

Understanding Migration

David Elward
Taurus Software, Inc.
770 Welch Road Suite 3A
Palo Alto, CA 94304

Introduction

If you're considering purchasing an HP3000 Series 900 machine, then there's a very good chance you will be "migrating" once it arrives. By migrating, I mean moving your existing programs and data files from your "classic HP3000" to your new "Spectrum Machine".

Migration has never been much of an issue within the Hewlett-Packard world, because up until now, all HP3000's have had the same hardware instruction set. This means that compiled programs will run on any HP3000, with the only difference being performance.

As you probably know by now, the Series 900 machines have a completely different instruction set than classic HP3000's. This instruction set has fewer instructions than traditional computer instruction sets and is based up RISC (Reduced Instruction Set Computer) technology. Hewlett-Packard calls this technology Hewlett-Packard Precision Architecture (HPPA). Within this paper, the terms HPPA machine, Spectrum machine, Series 900 machine, and MPE XL machine will all be used interchangeably to indicate HP's new computer. The term "classic HP3000" will be used to indicate older machines that run MPE/V.

When you begin learning about migrating and the new HPPA machines, you soon realize that there is an abundance of terms you've never heard before. Terms like "native mode", "mapped files", "Sysgen", and "MPE XL" are some of them. This presentation will focus on what these terms mean to you and your migration. Specific topics that will be discussed are as follows:

1. What is migration?
2. What is the difference between Compatibility Mode and Native Mode?
3. How do I prepare for migration?
4. What are some of the new MPE XL features and commands?
5. What are some of the new MPE XL utility functions?

What Is Migration?

Migration is simply moving programs and files which run on MPE/V based machines to HPPA based machines. The question then is: "Why is any migration effort needed at all?"

When the HP3000 was first introduced, it had an operating system called MPE, meaning Multi Programming Executive. Subsequent revisions of MPE were released, and most of you are probably now running the MPE V/E operating system. On the HP3000 Series 900 machines, the operating system is called MPE XL. The operating system on all classic HP3000's is written in a language specifically designed for the HP3000 hardware called SPL, Systems Programming Language. For the most part, MPE XL is a complete rewrite of MPE/V in a language called Pascal/XL. Pascal/XL is standard Pascal plus a variety of programming extensions added by Hewlett-Packard.

When MPE XL was designed, paramount importance was given to the issue of compatibility. With few exceptions, MPE XL is, and was designed to be, completely compatible with all MPE/V software. To make all MPE/V software run on MPE XL, Hewlett-Packard had a big problem because all MPE/V programs contained instructions for the old hardware which won't run on the new machines. The solution HP chose, was to write software that would emulate the classic HP3000 hardware and to integrate this emulator deep within MPE XL. By doing this, Hewlett-Packard has made the differences between MPE/V and MPE XL almost completely invisible.

Compatibility Mode versus Native Mode

"Compatibility Mode" (CM) is the term HP has chosen to mean running a program that contains instructions for the classic HP3000. When a compatibility mode program runs on MPE XL, the instructions are actually being emulated by the emulation software.

Almost all, if not all, of your programs will execute in compatibility mode with little or no effort on your part. Compatibility mode programs are typically created on an MPE/V machine, and then moved with :STORE/:RESTORE, or :DCOPY to an MPE XL machine. You can create a compatibility mode program by running an MPE/V compiler in compatibility mode on a MPE XL machine. Running an MPE/V compiler in compatibility mode generates object code with classic HP3000 instructions. This means they can run on the classic HP3000 using its instruction set or in compatibility mode on the HPPA machine emulating the classic HP3000 instruction set.

"Native Mode" (NM) is the term HP has chosen to mean the running of a program that executes HPPA instructions. A native mode program must be created by a native mode compiler on MPE XL, and will not run on MPE/V.

Native mode programs execute much faster than compatibility mode programs because the process of emulation takes time. On the average, a native mode program will execute about 12 times faster than its compatibility mode counterpart. Currently, there are four native mode compilers available on MPE XL. They are: Pascal/XL, FORTRAN 77/XL, COBOL/II/XL, and C/XL.

OCT

You are probably not surprized that HP has created a middle ground. The OCT (Object Code Translator) produces programs that can be executed in native mode and on the classic HP3000 with out having to recompile.

What the object code translator does is translate the classic HP3000 instructions contained in executable code (PROG or SL files) to native mode instructions. The native mode instructions are then added to the program or SL file, leaving the classic HP3000 instructions in place.

There are two reasons for leaving the old instructions in place. One reason is because the old code is used by the translated instructions, and the other is so that the translated file may be moved back to MPE/V and still work. It will still work because MPE/V ignores the HPPA instructions within the file. Translated programs tend to run about 4 times faster than untranslated programs.

Switching Modes

Sometimes it is necessary to have a program run in both native mode and compatibility mode. For example, a native mode COBOL program needs to call an SPL procedure. Since there is no native mode SPL compiler, it must remain in compatibility mode. Hewlett-Packard has provided a mechanism for doing this called the "Switch Subsystem".

The Switch Subsystem is a set of procedures that enable a program to call a procedure that runs in the opposite mode than the program is currently running in. In order to understand how the switch stub works, we need to understand the internal data structures.

On MPE/V, the only data structure that is created when the program is executed is the program's stack. (See Figure 1) However on MPE XL three different data structures are created: the Compatibility Mode Stack (Same format as MPE/V), the Native Mode Stack, and the Native Mode Heap. (See Figure 2)

Environment for MPE/V Program

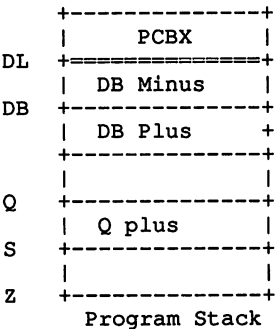


Figure 1.

Environment for MPE XL Program

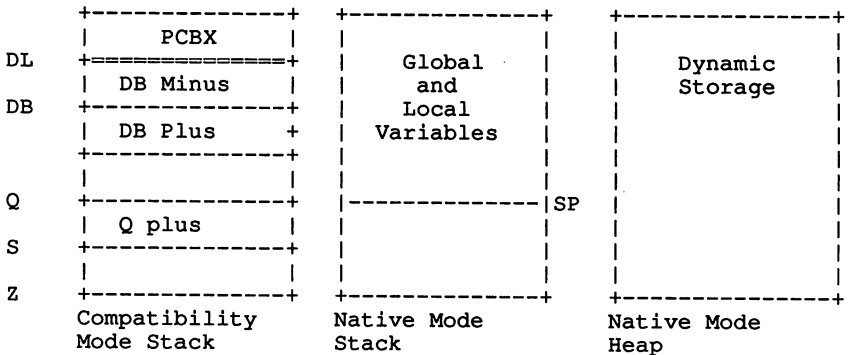


Figure 2.

When a program is executing in CM, it has access only to the Compatibility Mode stack. When a program is executing in NM, it has access only to the Native Mode Stack and Heap. Because of this, a "switch stub" is required to change modes. The switch stub is responsible for calling a switch procedure with the correct parameters to direct it to copy the procedure parameters from the current stack to the stack for the opposite mode. An easy fill-in-the-blank utility program is supplied with MPE XL to help the programmer write switch stubs.

Which is better for my program - CM or NM?

When it comes time for you to migrate, you'll be faced with many decisions, not the least of which is whether your programs will run in native mode or compatibility mode. The advantage of native mode is that your programs will run much faster. The advantage of compatibility mode is that you can move your programs from MPE/V to MPE XL and run them all within a matter of minutes.

In the long run, an attempt should be made to get as many programs as possible to run in native mode. Pascal and COBOL/II programs should not present much of a problem to migrate to a native mode compiler, and FORTRAN 77 programs should migrate to native mode easily but watch out for the IEEE floating point.

Relatively Easy Programs to Migrate to Native Mode:

1. Pascal programs. Pascal/XL has many new features over Pascal/V, however you should be careful when using them because they will not work on MPE/V.

2. Cobol/II programs. Cobol 66 programs will need to be changed to Cobol/II, before they can be compiled in native mode.

Potentially Difficult Programs to Migrate to Native Mode:

1. Fortran Programs. All Fortran 66 programs will have to be changed to Fortran 77 before they can be compiled in native mode. Floating point numbers are stored differently internally on HPPA machines than on classic HP3000's, so any floating point data that has been written to files or data bases must be changed to IEEE format using an intrinsic supplied with MPE XL.

Difficult or Impossible Programs to Migrate to Native Mode:

1. SPL Programs. There is no native mode SPL compiler available from Hewlett-Packard, although there is one called SPLASH available from a third party. SPL programs must run in compatibility mode or must be rewritten in another language.
2. Basic Programs. Basic/3000 will never be available in native mode on MPE XL, however HP Business Basic will be.
3. Programs that use privileged mode (PM). Many privileged mode programs will run correctly in CM, and many will not. Because the MPE XL operating system is completely different internally than MPE/V, there is very little chance that privileged mode programs can be run in native mode without major changes.

Native Mode Considerations

In general, there are two big differences between the behavior of compatibility mode programs and native mode programs. One is data alignment, and the other is floating point. Compilers like to align data on word boundaries; the trouble is that classic HP3000's have 16 bit words, and Series 900 machines have 32 bit words. This will cause the data to be stored differently in compatibility mode programs than in native mode programs. This could cause problems for programs that use existing data files, or programs that call external procedures expecting the data in a different format. Hewlett-Packard has supplied a compiler option for the native mode compilers to direct the data to be aligned the exact same way as it would be using a compatibility mode compiler.

Classic HP3000's use their own format for storing and manipulating floating point (Real) numbers, and HPPA machines use the IEEE Standard for storing and manipulating floating point numbers.

Single precision HP3000 floating point numbers have a precision of 6.9 digits and a range of $\pm 1.2\text{E}77$ to $\pm 8.6\text{E}-78$. Single precision IEEE floating point numbers have a precision of 7.2 digits and a range of $\pm 3.4\text{E}38$ to $\pm 1.4\text{E}-45$.

Double precision HP3000 floating point numbers have a precision 16.5 digits, and the same range as single precision numbers. Double precision IEEE floating point numbers have a precision of 15.9 digits and a range of $\pm 1.8\text{E}308$ to $\pm 4.9\text{E}-324$.

This format difference will probably have a negligible affect on mathematic results, however it is a problem for floating point data stored within files. Classic HP3000 floating point numbers stored within files, will have to be converted to the IEEE format using the `HPFP_CONVERT` intrinsic if you wish to access them in native mode.

Preparing for your Migration

HP has provided a number of facilities to allow you to prepare for your migration prior to your HPPA machine arriving. One of the utility programs which runs on the classic HP3000 is called RTM (Run Time Monitor).

The RTM is a utility intended to help you identify areas within your MPE/V programs that could cause a problem when ported to MPE XL. Because of fundamental differences between MPE/V and MPE XL, there are some things that may work on MPE/V that will fail on MPE XL. The RTM is intended to help you detect these things by logging calls to MPE/V intrinsics that could be a potential problem on MPE XL.

Logging is controlled using a program called `RTMSYS.PUB.SYS` and the logging results may be printed using a program called `RTMREP.PUB.SYS`. Typically, the user will monitor an application being considered for migration for a number of days. Then the RTM reports will be printed and any potential problems will be investigated. The advantage of the RTM is that it runs on MPE/V, and problems may be corrected long before migration actually begins.

OCA

Another utility which can be run on the classic HP3000 to help you prepare for migration is the OCA (Object Code Analyzer). The OCA is a utility program that scans program or SL files for potential MPE XL problems.

It is intended to perform the same function as the RTM, however its method of operation is different. The RTM logs intrinsic calls when they actually happen; the OCA scans a program or SL file looking for intrinsic calls. The OCA has the advantage of being able to obtain the results immediately without waiting for days of logging. Its disadvantage is that it may not be as accurate because it cannot always tell what parameters an intrinsic is being called with. Like the RTM, the OCA runs on MPE/V, and may be used before migration actually begins.

MPE XL New Features

A number of new features have been added to MPE XL machines. The features fall into three areas: intrinsics, mapped files, and command interpreter changes.

Intrinsics

As you may have guessed, new intrinsics have been added to MPE XL. Many of the new intrinsics have to do with switching between NM and CM. Others are provided to access features of the new command interpreter, including a new `HPCICOMMAND` intrinsic that can execute any MPE XL command including User Defined Commands (UDCs). Another new intrinsic is `HPFOPEN` that opens a file just as the `FOPEN` intrinsic does, but `HPFOPEN` is implemented with a couple of new features and is designed to be easily expanded.

Mapped Files

Using the `HPFOPEN` intrinsic, a program may now open a file "mapped". When a file is opened as a mapped file, a pointer is returned to the calling program. This pointer may be used to access the file directly without going through the file system. The mapped file may be treated exactly as if it was a data array within the program's stack. The advantage of mapped files is a tremendous performance improvement because the file system overhead is virtually eliminated.

MPE XL Command Interpreter

The MPE XL Command Interpreter has almost all of the commands that the MPE/V Command Interpreter has. The commands that have been deleted were deleted because they have no place within MPE XL. The deleted commands include `CACHECONTROL`, `DATA`, `FULLBACKUP`, `GIVE`, `LISTVS`, `PARTBACKUP`, `PTAPE`, and `VINIT`.

Some of the existing MPE/V commands have been modified or enhanced on MPE XL. Some of these commands are: `IF` has been greatly enhanced, `LISTF` has had some new options added, `LISTACCT`, `LISTGROUP`, `LISTUSER` have been changed to give output similar to `LISTDIR5`, `REDO` has been enhanced, `RUN` has had some options added.

Several new commands have been added to MPE XL that give dramatic improvement over MPE/V. The `SETVAR`, `SHOWVAR`, `DELETEVAR`, and `INPUT` commands have been added to manipulate variables. Variables are similar to JCWs except that they may contain string and boolean values in addition to numeric values. About 60 variables are predefined within MPE XL and they contain a variety of information about the users environment, such as the CI

prompt, user's jobname, user, group and account among others.

A redo stack has been added that saves that last 20 or more commands. A DO command that does any command within the redo stack has been added, and a LISTREDO command that displays the redo stack has also been added.

A fast COPY command has been added that copies files 10-20 times faster than FCOPY. A PRINT command has been added to display the contents of a file, or prints the file upon a line printer without having to use an editor.

Perhaps the greatest improvement is the addition of command files. Command files are similar to UDCs except that they may only contain one group of commands, and they do not need to be cataloged. Whenever a command is entered that MPE XL does not recognize, it is assumed to be a command file or program name, and MPE XL searches the user's group, PUB group, and PUB.SYS group file a program file or command file with the same name.

SYSGEN

There is no SYSDUMP program on MPE XL to configure your system. SYSDUMP has been replaced by a program called SYSGEN to configure your system and make cold load tapes. Backup is performed using STORE. SYSGEN has a much different approach than SYSDUMP; it is command oriented rather than question oriented. This approach is much more direct than SYSDUMP's.

In addition to SYSGEN, there is a program NMCONFIG that is used to configure your LAN (Local Area Network), DTC (Distributed Terminal Controllers), and terminals connected to DTC's. Since all terminals on MPE XL except the console must go through a DTC, the NMCONFIG program must be used to configure all of your terminals.

A program called VOLUTIL replaces VINIT and is used to configure your disc drives.

DIRMIG

If you wish to replace one of your classic HP3000's with a Series 900 machine, you will want to move your entire accounting structure along with all of your files to the new machine. Hewlett-Packard has supplied a program to assist you in doing this called DIRMIG. DIRMIG runs on your MPE/V system and creates a tape containing accounting and configuration information to be moved to MPE XL.

Utilities

Many of the system utility programs have been either removed or replaced on MPE XL. The following list summarizes the major changes:

1. **LISTDIR5** has been removed. Its functionality has been moved to standard MPE XL commands. **LISTF** options 3, and is similar to the **LISTDIR5 LISTF** command. **LISTF** option 4 is similar to the **LISTDIR5 LISTSEC** command. The **LISTUSER**, **LISTGROUP**, and **LISTACCT** commands now have output similar to **LISTDIR5**, and the octal dump mode has been eliminated.
2. The **FREE5** program has been replaced with the **DISCFREE** program. **DISCFREE** performs the same function as **FREE5**, but the output format is different.
3. **SPOOK** has been modified internally, but no differences are visible to the user.
4. The **LISTEQ5** program has been eliminated. It no longer has a use in MPE/V or MPE XL because of the **LISTEQ** and **LISTFTEMP** commands.
5. **DEBUG** has been completely rewritten and does not bear any resemblance to the MPE/V debugger. The MPE XL debugger is many times more powerful and makes extensive use of windows to display information.

Summary

The first step towards a successful migration is education. MPE XL contains many new things that at first can be overwhelming. What is comforting is that when you begin to use MPE XL, you don't even need to know you're using it. All of the commands you are likely to use perform just the same, and programs moved to MPE XL in compatibility mode just run. Only when you are ready to maximize the benefits of your new machine do you need to have a good understanding of the migration process.