900 Series HP 3000 Computer Systems

Migration Process Guide

HP Part No. 30367-90007
Printed in U.S.A. 1992

Fifth Edition
E0692

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 or fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for direct, indirect, special, incidental or consequential damages in connection with the furnishing 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.

This document contains proprietary information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Copyright © 1992 by Hewlett-Packard Company

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013. Rights for non-DoD U.S. Government Departments and agencies are as set forth in FAR 52.227-19 (c) (1,2).

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Restricted Rights Legend

Printing History

The following table lists the printings of this document, together with the respective release dates for each edition. The software version indicates the version of the software product at the time this document was issued. Many product releases do not require changes to the document. Therefore, do not expect a one-to-one correspondence between product releases and document editions.

Edition	Date	Software Version
First Edition	November 1987	A.01.00
Update 1	July 1988	A.10.00
Update 2	December 1988	A.20.00
Second Edition	October 1989	A.30.00
Third Edition	April 1990	A.40.00
Fourth Edition	December 1990	B.30.00
Fifth Edition	June 1992	B.40.00

Preface

MPE/iX, Multiprogramming Executive with Integrated POSIX, is the latest in a series of forward-compatible operating systems for the HP 3000 line of computers.

In HP documentation and in talking with HP 3000 users, you will encounter references to MPE XL, the direct predecessor of MPE/iX. MPE/iX is a superset of MPE XL. All programs written for MPE XL will run without change under MPE/iX. You can continue to use MPE XL system documentation, although it may not refer to features added to the operating system to support POSIX (for example, hierarchical directories).

Finally, you may encounter references to MPE V, which is the operating system for HP 3000s, not based on PA-RISC architecture. MPE V software can be run on the PA-RISC (Series 900) HP 3000s in what is known as *compatibility mode*.

Conventions

`UPPERCASE` In a syntax statement, commands and keywords are shown in uppercase characters. The characters must be entered in the order shown; however, you can enter the characters in either uppercase or lowercase. For example:

```
COMMAND
```

can be entered as any of the following:

```
command        Command        COMMAND
```

It cannot, however, be entered as:

```
comm           com_mand       comamnd
```

italics In a syntax statement or an example, a word in italics represents a parameter or argument that you must replace with the actual value. In the following example, you must replace *filename* with the name of the file:

`COMMAND` *filename*

bold italics In a syntax statement, a word in bold italics represents a parameter that you must replace with the actual value. In the following example, you must replace ***filename*** with the name of the file:

`COMMAND(`***filename***`)`

punctuation In a syntax statement, punctuation characters (other than brackets, braces, vertical bars, and ellipses) must be entered exactly as shown. In the following example, the parentheses and colon must be entered:

(*filename*):(*filename*)

underlining Within an example that contains interactive dialog, user input and user responses to prompts are indicated by underlining. In the following example, yes is the user's response to the prompt:

```
Do you want to continue? >>  yes
```

`{ }` In a syntax statement, braces enclose required elements. When several elements are stacked within braces, you must select one. In the following example, you must select either `ON` or `OFF`:

`COMMAND` $\left\{ \begin{array}{l} \texttt{ON} \\ \texttt{OFF} \end{array} \right\}$

`[ ]` In a syntax statement, brackets enclose optional elements. In the following example, `OPTION` can be omitted:

`COMMAND` *filename* `[OPTION]`

When several elements are stacked within brackets, you can select one or none of the elements. In the following example, you can select `OPTION` or *parameter* or neither. The elements cannot be repeated.

`COMMAND` *filename* $\left[\begin{array}{l} \texttt{OPTION} \\ \textit{parameter} \end{array} \right]$

Conventions (continued)

`[ ... ]`

In a syntax statement, horizontal ellipses enclosed in brackets indicate that you can repeatedly select the element(s) that appear within the immediately preceding pair of brackets or braces. In the example below, you can select *parameter* zero or more times. Each instance of *parameter* must be preceded by a comma:

`[,`*parameter*`][...]`

In the example below, you only use the comma as a delimiter if *parameter* is repeated; no comma is used before the first occurrence of *parameter*:

`[`*parameter*`][,...]`

`| ... |`

In a syntax statement, horizontal ellipses enclosed in vertical bars indicate that you can select more than one element within the immediately preceding pair of brackets or braces. However, each particular element can only be selected once. In the following example, you must select **`A, AB, BA`**, or **`B`**. The elements cannot be repeated.

$$\left\{ \begin{matrix} \texttt{A} \\ \texttt{B} \end{matrix} \right\} \texttt{| ... |}$$

`...`

In an example, horizontal or vertical ellipses indicate where portions of an example have been omitted.

`Δ`

In a syntax statement, the space symbol `Δ` shows a required blank. In the following example, *parameter* and *parameter* must be separated with a blank:

`(`*parameter*`)Δ(`*parameter*`)`

(key)

The symbol (key) indicates a key on the keyboard. For example, (RETURN) represents the carriage return key or (Shift) represents the shift key.

(CTRL)*character*

(CTRL)*character* indicates a control character. For example, (CTRL)**`Y`** means that you press the control key and the Y key simultaneously.

1

Introduction

This chapter introduces the migration stages. In order to help organize the activities involved in the migration process, a six-stage migration process is defined as follows:

1. Education
2. Analysis and planning
3. Preparation
4. Installation
5. Compatability Mode (CM) operation
6. Native Mode (NM) operation

These stages should be used as a guideline for migration. Stages can, and often will, overlap or be skipped altogether. Figure 1-1 summarizes the migration project stages.

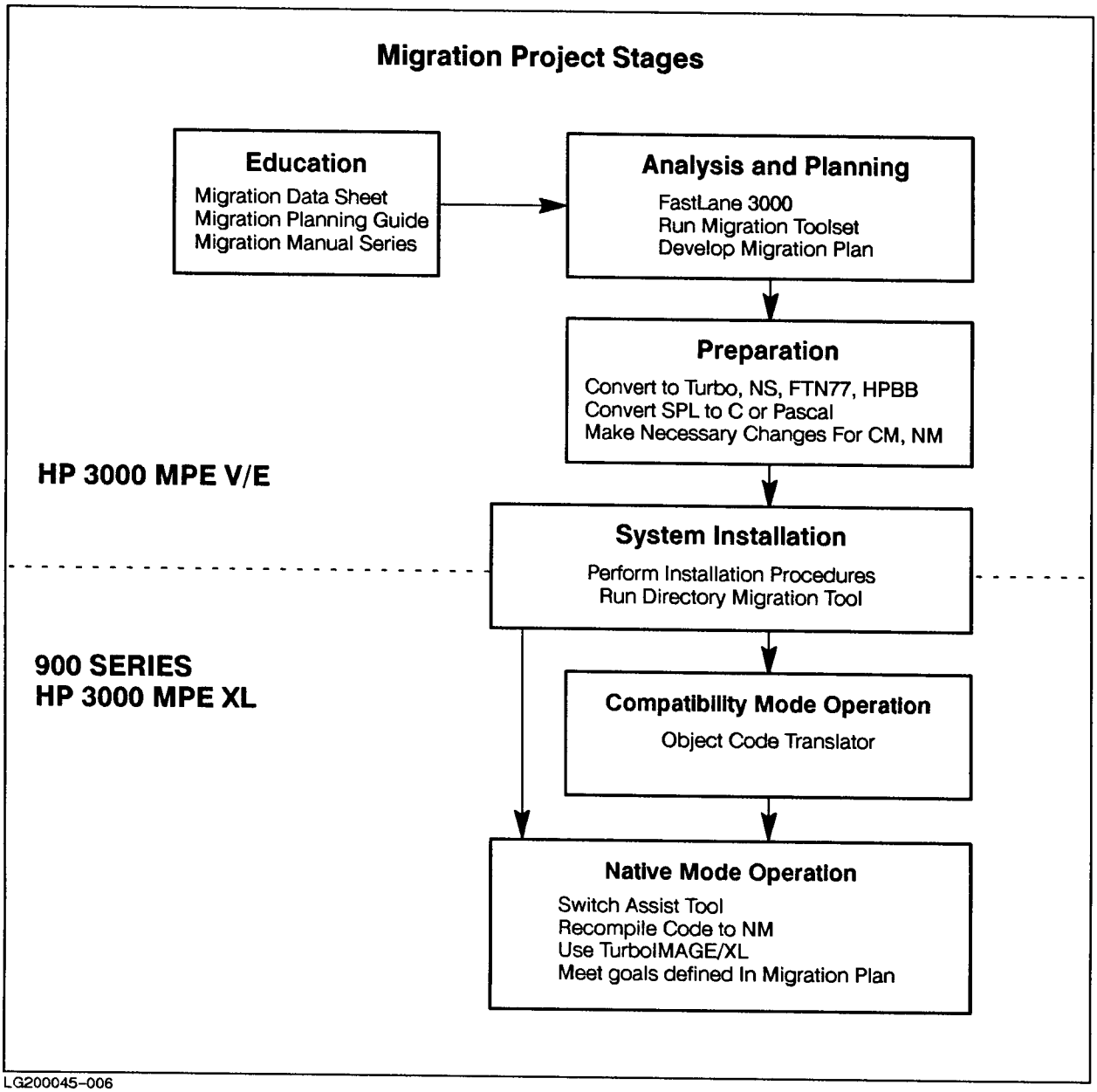


Figure 1-1. Migration Project Stages

Hewlett-Packard has provided solutions to ease the migration process at each stage: use the Migration Toolset during the analysis/planning stage, the Directory Migration Tool (DIRMIG) to migrate the operating environment during the installation stage, and the Switch Assist Tool (SWAT) to create mixed-mode applications during the migration to NM. You will find other migration documents and tools listed in Chapter 2.

Education

During education, the first stage of the migration process, the migration project team is selected. Also during this stage, they become familiar with MPE/iX, the generic migration process, and the documents, tools, and services that are available.

This manual is written to help you during this stage.

To migrate successfully your team must:

- Carefully analyze and formulate a realistic migration plan.
- Understand the benefits and trade-offs of the migration options.
- Take advantage of the ability to phase the migration process.

Analysis and Planning

During analysis and planning, the second stage of the migration process, the team develops the migration plan. Thorough analysis and planning is crucial to the success of migration.

Phased migration requires analysis of the existing application environment, consideration of MPE/iX functionality requirements and dependencies, and careful planning based on migration options available, time and resources required, and desired performance.

Analysis

The analysis process begins by taking a system-level view of the components to be migrated, and the functionality provided by the new MPE/iX system. The migration project team takes inventory of all applications and databases to be migrated. They must consider how much effort it takes to move each application.

The Hewlett-Packard Migration Toolset will be helpful during this stage of the migration process.

MPE/iX has been designed as a superset of MPE V/E to provide compatibility between the two systems, but there may be some incompatibilities. Any incompatibilities which may exist in each user-developed application should be identified by running the migration tools. Incompatibility issues may be identified in any of the following areas:

- Obsolete MPE V/E commands
- Intrinsics
- Subsystems
- Languages (SPL, in particular)
- Privileged Mode applications
- Data communications

You can estimate the migration effort needed by analyzing how many incompatibilities you find and how many changes are required to work around them. The analysis also determines the migration schedule and strategy you form for each application and database.

Planning

During planning, the team defines migration schedules and strategies for migrating each application and database. If application will be migrated in phases, the strategy describes those phases.

FastLane 3000 is a Hewlett-Packard tool for developing a migration plan.

MPE/iX has two environments, Compatability Mode (CM) and Native Mode (NM). The Switch subsystem enables mixed-mode operation, allowing applications to switch between the two environments.

Often, applications are moved to CM as an intermediate step, and then moved to NM in phases. CM is designed to provide code and data compatible to MPE V/E; it is usually easiest to migrate MPE V files to CM. NM is the native environment for MPE/iX; it is usually more productive to move to NM.

A migration strategy for a particular application could specify which portions of the application will execute in CM, NM, or mixed mode. For example, one phased migration plan might be:

1. Move all code and data to CM.
2. Test and verify that they meet or exceed performance expectations.
3. Recompile selected portions in NM with CM data alignment options.

4. Modify applications to access NM 32-bit data.

Preparation

During preparation, the third stage of the migration process, you update software to make migration easier. This stage may span the entire migration process. It can start before the 900 Series arrives.

Preparation involves updating to more recent versions of MPE software and HP products. For example:

- The MPE V/E-based system must be updated to U-MIT or later software prior to migrating to the 900 Series HP 3000.
- Databases must be converted from IMAGE/3000 to TurboIMAGE using the MPE V/E utility DBCONV.
- Data communications software should be upgraded from DS/3000 to NS3000/iX.
- All applications should be updated to the appropriate language version if NM performance is required. For example, FORTRAN/V applications should be updated to HP FORTRAN 77/V and BASIC/V applications should be updated to HP Business BASIC/V.

Any recoding necessary to remove incompatibilities found in the analysis and planning stage can begin in this preparation stage. The preparation stage should also include recompiling all source on MPE V/E to verify that the correct versions of source are available and used.

Installation

During installation, the fourth stage of the migration process, your 900 Series HP 3000 is installed and configured, and all programs and files are moved to the new machine. The objective is to have a functional MPE/iX-based system in place in order to begin the migration.

Use the Directory Migration Tool (DIRMIG) to migrate the MPE V/E operating environment.

The result of this stage is a 900 Series HP 3000 installed with the MPE/iX operating system and subsystems, configured with appropriate peripherals, and prepared for NM development.

Compatibility Mode Operation

During Compatibility Mode operation, the fifth stage of the migration process, object code files that ran on an MPE V/E-based system are run in CM on the MPE/iX system, and their results are validated with the originals.

It is suggested that you thoroughly test the application to ensure valid results.

Use the Object Code Translator (OCT) in this stage to gain additional performance for CM applications.

This stage allows for immediate productive use of the 900 Series HP 3000. Some migration plans may move directly to NM instead of using CM operation as an intermediate phase. The CM stage is recommended for users, however, so that program integrity can be verified before moving to NM. It may involve completing source code changes to remove CM incompatibilities found during analysis, thus easing NM conversion later.

Native Mode Operation

Native Mode operation is the sixth, and final, stage of the migration process. The objective of this stage is the implementation of the migration plan so that it meets your goals and performance expectations. The length of this stage depends on the scope of the migration plan, the strategy chosen, and the extent of changes that must be made to the application to meet performance goals and user expectations. Depending on the migration plan, this stage may overlap with the CM operation stage.

Activities included in this stage are:

- Recompiling source files using MPE/iX compilers.
- Determining what mixed-mode calls may be necessary for an application.
- Completing changes for NM incompatibilities found in the analysis and planning stage.

The results of this stage may include first developing applications in NM and CM, then using NM programs to manipulate MPE V/E-compatible data, and ultimately using NM programs to manipulate MPE/iX-compatible data.

In order to run in NM, programs developed on MPE V/E-based systems need to be recompiled with the MPE/iX version of compilers. Certain programs may require some source code changes to operate efficiently on the 900 Series HP 3000.

This last migration stage will most likely occur in phases, each one followed by a test period to verify the correctness of the application and an evaluation period to verify performance.

After a particular application has been validated under CM operation, the next phase may involve partial recompilation to NM using the Switch subsystem to access procedures in CM segmented libraries (SLs). SLs

can be accessed by applications executing in NM only through the Switch subsystem.

At the end of this phase, check performance of the application which uses Switch. If it does not met the goals defined in the migration plan, you may need to modify the plan to include an additional phase.

During this additional phase, you would convert the CM procedures to NM. Recompile CM procedures to NM if possible. If the CM procedures are implemented in a language not supported in NM you will have to rewrite them. The resultant NM object code can then be added to executable libraries (XLs) XLs in NM are the equivalent of SLs (Segmented Libraries) in CM. For more information about creating and maintaining executable libraries (XLs), refer to the *HP Link Editor/XL Reference Manual* (32650-90030). For more information about accessing XLs from CM programs, or accessing SLs from NM programs, refer to the *Switch Programming Guide* (32650-90014).

2

Education

This chapter discusses the first stage of migration: the education stage. It includes:

- "Migration Project Team", a definition of the project team you should form to complete migration.
- "System Overview", an introduction to the 900 Series HP 3000 hardware and software.
- "Migration Options and Solutions", a discussion to help you plan your migration methods.
- "Migration Tools and Documents", a list of Hewlett-Packard products available to help you migrate.

Migration Project Team

Before actually beginning the migration process, you should define the team of people assigned to complete the various tasks of migration. A well-balanced migration project team should consist of a project leader and representatives from:

- operations staff
- programming staff
- Hewlett-Packard staff
- users of major applications

Include one or more representatives from each area in your migration project team, depending on the focus and extent of your migration.

The most important keys to a successful migration effort is careful planning and a working knowledge of the existing applications and installations needs. The migration team must fully understand the migration options and the ramifications of migrating some applications and not others.

Project Leader Responsibilities

The project leader coordinates activities of the project team. To do this, the project leader must:

- Serve as a key contact person for Hewlett-Packard for FastLane 3000, the Hewlett-Packard consulting service that helps you migrate.
- Monitor the overall project.
- Maintain the Migration Project Notebook.
- Assign tasks and responsibilities to individual team members.
- Decide what documentation and training is needed and be sure it gets done.
- Schedule and conduct team meetings.
- Report progress to users and upper management.
- Resolve conflicts.

Project Team Responsibilities

The project team is responsible for project management. To do this, the project team must:

- Set the goal and scope of the project.
- Identify the key activities and milestones.
- Fevelop a detailed project schedule.
- Monitor the progress of assigned tasks.

System Overview

Before describing the options and issues involved in migrating to a 900 Series HP 3000 Computer System, it is important to understand the hardware and software components of the system.

The 900 Series HP 3000 family is based on HP Precision Architecture (HP-PA), an architecture founded on the concepts of RISC (Reduced Instruction Set Computer). The operating system for the 900 Series family is MPE/iX (MPE with extended large addressing).

HP Precision Architecture

HP-PA (Hewlett-Packard Precision Architecture) combines RISC principles with additional features to provide a flexible, expandable architecture and to further increase performance. Features include extended addressing (up to 64-bit virtual addressing), support for coprocessors and multiprocessors, and a memory-mapped I/O system.

RISC instructions are implemented directly in hardware to eliminate the system overhead associated with the microcode of conventional systems. Pipelining is enhanced by the uniformity of the HP-PA instructions. Improved performance also results from the memory hierarchy design of the new architecture. For a more detailed discussion of HP-PA, refer to the *General Information Manual* (5954-7418).

MPE/iX

MPE/iX is the operating system that runs on 900 Series HP 3000 computer systems. Improvements in hardware allow improvements in the operating system. Along with enhanced functionality and ease of use, it provides greater performance and capacity through a virtually addressed file system and higher availability of data.

MPE/iX is a superset of MPE V/E, the operating system that runs on the Series 37 through Series 70 HP 3000 computers. Although MPE/iX offers users a

superset of MPE V/E features, some incompatibilities exist due to fundamental differences between the systems software and hardware.

MPE/iX provides two environments (modes) for program execution: Compatibility Mode (CM) and Native Mode (NM). Program execution is not restricted to operating entirely in CM or NM; a program operating in one mode can call procedures that operate in the other. This switching between CM and NM is called mixed-mode operation.

An application or program executes in CM or NM with no noticeable differences to the user. Switching during mixed-mode operation is also transparent to the user.

Compatibility Mode

The CM environment emulates MPE V/E-based machine instructions on a 900 Series HP 3000 computer. This operating mode provides object code compatibility between MPE V/E-based systems and MPE/iX-based HP-PA machines; this means that programs that run on MPE V/E-based systems can run on 900 Series HP 3000 computers.

CM uses an emulator which functions as an object code interpreter. At run time, the emulator translates each MPE V/E machine instruction into a functionally equivalent sequence of HP-PA machine instructions. MPE V/E programs restored onto MPE/iX-based systems will run transparently under the emulator.

The MPE/iX environment also supports the translation of MPE V/E machine instructions through the Object Code Translator (OCT), a system utility. OCT accepts an MPE V/E program as input, performs an optimized translation of the object code into HP-PA machine instructions, and appends the translation to the end of the MPE V/E program.

If you run MPE V/E programs on MPE/iX, you will get better performance using OCT than you will using the emulator. With the emulator, instructions have to be fetched and decoded each time the program is run. You can cut overhead significantly by directly executing object code that OCT has already translated and appended to the program file.

The OCT translated file saves both the original program and the translation. This means the resulting file can be executed on both MPE V/E-based and MPE/iX-based systems. It is important to note, however, that it needs more disk space than the original file.

For either emulated or translated execution, MPE/iX preserves MPE V/E program and data structures and allows access to most MPE V/E callable intrinsics. The application is still subject to the same addressing limitations found on the MPE V/E-based HP 3000 computers.

Native Mode

NM is the natural operating mode of the 900 Series HP 3000 computer. NM allows users to utilize the features that the 900 Series HP 3000 offers, including higher performance and the expanded addressing capabilities of HP-PA. Programs developed and compiled with MPE/iX NM compilers automatically run in NM. Programs developed on MPE V/E-based systems need to be recompiled with the MPE/iX version of compilers in order to run in NM. The ability to recompile MPE V/E applications with MPE/iX compilers provides source code compatibility between the two systems.

Mixed Mode

MPE/iX uses the Switch subsystem to operate mixed-mode applications, applications that run in both CM and NM. Switch allows programs executing in

NM to call procedures which reside in CM Segmented Libraries (SLs). Switch also allows CM programs to call procedures located in NM Executable Libraries (XLs).

The Switch subsystem resolves the differences between NM and CM execution through a set of intrinsics that provide mixed-mode execution access. The switching is transparent to the user.

Because MPE/iX can mix modes of execution, applications to be migrated in phases, using CM as an intermediate step between MPE V/E and NM.

Phased Migration

You can move MPE V/E programs and data to NM over a period of time, using CM as an intermediate step. Moving everything to CM first is convenient in two ways. First, moving MPE V/E files to CM gives you immediate productivity on the 900 Series HP 3000 family; your applications work right away with little or no modification. Also, you may not be able to move files to NM quickly or easily because of such things as the size of applications, because you have object code but the original source code is lost, or because of language implementation differences.

You can recompile some parts to NM and leave others in CM, executing in mixed mode. MPE/iX uses the Switch subsystem moves between modes of execution, so it is an important part of phased migrations.

You can migrate database applications in phases, moving code and data separately. Move data directly to NM. Move code to CM; you can use it immediately, because MPE V object code and source code are compatible with CM. Recompile and move the code to NM part-by-part, in phases. Mixed mode operations will switch between NM and CM for you.

Figure 2-1 shows a phased migration.

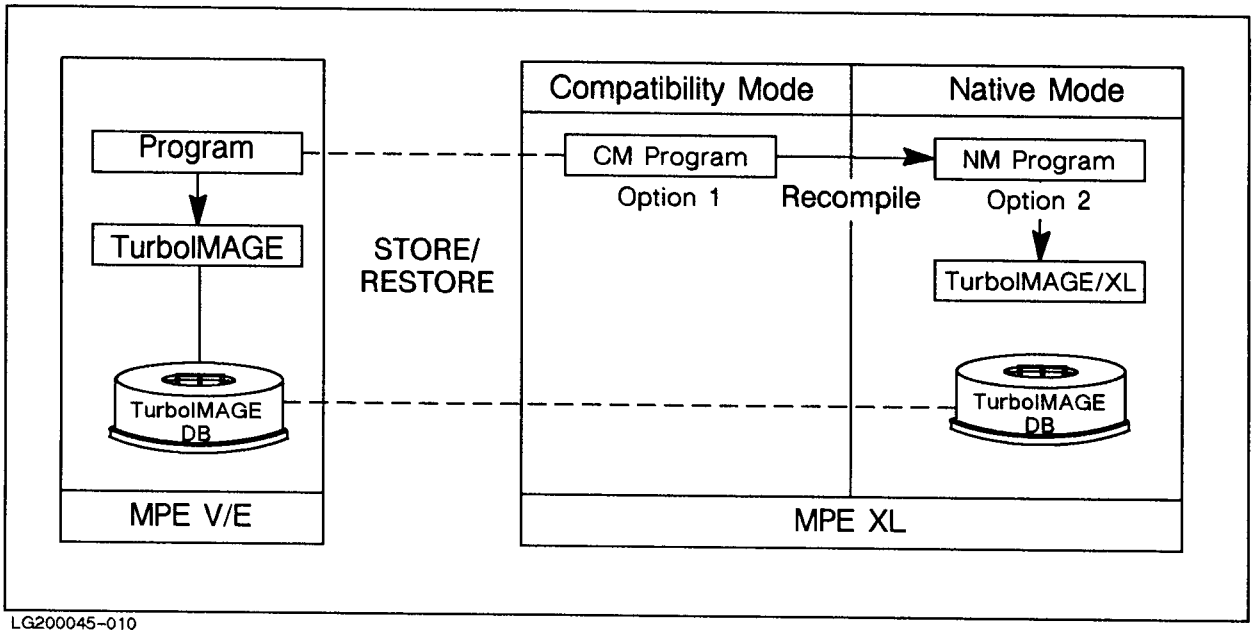


Figure 2-1. Summary of Phased Migration

Migration Options and Solutions

You will need to develop strategies to migrate your users' operating environment, data, programs, and databases. Hewlett-Packard has defined various migration options, created various tools, and outlined various strategies to help you to this successfully.

Migrating the Operating Environment

The user's operating environment contains:

- directory accounting structure.
- User Defined Command (UDC) structure.
- private volume configurations (volume sets and media).
- global Resource Identification Number (RIN) configurations.
- user logging identifiers.
- user files.

Hewlett-Packard recommends the use of the Hewlett-Packard supplied directory migration tool, DIRMIG, to migrate your MPE V/E operating environment to MPE/iX. Available on the MPE/iX installation tapes, DIRMIG is a flexible, one-step migration tool that migrates the MPE V/E operating environment from an MPE V/E SYSDUMP tape. Refer to Appendix G "Using the Directory Migration Tool" for details on DIRMIG operation.

Migrating Data

Data structures in MPE V are consistent with the data structures in CM on MPE/iX. However, there are two incompatabilities between CM and NM data structures that the migration team must look for:

- data alignment
- floating-point real numbers

Data Alignment

On MPE V/E-based systems and in CM, the natural alignment supported by the compilers is 16 bits. Data is aligned on 16-bit boundaries. In NM the natural alignment supported by the compilers and the architecture is 32 bits. The data is aligned on 32-bit boundaries.

Data ("flat") files generally contain byte streams of data. When the data is read by a program, the program interprets the data according to the program's internal data structures. The data structures are aligned by compilers, as defined by the program, using the system's natural alignment.

When MPE V/E programs are compiled to NM, the natural alignment changes from 16 bits to 32 bits. But the data in the files (the byte stream) does not change. What does change is the alignment of variables when CM programs are compiled to NM. Data accessed by NM programs may be interpreted inaccurately, even though the same data accessed by the same programs is interpreted correctly while running in CM.

For example, an MPE V/E-based program that depended upon a 16-bit alignment for its record structure may find that same record structure aligned on a 32-bit boundary once the program is compiled to NM.

The migration project team needs to identify all such possibilities before migrating to NM. Once identified, source code can be modified to retain data compatibility. For optimal performance, data files should be converted to the natural alignment for the 900 Series (32-bit).

You can use compiler options to minimize source modifications for data compatibility. Each NM compiler supports directives that select data compatibility with MPE V/E. For example, COBOL II/XL supports directives that align synchronized data either on 16-bit or 32-bit boundaries. HP Pascal/iX provides a compiler option that directs HP Pascal/XL to size and align data the same way as MPE V/E. HP FORTRAN 77/iX provides a similar compiler option.

Floating-point Representation

MPE V supports the HP 3000 format for representing floating-point real numbers. MPE/iX supports two

formats for representing floating-point real numbers: HP 3000 format is the default in CM, and IEEE format is the default in NM.

MPE V/E programs running in CM are not affected by the incompatability. For mixed-mode programs, select HP 3000 floating-point representation or IEEE floating-point representation by using the same compiler options and directives that are used to select data alignments. However, if full NM performance and precision is necessary, you need to convert all floating-point data to IEEE format.

Hewlett-Packard has provided an intrinsic, `HPFPCONVERT`, that will convert a value represented by HP 3000 floating-point format to a value represented by the IEEE floating-point format and vice versa.

Conversion of data files will affect program conversions. Thus, migration is an iterative process; once all programs are in NM, then all data should be converted to NM.

For complete information on data representation and floating-point conversion, consult the appropriate language migration manual and *Data Types Conversion Programmer's Guide* (32650-90015).

Migrating Data Communication Services

Data communication migration solutions apply to Hewlett-Packard manufactured networking subsystem. DS/3000 is not supported on MPE/iX. NS3000/iX is the supported data communication software on MPE/iX. NS3000/iX provides most of the features supported by NS3000/V.

Hewlett-Packard recommends two solutions in migrating software dependent upon data communications:

- Update the MPE V/E system from DS/3000 to NS3000/V; and convert all applications, job files,

and UDCs to use the NS3000/V subsystem. (All NS3000/V features are supported by NS3000/iX.)

- Migrate all applications, job files, and UDCs directly to MPE/iX and convert them to use the NS3000/iX subsystem.

Migrating Programs

The phased approach to migrating MPE V/E-based programs to NM over time allows your migration project team several options to achieve their goals. Hewlett-Packard has developed migration solutions for various system components: programs, operating environment, data files, and data communications. These solutions include:

- migration to CM.
- migration to NM.
- mixed mode applications.

Migrating database applications is a special case, and is discussed later in this chapter.

Migration to CM

CM is available to simplify the migration to the MPE/iX system, and to provide an intermediate stage, if needed, between MPE V and NM. CM offers an immediate migration path to MPE/iX-based systems and immediate productivity on the new system. CM program development is available in MPE/iX, but migration to CM should be considered the first stepping stone on the path to NM execution. The OCT may be used to increase performance with this solution.

This is extremely important in the cases where source code is not available for application programs.

Programs may need to be modified before they can execute on MPE/iX if they run in Privileged Mode or if incompatibilities have been identified. Most

non-Privileged Mode MPE V/E programs and data can be restored to an MPE/iX system and run in CM without the need for recompilation.

CM offers a variety of MPE V/E compatible compilers, such as COBOL II/V, HP FORTRAN 77/V, Pascal/V, BASIC/V, and HP Business ASIC/V. The MPE V Segmenter is available for building and maintaining program and library files for execution in CM on the 900 Series. After recompiling the application, it is suggested that you thoroughly test the application to ensure valid results.

Migration to Native Mode

To get the full benefit of the features of MPE/iX and HP-PA, CM programs need to be migrated to NM. This is achieved through source code recompilation with MPE/iX compilers. Before attempting recompilation, the migration project team should analyze the program for possible incompatibilities between CM and NM. Any incompatibilities should be resolved before recompilation. After recompiling the application, it is suggested that you thoroughly test the application to ensure valid results.

COBOL II, HP FORTRAN 77, and Pascal programs can be compiled into NM using the appropriate MPE/iX optimizing compilers: COBOL II/XL, HP FORTRAN 77/iX, and HP Pascal/iX, respectively.

NM employs LinkEditor to build and maintain program files, relocatable libraries, and executable libraries.

The NM compilers offer three levels of optimization, levels 0 through 2. Each incremental level provides a higher degree of execution performance. Optimization level 0 is the equivalent of no optimization, while level 2 provides full optimization. The optimization levels are invoked through compiler options and directives.

Hewlett-Packard recommends that programs be recompiled with level 0 optimization, tested, and subsequently recompiled with other optimization levels. Refer to the appropriate language reference manual for specific details on compiler optimization.

Migrating SPL programs to NM is more complex. Due to its dependence on the architecture of MPE V/E-based systems, an NM SPL compiler is not supported by Hewlett-Packard. Several solutions are available for SPL programs:

- Leave SPL programs in CM and maintain them with the SPL/V CM compiler available on MPE/iX.
- Translate SPL programs into HP-PA machine instructions using OCT. Note that while the newly translated program will not incur the overhead of the emulator, the program still adheres to the restrictions imposed on any MPE V/E object file running in CM.
- Rewrite SPL programs using a high-level language that can be compiled into NM; HP Pascal/iX or C/iX, for example.
- Leave the SPL routines in CM segmented libraries (SLs); migrate the program to NM and access the SPL routines from SLs via the Switch subsystem.

Mixed Mode Programs

Recompilation to NM may not be a viable alternative for all programs because source code may be missing, the program may be written in SPL, or the program may be written in another language for which an NM optimizing compiler is not available. The same holds true for segmented libraries (SLs).

You may want to operate in mixed-mode if programs are executed with group or account libraries, and the libraries cannot be migrated to NM. It is possible to migrate the program to NM and leave the libraries in

CM. The NM program, however, must access the SL procedures via the MPE/iX Switch subsystem because SLs are not directly accessible by programs executing in NM.

Switching between modes usually does not require source code modifications. In most cases, a program is recompiled to NM while the libraries accessed by the program remain in CM. A switch stub is required for every procedure that the program accesses from the CM SL. These switch stubs are placed in a NM library which is linked with the program. A switch stub is used to inform the Switch subsystem of the specifics of a procedure in a CM SL. You can create switch stubs either using the Switch Assist Tool (SWAT) or on your own, as discussed in the *Switch Programming Guide* (32650-90014).

Note that mixed mode program execution can also define a CM program accessing a routine in a NM executable library (XL). NM XLs (the NM equivalent to SLs), are not directly accessible to programs executing in CM, and must therefore be accessed through the Switch subsystem. Once again, neither the program making the call, nor the procedure being accessed, need to be modified.

Some overhead is incurred when using the Switch subsystem. The user needs to understand the trade-off between the performance implications of using the Switch subsystem and the effort necessary to recompile routines to NM. A mixed-mode program can be either an intermediate step to full NM operation or an alternative to rewriting or locating source code.

Mixed mode operation offers the highest degree of flexibility for the phased migration of programs. Even users who choose to recompile existing source code with the MPE/ iX version of compilers may need Switch if the source contains calls to user-written procedures

which still remain in CM SLs. Switch is also needed by developers of new NM applications that require the services of CM SL procedures.

If application size is a consideration, performance-sensitive portions of the CM application may be rewritten or recompiled in NM first, allowing CM programs to continue to call the procedures. Migrating critical portions of an application to NM while the remainder of the application executes in CM results in performance gains and allows for a continual, incremental upgrade path.

Phased Migration of Databases

MPE/iX supports two database management systems: TurboIMAGE/XL in NM and ALLBASE/SQL in NM.

Migrating ALLBASE

The MPE V version of ALLBASE is ALLBASE/SQL, a database management system that provides a relational database model called HP SQL. The MPE/iX version is ALLBASE/SQL.

Note

It is not possible to run the MPE/V version of ALLBASE/SQL in compatibility mode on an MPE/iX system. You must migrate to ALLBASE/XL.

Before migrating ALLBASE/SQL software and data from MPE/V to MPE/iX, you must UNLOAD EXTERNAL and STORE the data in all tables. After migrating, you must RESTORE the data, re-create all DBEnvironments, LOAD tables EXTERNAL, and then re-preprocess and recompile all application programs.

The following steps are recommended in migrating an ALLBASE/SQL DBEnvironment from MPE/V to MPE/iX:

1. On the MPE/V side, use the ISQL UNLOAD command with the EXTERNAL option to create ASCII file containing all data currently stored in each table in the DBEnvironment.
2. Copy the ASCII files and the description files created by the UNLOAD command over to the MPE/iX system. You can STORE and RESTORE the files, or you can use a disk copy utility.
3. Re-create the DBEnvironment on the MPE/iX system. Be sure to create tables with the same column descriptions as in the MPE V DBEnvironment.
4. On the MPE/iX side, use the ISQL LOAD command with the EXTERNAL option to load the newly created tables in to the MPE/iX DBEnvironment.
5. Copy all source files for application programs over to the MPE/iX system.
6. Re-preprocess and re-compile each application program.

Migrating TurboIMAGE

You can move data and applications from MPE V to MPE/iX separately. You can move applications from MPE V to CM or to NM. You can move applications from CM to NM in phases. These options give your migration project team has some flexibility in planning and executing the migration applications.

Whether you decide to move the applications to CM or to NM, the data files are taken from MPE V and restored directly to NM on MPE/iX. DBSTORE is used to store the database from the MPE V/E system and DBRESTOR may be used to restore the database onto the MPE/iX system.

Additional information on migrating TurboIMAGE/V databases can be found in Appendix H.

Option 1: CM. With this option, you move your data to TurboIMAGE/XL in NM, and your applications to CM. The CM programs rely on MPE V/E object code compatibility. In most cases, the programs and databases are restored from an MPE V/E STORE tape.

This option has several advantages:

- MPE V/E program files run with few or no modifications.
- The TurboIMAGE/XL database can be immediately accessed by the program.
- Users are immediately productive.
- No overhead is incurred for migration software.

The CM option is considered the first step in application migration to NM. While applications execute in CM, a migration project team can perform program migration to NM without impacting productivity.

TurboIMAGE/XL runs in NM, and its intrinsics are NM intrinsics. When your CM program calls a TurboIMAGE intrinsic, the process runs in mixed mode. TurboIMAGE/XL uses the Switch subsystem to invoke the CM intrinsics. You do not need to modify the intrinsic calls. Access to NM TurboIMAGE/XL intrinsics is transparent to the application because TurboIMAGE/XL intrinsic switch stubs exist in the MPE/iX CM System Library.

Option 2: NM. With this option, the application is recompiled with an NM compiler to run in NM. It accesses the TurboIMAGE database retaining the TurboIMAGE/XL intrinsic calls. The database remains in TurboIMAGE format and is accessed in NM by existing TurboIMAGE/XL intrinsics. This option offers the following advantages:

- The application's performance may improve over CM execution.
- The TurboIMAGE/XL intrinsic calls remain intact.
- The database is compatible with MPE V/E-based systems. Therefore, programs that access the same database can be systematically migrated to NM without impacting productivity of other CM programs accessing the database.
- Remote TurboIMAGE/XL access is preserved.

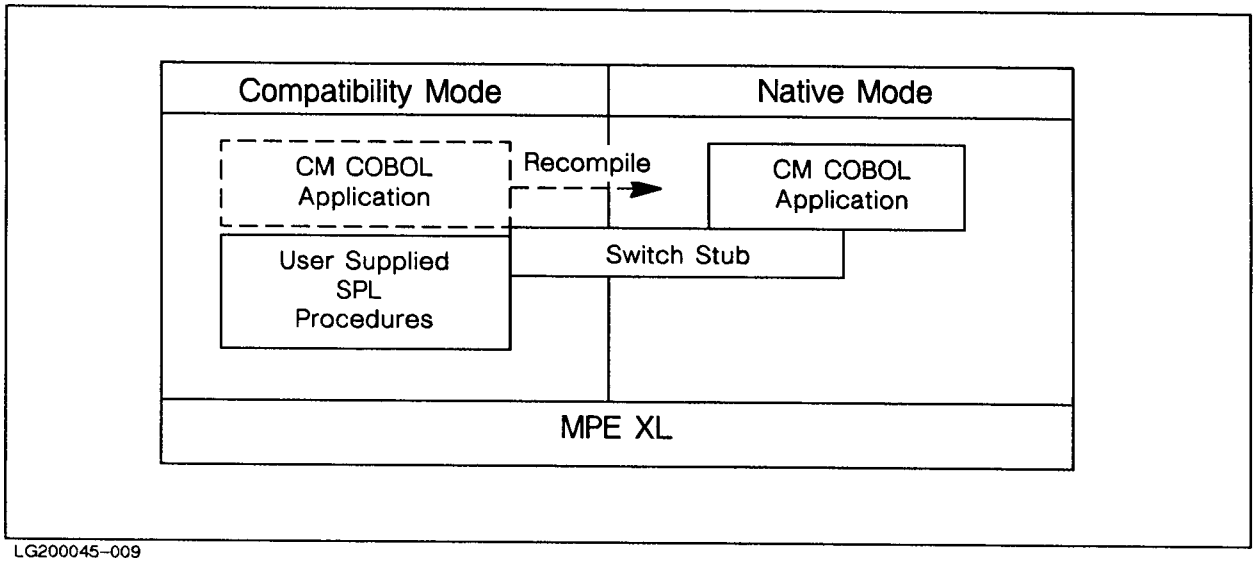


Figure 2-2. Option 2 with Switch

Migration Documentation, Training, and Tools

In addition to this manual, Hewlett-Packard has developed other manuals and tools to assist with various aspects of migration.

Migration Documents

- *General Information Manual* (5954-7418) provides a general overview of the 900 Series HP 3000 family.
- *HP 3000 Computer Systems System Configuration Guide* (5954-9354) provides current information on supported peripherals.
- *Introduction to MPE XL for MPE V Programmers* (30367-90005) updates programmers on topics that have changed from MPE V/E to MPE/iX and pertain to the MPE programming environment. This manual is useful when analyzing incompatibilities detected by the Migration Toolset.
- *FORTRAN 77/XL Migration Guide* (31501-90004) informs the FORTRAN 77 programmer of changes in the FORTRAN 77/iX environment.
- *HP COBOL II/XL Migration Guide* (31500-90004) informs the COBOL II programmer of changes in the COBOL II environment.
- *HP Pascal/XL Migration Guide* (31502-90004) informs the Pascal programmer of changes in the Pascal environment.
- *Introduction to MPE XL for MPE V System Administrators* (30367-90003) updates the System Administrator on topics that have changed from MPE V/E to MPE/iX that relate to their day-to-day activities.
- *MPE V to MPE XL: Getting Started Self-Paced Training* (30367-90002) updates the general user on topics that have changed from MPE V/E to MPE/iX that relate to their day-to-day activities.
- *Switch Programming Guide* (32650-90014) instructs the programmer on Switch stub generation.

Migration Training

FastLane 3000 is a Hewlett-Packard consulting service that helps you achieve more efficient use of time and resources as you migrate your application to the 900 Series HP 3000 system. It helps you understand how the migration process could impact your particular data processing environment, and what actions you can take to streamline that process.

FastLane 3000 is delivered to your migration project team by a specially trained Hewlett-Packard engineer and consists of two component services:

- System Planning builds the foundation of knowledge essential to migration analysis and planning.
- Application Planning guides your project team through the steps of developing a complete migration plan for one of your applications.

Migration Tools

HP has developed the following tools to help with migration:

- Migration Toolset (Version A.00.03) is three tools that find possible incompatibilities between MPE V/E and MPE/iX: Migration Planning Tool (MPT), Object Code Analyzer (OCA), and Run Time Monitor (RTM).
  - MPT, available on MPE V/E-based systems, provides an initial look at an MPE V/E system by generating reports, from a system-level down to a file detail level, examining incompatibilities and MPE/iX disk space projections for MPE V/E files. Operation of MPT is discussed in Appendix I, "Using the Migration Planning Tool".
  - OCA, available on both MPE V/E-based and MPE/iX-based systems, uses MPE V/E object code files as input and generates a list of possible incompatibilities between MPE V/E and MPE/iX

as output. Operation of OCA is discussed in Appendix B, "Using the Object Code Analyzer".

- RTM, available on MPE V/E-based systems, provides customers with a list of run-time incompatibilities by monitoring applications as they execute on an MPE V/E-based system and logging incompatibilities as they occur. Operation of RTM is discussed in Appendix C, "Using the Run Time Monitor".

- Directory Migration Tool (DIRMIG) is an MPE/iX tool that migrates the operating environment. Operation of DIRMIG is discussed in Appendix G, "Using the Directory Migration Tool".
- Switch Assist Tool (SWAT) is an MPE/iX tool that facilitates phased application migration from CM to NM and builds switch stubs to allow mixed mode execution of applications. Operation of SWAT is discussed in *Switch Programming Guide* (32650-90014).
- The Object Code Translator (OCT) is an MPE/iX tool that performs an optimized translation of MPE V/E-based object code to MPE/iX-based object code.
- Language conversion utilities are available to change applications from FORTRAN 66/V to HP FORTRAN 77/V and from BASIC/V to HP Business BASIC/V. The appropriate conversion utility is supplied with HP FORTRAN 77/V, HP FORTRAN 77/iX, and HP Business BASIC/V.
- A data alignment compiler option is available with all MPE/iX NM compilers, so that data can be aligned on 16-bit or 32-bit boundaries for access from CM or NM programs, respectively.
- `HPFPCONVERT` is an MPE/iX intrinsic that converts between HP 3000 and IEEE floating-point representations.

- DBCONV is a utility that will convert your IMAGE/3000 database to a TurboIMAGE database.
- SDCONV is a program that converts and loads data from Dictionary/V to HP System Dictionary/iX.

3

Analysis and Planning

This chapter discusses the second stage of migration: analysis and planning. During this stage, you gather and analyze the applications to be migrated so that you know how much time and effort you will need to migrate. At the end of this stage you will have your migration plan based on this analysis. This phase, then, involves two activities :

- Analyzing the applications
- Planning the migration

Careful and thorough analysis will ensure a successful migration plan. When analyzing an application to be migrated, consider the following:

- Which applications will migrate to an MPE/iX-based system?
- Does the application have any incompatibilities?
- What are the characteristics of the application?
- How much time do you have to complete the migration?
- What resources are needed to complete the migration?
- What are the migration goals for the application?

Detecting Incompatibilities

Use the Migration Toolset during the analysis and planning stage. Using the tools, you can detect most migration-related problems in MPE V/E files before you migrate them to an MPE/iX-based system.

Some problems cannot be detected by the Migration Toolset. They can only be determined by someone who has a working knowledge of what the application does and how it does it. These problems are summarized in Appendix D, "Incompatibilities".

The Migration Toolset, discussed later in this chapter, has three tools:

- MPT, Migration Planning Tool,
- OCA, Object Code Analyzer,
- RTM, Run Time Monitor,

The Migration Toolset detects and reports on incompatibilities found in two types of files:

- Program and segmented library (SL) files containing object code.
- Job stream files and UDC files.

Appendix D lists known incompatibilities between MPE V/E and MPE/iX including those incompatibilities detected by the Migration Toolset.

Searching Program and SL Files

The incompatibilities found in program and SL (segmented library) files are generally caused by the program or SL calling certain external procedures (or system intrinsics) or by the use of Privileged Mode capability. The Migration Toolset uses two methods of detecting incompatibilities found in object code:

- Static search through files containing object code
- Dynamic search during program execution

Using the static search method, program and SL files are searched for instances of calls to external procedures and system intrinsics that may cause problems when the object code is executed on an MPE/iX-based system (either in CM or when recompiled in NM). This method reports an incompatibility whether or not the call is executed at run time.

Two migration tools, MPT and OCA, report on incompatibilities found using the static search method of detection.

Using the dynamic search method, applications are monitored during execution on an MPE V/E-based system for occurrences of calls to MPE V/E intrinsics and programmatically executable commands which are either not supported, or have modified parameters or return values in MPE/iX. This method reports an incompatibility only when the problem code is executed in a manner that would cause a problem when executed on an MPE/iX-based system.

RTM reports on incompatibilities found using the dynamic search method of detection. In addition, the MPT is able to access the information gathered by the RTM and include the information in its own reports.

Searching Job Stream Files and UDC Files

Incompatibilities found in job stream files and UDC (user-defined command) files are caused by instances in the file of calls to incompatible or modified commands. Using the static method of search, job stream and UDC files are searched line-by-line for instances of calls to commands that may cause problems when the file is used on an MPE/iX-based system. MPT detects and reports on incompatibilities found in job stream and UDC files.

Migration Planning Tool

MPT provides a first look at the potential issues arising from migrating files from an MPE V/E-based computer system to an MPE/iX-based computer system. MPT is not intended to be all-encompassing, but rather a vehicle for providing a sense of direction to the migration process.

MPT is easy to use by inexperienced programmers and analysts, yet provides in-depth detail for more sophisticated users. MPT can combine the information compiled by RTM into the final MPT reports (when RTM is run prior to creating the MPT database). MPT is also used to identify those program and segmented library files that require further analysis by OCA.

MPT is a package of programs and parameter files that you use to generate the following:

1. a database containing comprehensive information about files on your MPE V/E-based HP 3000 Computer System.
2. migration planning reports (up to six levels) from information collected from the database. The reports contain information on:
3. a report listing the Hewlett-Packard software products that are fully or partially installed on your MPE V/E-based system.
   - estimates of disk capacity for files migrating from an MPE V/E-based system to an MPE/iX-based system.
   - places in files where problems may be encountered after the files are migrated to an MPE/iX-based system.

The MPT package runs entirely on an MPE V/E-based HP 3000 Computer System in either batch mode or interactive mode. The MPT package does not deal with the implications of adding new applications nor any

consequences related to system performance. Its purpose is to assess disk capacity requirements and to uncover incompatibilities that might affect the migration effort.

Refer to Appendix I, "Using the Migration Planning Tool" for detailed information about MPT.

Object Code Analyzer

OCA is a utility to aid in migrating application programs from an MPE V/E-based HP 3000 (running U-MIT or later) to an MPE/iX-based HP 3000. OCA reports definite problem areas and warns of potential problems that may be encountered when migrating an application to an MPE/iX-based system. The OCA can scan program and SL files on MPE V/E-based HP 3000 systems in either interactive or batch mode. OCA will be able to locate any potential incompatibilities with only one pass.

Using OCA, you can:

- locate program and SL files to be analyzed.
- identify incompatibilities independent of the source language.
- find all occurrences of a potential incompatibility. OCA does not depend on run-time execution.
- Identify the location of incompatibilities in code by reporting the code segment and the offset within the segment.
- Determine which SL files are required to migrate an application.
- Look for calls to uncallable entry points to the MPE V/E operating system.
- Look for calls to any user-selected external procedure.

OCA can be used to provide code segment and offset information when an incompatibility is found. You can

then determine the procedure and the offset within the procedure by using a `PMAP` to trace an incompatibility back to a location in the source code. This information simplifies the task of identifying the exact location of the incompatibility. By examining each of these locations you can determine if a true incompatibility exists.

Refer to Appendix B, "Using the Object Code Analyzer" for detailed information about the OCA.

Run Time Monitor

RTM is a data capture facility you use at run time to identify incompatible areas within MPE V/E-based applications. The following events are logged

- calls to MPE V/E intrinsics
- programmatically executable commands which are not supported, or which have modified parameters or return values in MPE/iX.

When a user program or library procedure makes a call to system software to invoke an incompatible feature, the event and location of the call is logged. You can select from predefined classes of events to detect with the program `RTMSYS`. A report program, `RTMREP`, provides summaries according to program and event, in addition to a record-level detailed report. MPT can access the data captured by RTM and include RTM data in its own reports on incompatibilities.

You should run RTM long enough to have a complete activity cycle monitored. For instance, if you want to determine if an accounting application has any run-time incompatibilities, you will need to have RTM enabled long enough to monitor not only end-of-week or end-of-month activities, but also end-of-quarter or end-of-year activities to ensure that all portions of the application program paths are properly exercised and monitored.

Using RTM, you can:

- Identify incompatibilities independent of the source language.
- Trap all events from all library and program paths executed.
- Trace an event to the code segment and offset in code. Use this information, with a PMAP and source listing, to locate the incompatibility.
- Flag a potential problem in cases where return codes or values have been modified for MPE/iX. However, no incompatibility exists if your program does not depend on the modified return value.

Refer to Appendix C, "Using Run Time Monitor" for detailed information about RTM.

Migration Toolset: A Comparison Between the Tools

There are some inherent differences between MPT, OCA, and RTM. While one tool may be better suited for detecting a particular type of event, it is recommended that you use all three tools to get an accurate picture of your potential incompatibilities.

MPT is best suited for:.

- Getting an initial approximation of the complexity of the migration effort.
- Summarizing migration issues at a system level, account level, group level, and file level.
- Generating reports that incorporate the results of both static search methods and dynamic search methods.
- Projecting MPE/iX disk space requirements for MPE V/E files.
- Identifying Hewlett-Packard products installed on the MPE V/E-based system.

- Reporting on incompatibilities found in job stream files and UDC files as well as in program and SL files.

OCA is best suited for:.

- Analyzing all programs and SL files to be migrated to a 900 series HP 3000.
- Detecting incompatibilities in code paths that may not have been executed while RTM was enabled.
- Locating calls to user-selected external procedures.
- Quickly determining potential incompatibilities.
- Detecting calls to unsupported routines.

RTM is best suited for:.

- Investigating a particular class of events based on an OCA report.
- Detecting incompatibilities that can only be determined at run time.
- Analyzing applications on a long-term basis.
- Obtaining information on the frequency of particular incompatibilities.

Analyzing the Reports

Once potential incompatabilities are listed, use the information to play your migration strategy and identify any changes that you need to make in your applications.

Look in Appendix D for information listed under "Undetected Incompatibilities" and under "Incompatibilities", which lists the incompatibilities detected by the MPT, OCA and RTM. While this list of possible incompatibilities appears long, most files (either program, SL, job stream, or UDC files) contain only a few, if any, of these. Also included is the cause of the incompatibility and any correction for the problem.

Potential incompatibilities reported by MPT and OCA can fall into one of four categories. These categories relate to whether the event was executed while RTM was in effect and if the event detected by MPT or by OCA is a real incompatibility.

- An incompatible event did not execute while RTM was enabled.
- A compatible event did not execute while RTM was enabled.
- An incompatible event executed, and was detected and logged by RTM.
- A compatible event executed, and was detected but not logged by RTM.

Consider the example where OCA detects a `COMMAND` intrinsic call, event 200, but is unable to determine if the command executed in the call is incompatible. RTM reports on the call only if it is executed with an incompatible command. Since a different command may be executed at different times by the same `COMMAND` intrinsic call, what is compatible at one time may not be compatible at another.

Thus, the fact that OCA detected a possible incompatibility with the `COMMAND` intrinsic, event 200, and that RTM did not detect a problem with the `COMMAND` intrinsic, events 201 through 218, does not necessarily mean that the `COMMAND` call detected by OCA is not problematic with respect to events 201 through 218.

Careful inspection of the code in question may be necessary to resolve all potential incompatibilities. When analyzing the reports generated by MPT, OCA, and RTM, take into account the limitations of each tool in detecting incompatibilities.

Your migration strategy should be influenced by the number and complexity of the changes you need to make to your application to have it function properly on MPE/iX-based systems. Also, you should note the changes which are necessary to run in CM or NM, as this, too, influences your migration strategy. A thorough understanding of the application may be necessary to assess the impact of the incompatibility. Once all necessary changes are identified, and the migration plan is developed, you can estimate the resources needed to complete your migration.

Severity of Incompatibilities

Once you have determined what potential incompatibilities exist in your application, you need to determine how your application depends on the item in question. In some cases, what is flagged as an incompatibility may not cause your application to run incorrectly but it is something that you should investigate to determine the impact on your application.

To determine accurately what, if any, modifications your application will require to run on the 900 Series HP 3000, you have to decide whether the application will be run in CM or NM. Additionally, you need to know whether it needs to call NM or CM procedures.

The most severe incompatibilities include the use of Privileged Mode, uncallable routines, or undocumented entry points. Resolution of these potential incompatibilities will require a detailed study of the application in question.

Application Characteristics

Your application may include more than just the program file. A complete application may consist of programs, SLs, data files, job streams, and UDCs. The application may have other characteristics that will affect your migration strategy such as:

- The Hewlett-Packard languages used by the application.
- The number of lines of code in each program in the application.
- If the source code is available for all the programs in the application.
- Whether or not the application relies on Hewlett-Packard or third party software.
- The MPE V/E features that the application uses.

- How the application uses MPE intrinsics.
- How the application accesses data.
- Whether or not the application encounters storage or performance limits.
- Whether or not the application interfaces with another application.
- What related applications are being migrated.
- What the application's MPE/iX disk space requirements are when migrated to an MPE/iX-based system.

Migration Goals

After analyzing the incompatibilities and characteristics of the application, you need to establish migration goals. These goals should define the intended mode of operation for the application, and what data alignment is expected by the application.

You may also want to identify interim goals. For instance, if you have an application that you ultimately want to migrate to NM, but it shares data with another application which will not be migrated for some time, an interim migration goal might be CM or mixed mode, until the second application can be migrated to NM.

To accurately determine the goals for migrating the application you need to consider the following:

- What languages are available in the desired mode?
- Is the source code available for making changes and recompiling?
- What incompatibilities need to be resolved?
- Does going to NM increase the number of incompatibilities?
- What mode do dependencies work in?

- What data format is required?
- Is all required Hewlett-Packard or third party software (or equivalent) available for use on the MPE/iX-based system?
- Do you need to share files with MPE V/E systems in a network?
- Will source and program files be sharable across both MPE V/E-based machines and MPE/iX-based machines?

Time and Resources

Once you have established the migration goals, you will need to determine what time and resources you need to commit to completing those goals. Time and resource requirements needed may be influenced by the following factors:

- The number of lines of code in the application.
- Availability of source code.
- Tools used in software development.
- Activities needed to properly exercise all code paths for RTM.
- The needs of the application user group.
- The length of time both machines will be available to complete parallel testing.

Planning the Migration

Now that the analysis is complete, and you have developed a strategy for migrating your application, it is time to develop the migration plan.

Your complete migration plan should include:

- Purpose and scope
- Project identification

- Migration strategy
- Migration schedule
- Resource requirements
- Documentation and training requirements
- Detailed application analysis
- Test plan

Purpose and Scope

The scope should include all areas addressed by the plan. The purpose is a statement of how you will use the migration plan.

Project Identification

Project identification includes both customer and application identification. Application identification is a brief description of the application covered by this migration plan.

Migration Strategy

The migration strategy includes a statement about:

- migration goals.
- target state.
- major milestones.
- project completion criteria.
- project team members.
- support staff.
- your Hewlett-Packard support team.

If the application has no incompatibilities, you may be a ble to move the application directly to NM. If however, your application has some incompatibilities, then your

migration strategy will be slightly more complex. You may need diferent migration strategies for different applications. Hewlett-Packard gives you flexibility; you decide which migration options to choose for your applications.

A migration goal is a statement describing the functionality and performance expectations, as well as a timeframe for completion.

The target state describes the operating mode and options for all programs and databases that make up the application. The target state should also specify which portions of the application will be in CM, NM, or mixed mode with Switch stubs. Major milestones list each major step involved in reaching the target state. These milestones will be the basis for the migration schedule.

Project completion criteria lists the specific criteria for project completion.

The migration strategy should also list the project team members, support staff, and Hewlett-Packard personnel involved.

Migration Schedule

The migration schedule should represent milestones and durations. For example, a graph where the vertical axis lists milestones and the horizontal axis lists dates pertaining to duration and completion.

Your milestones should include: detailed analysis, migration plan, software preparation, installation, CM testing, recompilation to NM, database conversions, NM testing, and project completion. Include intermediate milestones.

Resource Requirements

Resource requirements should list the hardware and software requirements for completing migration.

Hardware requirements list the 900 Series, disk drives, tape drives, printers, terminals, and any other hardware needed for migration. Software requirements list MPE/iX with version number, compilers, subsystems, tools, database management systems, and any other software needed for this project.

Documentation and Training Requirements

Documentation and training requirements should list the training and documentation needs for all personnel involved in the migration project.

Under documentation needs, list migration guides and reference manuals. Indicate how many copies of each are needed and when they are needed. Be sure to plan time for individuals to read and study the documentation. Other documentation considerations include updating in-house documentation that needs to be changed due to migration.

Under training needs, list all training courses, dates, and who will attend.

Detailed Application Analysis

This is the seventh section of your migration plan. A detailed analysis of every application being migrated includes:

- application organization.
- languages used.
- number of lines of code.
- system dependencies.
- user interfaces.
- networking.
- database access.

- third party software.
- contributed library dependencies.
- known incompatibilities.
- potential problem areas.
- list of intrinsics and external references.

A procedure flowchart or call map can be very useful in describing the application. Reports from the Migration Toolset (MPT, OCA, and RTM) should be included here.

Test Plan

This is the eighth and last section of your migration plan. The test plan should include the test data and procedures that will be used at each stage of the migration.

A complete test plan ensures that most problems are avoided before the application is in full use and should ensure that any change made to the application is thoroughly tested.

4

Preparation

Preparation readies the existing MPE V/E system for migration. Software updates can be done before the 900 Series hardware arrives. This involves updating to U-MIT or later software, including databases, data communications, and languages.

Compiler Conversions

Most MPE V/E applications will run in CM on MPE/iX without recompilation. Simply restoring the application onto an MPE /iX-based system enables you to immediately execute the application in CM.

You may choose to continue development and maintenance of an MPE V/E application after its migration to an MPE/iX-based system. Many MPE V/E language compilers are available also on MPE/iX in CM to provide support for migrated MPE V/E applications. Because there is no COBOL 68 compiler available on an MPE/iX-based system, COBOL 68 applications must first be converted to COBOL II/V. Refer to the *COBOL II/3000 Conversion Guide (32233-90005)*.

Recompiling an application using NM compilers can provide increased performance and greater access to features offered by HP-PA, although modifications to the migrated source code may be necessary. To determine if there are any differences between the MPE V/E implementation of a language compiler and the MPE/iX implementation (either CM or NM),

review the appropriate language migration guide (listed under "Documentation and Training" in Chapter 1, "Introduction and Migration Overview").

Subsystem Changes

Convert all databases to TurboIMAGE before migrating to the 900 Series HP 3000.

Convert from DS/3000 to NS3000. NS3000/XL is the data communications system for the 900 Series HP 3000. To convert DS/3000 to NS3000, examine job streams and UDCs for DS/3000 control commands that may not work on NS3000, and make the appropriate changes.

Application Changes

Decide whether to move the application to CM. Examine Migration Toolset report output for CM incompatibilities. Determine when to convert the application to NM. Then, examine Migration Toolset report output for CM/NM incompatibilities.

The use of undocumented intrinsics, nonintrinsic SPL procedures, and other procedures that remain in CM that are called from high-level languages, require Switch stubs. Examine cross-mode calls for performance trade-offs. For more information on creating Switch stubs, refer to *Switch Programming Guide (32650-90014)*.

Preparing the HP 3000 MPE V/E System for Migration

DIRMIG allows the system manager to duplicate the MPE V/E operating environment on the MPE/iX system. The operating environment consists of account structure information, including the UDC environment, private volume information, and system tables information. DIRMIG runs on your MPE/iX-based system and takes as input your MPE V/E SYSDUMP tape. Before using an MPE V/E SYSDUMP tape and DIRMIG to migrate your operating environment, please review the following checklists.

Accounting Structure

- Certain MPE/iX accounts and groups are reserved for system use. Check for accounts and groups on the MPE V/E system that are reserved on the MPE/iX system (`DIAG.SYS`, `CONFIG.SYS`, `MPE XL.SYS`, `CONFG950.SYS`, `CONFG925.SYS`). DIRMIG will not migrate these groups and accounts.
- Use `:LISTFILE` or `:LISTF`to check for files that exist on both the MPE V/E system and the MPE/iX system. DIRMIG will not restore user files that already exist on MPE/iX. Purge duplicate files, or create groups and accounts on the MPE V/E system to accommodate duplicate files, if needed. Move duplicate files into new groups and accounts on the MPE V/E system.

User-Logging ID Table

On the MPE V/E system:

- Use `:LISTLOG` to determine which user-logging IDs to migrate.
- Use `:RELLOG` to release user-logging identifiers that should not be migrated or which are no longer in use.
- Define user-logging parameters via SYSDUMP. This includes the number of user-logging processes and the number of users per logging process.

UDCs and User Files

Disable any system-level, account-level, and user-level UDCs that users do not want to migrate. Run the GETUDC utility on the resultant MPE V/E system. GETUDC aids in the migration of the MPE V/E UDC environment by creating a text file containing the names of all system-recognized UDC files.

Refer to Appendix F, "Using the GETUDC Utility" for details on using GETUDC.

The GETUDC-generated file should be supplied to SYSDUMP's `ENTER DUMP FILE SUBSET` prompt so that all UDC files are stored at the beginning of the STORE portion of the SYSDUMP tape. Placement of UDC files at the beginning of the SYSDUMP tape greatly reduces the amount of time DIRMIG takes to search for and restore these files.

Create a SYSDUMP Tape

DIRMIG requires an MPE V/E SYSDUMP tape as input. Accomplish these tasks when running SYSDUMP:

- Delete unused global RINs that should not be migrated. Configure RIN parameters.
- Define user-logging parameters. Configure appropriate use-logging ID parameters, including the number of user-logging processes and the number of users per logging process.
- Enter the name of the GETUDC-generated text file at the SYSDUMP prompt `ENTER DUMP FILE SUBSETS`.

Note

A non-carriage-return full backup is recommended.

Preparing to Perform Migration

Ensure that all necessary components are available before continuing.

A Hewlett-Packard representative should have installed and configured the 900 Series HP 3000. Check to be sure that the system is up and loaded.

Note

Please refer to the *HP 3000 Computer Systems System Configuration Guide (5954-9354)* for current information on supported peripherals.

Software Requirements

An MPE/iX load tape, and a SYSDUMP tape from an MPE V/E-based system are required to complete system migration.

5

System Installation

During System Installation, the migration team completes the installation, initial system test, and configuration of the 900 Series HP 3000.

The team uses `DIRMIG.PUB.SYS` to migrate most of the MPE V/E operating environment to MPE/iX.

Tasks performed during the installation stage include:

- Installing MPE/iX (*System Startup Configuration and Shutdown (32650-90034)*).
- Configuring the MPE/iX system (*System Configuration (32650-90042)*).
- Completing System Migration (Appendix G of this manual).
- Installing HP subsystems with AUTOINST/XL (*MPE/iX Installation and Update Manual (36123-90001)*).

Directory Migration Tool (DIRMIG)

DIRMIG migrates the MPE V/E operating environment to MPE/iX. The operating environment includes:

- System tables information
- Global Resource Identification Numbers (RINs)
- User-logging identifiers
- Directory structure/user files
- UDC environment
- Private volume information

DIRMIG uses an MPE V/E SYSDUMP tape as input.

DIRMIG is menu-driven and provides the user with an interactive means of selecting any combination of MPE V/E components to migrate:

- Global RINs
- User-logging identifiers
- The directory, including UDC/user files
- Private volume information

DIRMIG features partial directory migration and the choice of overriding or not overriding existing directory information on the MPE/iX system with MPE V/E directory information, if accounts conflict. This permits directory and UDC information from multiple MPE V/E systems to be merged on one MPE/iX system.

DIRMIG also has the ability to detect MPE V/E directory corruption.

DIRMIG restores user files with the KEEP option specified. This prevents the user from restoring over files installed from the MPE/iX installation tape. The user has the option of choosing which file sets to restore. A log file, `DIRLOG`*nn*`.PUB.SYS`, is created each time DIRMIG is run, to track the progress of migration. The

logging facility records the migration status of each component and provides detailed information in the event that any errors occur.

To assist with private volume migration, DIRMIG creates three other files: `PVASSIST.PUB.SYS`, `PVSUMARY.PUB.SYS`, and VOLUTIL command files as specified by the user. `PVASSIST` contains private volume information for the accounts DIRMIG migrates. `PVSUMARY` is a VOLUTIL command file generated from all the information contained in `PVASSIST` to initialize media with MPE V/E directory information. The volume utility, `VOLUTIL.PUB.SYS`, allows the user to create, modify, and delete volumes in MPE/iX. Refer to Appendix G, "Using the Directory Migration Tool" for details on completing private volume migration using both DIRMIG and VOLUTIL. Refer to *Volume Management (32650-90045)* for complete documentation on VOLUTIL.

`PVSUMARY` must be modified to make it usable by VOLUTIL and the VOLUTIL command files. The user must:

- Provide logical device information for media initialization.
- Modify member and class information for volume sets, if any changes are desired.
- Delete/modify spanning information.

These modifications can be made using any HP 3000 standard text editor, such as EDIT/V.

To use DIRMIG:

1. Log on as `MANAGER.SYS,PUB`.
2. Mount the MPE V/E SYSDUMP tape.
3. Enter `:RUN DIRMIG.PUB.SYS`.

The main menu offers several options:

- Complete migration with no dialog
- Complete migration with dialog
- Migration of global RINs
- Migration of user-logging IDs
- Directory migration
- Private volume environment migration

The dialog option of complete migration displays more detailed menus at each step of the process. These detailed menus allow the user to continue without migrating a particular component, to display information from the MPE V/E SYSDUMP tape, and to use the HELP feature.

The step-by-step procedure for using DIRMIG is located in Appendix G, "Using the Directory Migration Tool". Appendix G also contains detailed procedures for using the MPE/iX VOLUTIL utility to complete private volume migration.

6

Compatibility Mode Operation

This is the fifth stage of migration. The objective of this stage is to ensure that programs that run successfully on MPE V/E-based HP 3000 systems also run successfully on MPE/iX-based HP 3000 systems in CM. The result of this stage is a set of CM applications whose results have been validated with the original versions. Activities of this stage include:

- Completing the changes identified in the analysis and planning stage necessary for operation in CM.
- Verifying program integrity after changes have been made.
- Translating code with OCT.

Compatibility Mode

CM provides most of the functions available to programmers in MPE V/E-based systems. Additionally, CM programs can access NM data files or procedures. All MPE V/E object code is supported in CM. In addition COBOL II/V, HP FORTRAN 77/V, Pascal/V, SPL/V, RPG/V, BASIC/V, and HP Business BASIC/V compilers are available in CM at first release. Other subsystems available for CM use include: KSAM/V, Transact/V, Report/V, VPLUS/V, Dictionary/V, Inform/V, and HP Access Central/V.

By default, CM programs will access data that is compatible with MPE V/E-based systems.

NM procedures can be called through the Switch subsystem. For more information on the Switch subsystem, refer to the *Switch Programming Programmer's Guide (32650-90014)*.

If recompiling is necessary for CM operation, you should use one of the MPE V/E compilers available in CM on MPE/iX and the MPE V Segmenter if necessary.

Program Validation

A complete CM test plan should be part of your migration plan. Once all necessary changes have been made to get your application running in CM, it is time to complete the test procedures. When you first run your program in CM, you may want to continue to run the original version on the MPE V/E-based system to ensure that the migrated version gives the same results as the original version of the program. Testing should include monthly, quarterly, or yearly activities that are critical to your operation.

Improving Performance

OCT converts CM object code to the 900 Series instruction set for increased performance. The effort involved in using the OCT is equivalent to recompiling, and, like recompiling, the application should be retested to verify proper execution. You can invoke the OCT through the MPE/iX `:OCTCOMP` command.

OCT translates most CM instructions into HP-PA instructions and appends them to the end of the destination file. The resulting file can be executed on either MPE V/E-based or MPE/iX-based systems.

By using the MPE/iX command `:OCTCOMP`, you can create a new file with translated object code, translate

only selected segments of the object code, or add translated segments to another file. Object code translation will increase the size of the file.

A

Maintaining Backward Compatibility

This appendix describes how to maintain backward compatibility for programs and files. It is especially useful for developing programs on an MPE/iX-based system that may be transported to an MPE V/E-based system.

How to Ensure Compatibility

Programs can be developed in CM on MPE/iX for use on MPE V/E-based systems. It is important to understand that you are basically working in the MPE V/E environment when you are in CM. Consequently, most of the features and limitations of MPE V/E are in effect. Therefore, to maintain backward compatibility, your programs should adhere to the following guidelines:

- Use only the standard language features.
- Use the intrinsic mechanism, except for calling VPLUS from COBOL programs.
- Use only User Mode programming techniques.
- Avoid all migration issues listed in Appendix D.
- Use only compatible compiler options.
- Maintain MPE V/E data alignment schemes.
- Stay within MPE V/E limitations.
- Use only MPE V/E compatible subsystems and products.

- Use compatible database products.

Additional information on these guidelines can be found in the *Introduction to MPE XL for MPE V Programmers (30367-90005)* or the appropriate language or subsystem guide.

When changes are made to an MPE/iX CM program, the same changes should be made to the MPE V/E program, and both programs should be validated for consistency and accuracy.

Unavoidable Problems

There are certain differences between MPE V/E and MPE/iX that are unavoidable. These include:

- Different job streams and UDCs must be used for compiles.
- System usage of the stack is different. This can be a problem for very large programs on MPE V/E.

B

Using the Object Code Analyzer

This appendix describes how to use OCA (Version A.00.05). The OCA is intended to be used in conjunction with its companion Migration Toolset utilities, MPT and RTM, during the analysis and planning stage of the migration effort. OCA runs on both MPE V/E-based and MPE/iX-based systems. Included in this appendix are discussions on the following subjects:

- "OCA Operation", a general overview of how the tool operates.
- "Using OCA", a detailed description of the user interface.
- "OCA Report Formats", sample report formats created by OCA.
- "OCA Program Error Messages", a comprehensive list of error and warning messages returned by OCA, as well as recommended methods of error recovery.

OCA Operation

OCA is a tool designed to aid in migrating applications from an MPE V/E-based system to an MPE/iX-based system. OCA is a dialog-driven program that scans specified MPE V/E program and SL files in either batch or interactive mode and generates either brief or detailed reports of both definite and potential problem areas that may be encountered when migrating an application to an MPE/iX-based system. OCA allows you to specify the following options prior to entering the scanning portion of the program:

- The `Scan User Externals` option enables you to add your own list of incompatible procedures to OCA's set of predefined system incompatibilities. This list can include either system entry points or user-written procedures which are called as externals of a program or SL being scanned.
- The `Build Indirect File` option enables you to construct indirect files (files that contain file specifications, including other indirect file names) from generic file sets.

These two options can be used only one time per execution of OCA. Thus, one set of options remains in effect for all program or SL files scanned. If only default values are desired, the first part can be skipped, and OCA proceeds to the scanning part of the program.

During the scanning portion of the program, you direct OCA to scan MPE V/E program and SL files and report both definite and potential problem areas that may be encountered when migrating an application to an MPE/iX-based system. The scanning portion of the program can be repeated until all specified program and/or SL files are scanned.

OCA outputs a report of incompatibilities in either a brief report format or a detailed report format. The brief report format (the default) includes names of externally

referenced procedures that have been identified as incompatibilities for the purposes of migration. Use the brief report format if you are interested primarily in a migration report. Appendix D, "Incompatibilities", gives greater detail on each of the incompatibilities detected by OCA, as well as corrective actions to take.

When you specify a detailed report format, OCA reports on all of the program file's externally referenced procedures, as well as general information about the program's structure. Thus, the detailed report creation may take much longer to produce than the brief report, and the output volume is much greater.

During the scanning portion of the program, OCA opens each file you specify and determines if it is an SL file or a program file. If the file is a program file, OCA satisfies the program's external references before proceeding. If the file is an SL file, OCA does not satisfy external references. For either type of file, OCA compares the procedure names against OCA's predefined set of incompatible system procedures. If a detailed report is specified, then OCA reads every record of the file and determines the locations of the procedure call (PCAL) instructions to incompatible procedures.

Note

In extremely rare cases, OCA cannot distinguish between a valid PCAL and a piece of data that resembles a PCAL. Some PCALs, therefore, might be detected and reported as incompatibilities erroneously.

Using OCA

OCA is initiated by running the program file `OCA.PUB.SYS` either interactively or in batch mode. You can get an explanation of the input expected for any particular prompt by entering a question mark (?). You may also issue any MPE command that can be executed programmatically by prefixing it with a colon (:). After the HELP text is displayed, or the MPE command is executed, the original prompt is redisplayed. A double slash (//) at any prompt causes OCA to terminate.

Figure B-1 charts the flow of control for OCA prompts. Additional information on the prompts is provided in subsequent headings. Defaults for all prompts are shown in upper case.

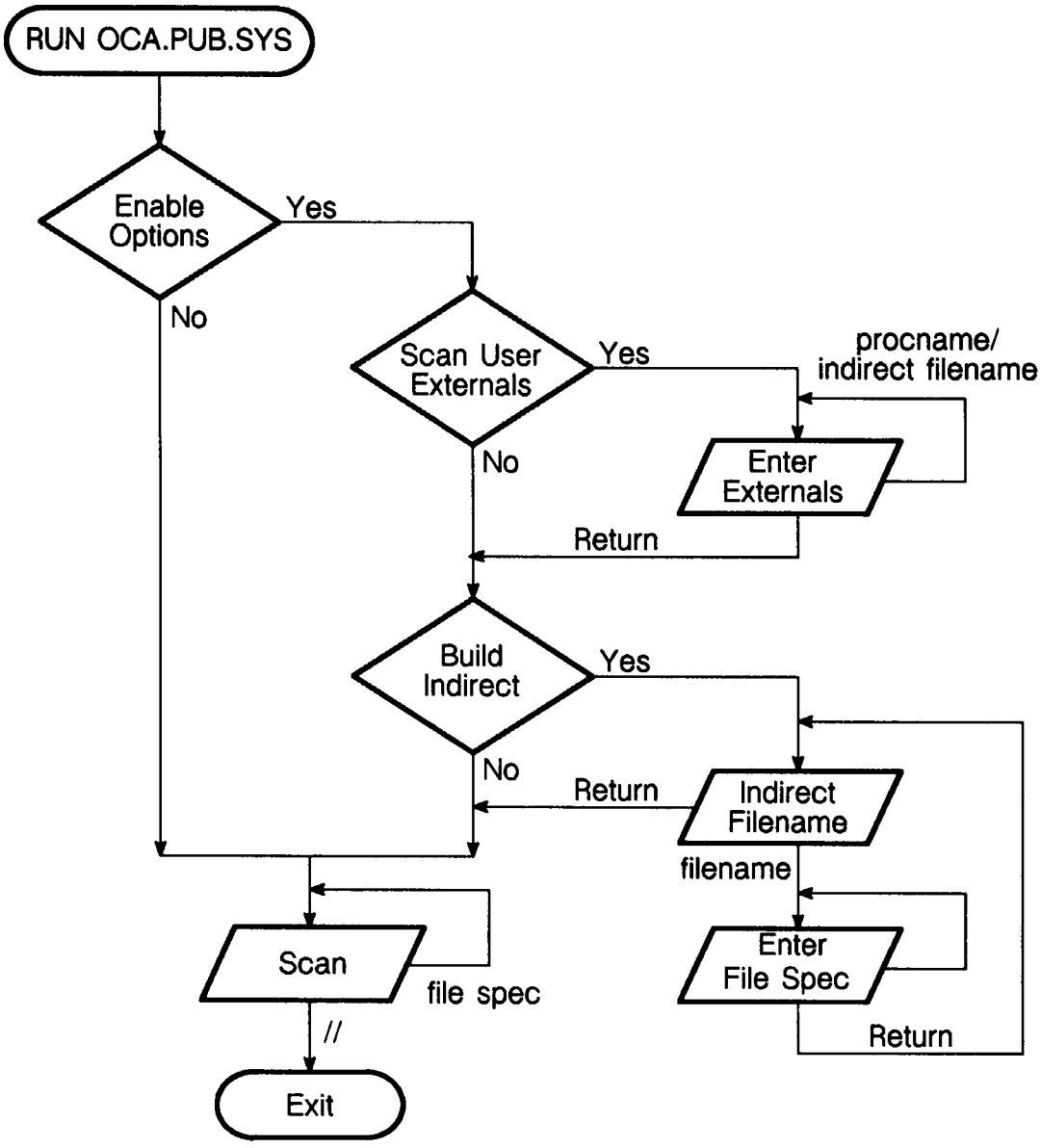


Figure B-1. Object Code Analyzer

Enable Options Prompt

```
Do you want to enable any options (yes/NO)?
```

This is the first prompt from OCA. You are being asked if you wish to access the two optional features of OCA prior to entering the scanning portion of the program.

- A positive response takes you to the User Externals prompt.
- A negative response (the default) or a Return
- A "//" terminates OCA.

Respond `Yes` only if you need to add user-selected, externally referenced procedures or build an indirect file of file specifications identifying files you want scanned.

Scan User Externals Prompt

```
Scan for user selected external procedure
references (yes/NO)?
```

OCA scans files for a set of predefined system incompatibilities. You are being asked if you have additional external procedures that you want temporarily added to the set of predefined system incompatibilities detected by OCA:

- A positive response indicates that you have additional external procedures that you want OCA to look for and takes you to the Enter External prompt.
- A negative response (the default) or a Return indicates that you want OCA only to look for its set of predefined system incompatibilities, and OCA takes you to the next option, the Build Indirect File prompt.
- A "//" terminates OCA.

When OCA scans program files, user-selected external procedure references are reported if they resolve to any SL. Predefined system incompatibilities are reported only when the procedure resolves to the system library `SL.PUB.SYS`. If a user-selected procedure name is identical to a predefined incompatible system procedure,

then two different messages may be reported. OCA prints a message if you add an external procedure name already in the list.

Enter External Prompt

`Enter external procedure name or indirect file name:`

You are being prompted for the external procedures that you want added to the events detected by OCA. You can specify system entry points and user-written procedures. The object code scanner looks for references to these external procedures in addition to the set of predefined system incompatibilities that OCA scans.

Press Return when you are finished entering procedure names and wish tp proceed to the next prompt

The syntax for this prompt is as follows:

[*external procedure name*]
[*!indirect file name*]
[*indirect file name*]

All of the external procedure names you enter at this prompt, either directly or via an indirect file, are written to a temporary file named `OCAPNAME`. If you wish, you can save this file as a permanent file using the MPE V/E `:SAVE` command and use it as an indirect file on subsequent invocations of OCA.

External Procedure Names

OCA accepts a very general, language independent syntax for procedure names. Names are required to start with an alpha character and then can contain any combination of alpha and numeric characters plus the apostrophe (') and underbar (_) characters.

OCA returns an error message if the procedure name is syntactically incorrect, and returns the prompt.

Because of restrictions in the MPE V Segmenter and the program/SL file format, procedure names are truncated to 15 characters. OCA follows this convention and issues a warning message when a name is over 15 characters in length.

External procedure names can be quoted (with either single or double quotes) or unquoted:

- Unquoted procedure names are upshifted as is the usual convention on an MPE V/E-based system.
- Quoted procedure names are treated as literals and are not upshifted or downshifted. (Use double quotes around procedure names containing imbedded single quotes.)

Indirect File Names

As an alternative to entering the names of all your external procedures one at a time, you may specify an indirect file that contains the names of external procedures you wish added to the set of predefined system incompatibilities OCA searches for during the scanning portion of the program.

As is the case with all OCA indirect files, the file name must be preceded by either an exclamation point (!) or a percent symbol (%). The indirect file must be an EDIT/V compatible file, numbered or unnumbered, with the following physical characteristics:

- Standard file type (the default)
- ASCII
- Permanent, existing in the system file domain
- File code must be 0 (the default)

If you respond with an incorrect or nonexistent file name, OCA produces an error message and issues the prompt again.

The general format of this file is similar to the format of indirect files described later in this appendix, except that nested indirect files are not allowed at this prompt. The file should contain one procedure name per line (leading and trailing blanks are ignored). The conventions for specifying external procedures are described in "External Procedure Names" above. You can use the string "—" to delineate the rest of the line as a comment that is ignored by OCA. The end of line ends the comment. Blank lines are likewise ignored. In addition, only the first 72 bytes of each record are read by OCA. Everything past column 72 is ignored.

Following is an example of the file of procedure names created by OCA:

```
5     READ'TAB'1      --comment to end of line
6     PROC1
7
8     KEEPTIME
9     UPDATE_TAB      --comment to end of line
10    "Abort'IO"      --a quoted procedure name
11    'AbortProg'     --another quoted procedure name
```

Build Indirect File Prompt

```
Build an indirect file of file names (yes/NO)?
```

You are being asked if you want OCA to build a file that contains the MPE file designators of the files you want OCA to scan later during the scan portion of the program.

- A positive response to this prompt causes OCA to build an indirect file you can later specify at the SCAN prompt and takes you to the Indirect File Name prompt.
- A negative response (the default) or a Return takes you to the SCAN prompt.

- A "//" terminates OCA.

This "Build Indirect File" feature can be useful if you have a large number of files to scan that are centralized in a few key groups and accounts on the system, and you want to save the set of file names for future use. A second use of this feature is generating a list of the names of programs and SL files to be considered as candidates for migration.

Indirect File Name Prompt

```
Enter name for indirect file:
```

You are being prompted for the name of the file you want to associate with the indirect file you are about to create. This file will contain the names of files that you later want OCA to scan during the scanning portion of the program. Valid responses to this prompt are:

- A file name following MPE V/E file naming conventions.
- A Return takes you to the Scan prompt.
- A "//" terminates OCA.

If you specify a valid MPE file name and press Return, creates and opens a new file with the name you specified at the prompt, then proceeds to the "Enter File specification" prompt (described below.).

Enter File Specification Prompt

```
Enter file specification:
```

You are being prompted for the information to include in the indirect file you just created. The indirect file is used to specify the files to be scanned in response to the SCAN prompt and is helpful in identifying program and SL files from generic file sets. Each line of the indirect file will be treated by the SCAN prompt as if you had entered it interactively (in other words, as a command image). Thus, the indirect file can contain any valid response to the SCAN prompt.

After OCA has opened the indirect file, it repeatedly prompts you for files or file sets to be scanned. The file set specifications may include wild cards. OCA automatically selects all program and SL files residing in the file set. The fully qualified names of individual program and SL files are written to the indirect file.

Press Return when you are finished entering file specifications and wish to proceed to the next prompt. Once you have entered the contents of the indirect file,

OCA attempts to close the indirect file as a permanent file.

If a file of the same name already exists in the permanent file directory, OCA asks you if the old file should be purged. If you respond `YES`, OCA purges the old file before it closes the indirect file.

If you respond `NO` to the purge request (or press Return), OCA asks if you want to save the file under a different name. If you respond `NO` to this prompt (or press Return), the indirect file you just built is lost. If you respond `YES`, you are prompted for a different name for the indirect file you just built. If you press Return in response to the prompt for a different name, the indirect file you just built is lost.

In both cases where the file is lost, OCA returns you to the Indirect File Name prompt and allows you to create a new indirect file. When you succeed at saving the indirect file, OCA takes you to the SCAN prompt. The syntax for the `Enter File Specification` prompt is:

file specification [library specification] [report level] [report destination]

The syntax for the required *file specification* is:

```
{ filename                 }
{ fileset                  }
{ !indirect filename       }
{ #indirect filename       }
```

The syntax for the optional *library specification* is:

```
          { GROUP  }
[; LIB = { PUBLIC } ]
          { SYSTEM }
```

The syntax for the optional *report level* is:

```
[; DETAILED ]
[; BRIEF    ]
```

The syntax for the optional report destination is:

```
[; OFFLINE ]
```

As a convention OCA keywords are shown in upper case, although you do not have to enter them in upper case on the command line. In addition, only the minimum number of characters necessary to recognize the keyword must be specified. For example, `;DETAILED` can be entered as `;DET` or as `;D`.

Note

Each line of the indirect file is treated as a command image at the SCAN prompt. Therefore, the syntax and meaning of the options listed above are identical to the optional parameters available throughout the SCAN prompt.

Details of the *file specification*, *library specification*, *report level*, and *report destination* parameters are discussed below.

File Specifications

OCA scans only MPE V/E format SL or program files. You can specify the files you want placed in the indirect file in the following ways:

- By file name
- By file set
- By indirect file name

By File Name. You can enter the name of a single program or SL file. OCA displays an error message if the file is not of the correct type. Program and SL files are treated slightly differently; the differences are reflected only in the detailed report.

By File Set. You can enter a file set. You can select groups of program and/or SL files using MPE wild card characters (*@*, *#*, *?*) in conjunction with file designators. OCA searches the specified file set and includes in the indirect file the fully qualified file designators of only the program and SL files located in the specified file set. For example, if you specify `@.mygroup.myacct`, OCA places in the indirect file all program and SL files located in group `mygroup`, account `myacct`.

Note

The *file set* specification can contain MPE wild card characters (*@*, *#*, *?*) except that OCA interprets the single question mark followed by a space or Return as a request for OCA `HELP` facility instead of as a wild card character. Therefore, if you want to specify all single-character file names, you must use a more qualified file specification, for example:

```
?.mygroup
?.mygroup.myacct
```

By Indirect File Name. You can enter the name of an indirect file that you have previously created either by the `Build Indirect File` option of OCA or from an editor.

As is the case with all OCA indirect files, the file name must be preceded by either a ! or a %. The indirect file must be an EDIT/V compatible file, numbered or unnumbered, with the following physical characteristics:

- Standard file type (the default)

- ASCII
- Permanent, existing in the system file domain
- File code must be 0 (the default)

If you respond with an incorrect or nonexistent file name, OCA produces an error message and issues the prompt again.

The general format of this file is similar to the format of indirect files described earlier in this appendix, except that indirect files of file specifications can be nested up to five levels deep. The file should contain one command per line (following the syntax described above). Leading and trailing blanks are ignored. Use the string "—" to delineate the rest of the line as a comment that is ignored by OCA. The end of line ends the comment. Blank lines are likewise ignored. In addition, only the first 72 bytes of each record are read by OCA (everything past column 72 is ignored). Following is an example of the contents of an indirect file of file specifications:

```
1   $OMRPT;LIB=G
2   $CRNDVR.PUB.DRIVERS  --comment to end of line
3   @.PUB.OURSYS;DETA
4   $EDIT;BRIEF;OFFLINE
5   !INDIRECT.PUB.MYACCT;BRIEF  --an indirect file
```

Note

The additional options you specify with the *indirect filename* option will override whatever options may have been specified within that indirect file. In the example above, specification of the `;BRIEF` option for the indirect file `!INDIRECT.PUB.MYACCT` overrides any specifications for a detailed report format that may be made within `!INDIRECT.PUB.MYACCT`.

LIB= Option

The LIB= option tells OCA what system libraries the loader would search if the specified program file were to be loaded. It is interpreted in exactly the same way as the LIB= option on the MPE V/E :RUN command. The default library indicator is "S". Specifying LIB=S is equivalent to omitting the LIB= option entirely.

The LIB= option should be specified only with program files. If you specify LIB with an SL file, the LIB= parameter is ignored.

When the LIB= option is used with a file set or with an indirect file, the specified library is applied to those files in the file set (or in the indirect file) that are program files and is ignored for the SL files.

If you request that a file is to be scanned LIB=G or LIB=P and a group or account SL does not exist, OCA does not print an error message. This is consistent with the behavior of the MPE V/E :RUN command. However, the OCA detailed report notes when an SL is not required.

OCA always tells you which SL file the external procedure will actually be resolved to when the program is loaded. It also tells you if you specified a group or account SL and none of the externals resolved to the group or account SL specified. This information is useful for determining which SLs are needed (and not needed) for migration.

Note

If you request that an SL file is to be scanned, OCA scans the entire SL file. If you request that a program file be scanned the LIB= option, OCA scans only the segments that contain external procedures directly referenced by the program, or indirectly referenced by externals of SL segments.

Brief and Detailed Options

By default, OCA produces a report on *file specification* in a brief format that provides only information directly related to migrating the program file successfully, including names of external procedures that have been identified as incompatibilities for the purposes of migration. Specifying `;BRIEF` with a *file name* or *file set* is the same as not specifying `;BRIEF`.

You can optionally specify `;DETAILED` if you want much greater detail about all of *file specification's* externally referenced procedures. In addition, the `;DETAILED` option produces general information about the program structure of *file specification*.

Specifying either the `;BRIEF` or the `;DETAILED` option with an indirect file overrides all uses of this option that may be found within that indirect file.

Offline Option

By default, OCA output is directed to the job/session list device `$STDLIST`. If you use the `;OFFLINE` option, you direct OCA output to the OCA list file `OCALIST`. By default this file is assigned to the system line printer (`DEV=LP`). If you want to redirect output to a different device, you can create an MPE file equation using the `:FILE` command to redefine the characteristics for `OCALIST`. This file equation is invoked for *file specification* only if you specify `;OFFLINE`. In the following example, the `:FILE` command is invoked at the SCAN prompt to create a file equation for `OCALIST`, then a file name is specified with the `;OFFLINE` option:

```
SCAN> :FILE OCALIST;DEV-EPOC
SCAN> @.TOOLS.SWAT;OFFLINE
SCAN>
```

SCAN Prompt

SCAN>

You are being prompted for the name of the file(s) you want OCA to scan for incompatibilities.

Valid responses to this prompt are:

- A file specification (with optional keyword parameters), following the syntax description described below.
- Return, which returns you to the SCAN prompt.
- A "//", which terminates OCA.

A report is generated for each file scanned, and you are prompted for the name of the next file to scan. If a file set has been specified, OCA displays progress messages indicating the start of the directory search and the number of program and SL files found. OCA always displays the name of the file being scanned.

Normally, OCA automatically checks the files against a set of predefined system incompatibilities located in the file OCAINCOM.PUB.SYS. Optionally, a set of user-selected incompatibilities created by the Build Indirect File option temporarily disables access to OCAINCOM.PUB.SYS by using a file equation to equate OCAINCOM.PUB.SYS to $NULL prior to executing OCA. You can later invoke the :RESET command to enable OCAINCOM.PUB.SYS. Following is a sequence of SCAN prompt entries where this is accomplished:

```
:FILE OCAINCOM.PUB.SYS=$NULL
:RUN OCA.PUB.SYS
:SCAN>//
:RESET OCAINCOM.PUB.SYS
```

The syntax for the SCAN prompt is:

file specification [library specification] [report level] [report destination]''

The syntax for the required *file specification* is:

```
{ filename     }
{ fileset }
{ !indirect filename           }
{ #indirect filename           }
```

The syntax for the optional *library specification* is:

```
          { GROUP  }
[; LIB = { PUBLIC } ]
          { SYSTEM }
```

The syntax for the optional *report level* is:

```
[; DETAILED ]
[; BRIEF    ]
```

The syntax for the optional *report destination* is:

```
[; OFFLINE ]
```

As a convention, OCA keywords are shown in upper case, although you do not have to enter them in upper case on the command line. In addition, only the minimum number of characters necessary to recognize the keyword must be specified. For example, `;DETAILED` can be entered as `;DET` or as `;D`.

If you specify the file name `EXIT` (or any of its substrings `EXI`, `EX`, `E`) and a file by that name does not exist in the current logon group, OCA terminates. To be safe, qualify `EXIT` (or its substring) with the group name (for example, `EXI.MYGROUP`).

Details on the *file specification*, *library specification*, *report level*, and *report destination* parameters are discussed below.

File Specifications

OCA scans only MPE V/E format SL or program files. You can specify the files you want scanned in the following ways:

- By file name
- By file set
- By indirect file name

By Filename. You can enter the name of a single program or SL file. OCA displays an error message if the file is not of the correct type. Program and SL files are treated slightly differently. The differences are reflected only in the detailed report.

By File Set. You can enter a file set by selecting groups of program and/or SL files using MPE wild card characters (`@, #, ?`) in conjunction with file designators. OCA searches the specified file set and scans only the program and SL files located in the specified file set. For example, if you specify `@.mygroup.myacct`, OCA scans all program and SL files located in group `mygroup`, account `myacct`.

Note

The *fileset* specification can contain MPE wild card characters (`@, #, ?`), except that OCA interprets the single question mark followed by a space or Return. Therefore, if you want to scan single-character file names you must use a more qualified file specification, for example:

```
?.mygroup
?.mygroup.myacct
```

By Indirect File Name. You can enter the name of an indirect file that you have previously created either by the `Build Indirect File` option of OCA or from an editor. As is the case with all OCA indirect files, the file name must be preceded by either a ! or a %. The indirect file must be an EDIT/V compatible file, numbered or unnumbered, with the following physical characteristics:

- Standard file type (the default)
- ASCII
- Permanent, existing in the system file domain
- File code must be 0 (the default)

If you respond with an incorrect or nonexistent file name, OCA produces an error message and issues the prompt again.

The general format of this file is similar to the format of indirect files described earlier in this appendix. Indirect files of file specifications can be nested up to five levels deep. The file should contain one file specification per line (following the syntax described above). Leading and trailing blanks are ignored. You can use the string "—" to delineate the rest of the line as a comment that is ignored by OCA. The end of line ends the comment. Blank lines are likewise ignored. In addition, only the first 72 bytes of each record are read by OCA. (Everything past column 72 is ignored.) Following is an example of the contents of an indirect file of file specifications:

```
1    EOMRPT;LIB=G
2    SCRNDVR.PUB.DRIVERS  --comment to end of line
3    SCRAMBLE.PUB.OURSYS
4    ZEDIT;BRIEF;OFFLINE
5    !FILESPECS.PUB.MYACCT;DETAILED  --an indirect file
```

Note

The additional options you specify with the indirect file name option at the SCAN prompt will override whatever options that may have been specified within that indirect file. For example, if you specify at the SCAN prompt the indirect file described above with the `;BRIEF` option, the contents of the file set `@.PUB.OURSYS` and the contents of the indirect file `!FILESPECS.PUB.MYACCT` will be reported in the brief report format.

LIB= Option

The `LIB=` option tells OCA what system libraries would be searched if the program file were to be loaded. It is interpreted in exactly the same way as the `LIB` parameter on the MPE V/E `:RUN` command. The

default library indicator is "S". Specifying LIB=S is equivalent to omitting the LIB= parameter.

The LIB= option should be used only if scanning a program file. If you specify LIB= with an SL file, the LIB= parameter is ignored.

When the LIB= option is used with either a file set or an indirect file, the specified library is applied to those files in the file set (or in the indirect file) that are program files, and ignored for the SL files.

If you request that a file is to be scanned LIB=G or LIB=P and a group or account SL does not exist, OCA does not print an error message. This is consistent with the behavior of the MPE :RUN command. The OCA detailed report, however, notes when an SL is not required.

OCA always tells you which SL file the external procedure will actually resolve when the program is loaded. It also tells you if you specified a group or account SL and none of the externals would resolve to the group or account SL specified. This information is useful for determining which SLs are needed (and not needed) for migration.

Note

If you request that an SL file is to be scanned, OCA scans the entire SL file. If you request that a program file be scanned with the LIB parameter, OCA scans only the segments that contain external procedures directly referenced by the program, or indirectly referenced by externals of SL segments.

Brief and Detailed Options

By default, OCA produces a report on *file specification* in a brief format that provides only information directly related to the ability of the program file to migrate successfully, including names of external procedures that have been identified as incompatibilities for the purposes

of migration Specifying ;BRIEF with a file name or file set is the same as not specifying ;BRIEF.

You can optionally specify ;DETAILED if you want much greater detail about all of *file specification's* externally referenced procedures. In addition, the ;DETAILED option produces general information about the program structure of *file specification*.

Specifying either the ;BRIEF or the ;DETAILED option with an indirect file overrides all uses of this option that may be found within that indirect file.

Offline Option

By default, OCA output is directed to the job/session list device $STDLIST. If you use the ;OFFLINE option, you direct OCA output to the OCA list file OCALIST. By default this file is assigned to the system line printer (DEV=LP). If you want to redirect output to a different device, you can create an MPE file equation using the :FILE command to redefine the characteristics for OCALIST. This file equation is invoked for *file specification* only if you specify ;OFFLINE. In the following example, the :FILE command is invoked at the SCAN prompt to create a file equation for OCALIST, then a file name is specified with the ;OFFLINE option:

```
SCAN>:FILE OCALIST;DEV=EPOC
SCAN>@.TOOLS.SWAT;OFFLINE
SCAN>
```

Interactive Pagination

When using OCA interactively and the report fills one page on your screen, OCA displays the More prompt. OCA does not display the More prompt when run from batch mode.

```
More (YES/no/all)?
```

The following responses are valid to the More prompt:

- `YES` (the default) tells OCA that you want to continue listing the text. OCA will display the More prompt each time it fills one page on your screen.
- `NO` tells OCA that you do not want to see the rest of the text. OCA stops scanning any other files specified, and returns you to the SCAN prompt.
- `ALL` tells OCA that you want to see all of the report without intervening More prompts.

OCA uses two session level job control words (JCWs) to control page length for output (help text or reports). If OCA determines that the output destination is offline then OCA will use the JCW `OCAPAGESIZE` to determine the number of lines per page. This would be the case if you specified `;OFFLINE` on the command line, or OCA was being run from a `:JOB`. If the output destination is your terminal screen, OCA will use the `OCASCREENSIZE` JCW. If these JCWs do not exist or are set to values out of range, OCA will use default values for the page size. If the appropriate JCW is set to zero, then the paging mechanism is disabled.

Below are the default values and legal ranges for each of the JCWs described above:

- For JCW `OCASCREENSIZE`, default value is 23; legal range is 0, 1 ... 100
- For JCW `OCAPAGESIZE`, default value is 60; legal range is 0, 5 ... 100

Exiting OCA

Entering a double slash (//) at any prompt terminates OCA. Additionally, if you specify `EXIT` (or any of it's substrings `EXI`, `EX`, `E`) at the SCAN prompt, or if `EXIT` is encountered in an indirect file, and a file by that name does not exist in the current logon group, OCA terminates.

Security Considerations

There are no special restrictions on who can run OCA. OCA, however, places restrictions on which files may be scanned. You must have READ access to the file to scan it. In addition, you must provide lockwords for files requiring them. Users possessing Account Manager capability may scan any file residing in their accounts without specifying the lockword. Similarly, system manager capability allows all files on the system to be scanned without specifying the lockword. Access restrictions, however, may prevent you from scanning a file despite your capabilities. In most cases, if you can run the program, you can scan it.

Running OCA in Batch Mode

Figure B-2 illustrates how you can use a job stream file to invoke OCA in batch mode. Example B1 shows all the major operations available through `OCA.PUB.SYS`. Note how MPE V/E commands are executed and then control is returned to the same OCA prompt. In addition, the string "—" indicates to OCA that the remainder of the input line is a comment. Because OCA deletes these comments from the input line before parsing occurs, the comments can be included on MPE V/E command execution lines (see OCA invocation of `:FILE` command).

```
:job ocajob, manager.sys
:run oca.pub.sys
y
y                          -- want to enable some options
procedure1                 -- want to add user selected externals
procedure2                 -- enter the names of some procedures you
procedure3                 -- want to add to predefined list of system
procedure4                 -- incompatibilities
procedure5
                           -- blank line indicates no more procedure names
:comment save the procedure names entered above for later use
:save ocapname
y                          -- want to build an indirect file
indirect                   -- call the file indirect
@.@.finance                -- get all programs/SLs in finance account
@.tools.@;lib=g            -- get all tools groups in all accounts ;LIB=G
                           -- blank line to indicate no more filesets
                           -- blank line to indicate no more indirect files

-- We are at the Scan >prompt now, blank lines or comment have no effect.
-- We are going to do a :showtime before and after we generate the
-- report just so we can tell how long it took.
:showtime
:file ocalist;dev=epoc     --direct output to epoc printer
!indirect;brief;offline    --scan the indirect file created above
:showtime
//                         --terminates OCA
:eoj
```

Figure B-2. Job Stream to Execute OCA

OCA Report Formats

A report is automatically generated for each file that OCA scans. There are two formats of the report:

- Brief format
- Detailed format

The brief report format provides only information directly related to the ability of the program or SL file to migrate successfully. This information includes a list of potential incompatibilities detected when OCA scanned the selected program or SL files. Appendix D, "Incompatibilities", provides detailed descriptions and corrective actions to take for each of the incompatibilities detected by OCA.

Note

It is highly recommended that you run the brief report first. Run the detailed report only on programs which have potential incompatibilities as identified in the brief report, or if the additional information is needed.

The detailed report format provides more information about the program than in the brief report format. It also provides the code locations of potential incompatibilities. This should be done only on programs which have incompatibilities as reported by the OCA brief report (and not disqualified by RTM). A detailed report can be used to locate privileged segments or provide a better understanding of the program's structure. The content of this report differs slightly depending on whether the file is a program file or an SL file.

If you have the source file, following are two alternatives to using the detailed report to obtain the code location of incompatibilities. After you have used the OCA brief report format to identify the names of incompatible external procedures:

- Use an editor to find the exact location in your source file of the call to the incompatible procedure.
- Use the compiler listing and a cross reference to find the exact location of the call to the incompatible procedure.

Output Device Specification

If you use the `;OFFLINE` option, you can direct OCA report or HELP text output to the OCA list file `OCALIST`. By default this file is assigned to the system line printer (`DEV=LP`). If you want to redirect output to a different device, you can create an MPE file equation using the `:FILE` command (either at the SCAN prompt or prior to running OCA) to redefine the characteristics for `OCALIST`. This file equation is invoked for file specification only if you specify `;OFFLINE`. In the following example, the `:FILE` command, is invoked at the SCAN prompt to create a file equation for OCALIST then a file name is specified with the `;OFFLINE` option:

```
SCAN>:FILE OCALIST;DEV=EPOC
SCAN>@TOOLS.SWAT;OFFLINE
SCAN>?;OFFLINE
```

At any prompt, you can use the `;OFFLINE` option immediately after a question mark "?" to redirect the HELP text output associated with that prompt. In the example above, HELP text about the `SCAN` prompt is redirected to device `EPOC`.

Contents of Brief and Detailed Report Formats

The following tables summarize the types of information returned in the brief and detailed reports created by OCA. The tables mirror the sequence of sections of the OCA report. Sections printed only for program files are noted in the table description:

- General Information (returned for program files only)
- Segment Information

- Resolved External Procedures
- Unresolved External Procedures (returned for program files only)
- Potential Incompatibilities
- Intrinsic Mechanism Information

General Information

Table B-1 summarizes the information returned in the "General Information" section of an OCA report. This information is returned only for program files; segmented library (SL) files do not have equivalent characteristics. The first two categories of information indicate potential migration issues. You should pay particular attention to whether or not the program file was prepped with Privileged Mode (PM) capability. In this case, check to see if your program file contains Privileged Mode segments. In general, use of Privileged Mode code in an application is much more likely to cause migration-related problems.

Table B-1.
General Information (program files only)

Information	Brief	Detailed
Program capabilities(for example, IA,BA,DS,PM)	X	X
Are there privileged segments?	X	X
Initial stack info: size of program global area (DB-QI)		X
Initial stack info: size of program global (DL-DB)		X
Initial stack info: size of initial DL area		X
Initial stack info: program of MAXDATA size (DL-Z)		X
Was a program prepped with; FPMAP option		X

Segment Information

Table B-2 summarizes the information returned in the "Segment Information" section of an OCA report. The "Segment Information" section returns information about all program/SL segments, as well as information about possible patch areas if the program/SL file is patched. If the "General Information" section of an OCA report indicates that the program file contains Privileged Mode segments, the "Segment Information" section is useful for determining which program/SL segments are privileged.

Table B-2. Segment Information

Information	Brief	Detailed
Table containing the segment name (SL files only), segment number, segment length, and mode (User or Priv)		X
Summary information: total code size		X
Summary information: average code segment size		X
Patch area information: program name(program fine patch areas only)		X
Patch area information: segment name		X
Patch area information: prep date/time		X
Patch area information: patch data/time		X
Patch area information: whether or not the patch area has been altered		X
Patch area information: an octal dump of the patch area (only if the patch area has been altered)		X

Resolved External Procedures

Table B-3 summarizes the information returned in the "Resolved External Procedures" section of an OCA report. The "Resolved External Procedures" section of the report provides a complete list of all external procedures called by a program directly or indirectly. This section also provides a cross-reference of all the calls. External procedure calls originating from a segment in the system library `SL.PUB.SYS` will not be in the list. This section may be very long for

large programs and for programs that use SL files. Hewlett-Packard recommends that you request this information only if you need to get a "feel" for what a program is doing, and how it does it (for example, if you are migrating a program you did not write).

Table B-3. Resolved External Procedures

Information	Brief	Detailed
The name of the external procedure		X
Library in which the external procedure resolved (Group, Pub, System)		X
Segment number of the segment(s) calling the procedure		X
SST number(s) for the PCAL to the procedure		X
PB relative offsets for the calls to the procedure		X

Unresolved External Procedures

Table B-4 summarizes the information returned in the "Unresolved External Procedures" section of an OCA report. This section is printed only if the program file contains one or more unresolved external procedure references. This usually occurs when you do not specify the correct library at the SCAN prompt using the `LIB=` option. For best results, you should scan the program file using the same `LIB=` option as you would when you run the program.

Table B-4.
Unresolved External Procedures (program files only)

Information	Brief	Detailed
A list of the external procedures which were unresolved	X	X
The reason why the procedure was unresolved	X	X

Potential Incompatibilities

Table B-5 summarizes the information returned in the "Potential Incompatibilities" section of an OCA report. This section lists the potential migration issues you should expect to encounter.

Note

You should read the messages encountered in this section, and refer to Appendix D, "Incompatibilities", for a more detailed explanation of the cause of the incompatibility, as well as possible corrective actions to take to remedy the problem.

It is possible that the OCA report may list calls to system intrinsics that you will be unable to locate in your code. This should not alarm you. Many MPE V/E compilers generate calls to system intrinsics on your behalf. For example, the COBOL II/V compiler generates a call to the system intrinsic `XCONTRAP` if you compile with the `$CONTROL DEBUG` option enabled. When you recompile your source code using an MPE/iX NM compiler, all compiler-generated external references will be properly handled by the compiler.

It is also possible that the OCA report may list incompatibilities against SL procedures not actually referenced by the program. If your program refers to an

entry in an SL segment (whether directly or indirectly), OCA scans all procedures found in the SL segment, whether or not they are actually referenced. This occurs because the whole SL segment (not just the referenced procedures) is bound to the program at run time. The recommended method of determining whether or not a reported procedure is actually called by a program is to enable RTM and run the program again. (For details on this method, refer to Appendix C, "Using the Run Time Monitor".)

Table B-5. Potential Incompatibilities

Information	Brief	Detailed
A list of all the incompatibilities detected	X	X
The name of the incompatible procedure that is referenced.	X	X
A brief message indicating the nature of the incompatibility.	X	X
An indication of the mode in which a user would expect to encounter the problem (that is,CM only, CM/NM,NM only).	X	X
A message number which can be cross-referenced with a number from RTM and looked up in Appendix D.	X	X

You can also use the source code of both the program and SL segments to determine which procedures are actually referred to by the program by successfully tracing reported procedures back to the main program. If you are an advanced user of MPE V/E, you can use the OCA detailed report with `LIB=G/P` along with PMAPs (created with the MPE V/E SEGMENTER) of all SL segments in question. Cross reference all reported

procedures and determine whether the procedures are actually referred to.

Summary Information

Table B-6 summarizes the information returned in the "Summary Information" section of an OCA report. This section reports total numbers of events listed in the "Potential Incompatibilities" section of the report.

Table B-6. Summary Information

Information	Brief	Detailed
Total number of CM only incompatibilities detected	X	X
Total number of CM/NM incompatibilities detected	X	X
Total number of NM only incompatibilities detected	X	X
Total number of user-defined incompatibilities detected	X	X
Total number of uncallable procedures detected	X	X

Intrinsic Mechanism Information

Table B-7 summarizes the information returned in the "Intrinsic Mechanism Information" section of an OCA report. This section briefly describes the intrinsic mechanism issue related to migration, and lists all of the names of the system intrinsics your program references. Using this list, you can consult your source code to determine if you actually used the intrinsic mechanism. The intrinsic mechanism issue related to migration is discussed below.

Table B-7. Intrinsic Mechanism Information

Information	Brief	Detailed
A message briefly explaining the intrinsic mechanism issue	X	X
A list of all the intrinsics that the application references	X	X

Although you may not recognize it by the name intrinsic mechanism, the intrinsic mechanism has been an MPE V/E feature for a long time. All of the programming languages supported on MPE V/E allow users to call system-supplied routines, called intrinsics, without explicitly declaring the intrinsic parameters. The compilers determine if you are calling the procedure correctly by examining the file `SPLINTR.PUB.SYS`, which contains the declarations for all the system intrinsics.

The MPE V/E intrinsic mechanism is provided as a convenience for users. It is a shorthand notation for calling external procedures, but it does not cause the compiler to generate code any differently than if the procedure is declared as an external procedure and is called as in the example above. MPE/iX also supports a set of NM compilers which handle the intrinsic mechanism in a slightly different manner.

On MPE/iX, the intrinsic mechanism plays a more important part in the compilation process. The intrinsic file on MPE/iX, `SYSINTR.PUB.SYS` contains more information about the procedure which is being called. For example, it contains information about the mode in which the procedure executes (CM or NM) and about how it can be accessed via the Switch subsystem. Furthermore, it contains crucial information about the parameter alignment necessary for calling the procedure.

If you plan to migrate your application to NM on MPE/iX, you must use the intrinsic mechanism to get correct results. By mandating the use of the intrinsic mechanism and enhancing it to contain several new features, MPE/iX will be able to provide a programmatic interface which is more flexible and extensible than the one currently provided by MPE V/E.

Each of the MPE V/E programming language reference manuals contain a section on how to call system intrinsics. These manuals can be consulted to determine the appropriate syntax for the language you are using. If you are already using the intrinsic mechanism (not explicitly declaring the procedures or allowing the compiler to do it implicitly), you will not encounter any problems in this area when migrating to NM on MPE/iX.

All of the languages supported on MPE/iX in NM have their own migration guides. These guides will provide you with additional information on the subject of the intrinsic mechanism and how to deal with any potential issues which may arise when recompiling to NM. Some of the MPE/iX compilers support special migration related compiler options which can be of great assistance in determining if you have a problem in this area. COBOL II/XL, for example, supports a `$CONTROL CALLINTRINSIC` option which will generate a warning if your program contains calls to system intrinsics without using the intrinsic mechanism.

It is possible that the OCA report may list calls to system intrinsics that you will be unable to locate in your code. This should not alarm you. Many of the MPE V/E compilers will generate calls to MPE V/E system intrinsics on your behalf. For example, most of the compilers will generate a call to the system intrinsic `TERMINATE` which handles terminating your process when your program is done executing. Any procedure names which you find in the list generated by OCA but

cannot find references to in your source code fall into this category.

When you recompile your source code on MPE/iX using NM compilers, all compiler-generated external references are handled through the intrinsic mechanism.

Sample Brief Report

Figure B-3 provides a sample of a brief OCA report.

```
OCA HP30365A.00.05 (Cat A.00.05) - THU, OCT 23, 1986, 11:01 AM  Page
No.  1 Report for STICP.SOOL.MPEV ;LIB=G ;BRIEF

  G E N E R A L    I N F O R M A T I O N

Program was prepped with ;CAPS=BA,IA.
Program contains only user mode segments.

  P O T E N T I A L     I N C O M P A T I B I L I T I E S

Incompatibilities detected in the program file "STICP.SOOL.MPEV".

FCONTROL:  *** System Defined Incompatibility ***
      - Some FCONTROL controlcodes are no longer valid
        on MPE/iX (CM/NM 500).
        These control codes include:  03, 48.

Incompatibilities detected in the group SL (SL.SOOL.MPEV).

CREATEPROCESS:  *** System Defined Incompatibility ***
      - CREATEPROCESS may encounter byte pointer problems
        in NM on MPE/iX (NM 107).

FCONTROL:  *** System Defined Incompatibility ***
      - Some FCONTROL controlcodes are no longer valid
        on MPE/iX (CM/NM 500).
        These controlcodes include:  03, 48.

  S U M M A R Y      I N F O R M A T I O N

NM Only Incompatibilities:1
CM/NM Incompatibilities:2
Uncallable Procedures Detected:0

  I N T R I N S I C      M E C H A N I S M     I N F O R M A T I O N


Applications being recompiled to native mode must
use the intrinsic mechanism for system intrinsics.

Number of procedures requiring the intrinsic mechanism:  3

Procedures from STICP.SOOL.MPEV requiring the intrinsic mechanism:FCONTROL

Procedures from SL.SOOLMPEV requiring the intrinsic mechanism:
CREATPROCESS       FCONTROL
```


Figure B-3. Sample Brief Report

Sample Detailed Report

Figure B-4 provides a multipage sample of a detailed OCA report. It provides an expanded listing of general information, segment information, and resolved external procedures.

```
OCA HP30365A.00.05 (Cat A.00.05) - THU, OCT 23, 1986, 11:01 AM  Page
No.  1 Report for STICP.SOOL.MPEV ;LIB=G ;DETAILED

  G E N E R A L    I N F O R M A T I O N

Program was prepped with ;CAPS=BA,IA.
Program contains only user mode segments.

Program was prepped with ;CAPS=BA,IA.
Program contains only user mode segments.

Initial Stack Information:
    Size of initial DL area (DL-DB):           0  (%000000)
    Size of program global area (DB-QI):     284  (%000434)
    Size of initial stack (QI-Z):           4256  (%010240)
    ;MAXDATA= was not specified on :PREP command.

  S E G M E N T    I N F O R M A T I O N

Segment information for the program file "STICP.SOOL.MPEV".
Total number of segments: 1
Starting segment number:  0

 Segment           Segment
 Number            Length (%octal)           Mode
 ================================================
 %00000              76 (%00114)             USER

 Total code segment size: 76   Average code segment size: 76

 Information for referenced segments in the group SL (SL.SOOL.MPEV).
 Number of segments referenced: 1

 Segment            Segment        Segment
 Name               Number         Length (%octal)        Mode
 ============================================================
 STICK              %00000            72 (%00110)    USER

 Total code segment size: 72   Average code segment size: 72

 Information for referenced segments in the account SL (SL.PUB.MPEV).
 No externals resolved to this SL.
 NOTE :  This application does not require this SL to migrate.

 Entry Point       PB Relative Address     Stt Number
 ================================================
 OB'                     %000013            %000001
```

```
  R E S O L V E D    E X T E R N A L    P R O C E D U R E S

External references of the program "STICP.SOOL.MPEV".
                     Called   At          Resolved To
                    ------------------      -----------
Procedure Name    Seg Num   STT Num           LIB
=====================================================
FCONTROL          %00000      8             SYSTEM

  ****  O F F S E T S  ****

%000061

Total Number of calls in segment 0: 1

Total number of calls to FCONTROL:      1
Number of segments which call FCONTROL: 1

                     Called   At          Resolved To
                    ------------------      -----------
Procedure Name    Seg Num   STT Num           LIB
=====================================================
P'CLOSEIO         %00000      5             SYSTEM

  ****  O F F S E T S  ****

%000072

Total Number of calls in segment 0: 1

Total number of calls to P'CLOSEIO:      1
Number of segments which call P'CLOSEIO: 1

                     Called   At          Resolved To
                    ------------------      -----------
Procedure Name    Seg Num   STT Num           LIB
=====================================================
P'INITHEAP'3000    %00000      6             SYSTEM
```



```
  ****  O F F S E T S  ****

%000016

Total Number of calls in segment 0: 1

Total number of calls to P'INITHEAP'3000:      1
Number of segments which call P'INITHEAP'3000: 1
```

```
                        Called   At          Resolved To
                      ------------------      -----------
Procedure Name    Seg Num   STT Num             LIB
=================================================
P'RESET            %00000      3              SYSTEM

                ****  O F F S E T S  ****
     %000052

     Total Number of calls in segment 0: 1

Total number of calls to P'RESET:      1
Number of segments which call P'RESET: 1

                        Called   At          Resolved To
                      ------------------      -----------
Procedure Name    Seg Num   STT Num             LIB
=================================================
P'REWRITE          %00000      4              SYSTEM

                ****  O F F S E T S  ****
     %000035

     Total Number of calls in segment 0: 1

Total number of calls to P'REWRITE:      1
Number of segments which call P'REWRITE: 1

                        Called   At          Resolved To
                      ------------------      -----------
Procedure Name    Seg Num   STT Num             LIB
=================================================
PROCA              %00000      7              GROUP

                ****  O F F S E T S  ****
     %000067

     Total Number of calls in segment 0: 1

Total number of calls to PROCA:      1
Number of segments which call PROCA: 1

                        Called   At          Resolved To
                      ------------------      -----------
Procedure Name    Seg Num   STT Num             LIB
=================================================
TERMINATE'         %00000      2              SYSTEM
```

```
                ****  O F F S E T S  ****

     %000073

     Total Number of calls in segment 0: 1

Total number of calls to TERMINATE':      1
Number of segments which call TERMINATE': 1

External references inherited from the group SL (SL.SOOL.MPEV).
                    Called   At          Resolved To
                  ------------------      -----------
Procedure Name    Seg Num   STT Num           LIB
=====================================================
CREATEPROCESS      %00000      2             SYSTEM

                ****  O F F S E T S  ****

     %000072

     Total Number of calls in segment 0: 1

Total number of calls to CREATEPROCESS:      1
Number of segments which call CREATEPROCESS: 1

                    Called   At          Resolved To
                  ------------------      -----------
Procedure Name    Seg Num   STT Num           LIB
=====================================================
FCONTROL           %00000      3             SYSTEM

                ****  O F F S E T S  ****

     %000032

     Total Number of calls in segment 0: 1

Total number of calls to FCONTROL:      1
Number of segments which call FCONTROL: 1

    P O T E N T I A L    I N C O M P A T I L I T I E S

Incompatibilities detected in the program file "STICP.SOOL.MPEV".

FCONTROL: *** System Defined Incompatibility ***
  - Some FCONTROL controlcodes are no longer valid
    on MPE/iX (CM/NM 500).
```

```
 Incompatibilities detected in the group SL (SL.SOOL.MPEV).

 CREATEPROCESS: *** System Defined Incompatibility ***
   - CREATPROCESS may encounter byte pointer problems
     in NM on MPE/iX (NM 107).

 FCONTROL: *** System Defined Incompatibility ***
   - Some FCONTROL controlcodes are no longer valid
     on MPE/iX (CM/NM 500
     These controlcodes include: 03,48.

    S U M M A R Y    I N F O R M A T I O N

 NM Only Incompatibilities:        1
 CM/NM Incompatibilities:          2
 Uncallable Procedures Detected:   0

    I N T R I N S I C    M E C H A N I S M    I N F O R M A T I O N

Applications being recompiled to native mode must use the intrinsic
mechanism for system intrinsics.

Number of procedures requiring the intrinsic mechanism: 3

Procedures from STICP.SOOL.MPEV requiring the intrinsic mechanism:
FCONTROL

Procedures from SL.SOOL.MPEV requiring the intrinsic mechanism:
CREATEPROCESS    FCONTROL
```

Figure B-8. Sample Detailed OCA Report Page 5

OCA Program Error Messages

There are two classes of program error messages generated by OCA:

- User input errors that cause syntax and semantic errors
- Internal error messages

User Input Errors

OCA generates and displays to `$STDLIST` error messages for any syntax and semantic errors associated with user input. OCA follows these guidelines when generating this class of error messages:

- Displays file system error messages.
- Echoes user input.
- Attempts to point to the error with a carat (^).
- Numbers the errors in the order they were discovered (usually left to right).
- Identifies the file that OCA was reading when the error occurred and prints the name at the end of the error message. ("****" is printed instead of`$STDIN` or `$STDINX`.)

Figure B-9 illustrates incorrect user input at the SCAN prompt, and the resulting user input errors that are displayed to the screen. The user responds to the SCAN prompt with the indirect file `IFILE`. Errors occur when OCA scans the first line in `IFILE`, `BADFILG;;LIB;&`. Next, the user responds to the SCAN prompt with the file name `NOTFILE;` a file system error is displayed.

```
SCAN>    ! IFILE;LIB=G  --  read from IFILE.PUB.PERS
BADFILE;;LIB;         ^1  ^2,3
             ^4,5
ERROR 1:  Missing option.  Expected one of LIB, BRIEF, or
          DETAILED (IFILE.PUB.PERS)
ERROR 2:  Missing = after LIB (IFILE.PUB.PERS)
ERROR 3:  Missing G, P, or S after = (IFILE.PUB.PERS)
ERROR 4:  Missing option. Expecting one of LIB, BRIEF, or
          DETAILED (IFILE.PUB.PERS)
ERROR 5:  Extraneous input after command (IFILE.PUB.PERS)
SCAN>NOTFILE
NONEXISTENT PERMANENT FILE (FSERR 52)
NOTFILE
   ^1
ERROR 1:  Unable to open NOTFILE.PUB.PERS ****
```

Figure B-9. Sample OCA Error Message Display

Internal Error Messages

The following section lists internal program error numbers and the corresponding messages generated by OCA. This error information contains:

- Internal error numbers
- The complete error message
- The probable cause of the error
- Action to take to correct the error

101	MESSAGE	INTERNAL ERROR 101: CATREAD failed in REPORT. Display of message with NLS failed. Msg_num = *nnn* Set_num = *nnn* CATREAD error = *nnn*
	CAUSE	Attempting to display a message from the catalog file via the REPORT procedure. CATREAD failed for some reason. Possible causes of the CATREAD failure are explained in the *MPE V/E Native Language Support Reference Manual (32414-190001).*
	ACTION	This error is likely to occur if the OCA catalog has been incorrectly localized. If the catalog name is OCCAT000.PUB.SYS, then this may be because of a bug in OCA, not a problem with localization. You should attempt to verify that the catalog file is not corrupt by restoring it from the product tape. If this failure persists, please contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.
102	MESSAGE	INTERNAL ERROR 102: NLSCANMOVE error. NLSCANMOVE error *nnn* Attempting to NLUPSHIFT. (or) NLSCANMOVE error *nnn* Attempting to NLDOWNSHIFT.
	CAUSE	NLSCANMOVE intrinsic failed attempting to either upshift or downshift a string. Possible causes of the NLSCANMOV E failure are explained in the *MPE V/E Native Language Support Reference Manual (32414-90001).*
	ACTION	Ask your System Manager to confirm that NLS has been correctly configured for your system (especially for the language associated with the number assigned to the JCW NLUSERLANG). If you are not using a localized version of OCA then this should not be an issue. Please contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.

103	MESSAGE	INTERNAL ERROR 103: CATREAD failed during NLDISPLAY. Display of message with NLS failed. Msg_num=*nnn* Set_num=*nnn* CATREAD error=*nnn*
	CAUSE	Attempting to display a message from the catalog file via the NLDISPLAY procedure. CATREAD failed for some reason. Possible causes of the CATREAD failure are explained in the *MPE V/E Native Language Support Reference Manual (32414-90001).*
	ACTION	This error is likely to occur if the OCA catalog has been incorrectly localized. If the catalog name is OCCAT000.PUB.SYS, then this may be because of a bug in OCA, not a problem with localization. You should attempt to verify that the catalog file is not corrupt by restoring it from the product tape. If this failure persists, please contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.
104	MESSAGE	INTERNAL ERROR 104: CATREAD failed during NLPROMPT. Display of message with NLS failed. Msg_num=*nnn* Set_num=*nnn* CATREAD error=*nnn*
	CAUSE	Attempting to display a message from the catalog file via the NLPROMPT procedure. CATREAD failed for some reason. Possible causes of the CATREAD failure are explained in the *MPE V/E Native Language Support Reference Manual (32414-90001).*
	ACTION	This error is likely to occur if the OCA catalog has been incorrectly localized. If the catalog name is OCCAT000.PUB.SYS, then this may be because of a bug in OCA, not a problem with localization. You should attempt to verify that the catalog file is not corrupt by restoring it from the product tape. If this failure persists, please contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborts.

105	MESSAGE	INTERNAL ERROR 105: Unable to retrieve localized string from OCCAT*nnn*.PUB.SYS. CATREAD error nnn OCA cannot continue. (or) String read from catalog is too long. (MAX=*nnn* Chars.) String: *the actual string read from catalog(nnn)*
	CAUSE	Attempting to CATREAD a localizable string from the OCA catalog, a problem is encountered. The problem is either a CATREAD error (which can be diagnosed by looking up the CATREAD error number in the *MPE V/E Native Language Support Reference Manual (32414-90001)*) or a problem with the length of an internal buffer passed inside of OCA.
	ACTION	This error is likely to occur if the OCA catalog has been incorrectly localized. If the catalog name is OCCAT000.PUB.SYS, then this may be because of a bug in OCA, not a problem with localization. You should attempt to verify that the catalog file is not corrupt by restoring it from the product tape. If this failure persists, please contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.
106	MESSAGE	INTERNAL ERROR 106: Unable to find keyword in catalog message. Catalog message: *actual message read from catalog* Keyword: *expected keyword* Message number: *nnn*
	CAUSE	After reading a localizable message from the catalog, it was searched for a keyword. This keyword was not found. The message read from the catalog and the expected keyword are included in the error message.
	ACTION	Contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.
107	MESSAGE	INTERNAL ERROR 107: Cannot Obtain NLUPSHIFT Table For Language *nnn.* NLINFO Error nnn during SETUP.
	CAUSE	NLS intrinsic NLINFO returned an error when the UPSHIFT table for language *nnn* was requested. Possible causes of the NLINFO failure are explained in the *MPE V/E Native Language Support Reference Manual (32414-90001).*
	ACTION	Ask your system manager to confirm that NLS has been correctly configured for your system (especially for the language associated with the number assigned to the JCW NLUSERLANG). If you are not using a localized version of OCA,

108	MESSAGE	`INTERNAL ERROR 108: Cannot Obtain NLDOWNSHIFT Table For Language` *nnn*. `NLINFO Error` *nnn* `during SETUP.`
	CAUSE	NLS intrinsic **NLINFO** returned an error when the **DOWNSHIFT** table for language nnn was requested. Possible causes of the **NLINFO** failure are explained in the *MPE V/E Native Language Support Reference Manual (32414-90001).*
	ACTION	Ask your system manager to confirm that NLS has been correctly configured for your system (especially for the language associated with the number assigned to the JCW **NLUSERLANG**). If you are not using a localized version of OCA, then this should not be an issue. Contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.
109	MESSAGE	`INTERNAL ERROR 109: Unable to get localized yes/no string.` `NLINFO error` *nnn* `Attempting to get NLS yes/no for language` *nnn*.
	CAUSE	NLS intrinsic **NLINFO** returned an error when the localized yes/no string for language *nnn* was requested. Possible causes of the **NLINFO** failure are explained in the *MPE V/E Native Language Support Reference Manual (32414-90001).*
	ACTION	Ask your system manager to confirm that NLS has been correctly configured for your system (especially for the language associated with the number assigned to the JCW **NLUSERLANG**). If you are not using a localized version of OCA then this should not be an issue. Contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.

110	MESSAGE	`INTERNAL ERROR 110: NLAPPEND Error` *nnn*.
		`Unable To Open Catalog File OCCAT`*nnn*`.PUB.SYS.`
		`OCA Cannot Continue.`

	CAUSE	**NLAPPEND** intrinsic returned error *nnn*. Possible causes of the **NLAPPEND** failure are explained in the *MPE V/E Native Language Support Reference Manual (32414-90001).*
	ACTION	Ask your system manager to confirm that NLS has been correctly configured for your system (especially for the language associated with the number you assigned to the JCW **NLUSERLANG**). If you are not using a localized version of OCA, then this should not be an issue. Contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.

111	MESSAGE	`INTERNAL ERROR 111: CATOPEN Error` *nnn*`.`
		`Unable To Open Catalog File OCCAT`*nnn*`.PUB.SYS.`
		`OCA Cannot Continue.`

	CAUSE	`CATOPEN` intrinsic returned error *nnn* when attempting to open the OCA catalog file. Possible causes of the `CATOPEN` failure are explained in the *MPE V/E Native Language Support Reference Manual (32414-90001).*
	ACTION	Make sure the catalog file mentioned in the message exists in the correct group and account, and has the current file code (MGCAT). If it does not, then obtain a copy of the correct formatted catalog file from your System Manager. If you cannot take care of the problem in this manner, contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.
200 to 299	MESSAGE	200 ... 299 Message (Various Messages)
	CAUSE	Corruption of internal data structures.
	ACTION	Contact a Hewlett-Packard Support Representative.
300	MESSAGE	`INTERNAL ERROR 300: Duplicate/ambiguous keyword in table.`
	CAUSE	An attempt was made to add a duplicate of ambiguous entry to one of the OCA keyword tables.
	ACTION	This error is likely to occur if the OCA catalog has been incorrectly localized. If the catalog name is `OCCAT000.PUB.SYS`, then this may be because of a bug in OCA, not a problem with localization. You should attempt to verify that the catalog file is not corrupt by restoring it from the product tape. If this failure persists, please contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.
301	MESSAGE	`INTERNAL ERROR 301: Attempt to insert too many keywords in table.`
	CAUSE	One of OCA's internal keyword tables is full and an attempt has been made to make another entry. This table needs to be expanded.
	ACTION	Contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.

302	MESSAGE	INTERNAL ERROR 302: Bad keyword index.
	CAUSE	Attempting to access an entry in one of OCAs internal keyword tables using an invalid index.
	ACTION	Contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.
303	MESSAGE	INTERNAL ERROR 303: SL directory entry points to deleted segment. Bad SL file, *n*1, detected.
	CAUSE	OCA has determined that the directory contained in the SL file it scanned is corrupted. Specifically it determined that a directory entry points to an entry point which has been deleted from that SL file. This is not an internal error in OCA, but rather an internal error in the SL file scanned by OCA.
	ACTION	Do not contact a Hewlett-Packard Support Representative unless the SL file corruption persists, and you can determine what is causing the corruption. In that case, submit the Service Request against the product which is corrupting the SL file, not against OCA. The best corrective action is to rebuild the SL file. If this is not possible (because you do not have access to the source or the USL file), contact the creator of the SL file to get a new copy.
304	MESSAGE	INTERNAL ERROR 304: I/O error occurred while writing to the report file (OCALIST).
	CAUSE	When attempting to write (either via FWRITE or CATREAD) a message to the formal file designator OCALIST, an I/O error occurred. This will have occurred either during the generation of an OCA report or during the printing of OCA HELP text. In either case, if you specified the ;OFFLINE option, the characteristics of the file OCALIST can be controlled by issuing an MPE :FILE equation. If you did not specify ;OFFLINE, then your output is being directed to $STDLIST, and :FILE equations will be ignored.
	ACTION	If the ;OFFLINE option was specified, then check to see if a :FILE equation is in effect by issuing the MPE command :LISTEQ. Look for a :FILE equation for OCALIST. The most common cause of this problem is encountering End Of File on OCALIST when OCALIST is redirected to a disk file. If that is the case, use the ;DISC= option on the :FILE command to increase the limit on the disk file. If you did not specify ;OFFLINE or cannot figure out how to correct the problem with a file equation contact your Hewlett-Packard Support Representative.

305	MESSAGE	`INTERNAL ERROR 305: FREADDIR intrinsic failed.`
	CAUSE	The most likely cause of this problem is either program or SL file corruption (of the program or SL file being scanned, not of the OCA program file). If you are scanning a program file when you get this error, you can double check to see if it is corrupt by attempting to `:RUN` the program. If it is corrupted, the MPE loader will probably print a similar message to the one printed by OCA. If you are scanning a program file with the `;LIB=` option and get this error, then the corruption may be in the SL file, not the program file. The error message should indicate which file is corrupt. Again, you can verify that the file is corrupted by running the program with the same ;LIB= parameter. Hewlett-Packard does not suggest that you attempt to run the program if it contains any Privileged Mode code and you get this error from OCA. Privileged Mode code could be dangerous if the program file is indeed corrupted.
	ACTION	Do not contact a Hewlett-Packard Support Representative unless the file corruption persists, and you can determine what is causing the corruption. In that case, submit the Service Request against the product which is corrupting the file, not against OCA. The best corrective action is to rebuild the program or SL file. If this is not possible (because you do not have access to the source or the USL file), contact the creator of the SL file to get a new copy.
306	MESSAGE	`INTERNAL ERROR 306: FFILEINFO intrinsic failed on` *filename*.
	CAUSE	The MPE `FFILEINFO` intrinsic failed when attempting to get information about the file *filename*.
	ACTION	Contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted.
307	MESSAGE	`INTERNAL ERROR 307: FCLOSE failed on SL file` *filename*.
	CAUSE	OCA attempted to close the SL file *filename* using the MPE `FCLOSE` intrinsic. The `FCLOSE` failed. This message will be followed by a file system error message which should give you a better idea as to why the `FCLOSE` failed.
	ACTION	Contact a Hewlett-Packard Support Representative. Include all information printed by OCA when it aborted. Please note that this error does not invalidate in any way the results which were printed by OCA. The reports and migration information contained within the reports will still be accurate.

C

Using the Run Time Monitor

This appendix describes how to use RTM (Version A.00.01). RTM is intended to be used in conjunction with its companion Migration Toolset utilities, MPT and OCA, during the analysis and planning stage of the migration effort. RTM runs on MPE V/E-based systems. Included in this appendix are discussions on the following subjects:

- "Run Time Monitor Operation", a general overview on how the tool operates.
- "Setting up Run Time Monitor", how to install RTM on your MPE V/E-based system.
- "Using RTMSYS", how to enable the RTM event monitoring facility.
- "Using RTMREP", how to generate RTM event reports.
- "Run Time Monitor Program Error Messages", a comprehensive list of program error and warning messages, as well as recommended methods of error recovery.

Run Time Monitor Operation

RTM is a tool designed to aid in migrating applications from an MPE V/E-based system to an MPE/iX-based system. The task of RTM is to monitor executing applications for occurrences of predefined events. An event is defined as an MPE V/E intrinsic feature that will be incompatible when executed on an MPE/iX-based system.

RTM is made up of three parts:

- A controlling program, `RTMSYS.PUB.SYS`, used to enable RTM and to specify the event classes to monitor.
- A report program, `RTMREP.PUB.SYS`, used to generate reports for monitored applications.
- An SL, `RTMSL.PUB.SYS`, used by RTM to control event monitoring.

RTM uses the MPE V/E System Logging facility to log events during program execution. Therefore, system logging must be enabled for RTM to log events. To determine if RTM is operating, execute an event that you know will be logged, close the current log file and create a new log file with `:SWITCHLOG`, and generate a report with RTMREP.

RTMSYS

A user with SM or OP capability can use RTMSYS to enable or disable RTM and monitoring of specific event classes. These classes are described in Table C-1.

Table C-1. Run Time Monitor Event Classes

Class Number	Class Name	Description
0	Logging Enabled	Enable/disable RTM operation
1	NM Events	Enable disable logging of uniquely NM events (in addition to events common to both CM and NM.
2	Command Intrinsic Events	Enable/disable logging of any incompatible command executed by the `COMMAND` intrinsic. Commands are not associated with a specific mode. Therefore, the NM Events (class 1) does not effect this event class.
3	`FFILEINFO` Intrinsic Events	Enable/disable logging of `FFILEINFO` related incompatibilities. There are several differences relevant to `FFILEINFO` mainly associated with device information.
4	`FGETINFO` Intrinsic Events	Enable/disable logging of `FGETINFO` related incompatibilities.
5	`FCONTROL` Intrinsic Events	Enable/disable logging of `FCONTROL` related incompatibilities.
6	General File System Events	Enable/disable logging of all other file system incompatibilities.
7		Reserved for future use.
8	Miscellaneous Events	Enable/disable logging of events that do not fall under the other event classes.

RTMREP A report of events can be generated by using the RTMREP program. RTMREP reads the specified system log files and generates a report to the formal file designator `RTMLIST` of the events found. Two report formats are available:

- A brief report (the default) is a program summary which shows the events generated by each program. A counter shows the number of times that a particular call caused the event. The report is sorted by date, program name, and event class.
- A detailed report shows all events logged, sorted by date, program name, and event class. All information in the log record is formatted and displayed in chronological order.

Appendix D "Incompatibilities" contains detailed descriptions of all events detected by RTM and reported in RTMREP. The descriptions include possible causes of the event, as well as recommended actions you can take to work around the detected incompatibility.

RTMSL RTM uses an SL file, `RTMSL.PUB.SYS`, to find areas of incompatibility by intercepting calls to specific system procedures. The SL is bound to the program at load time and allows you to control which events are logged. The relevant information is logged by RTM, which then completes the procedure call by calling the actual procedure in `SL.PUB.SYS`. With RTM enabled, the LOADER inserts `RTMSL.PUB.SYS` into the load sequence when any program not residing in `PUB.SYS` is executed or allocated.

Managing Run Time Monitor Stack Space Usage

When RTM is enabled, your programs will use slightly more stack space. The amount of additional stack depends on the intrinsics called by your program. Intrinsics with more parameters will incur more stack overhead when RTM is enabled. The additional stack overhead ranges from 8 to 60 words. Your applications will use more stack from the time the intrinsic is called until control is returned to your program. In other words, your application does not pay a penalty in stack overhead during the entire time it is executing, and the stack overhead is not cumulative for all the intrinsics your application calls. Instead, the additional overhead impacts your application only when RTM intercepts the call to a potentially incompatible procedure. Nevertheless, it is possible that the additional stack space used by RTM will cause your program to abort with a stack overflow message. If this happens you have 2 options.

1. You can attempt to run your program using the `;NOCB` option on the `:RUN` command. You will also want to make sure you have specified the maximum value for the `;MAXDATA=` option on the run command or when the program was prepped with the `:PREP` command.

   If this option does not work or is too cumbersome, because you would have to modify several system UDCs to make it work, attempt to use option 2.

2. You can still use RTM to monitor other programs on your system while continuing to use the application which is aborting. To do this you, should disable RTM and make sure the program in question is not being run and is not allocated. Having accomplished this you may allocate the program using the MPE `:ALLOCATE` command and enable RTM. Programs which are allocated before RTM is enabled will not be loaded through `RTMSL.PUB.SYS` and therefore, will not

incur any overhead due to RTM. Unfortunately, RTM will not be able to monitor these programs so you will need to rely on OCA for your migration information.

Managing Run Time Monitor Disk Space Usage

RTM uses the MPE V/E System Logging facility to keep track of migration events as they occur on your system. As log file records are generated, RTM will consume disk space on your system. The rate of consumption varies based on several different factors. These factors include the following:

- Other activities also being logged on your system (for example, console logging).
- System load (for example, numbers of users, extent of activity on your system).
- Which RTM event classes have been enabled using RMTSYS.PUB.SYS.
- Period of operation of RTM.
- Number of migration issues encountered in the programs being monitored.

If you are concerned about disk space usage on your system you may wish to monitor the system log files when you first enable RTM (or when you change the event classes being monitored via `RTMSYS.PUB.SYS`). One easy way to do this is to issue the following `:LISTF` command.

```
:LISTF LOG####,PUB,SYS,2
ACCOUNT= SYS   GROUP= PUB

FILENAME CODE              LOGICAL RECORD                    SPACE
                          SIZE TYP     EOF    LIMIT R/B    SECTORS #X MX

LOG1327                  1022W VB  27165     1023   1      8192    16 16
LOG1328                  1022W VB  13012     1023   1      4096     8 16
LOG1329                  1022W VB   5402     1023   1      2048     4 16
```

By totaling the SECTORS column you can determine how much disk space is being consumed by the system log files (in the example above, 14,336 sectors have been used).

If you find that disk space is being consumed too rapidly the following corrective steps are recommended:

1. Store system log files on tape for future analysis as they are generated.
2. Disable monitoring for event classes once you feel you have sufficient data on events in those classes.
3. Temporarily disable other types of system logging while using RTM.

Setting Up Run Time Monitor

System logging and the program file event (type 16) must be enabled through the MPE V/E SYSDUMP facility for RTM to work. (Refer to the *MPE V/E System Operation and Resource Management Manual (32033-90005)* for details on enabling system logging and the program file event.) Since any program already loaded or allocated is not linked through `RTMSL.PUB.SYS` when run-time event monitoring is enabled, it is necessary to deallocate all programs first, or make sure that RTM is enabled immediately after a system start.

Note

The auto-allocate feature of MPE V/E does not affect RTM operation. RTMSYS ensures that auto-allocate is disabled before any RTM status changes are made. Once the changes are made, RTMSYS enables auto-allocate if it was enabled prior to the change.

Caution

Running RTM on the PREDICTIVE software can cause your system to hang.

Initially, all event classes recognized by RTMSYS are off. For RTM to work, you need to turn on the desired event classes.

There are two methods available to automatically enable RTM after a WARMSTART/COOLSTART/COLDSTART:

- Through an operator logon UDC
- Through the system Startup State Configurator file

Operator Logon UDC

The first method you can use to automatically enable RTM is to have OPERATOR.SYS execute a logon UDC that runs the RTMSYS program with an info string. The info string contains the name of a file that has the event classes turned on or off, as desired. Each entry in the file looks exactly as if it was typed interactively.

```
******
EVENTUDC
OPTION LOGON
: RUN RTMSYS.PUB.SYS;info="RTEVENTS.PUB.SYS"
******
```

The file RTEVENTS.PUB.SYS is a standard EDIT/V compatible file, either numbered or unnumbered, that you create. Each line contains a class number and the desired status for that particular class.

System Startup File

The second method you can use to enable RTMSYS by is streaming a job stream file from the Startup State Configurator file, SYSTART.PUB.SYS. A job stream file must be used since the :RUN command can not execute from the Startup State Configurator file.

A sample job stream file to be invoked from a startup state file:

```
!JOB RTMINIT,OPERATOR.SYS
!RUN RTMSYS.PUB.SYS;INFO="RTEVENTS.PUB.SYS"
!TELLOP; **** RTM ENABLED
!EOJ
```

Using RTMSYS

RTM is controlled by the program RTMSYS. RTMSYS can be run interactively or in batch mode by any user possessing OP or SM capability by entering:

```
: RUN RTMSYS.PUB.SYS
```

You only need to run RTMSYS to change the status of an event class.

Figure C-1 charts the flow of control with RTMSYS prompts. Additional information on the prompts is provided in subsequent headings. Defaults for all prompts are shown in upper case.

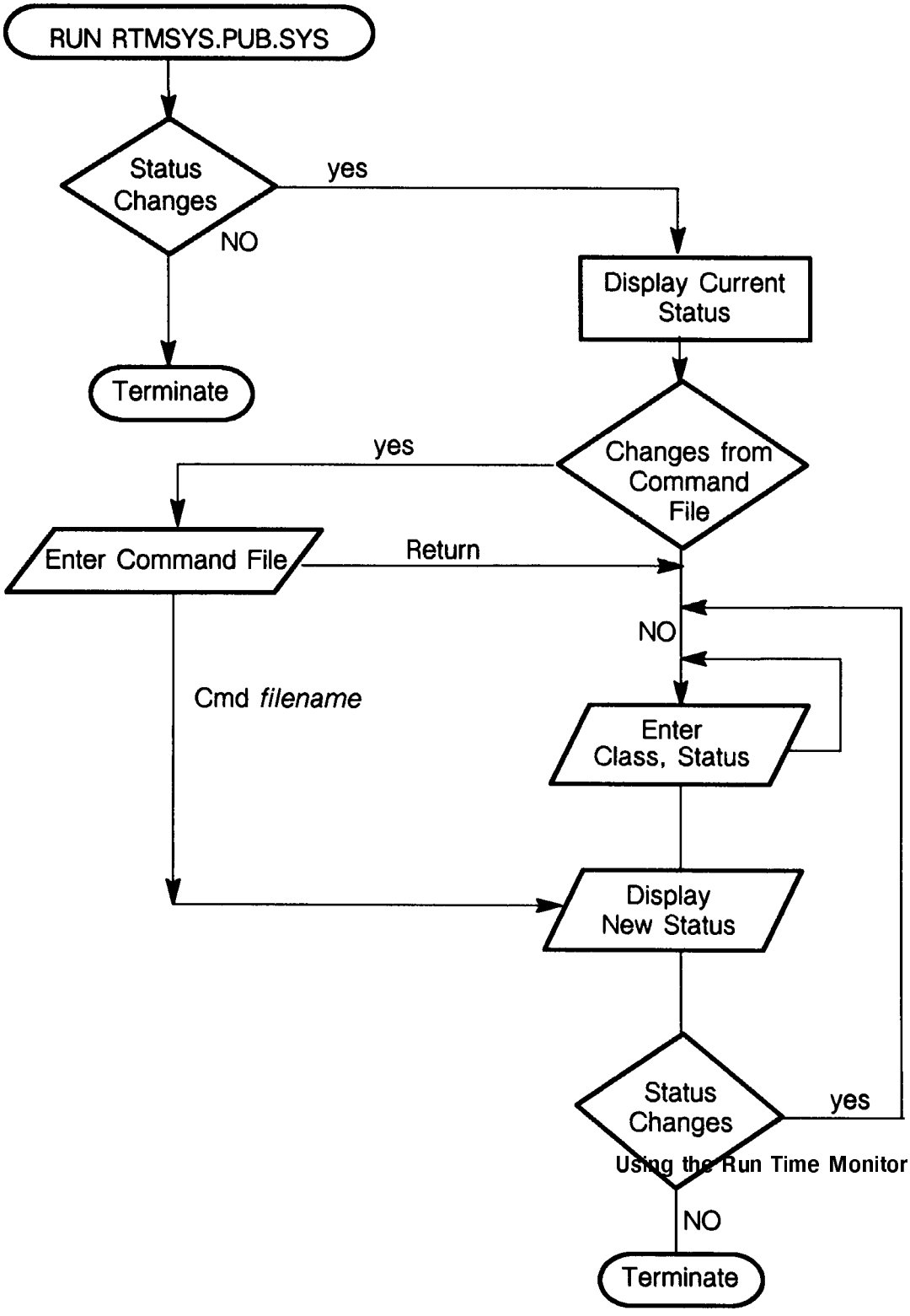


Figure C-1. RTMSYS

Status Changes Prompt

RTMSYS initially displays all event classes and their current status. The first prompt asks you if you want to make changes to the classes of events that RTM detects:

- A positive response causes RTMSYS to continue and takes you to the command file prompt.
- A negative response, or Return, terminates RTMSYS

```
HP30364A.00.00 Run Time Monitor (C) HEWLETT PACKARD COMPANY 1986

Enter a question mark (?) at any prompt for help.
Default answers are shown in upper case.

Do you want to make RTM status changes [yes/NO]?


CLASS DESCRIPTION      STATUS

0 - LOGGING ENABLED     OFF
1 - Native Mode EVENTS  OFF
2 - COMMAND INTRINSIC   OFF
3 - FFILEINFO           OFF
4 - FGETINFO            OFF
5 - FCONTROL            OFF
6 - GENERAL FILESYS     OFF
7 - RESERVED
8 - MISCELLANEOUS       OFF
```

Command File Prompt

```
Do you want to make changes from a command file [yes/NO]?
```

You are being asked if you have a file that contains information regarding the classes of events that you want to alter the status of.

- A positive response takes you to the Enter Command File prompt.
- A negative response (the default), takes you to the Enter Class prompt.

Enter Command File Prompt

```
Enter name of command file:
```

At this prompt, you should enter the name of the command file. Valid responses include any valid file name of a file containing the event classes and desired status. This file can be any standard, EDIT/3000 compatible file, either numbered or unnumbered, that contains records indicating the class number and desired status. Each line contains a class number and the desired status for that particular class.

```
1       0,ON
2       1,ON
3       5,OFF
```

Enter Class Prompt

```
ENTER CLASS, ON/OFF?
```

You are being prompted for event class changes. To change the status of an event class, enter the class number and the desired status, ON or OFF, separated by a comma. Continue to enter class numbers and desired status until all changes have been made. When all desired changes have been entered, press Return. This returns you to the Status Changes prompt.

Exiting RTMSYS

A negative response, or Return, at the Status Changes prompt causes RTMSYS to terminate.

Running RTMSYS in Batch Mode

Figure C-2, Figure C-3, and Figure C-4 are three sample job stream files that illustrate how you can enable and disable events from batch mode, and generate a report on all programs run from a specified account.

```
:job rtmstart, manager.sys, pub
:comment
:comment check to see if rtmstart.pub.sys exists
:comment
:setjcw cierror=0
:listf rtmstart.pub.sys
:comment
:comment if rtmstart.pub.sys does not exist then create it
:comment
:if cierror = 907 then
:editor
a
0,on
1,on
2,on
3,on
4,on
5,on
6,on
7,on
8,on
//
k rtmstart.pub.sys
e
:endif
:comment
:comment run rtmsys.pub.sys with command file created above
:comment
:run rtmsys.pub.sys;info="rtmstart.pub.sys"
:eoj
```

Figure C-2. Job Stream to Enable RTM for All Event Classes

```
:job rtmstop, manager.sys, pub
:comment
:comment check to see if rtmstart.pub.sys exists
:comment
:setjcw cierror=0
:listf rtmstop.pub.sys
:comment
:comment if rtmstop.pub.sys does not exist then create it
:comment
:if cierror = 907 then
:editor
a
0,off
1,off
2,off
3,off
4,off
5,off
6,off
7,off
8,off
//
k rtmstop.pub.sys
e
:endif
:comment
:comment run rtmsys.pub.sys with command file created above
:comment
:run rtmsys.pub.sys;info="rtmstop.pub.sys"
:eoj
```

Figure C-3. Job Stream to Disable RTM for All Event Classes

```
:job rtmrep, manager.sys, pub
:comment
:comment  The responses to the RTM prompts below have the
:comment  following meanings:
:comment
:comment            Response                              Meaning
:comment
:comment            :file rtmlist;dev=epoc        redirect the report to the
:comment                                          epoc printer.
:comment            n                             do not want a detail line for
:comment                                          each incompatibility detected
:comment            1300                          start with log file 1300
:comment            1305                          end with log file 1305
:comment            y                             subset the data
:comment            @.@.finance                   report on all programs run
:comment                                          out of the finance account
:comment            1,2,3,4,5                     report on events in event
:comment                                          classes 1 through 5
:comment
:run rtmrep.pub.sys
:file rtmlist;dev=epoc
n
1300
1305
y
@.@.finance
1,2,3,4,5
:eoj
```

Figure C-4. Job Stream to Generate a Report for All Finance Programs

Using RTMREP

RTM does not automatically generate reports. However, all run-time events that are captured are recorded in a system log file. A report of events logged can be generated by running RTMREP. Descriptions of the events logged by RTMSYS are found in Appendix D, "Incompatibilities".

There are two report formats.

- A program summary (the default), a brief report that shows all events generated by each program. A counter shows the number of times that a particular segment/offset caused the event. The report is sorted by date, program name, and event class.
- A detailed report shows all events logged. These events are sorted by date, program name, and event class. All information in the log record is formatted and displayed in chronological order.

Note

RTMREP cannot access the current system log file. If you want to generate a report from the current system log file, you must first close it with the `:SWITCHLOG` command.

To generate a report, enter the following:

```
:RUN RTMREP.PUB.SYS
```

:RTMREP prompts you for the choice of a program summary or detailed report.

Additionally, `LISTLOG5.PUB.SYS` can be used to examine the log file for entries. Refer to the *MPE V/E Utilities Reference Manual (32033-90008)* for more information on `LISTLOG5`.

RTMREP has a HELP facility can be invoked at any time by entering `?`. An explanation of the current prompt is then displayed, followed by the same prompt.

You may also issue any MPE V/E command that can be executed programmatically be prefixing it with a colon (:).

Output Device Specification

The output formal file designator is `RTMLIST`, which by default is opened with device class `LP`. The `:FILE` command can be used to redirect the output as necessary.

Figure C-5 charts the flow of control with RTMREP prompts. The prompts are described under the following headings. Defaults for all prompts are shown in upper case.

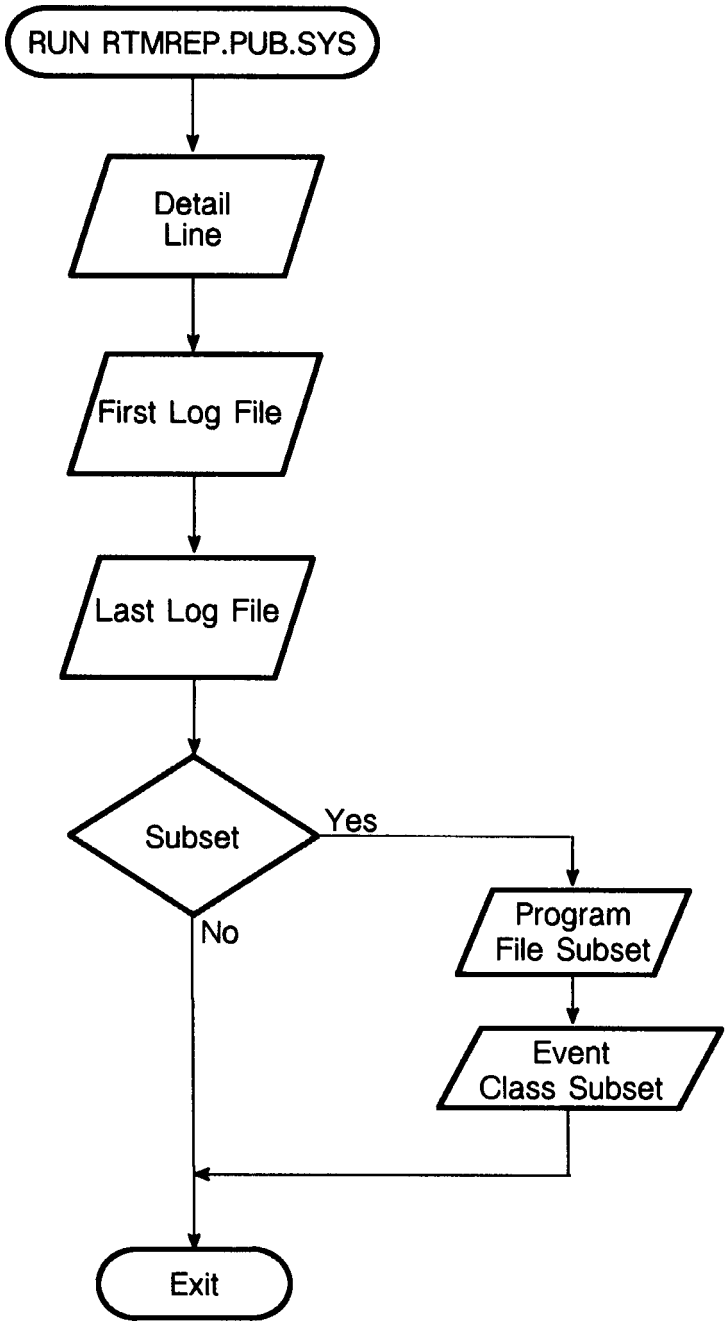


Figure C-5. RTMREP

Detail Line Prompt

```
HP30364A.00.00 Run Time Monitor (C)
HEWLETT PACKARD COMPANY 1986

Enter a question mark (?) at any prompt for help.
Default answers for all prompts are shown in upper case.

Do you want a detail line for each event [yes/NO]?
```

- A positive answer to this prompt gives you the detailed report.
- A negative answer (the default) to this prompt gives you the summary report. Either response takes you to the Log File Specification prompts.

Log File Specification Prompt

```
You may specify a range of log files to be
analyzed for the report
Enter first log file:
Enter last log file:
```

To examine a range of log files, type the starting log file number and the last log file number.

To examine only one log file, enter the log file number in response to the Enter First Log File prompt and a Return to the Enter Last Log File number.

Either response takes you to the Subset prompt.

Subset Prompt

```
Do you want to report on a subset of available data [yes/NO]?
```

- A positive response allows you to tailor the report to your needs by selecting specific a subset of programs. This takes you to the Enter Program File Subset prompt.

- A negative response (the default) reports on all available information.

Enter Program File Subset Prompt

```
Enter program file subset [CR will select all]:
```

You are being prompted for the name of a file(s) you want RTMREP to create a report about. In addition, you can specify a set of program files using wild card characters (@, #, ?) in conjunction with MPE V/E file specifications. For example, the following file set specification indicates all program files that start with the letter "C":

```
C@.MYGROUP.MYACCT
```

Enter Class Numbers Prompt

```
Enter class numbers, separated by commas, or CR for all:
```

You are being prompted for the event classes you want included in the report RTMREP creates.

Exiting RTMREP

After report specifics are complete, RTMREP prints the report, then exits.

Sample Report

The following lists a sample RTM report. The numbers in the `EVENT` column correspond to message numbers listed in Appendix D, "Incompatibilities".

Note

RTM report format uses 132 columns. The report that follows has been truncated to 80 columns to fit this page. A complete report would include a field to the far right listing a brief description of the incompatibility.

```
1HP30364X.01.02  RUN TIME EVENTS BY PROGRAM FILE    LOG DATE: THU, OCT 23,1986
OPROGRAM FILE          TIME      #J/S      SEGMENT     DELTAP  STATUS  EVENT

 STICP.SOOL.MPEV      10:29 AM   #S127     GSL  %000   000072  060002  107
                      10:29 AM   #S127     GSL  %000   000032  060002  503
                      10:29 AM   #S127     PROG %000   000061  060001  503
```

Run Time Monitor Program Error Messages

The following list of error and warning messages are generated by the two programs that make up the RTM utility, RTMSYS and RTMREP.

12	MESSAGE	`Classes must be numeric in the range 1..8`
	CAUSE	In RTMREP you have selected a subset class that was either not a numeric value or was beyond the specified range (1 . . . 8)
	ACTION	Respecify the selection you want: The correct syntax is discussed under "Enter Class Numbers Prompt".
14	MESSAGE	`Is incomplete - a maximum of 100000 records per run allowed.`
	CAUSE	In any single run of RTMREP, the program can only sort 100,000 records of selected migration events. You have selected a subset of log files and events that has exceeded this amount.
	ACTION	Analyze a subset of the data by splitting the analysis into two or more smaller analyses. You can analyze a subset of events classes in the same number of log files or analyze the same number of event classes, but reduce the number of log files selected for each analysis.
31	MESSAGE	`User must have OP or SM capability to run RTMSYS.`
	CAUSE	You need to have OP or SM capability in order to run RTMSYS.
	ACTION	Sign on to an account or user that has either OP or SM capability, or have your system manager increase your account and user capabilities to include one or the other of OP or SM.
39	MESSAGE	`Changes from Command File not made due to file error.`
	CAUSE	RTMSYS was unable to open the file name you specified as a Command File. RTMSYS was attempting to open the file as a permanent ASCII file (`FOPEN`*foptions* = %5).
	ACTION	Examine the file system information displayed prior to this message and either enter your changes manually, respecify the file name, or create a new Command File and try again. Entering command file names is discussed under "Command File Prompt"

41	MESSAGE	`Input must be in the form: "1, ON" or "1, OFF".`
	CAUSE	RTMSYS was expecting to find a comma between the two tokens you provided.
	ACTION	Respecify your input with a comma between the event class and the requested status (ON or OFF). The correct syntax is discussed under "Enter Class Prompt".
42	MESSAGE	`Class must be numeric, for example "1,ON".`
	CAUSE	RTMSYS found that the first token you specified was not a numeric character.
	ACTION	Respecify your input with the correct numeric value. The correct syntax is discussed under "Enter Class Prompt".
43	MESSAGE	`Class is out of range, valid classes are 0 through 8.`
	CAUSE	You have specified a migration event class value that is not in the range (0 ... 8) expected by RTMSYS.
	ACTION	Respecify your input with the a numeric character in the correct range (0 ... 8). The correct syntax is discussed under "Enter Class Prompt".
44	MESSAGE	`Specify "ON" or "OFF", for example "1,ON".`
	CAUSE	RTMSYS recognized the first token as being a valid event class, but did not recognize the second token as being either ON or OFF.
	ACTION	Respecify your input ensuring that the second token is either ON or OFF. The correct syntax is discussed under "Enter Class Prompt".
45	MESSAGE	`WARNING - system logging is disabled - use SYSDUMP to enable it.`
	CAUSE	In order to successfully log migration events, both overall system logging and logging class 16 (program file events) need to be enabled. This is currently not the case on the system.
	ACTION	Use the SYSDUMP facility to enable both system logging and logging class 16. Refer to the *MPE V/E System Operation and Resource Management Reference Manual (32033-90005)* for details on accomplishing this task.

46	MESSAGE	`WARNING Logging class 16 is disabled - use SYSDUMP to enable it.`
	CAUSE	Refer to Cause of Error Message 45.
	ACTION	Refer to Action of Error Message 45.
48	MESSAGE	`The VUFs of RTM segments in SL.PUB.SYS and RTMSL.PUB.SYS do not match.`
	CAUSE	RTMSYS has done internal checking of the modules necessary to run event logging. RTMSYS has discovered an inconsistency between these modules.
	ACTION	RTM may not be correctly installed on your system. Contact a Hewlett-Packard Support Representative for assistance in reinstalling the product or reporting a problem to Hewlett-Packard.
49	MESSAGE	`RTM is not enabled. Version information from RTMSL.PUB.SYS is unavailable.`
	CAUSE	Refer to Cause of Error 48.
	ACTION	Refer to Action of Error 48.
54	MESSAGE	`RTM ERROR: LOADPROC intrinsic error occurred.`
	CAUSE	Refer to Cause of Error 48.
	ACTION	Refer to Action of Error 48.
55	MESSAGE	`WARNING - RTMSL.PUB.SYS does not exist.`
	CAUSE	RTMSYS received file system error 52 when it attempted to open the file `RTMSL.PUB.SYS`.
	ACTION	Refer to Action of Error 48.

D

Incompatibilities

This appendix lists all known incompatibilities between MPE V/E and MPE/iX. These incompatibilities are divided into two categories, undetected incompatibilities and detected incompatibilities.

Undetected Incompatibilities

Incompatibilities that cannot be detected by the Migration Toolset (MPT, OCA, and RTM) are considered undetected incompatibilities. Undetected incompatibilities fall into the following groups:

- Peripheral dependent incompatibilities
- Intrinsic incompatibilities
- Command Incompatibilities
- Subsystem and compiler incompatibilities
- System Logging Incompatibilities

Peripheral Dependent Incompatibilities

Peripherals that are supported on MPE V/E-based systems may not be currently supported on MPE/iX. The *HP 3000 Computer Systems System Configuration Guide (5954-9354)* provides a list of the peripherals currently supported on MPE/iX. The following incompatibilities exist as long as the associated peripherals are not supported:

- `FREADBACKWARD` will result in a run-time error, since no tape drives with reverse read capability are supported on MPE/iX.
- `FWRITE` *controlcodes* %311, %312, %313, %314-317 are associated with devices that are not currently supported on MPE/iX.

Intrinsic Incompatibilities

The following intrinsic incompatibilities are not detected by the Migration Toolset:

- Applications that use undocumented intrinsics, execute in Privileged Mode, call routines that require Privileged Mode, or use privileged machine instructions may need to be modified when moved to the 900 Series. Only a detailed study of the program will determine this, as appropriate modifications must be made on a case-by-case basis. The Migration Toolset cannot provide sufficient details.

- Various MPE/iX subsystems require file numbers for their own use. Therefore, you should make no assumptions about file numbers returned to your program. All hardcoded file numbers should be removed and replaced with appropriate variables that contain the file number values returned by system intrinsics.
- Intrinsic parameter types and conventions have changed in NM. You should use the intrinsic mechanism to ensure compatibility.
- `FOPEN`, on MPE/iX, specifying a maximum number of extents, other than one, allows the file system to determine the number and size of extents independently.
- `FOPEN`, specifying an *foption* of zero or an *aoption* of zero always behaves as if the parameters were defaulted.
- `FOPEN`, if errors 90 and 91 are applicable at the same time for a failed `FOPEN` call, error 90 is the only error returned via the `FCHECK` intrinsic.
- `FOPEN`, the default for initial allocation is now zero extents rather than one extent in MPE V/E.
- `FOPEN`, on MPE/iX, any parameter passed to `FOPEN` must be initialized to a valid value, whether or not the intrinsic uses that parameter.

Note

For more information about intrinsic incompatibilities, refer to *Introduction to MPE XL for MPE V Programmers (30367-90005)*.

Command Incompatibilities

The Migration Toolset does not search for all command incompatibilities. Refer to the *MPE V to MPE XL: Getting Started Self-Paced Training* (30367-90002) and *Introduction to MPE XL for MPE V System Administrators* (30367-90003) for descriptions of the command changes from MPE V to MPE/iX. Note whether the iX command changes will affect current programs or UDCs and make the necessary changes.

To specify a volume set name for private volumes on MPE/iX, the keyword `ONVS=` is used, rather than the former `VS=` keyword in MPE V. Change the `VS=` parameter keyword in current programs to `ONVS=`. This keyword is used by the `ALTACCT`, `ALTGROUP`, `NEWACCT`, `NEWGROUP`, `PURGEACCT`, `PURGEGROUP`, and `REPORT` commands. Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for syntax requirements for these commands.

Subsystem and Compiler Incompatibilities

This section summarizes non-Operating System specific incompatibilities. For more information, refer to the appropriate migration guide. Other areas that have changed from MPE V/E to MPE/iX include:

- Applications which use SPL to assemble any of the following nonprivileged machine code instructions must be removed before moving to the 900 Series:
  - RMSK
  - RSW
  - RCLK
- On the 900 Series, XON/XOFF control characters are not stripped from the data stream while XON/XOFF is set OFF. Applications using this feature must be modified.
- Applications that depend on specific DSERROR codes may have to be modified, since the error codes on MPE/iX comply with the new NS3000/XL error codes.
- Pascal applications written on MPE V/E-based systems which depend on the specific data layout using the variant parts of records may require modification before recompilation into NM. This is because pointers are sized and aligned on 32-bit boundaries.
- Pascal applications which take advantage of undetected errors may have to be modified. These undetected errors are listed in the *Pascal/3000 Reference Manual (32106-90001)*.
- HP FORTRAN 77 programs that use `COMMON` or `EQUIVALENCE` should be modified so that 32-bit data types are aligned on 32-bit boundaries and so that 64-bit data types are aligned on 64-bit boundaries before being moved to NM, in order to ensure maximum performance. For more information on data alignment, refer to "Data Types Conversion" in

Introduction to MPE XL for MPE V Programmers (30367-90005).

- SPL is available in CM both for run-time support and for development of SPL applications. Your SPL applications and SPL procedures will run in CM. However, because of its close relationship to the MPE V/E-based architecture, Hewlett-Packard does not support a NM SPL compiler on the 900 Series. The OCT can be used to increase the performance of MPE V/E object code.
- High-level language applications that will be migrated to NM and that call user-supplied SPL procedures will require Switch stubs to access these procedures. For more information on Switch stubs, refer to the *Switch Programming Guide (32650-90014).*
- Procedures passed as parameters in HP FORTRAN 77 or Pascal require 16-bit parameters on MPE V/E and 32-bit parameters on MPE/iX.
- The passing of a field of a record, an element of an array, or an equivalenced variable as a reference parameter and requesting that more data be copied than is represented by the item size may produce unexpected results on MPE/iX.
- The passing of a pointer as an SPL integer or by value to a CM SPL routine.
- SPL procedures called with `OPTION VARIABLE`. Explicit parameter masks are supported in CM only. NM code calling a user SPL routine in CM must do so with a Switch stub. All NM calls to intrinsics must go through the intrinsic mechanism. For more information on Switch stubs, refer to the *Switch Programming Guide (32650-90014).*
- Variables located in programs should be properly initialized. Uninitialized variables that did not cause

problems on MPE V/E-based systems may cause programs to abort on MPE/iX-based systems.

System Logging Incompatibilities

The formats and record types of system log files have been modified. If a program processes records from a system log file, the program must be changed to specify the appropriate iX record types. Refer to *Getting System Information Programmer's Guide* (32650-90018) for a listing of the new record types and formats for the iX system log files.

Detected Incompatibilities

Incompatibilities that are detected by the Migration Toolset are listed in the following table. This table lists incompatibilities by number as reported by the Migration Toolset. This table contains:

- A message number as returned by the Migration Toolset.
- The complete incompatibility message.
- The mode in which the incompatibility would arise and the migration tool used to detect the incompatibility. Also specified is the severity level of the incompatibility assigned by MPT and the event class you must enable before the RTM will detect the incompatibility.
- The cause of the incompatibility.
- Action to take to work around the incompatibility.

Note

Message numbers reported by the A.00.01 version of the Migration Toolset (MPT version A.01.04, OCA version A.00.03, and RTM version A.00.01) may differ from those reported by previous versions of the Migration Toolset.

You can find further information about differences between MPE V/E and MPE/iX in the *Introduction to MPE XL for MPE V Programmers (30367-90005)*, where potential incompatibilities associated with system intrinsics are described in greater detail than in this appendix.

101	MESSAGE	`XARITRAP` intrinsic may require a new trap handler in MPE/iX NM.
	MODE	NM incompatibility detected by MPT (severity level WARNING), OCA, and RTM (Event Class 1).
	CAUSE	HP-PA is a 32-bit architecture, while the architecture of MPE V/E-based computer systems is a 16-bit architecture. In NM, memory addresses represented as pointers or plabels must be 32-bit entities, while in MPE V/E and in CM, pointers and plabels must be 16-bit entities. The `XARITRAP` intrinsic uses plabels.
	ACTION	Redeclare the variables you pass through the required *plabel*, *oldplabel*, and *mask* parameters of `XARITRAP` to be 32-bit entities. If you wish to have arithmetic traps enabled in both CM and NM for a mixed-mode program, you must have two trap routines—one in CM and one in NM.
102	MESSAGE	`XCONTRAP` intrinsic may require a new trap handler in MPE/iX NM.
	MODE	NM incompatibility detected by MPT (severity level WARNING), OCA, and RTM (Event Class 1).
	CAUSE	HP-PA is a 32-bit architecture, while the architecture of MPE V/E-based computer systems is a 16-bit architecture. In NM, memory addresses represented as pointers or plabels must be 32-bit entities, while in MPE V/E and in CM, pointers and plabels must be 16-bit entities. The `XCONTRAP` intrinsic uses plabels.
	ACTION	Redeclare the variables you pass through the required *plabel* and *oldplabel* parameters of `XCONTRAP` to be 32-bit entities.

103	MESSAGE	`iXIBTRAP` intrinsic may require a new trap handler in MPE/iX NM.
	MODE	NM incompatibility detected by MPT (severity level WARNING), OCA, and RTM (Event Class 1).
	CAUSE	HP-PA is a 32-bit architecture, while the architecture of MPE V/E-based computer systems is a 16-bit architecture. In NM, memory addresses represented as pointers or plabels must be 32-bit entities, while in MPE V/E and in CM, pointers and plabels must be 16-bit entities. The `iXIBTRAP` intrinsic uses plabels.
	ACTION	Redeclare the variables you pass through the required *plabel* and *oldplabel* parameters of `iXIBTRAP` to be 32-bit entities. If you wish to have library traps enabled in both CM and NM for a mixed-mode program, you must have two trap routines—one in CM and one in NM.
104	MESSAGE	`XSYSTRAP` intrinsic may require a new trap handler in MPE/iX NM.
	MODE	NM incompatibility detected by MPT (severity level WARNING), OCA, and RTM (Event Class 1).
	CAUSE	HP-PA is a 32-bit architecture, while the architecture of MPE V/E-based computer systems is a 16-bit architecture. In NM, memory addresses represented as pointers or plabels must be 32-bit entities, while in MPE V/E and in CM, pointers and plabels must be 16-bit entities. The `XSYSTRAP` intrinsic uses plabels.
	ACTION	Redeclare the variables you pass through the required *plabel* and *oldplabel* parameters of `XSYSTRAP` to be 32-bit entities. If you wish to have system traps enabled in both CM and NM for a mixed-mode program, you must have two trap routines—one in CM and one in NM.
105	MESSAGE	`SEARCH` intrinsic may encounter pointer problems in MPE/iX NM.
	MODE	NM incompatibility detected by MPT (severity level POSSIBLE), OCA, and RTM (Event Class 1).
	CAUSE	HP-PA is a 32-bit architecture, while the architecture of MPE V/E-based computer systems is a 16-bit architecture. In NM, memory addresses represented as pointers or plabels must be 32-bit entities, while in MPE V/E and in CM, pointers and plabels must be 16-bit entities. The `SEARCH` intrinsic may use pointers.
	ACTION	If you use the optional *defn* parameter of `SEARCH`, you must redeclare the variable you pass through it to be a 32-bit entity.

106	MESSAGE	`MYCOMMAND` intrinsic may encounter pointer problems in MPE/iX NM.
	MODE	NM incompatibility detected by MPT (severity level WARNING), OCA, and RTM (Event Class 1).
	CAUSE	HP-PA is a 32-bit architecture, while the architecture of MPE V/E-based computer systems is a 16-bit architecture. In NM, memory addresses represented as pointers or plabels must be 32-bit entities, while in MPE V/E and in CM, pointers and plabels must be 16-bit entities. The `MYCOMMAND` intrinsic uses pointers.
	ACTION	Redeclare the variable you pass through the required *params* parameter of `MYCOMMAND` to be a an array of 32-bit entities. In addition, if you use the optional *defn* parameter, you must redeclare the variable you pass through it to be a 32-bit entity.
107	MESSAGE	`CREATEPROCESS` intrinsic may encounter pointer problems in MPE iX NM.
	MODE	NM incompatibility detected by MPT severity level POSSIBLE), OCA, and RTM (Event Class 1).
	CAUSE	HP-PA is a 32-bit architecture, while the architecture of MPE V/E-based computer systems is a 16-bit architecture. In NM, memory addresses represented as pointers or plabels must be 32-bit entities, while in MPE V/E and in CM, pointers and plabels must be 16-bit entities. The `CREATEPROCESS` intrinsic may use pointers.
	ACTION	If you use the optional itemnums and items parameters of `CREATEPROCESS`, you must redeclare the variables you pass through them to be arrays of 32-bit entities.
108	MESSAGE	`GENMESSAGE` intrinsic may encounter pointer problems in MPE/iX NM.
	MODE	NM incompatibility detected by MPT (severity level POSSIBLE), OCA, and RTM (Event Class 1).
	CAUSE	HP-PA is a 32-bit architecture, while the architecture of MPE V/E-based computer systems is a 16-bit architecture. In NM, memory addresses represented as pointers or plabels must be 32-bit entities, while in MPE V/E and in CM, pointers and plabels must be 16-bit entities. The `GENMESSAGE` intrinsic may use pointers.
	ACTION	If you use the optional *param* parameter of `GENMESSAGE`, you must redeclare the variable you pass through it to be a 32-bit entity.

109	MESSAGE	`SWITCHDB` intrinsic not supported in NM.
	MODE	NM incompatibility detected by MPT (severity level ERROR) and OCA.
	CAUSE	The `SWITCHDB` intrinsic changes the DB register so that it points to the base of an extra data segment instead of the base of the stack on MPE V/E and in CM. The `SWITCHDB` intrinsic is not available to programs executing in NM.
	ACTION	Remove all `SWITCHDB` invocations and dependent code from programs which will execute in NM.
110	MESSAGE	`ARITRAP` intrinsic *state* parameter is 32-bit signed integer.
	MODE	NM incompatibility detected by MPT (severity level WARNING) and OCA.
	CAUSE	The `ARITRAP` intrinsic collectively enables or disables arithmetic traps. On MPE V/E and in CM, the *state* parameter is a required logical by value entity. In NM, `ARITRAP` uses the *trapstate* parameter instead, which is a required 32-bit signed integer by value entity.
	ACTION	Redeclare the variables you pass through the required *trapstate* parameter to be a 32-bit entity for programs which will execute in NM. If you wish to have arithmetic traps collectively enabled in both CM and NM for a mixed-mode program you must have two trap routines—one in CM and one in NM.
196	MESSAGE	`LOADPROC` intrinsic -> `HPGETPROCPLABEL` intrinsic for NM.
	MODE	NM incompatibility detected by MPT (severity level WARNING) and OCA.
	CAUSE	Library routines may exist in either CM libraries (SLs) or in NM libraries (iXs). The internal structures of these libraries differ significantly. The `LOADPROC` intrinsic dynamically loads procedures from MPE V/E or CM libraries only.
	ACTION	If your library routine is in a CM SL, you must use `LOADPROC` from a CM program, or either `HPLOADCMPROCEDURE` or `LOADPROC` from a NM program to dynamically load this procedure. If you move your library routine to a NM iX, you must use `HPGETPROCPLABEL` from a NM program, or `HPLOADNMPROC` from a CM program.

200	MESSAGE	`COMMAND` intrinsic possible programmatic execution of incompatible commands.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and OCA.
	CAUSE	Some MPE V/E commands have either been modified for use on MPE/iX or are no longer supported on MPE/iX. The `COMMAND` intrinsic may execute one of the following modified or unsupported commands: `:LISTF, :LISTACCT, :LISTUSER, :LISTGROUP, :PTAPE, :TUNE, SHOWCACHE, :CACHECONTROL, :STARTCACHE, :STOPCACHE, :SHOWCOM, :DSCONTROL, :STARTSESS,` and `:SPEED.`
	ACTION	If OCA or MPT reports this event, enable Event Class 2 of RTM to detect when `COMMAND` executes one of the commands listed above. Check for event numbers 201 through 218, which indicate that RTM detected execution of a modified or unsupported command. In all cases, analyze the program to determine whether the command executed was acquired interactively or embedded in code and evaluate how the command is used.
201	MESSAGE	Programmatic `:LISTF`.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and RTM (Event Class 2).
	CAUSE	Because of differences between HP-PA and the architecture of MPE V/E-based computer systems, MPE/iX `:LISTF -1` output is different from MPE V/E `:LISTF -1 format`. In addition, because of changes in extent management on MPE/iX, the MX (maximum extents) field of `:LISTF -2` will contain an asterisk if a maximum number of extents was not specified at file creation.
	ACTION	Remove code dependent upon `:LISTF -1` output. Determine what data you needed from `:LISTF -1` and evaluate alternative ways to retrieve that data, for example, through the `FLABELINFO` intrinsic. Furthermore, MPE/iX enhancements to the `:LISTF` command may also provide this data. If your code is dependent upon data retrieved from the MX field output by `:LISTF 2`, you may have to modify your code to handle the asterisk.
202	MESSAGE	Programmatic `:LISTACCT` to a file.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and RTM (Event Class 2).
	CAUSE	Because of enhancements to the MPE/iX operating system, output from the `:LISTACCT` command appears in ASCII format.
	ACTION	If you are doing programmatic analysis of the `:LISTACCT` output, you must modify your code to handle the new output format.

203	MESSAGE	Programmatic `:LISTUSER` to a file.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and RTM (Event Class 2).
	CAUSE	Because of enhancements to the MPE/iX operating system, output from the `:LISTUSER` command appears in ASCII format.
	ACTION	If you are doing programmatic analysis of the `:LISTUSER` output, you must modify your code to handle the new output format.
204	MESSAGE	Programmatic `:LISTGROUP` to a file.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and RTM (Event Class 2).
	CAUSE	Because of enhancements to the MPE/iX operating system, output from the `:LISTGROUP` command appears in ASCII format.
	ACTION	If you are doing programmatic analysis of the :LISTGROUP output, you must modify your code to handle the new output format.
205	MESSAGE	Programmatic `:PTAPE` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and RTM (Event Class 2).
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX, resulting in paper tape readers being unsupported on MPE/iX-based computer systems. The `:PTAPE` command is not supported on MPE iX-based computer systems.
	ACTION	Remove all `:PTAPE` invocations and dependent code. If you are using `:PTAPE` for uses other than reading from paper tape devices, you must determine the original need for the call to `:PTAPE` and develop alternative solutions.
207	MESSAGE	Programmatic `:TUNE`.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and RTM (Event Class 2).
	CAUSE	Because of enhancements to the MPE/iX operating system, the required *minclockcycle* parameter of the `:TUNE` command is ignored.
	ACTION	No action is required, as the *minclockcycle* parameter of `:TUNE` is ignored by the operating system.

208	MESSAGE	Programmatic `:SHOWCACHE`.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and RTM (Event Class 2).
	CAUSE	HP-PA features include complete operating system management of all cache activities. Programmatic control of cache activities is not allowed. MPE V/E cache management commands, including the `:SHOWCACHE` command are not supported on MPE/iX.
	ACTION	Remove all `:SHOWCACHE` invocations and dependent code.
209	MESSAGE	Programmatic `:CACHECONTROL`.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and RTM (Event Class 2).
	CAUSE	HP-PA features include complete operating system management of all cache activities. Programmatic control of cache activities is not allowed. In addition, `BLOCKONWRITE` on a system-wide basis is not supported on MPE/iX. MPE V/E cache management commands, including the `:CACHECONTROL` command, are not supported on MPE/iX.
	ACTION	Remove all `:CACHECONTROL` invocations and dependent code. `BLOCKONWRITE` on a file-by-file basis is available through the `FSETMODE` intrinsic.
210	MESSAGE	Programmatic `:STARTCACHE`.
	MODE	CM and NM Mode incompatibility detected by MPT (severity level ERROR) and RTM (Event Class 2).
	CAUSE	HP-PA features include complete operating system management of all cache activities. Programmatic control of cache activities is not allowed. MPE V/E cache management commands, including the `:STARTCACHE` command, are not supported on MPE/iX.
	ACTION	Remove all `:STARTCACHE` invocations and dependent code.
211	MESSAGE	Programmatic `:STOPCACHE`.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and RTM (Event Class 2).
	CAUSE	HP-PA features include complete operating system management of all cache activities. Programmatic control of cache activities is not allowed. MPE V/E cache management commands, including the `:STOPCACHE` command, are not supported on MPE/iX.
	ACTION	Remove all `:STOPCACHE` invocations and dependent code.

215	MESSAGE	Programmatic `:SHOWCOM`.
	MODE	CM and NM event detected by MPT (severity level ERROR) and RTM (Event Class 2).
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX; the Intelligent Network Processor (INP) and the associated `:SHOWCOM` command are not supported on MPE/iX.
	ACTION	Remove all `:SHOWCOM` invocations and dependent code. Evaluate your original need for `:SHOWCOM` and determine alternative solutions. Equivalent functionality may be available through the NS3000/XL subsystem.
216	MESSAGE	Programmatic `:DSCONTROL`.
	MODE	CM and NM event detected by MPT (severity level ERROR) and RTM (Event Class 2).
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX, resulting in the DS/3000 subsystem being replaced on MPE/iX with the NS3000/XL subsystem (an enhanced version of the NS3000/V subsystem available on MPE V/E). The `:DSCONTROL` command is not supported on MPE/iX.
	ACTION	Remove all `:DSCONTROL` invocations and dependent code. Evaluate your original need for `:DSCONTROL` and determine alternative solutions, for example, the `:NSCONTROL` command currently available through the NS3000/V subsystem. In addition, equivalent functionality may be available through the NS3000/XL subsystem.
217	MESSAGE	Programmatic `:STARTSESS`.
	MODE	CM and NM event detected by MPT (severity level POSSIBLE) and RTM (Event Class 2).
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX; the following *termtypes* are not supported: 4, 6, 12, 13, 14, 15, 16, 19, and 20. The required *logonstring* parameter of the `:STARTSESS` command may specify an unsupported *termtype*.
	ACTION	If the required *logonstring* parameter of `:STARTSESS` includes a *termtype*, you may have to modify the variable passed through *logonstring* to specify a supported *termtype* (10, 18, 21, and 22).

218	MESSAGE	Programmatic `:SPEED`.
	MODE	CM and NM event detected by MPT (severity level POSSIBLE) and RTM (Event Class 2).
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX; the following baud rates are not supported: 110, 150, and 600. The `:SPEED` command uses a baud rate.
	ACTION	Modify only the `:SPEED` invocations that specify the unsupported baud rates listed above, replacing unsupported baud rates with supported baud rates.
250	MESSAGE	Error numbers returned by the `COMMAND` intrinsic may be different.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING), OCA, and RTM (Event Class 2). When RTM detects events 250, 251, and 252, only event 250 is actually logged to a system log file located on disk. This minimizes disk space consumption associated with each detected invocation of the `COMMAND` intrinsic.
	CAUSE	Because of differences between HP-PA and the architecture of MPE-V/E based computer systems, a few MPE V/E Command Interpreter (CI) error numbers and their associated messages have been replaced on MPE/iX with more precise error numbers and messages. The required *error* parameter of the COMMAND intrinsic may , in a very few cases, return a different CI error number on an MPE/iX-based computer system than it would on an MPE V/E-based computer system for the same type of error.
	ACTION	If you have code that relies upon specific non-zero values returned in the required *error* parameter of `COMMAND`, you may have to modify your code. Analyze your source code and list the error numbers upon which your MPE V/E code relies. When MPE/iX error numbers are available, check for incompatibilities. In addition, if this event is logged by RTM, analyze your code to determine whether the incompatibility is associated with event 250, 251, or 252.

251	MESSAGE	The meaning of the value returned in *param* has changed for the `COMMAND` intrinsic.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and OCA. If RTM logs event 250, event 251 may apply to the event detected by RTM. Analyze your code to determine if this is the case.
	CAUSE	Because of enhancements to the MPE/iX operating system, a value returned by the required *param* parameter of the `COMMAND` intrinsic has a different interpretation on MPE/iX than it does on MPE V/E. On MPE V/E, a positive value *n* less than 12 indicates that an error occurred in the *nth* parameter of the character array passed through the *comimage* parameter; a positive value greater than 12 is a file system error number. On MPE/iX, a negative value indicates the character position within the *comimage* array which contains the first character of the parameter where an error occurred; a positive value indicates a file system error number.
	ACTION	If you have code that relies upon the value returned by the *param* parameter of `COMMAND`, determine what, if any, changes you need to make.
252	MESSAGE	`COMMAND` intrinsic supports process creation on MPE/iX.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and OCA. If RTM logs event 250, event 252 may apply to the event detected by RTM. Analyze your code to determine if this is the case.
	CAUSE	Because of enhancements to the MPE/iX operating system, you can use the `COMMAND` intrinsic to invoke commands that create processes, for example, `:RUN`, `:SEGMENTER`, and so forth.
	ACTION	If your program is using the `COMMAND` intrinsic to invoke a CI command that has been input by a user, you should investigate whether or not you want the user to invoke process creation commands not allowed on MPE V/E.

300	MESSAGE	Some `FFILEINFO` intrinsic *item* values may not be valid on MPE/iX.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and OCA.
	CAUSE	Some values returned in the optional *itemvalue* parameter of the `FFILEINFO` intrinsic either have modified meanings or have no meaning on an MPE iX-based computer system. Affected *itemvalues* are those associated with the following values passed in an associated *itemnums* parameter: 5, 7, 13, 15, 16, 40, 41, 42, 44, 47, 48, and 49.
	ACTION	If OCA or MPT reports this event, enable Event Class 3 of RTM to detect when `FFILEINFO` returns one of the *itemvalues* listed above. Check for event numbers 305 through 349 which indicate that RTM detected that `FFILEINFO` returned a modified or meaningless *itemvalue*.
305	MESSAGE	`FFILEINFO` intrinsic *itemvalue* 5 device type/subtype.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	When the specified file is a standard disk file (as opposed to a KSAM, RIO, or CIR disk file), MPE/iX does not distinguish between the supported disk devices' device types. In this specific case, the value returned by *itemvalue* 5 always represents device type 3, subtype 8 (indicating a 793X disk drive).
	ACTION	If the standard disk file specified in the `FFILEINFO` call will not reside on a 793X disk drive, you must remove dependency upon *itemvalue* 5.
306	MESSAGE	`FFILEINFO` intrinsic - Item 6 (ldev) may be invalid for disk.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	When the specified file is a standard disk file (as opposed to a KSAM, RIO, or CIR disk file), MPE/iX returns the logical device number where the disk file's file label resides. This may not be the same logical device that contains the file's data.
	ACTION	If your code relies upon the disk file label's logical device number being the same as the logical device number for the file's data, you will have to modify your code to remove this dependency.

307	MESSAGE	`FFILEINFO` intrinsic *itemvalue* 7 hardware device address.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	MPE/iX does not support the DRT number/unit number hardware address. When the specified file is located on a disk, the value returned by *itemvalue* 7 always represents DRT number 8, unit number 0, meaningless values. For MPE/iX disk files, the DRT number/unit number representation for a hardware device address has been replaced by device path, available through *itemvalue* 75 of the `FFILEINFO` intrinsic.
	ACTION	Remove dependency upon *itemvalue* 7 of `FFILEINFO`. Evaluate why your program used this value. You may be able to perform an equivalent action using the device path returned by *itemvalue* 75 of `FFILEINFO`.
313	MESSAGE	`FFILEINFO` intrinsic *itemvalue* 13 physical I/O count.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and RTM (Event Class 3).
	CAUSE	Because of enhancements to the MPE/iX operating system, the value returned by *itemvalue* 13 of the `FFILEINFO` intrinsic is the buffered physical count of data blocks transferred. If the file was opened using other than buffered access (for example, `NOBUF`), the value returned will not be meaningful.
	ACTION	If your code relies upon *itemvalue* 13 of `FFILEINFO` returning the physical I/O count for unbuffered files, you must remove dependency on physical I/O count.
315	MESSAGE	`FFILEINFO` intrinsic - Item 15 extent size.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	On MPE/iX, the file system manages extent allocation, where variable numbers of extents, each of variable size, are allocated based on need. There is no fixed extent size for a file. Extent information specified at file creation is meaningless on an MPE/iX-based computer system and is stored in the file label only to maintain backward compatibility with MPE V/E-based computer systems. Therefore, *itemvalue* 15 of the `FFILEINFO` intrinsic returns a value that is meaningless on MPE/iX.
	ACTION	If your code relies upon the extent size returned by *itemvalue* 15 of `FFILEINFO`, you will have to modify your code to remove dependency upon extent size.

316	MESSAGE	`FFILEINFO` intrinsic - Item 16 maximum number of extents.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	On MPE/iX, the file system manages extent allocation, where variable numbers of extents, each of variable size, are allocated based on need. There is no practical limit to the number of extents for a file. Extent information specified at file creation is meaningless on an MPE/iX-based computer system and is stored in the file label only to maintain backward compatibility with MPE V/E-based computer systems. Therefore, *itemvalue* 16 of the `FFILEINFO` intrinsic returns a value that is meaningless on MPE/iX.
	ACTION	If your code relies upon the number of extents returned by *itemvalue* 16 of `FFILEINFO`, you will have to modify your code to remove dependency upon number of extents.
319	MESSAGE	`FFILEINFO` intrinsic - Item 19 (flabel disk sec addr) returns 0.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	When the specified file is a standard disk file (as opposed to a KSAM, RIO, or CIR disk file), the file label disk sector address returned in *itemvalue* 19 of the `FFILEINFO` intrinsic has no meaning, but always returns a 0. This *itemvalue* is kept only to maintain backward compatibility with MPE V/E-based computer.
	ACTION	If your code relies upon the file label disk sector address returned by *itemvalue* 19 of `FFILEINFO`, you will have to modify your code to remove this dependency.
340	MESSAGE	`FFILEINFO` intrinsic - Item 40 disk device status.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	When the specified file is a standard disk file (as opposed to a KSAM, RIO, or CIR disk file), the disk device status returned in *itemvalue* 40 of the `FFILEINFO` intrinsic has no meaning, but always returns a 0. This *itemvalue* is kept only to maintain backward compatibility with MPE V/E-based computer systems.
	ACTION	If your code relies upon the disk device status returned by *itemvalue* 40 of `FFILEINFO`, you will have to modify your code to remove the dependency.

341	MESSAGE	`FFILEINFO` intrinsic - Item 41 device type.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	When the specified file is a standard disk file (as opposed to a KSAM, RIO, or CIR disk file), MPE/iX does not distinguish between the supported disk devices' device types. In this specific case, the value returned by *itemvalue* 41 of the `FFILEINFO` intrinsic always represents device type 3 (indicating a 79xx disk drive).
	ACTION	If the standard disk file specified in the `FFILEINFO` call will not reside not a 79xx disk drive, you must remove dependency upon *itemvalue* 41.
342	MESSAGE	`FFILEINFO` intrinsic - Item 42 device subtype.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	When the specified file is a standard disk file (as opposed to a KSAM, RIO, or CIR disk file), MPE/iX does not distinguish between the supported disk devices' device types. In this specific case, the value returned by *itemvalue* 42 always represents device subtype 8 (indicating a 793x disk drive).
	ACTION	If the standard disk file specified in the `FFILEINFO` call will not reside not a 793x disk drive, you must remove dependency upon *itemvalue* 42.
344	MESSAGE	`FFILEINFO` intrinsic - Item 44 last extent allocated.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	Because of enhancements to the MPE/iX operating system, *itemvalue* 44 of the `FFILEINFO` intrinsic returns a value that represents the number of extents that have been accessed during the current open. On MPE V/E-based computer systems, this same *itemvalue* returns the cardinal number of the last extent accessed.
	ACTION	If your code relies upon the information returned in *itemvalue* 44 of `FFILEINFO`, you will have to either remove the dependency or modify your code to rely upon the MPE/iX interpretation of *itemvalue* 44.

347	MESSAGE	`FFILEINFO` intrinsic - Item 47 DRT number.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	MPE/iX does not support the DRT number/unit number hardware address. When the specified file is located on a disk, the value returned in *itemvalue* 47 of the `FFILEINFO` intrinsic always represents DRT number 8, a meaningless value. For MPE/iX disk files, the DRT number/unit number representation for a hardware device address has been replaced by device path, available through *itemvalue* 75 of `FFILEINFO`.
	ACTION	Remove dependency upon *itemvalue* 47 of `FFILEINFO`. Evaluate why your program used this value. You may be able to perform an equivalent action using the device path returned by *itemvalue* 75 of `FFILEINFO`.
348	MESSAGE	`FFILEINFO` intrinsic - Item 48 device unit number.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	MPE/iX does not support the DRT number/unit number hardware address. When the specified file is located on a disk, the value returned the value returned by *itemvalue* 48 of the `FFILEINFO` intrinsic always represents unit number 0, a meaningless value. For MPE/iX disk files, the DRT number/unit number representation for a hardware device address has been replaced by device path, available through *itemvalue* 75 of `FFILEINFO`.
	ACTION	Remove dependency upon *itemvalue* 48 of `FFILEINFO`. Evaluate why your program used this value. You may be able to perform an equivalent action using the device path returned by *itemvalue* 75 of `FFILEINFO`.
349	MESSAGE	`FFILEINFO` intrinsic - Item 49 software interrupt plabel (MSG files).
	MODE	NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 3).
	CAUSE	HP-PA is a 32-bit architecture, while the architecture of MPE V/E-based computer systems is a 16-bit architecture. In NM, memory addresses represented as pointers or plabels must be 32-bit entities, while in MPE V/E and in CM, pointers and plabels must be 16-bit entities. An integer equivalent to a plabel of a software interrupt handler is returned in *itemvalue* 49 of the `FFILEINFO` intrinsic.
	ACTION	Redeclare the variable you use to hold the value returned by *itemvalue* 49 of `FFILEINFO` to be a 32-bit entity.

400	MESSAGE	Some `FGETINFO` intrinsic item values may not be valid on MPE/iX.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and OCA.
	CAUSE	The values returned in the optional *devtype*, *hdaddr*, *physcount*, *extsize*, and *numextents* parameters of the `FGETINFO` intrinsic have either modified meanings or no meaning on an MPE/iX-based computer system.
	ACTION	If MPT or OCA reports this event, enable Event Class 4 of RTM to detect when `FGETINFO` returns one of the optional parameters listed above. Check for event numbers 405 through 416, which indicate that RTM detected that `FGETINFO` returned a modified or meaningless parameter value.
405	MESSAGE	`FGETINFO` intrinsic *devtype* parameter.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 4).
	CAUSE	When the specified file is a standard disk file (as opposed to a KSAM, RIO, or CIR disk file), MPE/iX does not distinguish between the supported disk devices' device types. In this specific case, the value returned by the optional *devtype* parameter of the `FGETINFO` intrinsic always represents device type 3, subtype 8 (indicating a 793x disk drive).
	ACTION	If the standard disk file specified in the `FGETINFO` call will not reside on a 793x disk drive, you must remove dependency upon the *devtype* parameter.
407	MESSAGE	`FGETINFO` intrinsic *hdaddr* parameter.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 4).
	CAUSE	MPE/iX does not support the DRT number/unit number hardware address. When the specified file is located on a disk, the value returned by the optional *hdaddr* parameter of the `FGETINFO` intrinsic always represents DRT number 8, unit number 0. For MPE/iX disk files, the DRT number/unit number representation for a hardware device address has been replaced by device path, available through *itemvalue* 75 of the `FFILEINFO` intrinsic.
	ACTION	Remove dependency upon the *hdaddr* parameter of `FGETINFO`. Evaluate why your program used this value. You may be able to perform an equivalent action using the device path returned by *itemvalue* 75 of `FFILEINFO`.

413	MESSAGE	`FGETINFO` intrinsic *physcount* parameter.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and RTM (Event Class 4).
	CAUSE	Because of enhancements to the MPE/iX operating system, the value returned by the optional *physcount* parameter of the `FGETINFO` intrinsic is the buffered physical count of data blocks transferred. If the file was opened using other than buffered access (for example, `NOBUF`), the value returned will not be meaningful.
	ACTION	If your code relies upon the *physcount* parameter of `FGETINFO` returning the physical I/O count for unbuffered files, you must remove dependency on physical I/O count.
415	MESSAGE	`FGETINFO` intrinsic *extsize* parameter.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 4).
	CAUSE	On MPE/iX, the file system manages extent allocation, where variable numbers of extents, each of variable size, are allocated based on need. There is no fixed extent size for a file. Extent information specified at file creation is meaningless on an MPE/iX-based computer system and is stored in the file label only to maintain backward compatibility with MPE V/E-based computer systems. Therefore, the optional *extsize* parameter of the `FGETINFO` intrinsic returns a value that is meaningless on MPE/iX.
	ACTION	If your code relies upon the extent size returned by the *extsize* parameter of `FGETINFO`, you will have to modify your code to remove dependency upon extent size.
416	MESSAGE	`FGETINFO` intrinsic *numextents* parameter.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 4).
	CAUSE	On MPE/iX, the file system manages extent allocation, where variable numbers of extents, each of variable size, are allocated based on need. There is no practical limit to the number of extents for a file. Extent information specified at file creation is meaningless on an MPE/iX-based computer system and is stored in the file label only to maintain backward compatibility with MPE V/E-based computer systems. Therefore, the optional *numextents* parameter of the `FGETINFO` intrinsic returns a value that is meaningless on MPE/iX.
	ACTION	If your code relies upon the number of extents returned by the *numextents* parameter of `FGETINFO`, you will have to modify your code to remove dependency upon number of extents.

500	MESSAGE	Some `FCONTROL` intrinsic *controlcodes* are not valid on MPE/iX.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE) and OCA.
	CAUSE	Some values that may be contained in the required *param* parameter of the `FCONTROL` intrinsic have either been modified or have no meaning on an MPE/ iX-based computer system. Affected *params* are those associated with *controlcodes* 3 and 48.
	ACTION	If MPT or OCA reports this event, enable Event Class 5 of RTM to detect when `FCONTROL` calls *controlcode* 3 and 48.
503	MESSAGE	`FCONTROL` intrinsic controlcode 3 Read hardware status word.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and RTM (Event Class 5).
	CAUSE	Because of differences between HP-PA and the architecture of MPE V/E-based computer systems, the system hardware status word returned by `FCONTROL` has no meaning on MPE/iX. A meaningless value is returned in the required *param* parameter of the `FCONTROL` intrinsic when a value of 3 is passed in required *controlcode* parameter.
	ACTION	Remove dependencies upon *controlcode* 3 of `FCONTROL`.
548	MESSAGE	`FCONTROL` intrinsic *controlcode* 48 Enable/Disable software interrupts.
	MODE	NM incompatibility detected by MPT (severity level WARNING) and RTM (Event Class 5).
	CAUSE	HP-PA is a 32-bit architecture, while the architecture of MPE V/E-based computer systems is a 16-bit architecture. In NM, memory addresses represented as pointers or plabels must be 32-bit entities, while in MPE V/E and in CM, pointers and plabels must be 16-bit entities. The plabel of a software interrupt handler may be passed through the required *param* parameter of `FCONTROL` if the required *controlcode* parameter passes the value 48.
	ACTION	Redeclare the variable you pass through the *param* parameter of `FCONTROL` to be a 32-bit entity when you also pass a *controlcode* value of 48.

601	MESSAGE	The `PTAPE` intrinsic is not available on MPE/iX.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR), OCA, and RTM (Event Class 6).
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX, resulting in paper tape readers being unsupported on MPE/iX-based computer systems. The `PTAPE` intrinsic is not supported on MPE/ iX-based computer systems.
	ACTION	Remove all `PTAPE` invocations and dependent code. If you are using `PTAPE` for uses other than reading from paper tape devices, you must determine the original need for the call to `PTAPE` and develop alternative solutions.
602	MESSAGE	The `FCARD` intrinsic is not available on MPE/iX.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR), OCA , and RTM (Event Class 6).
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX; the HP 7260 series of card readers is not supported on MPE/iX-based computer systems. Because the `FCARD` intrinsic accesses only these card readers, `FCARD` likewise is not supported on MPE/iX.
	ACTION	Remove all `FCARD` invocations and dependent code.
604	MESSAGE	`FDEVICECONTROL` intrinsic *controlcode* parameter value of 192.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING) and OCA.
	CAUSE	`FDEVICECONTROL` with *controlcode* of 192 is used for devices connected through the terminal subsystem. Modifications have been made to the terminal subsystem of MPE/iX, resulting in `FDEVICECONTROL` being unsupported for these devices on the 1.0 version of MPE/iX-based computer systems.
	ACTION	If you are migrating to the 1.0 release of MPE/iX, remove all `FDEVICECONTROL` calls using a *controlcode* of 192 and dependent code. Evaluate your original need for `FDEVICECONTROL` and determine alternative solutions. Some of the functions of such calls are no longer necessary; for example, changes to ENQ/ACK protocol are not needed since this protocol is not used on MPE/iX. Other functions that `FDEVICECONTROL` provides are also available through the `FCONTROL` intrinsic, such as using parity checking and setting new values for the Subsystem Break or Alternate End-of-Record. On release 1.1 or later, `FDEVICECONTROL` with control code 192 and its dependent codes is implemented as on MPE V, with the exception of Alternate End-of-Record. On MPE V systems, you are allowed to designate multiple AEOR characters for a single device. MPE/iX, however, allows you to designate only one AEOR character for a device at any given time. Modifications may be necessary to provide only one AEOR character in existing code. For more information refer to *MPE XL Asynchronous Serial Communications Programmer's Reference Manual* (32022-90012) for further information.

605	MESSAGE	KSAM record-level locking not supported on MPE/iX systems.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and OCA.
	CAUSE	Because of differences between HP-PA and the architecture of MPE V/E-based computer systems, KSAM record-level locking is not supported on MPE/iX-based computer systems.
	ACTION	Remove all KPTR invocations and dependent code. Evaluate your original need for KSAM record-level locking and determine alternative solutions.
800	MESSAGE	CS data communication routines are not available on MPE/iX.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and OCA.
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX; CS data communication routines are not supported on MPE/iX.
	ACTION	Remove all CS routines and dependent code. Analyze the original reasons for having the CS routines and develop alternative solutions. Equivalent functionality on MPE/iX may be available through the NS3000/XL subsystem.
801	MESSAGE	`STARTSESS` intrinsic *logonstr* parameter may specify an unsupported *termtype*.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE), OCA, and RTM (Event Class 8).
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX; the following *termtypes* are not supported: 4, 6, 12, 13, 14, 15, 16, 19, and 20. The required *logonstr* parameter of the `STARTSESS` intrinsic may specify an unsupported *termtype*.
	ACTION	If the required *logonstr* parameter of `STARTSESS` includes a *termtype*, you may have to modify the variable passed through *logonstring* to specify a supported *termtype* (10, 18, 21, and 22).

802	MESSAGE	RFA (1000 to 3000) data communication routines are not available on MPE/iX.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and OCA.
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX; RFA data communication routines for communicating between HP 3000 Series computer systems and HP 1000 Series computer systems are not supported on MPE/iX.
	ACTION	Remove affected RFA routines and dependent code. Analyze the original reasons for having these routines and develop alternative solutions. Equivalent functionality on MPE/iX may be available through the NS3000/XL subsystem.
803	MESSAGE	PTOP data communication routines are not available on MPE/iX.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and OCA.
	CAUSE	Modifications have been made to the attached peripheral environment of MPE/iX; PTOP data communication routines are not supported.
	ACTION	Remove all PTOP routines and dependent code. Analyze the original reasons for having the PTOP routines and develop alternative solutions. For example, you may be able to immediately replace PTOP routines with equivalent NetIPC routines available in the NS3000/V subsystem. Equivalent functionality on MPE/iX may be available through the NS3000/XL subsystem.
810	MESSAGE	DEL routines not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and OCA.
	CAUSE	The Data Entry Library (DEL) screen-handling package is not supported on MPE/iX.
	ACTION	Analyze the original reasons your program required the Data Entry Library routines and determine if equivalent functionality is available using VPLUS intrinsics.
850	MESSAGE	Data capture routines are not available on MPE/iX.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR) and OCA.
	CAUSE	Data capture routines are not supported on MPE/iX.
	ACTION	Analyze the original reasons your program required the data capture routines and determine if equivalent functionality is available using VPLUS intrinsics.

851	MESSAGE	Printer Support Package (PSP) routines are not callable from NM on MPE/iX.
	MODE	NM incompatibility detected by MPT (severity level WARNING) and OCA.
	CAUSE	On MPE/iX PSP routines are not directly callable from a NM program.
	ACTION	On MPE/iX, programs accessing the PSP routines must either run in CM or use the Switch subsystem from NM to access the PSP routines.
897	MESSAGE	THISCPU procedure returns new values.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE), OCA, and RTM (Event Class 8).
	CAUSE	The THISCPU procedure is an undocumented but user-callable procedure that returns an integer value indicating the type of HP 3000 system on which the program is running. While this procedure will function properly in both CM and NM, it returns a new value, 16, for the 900 Series systems.
	ACTION	Ensure that the program can accommodate the new value when it is run on 900 Series systems. This may require minor program logic changes.
898	MESSAGE	MPE procedure returns new values.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE), OCA, and RTM (event Class 8).
	CAUSE	The MPE procedure is an undocumented but user-callable procedure that returns a value related to the version of MPE on the system. While this procedure will function properly in both CM and NM, it returns a new value, 41, for MPE/iX.
	ACTION	Ensure that the program can accommodate the new values when it is run on 900 Series systems. This may require minor program logic changes. New programs developed for MPE/iX computer systems should use the supported system variable `HPVERSION` to get the operating system version instead of calling the unsupported MPE procedure.

899	MESSAGE	`GETPRIVMODE` intrinsic may access incompatible features
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING for CM, ERROR for NM), OCA, and RTM (Event Class 8).
	CAUSE	While the `GETPRIVMODE` intrinsic is supported in both CM and NM, the use of privileged mode may access features of MPE or of the system hardware which are no longer available on the 900 Series systems or on the MPE/iX operating system.
	ACTION	Examine the program's use of Privileged Mode to determine if it accesses features no longer available on the 900 Series. If it does, the program must be modified before it can be used on those systems safely and correctly. In addition, equivalent functionality may be available on MPE/iX without the use of Privileged Mode.
2000	MESSAGE	`:CACHECONTROL` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	Disk caching is part of MPE/iX Transaction Management. Thus, the `:CACHECONTROL` command is not available to manipulate its operational parameters.
	ACTION	Remove all references to the `:CACHECONTROL` command from the job or UDC file.
2001	MESSAGE	`:DISCRPS` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	The `:DISCRPS` command was used to enable or disable the Rotational Position Sensing (RPS) feature on some disk drives. On MPE/iX, the default is to disable RPS because MPE/iX cannot detect the firmware version of the disk drive to ensure that data corruption will not occur.
	ACTION	Examine the firmware versions of any 7933/35 disk drives using `CS80DIAG` to ensure that the firmware versions are 5.1 or later. Then, if `:DISCRPS` is enabled for those disks on MPE/iX, disk corruption will not occur.
2002	MESSAGE	`:FULLBACKUP` command not supported (use `:SYSGEN` or `:STORE` instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:FULLBACKUP` command is not supported on MPE/iX due to changes in the method for doing system backups.
	ACTION	Remove all references to the `:FULLBACKUP` command from the job or UDC file, replacing them with equivalent `:STORE` commands.

2003	MESSAGE	`PARTBACKUP` command not supported (use `:SYSGEN` or `:STORE` instead).
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `PARTBACKUP` command is not supported on MPE/iX due to changes in the method for doing system backups.
	ACTION	Remove all references to the `PARTBACKUP` command from the job or UDC file, replacing them with equivalent `:STORE` commands. This change may also require changes to Operator procedures in order to provide the relative full dump date. (This date was automatically supplied by the `:FULLBACKUP` or `PARTBACKUP` commands.)
2004	MESSAGE	`:PTAPE` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:PTAPE` command was originally designed to read paper tapes from teletype terminals on the earliest HP 3000 systems. Since these terminals are not supported on MPE/iX, the `:PTAPE` command has been removed.
	ACTION	Remove all references to the `:PTAPE` command from the job or UDC file.
2005	MESSAGE	`:SHOWCACHE` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	Disk caching is part of MPE/iX Transaction Management. You may use the System Monitor Tool (SMT) to examine its function.
	ACTION	Remove all references to the `:SHOWCACHE` command from the job or UDC file. If you have access to SMT, you can use it to report on much of the same information as `:SHOWCACHE`.
2007	MESSAGE	`:STARTCACHE` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	Disk caching is part of MPE/iX Transaction Management. It is not necessary to enable it using MPE/iX commands.
	ACTION	Remove all references to the `:STARTCACHE` command from the job or UDC file.

2008	MESSAGE	`:STOPCACHE` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	Disk caching is part of MPE/iX Transaction Management. It is not necessary to disable it using MPE/iX commands.
	ACTION	Remove all references to the `:STOPCACHE` command from the job or UDC file.
2009	MESSAGE	`:SYSDUMP` command not supported (use `:SYSGEN` or `:STORE` instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:SYSDUMP` command is not used on MPE/iX. Use the SYSGEN program to configure the system and to generate the MPE/iX equivalent of coldload tapes. Use the `:STORE` command to back up user files.
	ACTION	Remove all references to the `:SYSDUMP` command from the job or UDC file, replacing them with `:STORE` or `:SYSGEN` commands. The command syntax for `:SYSGEN` is different from that for `:SYSDUMP`, which may require additional changes.
2010	MESSAGE	`:SHOWCOM` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:SHOWCOM` command is not supported on MPE/iX since the DS/3000 subsystem is not available on MPE/iX.
	ACTION	Remove all references to the `:SHOWCOM` command from the job or UDC file.
2011	MESSAGE	`:TUNE` command - syntax has changed.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	The *minclockcycle* parameter of the `:TUNE` command is ignored on MPE/iX.
	ACTION	None required in most cases - simply be aware that the parameter value is ignored.

2012	MESSAGE	`:VINIT` command not supported (use VOLUTIL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:VINIT` command is not supported on MPE/iX. The VOLUTIL program performs similar functions.
	ACTION	Modify the job or UDC file to use the VOLUTIL program instead of `:VINIT`. The command syntax of VOLUTIL is different from that of `:VINIT`, which may require additional changes.
2013	MESSAGE	`:LISTACCT` command - output format has changed.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	The `:LISTACCT` command is valid on MPE/iX; however, the listing format is different from that on MPE V/E. Instead of providing the account information in unformatted octal words, as MPE V/E does, a formatted listing similar to that produced by the LISTDIR5 utility on MPE V/E is displayed.
	ACTION	None required unless the command directs its output to a file which is read by a program; in that case, the program must be modified to read the new format.
2014	MESSAGE	`:LISTGROUP` command - output format has changed.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	The `:LISTGROUP` command is valid on MPE/iX; however, the listing format is different from that on MPE V/E. Instead of providing the group information in unformatted octal words, as MPE V/E does, a formatted listing similar to that produced by the LISTDIR5 utility on MPE V/E is displayed.
	ACTION	None required unless the command directs its output to a file which is read by a program; in that case, the program must be modified to read the new format.

2015	MESSAGE	`:LISTUSER` command - output format has changed.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	The `:LISTUSER` command is valid on MPE/iX; however, the listing format is different from that on MPE V/E. Instead of providing the user information in unformatted octal words, as MPE V/E does, a formatted listing similar to that produced by the LISTDIR5 utility on MPE V/E is displayed.
	ACTION	None required unless the command directs its output to a file which is read by a program; in that case, the program must be modified to read the new format.
2016	MESSAGE	`:RUN DISKED#` - not supported (use `:DEBUG` instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The DISKED5 program is not supported on MPE/iX. However, DISKED5 functionality is available in `:DEBUG`.
	ACTION	Modify the job or UDC file to use DEBUG commands instead of DISKED5.
2017	MESSAGE	`:RUN FREE#` - not supported (use DISCFREE instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The FREE5 program has been replaced with a new utility on MPE/iX.
	ACTION	Modify the job or UDC file to use the DISCFREE program instead of the FREE5 program.
2018	MESSAGE	`:RUN LISTDIR#` - not supported (use `:LISTF/ACCT/GROUP/USER`).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The LISTDIR5 program has been replaced by improved functionality in other MPE/iX commands.

	ACTION	Modify the job or UDC file to use one of the following commands instead of the LISTDIR5 program:
		`:LISTACCT :LISTF,3 :LISTF,-3`
		`:LIST,4 :LISTGROUP :LISTUSER`

2019	MESSAGE	`#:RUN LISTEQ#` - not supported (use `:LISTEQ` or `:LISTFTEMP` instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The LISTEQ5 program has been replaced by two MPE commands: `:LISTEQ` and `:LISTFTEMP`. Note that these commands are available on both MPE V/E and MPE/iX systems.
	ACTION	Modify the job or UDC file to use the `:LISTEQ` and/or `:LISTFTEMP` commands instead of the `LISTEQ5` program.
2020	MESSAGE	`:RUN LISTLOG#` - not supported (use LOGTOOL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The LISTLOG5 program has been replaced by the LOGTOOL utility within the Diagnostic User Interface (DUI) on MPE/iX.
	ACTION	Modify the job or UDC file to use the LOGTOOL utility's LIST command. LOGTOOL is accessed through the SYSDIAG (DUI) program.
2021	MESSAGE	`:RUN RECOVER#` - not supported (use VOLUTIL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The RECOVER5 program has been replaced by the RECOVER function of the VOLUTIL program on MPE/iX.
	ACTION	Modify the job or UDC file to use the VOLUTIL program instead of the RECOVER5 program.
2022	MESSAGE	`:RUN MEMTIMER` - not supported (use LOGTOOL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The MEMTIMER program was used on MPE V/E systems to set the timing intervals for memory error logging. This function is replaced by the LOGTOOL utility within the Diagnostic User Interface (DUI) of MPE/iX.
	ACTION	Modify the job or UDC file to use the LOGTOOL utility's MEMTIMER command. LOGTOOL is accessed through the SYSDIAG (DUI) program.

2023	MESSAGE	`:RUN DPAN#` - not supported (use DAT instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The batch type Dump Analyzer Tool (DPAN5) from MPE V/E has been replaced by an online dump analysis tool on MPE/iX.
	ACTION	Modify the job or UDC file to use the DAT program instead of DPAN5.
2024	MESSAGE	`:RUN OPT` - not supported (use SMT instead).
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The OnLine Performance Tool (OPT/3000) on MPE V/E has been replaced with the System Monitor Tool (SMT) on MPE/iX. On MPE XL Release 1.0, SMT is available only as a field tool.
	ACTION	Modify the job or UDC file to remove all references to OPT unless you have access to SMT. In that case, you should use the SMT program instead of OPT.
2025	MESSAGE	`:PREP` command - use `:LINKEDIT` for NM programs.
	MODE	NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:PREP` command was encountered. This command can be used for CM programs, but should be replaced with the MPE/iX `:LINKEDIT` command when migrating programs to NM.
	ACTION	Modify the job or UDC file to use `:LINKEDIT` instead of `:PREP` if this job or UDC file is to be used in NM program development. `:LINKEDIT's` command syntax is different from `:PREP's`; thus, additional changes may be required.
2026	MESSAGE	`:QUANTUM` command not supported (use `:TUNE` instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:QUANTUM` command is not valid on MPE/iX.
	ACTION	Modify the job or UDC file to use the `:TUNE` command instead.

2029	MESSAGE	This command is used by MPE/iX. Check for UDC usage.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).

	CAUSE	The following are new commands available under MPE/iX. One of them was discovered in a job or UDC file.
		`:CALC :CHGROUP :COPY`
		`:DELETEVAR :DO :ECHO :ENDWHILE :EXIT`
		`:INPUT :LINK :LINKEDIT :LISTREDO`
		`:OCTCOMP :OPTION :PRINT :SETVAR`
		`:SHOWVAR :SYSGEN :VSCLOSE :VSOPEN`
		`:VSRELEASE :VSRELEASESYS :VSRESERVE :VSRESERVESYS`
		`:WHILE :XEQ`

	ACTION	Examine the job or UDC file to ensure that none of these commands is being redefined on your system. If one is redefined, the actual MPE command will be inaccessible for any user for whom the UDC with the duplicate name is cataloged unless the name of the UDC is changed.
2035	MESSAGE	`:SPEED` command - some speeds not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	The `:SPEED` command was executed by a UDC file. While this is a valid MPE/iX command, some MPE V/E baud rates are not supported on MPE/iX.
	ACTION	Modify the UDC if it specifies any of the following baud rates: 100, 140, 150, or 600.
2036	MESSAGE	`:LISTF` command - output format has changed.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	A job stream or UDC has invoked the `:LISTF` MPE command. Although `:LISTF` is a valid MPE/iX command, the output format for `:LISTF, -1` has changed.
	ACTION	None required unless the command directs its output to a file which is read by a program; in that case, the program must be modified to read the new format.

2037	MESSAGE	`:LISTFTEMP` command - output format has changed.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	A job stream or UDC has invoked the `:LISTFTEMP` MPE command. Although `:LISTFTEMP` is a valid MPE/iX command, the output format for `:LISTFTEMP,-1` has changed.
	ACTION	None required unless the command directs its output to a file which is read by a program; in that case, the program must be modified to read the new format.
2038	MESSAGE	`:DSCONTROL` command not supported (use NS3000 commands instead).
	MODE	compatibility Mode and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:DSCONTROL` command is not valid on MPE/iX. DS/3000 is not available on MPE/iX; only NS3000/XL, which uses different commands to start and stop communications.
	ACTION	Modify the job or UDC file to use NS3000/XL commands instead. Refer to the *NS3000/XL Network Manager's Reference Manual (36920-90002)* for details.
2039	MESSAGE	`:SEGMENTER` command - use `:LINKEDIT` for NM object modules.
	MODE	NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:SEGMENTER` command was encountered. This command can be used for CM object modules, but should be replaced with the MPE/iX `:LINKEDIT` command when migrating code to NM.
	ACTION	Modify the job or UDC file to use `:LINKEDIT` instead of `:SEGMENTER` if this job or UDC file is to be used in NM code development. The `:LINKEDIT` command's syntax is different from the `:SEGMENTER` command's; thus, additional changes may be required.
2041	MESSAGE	`:FOREIGN` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:FOREIGN` command is not valid on MPE/iX. Foreign disks are not supported.
	ACTION	Remove all references to the `:FOREIGN` command from the job or UDC file.

2042	MESSAGE	`:GIVE` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:GIVE` command is not valid on MPE/iX. This command was used to allow certain diagnostics to access a downed device on some versions of MPE. The MPE/iX diagnostics interface does not require this functionality.
	ACTION	Remove all references to the `:GIVE` command from the job or UDC file.
2043	MESSAGE	`:LISTVS` command not supported (use VOLUTIL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:LISTVS` command is not valid on MPE/iX.
	ACTION	Modify the job stream or UDC file to use the VOLUTIL utility instead.
2044	MESSAGE	`:MPLINE` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:MPLINE` command is not valid on MPE/iX. MTS/3000 is not supported on the 900 Series.
	ACTION	Remove all references to the `:MPLINE` command from the job or UDC file.
2045	MESSAGE	`:NEWVSET` command not supported (use VOLUTIL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:NEWVSET` command is not valid on MPE/iX.
	ACTION	Modify the job stream or UDC file to use the VOLUTIL utility instead.
2046	MESSAGE	`:TAKE` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:TAKE` command is not valid on MPE/iX. This command was used to take back a device that had been given to a diagnostic with `:GIVE`. The MPE/iX diagnostics interface does not require this functionality.
	ACTION	Remove all references to the `:TAKE` command from the job or UDC file.

2047	MESSAGE	`:PURGEVSET` command not supported (use VOLUTIL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:PURGEVSET` command is not valid on MPE/iX.
	ACTION	Modify the job stream or UDC file to use the VOLUTIL utility instead.
2048	MESSAGE	`:ALTVSET` command not supported (use VOLUTIL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The `:ALTVSET` command is not valid on MPE/iX.
	ACTION	Modify the job stream or UDC file to use the VOLUTIL utility instead.
2049	MESSAGE	`:EOF` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:EOF` command is used to simulate a hardware end-of-file from any input device. It is not supported on MPE/iX.
	ACTION	Remove all references to the `:EOF` command from the job or UDC file.
2050	MESSAGE	`:APL` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:APL` command is used to invoke the APL subsystem, which is not available on MPE/iX.
	ACTION	Remove all references to the `:APL` command from the job or UDC file.
2051	MESSAGE	`:COBOL` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:COBOL` command is used to invoke the COBOL 68 compiler, which is not available on MPE/iX.
	ACTION	Remove all references to the `:COBOL` command from the job or UDC file.

2052	MESSAGE	`:COBOLGO` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:COBOLGO` command is used to compile, prep, and run a COBOL 68 program at one time. The COBOL 68 compiler is not available on MPE/iX.
	ACTION	Remove all references to the `:COBOLGO` command from the job or UDC file.
2053	MESSAGE	`:COBOLPREP` command not supported.
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	The `:COBOLPREP` command is used to compile and prep a COBOL 68 program at one time. The COBOL 68 compiler is not available on MPE/iX.
	ACTION	Remove all references to the `:COBOLPREP` command from the job or UDC file.
2054	MESSAGE	`:RUN MEMLOGAN` - not supported (use LOGTOOL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The MEMLOGAN program was used on MPE V/E systems to print the memory error-logging file. This function is replaced by the LOGTOOL utility within the Diagnostic User Interface (DUI) of MPE/iX.
	ACTION	Modify the job or UDC file to use the LOGTOOL utility's MEMRPT command. LOGTOOL is accessed through the SYSDIAG (DUI) program.
2055	MESSAGE	`:RUN ASOCTABL` - not supported (use ASOCTBL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The ASOCTABL program was used on MPE V/E systems to associate a user with a device class. This function is replaced by the ASOCTBL program on MPE iX.
	ACTION	Modify the job or UDC file to use the ASOCTBL program instead of the ASOCTABL program.

2056	MESSAGE	`:RUN ASOCTBL#` - not supported (use ASOCTBL instead).
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The ASOCTBL5 program was used on MPE V/E systems to associate a user with a device class. This function is replaced by the ASOCTBL program on MPE iX.
	ACTION	Modify the job or UDC file to use the ASOCTBL program instead of the ASOCTBL5 program.
2057	MESSAGE	`:RUN SPOOK#` - not supported (See Migration Process Guide)
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	The Spook utility has been replaced with Native Mode Spooler on MPE/iX.
	ACTION	Modify the job or UDC file. Instead of Spook utility commands, use Command Interpreter NM Spooler commands, the `:STORE` and `:RESTORE` commands, the PRINTSPF utility, the SPFXFER utility, or editor subsystems. For more information, see *Native Mode Spooler Reference Manual* (32650-90166) and *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364).
2100	MESSAGE	User SL procedure redefines a system entry point.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	A procedure was found in a Segmented Library which has the same name as a system SL entry point. Use of this SL may intercept calls to the system procedure and prevent proper program operation.
	ACTION	Examine the SL to see if this redefinition of entry points is desired and take appropriate action.
2101	MESSAGE	User SL procedure redefines UNCALLABLE system entry point.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	A procedure was found in a Segmented Library which has the same name as an UNCALLABLE system SL entry point. Use of this SL may intercept calls to the system procedure and prevent proper program operation.
	ACTION	Examine the SL to see if this redefinition of entry points is desired and take appropriate action.

2102	MESSAGE	User SL has externals not satisfied in system SL.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	This SL has external references which are not found in the system SL. This may not be a problem provided the externals are found in an account (PUB) SL. Otherwise, programs using this SL may fail to run.
	ACTION	Examine the SL to see whether the external references will be properly satisfied or not and take appropriate action.
2103	MESSAGE	User SL calls system UNCALLABLE procedure.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	This SL makes calls to an UNCALLABLE system procedure. Normally uncallable procedures should be referenced only by MPE, and they require that the calling SL routine run in Privileged Mode.
	ACTION	Examine this SL and its use of unsupported system procedures to determine if these procedures are changed or eliminated in the new operating system. Use of these procedures is not supported on any version of MPE, and attempts to use them on MPE/iX may cause system failures or loss of data.
2104	MESSAGE	Program has unsatisfied external references.
	MODE	CM and NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	This program has external references which cannot be satisfied from an SL (System, Pub, or Group). In order to run this program, these external references must be resolved.
	ACTION	Determine which externals are missing by attempting to run the program (`:RUN` *program* `;LIB=G`). If these routines are normally supplied from an SL, be sure to migrate the SL along with the program. This program should not work at present on the MPE V/E system due to these unresolved externals.

2105	MESSAGE	Program calls system UNCALLABLE procedure.
	MODE	CM and NM incompatibility detected by MPT (severity level WARNING).
	CAUSE	This program makes calls to an UNCALLABLE system procedure. Normally uncallable procedures should be referenced only by MPE, and they require that the program run in Privileged Mode.
	ACTION	Examine this SL and its use of unsupported system procedures to determine if these procedures are changed or eliminated in the new operating system. Use of these procedures is not supported on any version of MPE, and attempts to use them on MPE/iX may cause system failures or loss of data.
2106	MESSAGE	Program requires `;LIB=P` to satisfy external references.
	MODE	NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	This program has external references which could not be found in the system SL. An SL was found in the PUB group of this account which contained the necessary routines. In order to run this program on the current system, you must run with either `;LIB=P` or `;LIB=G`.
	ACTION	Ensure that the SL file in the PUB group of this account is migrated along with the program. If migrating to NM, you should decide whether to migrate the SL routine(s) to a NM library at the same time or to use Switch procedures to access the CM SL.
2107	MESSAGE	Program requires `;LIB=G` to satisfy external references.
	MODE	NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	This program has external references which could not be found in the system SL. An SL was found in the local group of this account which contained the necessary routines. In order to run this program on the current system, you must run with `;LIB=G`.
	ACTION	Ensure that the SL file in the local group of this account is migrated along with the program. If migrating to NM, you should decide whether to migration the SL routine(s) to a NM library at the same time or to use Switch procedures to access the CM SL.

2108	MESSAGE	Group SL requires `PUB` SL routines.
	MODE	NM incompatibility detected by MPT (severity level POSSIBLE).
	CAUSE	A group SL has external references which are not satisfied from the system SL. An SL was found in the `PUB` group of this account which satisfies these references.
	ACTION	Ensure that the `PUB` SL is migrated at the same time as the group SL. Programs which reference routines from this group SL may also require the `PUB` SL to be present.
2999	MESSAGE	`:RUN` ??? other selected program(s).
	MODE	CM and NM incompatibility detected by MPT (severity level ERROR).
	CAUSE	A job stream or UDC file has run a program which was selected by the System Manager to be flagged.
	ACTION	Consult your System Manager for a list of such programs and the action to take when they are encountered.

E

Notes on Migrating VPLUS

For the most part, VPLUS operates the same on MPE/iX-based machines as it does on MPE V/E-based machines. This is because all of VPLUS operates in CM. NM stubs are supplied for all of the VPLUS intrinsics so that they may be called from NM applications. The following VPLUS utilities are available in CM:

- FORMSPEC
- REFSPEC
- REFORMAT
- ENTRY

Migration Issues

The following headings discuss VPLUS migration issues.

Terminal Configuration

All terminals must be configured for XON/XOFF handshaking.

Supported Terminals

The following terminals are supported:

- HP 2392A
- HP 2394A
- HP 150
- HP 2393A
- HP 2397A

- HP 2625A
- HP 2628A
- HP 7009X

The following terminals are supported if the firmware revision conditions in parentheses are met:

- HP 2622A (rev D 1818-3199)
- HP 2623A (rev B 1818-3223)
- HP 2624B (rev B 1818-3139)
- HP 2627A (rev B 1818-3487)

Unsupported Terminals

The following terminals are not supported:

- HP 2382A
- HP 2621A
- HP 2621B
- HP 2621P
- HP 2626A
- HP 2626W
- HP 264x
- HP 3075
- HP 3076
- HP 3081A

Pascal Integers

In HP Pascal/iX, all references to integers in the `COMAREA` must be of type `SHORTINT` (a new 2-byte compiler-defined type for the HP Pascal/XL compiler) as opposed to type `INTEGER` (-32768 ... 32767) which is interpreted as 4 bytes.

Data Alignment

VPLUS intrinsics continue to expect data that is aligned on 16-bit boundaries. For example, in a Pascal program, the record structures declared for the VPLUS `INFO` intrinsics may need to be packed to ensure 16-bit data alignment.

Real Data Types

Real data type conversions continue to assume Hewlett-Packard real format for `VGETREAL`, `VGETLONG`, `VPUTREAL`, and `VPUTLONG`. New IEEE conversion intrinsics (`VGETIEEEREAL`, `VGETIEEELONG`, `VPUTIEEEREAL`, and `VPUTIEEELONG`) are available.

NM Stubs

Programs with incorrect parameter values which run successfully in CM may not run in NM. Many of these incorrect parameters will be detected by the native mode stubs which have an additional level of parameter checking not present in the VPLUS intrinsics.

Terminal Refresh

After a powerfail or hard reset, refresh works differently on an MPE/iX-based system than it does on an MPE V/E-based system. On MPE/iX, the terminal falls into character mode while the driver still expects block mode transmissions. A singular pressing of a terminal function key will not be "heard". The user must additionally terminate the read by pressing Shift CTRL 6, the record separator control character.

Vectra With Type Ahead Enabled

Verify your Vectra's terminal configuration before using it to run block mode applications. Ensure that the "Type Ahead" field on the Terminal Configuration Screen is set to NO. To verify the current value for "Type Ahead", return to Advancelink Main. Press the F6 key (the function) Again press key labeled "Terminal"). Again press the F6 key (now labeled the "Config Keys"). Press the F5 key to get to the Terminal Configuration Screen. If the "Type Ahead"

field contains YES, change the value to NO. Save the new value by pressing the [F1] key.

If a block mode application is run on a Vectra while the "Type Ahead" value is set to YES, a form will be painted. However, the keyboard will remain in a locked state. To clear this terminal hang, press [CTRL] [F10] to flush the type ahead buffer. Next, press the [F8] key to exit the application. Follow the directions above to reconfigure the "Type Ahead" field on the Terminal Configuration Screen.

Call Intrinsic

In general, the NM COBOL II/XL compiler requires that intrinsics be called with the Call Intrinsic mechanism. The only exceptions to this are the VPLUS and IMAGE intrinsic calls. For compatibility reasons the COBOL II/XL compiler will recognize these calls and assume Call Intrinsic. A warning will be issued when using Call Intrinsic on MPE V/E. While Customers preparing for migration of COBOL applications are not required to use the Call Intrinsic mechanism for VPLUS and IMAGE intrinsics, they must still plan to use the Call Intrinsic mechanism for other intrinsics.

Note

On MPE V, the Call Intrinsic mechanism examines the `SPLINTR` file and determines the type of addressing that the intrinsic expects.

VPLUS examines the language ID and, if it is 0 (that is, COBOL), VPLUS assumes all the addresses are word addresses. If the Call Intrinsic mechanism is used, this is an invalid assumption and erroneous results are likely. Therefore, the programmer should "fool" VPLUS by using a language ID of 5 (Pascal).

On MPE/iX, the COBOL II/XL compiler generates byte addresses and the VPLUS stub will prevent VPLUS from doubling the address (in an attempt to convert from word addresses to byte addresses).

Non-VPLUS I/O

Often, application designers resort to using FREAD, FWRITE, and input/output verbs of a programming language to display and retrieve information from the terminal within the VPLUS context. The situation becomes more complex when the application also controls the terminal settings as well as the driver settings. A likely end result is an application which runs in a restricted environment, such as on an ATP, and on one kind of terminal only. The same application may not run in a different environment such as over X.25 or on a different kind of terminal. Two new VPLUS procedures are provided to assist the application designer in producing more portable and more easily maintainable applications. These two procedures are:

- VTURNOFF
- VTURNON

VOPENTERM configures the driver and the terminal for block mode access. VCLOSETERM configures the terminal and driver for character mode access. These procedures initialize the terminal in many ways including clearing the screen image. VTURNON and VTURNOFF reconfigure the terminal and driver but leave the terminal screen image intact. The procedure VTURNOFF allows the application to momentarily switch to character mode from block mode without disturbing the screen. A call to VTURNON will reconfigure the terminal back to block mode.

For example, a program which accesses a printer slaved off a terminal while that terminal is using VPLUS should use VTURNON and VTURNOFF. VTURNOFF would be called, followed by the FWRITEs to the printer. Then VTURNON should be called.

Refer to the VTURNON and VTURNOFF intrinsics descriptions later in this appendix for further discussion.

VGETIEEEREAL

The VGETIEEEREAL intrinsic copies character-coded data from a specified field in the form data buffer in memory into an application, converting the numeric value to IEEE floating-point format. The syntax and parameter descriptions for this intrinsic are provided below.

```
VGETIEEEREAL {COMAREA,FIELDNUM,VARIABLE}
```

COMAREA — Must be COMAREA name specified when the forms file was opened with VOPENFORMF. If not already set, the following COMAREA items must be set before calling VGETIEEEREAL:

- CSTATUS — Set to zero.
- COMAREALEN — Set to total number of 2-byte words in COMAREA.

VGETIEEEREAL may set the following COMAREA fields:

- CSTATUS — Set to nonzero value if call is unsuccessful or if field's error flag is set.

FIELDNUM — Two-byte integer variable containing the field number assigned by FORMSPEC to the field in the data buffer from which the value is retrieved. Note that the field identified by FIELDNUM may not be longer than 80 characters.

VARIABLE — Variable of type IEEE floating-point real number (32-bits) in which the converted value will be placed.

This intrinsic reads the field identified by its field number from the data buffer. (Note that this field number is a unique number assigned to each field by FORMSPEC and is totally independent of the field

position in the data buffer.) The field's value must be numeric, but its data type need not be. That is, numbers in a character type field can be converted.

The numeric value, stored in the buffer in character coded form, is converted to IEEE format and then moved to the variable in the application. If errors occur during conversion, CSTATUS is set to an error code. If the requested field's error flag is set, its value is moved to the variable, but CSTATUS is set to an error code.

Note that a field's number never changes unless the batch command RENUMBER is used. It is not changed, even if the position of the field in the form is changed or its length or other characteristics are changed. The field number should not be confused with the position of the field in the data buffer, which is based on the field position within the form. Thus, the field number provides a way to locate fields regardless of their position.

The following examples illustrate a call to VGETIEEEREAL using common programming languages:

COBOL:

```
CALL "VGETIEEEREAL" USING COMAREA FIELDNUM VARIABLE.
```

FORTRAN:

```
CALL VGETIEEEREAL(COMAREA,FIELDNUM,VARIABLE)
```

Pascal:

```
VGETIEEEREAL(COMAREA,FIELDNUM,VARIABLE);
```

VGETIEEELONG

The VGETIEEELONG intrinsic copies character-coded data from a specified field in the form data buffer in memory into an application, converting the numeric value to IEEE long floating-point format. The syntax and parameter descriptions for this intrinsic are described below.

```
VGETIEEELONG {COMAREA,FIELDNUM,VARIABLE}
```

COMAREA — Must be the COMAREA name specified when the forms file was opened with VOPENFORMF. If not already set, the following COMAREA items must be set before calling VGETIEEELONG:

- CSTATUS — Set to zero.
- COMAREALEN — Set to total number of 2-byte words in COMAREA.

VGETIEEELONG may set the following COMAREA fields:

- CSTATUS — Set to non-zero value if call is unsuccessful or if field's error flag is set.

FIELDNUM — Two-byte integer variable containing the field number assigned by FORMSPEC to the field in the data buffer from which the value is retrieved. Note that the field identified by FIELDNUM may not be longer than 80 characters.

VARIABLE — Variable of type IEEE long floating-point real (64-bits) in which the converted value will be placed.

This intrinsic reads the field identified by its field number from the data buffer. (Note that this field number is a unique number assigned to each field by

FORMSPEC and is totally independent of the field position in the data buffer). The field's value must be numeric, but its data type need not be. That is, numbers in a character type field can be converted.

The numeric value, stored in the buffer in character-coded form, is converted to IEEE long format and then moved to the variable in the application. If errors occur during conversion, `CSTATUS` is set to an error code. If the requested field's error flag is set, its value is moved to the variable, but `CSTATUS` is set to an error code.

Note that a field's number never changes unless the batch command `RENUMBER` is used. It is not changed even if the position of the field in the form is changed, or its length or other characteristics are changed. The field number should not be confused with the position of the field in the data buffer, which is based on the field position within the form. Thus, the field number provides a way to locate fields regardless of their position.

The following examples illustrate a call to `VGETIEEELONG` using common programming languages:

COBOL:

```
CALL "VGETIEEELONG" USING COMAREA FIELDNUM VARIABLE.
```

FORTRAN:

```
CALL VGETIEEELONG(COMAREA,FIELDNUM,VARIABLE)
```

Pascal:

```
VGETIEEELONG(COMAREA,FIELDNUM,VARIABLE);
```

VPUTIEEEREAL

The VPUTIEEEREAL intrinsic writes a floating-point number in IEEE standard format from an application to a specified field in the form data buffer in memory, converting the value to character-set-coded external representation. The syntax and parameter descriptions for this intrinsic are provided below.

```
VPUTIEEEREAL {COMAREA,FIELDNUM,VARIABLE}
```

COMAREA — Must be the COMAREA name specified when the forms file was opened with VOPENFORMF. If not already set, the following COMAREA items must be set before calling VPUTIEEEREAL:

- CSTATUS — Set to zero.
- COMAREALEN — Set to total number of 2-byte words in COMAREA.

VPUTIEEEREAL may set the following COMAREA fields:

- CSTATUS — Set to non-zero value if call is unsuccessful.
- NUMERRS — Will be decremented if new value replaces the value of a field with an error.

FIELDNUM — Two-byte integer variable containing the field number assigned by FORMSPEC to the field in the data buffer to which the is written. Note that the field identified by FIELDNUM may not be longer than 80 characters.

VARIABLE — Variable of type IEEE floating-point real number (32-bits) that contains the value to be converted to character-set-coded

external representation and copied to a field in the data buffer.

This procedure converts an IEEE floating-point real number to its character-coded form and writes the converted value to a particular field in the data buffer, right justified. The exact format of the written data depends on the type of the destination field.

For example, if the number "34.56" were to be put in a field of type `DIG`, the result would be "34", since a field of type `DIG` may only contain integer values. The destination field is identified by the field number assigned by FORMSPEC. The field to which the value is written must be defined as a numeric field (type `NUM`,`IMP`, or `DIG`).

Note that a field's number never changes unless the batch command `RENUMBER` is used. It is not changed, even if the position of the field in the form is changed or its length or other characteristics are changed. The field number should not be confused with the position of the field in the data buffer, which is based on the field position within the form. Thus, the field number provides a way to locate fields regardless of their position. If the specified field had an error, then `VPUTIEEEREAL` will clear the field's error flag and decrement `NUMERRS`.

The following examples illustrate a call to `VPUTIEEEREAL` using common programming languages:

COBOL:

```
CALL "VPUTIEEEREAL" USING COMAREA FIELDNUM VARIABLE.
```

FORTRAN:

```
   CALL VPUTIEEEREAL(COMAREA,FIELDNUM,VARIABLE)
```

Pascal:

```
VPUTIEEEREAL(COMAREA,FIELDNUM,VARIABLE);
```

VPUTIEEELONG

Writes a floating point number in IEEE long standard format from an application to a specified field in the form data buffer in memory, converting the value to character-set-coded external representation. The syntax and parameter descriptions for this intrinsic are provided below.

```
VPUTIEEELONG {COMAREA,FIELDNUM,VARIABLE}
```

`COMAREA` Must be the `COMAREA` name specified when the forms file was opened with `VOPENFORMF`. If not already set, the following `COMAREA` items must be set before calling `VPUTIEEELONG`:

`CSTATUS`	Set to zero.
`COMAREALEN`	Set to total number of 2-byte words in `COMAREA`.

`VPUTIEEELONG` may set the following `COMAREA` fields:

`CSTATUS`	Set to non-zero value if call is unsuccessful.
`NUMERRS`	Will be decremented if new value replaces the value of a field with an error.

`FIELDNUM` Two-byte integer variable containing the field number assigned by FORMSPEC to the field in the data buffer to which the value is written. Note that the field

identified by FIELD may not be longer than 80 characters.

VARIABLE	Variable of type IEEE long floating-point real numbers (64-bits) that contains the value to be converted to character-set-coded external representation and copied to a field in the data buffer.

This intrinsics converts an IEEE long floating-point real number to its character-coded form and writes the converted value to a particular field in the data buffer, right-justified. The exact format of the written data depends on the type of the destination field.

For example, if the number "34.56" were to be put in a field of type DIG, the result would be "34", since a field of type DIG may only contain integer values. The destination field is identified by the field number assigned by FORMSPEC. The field to which the value is written must be defined as a numeric field (type NUM,IMP, or DIG).

Note that a field's number never changes unless the batch command RENUMBER is used. It is not changed, even if the position of the field in the form is changed or its length or other characteristics are changed. The field number should not be confused with the position of the field in the data buffer, which is based on the field position within the form. Thus, the field number provides a way to locate fields regardless of their position. If the specified field had an error, then VPUTIEEELONG will clear the field's error flag and decrement NUMBERS.

The following examples illustrate a call to VPUTIEEELONG using common programming languages:

COBOL:

```
CALL "VPUTIEEELONG" USING COMAREA FIELDNUM VARIABLE.
```

FORTRAN:

```
CALL VPUTIEEELONG(COMAREA,FIELDNUM,VARIABLE)
```

Pascal:

```
VPUTIEEELONG(COMAREA,FIELDNUM,VARIABLE);
```

VTURNOFF

The VTURNOFF intrinsic turns off VPLUS block mode and enables character mode access without disturbing the terminal screen. The syntax and parameter descriptions for this intrinsic are provided below.

```
VTURNOFF {COMAREA}
```

COMAREA — Must be COMAREA named when file was opened by VOPENTERM. If not already set, the following COMAREA items must be set before calling VTURNOFF.

CSTATUS	Set to zero.
COMAREALEN	Set to total number of 2-byte words of COMAREA.

VTURNOFF may set the following COMAREA items:

CSTATUS	Set to nonzero value if call is unsuccessful.
FILERRNUM	Set to file error code if MPE file error.

VTURNOFF is used for momentarily switching from VPLUS block mode to character mode. This procedure is designed for use after a terminal has been previously opened by VOPENTERM or after a VTURNON.

VTURNOFF reconfigures the terminal and driver for character mode operations without disturbing the screen image on the terminal. The following operations normally performed in VCLOSETERM are not performed in VTURNOFF:

- Clear local form storage
- Enable the USER/SYSTEM keys
- Disable touch reporting, delete touch fields
- Clear screen
- Unlock keyboard
- Close terminal file

Note that VTURNOFF does not close the terminal file. To close the file and completely reset the driver and the terminal, VCLOSETERM must be used.

The following examples illustrate a call to VTURNOFF:

COBOL:

```
CALL "VTURNOFF" USING COMAREA.
```

BASIC:

```
200 CALL VTURNOFF(C(*))
```

FORTRAN:

```
CALL VTURNOFF(COMAREA)
```

SPL:

```
VTURNOFF(COMAREA);
```

Pascal:

```
VTURNOFF(COMAREA);
```

VTURNON

The VTURNON intrinsic turns on VPLUS block mode without disturbing the terminal screen. The syntax and parameter descriptions for this intrinsic are provided below.

```
VTURNON {COMAREA,TERMFILE}
```

`COMAREA` The `COMAREA` name must be unique for each open forms file. The `COMAREA` must be the same `COMAREA` used in `VOPENTERM`. The following `COMAREA` items must be set before the call, if not already set:

`CSTATUS`	Set to zero.
`LANGUAGE`	Set to code that identifies the programming language of the calling program.
`COMAREALEN`	Set to total number of 2-byte words in `COMAREA`.

`VTURNON` may set the following `COMAREA` fields:

`CSTATUS`	Set to nonzero value if call is unsuccessful.
`FILERRNUM`	Set to file error code if MPE file error.
`FILEN`	Set to MPE file number of terminal .
`IDENTIFIER`	Set to appropriate VPLUS/V terminal ID.
`LAB'INFO`	Set to appropriate number and length of labels.

TERMFILE	Must be the same terminal file name used in VOPENTERM.

VTURNON is normally used in an application when the terminal is already opened by VOPENTERM, and VTURNOFF was called to switch out of VPLUS block mode. VTURNON switches the application back to VPLUS block mode without disturbing the image on the terminal screen.

VTURNON reconfigures the terminal and the driver without performing the following operations which are normally performed by VOPENTERM:

- Initialize local form storage
- Clear screen
- Enable the USER function keys
- Disable or enable the USER/SYSTEM key, as specified in the SHOWCONTROL word

Unlike VOPENTERM, VTURNON will not ask you to press the BLOCK MODE key, if you are using an HP 2640B or HP 2644 terminal when the terminal is not in block mode.

The following examples illustrate a call to VTURNON using common programming languages:

COBOL:

```
CALL "VTURNON" USING COMAREA, T1.
```

BASIC:

```
90 T1$="  "

100 CALL VTURNON(C(*),T1$)
```

FORTRAN:

```
T1="  "

VTURNON(COMAREA,T1);
```

SPL:

```
MOVE T1:="  ";

VTURNON(COMAREA,T1);
```

Pascal:

```
T1:=' ';

VUTURNON(COMAREA,T1);
```

VBLOCKWRITE

The VBLOCKWRITE intrinsic writes a block of characters to a terminal in block mode. The syntax and parameter descriptions for this intrinsic are provided below.

```
VBLOCKWRITE {COMAREA,BUF,LEN,TMODE,LOC,TC}
```

COMAREA — The following COMAREA fields must be set before calling VBLOCKWRITE, if not already set:

- LANGUAGE — Set to code identifying the programming language of the calling program.
- COMAREALEN — Set to total number of 2-byte words in COMAREA.

VBLOCKWRITE may set the following COMAREA fields:

- CSTATUS — Set to nonzero value if call is unsuccessful.
- FILERRNUM — Set to file error code if MPE file error.

`BUF`	Byte array containing characters to be written to the terminal.	
`LEN`	Number of bytes in the `BUF` array (2-byte integer).	
`TMODE`	Terminal mode (2-byte integer).	
	0	do not change terminal mode
	1	change to format mode
	2	change to unformatted mode
`LOC`	Start position of write (array of two 2-byte integers). Absolute cursor addressing is not allowed in format mode. An error will be returned.	
	[0] [0]	home cursor before `WRITE`
	[x] [y]	start from absolute row x, column y. (Not allowed in format mode.)
	[-1] [0]	start from current position
	[-2] [0]	start from first available line of display memory, for example, the first available line after the end of a previous form. (Not allowed in format mode.)
`TC`	Terminal control (2-byte integer)	
	0	do not lock keyboard at the beginning of write;

	unlock at the end of write.
1	lock keyboard at the beginning of write; unlock at the end of write.

This procedure writes the content of a user buffer to a terminal. `TMODE` options can be used to change the terminal to format or unformatted mode before the write. `LOC` options allow the programmer to specify the position of the screen where the write is to begin. Terminal control (`TC`) options can be used to control keyboard locking for the protection of data as it is being written to the terminal. `TC` = 1 is recommended for applications which do multiple writes to the terminal with no intervening reads. Procedures, such as `VBLOCKREAD` or `VREADFIELDS`, lock the keyboard as soon as the terminal begins transmitting data when triggered by the Enter key or a function key.

`VOPENTERM` must be called before using this procedure. This procedure is intended only for advanced programmers who are proficient with terminal control operations and VPLUS terminal settings. Terminal keyboard operations, such as PREV PAGE and NEXT PAGE, can be performed programmatically by sending the appropriate escape sequences to the terminal via `VBLOCKWRITE`. `VBLOCKWRITE` can also be used to write large blocks of unformatted text or multiple report lines in between uses of predefined VPLUS forms. To ensure portability of the application from one driver to another, alteration of terminal straps using `VBLOCKWRITE` is not recommended. See `VTURNON` and `VTURNOFF` for information on how to switch between character mode and block mode without disturbing the screen.

The following examples illustrate a call to `VBLOCKWRITE` using common programming languages:

COBOL:

```
CALL "VBLOCKWRITE" USING COMAREA @BUF LEN TMODE LOC TC.
```

BASIC:

```
CALL VBLOCKWRITE(C(*),B1$,L1,M1,L(*),T1)
```

FORTRAN:

```
CALL VBLOCKWRITE(COMAREA,BUF,LEN,TMODE,LOC,TC)
```

SPL:

```
VBLOCKWRITE(COMAREA,BUF,LEN,TMODE,LOC,TC);
```

Pascal:

```
VBLOCKWRITE(COMAREA,BUF,LEN,TMODE,LOC,TC);
```

VBLOCKREAD

The VBLOCKREAD intrinsic reads a block of characters from a terminal in block mode. The syntax and parameter descriptions for this intrinsic are provided below.

```
VBLOCKREAD {COMAREA,BUF,LEN,ACTLEN,TMODE,LOC,BC,TC}
```

COMAREA — The following COMAREA fields must be set before calling VBLOCKREAD, if not already set:

- LANGUAGE — Set to code identifying the programming language of the calling program.
- COMAREALEN — Set to total number of 2-byte words in COMAREA.

VBLOCKWRITE may set the following COMAREA fields:

CSTATUS	Set to nonzero value if call is unsuccessful.
FILERRNUM	Set to file error code if MPE file error.

BUF	Byte array to receive data from the terminal.
LEN	Maximum number of bytes to read from terminal (2-byte integer).
ACTLEN	Actual number of bytes read from terminal (2-byte integer).
TMODE	Terminal setting at the time of read (2-byte integer).

1	assume terminal is in format mode
2	assume terminal is in unformatted mode

LOC	Array of two 2-byte integers containing start position of read.

[0] [0]	home cursor before read
[-1] [0]	start from current cursor position

BC	Buffer control (2-byte integer)—not currently used. Must initialize to zero.
TC	Terminal control (2-byte integer)—not currently used. Must initialize to zero.

This intrinsic reads a block of data from the terminal with a number of options. There are two major differences between VREADFIELDS and this procedure. First, it provides more options for reading data from the terminal. Second, data read is returned directly to the

application buffer. There is no VPLUS form associated with the read.

Like the companion intrinsic, VBLOCKWRITE, this procedure is recommended only for advanced programmers who are proficient with terminal input/output. VOPENTERM must be called before using VBLOCKREAD. The keyboard must be unlocked before calling VBLOCKREAD. (Refer to keyboard unlock options in VBLOCKWRITE.) VBLOCKREAD will lock the keyboard immediately after Enter or a function key is pressed to ensure data integrity.

This procedure is primarily designed for unformatted reads. For users who do not use VREADFIELDS but use VBLOCKREAD to read data in format mode, the application data interpretation algorithm should accommodate both MDT (Modified Data Tag) and non-MDT inputs. When MDT is on, unmodified blanks and data are not transmitted from the terminal. Refer to the appropriate terminal reference manuals for further explanation of the MDT feature.

The following examples illustrate a call to VBLOCKREAD using common programming languages:

COBOL:

```
CALL "VBLOCKREAD" USING COMAREA @BUF LEN ACTLEN
                        TMODE LOC BC TC.
```

BASIC:

```
CALL VBLOCKREAD(C(*),B1$,L1,L2,M1,L(*),U1,U2)
```

FORTRAN:

```
CALL VBLOCKREAD(COMAREA,BUF,LEN,ACTLEN,TMODE,LOC,BC,TC)
```

SPL:

```
VBLOCKREAD(COMAREA,BUF,LEN,ACTLEN,TMODE,LOC,BC,TC);
```

Pascal:

```
VBLOCKREAD(COMAREA,BUF,LEN,ACTLEN,TMODE,LOC,BC,TC);
```

F

Using the GETUDC Utility

This appendix describes how to use the GETUDC utility. GETUDC is intended to be used during the preparation stage of the migration effort. GETUDC runs on MPE V/E-based systems (release G.03.01 and later). Included in this appendix are discussions on the following subjects:

- "GETUDC Operation", a general overview on when to use the tool and how the tool operates.
- "Using GETUDC", a detailed description of the user interface.
- "GETUDC Program Error Messages", a comprehensive list of error and warning messages returned by GETUDC, as well as recommended methods of recovery.

GETUDC Operation

GETUDC is an MPE V/E utility that aids you in migrating your MPE V/E User Defined Command (UDC) environment from an MPE V/E-based system to an MPE/ iX-based system. GETUDC is designed to be used by System Managers (users with SM capability) who are preparing to migrate an existing MPE V/E UDC environment to an MPE/iX-based system using DIRMIG.

The GETUDC utility places in a text file:

- The name COMMAND.PUB.SYS, a system file that keeps track of the names of all system-recognized UDC files, as well as the users associated with those UDC files.
- The names of all system-recognized UDC files located on your MPE V/E-based computer system.

Note

A system-recognized UDC file is one that a user has specified with the :SETCATALOG command as being either a system-level, account-level, or user-level UDC file.

You should use GETUDC immediately before creating the SYSDUMP tape you are planning to use with DIRMIG. When you run the MPE V/E SYSDUMP utility to create the tape, you can enter the name of the GETUDC-generated file at the following SYSDUMP prompt:

```
ENTER DUMP FILE SUBSETS
```

Using the GETUDC-generated file as an "indirect STORE file" (a file containing valid responses to the above SYSDUMP prompt) ensures that files are placed at the beginning of the STORE portion of the SYSDUMP tape in the order you specified in the indirect STORE file (COMMAND.PUB.SYS followed by UDC files).

When migrating the MPE V/E UDC environment to an MPE/iX system, DIRMIG invokes the RESTORE facility twice; once to restore COMMAND.PUB.SYS, and once to restore UDC files whose names DIRMIG reads from COMMAND.PUB.SYS. Placement of COMMAND.PUB.SYS and UDC files at the beginning of the STORE portion of the SYSDUMP tape greatly reduces the amount of time DIRMIG takes to search for and restore these files.

Using GETUDC

Initiate GETUDC by running the program file `GETUDC.PUB.SYS` either interactively or in batch mode. GETUDC progresses through the following steps in order to accomplish its task:

1. Asks you for an MPE file name, then creates and opens an EDIT/V compatible text file you can later use as an indirect STORE file.
2. After writing the name COMMAND.PUB.SYS to the indirect STORE file, GETUDC reads system-recognized UDC file names from a temporary copy of COMMAND.PUB.SYS and writes the names to the indirect STORE file.
3. Asks you to append to the indirect STORE file additional file names you plan to store to the STORE portion of the SYSDUMP tape.

Once these steps are accomplished, GETUDC saves the indirect STORE file, then terminates.

Creating the Indirect STORE File

GETUDC prompts you for a file name which will hold the gathered UDC file names:

```
Enter file name to hold UDC file names [enter ? for help]
>>
```

The possible responses to this prompt are:

- Any legal MPE file designator, followed by a Return. The default group and account are the logon group and account. If the account is specified, it must be that of the user's logon account. You can specify the file designator with a leading asterisk (*) if you want to backreference a file equation in order to override default file characteristics.

- A question mark (?) displays pertinent HELP text, then returns you to the prompt.
- A Return terminates GETUDC (a file is not created).

If you specify an illegal file name, or if a duplicate file name already exists on the system, GETUDC displays an error message and prompts you again for a file name.

Once a file name is selected, GETUDC creates an EDIT/V-compatible text file you can later use as an indirect STORE file at the appropriate SYSDUMP prompt.

Gathering UDC File Names

In order to minimize possible impact upon system users, GETUDC copies the contents of `COMMAND.PUB.SYS` to a temporary file. GETUDC reads UDC file names from this temporary file.

The file name `COMMAND.PUB.SYS` is copied to the first line of the indirect STORE file. GETUDC then displays the following progress message:

```
0000  Number of UDC Files Collected in The System
```

The number in the progress message automatically increments to indicate the number of UDC file names collected from the temporary copy of `COMMAND.PUB.SYS` and placed (on a file name per line basis) in the indirect STORE file.

To ensure system security, GETUDC does not write lockwords to the indirect STORE file. (Lockwords will not be required for SYSDUMP to successfully STORE the files.)

GETUDC displays the following message after it finishes writing UDC file names:

`All` *#nnnn* `UDC file names are stored in file` *filename*

A lockword, if previously specified, is not displayed in the above message.

Appending Other File Sets

You can append additional file names to the indirect STORE file. When the indirect STORE file name is used as a response to the `ENTER DUMP FILE SUBSETS` prompt of the SYSDUMP, these files will be stored in the STORE portion on the SYSDUMP tape immediately following `COMMAND.PUB.SYS` and UDC files.

Note

The `ENTER DUMP FILE SUBSETS` prompt of SYSDUMP is not repeated after the first time you specify the filesets (or indirect file name containing the file sets) you want stored. It is recommended that you place in the indirect STORE file the names of all the file sets you want SYSDUMP to store.

GETUDC displays this prompt:

```
Enter any file sets you want to append [ enter ? for help ]
>>
```

The possible responses to this prompt are:

- Any STORE/RESTORE-compatible file set name. Please enter this data correctly, as GETUDC does not check for incorrect names. You are prompted repeatedly until you press Return without entering a file set name.
- A question mark (?) displays pertinent HELP text, then returns you to the prompt.
- A Return causes GETUDC to save and close the indirect STORE file, terminate.

Note

When running in batch mode, a blank line in the job file (instead of a file set name) is treated as a Return.

GETUDC Execution Example

Example F-1 illustrates a typical execution of the GETUDC utility. User input is highlighted.

```
:RUN GETUDC.PUB.SYS

GETUDC  G.03.01 COPYRIGHT HEWLETT-PACKARD CO. 1987.

Enter file name to hold UDC file names [ enter ? for help ]
>>TESTFILE/LOCK

0123    Number of UDC Files Collected in The System

All 123 UDC file names are stored in file TESTFILE.

Enter any file sets you want to append [ enter ? for help ]
>> @.PUB.SYS
```

```
>> @M
>> A@#ACCT@
>> @.@.FINANCE
>>  Return

END OF PROGRAM
:
```

Example F-1. GETUDC Execution Example

GETUDC Program Error Messages

GETUDC sets the user-defined Job Control Word (JCW) GUERR to a value that indicates the error or warning that occurred. If no error or warning occurred, GUERR is set to zero.

When running interactively, some errors do not cause GETUDC to terminate. When running in batch mode, if any error is encountered, GETUDC terminates with an appropriate error message.

The following list of error and warning messages are issued by GETUDC. Error messages are in the range 0 to 1999. Warning messages are in the range 2000 to 2999.

0000 to 1000	MESSAGE	The GUERR value is equivalent to a file system error number. The returned message is a file system error message.
	CAUSE	A file system error has occurred.
	ACTION	Look up the indicated file system error in the *MPE V/E Intrinsics Reference Manual (32033#90007)* and take appropriate action.
1001	MESSAGE	CATOPEN INTRINSIC ERROR: *error number* FILE: GUCAT*nnn*.PUB.SYS UNABLE to OPEN THE MESSAGE CATALOG FILE (GUERR 1001)
	CAUSE	GETUDC is unable to open the NLS language catalog file.
	ACTION	The user should look up the indicated CATOPEN intrinsic error in the *MPE V/E Native Language Support Reference Manual (32414-90001)* and troubleshoot accordingly.
1002	MESSAGE	CATREAD INTRINSIC ERROR: *error number*FILE: GUCAT*nnn*.PUB.SYS UNABLE TO DISPLAY CATALOG MESSAGE SET = *n* MSG = *m* (GUERR 1002)
	CAUSE	GETUDC is unable to display the message desired, because a read error occurred on the catalog file.
	ACTION	The user should look up the indicated CATREAD intrinsic error in the *MPE V/E Native Language Support Reference Manual (32414-90001)* and troubleshoot accordingly.
1003	MESSAGE	SM CAPABILITY REQUIRED TO RUN THIS UTILITY. (GUERR 1003)
	CAUSE	The user does not have System Manager (SM) capability.
	ACTION	The user should have SM capability to run this utility.
1004	MESSAGE	*file system error message* (FSERR *nnn*) UNABLE TO OPEN $STDINX. (GUERR 1004)
	CAUSE	The cause is determined by the file system error returned.
	ACTION	Look up the indicated file system error, refer to the *MPE V/E Intrinsics Reference Manual (32033#90007)*, and troubleshoot accordingly.

1005	MESSAGE	*file system error message* (FSERR *nnn*) FILE: COMMAND.PUB.SYS UNABLE TO OPEN THE FILE COMMAND.PUB.SYS. (GUERR 1005)
	CAUSE	GETUDC needs the file COMMAND.PUB.SYS in order to gather all UDC files from the system. It is unable to open the file, because the file is not present or because of other file system errors.
	ACTION	Look up the indicated file system error and take appropriate action.
1006	MESSAGE	ATTEMPT TO WRITE BEYOND END OF FILE. (GUERR 1006)
	CAUSE	The file created by GETUDC is not large enough to hold all UDC file names.
	ACTION	Use a file equation to override the default file size (1023 records).
1007	MESSAGE	*file system error message* (FSERR*nnn*) FILE: *filename* FATAL ERROR RETURNED FROM FWRITE INTRINSIC. (GUERR 1007)
	CAUSE	A fatal error is returned from FWRITE intrinsic call.
	ACTION	Look up file name and the file system error, refer to the *MPE V/E Intrinsics Reference Manual (32033-90007)*, and take appropriate action.
1008	MESSAGE	*file system error message* (FSERR *nnn*) FILE: *filename* FATAL ERROR RETURNED FROM FREAD INTRINSIC. (GUERR 1008)
	CAUSE	A fatal error is returned from FREAD intrinsic call.
	ACTION	Look up file name and the file system error, refer to the *MPE V/E Intrinsics Reference Manual (32033-90007)*, and take appropriate action.
1009	MESSAGE	*file system error message* (FSERR*nnn*) FILE: *filename* FATAL ERROR RETURNED FROM FCLOSE INTRINSIC. (GUERR 1009)
	CAUSE	A fatal error is returned from FCLOSE intrinsic call.
	ACTION	Look up file name and the file system error, refer to the *MPE V/E Intrinsics Reference Manual (32033-90007)*, and take appropriate action.

1010	MESSAGE	*file system error message* (FSERR *nnn*) FILE: DUPCOMMD.*group.account* UNABLE TO OPEN TEMPORARY COPY OF COMMAND.PUB.SYS. (GUERR 1010)
	CAUSE	GETUDC copies the file COMMAND.PUB.SYS to a temporary file and scans this temporary file instead. It is unable to open this file.
	ACTION	Look up the file system error, refer to the *MPE V/E Intrinsics Reference Manual (32033-90007)*, and take appropriate action.
1011	MESSAGE	*file system error message* (FSERR *nnn*) FILE: DUPCOMMD.*group.account* UNABLE TO CLOSE DUPLICATE COPY OF COMMAND.PUB.SYS. (GUERR 1011)
	CAUSE	GETUDC copies the file COMMAND.PUB.SYS to a temporary file and scans this temporary file instead. It is unable to close this file.
	ACTION	Look up the file system error, refer to the *MPE V/E Intrinsics Reference Manual (32033-90007)*, and take appropriate action.
1012	MESSAGE	*file system error message* (FSERR *nnn*) FILE: *filename* FATAL ERROR RETURNED FROM FOPEN INTRINSIC. (GUERR 1012)
	CAUSE	The cause is determined by the file system error returned.
	ACTION	Look up the file name and file system error returned, refer to the *MPE V/E Intrinsics Reference Manual (32033-90007)*, and take appropriate action.
1013	MESSAGE	*file system error message* (FSERR *nnn*) FILE: *filename* FATAL ERROR RETURNED FROM FREADDIR INTRINSIC. (GUERR 1013)
	CAUSE	GETUDC scans the temporary copy of the file COMMAND.PUB.SYS by means of the FREADDIR intrinsic. A fatal error is returned from the intrinsic call.
	ACTION	Look up the file name and file system error returned, refer to the *MPE V/E Intrinsics Reference Manual (32033-90007)*, and take appropriate action.

1014	MESSAGE	INCORRECT FILE FORMAT FOR COMMAND.PUB.SYS. (GUERR 1014)
	CAUSE	The file COMMAND.PUB.SYS is in an incorrect file format (record size is less than 20 words and/or the file is an ASCII file).
	ACTION	Purge the file COMMAND.PUB.SYS, rebuild the file in correct format, and perform :SETCATALOG commands appropriately to recreate the UDC environment in the system.
1015	MESSAGE	COMMAND.PUB.SYS IS CORRUPTED. (GUERR 1015)
	CAUSE	The file COMMAND.PUB.SYS is corrupted. (Maximum number of entries in the file is greater than number of records (EOF) of the file.)
	ACTION	Purge the file COMMAND.PUB.SYS, rebuild the file in correct format, and perform :SETCATALOG commands appropriately to recreate the UDC environment in the system.
2001	MESSAGE	CATCLOSE INTRINSIC ERROR: *error number* FILE: GUCAT*nnn*.PUB.SYS UNABLE TO CLOSE THE MESSAGE CATALOG FILE. (GUWARN 2001)
	CAUSE	The cause is determined by the error number returned.
	ACTION	No action required.
2002	MESSAGE	*file system error message* (FSERR *nnn*) FILE: COMMAND.PUB.SYS UNABLE TO CLOSE THE FILE COMMAND.PUB.SYS.(GUWARN 2002)
	CAUSE	The cause is determined by the file system error.
	ACTION	No action required.
2003	MESSAGE	*file system error message* (FSERR *nnn*) FILE: *filename* UNABLE TO CLOSE $STDINX. (GUWARN 2003)
	CAUSE	The cause is determined by the file system error.
	ACTION	No action required.

2004	MESSAGE	COMMAND.PUB.SYS IS EMPTY. (GUWARN 2004)
	CAUSE	The file COMMAND.PUB.SYS is empty.
	ACTION	No action required.
2005	MESSAGE	ERROR RETURNED FROM FCHECK INTRINSIC. (GUWARN 2005)
	CAUSE	Unknown.
	ACTION	No action required.
2006	MESSAGE	*file system error message* (FSERR *nnn*) FILE: DUPCOMMD.*group.account* UNABLE TO CLOSE TEMPORARY COPY OF COMMAND.PUB.SYS. (GUWARN 2006)
	CAUSE	GETUDC is unable to close the temporary copy of COMMAND.PUB.SYS after the scan is completed.
	ACTION	No action required.

G

Using the Directory Migration Tool

DIRMIG is intended to be used during the installation stage of the migration process, after the MPE/iX operating system has been installed and configured on a 900 Series HP 3000 computer system. DIRMIG runs on MPE iX-based systems. Included in this appendix are discussions on the following subjects:

- "DIRMIG Operation", a general overview of the tool and how it operates.
- "Using DIRMIG", a detailed description of the user interface.
- "Using VOLUTIL to Complete Private Volume Migration", a detailed description of how to use the MPE/iX utility VOLUTIL to complete the task started by DIRMIG.
- "DIRMIG Program Error Messages", a comprehensive list of diagnostic error messages returned by DIRMIG, as well as recommended methods of error recovery.

DIRMIG Operation

DIRMIG is an MPE/iX tool you use to migrate an MPE V/E operating environment (U-MIT or later) to an MPE/iX-based computer system. DIRMIG provides a friendly, menu-driven interface that enables you to easily migrate operating environment information located on the MPE V/E SYSDUMP tape created during the preparation stage of the migration process. Refer to the discussion of creating the SYSDUMP tape in Chapter 3, "Preparation".

DIRMIG enables you to migrate to the MPE/iX-based system these MPE V/E operating environment components:

- Global RINs.
- User-logging identifiers.
- Directory structures, including account, group, and user characteristics.
- Private volume environments associated with the migrated directory structures.
- UDC environments and user files associated with the migrated directory structures.

DIRMIG's menu-driven interface gives you the flexibility to select different migration paths. You can choose to migrate the entire MPE V/E operating environment at once, or migrate components separately. Figure G-1 illustrates the different migration paths available in DIRMIG.

You will commonly run DIRMIG immediately after the MPE/iX operating system has been successfully installed on the 900 Series HP 3000 computer system. However, if you plan to migrate the MPE V/E operating environment incrementally, you can run DIRMIG after any MPE/iX startup procedure.

DIRMIG creates a log file each time it executes. DIRMIG log files contain migration progress messages as well as error messages about components that could not be migrated successfully. The log file provides information for debugging and troubleshooting if a problem occurs during migration. The DIRMIG log files are numbered so that you can maintain a log file history of every execution of DIRMIG. You can also enable the MPE/iX tracing facility to provide additional troubleshooting information by running DIRMIG with the optional `PARM=` parameter equal to one. (For more information about DIRMIG log files, refer to "Examining DIRMIG Log Files".)

Who Can Use DIRMIG

DIRMIG should be run during the installation stage of the migration process by a person with both System Manager (SM) and System Supervisor (OP) capabilities. This person should have knowledge of:

- The overall migration process developed by the Migration Project Team. The Migration Project Team is described in Chapter 2, "Analysis and Planning".
- Detailed knowledge of the type of operating environment planned for the MPE/iX-based system.

DIRMIG Security Considerations

To protect itself against accidental user intervention, DIRMIG enforces the following security provisions:

- The user must be logged on to `MANAGER.SYS,PUB` and have both System Manager (SM) and System Supervisor (OP) capabilities.
- No other jobs and sessions may be logged on to the system (other than the session running DIRMIG either interactively or streaming the one job running DIRMIG in batch mode).
- `CONTROL-Y` is not armed, although `BREAK` remains enabled.

- There is no access to MPE/iX commands.

Caution

You should not interrupt or abnormally terminate DIRMIG once it is started, since it is running in Privileged Mode (PM) updating MPE/iX system tables and allocating directory space. Instead, use the exit options provided to exit DIRMIG safely.

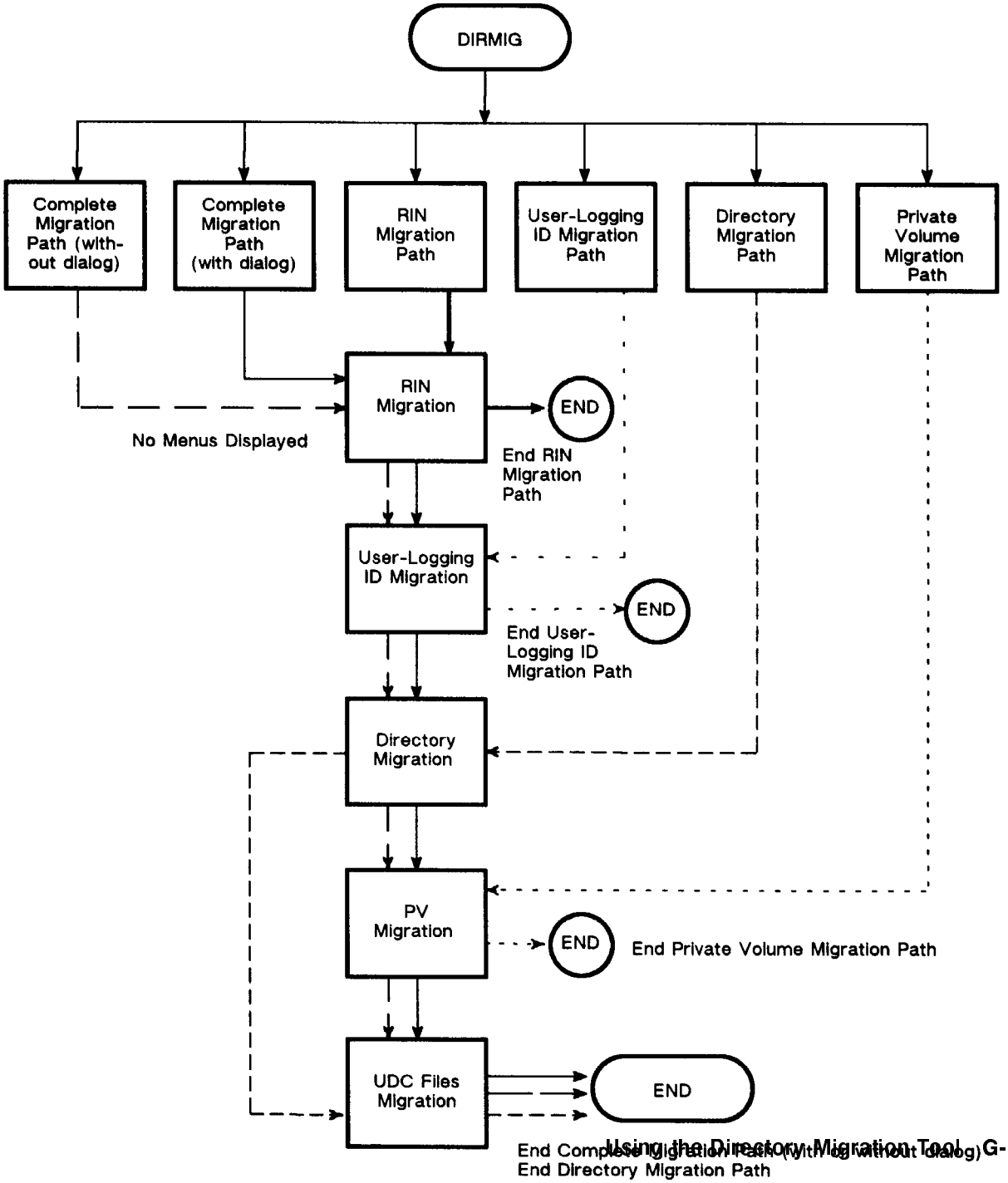


Figure G-1. Migration Paths Available in DIRMIG

If DIRMIG is executed interactively, it is recommended that the system be in "single-user mode". Single-user mode is achieved by executing the `LOGOFF` command at the system console:

```
CTRL A
=LOGOFF #Snnn
```

where *#Snnn* is the session number assigned to `MANAGER.SYS` which remains logged on. Single-user mode guarantees that only one user is on the system, and that no one else can log on until you disable single-user mode after DIRMIG finishes execution. (You can also use the `:LIMIT` command to prevent other users from logging on.)

If DIRMIG is executed in batch mode, it is recommended that the only session on the system be the session that streamed the DIRMIG job.

Note

It is recommended that you run DIRMIG from the System Console where you can maintain closer control over your MPE/iX system, as well as monitor and reply to DIRMIG tape mount requests.

Using DIRMIG

Follow these steps to correctly start DIRMIG:

1. Log on as `MANAGER.SYS,PUB`. Confirm that you have both SM and OP capabilities.
2. Ensure that all other jobs and sessions are logged off before running DIRMIG.
3. If device class `TAPE` is not configured on your system, create a file equation to redirect `MAGTAPE` (DIRMIG's formal designator for the tape device) to the device class or logical device number of your configured tape drive. For example:

```
:FILE MYTAPE;DEV=MAGTAPE
:FILE MYTAPE=*MYTAPE
```

4. If device class `LP` is not configured for your system, create a file equation to redirect `LIST` (DIRMIG's formal designator for the offline listing device) to the device class or logical device number for your line printer or an alternate device (refer to "Redirecting DIRMIG Output").
5. Enter `:RUN DIRMIG.PUB.SYS`. DIRMIG displays the Main Menu where you can select your migration options.

Under subsequent headings you will find discussions on how to use DIRMIG's menu-driven interface to select, then implement, the MPE V/E operating environment migration options of your choice:

1. Selecting Your Migration Options from the Main Menu
2. Migrating Without Dialog
3. Migrating With Dialog
4. Migrating Global RINs
5. Migrating User-Logging Identifiers
6. Migrating the Directory
7. Migrating Private Volume Environments
8. Migrating UDCs and User Files

Selecting Your Migration Options from the Main Menu

Once you have successfully passed DIRMIG's security checks, DIRMIG displays first the product banner and the Main Menu (shown in Figure G-2). From this menu, you can select the directory migration path you wish.

```
============================================================
DIRMIG.PUB.SYS HP30362A.00.00 (c) HEWLETT PACKARD CO., 1987.
MPE/iX Migration Utility. WED, SEP 23, 1987, 4:55 PM
============================================================
Current Log File:  DIRLOG03.PUB.SYS

CHOOSE FROM MIGRATION OPTIONS BELOW:
0        -  EXIT
1        -  HELP
2        -  COMPLETE MIGRATION (No Dialog)
3        -  COMPLETE MIGRATION (With Dialog)
4        -  RINS
5        -  USER LOGGING ID'S
6        -  DIRECTORY
7        -  PRIVATE VOLUME ENVIRONMENT
ENTER MIGRATION OPTION (S)
..
```

Figure G-2. DIRMIG Main Menu

You are being asked which migration paths you want to take. At the `ENTER MIGRATION OPTION(S)` prompt, you can select options 0, 1, 2, and 3 by themselves or select one or more of options 4, 5, 6, and 7 in the following formats:

- Option numbers separated by commas. The example below indicates you selected options 4, 6, and 7:

```
ENTER MIGRATION OPTION(S)
..4, 6, 7
```

- Option number separated by a "/" to show a range of selections. The example below indicates you selected options 4, 5, 6, and 7:

```
ENTER MIGRATION OPTION(S)
..4/7
```

If you select a migration path that requires the SYSDUMP tape (options 2, 3, 4, 5, or 6), DIRMIG instructs you to mount the SYSDUMP tape and place it online:

```
MOUNT MPE SYSDUMP TAPE AND PLACE TAPE DRIVE ON-LINE.
```

Mount and place online the MPE V/E SYSDUMP tape created during the preparation stage of the migration process on a reel-to-reel magnetic tape drive.

- If you select option 0 from the Main Menu, DIRMIG terminates. No migration occurs.
- If you select option 1 from the Main Menu, the HELP facility is invoked. When you exit the HELP facility you return to the Main Menu.
- If you select option 2 from the Main Menu, DIRMIG proceeds to migrate the entire MPE V/E operating environment found on the SYSDUMP tape without allowing user interaction. Refer to "Migrating Without Dialog".

- If you select option 3 from the Main Menu, DIRMIG enables you to step through each of the optional menus in turn so that you can implement your own migration strategy. Refer to "Migrating with Dialog".
- If you select option 4 from the Main Menu, you proceed directly to the Global RIN Migration Menu. Refer to "Migrating Global RINs".
- If you select option 5 from the Main Menu, you proceed directly to the User-Logging ID Migration Menu. Refer to "Migrating User-Logging Identifiers".

- If you select option 6 from the Main Menu, you proceed directly to the Directory Migration Menu. Refer to "Migrating the Directory". If you choose to migrate one or more accounts, the UDC Environment/User Files Migration Menu is displayed. Refer to "Migrating UDC Environments and User Files".
- If you select option 7 from the Main Menu, you proceed directly to the Private Volume Migration Menu. Refer to "Migrating Private Volume Environments".

Note

You can select option 7 from the Main Menu only if the file `PVASSIST.PUB.SYS` has already been created by DIRMIG. `PVASSIST.PUB.SYS` is created when DIRMIG encounters the first volume set definition when migrating accounts during directory migration. That volume set definition, and subsequent volume set definitions, are appended to `PVASSIST.PUB.SYS`. Future executions of DIRMIG will append volume set definitions to this file.

Migrating Without Dialog

If you select option 2 from the Main Menu, DIRMIG proceeds to migrate the entire operating environment found on the SYSDUMP tape. DIRMIG accomplishes the following tasks without interactive dialog:

- Migrates all MPE V/E global RINs.
- Migrates all MPE V/E user-logging identifiers.
- Migrates the entire MPE V/E system domain directory/UDC environment.
- Restores all user files residing on the tapes (except the MPE/iX reserved groups and accounts `MPEiX.SYS`, `CONFIG.SYS`, `DIAG.SYS`, `CONFG950.SYS`, and `CONFG925.SYS`)
- Writes volume set information to `PVASSIST.PUB.SYS` and generates a `PVSUMARY.PUB.SYS` command file for

all volume sets found. (Refer to "Migrating Private Volume Environments" for details about `PVASSIST` and `PVSUMARY` files.)

Migrating With Dialog

If you select option 3 from the Main Menu, DIRMIG enables you to step through each of the following menus in turn so that you can choose your own migration path. Once you are finished with each optional menu, DIRMIG moves you to the next menu on the path:

1. Global RIN Migration Menu (refer to "Migrating Global RINs").
2. User-Logging ID Migration Menu (refer to "Migrating User-Logging Identifiers").
3. Directory Migration Menu (refer to "Migrating the Directory").
4. Private Volume Migration Menu is displayed only if the file `PVASSIST.PUB.SYS` exists (refer to "Migrating Private Volume Environments").
5. UDC Environment/User Files Migration Menu is displayed only if one or more accounts were migrated using DIRMIG's Directory Migration Menu (refer to "Migrating the UDC Environment and User Files").

Migrating Global RINs

DIRMIG displays the RIN Migration Menu shown in Example G-2 after one of the following actions has occurred:

- You selected option 3 from the Main Menu; this is the first menu DIRMIG displays after you select option 3 from the Main Menu.
- You selected option 4 from the Main Menu.

Selecting Your Menu Option

From the RIN Migration Menu (shown in Figure G-3), you can choose the global RIN migration option you wish.

```
RIN MIGRATION
-------------
CHOOSE FROM RIN OPTIONS BELOW:

      0     - EXIT / (No RIN Migration)
      1     - HELP
      2     - LIST GLOBAL RINS
      3     - DELETE GLOBAL RINS
      4     - MIGRATE RINs

ENTER RIN OPTION
>>
```

Figure G-3. RIN Migration Menu

DIRMIG prompts you to select one of the above options. Entering only a Return at the `ENTER RIN OPTION` prompt returns you to the same prompt. The following headings explain the different options available to you through the RIN Migration Menu.

Exit/No RIN Migration (Option 0)

If you select option 0 from the RIN Migration Menu, DIRMIG does not migrate RINs. DIRMIG's next action is dependent upon which option you previously selected from the Main Menu:

- If you selected option 3 from the Main Menu, DIRMIG takes you to the next menu in the migration path, the User-Logging ID Migration Menu (refer to "Migrating User-Logging Identifiers").
- If you selected only option 4 from the Main Menu, DIRMIG terminates.

HELP (Option 1)

If you select option 1 from the RIN Migration Menu, the HELP facility is invoked. When you exit the HELP facility, you return to the `ENTER RIN OPTION` prompt of the RIN Migration Menu.

List Global RINs (Option 2)

If you select option 2 from the RIN Migration Menu, DIRMIG displays, in MPE V/E SYSDUMP format, all global RINs DIRMIG previously copied to an MPE/iX table structure from the SYSDUMP tape (minus any RINs you may have deleted using option 3 of this menu). When DIRMIG finishes listing global RINs, you return to the `ENTER RIN OPTION` prompt of the RIN Migration menu.

If you are running DIRMIG interactively, DIRMIG enables you to redirect output to an offline listing by prompting you with the following message:

```
OFFLINE LISTING (YES/[NO])?
>>
```

- If you enter Y or YES at the prompt, DIRMIG redirects output to the device associated with the file `LIST`. The default device specification for `LIST` is `LP`, but a file equation can be created prior to running DIRMIG that redirects output to an alternate device (refer to "Redirecting DIRMIG Output").
- If you enter N, NO, or (Return), or if DIRMIG is running in batch mode, output is directed to `$STDLIST`.

Delete Global RINs (Option 3)

If you select option 3 from the RIN Migration Menu, DIRMIG allows you to select the global RINs you do not want migrated to MPE/iX. DIRMIG issues the prompt:

```
ENTER RIN NUMBER(S)>>
```

At this prompt, you can use one of the following formats to specify the RINs you do not want migrated:

- One RIN per line.
- Multiple RINs as well as ranges (using the symbol "/") on the same line separated by commas, for example:

```
ENTER RIN NUMBER(S)>>  2,4,7/11,20,55,70/200
```

When you follow the specified RIN(S) by pressing Return, DIRMIG confirms the deletion of each RIN number before redisplaying the prompt `ENTER RIN NUMBER(S)>>` (thus enabling you to specify more RINs you want deleted). For example:

```
RIN - 2 DELETED
RIN - 4 DELETED
RIN - 7 DELETED
RIN - 8 DELETED
      ⋮
```

When you enter only Return at the `ENTER RIN NUMBER(S)>>` prompt, DIRMIG takes you back to the `ENTER RIN OPTION` prompt of the RIN Migration Menu.

Note

Deleting RINs and compacting the resultant RIN table is time-consuming. For example, compacting a RIN table of 1024 RINs may take up to 30 minutes.

Migrate RINs (Option 4)

When you select option 4 from the RIN Migration Menu, DIRMIG migrates to MPE/iX the global RINs copied

from the SYSDUMP tape (minus any RINs you may have deleted). Once migration is successful, DIRMIG displays the message:

```
RIN MIGRATION SUCCESSFUL
```

Note

DIRMIG overwrites all global RIN data existing on the MPE/iX system with the MPE V/E global RIN data you migrated using DIRMIG's RIN Migration Menu.

DIRMIG's next action is dependent upon which option you previously selected in the Main Menu:

- If you selected option 3 from the Main Menu, DIRMIG takes you to the next menu in the migration path, the User-Logging ID Migration Menu (refer to "Migrating User-Logging Identifiers").
- If you selected only option 4 from the Main Menu, DIRMIG terminates.

Migrating User-Logging Identifiers

DIRMIG displays the User-Logging ID Migration Menu shown in Example G-3 after one of the following actions has occurred:

- You selected option 3 from the Main Menu; this is the second menu DIRMIG displays after you select option 3 from the Main Menu.
- You selected option 5 from the Main Menu.

Selecting Your Menu Option

From the User-Logging ID Migration Menu (shown in Figure G-4), you can choose the user-logging identifier migration option you wish.

```
USER-LOGGING ID MIGRATION
-------------------------
CHOOSE FROM USER-LOGGING OPTIONS BELOW:

     0      - EXIT/ (No Logging ID Migration)
     1      - HELP
     2      - LIST USER-LOGGING IDs
     3      - MIGRATE USER-LOGGING IDs

ENTER USER-LOGGING OPTION
>>
```

Figure G-4. User-Logging ID Menu

DIRMIG prompts you to select one of the above options. If you enter only Return at the ENTER USER-LOGGING OPTION prompt, DIRMIG returns you to the same prompt. The following headings explain the different options available to you through the User-Logging ID Migration Menu.

Exit/No Logging ID Migration (Option 0)

If you select option 0 from the User-Logging ID Migration Menu, DIRMIG does not migrate user-logging identifiers. DIRMIG's next action is dependent upon which option you previously selected in the Main Menu:

- If you selected option 3 from the Main Menu, DIRMIG takes you to the next menu in the migration path, the Directory Migration Menu (refer to "Migrating the Directory").
- If you selected only option 5 from the Main Menu, DIRMIG terminates.

HELP (Option 1)

If you select option 1 from the User-Logging ID Migration Menu, the HELP facility is invoked. When you exit the HELP facility, you return to the `ENTER USER-LOGGING OPTION` prompt of the User-Logging ID Migration Menu.

List User-Logging IDs (Option 2)

If you select option 2 from the User-Logging ID Migration Menu, DIRMIG displays, in MPE V/E `:LISTLOG` format, all user-logging identifiers found on the SYSDUMP tape. When DIRMIG finishes listing user-logging identifiers, you return to the `ENTER USER-LOGGING OPTION` prompt of the User-Logging ID Migration Menu.

If you are running DIRMIG interactively, DIRMIG enables you to redirect output to an offline listing by prompting you with:

```
OFFLINE LISTING (YES/[NO])?
>>
```

- If you enter Y or YES at the prompt, DIRMIG redirects output to the device associated with the file

LIST. The default device specification for LIST is LP, but a file equation can be created prior to running DIRMIG that redirects output to an alternate device (refer to "Redirecting DIRMIG Output").

- If you enter N, NO, or [Return], or if DIRMIG is running in batch mode, output is directed to $STDLIST.

Migrate User-Logging IDs (Option 3)

If you select option 3 from the User-Logging ID Migration Menu, DIRMIG migrates to MPE/iX all user-logging identifiers found on the MPE V/E SYSDUMP tape. Once migration is successful, DIRMIG displays the message:

```
USER-LOGGING ID MIGRATION SUCCESSFUL
```

Note

DIRMIG overwrites all user-logging identifier information existing on the MPE/iX system with the user-logging identifier information from the MPE V/E SYSDUMP tape.

DIRMIG's next action is dependent upon which option you previously selected in the Main Menu:

- If you selected option 3 from the Main Menu, DIRMIG takes you to the next menu in the migration path, the Directory Migration Menu (refer to "Migrating the Directory").
- If you selected only option 5 from the Main Menu, DIRMIG terminates.

Migrating the Directory

DIRMIG displays the Directory Migration Menu shown in Example G-4 after one of the following actions has occurred:

- You selected option 3 from the Main Menu; this is the third menu DIRMIG displays when option 3 is selected.
- You selected option 6 from the Main Menu.

Selecting Your Menu Option

From the Directory Migration Menu (shown in Figure G-5), you can choose the directory migration option you wish.

```
DIRECTORY MIGRATION
-------------------
CHOOSE FROM DIRECTORY OPTIONS BELOW:

     0     - EXIT/(No Directory Migration)
     1     - HELP
     2     - LIST ACCOUNTS
     3     - COMPLETE MIGRATION (All Accounts)
     4     - PARTIAL MIGRATION (Specified Accounts)

ENTER DIRECTORY OPTION
>>
```

Figure G-5. Directory Migration Menu

DIRMIG prompts you to select one of the above options. If you enter only Return at the `ENTER DIRECTORY OPTION` prompt, DIRMIG returns you to the same prompt. The following headings explain the different options available to you through the Directory Migration Menu.

Exit/No Directory Migration (Option 0)

If you select option 0 from the Directory Migration Menu, DIRMIG does not migrate MPE V/E directory structures. DIRMIG's next action is dependent upon which option you previously selected in the Main Menu:

- If you selected option 3 from the Main Menu, DIRMIG takes you to the next menu in the migration path, the Private Volume Migration Menu ("refer to Migrating Private Volume Environments") only if the file `PVASSIST.PUB.SYS` exists. If `PVASSIST.PUB.SYS` does not exist, DIRMIG terminates.
- If you selected only option 6 from the Main Menu, DIRMIG terminates.

HELP (Option 1)

If you select option 1 from the Directory Migration Menu, the HELP facility is invoked. When you exit the HELP facility, you return to the ENTER DIRECTORY OPTION prompt of the Directory Migration Menu.

List Accounts (Option 2)

If you select option 2 from the Directory Migration Menu, DIRMIG displays the names of all accounts located on the SYSDUMP tape. When DIRMIG finishes listing account names, you return to the ENTER DIRECTORY OPTION prompt of the Directory Migration Menu.

If you are running DIRMIG interactively, DIRMIG enables you to redirect output to an offline listing by prompting you with:

```
OFFLINE LISTING (YES/[NO])?
>>
```

- If you enter Y or YES at the prompt, DIRMIG redirects output to the device associated with the file `LIST`. The default device specification for `LIST is LP`, but a file equation can be created prior to running DIRMIG that redirects output to an alternate device (refer to "Redirecting DIRMIG Output").
- If you enter N, NO or Return, or if DIRMIG is running in batch mode, output is directed to `$STDLIST`.

Migration of Accounts (Options 3, 4)

If you select option 3 (Complete Migration of All Accounts) or Option 4 (Partial Migration of Specified Accounts) from the Directory Migration Menu, DIRMIG follows a two-step process for account migration:

1. Selects the accounts to migrate.

2. Migrates the accounts.

The only difference between option 3 and option 4 of the Directory Migration Menu concerns how DIRMIG selects accounts to migrate. The differences between option 3 and option 4 are discussed below.

Complete Migration of all Accounts.. If you select option 3 from the Directory Migration Menu, DIRMIG selects for migration to MPE/iX the entire directory structure found on the MPE V/E SYSDUMP tape. After the selection is made, DIRMIG follows the process described below in "Completing Account Migration".

Partial Migration of Specified Accounts. If you select option 4 from the Directory Migration Menu, DIRMIG allows you to migrate selected MPE V/E account structures. DIRMIG issues the prompt:

```
ENTER ACCOUNT SET(S) TO BE MIGRATED
>>
```

At this prompt you can enter subsets, in the form of "accountsets", of the MPE V/E structure at the account level, followed by Return (you cannot specify group and user subsets). If you enter only Return at this prompt (without specifying any accountsets), DIRMIG returns you to the `ENTER DIRECTORY OPTION` prompt of the Directory Migration Menu.

Accountsets are defined to be one or more MPE V/E account names, where:

- MPE wild cards are allowed; for example, `M@` specifies all accounts beginning with the letter "M".
- Minus subsets are allowed; for example `A@ACCT@` specifies all accounts beginning with the letter "A" except those accounts beginning with the four-letter string "ACCT".
- Accountsets must be separated by commas.
- You can enter as many accountsets as desired up to a maximum of 280 characters.

After you have indicated to DIRMIG the accounts you want migrated, DIRMIG follows the process described below in "Completing Account Migration".

Completing Account Migration. After DIRMIG has selected the directory structure to be migrated (but before directory migration proceeds), DIRMIG allows you to specify whether or not you want DIRMIG to overwrite current MPE/iX directory information with MPE V/E directory information whenever account/group names conflict. DIRMIG displays this prompt:

```
OVERRIDE DISC DIRECTORY INFORMATION
IF DIRECTORY ENTRIES FROM TAPE CONFLICT (YES/[NO])? >>
```

- If you enter Y or YES at the prompt, DIRMIG will overwrite MPE/iX directory information with the already-selected directory information provided on the MPE V/E SYSDUMP tape whenever account/group names conflict (exist both on the MPE/iX system disk and on the SYSDUMP tape). All files associated with the overwritten accounts are lost.

  For example, if the account `XTRA` exists on the MPE/iX system disk and also resides in the MPE V/E directory, the account password, capabilities, and resource accounting limits obtained from the SYSDUMP tape for account `XTRA` are used. Likewise, all user and group structures for account `XTRA` are migrated.

- If you enter N, NO, or Return, DIRMIG does not overwrite MPE/iX directory information when account/group names conflict.

Note

The following MPE/iX reserved groups and accounts are not overwritten even when you enter Y or YES at the above prompt and there is a conflict:

```
MPEiX.SYS
CONFIG.SYS
CONFG950.SYS
CONFG925.SYS DIAG.SYS
```

DIRMIG migrates each account, followed by the groups within the account. When a group associated with a private volume has been migrated, that group's volume set information is saved to PVASSIST.PUB.SYS. After the account's groups are migrated, the account's users are migrated. DIRMIG confirms directory migration with progress messages, for example:

```
DIRECTORY MIGRATION IN PROGRESS

SUCCESSFULLY MIGRATED ACCOUNT account

ACCOUNT account NOT SUCCESSFULLY MIGRATED (DUE TO ERRORS)

ACCOUNT account NOT MIGRATED (CURRENTLY EXISTS ON SYSTEM)

ACCOUNT account NOT MIGRATED (RESERVED BY HP)

DIRECTORY MIGRATION COMPLETED
```

Figure G-6. DIRMIG Progress Messages

If the MPE V/E SYSDUMP tape is MPE version G.03.00 (VMIT) or later, encrypted passwords may exist in the account, group, or user directory entries.

Since the MPE/iX system does not support encrypted passwords, DIRMIG must replace the encrypted passwords it encounters with passwords that are supported. When DIRMIG encounters this situation, it displays this prompt:

```
MPE DIRECTORY MAY CONTAIN ENCRYPTED PASSWORDS.
ENTER PASSWORD TO BE SUBSTITUTED
>>
```

If you supply an MPE/iX supported password (eight or less alphanumeric characters, beginning with an alphabetical character) at this prompt, DIRMIG replaces all encrypted passwords it encounters on the SYSDUMP tape with the password you specified.

If you enter blanks or Return, DIRMIG replaces each encrypted password with the name of the object being migrated. For example, if account `XTRA` contains an encrypted password, then `XTRA` is the replacement password for the `XTRA` account.

After migration has completed, individual users can reset the replaced passwords as necessary.

DIRMIG's next action is dependent upon the following considerations:

- If you selected option 3 from the Main Menu and volume set definitions were appended to the file `PVASSIST.PUB.SYS` when you migrated one or more accounts to MPE/iX , DIRMIG takes you to the next menu in the migration path, the Private Volume Migration Menu (refer to "Migrating Private Volume Environments").
- If you selected option 6 from the main menu, DIRMIG takes you directly to the UDC Environment/User Files Migration Menu (refer to "Migrating the UDC Environment and User Files").

Migrating Private Volume Environments

DIRMIG displays the Private Volume Migration Menu shown in Example G-5 after one of the following actions has occurred:

- You selected option 3 from the Main Menu, and private volume definitions were appended to the file `PVASSIST.PUB.SYS` when you migrated one or more accounts to MPE/iX using DIRMIG's Directory Migration Menu.
- You selected option 7 from the Main Menu, and the file `PVASSIST.PUB.SYS` exists.

The data and environment found on MPE V/E private volume disk packs are not directly transportable to an MPE/iX system. In order to migrate/recreate an MPE V/E private volume environment on your MPE/iX system, it is necessary that you initialize the MPE/iX volumes using VOLUTIL, MPE/iX's volume management utility. The main purpose of DIRMIG's Private Volume Migration Menu is to assist the private volume migration process by generating command files for VOLUTIL. These files contain the appropriate commands which, when supplied to VOLUTIL, initialize

MPE/iX volume media with MPE V/E directory information.

Note

Read Overview of "Private Volume Migration" before using DIRMIG's Private Volume Migration Menu to generate VOLUTIL command files.

For more information concerning VOLUTIL and volume management on MPE/iX, refer to the *Volume Management Reference Manual* (32650-90045).

Overview of Private Volume Migration

The Private Volume Migration Menu serves the purpose of generating command files you can use later with VOLUTIL, the MPE iX volume management utility. These command files enable you to recreate an MPE V/E private volume environment on your MPE/iX system. Private volume migration is a two-step process:

1. Using DIRMIG's Private Volume Migration Menu to generate VOLUTIL command files.
2. Using VOLUTIL to initialize MPE/iX volume media with MPE V/E directory information. Refer to Using VOLUTIL to "Complete Private Volume Migration" for details on this step of the private volume migration process.

Using DIRMIG's Private Volume Migration Menu, you can create VOLUTIL command files. These command files contain VOLUTIL commands that are necessary to recreate the MPE V/E private volume environment on MPE/iX. If desired, the command files can contain MPE/iX commands that will build groups and accounts on the MPE/iX volumes. The command files are built based on the MPE V/E volume set information archived in the MPE/iX file `PVASSIST.PUB.SYS`. (`PVASSIST` is a binary file whose format is understood only by DIRMIG.)

DIRMIG archives MPE V/E volume set information into `PVASSIST.PUB.SYS` from every Dirmig-migrated account containing volume set information. As additional accounts are migrated to MPE/iX, volume set information is appended to `PVASSIST.PUB.SYS`. By providing a central location of MPE V/E volume set information, you have the flexibility of migrating private volume information to MPE/iX at your leisure.

You can restore the files associated with a private volume after you have successfully recreated that private volume's directory structure on MPE/iX.

Two Types of Command Files. The Private Volume Migration Menu provides options to build two types of command files. The first type of file is a general command file that will migrate the entire MPE V/E private volume environment to MPE/iX. The name of this file is `PVSUMARY.PUB.SYS`. Before it can be used by VOLUTIL, however, you must customize it by providing specific logical device numbers. The second type of command file is called the volume set command file. Here, each command file will migrate one specific MPE V/E volume set to MPE/iX. The following points contrast the two types of files:

- For each appropriate VOLUTIL command in `PVSUMARY.PUB.SYS`, logical device numbers are denoted by unique place holders; in volume set command files, the logical device numbers are included with the VOLUTIL commands.

- Volume set command files contain a subset of information from `PVSUMARY.PUB.SYS`; specifically the commands to migrate one volume set to MPE/iX.
- Volume set command files may be used with VOLUTIL as soon as they are created, while `PVSUMARY.PUB.SYS` must be customized first.

Once created, either file type can be customized (edited) and/or used at any time. Refer to Using VOLUTIL to "Complete Private Volume Migration" for a description of customizing DIRMIG command files and using these files with VOLUTIL to complete private volume migration.

Note

Use volume set command files if the number of disk drives available on your MPE/iX-based system is less than the number of private volumes to be migrated. This facilitates incremental volume set creation.

New MPE/iX Volume Management Feature. A feature of MPE/iX volume management that sets it apart from its MPE V/E counterpart is that only the master of a volume set need be mounted to logically initialize several member volumes. Logically initializing member volumes means that the member volume information is written to the master volume, but the member volumes need not be mounted. When additional disk drives are configured, member volumes may be physically initialized with VOLUTIL.

For example, suppose an MPE/iX-based system has only two disk drives configured for volumes and a particular volume set consists of one master and more than one member. As long as the master volume remains mounted, each member may be consecutively mounted, physically initialized, and dismounted in the other disk drive. No consecutive mounting and dismounting is necessary if the number of available disks/disk

drives equal the number of members plus the master. Understanding this concept will facilitate both volume set command file generation and the customization of PVSUMARY.PUB.SYS. Command file customization is detailed in Using VOLUTIL to "Complete Private Volume Migration".

Selecting Your Menu Option

From the Private Volume Migration Menu shown in Figure G-7, you can choose the Private Volume migration option you wish.

```
PRIVATE VOLUME MIGRATION
------------------------
CHOOSE FROM PRIVATE VOLUME OPTIONS BELOW:

    0     - EXIT/(No Volume Set Migration)
    1     - HELP
    2     - LIST VOLUME SETS
    3     - VISIT VOLUME SETS/ (Specified Accounts)
    4     - GENERATE PVSUMARY COMMAND FILE/ (Specified Accounts)

ENTER PRIVATE VOLUME OPTION
>>
```

Figure G-7. Private Volume Migration Menu

DIRMIG prompts you to select one of the above options. Entering only Return at the ENTER PRIVATE VOLUME OPTION prompt returns you to the same prompt. The following headings explain the different options available to you through the Private Volume Migration Menu.

Exit/No Volume Set Migration (Option 0)

If you select option 0 from the Private Volume Migration Menu, DIRMIG does not generate VOLUTIL command files. DIRMIG's next action is dependent upon which option you previously selected in the Main Menu:

- If you selected option 3 from the Main Menu and you successfully migrated one or more accounts using DIRMIG's Directory Migration Menu, DIRMIG takes you to the UDC Environment/User Files Migration Menu (refer to "Migrating the UDC Environment and User Files").
- If you selected only option 7 from the Main Menu, DIRMIG terminates.

HELP (Option 1)

If you select option 1 from the Private Volume Migration Menu, the HELP facility is invoked. When you exit the HELP facility, you return to the `ENTER PRIVATE VOLUME OPTION` prompt of the Private Volume Migration Menu.

List Volume Sets (Option 2)

If you select option 2 from the Private Volume Migration Menu, DIRMIG displays all volume set information contained in `PVASSIST.PUB.SYS`. When DIRMIG finishes listing this information, you return to the `ENTER PRIVATE VOLUME OPTION` of the Private Volume Migration Menu.

If you are running DIRMIG interactively, DIRMIG enables you to redirect output to an offline listing by prompting you with the following message:

```
OFFLINE LISTING (YES/[NO])?
>>
```

- If you enter Y or YES at the prompt, DIRMIG redirects output to the device associated with the file `LIST`. The default device specification for `LIST` is `LP`,

but a file equation can be created prior to running DIRMIG that redirects output to an alternate device (refer to "Redirecting DIRMIG Output").

- If you enter N, NO or Return or if DIRMIG is running in batch mode, output is directed to `$STDLIST`.

If you are running DIRMIG interactively, DIRMIG allows you to specify the output format you want by prompting you with the following message:

```
LONG FORMAT (YES/[NO])?
>>
```

- If you enter Y or YES at the prompt, or if DIRMIG is run in batch mode, DIRMIG generates an output listing similar in format to the MPE V/E command :

  `LISTVS @.@.@,1`

- If you enter N or NO, or Return, DIRMIG generates an output listing similar in format to the MPE V/E command:

  `LISTVS @.@.@,0`

Visit Volume Sets (Option 3)

If you select option 3 from the Private Volume Migration Menu, DIRMIG allows you to generate a command file for each specific MPE V/E volume set you want migrated to MPE/iX. Before generating command files, DIRMIG allows you to specify the accounts associated with the volume sets you want migrated. DIRMIG issues the prompt:

```
ENTER ACCOUNT SUBSETS
>>
```

At this prompt, you can enter subsets, if the form of "accountsets", of the MPE V/E structures at the account level followed by Return (you cannot. specify group and user subsets). If you enter only Return at this

prompt (without specifying any accountsets) DIRMIG returns you to the `ENTER PRIVATE VOLUME OPTION` prompt of the Private Volume Migration Menu.

Accountsets are defined to be one or more MPE V/E account names, where:

- MPE wild cards are allowed; for example, `M@` specifies all accounts beginning with the letter "M".
- Minus subsets are allowed for example, `A@-ACCT@` specifies all accounts beginning with the letter "A" except for those accounts beginning with the four-letter string "ACCT".
- Accountsets must be separated by commas.
- You can enter as many accountsets as desired up to a maximum of 280 characters.

You are repeatedly prompted for accountsets until you enter only Return at the `ENTER ACCOUNT SUBSETS` prompt.

Once DIRMIG completes account selection, DIRMIG scans `PVASSIST.PUB.SYS` and locates all volume set definitions associated with the accounts you specified. For each volume set encountered, DIRMIG then displays the Volume Set Options Menu shown in Example G-6. DIRMIG will display this menu for each volume set definition located.

Selecting a Volume Set Migration Option. If you selected option 3 from the Private Volume Migration Menu and specified account name(s) for volume set migration, DIRMIG will find all volume sets within the account(s) entered. DIRMIG displays the Volume Set Options Menu (shown in Figure G-8) for each volume set encountered, allowing you to generate a VOLUTIL command file for each MPE V/E volume set you want migrated to an MPE/iX system.

```
CHOOSE VOLUME SET OPTION BELOW:

       0  - EXIT (No Volume Set Migration)
       1  - HELP
       2  - LIST VOLUME SET DEFINITION
       3  - GENERATE A COMMAND FILE
       4  - DELETE VOLUME SET DEFINITION
       5  - NEXT VOLUME SET DEFINITION

 ENTER VOLUME SET OPTION
 >>
```

Figure G-8. Volume Set Migration Menu

DIRMIG prompts you to select one of the above options. Entering only Return at the `ENTER VOLUME SET OPTION` prompt returns you to the same prompt. The following headings explain the different options available to you through the Volume Set Options Menu.

Exit/No Volume Set Migration (Option 0). If you select option 0 from the Volume Set Options Menu, DIRMIG does not generate a VOLUTIL command file for the selected volume set definition. Instead, DIRMIG returns you to the `ENTER PRIVATE VOLUME OPTION` prompt of the Private Volume Migration Menu.

HELP (Option 1). If you select option 1 from the Volume Set Options Menu, the HELP facility is invoked. When you exit the HELP facility, you return to the `ENTER VOLUME SET OPTION` prompt of the Volume Set Options Menu.

List Volume Set Definition (Option 2). If you select option 2 from the Volume Set Options Menu, DIRMIG displays to `$STDLIST` information about the currently selected volume set definition, including:

- The volume set name.
- The names of volume set's member volumes.
- The names of the volume set's member volume classes.

Note

Volume member type information is displayed for consistency with the MPE V/E `:NEWVSET` syntax. This information is not required for creating volume sets in MPE/iX.

Generate a Command File (Option 3). If you select option 3 from the Volume Set Options Menu, DIRMIG prompts you for a a command file name, where you can enter any legal MPE file designator (optionally qualified by group and account):

```
ENTER COMMAND FILE NAME
```

```
>>
```

DIRMIG creates a command file for the currently selected volume set definition.

DIRMIG prompts for logical device numbers in order to customize the command file for VOLUTIL use:

```
ENTER LDEV NUMBER FOR INITIALIZING MASTER volume name
>>

ENTER LDEV NUMBER FOR INITIALIZING MEMBER volume name>
>>

ENTER LDEV NUMBER FOR INITIALIZING MEMBER volume name
>>
```

The following rules apply for these prompts:

- The logical device number of zero for the master volume is not allowed. A command file will not be built if a zero is input for the master volume logical device number.
- A logical device number of zero or Return for member volumes. DIRMIG generates commands which will result in the logical initialization of those volumes when the command file is used with VOLUTIL.
- A logical device number of 1 ... 255 for member volumes when the command file is used with VOLUTIL.
- A logical device number may be repeated for member volumes. (The logical device number assigned to the master volume may not be repeated.)

After prompting for logical device information, DIRMIG prompts for spanning information of all home *group.accounts* matching the current volume set name. For each group and account found DIRMIG issues the prompt:

```
SPAN group.acct TO VOLUME SET volume set name
(YES/[NO])?
>>
```

Affirmative responses place MPE/iX commands in the command file to create groups and accounts on the volume set. The spanning commands are added to the command file.

Note

MPE/iX does not support spanning of volume classes. DIRMIG will not prompt for spanning information of home volume classes found within the current volume set name. All groups and accounts found in PVASSIST.PUB.SYS with home volume class information will be logged.

Delete Volume Set Definition (Option 4). If you select option 4 from the Volume Set Options Menu, DIRMIG deletes from `PVASSIST.PUB.SYS` the currently selected volume set definition. Specify this option when you do not want to migrate the current volume set definition.

Next Volume Set Definition (Option 5). If you select option 5 from the Volume Set Options Menu, DIRMIG displays the next volume set in the pool of available migratable volume sets (as bounded by the accounts you selected). DIRMIG displays the Volume Set Options Menu for the newly selected volume set definition. A command file may now be created for the currently selected volume set.

DIRMIG displays a Volume Set Options Menu for each volume set definition located, allowing you to choose your own migration option for each volume set definition. When no more volume set definitions are found in `PVASSIST.PUB.SYS`, DIRMIG returns you to the `ENTER PRIVATE VOLUME OPTION` prompt of the Private Volume Migration Menu.

Generate PVSUMARY Command File (Option 4)

If you select option 4 from the Private Volume Migration Menu, DIRMIG generates the command file `PVSUMARY.PUB.SYS` for MPE V/E private volume environments associated with selected accounts (you may specify volume sets at the account level only). DIRMIG issues the prompt:

```
ENTER ACCOUNT SUBSETS
>>
```

At this prompt, you can enter subsets, in the form of "accountsets", of the MPE V/E structures at the account level followed by Return (you cannot specify group and user subsets). If you enter only Return at this prompt (without specifying any accountsets) DIRMIG returns you to the Enter Private Volume Option of the Private Volume Migration Menu.

Accountsets are defined to be one or more MPE V/E account names, where:

- MPE wild cards are allowed; for example, `M@` specifies all accounts beginning with the letter "M".
- Minus subsets are allowed for example, `A@-ACCT@` specifies all accounts beginning with the letter "A" except those accounts beginning with the four-letter string "ACCT".
- Accountsets must be separated by commas.

- You can enter as many accountsets as desired up to a maximum of 280 characters.

You are repeatedly prompted for accountsets until you enter only Return at the `ENTER ACCOUNT SUBSETS` prompt.

Once DIRMIG completes account selection, DIRMIG proceeds to generate `PVSUMARY.PUB.SYS`. If `PVSUMARY.PUB.SYS` already exists, you are prompted for a new command file name. DIRMIG displays the following progress messages:

```
PVSUMARY FILE CREATION IN PROGRESS...

PVSUMARY FILE COMPLETED.
```

Note

The VOLUTIL command file `PVSUMARY.PUB.SYS` contains all the commands necessary to recreate the private volume environments for the volume sets associated with the specified accounts. However, this file must be customized in order to make it usable by VOLUTIL. Refer to "Using VOLUTIL to Complete Private Volume Migration" for details on customizing `PVSUMARY.PUB.SYS` and using it with VOLUTIL.

DIRMIG's next action is dependent upon which option you previously selected in the Main Menu:

- If you selected option 3 from the Main Menu and you have successfully migrated one or more accounts using DIRMIG's Directory Migration Menu, DIRMIG takes you to the UDC Environment/User Files Migration Menu.
- If you selected only option 7 from the Main Menu, DIRMIG terminates.

Migrating the UDC Environment and User Files

DIRMIG displays the UDC Environment/User Files Migration menu shown in Example G-7 only after one of the following actions has occurred:

- You selected option 3 from the Main Menu and you have successfully migrated one or more accounts from the Directory Migration Menu.
- You selected option 6 from the Main Menu and you have successfully migrated one or more accounts from the Directory Migration Menu.

Selecting Your Menu Option

From the UDC Environment/User Files Migration Menu (shown in Figure G-9), you can choose the UDC and user files migration option you wish.

```
UDC ENVIRONMENT/USER-FILES MIGRATION
------------------------------------
CHOOSE FROM UDC/RESTORE OPTIONS BELOW;

    0   -  EXIT/No UDC Environment/User-Files Migration)
    1   -  HELP
    2   -  MIGRATE UDC ENVIRONMENT ONLY
    3   -  MIGRATE UDC ENVIRONMENT/RESTORE USER FILE SUBSETS
    4   -  MIGRATE UDC ENVIRONMENT/RESTORE ALL FILES

ENTER UDC/RESTORE OPTION
>>
```

Figure G-9. Environment Migration Menu

DIRMIG prompts you to select one of the above options. If you enter only Return at the `ENTER UDC/RESTORE OPTION` prompt, DIRMIG returns you to the same prompt. The following headings explain

the different options available to you through the UDC Environment/User Files Migration Menu.

Note

Options 2,3, and 4 of the UDC Environment/User Files Migration Menu follow the same sequence of five steps to accomplish UDC environment migration. The only difference between these three options lies in the choices DIRMIG gives you concerning restoration of user files associated with the accounts you migrated from the MPE V/E SYSDUMP tape (using DIRMIG's Directory Migration Menu). For this reason, options 2, 3, and 4 are discussed together under the common heading "Migrate UDC Environment/Restore User Files".

Exit/No UDC Environment/User Files Migration (Option 0)

If you select option 0 from the UDC Environment/User Files Migration Menu, DIRMIG does not migrate UDC environments and user files. DIRMIG terminates.

HELP (Option 1)

If you select option 1 from the UDC Environment/User Files Migration Menu, the HELP facility is invoked. When you exit the HELP facility, you return to the `ENTER UDC/RESTORE OPTION` prompt of the UDC Environment/User Files Migration Menu.

Migrate UDC Environment/Restore User Files (Options 2, 3, 4)

If you select option 2, 3, or 4 from the UDC Environment/User Files Migration Menu, DIRMIG proceeds to migrate the UDC environments associated with the account structures you successfully migrated during directory migration (refer to "Migrating the Directory Structure").

Note

DIRMIG invokes the RESTORE facility twice during UDC migration. If you used the MPE V/E GETUDC utility to place UDC environment files at the beginning of the RESTORE portion of the SYSDUMP tape, DIRMIG takes substantially less time to restore UDC environment files. (Refer to Appendix F, "Using the GETUDC Utility."

In addition to migrating the UDC environment, DIRMIG gives you the opportunity to:

- Restore selected file subsets (if you select option 3).
- Restore all file subsets (if you select option 4).

  If you are running DIRMIG interactively, DIRMIG allows you to redirect output to an offline listing by prompting you with:

```
OFFLINE LISTING FOR FILES TO BE RESTORED (YES/[NO])?
>>
```

- If you enter Y or YES at the prompt, DIRMIG redirects output to the device associated with the file `LIST`. The default device specification for `LIST` is `LP`, but a file equation can be created prior to running DIRMIG that redirects output to an alternate device (refer to "Redirecting DIRMIG Output").

- If you enter N, NO, or Return, or if DIRMIG is running in batch mode, output is directed to `$STDLIST`.

UDC migration is divided into five steps, described below.

1. **Obtain MPE/iX UDC environment information.** If COMMAND.PUB.SYS exists on the MPE/iX system, DIRMIG performs these actions on COMMAND.PUB.SYS to obtain MPE/iX UDC information and to preserve the current MPE/iX UDC environment:

   - Copies UDC information from COMMAND.PUB.SYS to a temporary data structure.
   - Renames COMMAND.PUB.SYS to SAVECMD.PUB.SYS. DIRMIG later modifies SAVECMD.PUB.SYS when migrating the MPE V/E UDC environment to MPE/iX.
   - Copies the contents of COMMAND.PUB.SYS to the file COPYCMD.PUB.SYS. DIRMIG can return the original version of COMMAND.PUB.SYS if an error occurs during UDC environment migration.

   If the MPE/iX COMMAND.PUB.SYS has an EOF equal to zero, the file is purged and DIRMIG continues with the assumption that there is no UDC environment on the MPE/iX system.

2. **Obtain MPE V/E UDC environment from the SYSDUMP tape.** DIRMIG prompts you to mount the SYSDUMP tape in order to restore the MPE V/E file COMMAND.PUB.SYS:

```
MOUNT MPE SYSDUMP TAPE AND PLACE TAPE DRIVE ON-LINE
```

   After the SYSDUMP tape is mounted on the tape device and placed online, DIRMIG displays the following progress messages to indicate success of file restoration:

```
RESTORING  COMMAND.PUB.SYS
RESTORE OF COMMAND.PUB.SYS SUCCESSFUL
```

After the MPE V/E system file `COMMAND.PUB.SYS` is successfully restored on the MPE/iX system, DIRMIG collects the names of the UDC files for those accounts and users successfully migrated. These are the UDC files that are restored during Step 3 (see below). Once these names are collected, DIRMIG purges the `COMMAND.PUB.SYS` file restored from the SYSDUMP tape and renames the MPE/iX file `SAVECMD.PUB.SYS` (see Step 1) back to `COMMAND.PUB.SYS`.

3. **Restore UDC files/user files from the SYSDUMP tape.** Step 3 varies depending upon whether you selected option 2, option 3, or option 4 from the UDC Environment/User Files Migration Menu. In all three cases, DIRMIG restores those UDC files whose names were collected in Step 2. DIRMIG restores files by invoking the `:RESTORE` facility with the `KEEP` and `OLDDATE` options enabled. (Refer to the `:RESTORE` command in the *MPE/iX Commands Reference Manual* (32650-90003) for details.)

Note

DIRMIG assumes disk file space limits for group and account entries will be sufficient. DIRMIG lists files that are not restored because of insufficient disk space.

- **Migrate UDC Environment Only (Option 2).** DIRMIG proceeds to restore from the MPE V/E SYSDUMP tape the UDC files whose names were collected in Step 2. DIRMIG does not restore user files from the SYSDUMP tape. DIRMIG prompts you to mount the SYSDUMP tape:

```
MOUNT MPE SYSDUMP TAPE AND PLACE TAPE DRIVE ON-LINE
```

  After the SYSDUMP tape is mounted and placed online, DIRMIG displays the following messages to indicate progress and success of UDC file restoration:

```
RESTORING  UDC FILES
RESTORE OF UDC FILES SUCCESSFUL
```

- **Migrate UDC Environment/Restore User File Subsets (Option 3).** DIRMIG displays the following prompt:

```
ENTER FILE SUBSETS
>>
```

At this prompt you can enter file subsets you want to migrate. File subsets are defined to be one or more MPE V/E file names (valid to the MPE/iX `:RESTORE` command), where file subsets are separated by commas, and MPE wild cards are allowed; for example, `M@#.@.@` specifies all files beginning with the letter "M" and ending with a numeric character.

You are repeatedly prompted for filesets until you enter only Return at the `ENTER FILE SUBSETS` prompt. DIRMIG prompts you to mount the SYSDUMP tape:

```
MOUNT MPE SYSDUMP TAPE AND PLACE TAPE DRIVE ON-LINE
```

After the SYSDUMP tape is mounted and placed online, DIRMIG displays the following messages to indicate progress and success of UDC file restoration:

```
RESTORING UDC FILES/USER-DESIGNATED FILES
RESTORE OF UDC FILES/USER-DESIGNATED FILES SUCCESSFUL
```

- **Migrate UDC Environment/Restore all Files (Option 4).** DIRMIG proceeds to restore from the MPE V/E SYSDUMP tape the UDC files whose names were collected in Step 2, as well as all user files associated with the accounts migrated during directory migration (except those files in `PUB.SYS`). DIRMIG prompts you to mount the SYSDUMP tape:

```
MOUNT MPE SYSDUMP TAPE AND PLACE TAPE DRIVE ON-LINE
```

  After the SYSDUMP tape is mounted and placed online, DIRMIG displays the following messages to indicate progress and success of UDC file restoration:

```
RESTORING  UDC FILES/ALL USER-FILES
RESTORE OF UDC FILES/ALL USER-FILES SUCCESSFUL
```

4. **Modify the MPE/iX UDC Environment.** DIRMIG modifies the MPE XL `COMMAND.PUB.SYS` file to include UDC environment information migrated from the MPE V/E system.

Note If an MPE/iX user also exists on the MPE V/E SYSDUMP tape, that user's MPE/iX UDC environment is replaced by the MPE V/E UDC environment found on the MPE V/E SYSDUMP tape.

---

5. **Update the MPE/iX Directory.** User and account entries within the MPE/ iX directory are updated to reflect the information found in the new MPE/iX system file `COMMAND.PUB.SYS`. This action enables the MPE/iX UDC environment without the need to use the `:SETCATALOG` command.

   Once DIRMIG has completed updating the MPE/iX directory, DIRMIG terminates.

Exiting DIRMIG

Each menu in DIRMIG has an optional selection that allows you to exit DIRMIG normally.

Caution

You should not interrupt or abnormally terminate DIRMIG once it is started, since it is running in Privilege Mode updating MPE/iX system tables and allocating directory space. Use exit options provided to exit DIRMIG safely.

Redirecting DIRMIG Output

When you select the offline option available through several DIRMIG menus, you direct the text output to the DIRMIG list file `LIST`. By default this file is assigned to the system line printer (`DEV=LP`). If you want to redirect the output to a different device, you must, prior to running DIRMIG, create an MPE file equation to redefine the characteristics for `LIST`. This file equation is invoked only when you respond Y or YES to an `OFFLINE LISTING (YES)/[NO])` prompt. For example, the file equation below redirects output to device name `EPOC`:

```
:FILE OTHER;DEV=EPOC
:FILE LIST=*OTHER
```

Note

When DIRMIG is run in batch mode, the output always defaults to `:$STDLIST`; the detailed format is to be used where applicable.

Examining DIRMIG Log Files

`DIRLOG`*nn*`.PUB.SYS`, DIRMIG's log file, is created to track the progress of each component's migration. You should always examine `DIRLOG`*nn* after running DIRMIG. This file provides information for debugging and troubleshooting if a problem occurs. The files are numbered (maximum number 99) so you can maintain a log file history of every run of DIRMIG.

DIRMIG searches only for the maximum sequence numbered DIRLOG file before creating a new log file. For example, if DIRMIG finds that the log file `DIRLOG01.PUB.SYS` already exists, DIRMIG creates `DIRLOG02.PUB.SYS`.

If `DIRLOG99.PUB.SYS` already exists, DIRMIG disables the logging facility and asks you if you want to continue execution with logging disabled:

- If your enter Y or YES to the prompt, DIRMIG continues execution with logging disabled.

- If you enter N, NO, to the prompt, DIRMIG terminates. You can purge or rename DIRMIG log files (highest to lowest), then rerun DIRMIG.

`DIRLOG`*nn* contains:

- Progress messages displayed to `$STDLIST`.
- Progress messages for directory migration.
- Error messages indicating a component has not been migrated.

Verifying Directory Migration

To verify that the correct data has been migrated check the system components, listed in the first column, by using the commands listed in the second column.

To check:	Use:
RINs	`:SYSGEN` (MISC Configurator)
User-Logging IDs	`:LISTLOG`
Directory	`:REPORT :LISTACCT :LISTGROUP :LISTUSER`
Mountable Volumes	`:DSTAT :VOLUTIL`
UDCs	`:SHOWCATALOG`
Files	`:LISTFILE` or `:LISTF`

Using VOLUTIL to Complete Private Volume Migration

Discussed below is the method of migrating MPE V/E private volume environments using VOLUTIL command files built by DIRMIG, and the MPE/iX volume management utility VOLUTIL.

The discussion below is divided into the following sections:

- Customizing `PVSUMARY.PUB.SYS`
- Performing the Migration
- Demonstrating a Volume Set Migration

Read and understand "Customizing `PVSUMARY.PUB.SYS`" before attempting to complete private volume migration. It is assumed that the reader has already used the Private Volumes Migration Menu of DIRMIG to create VOLUTIL command files.

Note

For more information concerning the VOLUTIL utility and volume management on MPE/iX, refer to *Volume Management (32650-90045)*.

Customizing PVSUMARY.PUB.SYS

The command file `PVSUMARY.PUB.SYS` contains VOLUTIL commands to generate the MPE V/E private volume environment on MPE/iX. However, alphanumeric tokens are used in lieu of logical device numbers for VOLUTIL command parameters. You are responsible for editing `PVSUMARY.PUB.SYS` and substituting logical device numbers for the place holders in every VOLUTIL command.

`PVSUMARY.PUB.SYS` is organized by volume sets. For every volume set contained in `PVSUMARY.PUB.SYS`, the pattern of VOLUTIL commands remains the same, as in the following example.

```
     .
     .
     .
SCRATCHVOL XX1
NEWSET volume set name VSET1 XX1
SETDEFAULTSET volume set name
     .
     .
     .
SCRATCHVOL XX2
NEWVOL XX2
     .
     .
     .
SCRATCHVOL XXn
NEWVOL XXn
     .
     .
     .
```

Using an EDIT/V compatible editor, edit `PVSUMARY` keeping the following rules in mind:

- You must replace token `XX1` with the logical device number of the disk drive upon which the master volume is to be mounted. This value can not be zero.
- You must replace token `XX2` through `XX`*n* with the logical device numbers of disk drives upon which member volumes are to be mounted. Logical device numbers in the range 1 ... 255 are allowed. The logical device number associated with a member volume cannot be the logical device number associated with the master volume. More than one member

volume can be associated with the same logical device number.

- Tokens `XX2` through `XX`*n* can be zero only if the those volumes will be logically initialized on the master volume. You can physically initialize these volumes with VOLUTIL's command, `:INITVOL`, at a later time.
- Any VOLUTIL command or MPE/iX command can be modified, added, or deleted at your discretion. However, this is only recommended for those users who are intimately knowledgeable with VOLUTIL and MPE/iX's mountable volumes.
- All of the above changes should be made for each volume set definition in `PVSUMARY.PUB.SYS`.

The preceding comments identify which place holders should be replaced by logical device numbers within a particular volume set. Once you make these replacements, `PVSUMARY.PUB.SYS` is ready for use by VOLUTIL.

Performing the Migration

Once the VOLUTIL command files are ready (customized), the MPE V/E private volume environment can be migrated. The process is broken into three main steps:

- Prepare to Migrate Private Volumes
- Migrate Private Volumes with VOLUTIL
- Verify the Migration

Read "Performing the Migration" entirely before proceeding. Once the process is well understood, follow the steps described in "Prepare to Migrate Private Volumes" and continue through "Verify the Migration". These steps may be repeated as necessary.

Prepare to Migrate Private Volumes

During the migration, it is recommended that you perform the migration on the physical Console. If this is not possible, perform the migration at a logical Console where the physical Console is in sight, as volume mount and dismount messages are posted to the physical Console (even if the logical Console has been relocated).

Below are the necessary steps to take to prepare for private volume migration:

1. Enter the command `:SHOWVAR HPTIMEOUT`

   If the `HPTIMEOUT` value is less than 15 minutes but greater than 0, issue the MPE/iX command, `:SETVAR HPTIMEOUT 15`. This will set the Command Interpreter read timeout value to 15 minutes. If the `:SETVAR` fails, use SYSGEN to configure the maximum

Command Interpreter read timeout value to at least 15. Reboot the system. A value of 0 indicates that there is no read timeout in effect and no `HPTIMEOUT` changes are necessary. The 15-minute timeout is the approximate time to place HP 7933/35 disk drives online and offline. The command files built by DIRMIG use the 15-minute timer to wait for the Operator to place the volumes online and offline.

2. Identify those disk packs that will be used for the migration.

   The volumes should be new or unused. Don't use a volume if it is suspected that it is used for something else (for example, a system pack).

3. Mount the available volume media into the disk drives.

   Messages of the following type are posted to the Console:

   ```
   LONER VOLUME MOUNTED ON LDEV 4.  (AVR 11)
   ```

   or,

   ```
   UNKNOWN VOLUME MOUNTED ON LDEV 14.  (AVR 9)
   ```

Note

Do not proceed to the next step until the system posts AVR messages to the Console for each volume media mounted.

4. Enter the command `:DSTAT ALL`

   The status for each of the newly mounted volumes should be one of the following: `UNKNOWN`, `SCRATCH`, or `LONER`. If the status is `MASTER` or `MEMBER`, enter the MPE/iX command `:VSCLOSE` for each unique volume set definition associated with the logical device. `:VSCLOSE` closes the volume sets, and the volumes' status will become `LONER`. The DIRMIG command files will fail if the status is not `UNKNOWN, LONER,` or `SCRATCH`.

5. Verify the command files to be used by VOLUTIL. Ensure that the following is correct, and modify the file if necessary:

   - The member and class information
   - The logical device numbers
   - The accounting structure that will be built

Migrate Private Volumes with VOLUTIL

Below are the necessary steps you take to migrate private volumes with the MPE iX VOLUTIL utility:

Enter the command `:RUN VOLUTIL.PUB.SYS`

VOLUTIL drives the private volume migration. The VOLUTIL utility is command driven and provides a HELP facility. You may enter commands by hand to configure volume sets, however, the command files built by DIRMIG may be input to VOLUTIL to migrate specific volume set environments.

Note

Some of the actions described below are timed by the CI variable `HPTIMEOUT` to allow dismounting and mounting of volume media. Follow each directive in a timely fashion. If the media is not mounted by the time VOLUTIL continues, VOLUTIL will wait until the appropriate volumes are mounted.

Below is the procedure for using the command files. Read all the :COMMENT messages posted to the Console. The input is entered in response to VOLUTIL prompts:

1. USE *cmdfile*

   Enter the name of the VOLUTIL script file, *cmdfile*, via the VOLUTIL USE command. *cmdfile* may either be the user-modified copy of PVSUMARY.PUB.SYS, or one of the individual command files for a specific volume set.

2. Enter Y or YES in response to any SCRATCHVOL prompts of the following:

   ```
   SCRATCH VOLUME ON LDEV  (Y/N)?
   ```

   An error may occur if the volume is not a LONER volume, however the error will not effect the subsequent NEWSET or NEWVOL commands from executing. The SCRATCHVOL command is included since any type of volume may be mounted.

3. Enter Y or YES in response to any CONTINUE OPERATION message of the following:

```
VOLUME ON LDEV n IS A LONER VOLUME.  CONTINUE WITH
TRANSACTION (Y/N)?

CONTINUE WITH DISC INITIALIZATION ON LDEV  (Y/N)?
```

4. Press Return in response to an INPUT command.

If the user is requested to dismount and subsequently mount another volume, the user should do so in a timely fashion and press Return , ONLY after an AVR message is posted to the Console after a volume is mounted. If Return is pressed prematurely, then VOLUTIL will wait until the volume is mounted, and then prompt the user for a response. Respond as necessary.

- Continue responding and mounting/dismounting separate disk packs into the specified disk drives as directed by VOLUTIL.
- After the volume set information is migrated to the new volumes, accounts and groups will be built on the volume set.

Verify the Migration

Several commands are used to verify the proper migration:

- Run the VOLUTIL utility by entering the command `:RUN VOLUTIL.PUB.SYS`. Use the VOLUTIL command, `:SHOWSET`. The output should indicate the results of the most currently migrated volume set.
- Use the MPE/iX `:LISTACCT` command with the `ONVS` keyword to verify the span of *acctname* to *vsetname*. Enter:

  `:LISTACCT` *acctname* `;` `ONVS=`*vsetname*

- Use the MPE/iX `:LISTGROUP` command with the `ONVS` keyword to verify the span of groups and accounts to *vsetname*. Enter:

  `:LISTGROUP` *grpname* `[`*acctname*`]` `;` `ONVS=`*vsetname*

At least the master of each volume set must be mounted and on-line before these commands can be used to verify the success of the private volume migration.

Demonstrating a Volume Set Migration

The following sections illustrate the migration of a volume set with a volume set command file created by DIRMIG. The discussion is divided into two parts:

- Creating the Volume Set Command File
- Using a Volume Set Command File with VOLUTIL

Creating the Volume Set Command File

The following screens illustrate a DIRMIG execution to create a volume set command file named `VSET1CMD`. It is assumed that the file `PVASSIST.PUB.SYS` exists from a previous execution of DIRMIG.

```
CTRL A
=LOGOFF #S1222
:RUN DIRMIG.PUB.SYS

============================================================
DIRMIG.PUB.SYS HP30362A.00.00 (c) HEWLETT PACKARD CO.,1986.
MPE/iX Migration Utility. MON, MAR 23, 1987,  4:44 PM
============================================================
Current Log File: DIRLOG01.PUB.SYS.

CHOOSE FROM MIGRATION OPTIONS BELOW:

     0     - EXIT
     1     - HELP
     2     - COMPLETE MIGRATION (No Dialog)
     3     - COMPLETE MIGRATION (With Dialog)
     4     - RINS
     5     - USER LOGGING ID'S
     6     - DIRECTORY
     7     - PRIVATE VOLUME ENVIRONMENT

ENTER MIGRATION OPTION(S)
>> 7

PRIVATE VOLUME MIGRATION
------------------------
CHOOSE FROM PRIVATE VOLUME OPTIONS BELOW:

     0     - EXIT/(No Volume Set Migration)
     1     - HELP
     2     - LIST VOLUME SETS
     3     - VISIT VOLUME SETS/ (Specified Accounts)
     4     - GENERATE PVSUMARY COMMAND FILE/ (Specified Accounts)

ENTER PRIVATE VOLUME OPTION
>>  3
```

```
ENTER ACCOUNT SUBSETS
>> PVACCT1
========= BEGINNING PRIVATE VOLUME MIGRATION ===============

VOLUME SET MIGRATION FOR VSET1.PVGROUP1.PVACCT1.

CHOOSE VOLUME SET OPTION BELOW:

     0     - EXIT/(No Volume Set Migration)
     1     - HELP
     2     - LIST VOLUME SET DEFINITION
     3     - GENERATE A COMMAND FILE
     4     - DELETE VOLUME SET DEFINITION
     5     - NEXT VOLUME SET DEFINITION

ENTER VOLUME SET OPTION
>> 3

ENTER COMMAND FILE NAME
>> VSETICMD

ENTER LOGICAL DEVICE NUMBER FOR INITIALIZING MASTER VSET1
>> 3

ENTER LOGICAL DEVICE NUMBER FOR INITIALIZING MEMBER MEMBER2
>> 4

ENTER LOGICAL DEVICE NUMBER FOR INITIALIZING MEMBER MEMBER3
>> 4

ENTER LOGICAL DEVICE NUMBER FOR INITIALIZING MEMBER MEMBER4
>> 4

SPAN PVGROUP2.PVACCT2 TO VOLUME SET VSET1.PVGROUP1.PVACCT1 (YES/[NO])?
>> YES

SPAN PVGROUP1.PVACCT2 TO VOLUME SET VSET1.PVGROUP1.PVACCT1 (YES/[NO])?
>> YES

GENERATING COMMAND FILE VSET1CMD.PUB.SYS...

COMMAND FILE VSET1CMD.PUB.SYS SAVED.

VOLUME SET MIGRATION FOR VSET1.PVGROUP1.PVACCT1.
```

```
CHOOSE VOLUME SET OPTION BELOW:

     0     - EXIT/(No Volume Set Migration)
     1     - HELP
     2     - LIST VOLUME SET DEFINITION
     3     - GENERATE A COMMAND FILE
     4     - DELETE VOLUME SET DEFINITION
     5     - NEXT VOLUME SET DEFINITION ENTER VOLUME SET OPTION
>>   0
NO MORE VOLUME SETS TO PROCESS.
=========== END OF PRIVATE VOLUME MIGRATION ================

 PRIVATE VOLUME MIGRATION
------------------------
CHOOSE FROM PRIVATE VOLUME OPTIONS BELOW:

     0     - EXIT/(No Volume Set Migration)
     1     - HELP
     2     - LIST VOLUME SETS
     3     - VISIT VOLUME SETS/ (Specified Accounts)
     4     - GENERATE PVSUMARY COMMAND FILE/ (Specified Accounts)

ENTER PRIVATE VOLUME OPTION
>>  0

END OF PROGRAM
:
```

VSET1CMD is now ready for use by VOLUTIL.

Using a Volume Set Command File with VOLUTIL

The following screens demonstrates how to use a volume set command file with VOLUTIL. In the example, two disk drives are configured for the private volume usage.

The migration is performed on the System Console. The mount and dismount messages are shown. The following routine uses the command file `VSET1CMD` created in the previous routine. (`PVSUMARY` could also have been used.)

Note these points in the following routine:

- The user must issue a `:VSCLOSE` on `VSET2.GROUP1.ACCOUNT` on ldev 3 before continuing with VOLUTIL.
- The error issued by `:SCRATCHVOL` on ldev 14 for the `UNKNOWN` volume does not affect the subsequent `:NEWVOL` command.
- The user will dismount and mount two additional volumes on ldev 14 when prompted by VOLUTIL.

```
:SHOWVAR HPTIMEOUT
HPTIMEOUT = 0
:SETVAR HPTIMEOUT 15
:
VSET2.GROUP1.ACCOUNT VOLUME MOUNTED ON LDEV 3 (AVR 7)
UNKNOWN VOLUME MOUNTED ON LDEV 14 (AVR 9)
:DSTAT ALL

  LDEV-TYPE    STATUS      VOLUME (VOLUME SET - GEN)
 ----------  ---------   --------------------------
  1-079350     MASTER      MEMBER1         (MPEiX_SYSTEM_VOLUME_SET)
  2-079350     MEMBER      MEMBER2         (MPEiX_SYSTEM_VOLUME_SET)
  3-079350     MASTER      VSET2           (VSET2.GROUP1.ACCOUNT)
 14-079350     UNKNOWN

:VSCLOSE VSET2.GROUP1.ACCOUNT
:RUN VOLUTIL.PUB.SYS

Volume Utility 8.30.01 (C) Hewlett-Packard Co., 1986
volutil:USE VSET1CMD

volutil: :COMMENT ------------------------------------------------------------
volutil: :COMMENT              MPE/iX Directory Migration Tool
volutil: :COMMENT   Command File : VSET1CMD.PUB.SYS
volutil: :COMMENT   Created On   : MON, MAR 23, 1987,  4:45 PM
volutil: :COMMENT ------------------------------------------------------------
volutil: :COMMENT
volutil: :COMMENT ------------------------------------------------------------
volutil: :COMMENT   VOLUME SET COMMANDS FOR VSET1.PVGROUP1.PVACCT1
volutil: :COMMENT ------------------------------------------------------------
volutil: SCRATCHVOL 3

SCRATCH VOLUME ON LDEV 3 (Y/N)? Y

SCRATCH VOLUME MOUNTED ON LDEV 3 (AVR 8)
volutil: NEWSET VSET1.PVGROUP1.PVACCT1 VSET1 3

UNRECOGNIZABLE MPE/iX DISC VOLUME
CONTINUE WITH DISC INITIALIZATION ON LDEV 3 (Y/N)? Y

volutil: SETDEFAULTSET VSET1.PVGROUP1.PVACCT1
volutil: SCRATCHVOL 14
*ERROR: MOUNTED VOLUME ON LDEV IS OF WRONG TYPE (SCRATCH VOLUME).
volutil: NEWVOL MEMBER2 14
VOLUME ON LDEV 14 IS UNKNOWN VOLUME. CONTINUE WITH TRANSACTION (Y/N)? Y
```

```
UNRECOGNIZABLE MPE/iX DISC VOLUME
CONTINUE WITH DISC INITIALIZATION ON LDEV 14 (Y/N)? Y

volutil: :COMMENT
volutil: :COMMENT ---------------------------------------------------------
volutil: :COMMENT The following sequence of commands enables the user
volutil: :COMMENT to dismount a volume set member in order to initialize
volutil: :COMMENT a new member on the same ldev.
volutil: :COMMENT ---------------------------------------------------------
volutil: :VSCLOSE VSET1.PVGROUP1.PVACCT1
volutil: :COMMENT ********************************************************
volutil: :COMMENT * CURRENT VOLUME ON LDEV 14   MUST BE DISMOUNTED TO      *
volutil: :COMMENT * INITIALIZE VOLUME SET MEMBER MEMBER3  ON LDEV 14  .    *
volutil: :COMMENT * MOUNT VOLUME SET MEMBER TO BE INITIALIZED ON LDEV 14  .*
volutil: :COMMENT ********************************************************
volutil: :INPUT HPWAITVOL,"ENTER <RETURN> WHEN VOLUME IS ON-LINE:";WAIT=900

ENTER <RETURN> WHEN VOLUME IS ON-LINE:
DISMOUNT REQUEST FOR LDEV 14 GRANTED (AVR 12)
LONER VOLUME MOUNTED ON LDEV 14 (AVR 11)

Return

volutil: :VSOPEN VSET1.PVGROUP1.PVACCT1
VSET1.PVGROUP1.PVACCT1 VOLUME MOUNTED ON LDEV 3 (AVR 7)
volutil: SCRATCHVOL 14
SCRATCH VOLUME ON LDEV 14 (Y/N)?  Y

SCRATCH VOLUME MOUNTED ON LDEV 14 (AVR 8)
volutil: NEWVOL MEMBER3 14
UNRECOGNIZABLE MPE/iX DISC VOLUME
CONTINUE WITH DISC INITIALIZATION ON LDEV 14 (Y/N)? Y

volutil: :COMMENT
volutil: :COMMENT ---------------------------------------------------------
volutil: :COMMENT The following sequence of commands enables the user
volutil: :COMMENT to dismount a volume set member in order to initialize
volutil: :COMMENT a new member on the same ldev.
volutil: :COMMENT ---------------------------------------------------------
volutil: :VSCLOSE VSET1.PVGROUP1.PVACCT1
volutil: :COMMENT ********************************************************
volutil: :COMMENT * CURRENT VOLUME ON LDEV 14   MUST BE DISMOUNTED TO      *
volutil: :COMMENT * INITIALIZE VOLUME SET MEMBER MEMBER4  ON LDEV 14  .    *
volutil: :COMMENT * MOUNT VOLUME SET MEMBER TO BE INITIALIZED ON LDEV 14  .*
volutil: :COMMENT ********************************************************
volutil: :INPUT HPWAITVOL,"ENTER <RETURN> WHEN VOLUME IS ON-LINE:";WAIT=900
```

```
ENTER <RETURN> WHEN VOLUME IS ON-LINE:
DISMOUNT REQUEST FOR LDEV 14 GRANTED (AVR 12)
LONER VOLUME MOUNTED ON LDEV 14 (AVR 11)

Return

voluti1: :VSOPEN VSET1.PVGROUP1.PVACCT1
VSET1.PVGROUP1.PVACCT1 VOLUME MOUNTED ON LDEV 3 (AVR 7)
voluti1: SCRATCHVOL 14
SCRATCH VOLUME ON LDEV 14 (Y/N)? Y
SCRATCH VOLUME MOUNTED ON LDEV 14 (AVR 8)
voluti1: NEWVOL MEMBER4 14
UNRECOGNIZABLE MPE/iX DISC VOLUME

CONTINUE WITH DISC INITIALIZATION ON LDEV 14 (Y/N)? Y

voluti1: :COMMENT
voluti1: :COMMENT ------------------------------------------------------------
voluti1: :COMMENT    CLASS INFORMATION FOR VSET1.PVGROUP1.PVACCT1.
voluti1: :COMMENT ------------------------------------------------------------
voluti1: NEWCLASS CLASS1 VOLUMES= (VSET1, MEMBER2, MEMBER3, MEMBER4)
voluti1: NEWCLASS CLASS2 VOLUMES= (VSET1, MEMBER2, MEMBER3, MEMBER4)
voluti1: NEWCLASS CLASS3 VOLUMES= (VSET1, MEMBER2, MEMBER3, MEMBER4)
voluti1: NEWCLASS CLASS4 VOLUMES= (VSET1, MEMBER2, MEMBER3, MEMBER4)
voluti1: NEWCLASS CLASS5 VOLUMES= (VSET1, MEMBER2, MEMBER3, MEMBER4)
voluti1: :COMMENT
voluti1: :COMMENT ------------------------------------------------------------
voluti1: :COMMENT    SPANNING INFORMATION FOR VSET1.PVGROUP1.PVACCT1.
voluti1: :COMMENT ------------------------------------------------------------
voluti1: :NEWACCT PVACCT2,MGR; ONVS= VSET1.PVGROUP1.PVACCT1
voluti1: :NEWGROUP PVGROUP2.PVACCT2;ONVS= VSET1.PVGROUP1.PVACCT1
voluti1: :NEWGROUP PVGROUP1.PVACCT2;ONVS= VSET1.PVGROUP1.PVACCT1
voluti1: EXIT

END OF PROGRAM
:
```

Private Volume Migration Checklist

Below is a checklist of the major steps of private volume migration.

- Migrate private volume accounts to MPE/iX with `DIRMIG.PUB.SYS`.
- Create `PVSUMARY.PUB.SYS` and/or volume set command files with DIRMIG.
- Customize `PVSUMARY.PUB.SYS`.
- Identify all necessary volume media.
- Ensure that `HPTIMEOUT` is configured to at least 15 minutes.
- Verify the VOLUTIL command files for correctness.
- Run `VOLUTIL.PUB.SYS` and use the appropriate command files.
- Label the disk packs as they are generated.
- Verify the migration with VOLUTIL and `:LIST`*xxxx* commands.

DIRMIG Program Error Messages

Two classes of errors can occur during directory migration:

- Corruption identified within the directory being ported
- Operating system errors

Directory Corruption

Directory corruption errors are posted to `DIRLOG`*nn* which contains a directory path displaying the sector of the corrupt data structure.

If any accounts were not successfully migrated, examine `DIRLOG`*nn*`.PUB.SYS` for error messages. Each error message is accompanied by a description of the condition

which caused the error, as well as a description of recommended actions to take for error recovery.

Operating System Errors

Operating system errors generated by the directory service routines will be posted to `DIRLOG`*nn*.

DIRMIG will attempt to provide conclusive error messages if these errors occur.

If errors occur, the following rules apply:

1. If an error occurs during the creation or altering of an account, the users and groups will not be migrated. Also, volume set information will not be saved to `PVASSIST.PUB.SYS`.
2. If an error occurs during the creation or alteration of a group, the remaining groups for that account are not migrated. DIRMIG will attempt to migrate the users, however. Volume set information for that group and all remaining groups will not be saved to `PVASSIST.PUB.SYS`.
3. If an error occurs while DIRMIG is creating a user or altering the user of an account, the remaining users for that account are not created.
4. If the account is created or altered successfully, users are guaranteed to have at least a `PUB` group. If the account was created, the user `MGR` will be on the system. If the account was altered, it is assumed that previous users will remain on the system.

Error Recovery

If any corrupted objects exist on the SYSDUMP tape, corrective action will have to be taken on the MPE V/E system from which the tape was made. Contact your Hewlett-Packard Support Representative for assistance.

Any object not migrated to the MPE/iX system may be created via MPE/iX commands. The following MPE/iX

and MPE V/E commands are useful for creating objects not successfully migrated by DIRMIG:

MPE V/E, MPE/iX	MPE XL ONLY	MPE V/E ONLY
:NEWACCT	:VOLUTIL	:SYSDUMP
:ALTACCT	:SYSGEN	:RELOAD
:NEWGROUP	:INSTALL	
:ALTGROUP		
:NEWUSER		
:ALTUSER		
:PURGEACCT		
:PURGEGROUP		
:PURGEUSER		
:LISTACCT		
:LISTGROUP		
:LISTUSER		
:FREERIN		
:GETRIN		
:RELLOG		
:GETLOG		
:SETCATALOG		

- If RESTORE of the MPE V/E `COMMAND.PUB.SYS` fails, the original MPE/iX UDC environment, if one existed, will remain intact on the system. The UDC and user file migration will terminate.
- If the open of `COMMAND.PUB.SYS` (tape) fails, the MPE/iX UDC environment will remain intact. If UDC files were successfully restored, log on to each user, and use `:SETCATALOG`.

- If RESTORE of UDC files or user files fails, study the `SYSLIST` for errors. The original MPE/iX UDC environment will remain intact. If UDC environment modifications were successful, the UDC files may be restored when DIRMIG completes execution. If there are disk space errors, adjustments to the disk space can be made when DIRMIG terminates. The user can `:SETCATALOG` as necessary to complete the UDC migration process.
- If UDC modification not successful, examine the current `DIRLOG`*nn*`.PUB.SYS` file for additional information. Troubleshoot as necessary
- If UDC modifications are successful, but the UDC environment does not exist, the UDC files will probably exist on the system. When DIRMIG terminates, use `:SETCATALOG` commands on the MPE/iX-based system to complete the migration process. Examine the current `DIRLOG`*nn*`.PUB.SYS` file for additional information. Troubleshoot as necessary.

Error Messages Returned by DIRMIG

Following is a list of DIRMIG error messages. Each error message is followed by a description of the condition which causes the error and a description of the recommended action for the user. The following error number groupings identify the tasks being performed at the time of the error.

1001 to 1999	Main DIRMIG Errors
2001 to 2999	SYSDUMP Tape Errors
3001 to 3999	RIN Migration Errors
4001 to 4999	User-Logging ID Migration Errors
5001 to 5999	Directory Migration Errors
6001 to 6999	Private Volume Migration Errors

7001 to 7999	UDC Migration Errors

Note

Error messages 5020, 5021, and 5022 are multi-line directory migration errors that provide special detail. They are described in the following section "Directory Corruption Error Messages".

1001	MESSAGE	LOGON MUST BE MANAGER.SYS,PUB.(DIRERR 1001)
	CAUSE	The user is not logged on as MANAGER.SYS,PUB.
	ACTION	The user should log on as MANAGER.SYS,PUB, and run DIRMIG.PUB.SYS again.
1002	MESSAGE	THIS PROGRAM REQUIRES SYSTEM SUPERVISOR (OP) CAPABILITY. (DIRERR 1002)
	CAUSE	The user does not have System Supervisor (OP) capability.
	ACTION	The user should provide MANAGER.SYS,PUB with OP capability and run DIRMIG.PUB.SYS again.
1003	MESSAGE	ALL JOBS AND SESSIONS MUST BE LOGGED OFF.(DIRERR 1003)
	CAUSE	With the exception of MANAGER.SYS,PUB, all jobs and sessions are not logged off.
	ACTION	The user should log off all jobs and sessions, and run DIRMIG.PUB.SYS again.
1004	MESSAGE	UNABLE TO OBTAIN NEXT LOGFILE SEQUENCE NUMBER FOR DIRLOG*nn*.PUB.SYS.(DIRERR 1004) DIRMIG LOGGING FACILITY DISABLED. CONTINUE (YES/[NO]) ? >>
	CAUSE	DIRMIG increments the current log file sequence number with each run. In this case, a file system error occurred while determining the next log file sequence number from the existing log files.
	ACTION	DIRMIG is unable to create a log file for tracking the migration process. The user has the option of (not) continuing with the migration. If the user enters YES, DIRMIG will continue. Messages normally logged to DIRLOG*nn*.PUB.SYS will be displayed to $STDLIST. If the user enters NO, DIRMIG will terminate. The user should note the file system error message displayed and troubleshoot accordingly.

1005	MESSAGE	CURRENT LOGFILE SEQUENCE NUMBER EQUALS MAXIMUM SEQUENCE NUMBER ALLOWED (99) FOR DIRLOG*nn*.PUB.SYS. (DIRERR 1005) DIRMIG LOGGING FACILITY DISABLED. CONTINUE (YES/[NO]) ? >>
	CAUSE	DIRMIG increments the current log file sequence number with each run. The sequence numbers range from 01 through 99. In this case, DIRLOG99.PUB.SYS exists on the system and DIRMIG is unable to increment the sequence number.
	ACTION	DIRMIG is unable to create a log file for tracking the migration process. The user has the option of (not) continuing with the migration. If the user enters YES, DIRMIG will continue. Messages normally logged to DIRLOG*nn*.PUB.SYS will be displayed to $STDLIST. If the user enters NO, DIRMIG will terminate. The user should then remove all DIRLOG*nn* files from PUB.SYS running DIRMIG again.
1007	MESSAGE	UNABLE TO DISPLAY LISTING OFFLINE LIST FILE NOT SUCCESSFULLY OPENED (DIRERR 1007)
	CAUSE	DIRMIG did not successfully open file LIST for redirecting output offline. Device class LP does not exist and the user did not redirect file LIST to a configured device or class; or the user entered an invalid equation for LIST.
	ACTION	Before running DIRMIG, the user should configure device class LP or issue a valid file equation for the formal file designator LIST.
1016	MESSAGE	UNABLE TO OBTAIN JOB/SESSION INFORMATION. JOBINFO INTRINSIC ERROR:#*n* (DIRERR 1016)
	CAUSE	DIRMIG is unable to check number of jobs and sessions logged on to an error return from the JOBINFO intrinsic.
	ACTION	The user should look up the JOBINFO intrinsic error return in the *MPE/iX Intrinsics Reference Manual (32650-90028)* and troubleshoot accordingly.

1017	MESSAGE	UNABLE TO OBTAIN DATE AND TIME INFORMATION. (DIRERR 1017) NLFMTDATE INTRINSIC ERROR:#*n*.
	CAUSE	DIRMIG calls the NLFMTDATE intrinsic to display date and time in the appropriate language.
	ACTION	The user should look up the indicated NLFMTDATE intrinsic error in the *Native Language Programmer's Guide (32650-90022)* and troubleshoot accordingly.

1018	MESSAGE	UNABLE TO OBTAIN NATIVE LANGUAGE SUPPORT LANGUAGE ID. (DIRERR 1018) NLGETLANG INTRINSIC ERROR #*n*. NATIVE-3000 LANGUAGE ID WILL BE USED"
	CAUSE	DIRMIG calls the **NLGETLAND** intrinsic to obtain the user-interface language ID. This error usually occurs when the **NLUSERLANG** JCW is set to a language that is not configured on the MPE/iX system.
	ACTION	The user should look up the indicated **NLGETLANG** intrinsic error in the *Native Language Programmer's Guide (32650-90022)* and troubleshoot accordingly.
1019	MESSAGE	UNABLE TO OBTAIN NATIVE LANGUAGE INFORMATION. (DIRERR 1019) NLINFO INTRINSIC ERROR: #*n*.
	CAUSE	DIRMIG is unable to obtain the appropriate NLS upshift information table (**NLINFO** item number 15) or the appropriate YES/NO language string (**NLINFO** item number 8).
	ACTION	The user should look up the **NLINFO** intrinsic error reported in the *Native Language Programmer's Guide (32650-90022)* and troubleshoot accordingly.
1020	MESSAGE	DIRMIG INTERNAL ERROR WHILE READING INPUT. (DIRERR 1020). NLSCANMOVE INTRINSIC ERROR: #*n*.
	CAUSE	DIRMIG attempts to upshift user input using the **NLSCANMOVE** intrinsic and bad status was returned.
	ACTION	The user should look up the indicated **NLSCANMOVE** intrinsic error in the *Native Language Programmer's Guide (32650-90022)* and troubleshoot accordingly.

1021	MESSAGE	UNABLE TO REPLACE UNPRINTABLE CHARACTERS WITH '' (DIRERR 1021) NLREPCHAR INTRINSIC ERROR:#*n*.
	CAUSE	DIRMIG calls the NLREPCHAR intrinsic to replace unprintable characters with a period ('.') within a string before the string is printed. NLREPCHAR is called before printing or logging a string suspected of being corrupt. Note: DIRMIG verifies strings of alphanumeric names such as RIN entries, user-logging identifiers, and all directory-related names before performing their migration.
	ACTION	The user should look up the indicated NLREPCHAR intrinsic error in the *Native Language Programmer's Guide (32650-90022)* and troubleshoot accordingly. (logged only)
1022	MESSAGE	UNABLE TO CREATE STORE.PUB.SYS PROCESS. (DIRERR 1022) CREATEPROCESS INTRINSIC ERROR:#*n*
	CAUSE	An MPE/iX internal error occurred while DIRMIG was trying to create the STORE.PUB.SYS process.
	ACTION	The user should look up the indicated CREATEPROCESS intrinsic error in the *Intrinsics Reference Manual (32650-90028)* and troubleshoot accordingly. (logged only)
1002	MESSAGE	UNABLE TO LOG MESSAGE SET=*set#* ,MSG=*msg#*. (DIRERR 1031) CATREAD INTRINSIC ERROR:#*n*.
	CAUSE	DIRMIG is unable to log the message desired because a read error occurred on the DIRMIG catalog file; the set or message number specified was not found in the file; or an internal message facility error occurred.
	ACTION	The user should look up the indicated CATREAD intrinsic error in the *Native Language Programmer's Guide (32650-90022)* and troubleshoot accordingly.

1032	MESSAGE	UNABLE TO DISPLAY MESSAGE:SET=*set#*,MSG=*msg#*. (DIRERR 1032) CATREAD INTRINSIC ERROR: #*n*.
	CAUSE	DIRMIG is unable to display the message desired because a read error occurred on the DIRMIG catalog file; the set or message number specified was not found in the file; or an internal message facility error occurred.
	ACTION	The user should look up the indicated CATREAD intrinsic error in the *Native Language Programmer's Guide (32650-90022)* and troubleshoot accordingly. (hardcoded message)
1033	MESSAGE	UNABLE TO OPEN DIRMIG CATALOG FILE*filename*. (DIRERR 1033) NLAPPEND INTRINSIC ERROR; #*n*.
	CAUSE	DIRMIG is unable to append the NLS language ID (*nnn*) to the catalog file name, so it is unable to open the appropriate message catalog file.
	ACTION	The user should look up the indicated NLAPPEND intrinsic error in the *Native Language Programmer's Guide (32650-90022)* and troubleshoot accordingly. (hardcoded message)
1034	MESSAGE	UNABLE TO CLOSE DIRMIG CATALOG FILE *filename*. (DIRERR 1034) CATALOG INTRINSIC ERROR: #*n*.
	CAUSE	DIRMIG is unable to close the message catalog file, DRCAT*nnn*PUB.SYS.
	ACTION	The user should look up the indicated CATCLOSE intrinsic error in the *Native Language Programmer's Guide (32650-90022)* and troubleshoot accordingly. (hardcoded message)

1042	MESSAGE	UNABLE TO OBTAIN VIRTUAL ADDRESS OF DST. (DIRERR 1042).
	CAUSE	An MPE/iX internal error occurred while DIRMIG was trying to migrate system table information.
	ACTION	The user should report the error message and MPE/iX status information to their Hewlett-Packard Support Representative.
		(logged only)
1043	MESSAGE	UNABLE TO OBTAIN RIN TABLE SIR. (DIRERR 1043)
	CAUSE	An MPE/iX internal error occurred while DIRMIG was trying to migrate system table information.
	ACTION	The user should report the error message and MPE/iX status information to their Hewlett-Packard Support Representative.
		(logged only)
1044	MESSAGE	UNABLE TO OBTAIN DST SIZE. (DIRERR 1044).
	CAUSE	An MPE/iX internal error occurred while DIRMIG was trying to migrate system table information.
	ACTION	The user should report the error message and MPE/iX status information to their Hewlett-Packard Support Representative.
		(logged only)
1045	MESSAGE	UNABLE TO ALTER DST SIZE. (DIRERR 1045).
	CAUSE	An MPE/iX internal error occurred while DIRMIG was trying to migrate system table information.
	ACTION	The user should report the error message and MPE/iX status information to their Hewlett-Packard Support Representative.
		(logged only)

1046	MESSAGE	UNABLE TO RELEASE RIN TABLE SIR. (DIRERR 1046).
	CAUSE	An MPE/iX internal error occurred while DIRMIG was trying to migrate system table information.
	ACTION	The user should report the error message and MPE/iX status information to their Hewlett-Packard Support Representative.
2001	MESSAGE	INVALID FILE SPECIFICATION FOR"MIGTAPE". (DIRERR 2001) UNABLE TO CONTINUE WITH MIGRATION.
	CAUSE	The user did not specify a file equation for MIGTAPE and device TAPE does not exist, or the user has specified a file equation for MIGTAPE with invalid parameters.
	ACTION	The user should configure device class TAPE for a magnetic tape or issue a valid file equation for MIGTAPE.
2002	MESSAGE	INVALID MPE SYSDUMP TAPE - INVALID CHECKSUM. (DIRERR 2002) UNABLE TO CONTINUE WITH MIGRATION.
	CAUSE	The tape mounted by the user is not a valid MPE V/E SYSDUMP tape. It could be a scratch tape or a bad SYSDUMP tape.
	ACTION	The user should make sure the correct MPE V/E SYSDUMP tape is mounted, or the user should mount a different MPE V/E SYSDUMP tape.
2003	MESSAGE	DIRECTORY INFORMATION NOT FOUND ON TAPE. (DIRERR 2003) MIGRATION NOT POSSIBLE FROM THIS TAPE SET.
	CAUSE	The MPE V/E SYSDUMP tape does not contain accounting structure information because a carriage-return SYSDUMP was performed to create the tape.
	ACTION	The user should mount a different MPE V/E SYSDUMP tape making sure it is a full or partial backup SYSDUMP tape (that is, ensure that the tape is not a carriage-return SYSDUMP tape).

2004	MESSAGE	MPE SYSDUMP TAPE IS NOT FROM AN MPE V/E-BASED SYSTEM. (DIRERR 2004) MIGRATION NOT POSSIBLE FROM THIS TAPE SET.
	CAUSE	The MPE V/E SYSDUMP tape was generated on a system which is not running the appropriate MPE V/E-based software defined as the base for migration to MPE iX.
	ACTION	The user should update their system to the appropriate MPE V/E-based version and generate a new MPE V/E SYSDUMP tape to be used by DIRMIG. Contact your Hewlett-Packard Support Representative for alternatives and for information about migration consulting services.
2005	MESSAGE	INCONSISTENT DIRECTORY SIZES FOUND ON MPE SYSDUMP TAPE. (DIRERR 2005)
	CAUSE	DIRMIG performs an integrity check on the configuration value for the directory size and found inconsistent values, signifying possible tape corruption.
	ACTION	The user should mount a different MPE V/E SYSDUMP tape. (logged only)
2006	MESSAGE	ERROR OCCURRED WHILE READING MPE SYSDUMP TAPE. (DIRERR 2006) UNABLE TO CONTINUE WITH MIGRATION. REFER TO DIRLOGnn.PUB.SYS.
	CAUSE	A file system error occurred while reading the MPE V/E SYSDUMP tape.
	ACTION	The user should check the indicated log file for the exact file system error message and troubleshoot accordingly.
2007	MESSAGE	ERROR OCCURRED WHILE CREATING DIRINFO.PUB.SYS. (DIRERR 2007) UNABLE TO CONTINUE WITH MIGRATION. REFER TO DIRLOGnn.PUB.SYS
	CAUSE	A file system error occurred while creating DIRINFO.PUB.SYS, the temporary file built by DIRMIG to hold the directory information migrated. DIRMIG was unable to open, close, or write the directory information to DIRINFO.PUB.SYS, or a file system error occurred reading directory information from the MPE V/E SYSDUMP tape.
	ACTION	The user should check the indicated log file for the exact file system error message and troubleshoot accordingly.

2008	MESSAGE	ERROR OCCURRED WHILE CREATING DIRMIG TABLES. (DIRERR 2008) UNABLE TO CONTINUE WITH MIGRATION. REFER TO DIRLOGnn.PUB.SYS
	CAUSE	DIRMIG was unable to allocate the temporary table space it required for the MPE global RINs or the MPE user-logging IDs.
	ACTION	The user should check the indicated log file for the specific error and refer to the action recommended for that error. Report the MPE/iX error information to your Hewlett-Packard Support Representative and troubleshoot accordingly.
2009	MESSAGE	UNABLE TO CREATE TEMPORARY RIN TABLE OBJECT. (DIRERR 2009)
	CAUSE	An MPE/iX internal system error occurred while DIRMIG was trying allocate temporary space it required for MPE V system table information.
	ACTION	The user should report the error message and MPE/iX status information to their Hewlett-Packard Support Representative. (logged only)
2010	MESSAGE	UNABLE TO CREATE TEMPORARY LOG ID TABLE OBJECT. (DIRERR 2010)
	CAUSE	An MPE/iX internal system error occurred while DIRMIG was trying allocate temporary space it required for MPE V system table information.
	ACTION	The user should report the error message and MPE/iX status information to their Hewlett-Packard Support Representative. (logged only)
2003	MESSAGE	RIN MIGRATION NOT SUCCESSFUL. (DIRERR 3001) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	An error occurred while trying to migrate the MPE V/E global RINs to the MPE/iX system. An MPE/iX internal system error occurred or a file system error occurred while updating the system file MISCP.CONFIG.SYS.
	ACTION	The user should check the indicated log file for the exact error message and refer to the action recommended for that error. Report the error message and MPE/iX status information to

3002	MESSAGE	MPE GLOBAL RINS NOT MIGRATED DUE TO CORRUPTION. (DIRERR 3002) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	DIRMIG has detected the possibility of RIN table corruption.
	ACTION	The MPE global RINs will not be migrated and DIRMIG will continue with the migration process. The user should check the indicated file for information on the corrupted entries. The user has the following options: keep the MPE/iX default RIN configuration values; or obtain a valid listing of MPE global RINs allocated and reallocate these RINs on the MPE/iX system using the :GETRIN command
3003	MESSAGE	NUMBER OF RIN ENTRIES DOES NOT AGREE WITH RIN TABLE HEADER VALUE. (DIRERR 3003)
	CAUSE	DIRMIG performs an integrity check on the configuration value for global RINs and the actual value in the table header. The values do not agree, signifying possible MPE V/E table corruption.
	ACTION	The user has the options described above. (logged only)
3004	MESSAGE	INVALID USER/ACCOUNT NAME FOR RIN ENTRY#*n*. (DIRERR 3004)
	CAUSE	DIRMIG performs an integrity check on each RIN entry and has detected possible corruption for RIN entry *n*.
	ACTION	The user has the options described above. (logged only)
4001	MESSAGE	USER-LOGGING ID MIGRATION NOT SUCCESSFUL. (DIRERR 4001) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	An error occurred while trying to migrate the MPE V/E user logging IDs to the MPE/iX system. An MPE/iX internal system error occurred or a file system error occurred while updating the system file LOGP.CONFIG.SYS.
	ACTION	The user should check the indicated log file for the exact error message and refer to the action recommended for that error. Report the error message and MPE/iX status information to your Hewlett-Packard Support Representative if necessary.

4002	MESSAGE	MPE USER-LOGGING IDs NOT MIGRATED DUE TO CORRUPTION. (DIRERR 4002) REFER TO DIRLOG*nn*.PUB.SYS."
	CAUSE	DIRMIG has detected the possibility of user-logging ID table corruption.
	ACTION	The MPE V/E user-logging IDs will not be migrated and DIRMIG will continue with the migration process. The user should check the indicated file for information on the corrupted entries. The user has the following options: keep the MPE/iX default user-logging configuration values; or obtain a valid listing of MPE Logging IDs and reallocate the logging identifiers by using the MPE/iX :GETLOG command.
4003	MESSAGE	NUMBER OF LOGGING ID ENTRIES DOES NOT AGREE WITH TABLE HEADER VALUE. (DIRERR 4003)
	CAUSE	DIRMIG performs an integrity check on the configuration value for MPE user-logging IDs and the actual value in the table header. The values do not agree, signifying possible MPE V/E table corruption.
	ACTION	The user has the options described above. (logged only)
4004	MESSAGE	INVALID FIELD NAME(S) FOR LOGGING ID ENTRY #*n*. (DIRERR 4004)
	CAUSE	DIRMIG performs an integrity check on each user-logging ID entry has detected possible corruption.
	ACTION	The user has the options described above. (logged only)
5001	MESSAGE	UNABLE TO CONTINUE DIRECTORY MIGRATION DUE TO ERRORS WITH DIRINFO.PUB.SYS. (DIRERR 5001) REFER TO DIRLOG*nn*.PUB.SYS."
	CAUSE	A file system error has occurred on the temporary file, DIRINFO.PUB.SYS, which contains the directory information to be migrated. DIRMIG was unable to open the file or a read error has occurred.
	ACTION	The user should check the indicated log file for the exact file error message and troubleshoot accordingly.

5002	MESSAGE	DIRECTORY MIGRATION NOT POSSIBLE DUE TO CORRUPTION. (DIRERR 5002) REFER TO DIRLOGnn.PUB.SYS.
	CAUSE	DIRMIG has detected an invalid system directory pointer or a corrupted system account index block and is unable to migrate the directory.
	ACTION	Refer to "Actions for MPE Directory Corruption", at the end of this appendix for the recommended action.
5003	MESSAGE	UNABLE TO LIST ACCOUNT NAMES DUE TO DIRECTORY . CORRUPTION. (DIRERR 5003) REFER TO DIRLOG*nn*.PUB.SYS."
	CAUSE	DIRMIG has detected an invalid system directory pointer or system account index block and is unable to list account names.
	ACTION	Refer to "Actions for MPE Directory Corruption" at the end of this appendix for the recommended action.
5004	MESSAGE	ACCOUNT *acctname* NOT SUCCESSFULLY MIGRATED DUE TO ERRORS. (DIRERR 5005)
	CAUSE	DIRMIG was unable to migrate the account, users within the account, groups within the account, or volume set definitions within the account. The errors may be due to corruption or MPE/iX internal system errors.
	ACTION	The user should check the current log file for the users, groups, or volume set definitions which were not migrated due to either corruption or MPE iX internal errors. Contact your Hewlett-Packard Support Representative for assistance if necessary.
5005	MESSAGE	*n* ERRORS(S) DETECTED DURING DIRECTORY MIGRATION (DIRERR 5005)
	CAUSE	DIRMIG detected directory corruption and keeps track of the total number of errors found.
	ACTION	The user should check the current log file for the specific directory errors logged and refer to the action recommended for each error. Contact your Hewlett-Packard Support Representative for assistance if necessary and troubleshoot accordingly.

5006	MESSAGE	*n* ERROR(S) FOUND WHILE LISTING ACCOUNT NAMES. (DIRERR 5006)
	CAUSE	The user specified the option to list the MPE V/E account names obtained from the SYSDUMP tape. DIRMIG detects directory corruption and reports the total number of errors found.
	ACTION	The user should check the current log file for the specific directory errors logged and refer to the action recommended for each error. Contact your Hewlett-Packard Support Representative for assistance if necessary and troubleshoot accordingly.
5007	MESSAGE	UNABLE TO ARCHIVE PRIVATE VOLUME INFORMATION DUE TO ERRORS WITH PVASST.PUB.SYS. (DIRERR 5007) REFER DIRLOG*nn*.PUB.SYS.
	CAUSE	DIRMIG saves volume set information to the file PVASSIST.PUB.SYS while migrating the directory. In this case, DIRMIG is unable to save volume set information found due to a file system error while accessing PVASSIST.PUB.SYS
	ACTION	The user should check the current log file for the exact file system error message and troubleshoot accordingly. Run DIRMIG.PUB.SYS again and migrate the appropriate accounts to save volume set information to PVASSIST.PUB.SYS.
5008	MESSAGE	MPE/iX DIRECTORY ERROR DURING *directory routine*. (DIRERR 5008) LOW STATUS: *low status* HIGH STATUS: *hi status*.
	CAUSE	DIRMIG made a call to the MPE/iX system routine *directory routine* and the call failed.
	ACTION	The user should report the error message and the MPE/iX status information to an Hewlett-Packard Support Representative. (logged only)

5009	MESSAGE	MPE/iX DIRECTORY ERROR DURING: *directory routine*. (DIRERR 5009) LOW STATUS: *low status* HIGH STATUS: *hi status* *directory entry name* DOES NOT EXIST. (DIRERR 5009) MIGRATION NOT SUCCESSFUL.
	CAUSE	DIRMIG made a call to the MPE/iX system routine *directory routine* and the call failed. However, the directory was being migrated with the override option enabled and *directory entry name* was expected to exist in the directory and it did not.
	ACTION	This may occur during directory migration, in which case the account, group, or user was not altered correctly. Compare the account, group, or user characteristics on MPE V/E with their equivalents on MPE/iX. Alter the MPE/iX account, group, or user characteristics by using the :ALTACCT, :ALTGROUP, or :ALTUSER commands. This may also occur during UDC migration, in which case the UDC environment for the specified user was not migrated correctly. Manually perform the :SETCATALOG command for the user on the UDC files desired. The user should report the error message and the MPE/iX status information to an Hewlett-Packard Support Representative. (logged only)
5050	MESSAGE	UNABLE TO OPEN SYSTEM FILE MPEiXDIR.PUB.SYS. (DIRWARN 5050) ALL HP-RESERVED GROUPS AND ACCOUNTS MAY BE OVERRIDDEN.
	CAUSE	DIRMIG opens and reads the MPE/iX system file, MPEXLDIR.PUB.SYS. It is from this file that DIRMIG determines what MPE/iX V/E groups and accounts are reserved, and should not be overwritten by MPE groups and accounts of the same names. If an error occurs while opening or accessing MPEiXDIR, those reserved groups and accounts may be overwritten by MPE V/E information.
	ACTION	The migration will continue. Check the current log file for the specific error and troubleshoot as necessary. If necessary, restore MPEXLDIR.PUB.SYS from the MPE/iX installation tape if MPEXLDIR does not exist on the system. Contact your Hewlett-Packard Support Representative for assistance if necessary.

6001	MESSAGE	PVASSIST.PUB.SYS MUST EXIST IN ORDER TO CONTINUE WITH PRIVATE VOLUME MIGRATION. (DIRERR 6001)
	CAUSE	PVASSIST.PUB.SYS contains MPE V/E volume set information which is saved by DIRMIG during directory migration. This file is required by DIRMIG for private volume migration. The user has specified private volume migration, but PVASSIST.PUB.SYS does not exist or the file was not created by DIRMIG because no volume set information was encountered during directory migration.
	ACTION	User information only. Private volume migration is not necessary since the accounts migrated do not have volume sets. If the user wishes to perform private volume migration, run DIRMIG again and migrate those accounts with volume set information.
6002	MESSAGE	INCOMPATIBLE VERSION NUMBER IN PVASSIST.PUB.SYS. FILE. (DIRERR 6002)
	CAUSE	This message is for future releases of DIRMIG when different versions of PVASSIST.PUB.SYS may be necessary. It is intended to keep compatibility between DIRMIG and PVASSIST.PUB.SYS. If this error message is displayed, it may indicate a corrupted PVASSIST file or may suggest that thePVASSIST file should not be used with this version of DIRMIG.
	ACTION	The user should purge PVASSIST.PUB.SYS and run DIRMIG again to recreate the file. To perform private volume migration for accounts which have already been migrated, run DIRMIG again and migrate the appropriate accounts specifying the override option. (logged only)

6003	MESSAGE	CHECKSUM ERROR IN PVASSIST.PUB.SYS FILE. (DIRERR 6003)
	CAUSE	DIRMIG performs an integrity check on PVASSIST.PUB.SYS and has detected a bad checksum. This may indicate a corrupted PVASSIST.PUB.SYS file.
	ACTION	The user should purge PVASSIST.PUB.SYS and run DIRMIG again to recreate the file. To perform private volume migration for accounts which have already been migrated, run DIRMIG again and migrate the appropriate accounts specifying the override option. (logged only)
6004	MESSAGE	PRIVATE VOLUME MIGRATION NOT POSSIBLE DUE TO ERRORS WITH PVASSIST.PUB.SYS. (DIRERR 6004) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	DIRMIG encountered one of the following problems while trying to set up for private volume migration: a version or checksum problem with PVASSIST.PUB.SYS, an MPE/iX error while trying to allocate temporary space for PVASSIST information, or a file system error while reading PVASSIST.
	ACTION	The user should check the indicated log file for the exact error message and refer to the action recommended for that error.
6005	MESSAGE	UNABLE TO CREATE OBJECT FOR PVASSIST INFORMATION. (DIRERR 6005)
	CAUSE	An MPE/iX internal error occurred while DIRMIG was trying to allocate temporary space for PVASSIST information.
	ACTION	The user should report the error message and MPE/iX status information to their Hewlett-Packard Support Representative. (logged only)

6006	MESSAGE	ERROR OCCURRED WHILE MODIFYING PVASSIST.PUB.SYS. (DIRERR 6006) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	The user has deleted a volume set, and a file system error occurred while DIRMIG was updating PVASSIST.PUB.SYS. PVASSIST may not have been updated to reflect the deleted volume set.
	ACTION	The user should check the indicated log file for the file system message logged and troubleshoot accordingly. The user must again invoke the Private Volume Migration Menu and select the option to visit volume sets to determine if the volume set was deleted correctly.
6007	MESSAGE	NO HOME VOLUME SET FOUND FOR *group.account* HOME VOLUME SET SHOULD BE *vsetname*. (DIRERR 6007)
	CAUSE	This error is generated when a VOLUTIL command script is being generated for *vsetname*. The PVASSIST.PUB.SYS file indicates home volume set for *group.account* should be *vsetname* the PVASSIST entry for *group.account* shows no home volume information. This message indicates a problem with PVASSIST.PUB.SYS.
	ACTION	The VOLUTIL command script for *vsetname* will be successfully created. However, DIRMIG will not display any spanning prompts for the volume set indicated, and commands to create the accounting structure on the volume set will not be included in the VOLUTIL command script. The user will need to create the accounting structure on the volume set manually or the user may add the appropriate commands to the script before running VOLUTIL to initialize the volume set. (logged only)

6008	MESSAGE	INCORRECT HOME VOLUME SET NAME FOUND FOR *group.account*. HOME VOLUME SET SHOULD BE *vsetname*. (DIRERR 6008)
	CAUSE	This error is generated when a VOLUTIL command script is being generated for *vsetname*. The PVASSIST.PUB.SYS file indicates that the home volume set for *group.account* should be vsetname, but the PVASSIST entry for *group.account* shows a different home volume set name. This message indicates a problem with PVASSIST.PUB.SYS.
	ACTION	The VOLUTIL command script for *vsetname* will be successfully created. However, DIRMIG will not display any spanning prompts for the volume set indicated, and commands to create the accounting structure on the volume set will not be included in the VOLUTIL command script. The user will need to create the accounting structure on the volume set manually, or the user may add the appropriate commands to the script before running VOLUTIL to initialize the volume set. (logged only)
6009	MESSAGE	ERROR(S) OCCURRED WHILE GENERATING COMMAND SCRIPT FOR *vset*. (DIRERR 6009) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	DIRMIG uses the GENCAT message facility to generate the VOLUTIL file and a CATREAD intrinsic error has occurred.
	ACTION	Commands may not have been successfully written to the VOLUTIL command file. Determine which commands were not added and modify the command file appropriately. The user should look up the CATREAD intrinsic error(s) displayed in the *Native Language Programmer's Guide (32650-90022)* and troubleshoot accordingly. It may be necessary to regenerate the command file for this volume set.
6050	MESSAGE	THE FOLLOWING GROUPS AND ACCOUNTS MAY NOT BE SPANNED TO CLASS *class* (DIRWARN 6050): *group.account*.
	CAUSE	This error is generated when a VOLUTIL command script is being generated for a volume set which contains class information. For each class definition within the volume set, DIRMIG checks each *group.account* entry in PVASSIST for home volume set information equal to class. The group.account names listed have home volume class information referencing class. DIRMIG is warning the user that spanning to volume classes is not supported on MPE/iX.
	ACTION	User information only.

6051	MESSAGE	UNABLE TO OBTAIN FILES LIMIT FOR GROUP *group*. DEFAULT FILES LIMIT USED. (DIRERR 6052)
	CAUSE	DIRMIG obtains the disk file space limit for group by executing a :LISTGROUP command and redirecting the output to the temporary file, ACCTLIST.PUB.SYS. The group does not exist in the system directory or a file system error occurred while reading the information from the temporary file ACCTLIST.PUB.SYS.
	ACTION	User information only. The user can modify the VOLUTIL command script to reflect the correct disk file space limit for group. Refer to the current log file for detailed error information. (logged only)
6052	MESSAGE	UNABLE TO OBTAIN FILES LIMIT FOR ACCOUNT *account*. DEFAULT FILES LIMITUSED. (DIRWARN 6052)
	CAUSE	DIRMIG obtains the disk file space limit for account by executing a :LISTACCT command and redirecting the output to the temporary file ACCTLIST.PUB.SYS. A file system error occurred while reading the information from the temporary file ACCTLIST.PUB.SYS.
	ACTION	User information only. The user can modify the VOLUTIL command script to reflect the correct disk storage limit for account. Refer to the current log file for the file system error message. (logged only)
6053	MESSAGE	UNABLE TO OBTAIN FILES LIMIT FOR ACCOUNT *account*. DEFAULT FILES LIMIT USED FOR THIS ACCOUNT AND ITS GROUPS (DIRWARN 6053)
	CAUSE	DIRMIG obtains the disk file space limit for *account* by executing a :LISTACCT command. The account does not exist in the system directory.
	ACTION	User information only. The user can modify the VOLUTIL command script to reflect the correct disk storage limit for the account and groups. (logged only)

7001	MESSAGE	RESTORE OF COMMAND.PUB.SYS NOT SUCCESSFUL. (DIRERR 7001) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	DIRMIG needs to restore COMMAND.PUB.SYS from the MPE SYSDUMP tape in order to proceed with UDC migration. It was unable to restore the file successfully because COMMAND.PUB.SYS was not on the tape or a RESTORE error occurred.
	ACTION	DIRMIG logs the STOREJCW value after the restore is finished. The user should check the indicated log file for the STOREJCW logged and troubleshoot accordingly. In order to migrate the UDC environment using DIRMIG, directory migration must be performed again with the override option specified.
7002	MESSAGE	RESTORE OF UDC FILES NOT SUCCESSFUL. (DIRERR 7002) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	An error occurred while restoring UDC and/or user designated files.
	ACTION	The current log file contains the STOREJCW value returned by the RESTORE process. Look up this value in the *Performing System Operating Task (32650-90137)* or *Controlling System Activity Reference Manual (32650-90155)* and troubleshoot accordingly. DIRMIG will still proceed with UDC environment migration. The user should restore the UDC/user-designated files with the :RESTORE command after DIRMIG has terminated.
7003	MESSAGE	RESTORE OF UDC FILES/USER-DESIGNATED FILES NOT SUCCESSFUL. (DIRERR 7003) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	An error occurred while restoring UDC and/or user designated files.
	ACTION	The current log file contains the STOREJCW value returned by the RESTORE process. Look up this value in the *Performing System Operating Tasks (32650-90137)* or the *(Controlling System Activity Reference Manual (32650-90155)| and troubleshoot accordingly. DIRMIG will still proceed with UDC environment migration. The user should restore the UDC/user-designated files with the* :RESTORE *command after DIRMIG has terminated.*

7004	MESSAGE	RESTORE OF UDCFILES/ALL USER-FILES NOT SUCCESSFUL. (DIRERR 7004) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	An error occurred while restoring UDC and/or user designated files.
	ACTION	The current log file contains the STOREJCW value returned by the RESTORE process. Look up this value in the *Performing System Operating Tasks (32650-90137)* or the *Controlling System Activity Reference Manual (32650-90155)* and troubleshoot accordingly. DIRMIG will still proceed with UDC environment migration. The user should restore the UDC/user-designated files with the :RESTORE command after DIRMIG has terminated.
7005	MESSAGE	UNABLE TO ATTEMPT RESTORE OF UDC FILES. (DIRERR 7005) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	An error occurred while DIRMIG was creating UDCMASTR.PUB.SYS. the indirect file used when invoking STORE.PUB.SYS; or DIRMIG was unable to successfully create the STOREPUB.SYS process.
	ACTION	The user should check the indicated log file for the specific error and troubleshoot accordingly. DIRMIG will still proceed with UDC environment migration. The user should restore the UDC/user-designated files with the :RESTORE command after DIRMIG has terminated.
7006	MESSAGE	UNABLE TO ATTEMPT RESTORE OF UDC FILE/USER-DESIGNATED FILES. (DIRERR 7006) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	An error occurred while DIRMIG was creating UDCMASTR.PUB.SYS. the indirect file used when invoking STORE.PUB.SYS; or DIRMIG was unable to successfully create the STOREPUB.SYS process.
	ACTION	The user should check the indicated log file for the specific error and troubleshoot accordingly. DIRMIG will still proceed with UDC environment migration. The user should restore the UDC/user-designated files with the :RESTORE command after DIRMIG has terminated.

7007	MESSAGE	UNABLE TO ATTEMPT RESTORE OF UDC FILES/ALL USER FILES. (DIRERR 7007) REFER TO DIRLOG*nn*.PUB.SYS.
	CAUSE	An error occurred while DIRMIG was creating UDCMASTR.PUB.SYS. the indirect file used when invoking STORE.PUB.SYS; or DIRMIG was unable to successfully create the STOREPUB.SYS process.
	ACTION	The user should check the indicated log file for the specific error and troubleshoot accordingly. DIRMIG will still proceed with UDC environment migration. The user should restore the UDC/user-designated files with the :RESTORE command after DIRMIG has terminated.
7008	MESSAGE	UDC FILENAME *filename* NOT ADDED TO UDCMASTR.PUB.SYS (DIRERR 7008)
	CAUSE	DIRMIG searches the MPE V/E COMMAND.PUB.SYS file for all UDC files to restore from the MPE SYSDUMP tape. All file names encountered are written to UDCMASTR.PUB.SYS, the indirect file created by DIRMIG for use when invoking STORE.PUB.SYS. A file system error occurred while writing the indicated *filename* to UDCMASTR.
	ACTION	DIRMIG will not attempt to restore UDC files. However, UDC environment modifications will be performed. COMMAND.PUB.SYS will be modified to reflect the UDC environment for migrated users. The user should make sure that the appropriate UDC files are restored onto the MPE/iX system. This is accomplished by running GETUDC.PUB.SYS on the MPE/iX system to create an indirect file containing the names of UDC files to restore, and using the indirect file to restore the UDC files from the MPE SYSDUMP tape. Another alternative is to check the current log file for the file system error message, troubleshoot the problem, and run DIRMIG again performing directory migration with the override option. (logged only)

7020	MESSAGE	UDC ENVIRONMENT MODIFICATIONS NOT SUCCESSFUL. (DIRERR 7020) REFER TO DIRLOG*nn*.PUB.SYS
	CAUSE	While updating the MPE/iX directory to reflect new UDC file information in COMMAND.PUB.SYS, an error was returned from a call to an MPE/iX directory services routine.
	ACTION	The current log file will contain an error log for the failed directory service routine as well as an indication of which user and or account did not have their directory entry updated (thus not enabling UDCs for that user and/or account). The user should report the error message and the MPE/iX status information (provided in the error log) to their HP Support Representative.
7021	MESSAGE	UDC ENVIRONMENT MODIFICATIONS NOT POSSIBLE DUE TO MPE OR MPE/iX COMMAND.PUB.SYS ERRORS. (DIRERR 7021)
	CAUSE	If an error occurs while restoring or accessing either the MPE/iX or the MPE V/E COMMAND.PUB.SYS file, the MPE/iX directory can not be updated to reflect new UDC information in MPE/iX COMMAND.PUB.SYS.
	ACTION	Check the current log file for the indicated errors with either the MPE iX or the MPE V/E COMMAND.PUB.SYS file and troubleshoot as necessary. Run DIRMIG again performing directory migration with the override option.
7022	MESSAGE	UNABLE TO OBTAIN MPE UDC INFORMATION. (DIRERR 7022)
	CAUSE	DIRMIG was unable to access the MPE V/E COMMAND.PUB.SYS file because it was unable to restore the file from the MPE SYSDUMP tape or it was unable to open the file.
	ACTION	The user should check the current log file for the reason why the restore or the open of COMMAND.PUB.SYS failed and troubleshoot accordingly. To migrate the UDC environment using DIRMIG, run DIRMIG again and perform directory migration with the override option.

(logged only)

7023	MESSAGE	UNABLE TO OBTAIN MPE/iXUDC INFORMATION. (DIRERR 7023)
	CAUSE	DIRMIG was unable to open, access, or obtain file characteristics for the MPE/iX COMMAND.PUB.SYS file.
	ACTION	The user should check the current log file for the reason why the open or the access failed and troubleshoot accordingly. To migrate the UDC environment using DIRMIG, run DIRMIG again and perform directory migration with the override option. (logged only)
7024	MESSAGE	*MPE/iX file system error message* ERROR OCCURRED WHILE RENAMING COMMAND.PUB.SYS. UNABLE TO RENAME COMMAND.PUB.SYS TO SAVECMD.PUB.SYS. (DIRERR 7024)
	CAUSE	Before the MPE V/E COMMAND.PUB.SYS file is restored from the MPE SYSDUMP tape, DIRMIG saves the MPE/iX COMMAND.PUB.SYS file to SAVECMD.PUB.SYS using the FRENAME intrinsic. This message is logged if the FRENAME intrinsic fails.
	ACTION	The user should check the current log file for the file system error generated by the FRENAME failure and troubleshoot accordingly. Review the UDC environment migration documentation for error recovery. To migrate the UDC environment using DIRMIG, run DIRMIG again and perform directory migration with the override option. (logged only)

7025	MESSAGE	*MPE/iX file system error message* ERROR OCCURRED WHILE RENAMING SAVECMD.PUB.SYS. CANNOT RENAME ORIGINAL MPE/iX COMMAND.PUB.SYS BACK FROM SAVECMD.PUB.SYS. (DIRERR 7025)
	CAUSE	Before the MPE V/E COMMAND.PUB.SYS is restored from the tape, DIRMIG saves the MPE/iX COMMAND.PUB.SYS file to SAVECMD.PUB.SYS. When DIRMIG is finished processing the information from the MPE V/E COMMAND.PUB.SYS file, it is purged and the original MPE/iX COMMAND.PUB.SYS is renamed back from SAVECMD via the FRENAME intrinsic. This message is logged if the FRENAME intrinsic fails.
	ACTION	The specified UDC/user-designated files will be restored. However, DIRMIG will not proceed with UDC environment migration. The user should check the current log file for the file system error generated by the FRENAME intrinsic failure and troubleshoot accordingly. After DIRMIG has terminated, manually rename SAVECMD.PUB.SYS to COMMAND.PUB.SYS to re-activate the original command file. To migrate the UDC environment using DIRMIG, run DIRMIG again and perform directory migration with the override option. (logged only)
7026	MESSAGE	MPE/iX COMMAND.PUB.SYS IS CORRUPT. (DIRERR 7026) UNABLE TO CONTINUE WITH UDC ENVIRONMENT MIGRATION.
	CAUSE	DIRMIG detected possible corruption within the existing MPE/iX COMMAND.PUB.SYS file.
	ACTION	Purge and rebuild COMMAND.PUB.SYS, and perform :SETCATALOG commands appropriately to recreate the MPE/iX UDC environment. If the existing MPE/iX COMMAND.PUB.SYS file is not necessary for UDC migration, purge COMMAND.PUB.SYS and run DIRMIG again with the override option specified. (logged only)

7027	MESSAGE	UNABLE TO BUILD A NEW MPE/iX COMMAND.PUB.SYS FILE. (DIRERR 7027)
	CAUSE	DIRMIG was unable to create a new COMMAND.PUB.SYS file. An attempt is made to create a new COMMAND.PUB.SYS file when the migrating UDC environment information will not fit in the existing COMMAND.PUB.SYS file due to file limit restrictions.
	ACTION	DIRMIG will not proceed with UDC environment migrations for migrated accounts and users. The user should check the current log file for the exact error message which caused the failure and troubleshoot accordingly. The original MPE/iX command file should still be intact. To migrate the UDC environment using DIRMIG, run DIRMIG again and perform directory migration with the override option. (logged only)
7028	MESSAGE	UNABLE TO COPY COMMAND.PUB.SYS TO COPYCMD.PUB.SYS. (DIRERR 7028)
	CAUSE	DIRMIG was unable to copy COMMAND.PUB.SYS to COPYCMD.PUB.SYS. The copy is made in the event that the original UDC environment would need to be restored in case errors occur while modifying COMMAND.PUB.SYS. UDC migration terminates.
	ACTION	The user should check the current logfile for the exact error message which caused the failure and trouble-shoot accordingly. Purge COPYCMD.PUB.SYS and run DIRMIG again. Review the UDC environment migration documentation for error recovery. (logged only)
7029	MESSAGE	MODIFICATIONS TO MPE/iX COMMAND.PUB.SYS FAILED. WILL COPY BACK ORIGINAL MPE/iX COMMAND.PUB.SYS. (DIRERR 7029)
	CAUSE	DIRMIG detected errors while modifying MPE/iX COMMAND.PUB.SYS with the new MPE UDC information. An attempt is made to restore the original MPE/iX COMMAND.PUB.SYS from COPYCMD.PUB.SYS.
	ACTION	The user should check the current logfile for the exact error message which caused the failure and trouble-shoot accordingly. UDC migration is terminated. Review the UDC environment migration documentation for error recovery.

7030	MESSAGE	UNABLE TO COPY ORIGINAL MPE/iX COMMAND.PUB.SYS BACK FROM COPYCMD.PUB.SYS. (DIRERR 7030)
	CAUSE	DIRMIG was unable to copy the original MPE/iX COMMAND.PUB.SYS back from COPYCMD.PUB.SYS. While making modifications to MPE/iX COMMAND.PUB.SYS, an error occurred. DIRMIG then attempts to restore the original MPE/iX UDC environment by copying back the original COMMAND.PUB.SYS.
	ACTION	The user should check the current logfile for the exact error message which caused the failure and trouble-shoot accordingly. The user should also determine from the current logfile why the modifications to MPE/iX COMMAND.PUB.SYS failed and trouble-shoot accordingly. Since an error occurred while modifying MPE/iX COMMAND.PUB.SYS, the contents of COMMAND is suspect. Purge COMMAND and copy/rename COPYCMD.PUB.SYS to COMMAND.PUB.SYS. This will restore the original UDC environment. Review the UDC environment migration documentation for error recovery. (logged only)
7031	MESSAGE	UNABLE TO CLOSE EXISTING COMMAND.PUB.SYS. UNABLE TO COPY BACK ORIGINAL MPE/iX COMMAND.PUB.SYS. (DIRERR 7031)
	CAUSE	DIRMIG was unable to close the COMMAND.PUB.SYS file before copying back the original MPE/iX COMMAND.PUB.SYS from COPYCMD.PUB.SYS.
	ACTION	The user should check the current logfile for the exact error message which caused the failure and trouble-shoot accordingly. UDC migration is terminated and the contents of MPE/iX COMMAND.PUB.SYS is suspect. Purge COMMAND.PUB.SYS and copy/rename COPYCMD.PUB.SYS to COMMAND.PUB.SYS. This will restore the original UDC environment. Review the UDC environment migration documentation for error recovery. (logged only)

7050	MESSAGE	CANNOT FIND UDC FILES FOR USER *user.account.* NO UDC MIGRATION FOR THIS USER. (DIRWARN 7050)
	CAUSE	DIRMIG searches the MPE V/E COMMAND.PUB.SYS file for UDC files to restore for each migrated user. This is done for those users who have UDC environment information in the MPE V/E COMMAND.PUB.SYS according to their MPE V/E directory entry. This message is displayed when DIRMIG is unable to find UDC files for a migrated user who should have an entry in the MPE V/E COMMAND.PUB.SYS file. This situation may suggest that the COMMAND.PUB.SYS on the MPE V/E system is old or has been overridden by a COMMAND.PUB.SYS file from another system with a different accounting structure.
	ACTION	DIRMIG is unable to migrate the UDC environment for this user. When DIRMIG terminates, the user needs to restore the appropriate UDC files and perform the necessary :SETCATALOG commands. (logged only)

Directory Corruption Error Messages

Following is a list of directory corruption error messages returned by DIRMIG. The error text varies depending on the directory corruption identified and the logging messages displayed when the message is written to the `DIRLOG`*nn* file.

5020	MESSAGE	CORRUPT MPE GROUP INDEX BLOCK PREFIX (DIRERR 5020): *index block prefix error message* (See below) DIRECTORY PATH.... ACCOUNT INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object _name* GROUP INDEX BLOCK PTR: *sector_location object_name* GROUPS FOR *account name* NOT MIGRATED DUE TO CORRUPTION.
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

5020	MESSAGE	CORRUPT MPE USER INDEX BLOCK PREFIX (DIRERR 5020): *index block prefix error message* (See Below) DIRECTORY PATH.... ACCOUNT INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object_name* USER INDEX BLOCK PTR: *sector_location object_name* USERS FOR *account name* NOT MIGRATED DUE TO CORRUPTION.
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

5020	MESSAGE	CORRUPT MPE VSD/VCD INDEX BLOCK PREFIX (DIRERR 5020): *vsd/vcd index block prefix error message* (See Below) DIRECTORY PATH.... ACCOUNT|| INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object_name* GROUP INDEX BLOCK PTR: *sector_location object_name* GROUP ENTRY PTR: *sector_location object_name* VSD/VCD INDEX BLOCK PTR: *sector_location object_name* VSDS/VCDS FOR *group.account* NOT MIGRATED DUE TO CORRUPTION.
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

Depending upon the circumstances, one of the following messages will be appended to error message 5020 when DIRMIG writes the message to the logging file `DIRLOG`*nn*.

INDEX BLOCK INFO WORD.... BIT 0:1 MUST EQUAL 1, NOT 0.

INDEX BLOCK INFO WORD.... ILLEGAL INDEX BLOCK ENTRY WORD SIZE.
EXPECTED WORD SIZE = *expected_value*, NOT *existing_value*.

INDEX BLOCK INFO WORD.... ILLEGAL IDENTIFIER -
EXPECTED IDENTIFIER = *expected_value*, NOT *existing_value*.

INDEX BLOCK INFO WORD.... ILLEGAL INDEX BLOCK ENTRY SECTOR SIZE FOR THIS OBJECT TYPE (*object type*).
NUMBER SECTORS (*existing value*) EXCEEDS NUMBER ALLOWED (*expected value*).

TOO MANY INDEX ENTRIES (*existing value*) FOR THIS INDEX BLOCK. EXPECTED A VALUE = *expected value*.

TOO MANY ENTRY BLOCK ENTRIES (*existing value*) SPECIFIED. \\EXPECTED A VALUE = *expected value*.

ENTRY BLOCK INFO WORD.... BIT 0:1 MUST BE 0, NOT 1.

ENTRY BLOCK INFO WORD.... ILLEGAL ENTRY BLOCK ENTRYWORD SIZE.
EXPECTED WORD SIZE = *expected value*, NOT *existing value*.

ENTRY BLOCK INFO WORD.... ILLEGAL IDENTIFIER -
EXPECTED IDENTIFIER = *expected value*, NOT *existing value*.

ENTRY BLOCK INFO WORD.... ILLEGAL ENTRY BLOCK ENTRY SECTOR SIZE. \\EXPECTED SECTOR SIZE = *expected value*, NOT *existing value*.

INVALID FATHER POINTER TO ANCESTOR - EXPECTED *expected sector location*, NOT *existing sector location*.

INVALID FATHER NAME FOR ANCESTOR - EXPECTED *expected string*, NOT *existing string*.

5021	MESSAGE	CORRUPT MPE ACCOUNT INDEX BLOCK ENTRY (DIRERR 5021): *index block entry error message* (See Below) DIRECTORY PATH.... ACCOUNT INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object_name*
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

5021	MESSAGE	CORRUPT MPE GROUP INDEX BLOCK ENTRY (DIRERR 5021): *index block entry error message* (See Below) DIRECTORY PATH.... ACCOUNT INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object_name* GROUP INDEX BLOCK PTR: *sector location object name*
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

5021	MESSAGE	CORRUPT MPEUSER INDEX BLOCK ENTRY (DIRERR 5021): *index block entry error message* (See Below) DIRECTORY PATH.... ACCOUNT INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object_name* USER INDEX BLOCK PTR: *sector_location object_name*
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

5021	MESSAGE	CORRUPT MPE VSD/VCD INDEX BLOCK ENTRY (DIRERR 5021): *index block entry error message* (See Below) DIRECTORY PATH.... ACCOUNT INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object_name* GROUP INDEX BLOCK PTR: *sector_location object_name* GROUP ENTRY PTR: *sector_location object_name* VSD/VCD INDEX BLOCK PTR: *sector_location object_name*
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

Depending upon the circumstances, one of the following messages will be appended to error messages 5021 when DIRMIG writes error message 5021 to the logging file DIRLOG*nn*.

INVALID DIRECTORY ENTRY NAME.... CONTAINS ILLEGAL CHARACTERS: *existing string*

NUMBER OF ENTRY BLOCK ENTRIES (*existing value*)
OUTSIDE LEGAL RANGE OF *expected_min* - *expected_max*
FOR THIS OBJECT TYPE (*object_type*).

5022	MESSAGE	CORRUPT MPE ACCOUNT DIRECTORY ENTRY (DIRERR 5022): *entry block entry error message* (See Below) DIRECTORY PATH.... ACCOUNT INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object_name*
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

5022	MESSAGE	CORRUPT MPE GROUP DIRECTORY ENTRY (DIRERR 5022): *entry block entry error message* (See Below) DIRECTORY PATH.... ACCOUNT INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object_name* GROUP INDEX BLOCK PTR: *sector_location object_name* GROUP ENTRY PTR: *sector_location object_name*
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

5022	MESSAGE	CORRUPT MPE USER DIRECTORY ENTRY (DIRERR 5022): *index block entry error message* (See Below) DIRECTORY PATH.... ACCOUNT INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object_name* USER INDEX BLOCK PTR: *sector_location object_name* USER ENTRY PTR: *sector_location object_name*
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

5022	MESSAGE	CORRUPT MPE VSD/VCD DIRECTORY ENTRY (DIRERR 5022): *entry block entry error message* (See Below) DIRECTORY PATH.... ACCOUNT INDEX BLOCK PTR: *sector_location object_name* ACCOUNT ENTRY PTR: *sector_location object_nam* GROUP INDEX BLOCK PTR: *sector_location object nam* GROUP ENTRY PTR: *sector_location object_nam* VSD/VCD INDEX BLOCK PTR: *sector_location object_name* VSD/VCD ENTRY PTR: *sector_location object_name*
	CAUSE	Refer to "Causes of MPE Directory Corruption".
	ACTION	Refer to "Action for MPE Directory Corruption". (logged only)

Depending upon the circumstances, one of the following messages will be appended to error message 5022 when DIRMIG writes error message 5022 to the logging file DIRLOG*nn*.

INVALID DIRECTORY ENTRY NAME.... CONTAINS ILLEGAL CHARACTERS: *existing value*

INDEX ENTRY NAME DOES NOT EQUAL THE FIRST NAME IN THE ENTRY BLOCK - EXPECTED ENTRY NAME = *expected string*, NOT *existing value.*

INDEX ENTRY NAME DOES NOT EQUAL THE FIRST NAME IN THE ENTRY BLOCK - EXPECTED ENTRY NAME = *expected string,* NOT *existing value.*

Causes of MPE Directory Corruption

Directory corruption is detected by examining index blocks and entry blocks before the particular components

within those blocks are used for migration. Corruption is assumed when values examined within those blocks do not match predetermined expected values. Often the values provide a check-and-balance of the directory structure. Other values are specific to the type of block being examined, for example, values specific only to the account index entry blocks.

Detected directory corruption can be the result of actual corruption of the MPE V/E directory or data errors caused by transmission errors when performing I/O to and from the SYSDUMP tape. While unlikely, it is not uncommon for minor directory corruption to go unnoticed on a system. This is especially true if the corruption is detected in an obsolete account, group, or user. However, some corruption can be catastrophic. If DIRMIG detects corruption, those objects flagged should be examined.

When corruption is detected, DIRMIG provides a Directory Path to the erroneous directory block. Sector offsets (from DIRBASE) are provided to guide qualified individuals to the corrupted directory component. The names of ancestors are also provided within the directory path. The names also provide assistance in locating the corrupted component.

Actions for MPE Directory Corruption

If DIRMIG flags directory components as corrupt, the problem must be characterized. Several activities can be put into action concurrently:

- Attempt directory migration from another SYSDUMP tape and specify only those account subsets where the corruption was flagged. If the migration is successful, chances are that the first tape was bad.
- Verify that access to those flagged directory objects are restricted due to corruption. On the migrating MPE V/E system, use LISTDIR5 to dump accounts,

users, groups, and volume set definitions. If the directory is corrupt, LISTDIR5 output will reflect such symptoms.

Contact your Hewlett-Packard Support Representative who may have access to tools to verify directory corruption.

If it appears that the tape is bad, create a new tape and run DIRMIG again.

Caution

If it appears that the directory is corrupt do not attempt to fix it. Contact your Hewlett-Packard Support Representative and describe the problem's symptoms. The Hewlett-Packard Support Representative will advise you on how to proceed.

H

Notes On Migrating TurboIMAGE/V Database

This appendix is intended for anyone responsible for migrating TurboIMAGE databases between an MPE V/E-based system and an MPE/iX-based system. This appendix is divided into the following subjects:

- "Migrating a TurboIMAGE/V Database to MPE/iX" explains some basic considerations about migrating a TurboIMAGE database from MPE V/E to MPE/iX and shows how to move a single database. Examples are provided to illustrate the migration of a database from MPE V/E to MPE/iX.
- "Migrating a TurboIMAGE/iX Database Back to MPE V/E" explains some basic considerations about migrating a TurboIMAGE database from an MPE/iX-based system to an MPE V/E-based system. Examples are provided to illustrate the migration of a database from MPE/iX to MPE V/E.
- "Migrating Multiple Databases" explains some basic considerations about migrating multiple TurboIMAGE databases using the STORE/RESTORE facility and DIRMIG.
- "Compatibility Considerations" describes some compatibility issues for application programs.

Note

The examples found in this appendix assume that identical groups and accounts exist on both the MPE V/E-based system and the MPE/iX-based system.

Migrating a TurboIMAGE/V Database to MPE/iX

Migration of a database is a simple process and should take relatively little effort in most cases. The process consists of performing a DBSTORE/DBRESTOR of each database you need to move from an MPE V/E-based system to an MPE/iX-based system. The most important step to remember prior to migration is identifying the databases which have Intrinsic Level Recovery (ILR) enabled. ILR should be disabled for a successful migration. Please read the following steps completely before you try to migrate a TurboIMAGE database.

Using DBSTORE/DBRESTOR is a simple way to move single databases between systems where identical accounting structures exist. However, if you need to move more than one database (or application program), it may be more appropriate to migrate the entire operating system directory and all the associated databases. The procedure for migrating a directory and a number of databases is described in the next section.

The following steps summarize the process of migrating a single database from an MPE V/E-based system to an MPE/iX-based system. Following the list is an explanation and example of how to perform each of the steps.

1. Disable all flags (ILR, ROLLBACK, etc.) for the database (if in use). This is important to do since a database with ILR enabled which is moved from an MPE V/E-based to an MPE/iX-based system cannot be opened. Please refer to the *TurboIMAGE/V Reference Manual (32215-90050)* for more information on the ILR files.
2. Perform a DBSTORE or STORE of the database you want to move. An example of how to use DBSTORE follows this list. DBSTORE and its counterpart DBRESTOR should be in the `PUB.SYS` group and

account. Use of the :STORE command is explained in the next section.

3. Mount the DBSTORE tape on the tape drive connected to the MPE/iX-based system and perform a DBRESTOR or RESTORE. An example of using DBRESTOR follows. An example of using the :RESTORE command is given in the next section.

At this point, the basic migration is finished and the database can now be accessed. ILR does not need to be enabled to ensure database integrity. The use of ILR on MPE/iX is discussed in the next section.

The following examples show how to disable ILR and to perform a STORE and RESTORE of a TurboIMAGE database. It is recommended that ILR be disabled before migrating the database.

Example of Disabling ILR

This example shows how to disable ILR:

```
:RUN DBUTIL.PUB.SYS                    specify the database

>>SHOW ORDERS FOR FLAGS
>>DISABLE ORDERS FOR ILR
ILR DISABLED
>>EXIT
:
```

Example of Using DBSTORE (MPE V/E)

The following example shows how the ORDERS database was stored to tape on an MPE V/E-based system with the DBSTORE program. DBSTORE was run from the system console. The example shows both the prompts to the DBSTORE command and the replies to the tape request.

Comments to the example are shown to the right of the input and are bracketed by << >> symbols.

```
:RUN DBSTORE.PUB.SYS   <<run command>>

HP32215C.00.24 TurboIMAGE/3000: DBSTORE (C) COPYRIGHT HEWLETT-PACKARD CO. 1978

WHICH DATA BASE? ORDERS   <<database name>>

STORE/RESTORE, VERSION 2 (C) 1981 HEWLETT-PACKARD CO.
WED, AUG 5, 1987,  9:45 AM

?9:45/#S363/85/LDEV# FOR "DBSTORE" ON TAPE (NUM)?   <<system prompt>>
=REPLY 85,7   <<console reply>>
9:45/#S363/85/Vol (unlabeled) mounted on LDEV# 7   <<system reply>>

FILES STORED:  7   <<# of files>>

DATA BASE STORED   <<confirmation>>
```

Example of Using DBRESTOR (MPE/iX)

The following example shows how the ORDERS database was restored on an MPE/iX based system.

```
:RUN DBRESTOR.PUB.SYS   <<run command>>

HP30391C.00.00 TurboIMAGE/iX: DBRESTOR (C) COPYRIGHT HEWLETT-PACKARD CO. 1987

WHICH DATA BASE? ORDERS   <<database name>>

STORE/RESTORE, VERSION 2 (C) 1981 HEWLETT-PACKARD CO.
WED, AUB  5, 1987, 10:07 AM

?10:07/#S363/70/LDEV# FOR "DBRESTOR" ON TAPE (NUM)?   <<system prompt>>
REPLY 70,7   <<console reply>>

WILL RESTORE 1 DATABASES; NUMBER OF FILES ON TAPE = 7   <<# of files>>

FILES RESTORED:  7   <<confirmation>>
```

At this point the database ORDERS has been successfully migrated to an MPE/iX operating system. The database is now ready for access through either the QUERY facility running in CM or through application programs. ILR does not have to be enabled to ensure the logical integrity of the database under MPE/iX. Please refer to the discussion of ILR and transaction management in the next section.

Migrating a TurboIMAGE/XL Database Back to MPE V/E

It may be desirable to move a database from an MPE/iX-based system back to an MPE V/E-based system. Moving back to an MPE V/E-based system is almost identical to the move from an MPE V/E-based system and will take about the same amount of time. Note that the TurboIMAGE/XL data set files must not exceed the MPE V/E file system limits.

If a TurboIMAGE/XL database is going to be moved back to an MPE V/E system, the size of the data files in the database must not exceed 0.5 gigabytes. If any file exceeds the limit, you will not be able to migrate the database back to an MPE V/E system. (You can quickly check the file sizes by using the `:LISTF,2` command.) You will encounter this problem only if you have increased the capacities of a TurboIMAGE/XL database to exceed the MPE/V file system limits.

When storing a TurboIMAGE/XL database for migration, you must use the "transport" option of the `:DBSTORE` command. The transport option calls the `:STORE` command and writes a different format tape. The tape can then be read and restored on an MPE V/E-based system. The DBSTORE will execute successfully without the transport option, but the STORE tape will not restore on the MPE V/E-based system. The transport option also instructs STORE to check the size of the data set files. If any files greater than 0.5 gigabytes are encountered, the DBSTORE will terminate when it writes past this limit.

Because DBSTORE will not give you advance warning of oversize files, you should check the file sizes before running DBSTORE. The easiest way to check the file sizes is to use a `:LISTF ORDERS@,2` command to view all the files associated with the `ORDERS` database and the sizes of the file

The following examples show how the ORDERS database was migrated from an MPE/iX system back to an MPE V/E-based system.

Example of Using DBSTORE (MPE/iX)

This example shows how ORDERS is stored to tape on an MPE/iX-based system.

```
:RUN DBSTORE.PUB.SYS:INFO="TRANSPORT"   <<run with transport option>>

HP30391C.00.00 TurboIMAGE/XL: DBSTORE (C) COPYRIGHT HEWLETT-PACKARD CO. 1987

WHICH DATABASE? ORDERS   <<database name>>

Transport Store-Restore for MPE/iX, v. A.01.00 WED,   AUG 5, 1987, 10:03 AM

FILES STORED:   7   <<# of files>>

DATA BASE STORED   <<confirmation>>

END OF PROGRAM
:
```

Example of Using DBRESTOR (MPE V/E)

This example shows how the ORDERS database is restored back on an MPE V/E-based computer system.

```
:RUN DBRESTOR.PUB.SYS<<run command>>

HP30391A.01.00 TurboIMAGE: DBRESTOR (C) COPYRIGHT HEWLETT-PACKARD CO. 1987

WHICH DATABASE? ORDERS<<database name>>

STORE/RESTORE, VERSION A.01.00 (C) 1986 HEWLETT-PACKARD CO.
WED,AUG 5, 1987, 9:56 AM

WILL RESTORE 1 DATABASES; NUMBER OF FILES ON TAPE =   7   <<system response>>

FILES RESTORED:   7   <<# of files>>
```

Migrating Multiple Databases

The second method of migrating TurboIMAGE databases and applications is to use the STORE/RESTORE facility and DIRMIG. This method allows you to move all the databases and application programs on your system at the same time. In addition, DIRMIG recreates your existing account structure on the MPE/iX system. All user identities, passwords, UDCs, and permissions can be moved intact from an MPE V/E-based system to an MPE/iX-based system.

There are five basic steps to migrating a block of databases and application programs.

1. Disable ILR, if it has been enabled for any of the databases to be migrated.
2. Perform a full SYSDUMP of the system you want to migrate.
3. Install the SYSDUMP tape on the tape drive of the MPE/iX-based system you are moving onto.

4. Run the DIRMIG utility and specify the files and accounts you want recreated on the MPE/iX machine.
5. Perform a STORE of all the database files from the MPE V/E-based system and restore the files on the MPE/iX system.

After completing the steps listed, you should have a duplicate of your MPE V/E directory, databases, and application programs installed on the MPE/iX system. You will be able to run your existing applications (in CM) and access the databases in much the same manner as you did on the MPE V/E-based system.

Note

For more information about using DIRMIG to migrate your databases, refer to Appendix G, "Using the Directory Migration Tool".

Compatibility Considerations

There are some compatibility issues for application programs of which you should be aware. These compatibility issues are discussed below.

ILR and MPE/iX

ILR is enabled differently on MPE/iX than on MPE V/E. Due to these differences, ILR should be disabled before making a copy of a database you want to move across systems.

If an MPE V/E-based system fails during the execution of a `DBPUT` or `DBDELETE`, broken chains can result and the structural integrity of the database is lost. Enabling ILR on the MPE V/E-based system guarantees the structural integrity of the database since TurboIMAGE/V logs each `DBPUT` and `DBDELETE` to the ILR log file (`DBFILE00`) before modifying the database. If the system fails during the execution of the `DBPUT` or `DBDELETE`, the ILR file is used to recover the database.

On the MPE/iX based system, TurboIMAGE uses the operating system services (transaction manager) to ensure that there are no broken chains in the database. The structural integrity of the database is preserved even if ILR is not enabled on MPE/iX.

Enabling ILR in MPE/iX simply causes the operating system to flush some portions of memory to disk at the end of each `DBPUT` or `DBDELETE`. The result is that less work might be lost in the event of a system failure.

Known Incompatibilities in the Status Array

- New status codes all are negative (-) values.
- New error messages go with the new negative status codes.
- When running in CM, status array [10] is the PB-relative address pointing to the TurboIMAGE/XL stub and not to the NM intrinsic.
- When running in NM, the database address is 32 bits long.
- When running in NM, the PB-relative and DB-relative offsets are not returned in the status array.

DBEXPLAIN and the TurboIMAGE/XL Status Array

For NM programs, addresses are 32 bits rather than 16 bits. Since the status array is not large enough to hold all the information, some information is saved in run-time control blocks (RTCB). The information stored in the RTCB is from the recently executed intrinsic.

Because of this arrangement, you must call `DBEXPLAIN` immediately after the intrinsic which encounters an error in order to obtain an accurate display. When migrating application programs, look for multiple status arrays, since your program must use more than one status array to create this situation.

For example in CM programs `DBEXPLAIN` is compatible with the TurboIMAGE/V implementation with the

exception that the PB-offset of the CM stub is returned instead of the offset of the intrinsic.

For NM programs the status array is not compatible, however the display for `DBEXPLAIN` is compatible. For TurboIMAGE/XL `DBEXPLAIN` merges the status array and RTCB values. However be aware that the RTCB values are valid only for the last intrinsic. Therefore, `DBEXPLAIN` must be called before any other TurboIMAGE/XL intrinsic to obtain the correct status array. Values from an old status array cannot be saved and used to call `DBEXPLAIN` at a later time.

Native Mode Code Issues

There are two issues associated with running application programs in NM:

- Real data types
- Data alignment

Real Data Types

TurboIMAGE/iX is insensitive to the difference between HP 3000 and IEEE real data type formats (floating-point decimals). However, other products are sensitive to the differences. Some of the products which are sensitive are:

- CM compilers
- QUERY/CM
- DBchange/V
- VPLUSN
- other utilities

The NM compilers available on MPE/iX have options to indicate the real data type format.

Note that 900 series coprocessors require the use of IEEE real data types.

Real data type formats can be converted by customer code or utilities using the `HPFPCONVERT` intrinsic.

Data Alignment

NM programs must use the HP 3000 compiler options. TurboIMAGE/XL data buffers and internal data structures are still aligned on 16-bit boundaries.

Using the Migration Planning Tool

MPT is used in conjunction with its companion Migration Toolset utilities, RTM and OCA, during the analysis and planning stage of the migration effort. MPT runs on MPE V/E-based systems. Included in this appendix are discussions on the following subjects:

- "Migration Planning Tool Operation" provides a general overview of MPT. This section is reference in nature.
- "Generating an Installed Products Report" provides instructions on how to generate a listing of Hewlett-Packard software products installed on your system.
- "Creating the Migration Planning Tool Database" describes how to use MPT programs to create the MPT database required by the MPT report-generation program.
- "Defining User Migration Options" describes how you can instruct MPT to analyze MPE V/E files according to which MPE/iX environment (for example, CM) you want them to migrate to. The results of the analysis are important for determining accurate MPE/iX disk space requirements for your particular system.
- "Generating Migration Planning Reports" provides instructions on how to generate six types of migration planning reports.

- "Evaluating Migration Planning Reports" describes the formats and contents of the six types of migration planning reports. This section is reference in nature.
- "Running the Migration Planning Tool in Batch Mode" provides instructions on how to use a provided job stream file to create the MPT database and generate reports.
- "Migration Planning Tool Program Error Messages" provides a comprehensive list of program error messages returned by MPT.

Migration Planning Tool Operation

MPT is a package of programs that provides a first look at the potential issues arising from migrating files from an MPE V/E-based computer system to an MPE/iX-based computer system. As a part of the Migration Toolset, MPT is an addition to the capabilities of RTM and OCA. MPT is a TELESUP utility and, therefore, not a fully supported PUB.SYS product. When RTM is run prior to creating the MPT database, MPT can combine the information compiled by RTM into the final MPT reports. MPT can identify those program and segmented library files that require further analysis by OCA.

MPT performs the following tasks:

1. Generates a report listing the Hewlett-Packard software products that are fully or partially installed on your MPE V/E-based system.
2. Creates a database (MPTDB) containing migration-related information about files on your MPE V/E-based system.

3. Generates six types of migration planning reports from information collected from the database. The reports contain information on:

   - Disk capacity estimates for files migrating from an MPE V/E-based system to an MPE/iX-based system.
   - Problem areas in files that may be encountered after the files are migrated to an MPE/iX-based system.

MPT runs entirely on an MPE V/E-based HP 3000 Computer System in either interactive mode or batch mode. MPT is easy to use by inexperienced programmers and analysts, yet provides in-depth detail for more sophisticated users.

MPT is not intended to be an all-encompassing tool, but rather an aid for providing a sense of direction to the migration process. Its purpose is to communicate estimated MPE/iX disk capacity requirements and to uncover incompatibilities that might affect the migration effort. The MPT package does not deal with the implications of adding new applications nor any consequences related to system performance.

In order to obtain the greatest benefit from MPT, you should install RTM and have it log data for a period of time (at least one week, preferably throughout one entire business cycle) so that as many of your programs as possible will be run and monitored by it.

If you have been running RTM on your system, this data should be made available to MPT during its analysis run. Normally this is accomplished by making sure the MPE log files containing RTM data are present in `PUB.SYS`. If you have had to purge the MPE log files containing RTM data, you must restore them from backup tapes.

Figure I-1 illustrates the major modules of MPT and how they work together to create the MPTDB database and to generate reports. Descriptions of each of the major modules follow.

The following files are included in the Migration Planning Tool product:

File Name	Description
MPTPROD.MPT.TELESUP	The MPT program that identifies Hewlett-Packard products installed on the system.
MPT.MPT.TELESUP	The major data collection program for MPT. It scans each disk file found on the system for potential migration difficulties. It creates the MPTDB database then loads the collected data into this database
MPTRTM.MPT.TELESUP	An MPT program that extracts RTM migration event data from MPE log files and adds the data to the MPTDB database. You must have previously used RTM to collect and log the events.
MPTOPTNS.MPT.TELESUP	A parameter file used to select specific migration options. This file is used by the MPTREPT program.
MPTREPT.MPT.TELESUP	The program that generates six types of

	reports from information collected from the MPTDB database.
`MPTJOB.MPT.TELESUP`	A job stream file that can be used to create the MPTDB database. It also generates several levels of migration planning reports and the installed products report.

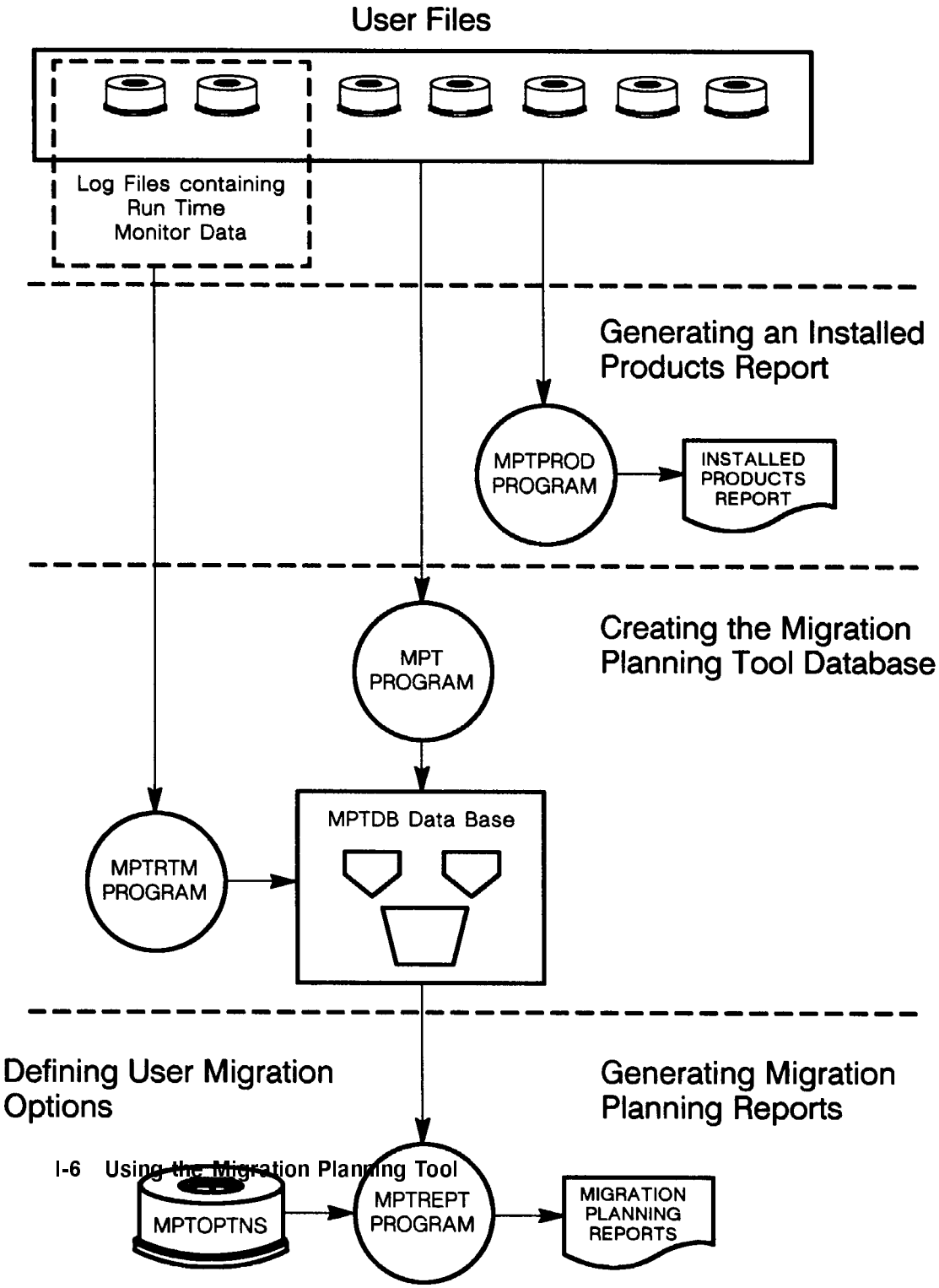


Figure I-1. Migration Planning Tool Program Flow

Generating an Installed Products Report

The MPTPROD program (`MPTPROD.MPT.TELESUP`) generates a report that enables you to identify Hewlett-Packard products installed on the system being analyzed. The purpose of this report is to alert you to the existence of products on your MPE V/E-based system. A full product integrity check is not actually performed for this report; rather, the system is checked to see if the proper files and system SL segments are present for each product. Installed product reports are created separately from the MPTDB database.

1. **Specify an Output Device:** To direct MPTPROD output to a device other than $STDLIST, use the :FILE command to equate the formal file designator LIST to the output device of your choice.
2. **Run the installed products report generation program:** You run the program MPTPROD by entering the following command:

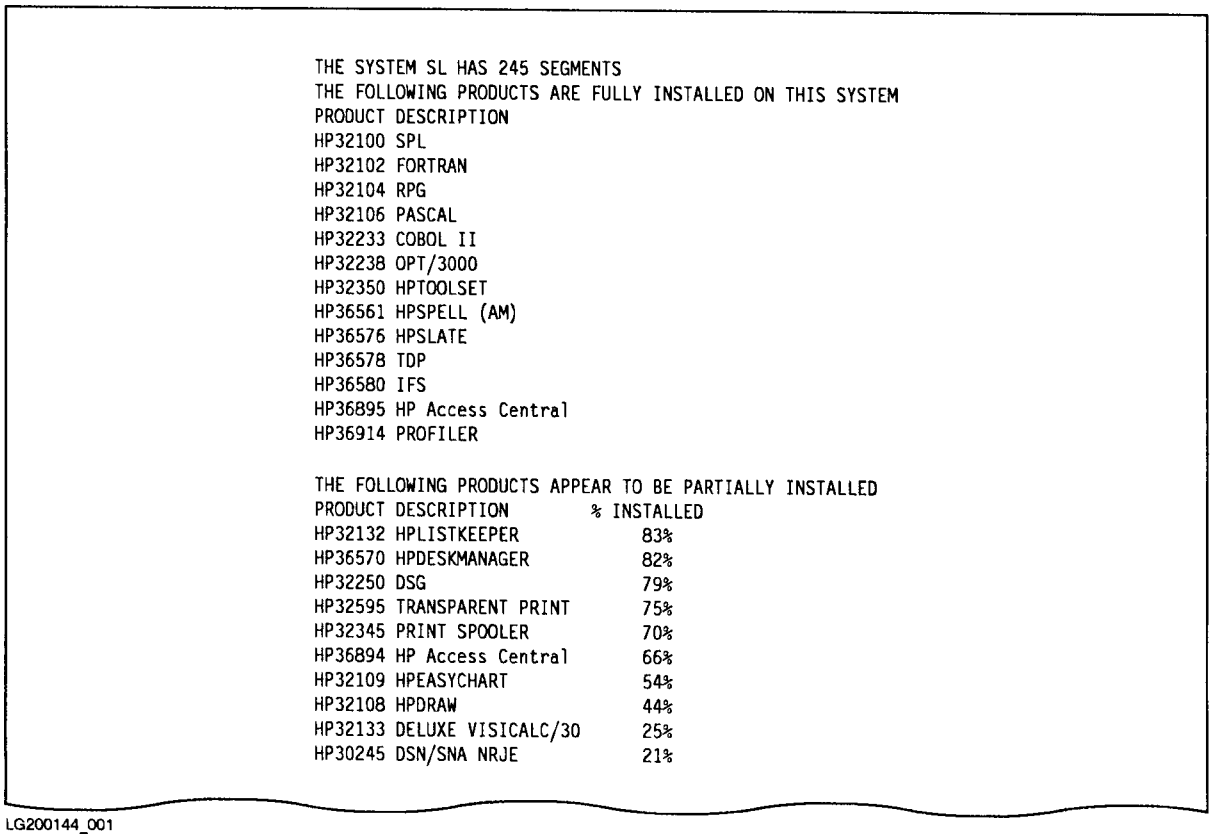
```
: RUN MPTPROD.MPT.TELESUP
```

The MPTPROD program collects data by scanning the MPE V/E-based system directly. The MPTPROD program does not log its results into the MPTDB database but rather produces a finished report directly.

Figure I-2 is a sample of an MPT Installed Products Report.

```
THE SYSTEM SL HAS 245 SEGMENTS
THE FOLLOWING PRODUCTS ARE FULLY INSTALLED ON THIS SYSTEM
PRODUCT DESCRIPTION
HP32100 SPL
HP32102 FORTRAN
HP32104 RPG
HP32106 PASCAL
HP32233 COBOL II
HP32238 OPT/3000
HP32350 HPTOOLSET
HP36561 HPSPELL (AM)
HP36576 HPSLATE
HP36578 TDP
HP36580 IFS
HP36895 HP Access Central
HP36914 PROFILER

THE FOLLOWING PRODUCTS APPEAR TO BE PARTIALLY INSTALLED
PRODUCT DESCRIPTION        % INSTALLED
HP32132 HPLISTKEEPER            83%
HP36570 HPDESKMANAGER           82%
HP32250 DSG                     79%
HP32595 TRANSPARENT PRINT       75%
HP32345 PRINT SPOOLER           70%
HP36894 HP Access Central       66%
HP32109 HPEASYCHART             54%
HP32108 HPDRAW                  44%
HP32133 DELUXE VISICALC/30      25%
HP30245 DSN/SNA NRJE            21%
```



LG200144_001

Figure I-2. Sample MPT Installed Products Report

The first part of the report generated by MPTPROD lists Hewlett-Packard products that are fully installed on the system. A product is considered fully installed if all system SL segments unique to the product are found in `SL.PUB.SYS` and if the required disk files for the product (or a core set of them) are present on the system.

The second part of the report generated by MPTPROD lists products that appear to be at least partially installed. A product is considered partially installed if all system SL segments unique to the product are found

in SL.PUB.SYS and at least one of the required disk files is found. (It is possible that there are no required SL segments for this product). The % INSTALLED column is the percent of files required by the product actually found by MPTPROD.

Note

Some Hewlett-Packard products have files which are purged as a part of the installation process. It is normal for these products to show up as less than fully installed. Products that are not installed are not listed.

Creating the Migration Planning Tool Database

You must create the MPTDB database before you can generate MPT migration planning reports. Two MPT programs, MPT.MPT.TELESUP and MPTRTM.MPT.TELESUP, are used to create the database and load it with data collected from files located on the system being analyzed. Normally, you need only create the database once for each system being analyzed. If the database already exists from a previous run, new information is appended.

It is recommended that you create the MPTDB database using the MPTJOB .MPT.TELESUP job stream file supplied by the MPT/3000 product. (Refer to "Running MPT in Batch Mode" for details.) This batch job will also generate a set of migration planning reports and an installed product report.

Note

You should run the RTM before running MPT. Migration event data placed by RTM in system log files can be collected by MPT into the MPTDB database and accessed whenever MPT generates migration reports. MPT can be run without benefit from RTM but the quality of the reports is greatly improved if this data is available.

To create the MPTDB database, follow the steps outlined below. Some of the steps are not required, but all must be considered:.

1. **Run RTM:** For details on using RTM, refer to Appendix C, "Using the Run Time Monitor".
2. **Size the Database:** In order to properly size the data sets in this database, MPT must know approximately how many disk files the database will hold. If the database is built too small, MPT or successive programs will abort with database errors. To avoid errors, size the database slightly larger than the estimated need.

   MPT determines the maximum allowable size of the database by checking the `MAXFILES` Job Control Word (JCW). If the default value for `MAXFILES` (30,000) is unsuitable for your needs, you will have to set it to your selected value prior to running MPT. For example, to create a database large enough to hold 50,000 files you would enter the following command prior to running MPT:

   ```
   : SETJCW MAXFILES=50000
   ```

3. **Create and Load the Database:** During this step, it is recommended that all other users be off the system, as the system will be extremely busy, and disk files should not be accessed by other programs. You must have System Manager (SM) capability prior to running the MPT program.

Note

Schedule adequate time to run the MPT program. The MPT program may take anywhere from thirty minutes to eight hours to create and load the MPTDB database, depending upon the number and types of disk files on your system.

Run the MPT.MPT.TELESUP program to create and load the MPTDB database used by MPT. Normally, MPT scans all disk files on the system including those on mounted and enabled private volumes. You can optionally limit the number of files scanned by MPT by using the INFO= parameter of the :RUN command to specify the MPE fileset you want scanned.

The following example executes the MPT program, specifying that only files in the ACCTING account are to be scanned:

```
: RUN MPT.MPT.TELESUP;INFO="@.@.ACCTING"
```

The MPT program creates a database (MPTDB) to contain the results of the analysis. Information regarding each file scanned is logged in the MPTDB database. The information logged can vary depending on the type of file. It includes some classification to identify the type of the file. The disk capacity requirement for the file is projected for each of the following MPE/iX modes: CM, NM, and Translated Mode

As each disk file is opened and scanned, program files, SL files, job stream files, and UDC files are examined more closely to see if they contain any migration events. If so, these events will also be logged in the database.

4. **Load RTM Events:** Run the MPTRTM program by entering:

```
:RUN MPTRTM.MPT.TELESUP
```

At the prompt LOG FILES ON TAPE OR DISC?, you must respond with DISC.

At the prompt LOG FILES TO BE SCANNED?, respond with either of the following:

`ALL`	MPTRTM scans through all log files found on the system.
m,n	MPTRTM scans through the system log files numbered *m* to *n*.

When RTM events are found recorded in the selected log files, they are entered into the MPTDB database.

This concludes the data collection portion of MPT. You now use the MPTREPT program to generate migration planning reports.

Defining User Migration Options

An MPE V/E file has different disk space requirements depending upon which of the following MPE/iX execution modes you plan to migrate it to:

- Compatibility Mode (CM)
- Native Mode (NM)
- Object code Translated Mode (TM)

For each file in the MPTDB database, MPT estimates the amount of disk space required by the file for each of these modes. The MPTREPT program includes this information in its migration planning reports. In addition, when MPT calculates final disk space considerations for all files included in the report, you select one of the modes for the calculations. The final disk space considerations are included in a general information summary automatically printed each time MPTREPT is run. To determine the chosen mode for migration, the MPTREPT program checks a special parameter file associated with the formal designator `MPTOPTNS`. If the file is not specified in `MPTOPTNS`, MPTREPT selects CM as the chosen migration mode.

You can encode user migration options for selected filesets into the special parameter file where the MPTREPT program can access it. The final disk space will be calculated using your chosen mode for each file (instead of the default, CM). A parameter file, `MPTOPTNS.MPT.TELESUP`, is supplied by MPT and already contains a set of migration options you would probably use.

To set or change the user migration options, use the text editor of your choice to modify `MPTOPTNS.MPT.TELESUP` or create your own parameter file and use a file equation to equate your file to the formal designator (`MPTOPTNS`) expected by the MPTREPT program. This parameter file contains key words to identify the files to select and some action for those files.

You can specify the files to select for an action in a variety of ways. The following key words can be used alone or in combination to select files:

`FNAME=`	Specify the fully qualified file name. (MPE wildcard characters are allowed.)
`FCODE=`	Specify the numeric MPE file code for the files.
`CLASS=`	Specify the mnemonic file classification (for example, `PROGRAM`).
`LANGUAGE=`	Specify the programming language chosen by MPT for the file.

In addition, you can specify the file size (in sectors) for files which you are adding to the report, but which are not located on the system being analyzed. Only the NM file size is required, but you can supply the others if you desire to see them on the detail level reports. Key words to supply file sizes are:

`CM=`	Specify the CM file size in sectors.
`NM=`	Specify the NM file size in sectors.

`TM=`	Specify the Translated Mode file size in sectors.
`OT=`	Specify the other mode file size in sectors.

Actions can be one of the following:

`CM`	Migrate these files into CM. (This is the default.)
`NM`	Migrate these files into NM.
`TM`	Migrate these files into object code translation mode.
`OT`	Migrate these files into "other" (as yet undefined).
`Delete`	Do NOT migrate these files to the MPE/iX system.
`Add`	Add a file to the MPE/iX system which does not already exist on the MPE V/E system.

Note

To ADD a file, you must also supply the file name and NM file size with the `FNAME=` and `NM=` key words (see below).

The MPTOPTNS file could look like this:

```
FCODE=1029;LANGUAGE=SPL;ACTION=TM
CLASS=PROGRAM;LANGUAGE=COBOL;ACTION=CM
CLASS=IMAGE;ACTION=NM;FNAME=@.@.ACCTNG
```

The sample MPTOPTNS file above is interpreted in the following manner:

- Use translation mode disk space estimates when calculating final disk space considerations for SPL program files included in the report.

- Use CM disk space estimates when calculating final disk space considerations for COBOL program files included in the report.
- Use NM disk space estimates when calculating final disk space considerations for IMAGE files located in the account `ACCTNG` included in the report.

Generating Migration Planning Reports

You generate MPT migration planning reports after you have created and loaded the MPTDB database. If you have not yet created the MPTDB database, follow the instructions found in "Creating the Migration Planning Tool Database" before you run the MPTREPT program.

You use the MPTREPT program (`MPTREPT.MPT.TELESUP`) to generate six types, or levels, of migration planning reports in addition to a general information summary generated each time a report is printed. The first levels are general summaries about the system and might be used to gain a general understanding of the migration issues involved. Succeeding levels of reporting are designed to further isolate the reported migration events identified in files. The following is a short description of the six levels of reports.

- Report Level One (System Summary) is a two page system summary report showing a global summary of migration issues on the system.
- Report Level Two (Account Summary) has one line per account on the system. It is used to determine if certain accounts will require more attention during the migration than others.
- Report Level Three (Group Summary) has one line per group on the system. It further identifies groups

within accounts that might require more attention than others.

- Report Level Four (File Summary) lists each file that was flagged as requiring migration attention . The number and severity of migration events is shown so you can pinpoint specific files which need attention during migration.
- Report Level Five (File Detail) has the same information as Report Level Four plus a brief description of each migration event found in the file.
- Report Level Six (Event Summary) lists each migration event detected and the files in which MPT located the event.

Follow the steps outlined below to generate MPT reports. Not all steps are required, but should be considered.

1. **Select a User Options Parameter File:** If you are using the `MPTOPTNS` parameter file (either your own, or the file `MPTOPTNS.MPT.TELESUP` supplied with MPT) to define migration options for disk space considerations, you must use the `:FILE` command to equate the MPT formal file designator `MPTOPTNS` to your parameter file. If you do not want user options defined, you must equate `MPTOPTNS` to `$NULL`. For example, the following file equation equates the MPT formal designator `MPTOPTNS` to your group and account:

```
:FILE MPTOPTNS=MPTOPTNS.MYGROUP.MYACCT
```

2. **Specify an Output Device:** To direct MPTREPT output to a device other than $STDLIST, use the `:FILE` command to equate the formal file designator MPTLIST to the output device of your choice, printing 132-character columns. For example, the following file equation equates MPTLIST to an HP2680 printer and selects an environment file that allows 132 column printing:

```
:FILE MPTLIST;DEV=HP2680;ENV=LP2.HPENV.SYS
```

3. **Include Deleted Files in Reports:** If you want to include in Migration Planning Reports (Levels Four and Five) those MPE V/E files normally deleted from migration planning reports, set the `SHOWDELETE` Job Control Word to a value of one, for example:

```
:SETJCW SHOWDELETE=1
```

4. **Run the Report Generation Program:** Run the `MPTREPT.MPT.TELESUP` program to generate migration planning reports from the data collected in the MPTDB database. Below are options you can select

through the :RUN command to customize the report output generated by MPTREPT are:

,SAVEPAPER	Eliminates page breaks at each major break in the report. (The default is a page break at major report breaks.)
;PARM=*page_length*	Selects the maximum number of lines printed on each page. The default is 60 lines per page.
;INFO=*custom_report_title*	Prints the selected character string at the top of each report page. The supplied string can be a maximum of 60 characters.

The following example executes MPTREPT, specifying all optional parameters:

```
:RUN MPTREPT.MPT.TELESUP,SAVEPAPER;PARM=80;INFO="ACME FINANCE REPORT"
```

4.1. Select Filesets to Include in Reports: The MPTREPT program first prints a product banner to $STDLIST, then asks that you select the files to be included in the report, giving directions for including and excluding filesets from reports. At the following prompt, enter the filesets you want included (or excluded) from migration planning reports:

```
ENTER FILESET:
```

You can respond with one valid MPE fileset per line. You can enter up to 100 filesets to be included and 100 filesets to be excluded (in any order). Exclude a fileset by preceding the fileset with a minus (-) sign. Pressing Return at the first prompt includes all files (`@.@.@`) in the report. Pressing Return at any other prompt ends fileset selection. Following are examples of valid responses to the `ENTER FILESET` prompt of MPTREPT:

```
@.@.GAMES
@.PUB.@
K#######          (translates to K#######.@.@)
-@.PUB.SYS         (do NOT include files in PUB.SYS)
-ERRORLOG.@.@
```

File selection is done once for all reports and remains unchanged for one run of the reporting program.

4.2. Select a Report Level: After all files have been extracted you will be asked to select a migration planning report level (1 through 6). At the following prompt, enter the level of the migration planning report you want:

```
ENTER THE LEVEL OF DETAIL YOU DESIRE (1-6)
```

Selecting a report outside this range (0 or 7, for example) or pressing Return terminates the program.

Report levels 1 through 5 all use a common data extraction from the database, but report level 6 must extract its own data. If you desire only report 6 then you can save time by pressing (CTRL) (Y) during the first data extraction (before choosing a report) and terminating that extraction early. When report 6 is chosen it will extract its own data each time it runs. If you select report level 4 or above (file summary, file detail, or event summary), you will be further prompted to see if you wish to restrict the report by event severity. (Refer to 4.3 below.)

You are repeatedly asked to select a migration planning report level until you respond by pressing a (Return). At this point the program terminates.

As you select the report, it will be generated in the MPTLIST file.

4.3. Restrict Reported Events: If you select report level 4 or above (file summary, file detail or event summary), you are further prompted to restrict the report by event severity . If you want to see all files, respond `NO` or press (Return) at the following prompt:

```
DO YOU WANT TO RESTRICT REPORTED EVENTS (Y/N)?
```

If you respond to the prompt with `Y(ES)`, MPTREPT prompts you to select which levels of severity will be reported both for CM migration events found in files you are migrating to CM, as well as for NM migration events in files you are migrating to NM:

```
ENTER LEVEL FOR COMPATIBILITY MODE (0-4)
ENTER LEVEL FOR NATIVE MODE (0-4)
```

The following are valid responses to the above prompts:

- 0 = report all events.
- 1 = report only POSSIBLE, WARNING, and ERROR events.
- 2 = report only WARNING and ERROR events.
- 3 = report only ERROR events.
- 4 = suppress all events.

Evaluating Migration Planning Reports

Migration planning reports generated by the MPTREPT program provide information that will assist you in deciding upon the MPE/iX execution mode best suited for migrated MPE V/E program and SL files. MPE/iX provides two major modes of program execution: CM and NM. A third mode of program execution is provided through object code translation of MPE V/E program code. For example, you may want to run a particular set of applications in CM for immediate productivity on the new system. You may want another set of applications to be immediately migrated to NM in order to fully utilize the features of MPE/iX and HP-PA.

MPT attempts to classify program files and generic ASCII data files according to the programming language they contain. This is a subjective classification which should be used as general guidance only. It is possible for this package to incorrectly classify the language of a program or source file.

Migration Events

MPT detects and reports on incompatibilities found in selected MPE V/E files. Incompatibilities are separated into individual migration events for clarity. Each migration event is defined as the presence or use of certain features of MPE V/E that are known to require attention when migrating those files to an MPE iX-based system.

The first look at MPT migration planning reports may seem to indicate that migration will be extremely difficult (due to the number of events reported). This is not usually the case. Many migration events are handled simply and quite a few can be dismissed. Remember, MPT gives you too much information rather than too little.

Not every reported event will cause difficulty in migrating to MPE/iX. Some events cause problems

only when going into NM. Others are, by their nature, uncertain in their difficulty. The design philosophy of MPT is to report any problem, even if the chances of it causing a problem are very slight. In this way the user can check the event at their leisure before migrating to MPE/iX.

A particular event may function properly when run in CM but not in NM. For this reason each event will be classified into one of the four severity levels both for CM and for NM. In general, the NM severity level will be at least as severe as CM and often more severe. If an application shows few or no problems in CM, but many severe problems in NM, you might decide that migrating this application to NM might not be warranted.

Severity Level of Migration Event

In order to help you decide which migration events are minor and which are major, each migration event is assigned one of four severity levels, described below:

OK	No problems exist when using this feature.
POSSIBLE	A potential problem exists, but to be certain, you require more information than MPT can obtain. For example, we know that the new operating system will not use DRT numbers to address its hardware devices. If a system intrinsic is called to return the DRT number on MPE/iX, then a zero will be returned. This will not abort the offending program, nor return an error condition. The potential problem arises based on what the application program actually does with the DRT once it gets it. POSSIBLE events might be no problem

at all, but they should be examined to be sure if migration is to be smooth.

WARNING — These events will not work on the new system without some modification. There is a work around or alternate method to accomplish the same task, but it might involve minor rework of the application. (For example, the `:SYSDUMP` command has been eliminated. You must use the `SYSGEN` utility instead. Also, the commands to SYSGEN are different from the interaction with `:SYSDUMP` . This is a correctable situation, but it won't work properly without changes).

ERROR — These events indicate that a feature is being used which is not supported on the MPE/iX system. There may be a work around available but it usually involves a significant amount of rework to the application. Examples include the calling of the `PTAPE` intrinsic, using the now obsolete DEL package, or calling undocumented MPE internal routines.

Examining File Types for Migration Events

The migration events detected in MPE V/E files may be further divided into general categories. Some migration events will apply only to compiled program files (such as the calling of certain intrinsic routines) while others relate mainly to batch jobs and UDC files. The current migration event types detected are:

PROG — Program file events. These events are discovered by examining compiled program files which are found on the system. Certain program information is

obtained for information purposes (for example, stack size and code segment data) and will be included in the reports. Incompatibilities are generally caused by the program calling certain external procedures (or intrinsics) or by its use of Privileged Mode capability.

This category will result in a lot of POSSIBLE events, since it can detect that a procedure is called, but not what parameters are passed to it or what the application does with the results of that call.

SL — Segmented Library (SL) events. These are the same events as for compiled program files, except that they were found in a Segmented Library rather than a program. SL files are included in the statistics gathered concerning the average size program and the number of program files with Privileged Mode.

RTM — Run Time Monitor events. In order to obtain more certain information about how the procedures are being used it was necessary to detect them dynamically during program execution. RTM inserts special routines into the system which detect when specific routines are called by a program.

Since RTM is used while the programs are running, parameters that are passed are examined by the interceptor routine. If a problematic combination is detected, a special record is written to the MPE log files. This record can be analyzed at a later time by reading those log files.

RTM can be more specific about an event than PROG events since it at least knows the parameters being passed. It still can't detect how an application uses the results of the routine so even many RTM events might not be actual problems. You may notice that the same event will be recorded as PROG and RTM. PROG says the application could call the routine in an incompatible way; RTM says it actually did.

Note that RTM detects events only as they occur. If a program is never run or is run but never executes a specific routine call, RTM can't detect incompatible events. For this reason MPT uses both program scanning (PROG) and RTM events in its reports.

JOB — Job stream events. These events are found by examining all job stream files on the system. (Job streams are ASCII files that contain a list of commands to be executed in batch mode). Each job stream is scanned for incompatible or changed MPE commands. If one is found, a JOB event is logged.

UDC — ASCII files are examined to see if they might be UDC files. Each UDC file is scanned for incompatible or changed MPE commands. If one is found, a UDC event is logged.

Selecting a Report Level

The MPTREPT program allows you to generate six levels of migration planning reports and a general information summary that is generated each time a report is printed. The first levels are general summaries about the system and might be used to gain a general understanding on the migration issues involved. Succeeding levels of reporting are designed to further isolate the reported migration events identified in files.

The sections that follow provide detailed descriptions of both the format and the contents of each of MPTREPT reports summarized below:

- General Information Summary is printed each time the MPTREPT program is run. This one-page report lists the parameters affecting the run and the overall disk capacity (before and after migration to MPE/iX).
- System Summary Report Level One is a two-page report showing a global summary of migration issues on the system. Use this report level to get an initial indication of what major migration issues exist on the system being analyzed.
- Account Summary Report Level Two has one line per account on the system. It can be used to determine if certain accounts will require more attention during the migration than others. For example, if the majority of migration issues are in an account you don't plan to migrate, there is little cause for concern. On the other hand, if an abnormally high number of issues are found in accounts you absolutely need to migrate to NM, you may wish to earmark that account for a detailed analysis using reporting levels four and five, as well as running OCA on the files in that account.
- Group Summary Report Level Three has one line per group on the system. It can further identify groups within accounts that might require more attention than others.

- File Summary Report Level Four lists each file that was flagged as requiring migration attention. The number and severity of migration events is shown so you can pinpoint specific files which need attention during migration. You can minimize the relatively large amount of output generated by this level of report by excluding those accounts or groups you are not planning to migrate to an MPE/iX- based system.
- File Detail Report Level Five has the same information as File Summary Report Level Four plus a line listing a brief description of each migration event found in the file. You can minimize the relatively large amount of output generated by this level of report by excluding those accounts or groups you are not planning to migrate to an MPE/iX- based system.
- "Event Summary Report Level Six" lists each migration event detected and the files in which MPT located the event.

Note

The disk space considerations included in MPT reports pertain only to permanent disk files and do not include the additional requirements for services such as virtual memory and spool files. The figures should be considered only as a minimum amount of additional disk space that will be required when migrating the selected files.

General Information Summary

A General Information Summary is printed each time the MPTREPT program is run. Figure I-3 is a sample of a General Information Summary. The General Information Summary lists the program run parameters MPTREPT uses when generating reports, including those you specified in the MPTOPTNS parameter file. Overall disk space considerations are also listed. Table I-2 describes the information returned in the MPTREPT General Information Summary.

```
1}  MPT/3000 A.01.04 REPORT (GENERAL INFORMATION)  ** Acme Rentals Inc. 05/10/88     ** Page 1
2}
3}  19213     FILES WERE SELECTED
4}  1         FILES WERE ADDED FOR MPE XL (   166.400 MEGABYTES)
5}  2067      FILES WERE DELETED          (   111.863 MEGABYTES)
6}  THE MPE V/E SYSTEM REQUIRED AT LEAST   2197.741 MEGABYTES OF DISC SPACE
7}  THE MPE XL SYSTEM WILL REQUIRE AT LEAST   2292.903 MEGABYTES OF  DISC SPACE
8}
9}  FOR A GROWTH RATE OF     95.162 MEGABYTES OR     4.3%
10}
11} THE FOLLOWING FILESETS WILL BE INCLUDED IN THE REPORT:
12}   @.@.@
13}
14} THE FOLLOWING FILESETS WILL BE EXCLUDED FROM THE REPORT:
15}  @.@.PATCH
16}  @.@.@NMC
17}
18} THE MIGRATION OPTIONS FOR THIS REPORT ARE :
19} 0  DELETE     ;FNAME='@.@.HPPL##                '
.   1  DELETE     ;FNAME='@.@.SUPPORT               '
.   2  DELETE     ;FNAME='@.@.TELESUP               '
.   3  DELETE     ;FNAME='ERRORLOG.@.@              '
    4  DELETE     ;FNAME='LOG####.PUB.SYS           '
    5  DELETE     ;CLASS='NLOG    '
    6  USE CM     ;CLASS='SL      ' ;LANGUAGE='SPL     '
    7  USE CM     ;FCODE= 1029 ;LANGUAGE='SPL     '
    8  USE NM     ;FCODE= 1029 ;LANGUAGE='COBOL   '
    9  USE NM     ;FCODE= 1029 ;LANGUAGE='PASCAL  '
   10  USE NM     ;FCODE= 1029 ;LANGUAGE='FTN77   '
   11  USE CM     ;CLASS='IMAGE   '
```

LG200144_002

Figure I-3. Sample MPTREPT General Information Summary

Table I-1. MPTREPT General Information Summary Description.

LINE#	DESCRIPTION
1	An MPT banner that identifies the report level printed on the current page (in this example, "GENERAL INFORMATION"), a custom report title you can specify using the `INFO=` parameter of the `:RUN` command (in this example, `:RUN MPTREPT.MPT.TELESUP; INFO= "Acme Rentals Inc."`), and the current date.
3	The total number of file descriptions in the MPTDB database that meet the selection criteria (your responses to the `ENTER FILESET:` prompt of the MPTREPT program).
4	The total number of files that meet the selection criteria but are not located in the MPTDB database because they represent new files that must be included in any report of MPE/iX disk space considerations. This number includes those files you specified in the MPTOPTNS parameter file (`ACTION=ADD`). Also listed is the disk space required by these files.
5	The total number of files that meet the selection criteria but are deleted from these MPT reports. This number includes those files you specified in the MPTOPTNS parameter file (`ACTION=DELETE`). Also listed is the disk space required by these files.
6	The MPE V/E disk space requirements for all files that met the selection criteria (including those that were deleted from the report, described in Line 5 above).
7	The estimated MPE/iX disk space requirements for all files selected (including those that were added to the report, described in Line 4 above). This estimate is computed for selected files using the chosen migration modes you specified in the MPTOPTNS file, either CM, NM, or TM. (The default migration mode is CM.)
9	The estimated growth rate indicated by the amount of additional disk space required by the selected (and `ADD`ed) files when they are migrated to MPE/iX. The percentage is this value compared against the original MPE V/E value.
11 ... 16	The selection criteria you specified in response to the `ENTER FILESET:` prompt of the MPTREPT program. The list includes both the files you included and the files you excluded from the report.
18 ... *n*	Migration options applicable to this run of the MPTREPT program. This list includes those migration options you specified in the MPTOPTNS parameter file, including the chosen action to take (for example, `DELETE, USE CM, or` [illegible] for the indicated filesets.

System Summary Report Level One

The System Summary Report Level One displays global migration information that may be present on your system. This report is printed each time you specify the value 1 in response to the `ENTER THE LEVEL OF DETAIL YOU DESIRE (1-6)` prompt of the MPTREPT program.

The System Summary Report is a three-part report:

- Part One displays total numbers of migration events sorted by the various file types recognized by MPT. Figure I-4 is a sample of Part One of a System Summary Report Level One. Table I-2 describes the information returned in Part One.
- Part Two displays general information about the SL and program files examined in the report. Figure I-5 is a sample of Part Two of a System Summary Report Level Two. Table I-3 describes the information returned in Part Two.
- Part Three displays disk space requirements for each of the migration options sorted by both MPE V/E file codes and special codes interpreted by MPT. Figure I-6 is a sample of Part Three of a System Summary Report Level Three. Table I-4 describes the information returned in Part Three.

Part One Description. Part One of the System Summary Report Level One summarizes the number of migration events, logged for both CM and NM according to file types and languages.

MPT/3000 A.01.04 REPORT LEVEL 1 (SYSTEM SUMMARY) ** 05/10/88 ** Page 1

LANGUAGE	SL & PROGRAM FILES	V-MEM(MB)	SOURCE FILES FILES	# LINES	COMPATIBILITY MODE POSS	WARN	ERROR	NATIVE MODE POSS	WARN	ERROR
BASIC	173	4.490	0	0	31	3	0	39	123	1
COBOL	74	4.974	60	12,542	52	4	1	70	19	2
DBSCHEMA	0	.000	57	7,270	0	0	0	0	0	0
FORTRAN	181	15.039	29	23,994	330	12	0	415	15	5
PASCAL	244	15.106	466	146,818	221	63	7	333	150	38
RPG	19	.510	2	668	9	0	0	12	3	0
SPL	1740	157.101	708	474,896	1699	669	76	2133	1296	345
STREAM	0	.000	1537	159,401	258	331	12	258	744	12
TDP FILE	0	.000	2524	492,670	0	0	0	0	0	0
TRANSACT	7	.000	65	8,007	0	0	0	0	0	0
UDC	0	.000	215	35,730	81	284	39	81	330	39
UNCLASS	22	.286	21	172,534	16	2	0	20	11	0

LG200144_003

Figure I-4. Sample System Summary Report Level One (Part One)

Table I-2. System Summary Report Level One Description

COLUMN	DESCRIPTION(Part One).	
LANGUAGE	The programming language for source, SL, and program files (for example, COBOL, FORTRAN, Pascal, or SPL). The following codes indicate further classification of the file in some cases, including:	
	`SOURCE`	Unclassifiable program source files.
	`STREAM`	ASCII files containing a job stream.
	`TDP FILE`	Unprocessed TDP document files.
	`UDC`	What MPT has interpreted to be ASCII UDC files.
	`AUTO`	IMAGE automatic master data sets.
	`DATASET1`	IMAGE files containing a data set.
	`DBSCHEMA`	IMAGE schema files.
	`DETAIL`	IMAGE detail data sets.
	`ILR`	IMAGE files used for Intrinsic Level Recovery.
	`MANUAL`	IMAGE manual master data sets.
	`REMOTE`	IMAGE remote database access files.
	`ROOT`	IMAGE root files.
	blank	The file does not fit any recognized category.
SL & PROGRAM FILES	The number of SL and program files found for this language.	
SL & PROGRAM VMEM(MB)	The size of virtual memory (code plus data stacks) for all program files and SLs in this language.	
SOURCE FILES FILES	The number of source files found for this language.	
SOURCE FILES # LINES	The total number of lines (records) for all source files in this language.	
COMPATIBILITY MODE POSS	The total number of CM migration events classified as "POSSIBLE" identified in these files.	
COMPATIBILITY MODE WARN	The total number of CM migration events classified as "WARNING" identified in these files.	
COMPATIBILITY MODE ERROR	The total number of CM migration events classified as "ERROR" identified in these files.	
NATIVE MODE POSS	The total number of NM migration events classified as "POSSIBLE" identified in these files.	
NATIVE MODE WARN	The total number of NM migration events classified as "WARNING" identified in these files.	
NATIVE MODE ERROR	The total number of NM migration events classified as "ERROR" identified in these files.	

Part Two Description. Part Two of the System Summary Report Level One displays a number of summary lines in a noncolumnar format. Figure I-5 is an sample of the second part of a System Summary Report Level One. Table I-3 describes the information returned in this part of the System Summary.

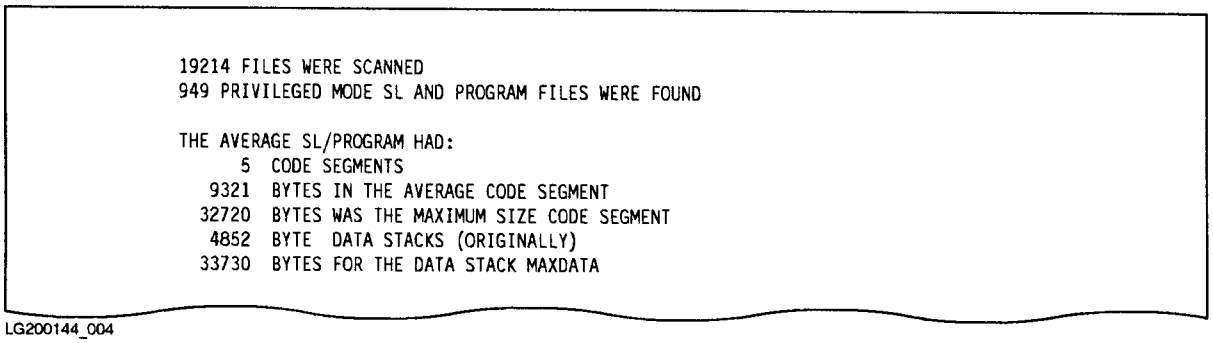


Figure I-5. Sample System Summary Report Level One (Part Two)

Table I-3. System Summary Report Level One Description (Part Two)

LINE#	DESCRIPTION
1	The total number of disk file descriptions included in this report (both those files examined in the MPTDB database and those files `ADD`ed to the report).
2	The total number of SL and program files assigned Privileged Mode (PM) capability.
5	The average number of code segments obtained by averaging the number of code segments in each SL and program file on the system.
6	The number of bytes in an average-sized code segment. Divide this number by two to get 16-bit (MPE V/E) words. Divide by four to get 32-bit (MPE/iX) words.
7	The largest code segment found in the SL and program files examined.
8	The average initial stack size that the average program starts as. The stack can usually be expanded as needed, up to the MAXDATA size.
9	The average MAXDATA size of the program files examined.

Part Three Description. Part Three of the System Summary Report Level One displays disk space requirements for each of the migration options sorted by both MPE V/E file codes and special codes interpreted by MPT. Figure I-6 is an sample of the third part of a System Summary Report Level One. Table I-4 describes the information returned in this part of the System Summary.

MPT/3000 A.01.04 REPORT LEVEL 1 (SYSTEM SUMMARY) ** 05/10/88 ** Page 2

DISC FILE SPACE OPTIONS: PERCENT ALL VALUES ARE MEGABYTES

TYPE	LANGUAGE	COUNT	OF TOTAL	CHOSEN	MPE-V	MPE-XL(CM)	MPE-XL(NM)	MPE-XL(OT)	MPE-XL(TM)	GROWTH
ANODE		1	.0%	.098	.097	.098	.098	.098		1.3%
ASCII	STREAM	1537	.8%	18.039	16.071	18.039	18.039	18.039		12.2%
ASCII	TDP FILE	2524	2.2%	50.047	46.816	50.047	50.047	50.047		6.9%
ASCII	TRANSACT	65	.0%	.848	.765	.848	.848	.848		10.9%
ASCII	UDC	215	.2%	3.566	3.291	3.566	3.566	3.566		8.4%
BASD		3	.0%	.671	.667	.671	.671	.671		.6%
BINARY		1938	50.7%	1161.781	1159.300	1161.781	1161.781	1161.781		.2%
BSAVE	BBASIC	1	.0%	.005	.004	.005	.005	.005		33.3%
CHARD		7	.0%	.271	.262	.271	.271	.271		3.4%
DIU		6	.1%	1.582	1.574	1.582	1.582	1.582		.5%
DRAW		264	.3%	6.083	5.745	6.083	6.083	6.083		5.9%
DSTOR		9	.0%	.488	.476	.488	.488	.488		2.4%

LG200144_005

Figure I-6. Sample System Summary Report Level One (Part Three)

Table I-4. System Summary Report Level One Description (Part Three).

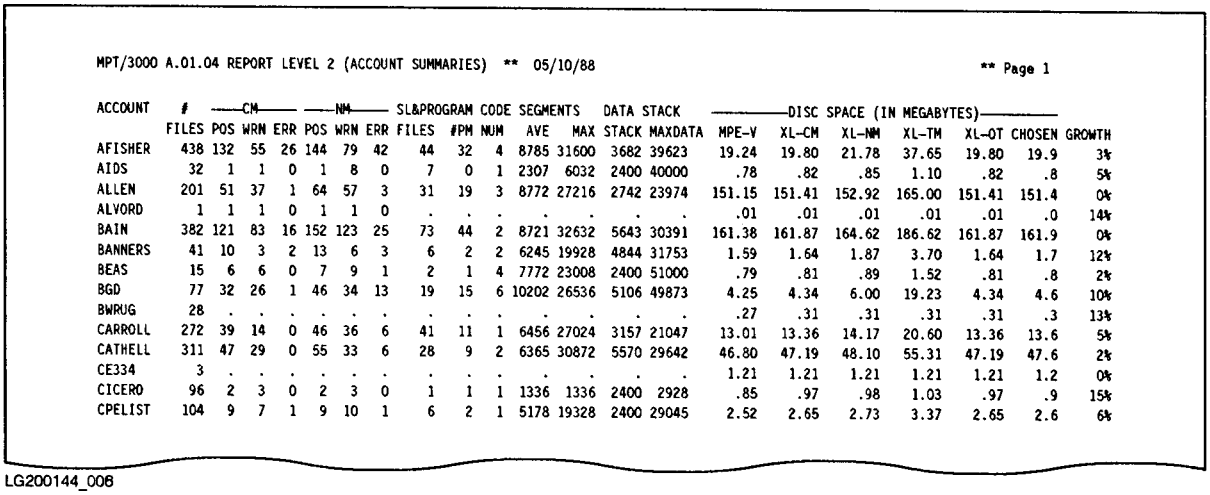
COLUMN	DESCRIPTION	
TYPE	The general MPE file type. This mnemonic comes from the file code (returned in `LISTF,2` command). If the file code is zero, the following special interpretations are obtained by the analysis program:	
	`ASCII`	General character files.
	`BINARY`	General binary data files.
	`CIR`	Files of type Circular.
	`MSG`	Files of type Message.
	`RIO`	Files of type Relative Input/Output.
	`KSAM`	Files of type KSAM.
LANGUAGE	The programming language for source, SL, and program files (for example, COBOL, FORTRAN, Pascal, or SPL). The following codes indicate further classification of the file in some cases, including:	
	`SOURCE`	Unclassifiable program source files.
	`STREAM`	ASCII files containing a job stream .
	`TDP FILE`	Unprocessed TDP document files.
	`UDC`	What MPT has interpreted to be ASCII `UDC` files.
	`AUTO`	IMAGE automatic master data sets.
	`DATASET1`	IMAGE files containing a data set.
	`DBSCHEMA`	IMAGE schema files.
	`DETAIL`	IMAGE detail data sets.
	`ILR`	IMAGE files used for Intrinsic Level Recovery.
	`MANUAL`	IMAGE manual master data sets.
	`REMOTE`	IMAGE remote database access files.
	`ROOT`	IMAGE root files.
	blank	The file does not fit any recognized category.

Table J-4. System summary Report Level One Description (Part Three).

COUNT	The number of disk files of this type examined in this report.
PERCENT OF TOTAL	The percentage of the total disk space used that the disk files of this type will occupy on the MPE/iX-based system (based upon the migration option you selected in the MPTOPTNS parameter file). CM is the default.
CHOSEN	The disk space requirements for the disk files of this type based upon the migration option you selected in the MPTOPTNS parameter file. CM is the default.
MPE-V	The MPE V/E disk space requirements for the disk files of this type.
MPE/iX(CM)	The estimated disk space requirements if the disk files of this type were migrated to an MPE/iX-based system and left in CM.
MPE/iX(NM)	The estimated disk space requirements if the disk files of this type were migrated to an MPE/iX-based system, then converted into NM. For example, program and SL files would be recompiled using NM compilers.
MPE/iX(OT)	Disregard this information. This column is reserved for expansion of the MPT tool to allow for the possibility of future migration paths.
MPE/iX(TM)	The estimated disk space requirements if the disk files of this type were object code translated using the OCT utility. This column is printed only if these files are program or SL files.
GROWTH	The relative growth of disk space usage for all disk files of this file type. This value represents the amount of additional disk space required when migrating these files to MPE/iX using the chosen migration option you identified in the MPTOPTNS parameter file. (Default chosen is CM.) A growth of 0% means that the files will occupy the same disk space on an MPE/iX-based system as they occupied on an MPE V/E-based system. A growth of 100% means the files will double in size. A negative growth indicates that the files will occupy less space on MPE/iX than they did on MPE V/E.

Account Summary Report Level Two

The Account Summary Report Level Two breaks down the migration information into account levels. The information returned in the Account Summary Report enables you to determine accounts that may be responsible for the majority of migration events or disk space. This report is printed each time you specify the value 2 in response to the ENTER THE LEVEL OF DETAIL YOU DESIRE (1-6) prompt of the MPTREPT program. Figure I-7 is a sample of a portion of an Account Summary Report. Table I-5 describes the information returned in the MPTREPT Account Summary Report Level Two.



MPT/3000 A.01.04 REPORT LEVEL 2 (ACCOUNT SUMMARIES) ** 05/10/88 ** Page 1

ACCOUNT	#	CM			NM			SL&PROGRAM CODE SEGMENTS					DATA STACK		DISC SPACE (IN MEGABYTES)					
	FILES	POS	WRN	ERR	POS	WRN	ERR	FILES	#PM	NUM	AVE	MAX	STACK	MAXDATA	MPE-V	XL-CM	XL-NM	XL-TM	XL-QT	CHOSEN GROWTH
AFISHER	438	132	55	26	144	79	42	44	32	4	8785	31600	3682	39623	19.24	19.80	21.78	37.65	19.80	19.9 3%
AIDS	32	1	1	0	1	8	0	7	0	1	2307	6032	2400	40000	.78	.82	.85	1.10	.82	.8 5%
ALLEN	201	51	37	1	64	57	3	31	19	3	8772	27216	2742	23974	151.15	151.41	152.92	165.00	151.41	151.4 0%
ALVORD	1	1	1	0	1	1	0	.	.	.	.	.	.	.	.01	.01	.01	.01	.01	.0 14%
BAIN	382	121	83	16	152	123	25	73	44	2	8721	32632	5643	30391	161.38	161.87	164.62	186.62	161.87	161.9 0%
BANNERS	41	10	3	2	13	6	3	6	2	2	6245	19928	4844	31753	1.59	1.64	1.87	3.70	1.64	1.7 12%
BEAS	15	6	6	0	7	9	1	2	1	4	7772	23008	2400	51000	.79	.81	.89	1.52	.81	.8 2%
BGD	77	32	26	1	46	34	13	19	15	6	10202	26536	5106	49873	4.25	4.34	6.00	19.23	4.34	4.6 10%
BWRUG	28	.	.	.	.	.	.	.	.	.	.	.	.	.	.27	.31	.31	.31	.31	.3 13%
CARROLL	272	39	14	0	46	36	6	41	11	1	6456	27024	3157	21047	13.01	13.36	14.17	20.60	13.36	13.6 5%
CATHELL	311	47	29	0	55	33	6	28	9	2	6365	30872	5570	29642	46.80	47.19	48.10	55.31	47.19	47.6 2%
CE334	3	.	.	.	.	.	.	.	.	.	.	.	.	.	1.21	1.21	1.21	1.21	1.21	1.2 0%
CICERO	96	2	3	0	2	3	0	1	1	1	1336	1336	2400	2928	.85	.97	.98	1.03	.97	.9 15%
CPELIST	104	9	7	1	9	10	1	6	2	1	5178	19328	2400	29045	2.52	2.65	2.73	3.37	2.65	2.6 6%

LG200144_008

Figure I-7. Sample Account Summary Report Level Two

Table I-5. Account Summary Report Level Two Description.

COLUMN	DESCRIPTION
ACCOUNT	The account name.
# FILES	The number of files selected in this account.
COMPATIBILITY MODE POS	The total number of CM migration events classified as "POSSIBLE" identified in the selected files in this account.
COMPATIBILITY MODE WRN	The total number of CM migration events classified as "WARNING" identified in the selected files in this account.
COMPATIBILITY MODE ERR	The total number of CM migration events classified as "ERROR" identified in the selected files is this account.
NATIVE MODE POS	The total number of NM migration events classified as "POSSIBLE" identified in the selected files in this account.
NATIVE MODE WRN	The total number of NM migration events classified as "WARNING" identified in the selected files in this account.
NATIVE MODE ERR	The total number of NM migration events classified as "ERROR" identified in the selected files in this account.
SL & PROGRAM FILES	The number of selected program and SL files located in this account.
SL & PROGRAM # PM	The number of selected program and SL files assigned Privilege Mode capability located in this account.
CODE SEGMENTS NUM	The average number of code segments obtained by averaging the number of code segments in each selected SL and program file in this account.
CODE SEGMENTS AVE	The number of bytes in an average-sized code segment. Divide this number by two to get 16-bit (MPE V/E) words. Divide by four to get 32-bit (MPE-iX) words.
CODE SEGMENTS MAX	The largest code segment (in bytes) found in the selected SL and program files in this account.

Table J-5. Account summary Report Level Two Description. (Cont.)

DATA STACK STACK	The average initial stack size (in bytes) for all selected programs in this account.
DATA STACK MAXDATA	The average MAXDATA size (in bytes) for all selected program files in this account.
DISC SPACE MPE-V	The MPE V/E disk space requirements (in megabytes) for all selected files in this account.
DISC SPACE MPE-iX(CM)	The estimated disk space requirements (in megabytes) if the selected files in this account were migrated to an MPE/iX-based system and left in CM.
DISC SPACE MPE/iX(NM)	The estimated disk space requirements (in megabytes) if the selected files in this account were migrated to an MPE/iX-based system, then converted into NM. For example, program and SL files would be recompiled using MPE/iX NM compilers.
DISC SPACE	The estimated disk space requirements (in megabytes) if the selected files in this MPE/iX(TM) account were object code translated using the OCT utility. This column is printed only for selected program and SL files.
DISC SPACE	Disregard this information. This column is reserved for expansion of the MPE-iX(OT) MPT product to allow for the possibility of future migration paths.
DISC SPACE CHOSEN	The disk space requirements (in megabytes) for the selected files in this account based upon the migration option you selected in the MPTOPTNS parameter file. (CM is the default.)
DISC SPACE GROWTH	The relative growth (in megabytes) of disk space usage for the selected files in this account. This value represents the amount of additional disk space required when migrating these files to MPE/iX using the "Chosen" migration option you identified in the MPTOPTNS parameter file. (The default "Chosen" is CM.) A growth of 0% means that the files will occupy the same disk space on an MPE/iX-based system they occupied on an MPE V/E-based system. A growth of 100% means the files will double in size. A negative growth indicates that the files will occupy less space on MPE/iX than they did on MPE V/E.

Group Summary Report Level Three

The Group Summary Report Level Three is quite similar to the Account Summary Report Level Two, except that it summarizes by group rather than by account. This report is printed each time you specify the value 3 in response to the `ENTER THE LEVEL OF DETAIL YOU DESIRE (1-6)` prompt of the MPTREPT program. Figure I-8 is a sample of a portion of a Group Summary Report Level Three.

The account name is given within the box of asterisks before each list of groups associated with that account. The columns in the Group Summary Level Three Report have exactly the same meaning as those in the Account Summary Level Two report, except that they summarize all the files in a group rather than in an account. Please refer to the description of the columns in Table I-5 for details.

```
MPT/3000 A.01.04 REPORT LEVEL 3 (GROUP SUMMARIES)  **  05/10/88                                           ** Page 1

*************
* AFISHER  *
*************

GROUP      #   ----CM----- ----NM----- SL&PROGRAM CODE SEGMENTS   DATA STACK   ----------DISC SPACE (IN MEGABYTES)----------
        FILES POS WRN ERR POS WRN ERR FILES  #PM NUM  AVE   MAX STACK MAXDATA MPE-V  XL-CM  XL-NM  XL-TM  XL-OT CHOSEN GROWTH
BULL        5   .   .   .   .   .   .    .    .   .    .     .     .       .   .02    .03    .03    .03    .03     .0    27%
CPE0831    60   .   .   .   .   .   .    .    .   .    .     .     .       .  1.08   1.15   1.15   1.15   1.15    1.1     7%
CPELIST    24   .   .   .   .   .   .    .    .   .    .     .     .       .   .37    .40    .40    .40    .40     .4     8%
FICHE      22   3   0   0   4   0   0    3    0   1 3238  8080  6474   39973   .27    .29    .32    .54    .29     .3    14%
LETTERS     1   .   .   .   .   .   .    .    .   .    .     .     .       .   .01    .01    .01    .01    .01     .0    17%
PNOTES     30   .   .   .   .   .   .    .    .   .    .     .     .       .   .17    .21    .21    .21    .21     .2    23%
PUB        37   9   4   1   9   7   1    4    2   1 6635 19328  2400   42218  1.55   1.60   1.68   2.29   1.60    1.6     4%
R000        3  10   4   3  11   4   5    3    3   2 7454 19216  2400   32000   .08    .09    .17    .83    .09     .0     5%
R001        6  11   4   3  12   4   5    3    3   2 7468 19328  2400   32000   .21    .21    .30    .96    .21     .2     4%
R002        6  11   4   3  12   4   5    3    3   2 7475 19328  2400   32000   .11    .12    .20    .86    .12     .1     7%
R004        5  10   4   3  11   4   5    3    3   2 7536 19712  4000   32000   .11    .12    .20    .87    .12     .1     6%
```

LG200144_007

Figure I-8. Sample Group Summary Report Level Three

File Summary Report Level Four

The File Summary Report Level Four provides summary information for each file selected for the report. This report displays the event number and severity level of all migration events located in each of the selected files. You can use the information returned in this report to pinpoint the files that need attention during migration.

This report is printed each time you specify the value 4 in response to the `ENTER THE LEVEL OF DETAIL YOU DESIRE (1-6)` prompt of the MPTREPT program. Figure I-9 is a sample of a portion of a File Summary Report Level Four. Table I-6 summarizes the information returned in the File Summary Report Level Four.

In the report, each group is identified in the box of asterisks preceding the list of files (`GROUP.ACCOUNT`). At the end of the list of files associated with each group is a group summary. The columns in this single line have meanings identical to those in the Account Summary Report Level Two, described in Table I-5.

```
MPT/3000 A.01.04 REPORT LEVEL 4 (FILE SUMMARIES)  **  05/10/88                                             ** Page 1

*********************
* BULL    .AFISHER  *
*********************

                          ---FILE SIZE--- ----CM----- ----NM----- PRV ---CODE SEGS--- -----DATA----- MPE-XL DISC SPACE (SECTORS)
FILENAME TYPE    LANGUAGE RECORDS SECTORS POS WRN ERR POS WRN ERR MODE  #   AVE   MAX STACK MAXDATA --CM-- --NM-- --TM-- --OT-- AC
DEFDATA  ASCII                 18      14  .   .   .   .   .   .        .    .     .    .       .    19     19     19     19
FCOPY    ASCII   TDP FILE      25      18  .   .   .   .   .   .        .    .     .    .       .    23     23     23     23
FLABEL   ASCII   TDP FILE      19      14  .   .   .   .   .   .        .    .     .    .       .    19     19     19     19
PRINTER  ASCII                 26      18  .   .   .   .   .   .        .    .     .    .       .    23     23     23     23
SILMAT   ASCII                 31      30  .   .   .   .   .   .        .    .     .    .       .    35     35     35     35
```

LG200144_008

Figure I-9. Sample File Summary Report Level Four

Table I-6. File summary Report Level Four Description.

COLUMN	DESCRIPTION
TYPE	The general MPE file type. This mnemonic comes from the file code (returned in `:LISTF,2` command). If the file code is zero, the following special interpretations are obtained by the analysis program: `ASCII` General character files. `BINARY` General binary data files. `CIR` Files of type Circular. `MSG` Files of type Message. `RIO` Files of type Relative Input/Output. `KSAM` Files of type KSAM.
LANGUAGE	The programming language for source, SL, and program files (for example, COBOL, FORTRAN, Pascal, or SPL). The following codes indicate further classification of the file in some cases, including: `SOURCE` Unclassifiable program source files. `STREAM` ASCII files containing a job stream. `TDP FILE` Unprocessed TDP document files. `UDC` What MPT has interpreted to be ASCII `UDC` files. `AUTO` IMAGE automatic master data sets. `DATASET1` IMAGE files containing a data set. `DBSCHEMA` IMAGE schema files. `DETAIL` IMAGE detail data sets. `ILR` IMAGE files used for Intrinsic Level Recovery. `MANUAL` IMAGE manual master data sets. `REMOTE` IMAGE remote database access files. `ROOT` IMAGE root files. blank The file does not fit any recognized category.
FILE SIZE RECORDS	The number of records in the file.
FILE SIZE SECTORS	The number of sectors of disk space used by the file on the MPE V/E-based system.

Table J-6. File Summary Report Level Four Description. (Cont.)

COMPATIBILITY MODE POS	The total number of CM migration events classified as "POSSIBLE" identified in this file.
COMPATIBILITY MODE WRN	The total number of CM migration events classified as "WARNING" identified in this file.
COMPATIBILITY MODE ERR	The total number of CM migration events classified as "ERROR" identified in this file.
NATIVE MODE POS	The total number of NM migration events classified as "POSSIBLE" identified in this file.
NATIVE MODE WRN	The total number of NM migration events classified as "WARNING" identified in this file.
NATIVE MODE ERR	The total number of NM migration events classified as "ERROR" identified in this file.
PRV MODE	Indicates if this file contains Privileged Mode (PM) code. This column can contain one of the following codes: (blank) The file does not contain Privilege Mode code. PM The file was assigned PM capability using `:PREP`. P The file contains at least one routine specifying `$CONTROL PRIVILEGED`. PP Both PM and P pertain to this file.
CODE SEGS#	The number of code segments in this file (if applicable).
CODE SEGS AVE	The average size of the code segments (in bytes).
CODE SEGS MAX	The size of the largest code segment (in bytes).
DATA STACK	The initial size of the data stack (in bytes).
DATA MAXDATA	The maximum size the data stack can grow to (bytes).
DISC SPACE MPE/iX(CM)	The estimated disk space requirements (in megabytes) if this file were migrated to an MPE/iX-based system and left in CM.
DISC SPACE MPE/iX(NM)	The estimated disk space requirements (in megabytes) if this file were migrated to an MPE/iX-based system, then converted into NM. For example, program and SL files would be recompiled using MPE/iX NM compilers.
DISC SPACE MPE-iX(TM)	The estimated disk space requirements (in megabytes) if this file were object code translated using the OCT utility. This column if this file were object is printed for only selected program and SL files.
DISC SPACE MPE-iX(OT)	Disregard this information. This column is reserved for expansion of the MPT tool to allow for the possibility of future migration paths.

Table J-6. File Summary Report Level Four Description. (Cont.)

AC	This field indicates when a migration option (action) has been "Chosen" for this file. The default action is to migrate the file into CM and will show up as a blank in this field. The codes are: C Migrate into CM (same as the default) N Migrate into NM T Migrate into object code translated mode O Migrate into the "other" mode (disregard) A This file was ADDED for MPE/iX (It didn't exist under MPE V/E) D This file was DELETED (not migrated from MPE V/E to MPE/iX)

File Detail Report Level Five

The File Detail Report Level Five returns information about the migration events detected for a file, in addition to the information contained in the File Summary Report Level Four (described in Table I-6). This report is printed each time you specify the value 5 in response to the `ENTER THE LEVEL OF DETAIL YOU DESIRE (1-6)` prompt of the MPTREPT program. Figure I-10 is a sample of a portion of a File Detail Report Level Five.

```
MPT/3000 A.01.04 REPORT LEVEL 5 (FILE DETAILS)  **  05/10/88                                                ** Page 5

*********************
* FICHE   .AFISHER  *
*********************

                           --FILE SIZE-- ----CM---- ----NM---- PRV ---CODE SEGS--- ----DATA----- MPE-XL DISC SPACE (SECTORS)
FILENAME TYPE     LANGUAGE RECORDS SECTORS POS WRN ERR POS WRN ERR MODE  #    AVE    MAX STACK MAXDATA --CM-- --NM-- --TM-- --OT-- AC
ANNIE    ASCII    TDP FILE      56      30  .   .   .   .   .   .        .     .      .     .       .    35     35     35     35
FILE     PROGRAM  PASCAL        14      15  .   .   .   .   .   .        1   992    992  8512   60000    20     35    155     20  N
FLOW     ASCII    TDP FILE     232      80  .   .   .   .   .   .        .     .      .     .       .    85     85     85     85
L1       ASCII    TDP FILE      19      15  .   .   .   .   .   .        .     .      .     .       .    20     20     20     20
L10      ASCII    TDP FILE      15      10  .   .   .   .   .   .        .     .      .     .       .    15     15     15     15
L11      ASCII    TDP FILE     145      55  .   .   .   .   .   .        .     .      .     .       .    60     60     60     60
MAKELIST PROGRAM  PASCAL        18      19  1   0   0   2   0   0        1  1880   1880  8512    9920    24     43    195     24  N

         PROG ERR#   200 OCCURRED      1 TIMES CM-POS NM-POS COMMAND INTRINSIC POSS. PROG. EXECUTION OF INCOMPATIBLE CMDS
         PROG ERR#   108 OCCURRED      1 TIMES CM-OK  NM-POS GENMESSAGE INTRINSIC MAY ENCOUNTER POINTER PROBLEMS IN NM
            .
            .
            .

         #    ----CM----  ----NM---- SL&PROGRAM  CODE SEGMENTS   DATA STACK   ----------DISC SPACE (IN MEGABYTES)----------
         FILES POS WRN ERR POS WRN ERR FILES  #PM NUM   AVE   MAX STACK MAXDATA  MPE-V   XL-CM   XL-NM   XL-TM   XL-OT CHOSEN GROWTH
           22   3   0   0   4   0   0     3    0   1  3238  8080  6474 39973      .27     .29     .32     .54     .29     .3    14%
```

LG200144_009

Figure I-10. Sample File Detail Report Level Five

You should recognize the majority of this report if you have examined the File Summary Report Level Four . You might notice that it is the same except for the lines following each file's information. These lines are descriptions of each migration event that was found in the file. The information returned is described below:

- The migration event type and unique identifying number is given (example: PROG ERR# 106). The event type identifies what type of event scanning detected the event. Possible events and their sources are described earlier in this document.
- The ERR# indicates a unique migration event. You may look up the event number in Appendix D, "Incompatibilities" to determine more information about what causes the event, how to tell if it is significant, and what corrective action can be taken if needed.

- The number of times the event occurred is listed. For RTM events, this is the actual number of times the event was detected and logged while RTM was enabled. For most other events, this number will be one. The exception to this rule is where more than one routine can cause the same event message. For example, calling a system routine that is marked as "uncallable" (it requires the calling program to be in Privileged Mode to make the call) will generate a migration event. In this case, the occurrence count will be the actual number of such routines detected in the program (or SL).
- The CM and NM severity levels for the event are listed next. CMOK or NMWRN would indicate that the event causes no problems if migration is to CM (CMOK), but requires some changes in order to migrate to NM (NMWRN).
- A brief description of the event is given. Remember that a more detailed description of each event is located in Appendix D, "Incompatibilities".

Event Detail Report Level Six

The Event Detail Report Level Six is unlike the other reports generated by MPTREPT in that the information in the report is sorted by migration event, not by file, fileset, group, or account. This report lists each migration event that occurred in the selected files, followed by all the files in which that event was found. In this way it should be simple to locate the files which need to be modified. You use this report once you decide that a certain event will need to be corrected for the migration and what steps you can take to correct it.

This report is printed each time you specify the value 6 in response to the `ENTER THE LEVEL OF DETAIL YOU DESIRE (1-6)` prompt of the MPTREPT program. Figure I-11 is a sample of a portion of an Event Detail Report Level Six.

```
MPT/3000 A.01.04 REPORT LEVEL 6 (EVENT DETAILS)  **  05/10/88                    ** Page 2

***************
* JOB  EVENTS *
***************

_____________
! EVENT 2005 ! SHOWCACHE COMMAND NOT SUPPORTED
_____________
Compatibility Mode Severity Level is ERROR
Native        Mode Severity Level is ERROR

Disc Caching is part of MPE XL Transaction Management.  You may
use the System Monitor Tool (SMT) to examine its function.

ACTION: Remove all references to the SHOWCACHE command from the job
        or UDC file.  If you have access to SMT, you may use it to
        report on much of the same information as SHOWCACHE.

ACCOUNT  GROUP    FILE(S)
________ ________ ________
SYS      DUMP     PDUMPMY  SYSINFO
```

LG200144_010

Figure I-11. Sample Event Detail Report Level Six

Running the Migration Planning Tool in Batch Mode

Part of the MPT product is a job stream file, MPTJOB.MPT.TELESUP, that you can use to create the MPTDB database and generate the installed products report as well as report levels 1, 2, 3, and 6 of migration planning reports for all files located on the system being analyzed.

Customizing the MPTJOB Job Stream File

If you want to customize the reports produced by MPTJOB, you have to modify the program using a text editor. Locate the :RUN command associated with MPTREPT and modify the line according to the options described in "Generating Migration Planning Reports".

You should also modify the appropriate !JOB command lines to add required passwords.

You can equate the formal designator MPTOPTNS to a parameter file of your own choosing, rather than the default file MPTOPTNS.MPT.TELESUP provided with the product. Find the appropriate file equation in the MPTJOB file and modify it as described in "Defining User Migration Options".

Streaming MPTJOB

It is best if you stream the MPT job after all other users have left the system. MPTJOB will purge any existing MPTDB databases before it creates a new database. During the job the system will be extremely busy and disk files should not be accessed so they can be analyzed. This job lasts from 30 minutes to more than eight hours, depending on the number and types of disk files on your system.

You initiate the MPT job stream file by entering the following:

```
:STREAM MPTJOB.MPT.TELESUP
```

Note

The job stream attempts to log on as `MANAGER.SYS` because the MPT program requires SM capability in order to scan all files on the system.

Progress reports will be sent to the console during the job execution.

Migration Planning Tool Program Error Messages

The error and warning message listed in the following four tables are generated by the four programs that make up the MPT product.

MPTPROD Program Error Messages

The error and warning messages listed below may result when running the MPTPROD program. Any of these errors may indicate some problems with the MPTPROD program, the MPT message catalog, or the MPTPARMS parameter file.

3	**MESSAGE**	`UNABLE TO OPEN THE PRODUCTS FILE (MPTPARMS).`
	CAUSE	The `FOPEN` intrinsic failed.
	ACTION	Be sure that the `MPTPARMS` file has been localized using the appropriate file equation. `MPTPROD` expects the file to be in the same group in which it is executing. For further information, refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative.
10	**MESSAGE**	`UNABLE TO OPEN THE 'BSTORE' DATABASE`
	CAUSE	The `DBOPEN` intrinsic failed while trying to open the 'BSTORE' database.
	ACTION	Refer to the *IMAGE Data Base Management System Reference Manual* (32215-90003) or *TurboIMAGE Data Base Management System Reference Manual* (32215-90050) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative.
11	**MESSAGE**	`UNABLE TO CREATE THE PRODUCTS FILE (MPTPARMS)`
	CAUSE	The `FOPEN` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved from this information, please contact a Hewlett-Packard Support Representative.

MPT Program Error Messages

The error and warning messages listed below may result when running the MPT program. Any of these errors may indicate some problems with the MPT program, the MPT message catalog, or the MPTPARMS parameter file.

3	**MESSAGE**	`UNABLE TO CREATE SON PROCESS (CREATEPROCESS ERROR` *errornumber*).
	CAUSE	The `CREATEPROCESS` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.
10	**MESSAGE**	`OPEN ERROR` *errornumber* `ON SL FILE` *filenumber*.
	CAUSE	The `FOPEN` intrinsic failed to open an SL file.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative.
11	**MESSAGE**	`READ ERROR` *errornumber* `ON SL FILE` *filenumber*.
	CAUSE	The `FREAD` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.
12	**MESSAGE**	`OPEN ERROR` *errornumber* `ON PROGRAM FILE` *filenumber*.
	CAUSE	The `FOPEN` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.

13	**MESSAGE**	`READ ERROR` *errornumber* `ON PROGRAM FILE` *filenumber*.
	CAUSE	The `FOPEN` intrinsic failed to open an SL file.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative.
14	**MESSAGE**	`OPEN ERROR` *errornumber* `ON ASCII FILE` *filenumber*.
	CAUSE	The `FOPEN` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.
15	**MESSAGE**	`READ ERROR` *errornumber* `ON ASCII FILE` *filenumber*.
	CAUSE	The `FREAD` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.
17	**MESSAGE**	`NO PROGRAMS WERE FOUND IN THE 'MPTPARMS' FILE. RUNNING INCOMPATIBLE PROGRAMS CAN'T BE FLAGGED IN UDCS OR STREAMS BUT THE FILE SCANNING WILL CONTINUE ANYWAY.`
	CAUSE	MPT reached the end of the file and failed to locate the `$PROGRAM` section.
	ACTION	MPTPARMS is part of the MPT product and should be on the store tape as MPTPARMS. Make sure the MPTPARMS file contains the `$PROGRAM` section. If not, restore MPTPARMS from the product tape and recheck. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative.
19	**MESSAGE**	`NO COMMANDS WERE FOUND IN THE 'MPTPARMS' FILE. NO UDC OR STREAM FILE INCOMPATIBILITIES CAN BE FLAGGED BUT THE FILE SCANNING WILL CONTINUE ANYWAY.`
	CAUSE	MPT reached the end of the file and failed to locate the `$COMMANDS` section.
	ACTION	Make sure that the `$COMMANDS` section of the MPTPARMS file was on the original product tape. If not, please contact a Hewlett-Packard Support Representative. If no information on UDC or commands is desired, make sure that no list of commands follows the `$COMMANDS` section.

21	**MESSAGE**	`NO INTRINSICS WERE FOUND IN THE 'MPTPARMS' FILE. NO PROGRAM FILE INCOMPATIBILITIES CAN BE FLAGGED BUT THE FILE SCANNING WILL CONTINUE ANYWAY.`
	CAUSE	MPT reached the end of the file and failed to locate the `$INTRINSICS` section.
	ACTION	Make sure the `$INTRINSICS` section of the MPTPARMS file was included on the original product tape. If not, please contact a Hewlett-Packard Support Representative.
28	**MESSAGE**	`UNABLE TO FIND THE FILE CODE PARAMETERS IN THE 'MPTPARMS' FILE. THIS INFORMATION IS CRITICAL TO THE OPERATION OF THIS PROGRAM. PLEASE LOCATE THE PROPER FILE AND MAKE IT ACCESSIBLE TO THIS PROGRAM.`
	CAUSE	MPT reached the end of the file and failed to locate the `$FCODE` section.
	ACTION	Make sure the `$FCODE` section of the MPTPARMS file was on the original store tape. If not, please contact a Hewlett-Packard Support Representative.
33	**MESSAGE**	`UNABLE TO OPEN THE MPT PARAMETER FILE` *filenumber*.
	CAUSE	The `FOPEN` intrinsic failed.
	ACTION	MPT expects the parameter file to be named MPTPARMS. Make sure the MPT parameter file was installed from the product tape. If not, restore it. If the MPT parameter file was installed, make sure the appropriate file equations are set up to localize the parameter file. If the parameter file is not on the tape, please contact a Hewlett-Packard Support Representative.
42	**MESSAGE**	`UNABLE TO OPEN OR CREATE THE 'MPTDB' DATABASE.`
	CAUSE	The `DBOPEN` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative.
43	**MESSAGE**	`UNABLE TO CREATE AN EXTRA DATA SEGMENT.`
	CAUSE	The `GETDSEG` intrinsic failed while allocating an extra data segment.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.

44	**MESSAGE**	`UNABLE TO RELEASE EXTRA DATA SEGMENT.`
	CAUSE	The `FREEDSEG` intrinsic failed.
	ACTION	Please contact a Hewlett-Packard Support Representative.
45	**MESSAGE**	`TOO MANY DATABASE ERRORS HAVE OCCURRED` *(> maxerrors).*
	CAUSE	The maximum number of allowable database errors (10) has been exceeded. Increase the default value by setting the MAXDBERR JCW.
	ACTION	Refer to the *IMAGE Data Base Management System Reference Manual* (32215-90003) or *TurboIMAGE Data Base Management System Reference Manual* (32215-90050) for information. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.
47	**MESSAGE**	`ERROR OCCURRED WHILE CREATING $STDIN FILE` *filenumber.*
	CAUSE	The `FOPEN` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative.
48	**MESSAGE**	`ERROR OCCURRED WHILE CREATING $STDLIST FILE` *filenumber.*
	CAUSE	The `FOPEN` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative.
50	**MESSAGE**	`UNABLE TO RUN DBSCHEMA.PUB.SYS (CREATEPROCESS ERROR` *errornumber*).
	CAUSE	The `CREATEPROCESS` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative.

51	**MESSAGE**	`ERROR OCCURRED WHILE READING THE $STDLIST FILE.`
	CAUSE	The `FREAD` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative.
52	**MESSAGE**	`ERROR OCCURRED WHILE WRITING TO THE $STDIN FILE.`
	CAUSE	The `FWRITE` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.
53	**MESSAGE**	`UNABLE TO RUN DBUTIL.PUB.SYS (CREATEPROCESS ERROR` *errornumber*).
	CAUSE	The `CREATEPROCESS` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.
54	**MESSAGE**	`UNABLE TO OPEN THE NEWLY CREATED 'MPTDB' DATABASE.`
	CAUSE	The `DBOPEN` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.
55	**MESSAGE**	`INVALID ENTRY IN THE 'FCODE' INFORMATION (NO "=" FOLLOWING` *number*).
	CAUSE	An equal sign is missing after the file code name in the `$FCODE` section of the MPPARMS file.
	ACTION	Be sure that entries in the `$FCODE` section of the MPTPARMS file have the proper syntax. Contact a Hewlett-Packard Support Representative.

56 **MESSAGE** `INVALID ENTRY IN THE 'FCODE' INFORMATION (NO FCODE"=" PARAMETER).`

CAUSE The format for one of the file codes in the `$FCODE` section of the MPTPARMS file is invalid.

ACTION Check the MPTPARMS file for errors in the file code formats. Make sure the `$FCODE` section of the MPTPARMS file consists of entries in the following format:

```
PROG  = 1029
SL    = 1031
      .      .
      .
FCODE = parameter
```

Contact a Hewlett-Packard Support Representative for the proper file code information.

---

57 **MESSAGE** `IMAGE ROOT FILE HAS INVALID CONDITION CODE` (*nnn*)

CAUSE The root file of a pre-TurboIMAGE database is invalid and cannot be analyzed by MPT. This error occurred before reading record 0 of the root file.

ACTION None.

---

58 **MESSAGE** `READ ERROR ON IMAGE ROOT FILE.`

CAUSE The `FREAD` intrinsic failed.

ACTION Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.

---

59 **MESSAGE** *??????* `ROOT FILE IS TOO LARGE FOR ANALYSIS.`

CAUSE The size of the database is too large for MPT to handle.

ACTION The database cannot be analyzed by MPT. It is unusual to find a root file of this size. Please contact a Hewlett-Packard Support Representative if this database must be analyzed.

60	**MESSAGE**	*???????* `ROOT FILE IS INVALID` *(nnn)*
	CAUSE	This error occurs on non-TurboIMAGE databases. MPT does not recognize the IMAGE version that created this root file, or the maximum root length (the number of items or number of sets) has been exceeded.
	ACTION	MPT cannot analyze the database associated with this root file. Please contact a Hewlett-Packard Support Representative.
61	**MESSAGE**	`ERROR ON MOVING DATA TO EXTRA DATA SEGMENT.`
	CAUSE	The `DMOVOUT` intrinsic failed. A root file could not be moved into the extra data segment.
	ACTION	The database cannot be analyzed. Please contact a Hewlett-Packard Support Representative.
62	**MESSAGE**	`UNABLE TO CLOSE IMAGE ROOT FILE.`
	CAUSE	The `FCLOSE` intrinsic failed.
	ACTION	Use the printed file information to determine the cause of the problem. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.
63	**MESSAGE**	`ERROR ON CALLING NM IMAGE DETAIL ROUTINE` *(nnn)*
	CAUSE	The `PGDETAILSPACE` routine failed.
	ACTION	Contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.
64	**MESSAGE**	`ERROR ON CALLING NM IMAGE MASTER ROUTINE` *(nnn)*
	CAUSE	The `PGMASTERSPACE` routine failed.
	ACTION	Contact a Hewlett-Packard Support Representative. Include all information listed by the MPT program.

MPTRTM Program Error Messages

The error and warning messages listed below may result when running the MPTRTM program. Any of these errors may indicate some problems with the MPTRTM program, the MPT message catalog, the MPTDB database, or the MPTPARMS parameter file.

2	**MESSAGE**	`UNABLE TO OPEN THE 'MPTDB' DATABASE.`
	CAUSE	The `DBOPEN` intrinsic failed.
	ACTION	Be sure the `MPTDB` has been localized properly by using the appropriate file equations. If this does not solve the problem, check to see if you have the appropriate capabilities to access the database. The SM capability is required. Check to see if the database has been created. If not, run MPT to create it. If the database is inaccessible, use the DBUTIL utility program to make it accessible. Refer to the *IMAGE Data Base Management System Reference Manual* (32215-90003) or *TurboIMAGE Data Base Management System Reference Manual* (32215-90050) for details.
8	**MESSAGE**	`UNABLE TO COMPLETE THE LISTF COMMAND.`
	CAUSE	The `COMMAND` intrinsic failed.
	ACTION	The file that MPT uses to scan the system logfiles is empty. Please contact a Hewlett-Packard Support Representative. Include all information listed by the MPTRTM program.
9	**MESSAGE**	`UNABLE TO OPEN THE LISTF FILE.`
	CAUSE	The `FOPEN` intrinsic failed to open the file with the names of the system log files.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPTRTM program.
13	**MESSAGE**	`UNABLE TO OPEN THE TAPE.`
	CAUSE	The `FOPEN` intrinsic failed to open the tape drive.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPTRTM program.

14	**MESSAGE**	`ATTEMPT TO ADDRESS OUT OF BLOCK.`
	CAUSE	MPTRTM is currently analyzing a bad system log file.
	ACTION	None. The problem is not an MPTRTM program error.
17	**MESSAGE**	`MISSING LOG FILES LOG`*nnnn* `to LOG`*nnnn*
	CAUSE	The `FOPEN` intrinsic failed to open the log files.
	ACTION	The log files are not on the system. If the log files are on the system, please contact a Hewlett-Packard Support Representative.
19	**MESSAGE**	`SORRY, THE DISC BLOCK SIZE IS TOO BIG` *(> maxsize)*.
	ACTION	Please contact a Hewlett-Packard Support Representative. Include all information listed by the MPTRTM program.

MPTREPT Program Error Messages

The error and warning messages listed below may result when running the MPTREPT program. Any of these errors may indicate some problems with the MPTREPT program, the MPT message catalog, the MPTPARMS parameter file, or the MPTOPTNS user option file.

2	**MESSAGE**	`UNABLE TO OPEN THE 'MPTDB' DATABASE.`
	CAUSE	The `DBOPEN` intrinsic failed.
	ACTION	The MPTDB database must be present on the system. If it is not, build it by running MPT. If the database is present, make sure the appropriate file equations are set up. Refer to the *IMAGE Data Base Management System Reference Manual* (32215-90003) or *TurboIMAGE Data Base Management System Reference Manual* (32215-90050) for further details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPTREPT program.
4	**MESSAGE**	`UNABLE TO CREATE THE 'SORTSCR' SCRATCH FILE.`
	CAUSE	The `FOPEN` intrinsic failed.
	ACTION	Refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPTREPT program.
7	**MESSAGE**	*filesetname* `REJECTED. TOO MANY FILESETS.`
	CAUSE	The maximum number of filesets has been exceeded.
	ACTION	MPT will continue to process a report. You can generate another report using only those filesets that are of interest or make your fileset descriptions broader. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPTREPT program.
15	**MESSAGE**	`UNABLE TO OPEN THE filename FILE.`
	CAUSE	The `FOPEN` intrinsic failed.
	ACTION	If the MPTPARMS file is missing, restore it from tape and set up the appropriate file equations if applicable. If the file specified in the error message is not MPTPARMS, refer to the *Migration Process Guide* (30367-90007) for details. If the problem cannot be resolved, please contact a Hewlett-Packard Support Representative. Include all information listed by the MPTREPT program.

27	**MESSAGE**	`THERE IS NO "=" FOLLOWING THE COMMAND` *commandname* `(COMMAND IGNORED).`
	CAUSE	An equal sign is missing in the MPTOPTNS file.
	ACTION	Edit the MPTOPTNS file and place an equal sign after the reported command.
29	**MESSAGE**	`YOU CAN'T ADD A FILE WITHOUT SPECIFYING THE 'FNAME' PARAMETER.`
	CAUSE	MPTREPT did not find the filename in the correct format in the MPTOPTNS file.
	ACTION	Edit the MPTOPTNS file and enter "FNAMES=" in front of the specified filename.
33	**MESSAGE**	`TOO MANY FILE CRITERIA WERE FOUND. THE REST WILL BE IGNORED.`
	ACTION	Please contact a Hewlett-Packard Support Representative. Include all information listed by the MPTREPT program.

Index

E

F

K

L