



hp business critical
systems

June 2003



technical
white paper

Porting HP OpenVMS Applications to Intel® Itanium® Architecture

Table of Contents

Executive Summary	2
Overview	2
Ease of Transition to Itanium®-based Platform and Protection of Investment	2
Parallelism Technology Project Established	3
Summary	3
Product Availability	3
OpenVMS Clusters	3
VAX Porting Plans	3
Compiler plans	4
Itanium Architecture Code Generators	4
Initial HP OpenVMS Compilers	4
Future C, C++, and Fortran Compilers	5
Run-time Libraries	5
Development Tools	5
DECset Tools	5
Binary Translator	5
Understanding the Differences	6
Calling Standard	6
Object File Format	6
Floating-point Data Types	7
Porting Process	7
Assessment of Porting Needs	7
Compiling/Linking/ Running	7
Unit Testing	8
System Testing	8
Regression Testing	8
Shipping the Revised Application	8
Operation and Installation of Processes and Procedures	8
Remote Support and Maintenance Considerations	8
Porting Checklist	9
Porting Issues	9
Architecture-specific Code	9
Object File Format Code	9
Alpha or VAX Logic	9
Porting Resources	10
Additional Dependencies and Considerations	10
For More Information	10

Executive Summary

This white paper describes aspects of porting HP OpenVMS applications from OpenVMS Alpha to the OpenVMS Industry Standard 64 operating system, which runs on the Intel® Itanium® architecture. The audience for this paper is system integrators, systems and software developers, independent software vendors (ISVs), and other partners.

Overview

In a bold move that promises to change the landscape of high-performance enterprise computing, Hewlett-Packard and Intel have committed to work together to accelerate the adoption of Itanium-based computing in the enterprise marketplace. The Itanium® architecture is targeted at the highest levels of the enterprise.

With this step both Hewlett-Packard and Intel will be the industry leaders in unique and complementary areas of expertise – high-volume superior microprocessor design and manufacturing, and high-performance enterprise solutions. They are working together, to establish the Itanium architecture as the industry leader for high-performance enterprise computing.

To maximize the advantages of working with Intel, Hewlett-Packard is standardizing its 64-bit enterprise server product line on the Itanium architecture and is bringing its OpenVMS operating system, middleware, and application portfolio to the Itanium architecture. Integral to this strategy is the transfer of Hewlett-Packard's tremendous Alpha microprocessor technology – backed by more than 10 years of technical leadership in 64-bit computing solutions – to accelerate and enhance ongoing development of the Itanium architecture. The Itanium architecture has already received considerable support throughout the industry, with more than 400 applications currently under development or released and numerous customers engaged in pilot programs prior to the initial platform release. The Hewlett-Packard and Intel relationship will further extend the reach of the Itanium architecture to the highest levels of the enterprise.

Ease of Transition to Itanium®-based Platform and Protection of Investment

To ensure that customers who want to take advantage of the Itanium architecture enjoy a transparent and smooth transition, Hewlett-Packard and Intel have committed significant resources to a robust technology transition program. This program includes the complete porting of the OpenVMS operating system, compilers, middleware, and application development tools to the Itanium architecture.

The Itanium architecture has a 64-bit model and basic system functions similar to the Alpha chip. These two characteristics will dramatically reduce the effort of porting the OpenVMS operating system to Itanium architecture and will allow Hewlett-Packard to use a single source-code base to produce both the Alpha and Itanium architecture variants. This will allow all nonhardware dependent and non-VAX functionality and performance improvements to be incorporated into both versions without multiple changes to the source code and will minimize the time required to perform qualification testing. This comprehensive strategy for transitioning Hewlett-Packard technologies to the Itanium microprocessor family will ensure that all of the high-performance features, functionality, and clustering capabilities of the OpenVMS operating system, along with compilers, layered products, and middleware will be available on the Itanium architecture. This strategy strongly reinforces the concept of "compile and go."

Using this common application development technique will allow ISVs and end-user developers to continue using their current and future Alpha systems while transitioning to the future Itanium-based platforms. Operating system, middleware, and application tools compatibility will protect customers' long-term investments in their OpenVMS operating environments and will deliver cost reductions across the board from suppliers, to integrators, to customers, to users. Extensive porting support for customers and ISVs

will also be available.

Parallelism Technology Project Established

To further advance the leadership of Hewlett-Packard in high-performance technical computing, Hewlett-Packard and Intel have recently initiated a Parallelism Technology Project (PTP) staffed with engineers from both companies. The project is conducting joint research and advanced development into parallel and supercomputing technologies, particularly the processor, compiler, software, and interconnect technologies necessary to meet the future needs of leading-edge, high-performance computing customers. PTP engineers will also engage directly with customers to understand their requirements, prototype solutions, integrate the results into volume product development at Hewlett-Packard and Intel, and drive new industry standards for high-end computing.

Summary

Hewlett-Packard's move to the Itanium architecture creates a strong new future for customers and partners. A long-term roadmap for growth has been established, and customer investments in applications will be protected and enhanced. Hewlett-Packard is committed to providing the most advanced, high-performance systems based on the volume economics of the Intel® Itanium® architecture. This architecture is the best choice for the enterprise and for Hewlett-Packard customers.

Product Availability

All of the basic OpenVMS operating system functionality that is currently available on Alpha will be available on the Itanium architecture as listed at the following URL:

http://www.hp.com/products1/evolution/alpha_retaintrust/openvms/products.html

OpenVMS Clusters

Hewlett-Packard plans to support mixed-architecture clusters on the Itanium architecture and Alpha platform with limited certification of potential configurations with the H1 2003 release (see

http://www.hp.com/products1/evolution/alpha_retaintrust/openvms/products.html). A separate whitepaper, entitled "OpenVMS Clusters Move to Intel® Itanium® Systems" has been published that details the cluster strategy and is available at the following URL:

http://www.hp.com/products1/evolution/alpha_retaintrust/openvms/resources.html

VAX Porting Plans

Although this white paper primarily addresses porting OpenVMS Alpha applications, the following information is provided for customers who need to port their VAX applications to the Itanium processor.

For customers who have their VAX application sources, the work required to port those sources to the Itanium architecture is roughly equivalent to the work that would have been required to port those sources to Alpha. These requirements are detailed in the manual *Migrating an Application from OpenVMS VAX to OpenVMS Alpha*, available at the following URL:

http://www.openvms.compaq.com/doc/73final/documentation/pdf/ovms_migr_app.pdf

IMPORTANT: Hewlett-Packard continues to formulate a VAX porting plan. The guide cited was written for the VAX to Alpha migration and should not be construed as a definitive manual for the VAX to Itanium architecture port. It will, however, provide customers with a basic understanding of how to modify their VAX sources.

For customers who do not have their VAX application sources, Hewlett-Packard recommends the following two-step process:

1. Translate the VAX image to an Alpha image using an upcoming version of DECmigrate
2. Translate the Alpha image to an Itanium architecture image using the binary translator tool. See the Binary Translator section (page 5) for more

information about this tool.

More VAX porting information will be forthcoming in a future document.

Compiler plans

As with the Alpha chip set, the compilers can make the difference between a highly optimized and well-tuned application and an application that is slow, requires additional hardware, and does not scale well with load at critical times.

Hewlett-Packard has joined with Intel to develop a suite of development tools that both maximizes the functionality of the new Itanium architecture and maintains the functionality and user interface that characterizes the Alpha development tool suite.

Hewlett-Packard has transferred some compiler technology to Intel. This provides the Intel compiler engineering group the knowledge to ensure a superior tool suite for Itanium-based platforms. In addition, two other engineering groups have been formed to support compiler development:

- A combined engineering group, staffed by both Intel and Hewlett-Packard engineering resources, to achieve expertise in compiler technology for the Itanium architecture.
- A Hewlett-Packard-based compiler development group to enhance, promote, and support a version of the Itanium compiler that is focused for the OpenVMS operating system.

These three engineering groups will share knowledge that allows the existing Hewlett-Packard compiler technology to remain effective and user friendly on HP platforms.

The OpenVMS compiler strategy can be summed up in one phrase: Compile and Go! The goal is to provide to Hewlett-Packard customers total compatibility between the Alpha and Itanium-based platforms so that application porting is as simple as possible. The following sections detail the OpenVMS compiler plans.

Itanium Architecture Code Generators

GEM is the Hewlett-Packard compiler back end that is used in the Alpha compilers to generate OpenVMS object files with Alpha architecture instructions. This back end will be enhanced to support the Itanium architecture, the ELF object file format, and the DWARF symbol table format (see page 6).

ECG is the Intel back end that is used in the Itanium compilers. This back end will be enhanced to support the OpenVMS Calling Standard (see page 6), and the VAX floating-point data types (see page 7). Hewlett-Packard and Intel will develop plans to move many of the compilers to the ECG back end over time. Both back ends will be supported by Hewlett-Packard. New optimization techniques and taking advantage of new hardware capabilities will be focused in the ECG back end.

Initial HP OpenVMS Compilers

The Hewlett-Packard compiler group is committed to helping the migration to the Itanium architecture succeed. That not only means that the individual compilers focus on the ability to compile the same programs as they can today, it also means that Hewlett-Packard provide support for each of the languages that are in use today.

The OpenVMS Product Availability chart located at http://www.hp.com/products1/evolution/alpha_retaintrust/openvms/products.html illustrates the broad range of languages that will be available and their respective time frames. The goal is to be compatible with the Alpha compilers, which will allow both Hewlett-Packard developed software and user-developed software to be used with little or no modification.

Hewlett-Packard will create the IMACRO compiler, which will compile VAX MACRO-32 code that was previously ported to the Alpha architecture.

There are specific languages and dialects of languages which Hewlett-Packard does not plan to make available for the Itanium platform. Hewlett-Packard will provide an Ada 95 compiler on the Itanium architecture but will not port the existing Hewlett-Packard Ada 83. Hewlett-Packard has no plans to make PL/1 available.

Future C, C++, and Fortran Compilers

Hewlett-Packard and Intel have established a joint engineering team chartered to deliver the best C, C++, and Fortran compilers available for the Itanium architecture. These compilers will be developed using the best technologies from both companies

The engineering teams have been working on this “best in class” effort since August 2001. These new compilers will have the features to support the compile and go strategy that was discussed earlier. Both companies would like to have these compilers available for the launch of OpenVMS on the Itanium architecture; the companies continue to work hard on this availability.

Run-time Libraries

Hewlett-Packard is porting OpenVMS run-time libraries from currently supported products to the Itanium architecture. All four Hewlett-Packard DECthreads interfaces will be ported to the Itanium Processor Family. The four interfaces are 1c, D4, CMA, and TIS.

Development Tools

Hewlett-Packard is porting the existing OpenVMS development tools and utilities that currently ship with the OpenVMS operating system to the Itanium architecture. This means that developers should be able to use their existing procedures for developing, debugging, testing, and deploying their applications on the Itanium-based platform.

DECset Tools

Hewlett-Packard will port the full suite of DECset tools to the Itanium architecture. These include:

- Language-Sensitive Editor/Source Code Analyzer (LSE/SCA)
- Code Management System (CMS)
- Module Management System (MMS)
- Digital Test Manager
- Performance and Coverage Analyzer (PCA)

All of the DECset products will be available with the first production release of OpenVMS on the Itanium architecture in H1 2004. For a detailed timetable, see http://www.hp.com/products1/evolution/alpha_retaintrust/openvms/index.html.

Binary Translator

A binary translator will be provided that will translate user mode OpenVMS Alpha images to OpenVMS for Itanium architecture images. The translator will translate:

- Native Alpha images
- VAX images that have been translated using DECmigrate into Alpha images

The translator will not translate all images. Among these are self-modifying code, user-written system services, and privileged instructions.

Understanding the Differences

Although the commonality in features between Alpha and the Itanium architecture is clear, there will be some significant implementation differences between the two platforms that may affect user-written applications. The purpose of this section is to identify those areas and to offer possible workarounds for surmounting those differences.

Calling Standard

The calling standard is an area where the chip architecture can influence applications. Hewlett-Packard is currently defining a new calling standard for OpenVMS applications running on the Itanium architecture. This standard will be based on the Intel calling standard and will include modifications that are required by the OpenVMS operating system.

Applications written in a high-level language typically will not have to deal with this change because the compilers will implement the calling standard. However, developers might have to modify code when it is ported to the Itanium architecture if the following conditions exist:

- Any code that directly implements the current OpenVMS Alpha Calling Standard
- Any code that is dependent on the details of the OpenVMS Alpha Calling Standard, for example, to directly manipulate parameters on the stack or to change the parameters that are passed to a routine

The design of the new calling standard has been completed and a draft of the new calling standard specification can be found at <http://h18003.www1.hp.com/hps/ipf-enterprise/openvms.html>.

Object File Format

Hewlett-Packard has determined that the most efficient way to utilize the compiler technologies developed by Intel is to adopt two industry standards, both of which are used by the Intel compiler:

- ELF (executable and linkable format) describes the object file and executable file formats.
- DWARF (debugging and traceback information format) describes the way debugging and traceback information is stored in the object and executable files.

One clear advantage of this decision is that development tools will be ported more easily to OpenVMS in the future. All of the compilers, development tools, and utilities provided with the OpenVMS operating system on the Itanium architecture will utilize and understand these new formats. Application developers will not need to be concerned with these formats unless their applications have specific knowledge of them.

In a typical application scenario in which an application is compiled, linked, debugged, and deployed, the details of the object file format, the executable image file format, and the debugging and traceback information are of no concern to the developer. However, software engineers who develop compilers, debuggers, analysis tools, or other utilities that are dependent on these file formats will need to understand how they are being implemented on OpenVMS.

At publishing time of this white paper, the implementation plans have not been completed; however, documentation will be made available once the design is finalized. See the following URL for the latest information and developments.

http://www.hp.com/products1/evolution/alpha_retaintrust/openvms/index.html

Floating-point Data Types

The Itanium architecture implements floating-point arithmetic in hardware using the IEEE floating-point formats. Currently, OpenVMS Alpha applications can use a number of floating-point formats, some of which have been carried forward from VAX. The list of floating point formats includes F, D, G, S, T, and X.

The Hewlett-Packard GEM-based compilers that are being ported from Alpha will support the same floating-point formats on the Itanium architecture as they do on Alpha. This means that floating-point data that was created by an application running on Alpha or VAX and that was stored on disk can be used by applications running on the Itanium architecture, assuming the applications are compiled using compilers that support the same-floating point format. These formats and their processing actions are as follows:

- The IEEE floating-point formats (S and T) will be processed directly in the hardware.
- F, D, and G formats will most likely be supported by converting the data to IEEE as loads/stores and using the floating-point instructions to do the operations.
- X format (128-bit IEEE) will most likely be supported by performing the operations in software.

Additional information about floating-point formats detailing accuracy and performance on the Itanium architecture will be available in a future document.

Porting Process

This section describes a typical set of tasks that comprise a porting project to a new platform.

1. Assessment of porting needs
2. Compiling/linking/running
3. Unit testing
4. System testing
5. Regression testing
6. Shipment of revised application
7. Operation or installation of processes or procedures
8. Remote support or maintenance

Although each task is completed sequentially, repetition of some steps may be required if problems arise.

Assessment of Porting Needs

The first step in any porting project is the assessment task. All components of the existing application need to be inventoried to determine all prerequisites, dependencies for differences in development, and operational tasks. Source code, infrastructure products, script, and procedures are examples of the areas that need detailed checking for development tasks. Installation requirements, on-disk structure, backup/restore functionality, operator look-and-feel, and system administration are examples of operational tasks. This inventory is then compared to the list of supported product, procedures, and functionality of the target platform. All deviations need to be researched for schedule mismatches, missing functionality, procedural conflicts, and supportability. All costs and time impacts are included in an engineering plan to provide a total picture of all porting issues.

Compiling/Linking/Running

This task involves converting source code modules to a single module or to a set of interdependent executable modules through a series of target-dependent operations. This

conversion is accomplished by a precompiler (in the case of some higher-level languages), a compiler (to create an intermediate object module), a linker (to combine one or a series of object modules), and the loading procedure (to insert the executable image into the running context of the operating system).

Unit Testing

Unit testing is the first level of testing and is usually performed by the source-code development or change engineer. Each addition or change to any part of the source code is exercised by changing the variable being processed to force the execution of the program through the modified code segment. The results of the code segments are checked for correctness, and then the version number of the module is updated to reflect and document the change. The module is then submitted to a code archiving system for distribution, protection, and accountability.

System Testing

When one or more interdependent modules have completed unit testing, the interaction between the modules is tested for correct operation. Specific testing is performed to check that the individual modules pass the correct information between them, and that the combination of modules perform the correct sequence of operations within the allotted time, as determined by the design, specification, and architecture of the application solution.

Regression Testing

Regression testing is a solution-oriented test that exercises the individual modules using a predetermined sequence of operational and performance events. Given a known set of inputs, the solution is run and the output and all operator-visible intermediary results are checked for compliance with a specification and with a previous version of the application. Both correct operations and error paths are checked as part of this testing level.

Shipping the Revised Application

The next release of the application is ready for archival and release to the customer base when:

- The executable part of the application has completion criteria.
- All shippable and engineering documentation is complete.
- All operational-dependent software is available on the target platform.

Operation and Installation of Processes and Procedures

At the customer site, a different set of parameters becomes important. The new revision of the application replaces the current version, but only after the customer performs various tasks specific to the environment. Installation is only a small part when the customer is using the application to run the business. Various degrees of testing should be performed to ensure that the new version is compatible with the customer's unique use of the product. All administrative and operator actions must be addressed by documentation and training. System and data backups are created to ensure that the customer can return to a known reference point if any ambiguities occur in the normal course of business. Parallel operation of using both versions of solution to run the business may be warranted based on the degree of change in the application environment (hardware or software), the criticality of the application to conduct business as usual, or just the large number of people and information that the application affects. Extra procedures must be instituted when the change invalidates any customer-retained archives that are required for legal or business considerations, such as accessibility to backup media.

Remote Support and Maintenance Considerations

The effort to maintain and support the customer base will be increased at the time of product revisions. New or modified customer-initiated procedures may create special needs for the support staff. Existing remote diagnostic techniques and tools may require changes to support the new revision and may necessitate training of both the customer and support staff resources.

Porting Checklist

Appendix A of this document consists of a porting checklist. In porting projects many issues and tasks are often overlooked that can cause schedules and costs to escalate beyond original estimates. Completing this checklist will help customers and ISVs assess the effort necessary for porting applications from Alpha to the Intel® Itanium® architecture.

Porting Issues

Hewlett-Packard intends to maintain a strict policy of ensuring and maintaining forward source compatibility so that “well-behaved” applications that currently run on recommended versions of OpenVMS Alpha run successfully on OpenVMS Industry Standard 64. ISVs and customers who take advantage of published system services and library interfaces can have a high level of assurance that their applications will move without modification to the latest version of OpenVMS Industry Standard 64.

OpenVMS applications will continue to have the same look and feel that customers have come to know and like. Changes will be necessary to accommodate the new architecture, but the basic structure and functionality of OpenVMS will remain the same. OpenVMS will share a common source pool for both the Alpha and Itanium architectures, with minimal conditional code and build procedures.

Similarly, most applications running on OpenVMS Alpha today will port easily to the Itanium architecture with few changes required to the source code or build procedures. Developers will need to simply compile and link their applications and then perform some level of testing.

Until OpenVMS is running on an Itanium-based system with the appropriate compilers, developers can examine their code to determine whether portions will require modification. This section provides a set of questions and guidelines to help developers decide the state of the code.

Architecture-specific Code

Developers will need to rewrite any code that is dependent on the Alpha architecture. For example, assembly language code that deals with specific machine instructions or that makes assumptions about the number of registers or the functions of specific registers will have to be modified. This type of architecture-specific code is usually not found in applications written in high-level languages.

Object File Format Code

Any applications that deal with the object or executable image file formats will have to be modified to handle the ELF and DWARF formats. This type of code is typically found in compilers, debuggers, and analysis tools that directly process or manipulate object files or executable images. For more information, see the Floating-point Data Types section on page 7.

Alpha or VAX Logic

Some code may include logic that assumes it is running on either a VAX or an Alpha System. Such code might not be able to deal with a third platform. For example, a build command procedure might contain the following statement:

```
$ if .not. Alpha
$ then
$ ! execute VAX specific commands
$ endif
```

Similar code might exist in a C or a C++ application. For example:

```
#ifndef VAX/* Alpha specific code
#endif
```

In either case, when these files are moved to the Itanium architecture, they may have to be modified to work correctly.

Porting Resources

Because no one organization will have expertise in all phases of the porting process, Hewlett-Packard has gathered a set of resources that can assist ISVs and customers as they go through the process. These resources can provide everything from porting guides to teams of personnel with equipment that will handle the entire porting task. A subset of these resources includes:

- Developer & Solution Partner Program (DSPP) <http://www.hp.com/go/dspp>
- Test Drive Program <http://www.testdrive.hp.com/>
- Hewlett-Packard Services <http://www.hp.com/go/services/>

As Hewlett-Packard's strategy related to Itanium architecture continues to develop, Sales Resources and Global Services will be assessing the impact to the long-term plans of Hewlett-Packard customers and ISVs. A number of template and custom service offerings will be created and documented to address generic and unique situations of each customer. Each service offering will be built to support the future directions of the customers by making sure there are tools and resources available to assist with a change whenever a decision point is reached.

Additional Dependencies and Considerations

Hewlett-Packard recognizes that not all layered software and middleware supplier products will port to Itanium -based platforms at first release. It will be in Hewlett-Packard's best interest to support the porting of all software to the Itanium architecture through a variety of techniques and tools. Hewlett-Packard is committed to supporting these software vendors with information, hardware, support and tools that will make this process as simple and straightforward as possible. Each vendor will have to monitor the progress of their supplier of prerequisite software to ensure that the required components are in place for their own development schedules.

Hewlett-Packard's commitment is to implement a tool set on the existing Alpha platform that will make the porting to the new platforms transparent for most developers in terms of code base, commands, and process. Nothing will replace the unit test, system test, and regression test that accompany any quality product release. Rather, this tool suite will reduce the time required for product porting.

The following web site provides a list of partners to date who have committed to porting their products to OpenVMS on the Itanium architecture:

http://www.hp.com/products1/evolution/alpha_retaintrust/openvms/partners.html

For More Information

Visit the following web sites for additional information:

- | | |
|---------------------------------------|---|
| Hewlett-Packard and Intel Initiative | http://www.hp.com/products1/evolution/alpha_retaintrust/index.html |
| CETS2001 presentations | http://www.OpenVMS.compaq.com/presentations/Cets2001/index.html |
| Intel Training Center | http://developer.intel.com/software/products/itc/architec/itanium/itan.htm |
| Intel® Itanium® architecture web site | http://intel.com/design/itanium/family |
| OpenVMS Systems | http://www.hp.com/go/openvms |

Appendix A: Porting Checklist

This checklist indicates the minimum set of information that customers and ISVs must collect before starting the porting effort. You can assess your porting effort based on the results of these questions.

- In one sentence, what does the application do?
- On what operating system and version does the current application environment run?
(Example: OpenVMS VAX 7.1, OpenVMS Alpha 7.3)
- To what operating system will the application be ported?
- List all languages and versions that were used to construct the application:
(Examples: Pro COBOL Version 2.7, Compaq C version 6.4 , Compaq Fortran V6.6,)

Language	Version	Comments

If more languages were used, list them on a separate page.

- List all Hewlett-Packard layered and third-party products the application is dependent on for proper operation
(Examples: Rdb, Oracle, COBOL PreCompiler, Compaq MessageQ, SMG, DECforms)

Language	Version	Comments

If more products were used, list them on a separate page.

- Are all source code files available to build the application? Yes No

Porting HP OpenVMS Applications to the Intel® Itanium® Architecture

- How many disk blocks or MBs of source code files are there? _____

- When was the last time the application environment was completely recompiled and rebuilt?

- If the application is re-built more than once a quarter, how often is the application built?

- How is the application tested or verified for proper operation after it is rebuilt?

- Which source-code management tool is used, if any?

- Do you have a separate development and test environment from which the application updates and tests?
Yes No

- What is the primary reason for porting the application?

© 2003 Copyright Hewlett-Packard Development Company, L.P
All brand names are trademarks of their respective owners.
Technical information in this document is subject to change without notice.

Intel® and Itanium® are trademarks of Intel Corporation in the US and other countries and are used under license. All other product names mentioned herein may be the trademarks of their respective companies.

Neither HP nor any of its subsidiaries shall be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for HP products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

