OpenVMS VAX/Alpha to Integrity Migration Experiences

Thomas Siebold
Senior Technology Consultant
thomas.siebold@hp.com
+49-(0)89-9392.4201

GET CONNECTED
People. Training. Technology.

Anonymous

“Most people don’t
plan to fail,…
they fail to plan.

Die meisten Menschen
planen nicht zu versagen,…
Sie versagen beim Planen.

Top 10 Porting Considerations…

- Do a complete inventory of all 3rd party software products and HP OpenVMS layered products before you start your port. These may be required for development, QA, or production deployment. Ensure you know the status of each of these on OpenVMS I64 before you go too far in your port.
- Make sure your application builds cleanly and runs on OpenVMS Alpha V8.2 (or greater) using the latest released compilers and development tools
- Check for hardware architecture consistently in all source code and DCL command procedures
- Have automated regression tests as much as possible and clearly documented manual regression tests where necessary
- Document your build procedure / process
- Read the Porting Guide and various Release Notes (Really do it!)
- Update any Fortran 77 code to Fortran 90
- Reduce / Recode / eliminate as much Alpha Macro (Macro64) code as possible
- Where possible, use IEEE floating point
- Have a working development / QA environment on OpenVMS Alpha near by so you can compare results easily between Alpha and Integrity systems.
- Sit back and just...

(re)compile

(re)link

(re)qualify

and run

done....

The information contained herein is subject to change without notice

Compiler Differences

Compiler Migration at a glance

Alpha		Porting	Integrity
Compiler	**Version**	Action	Version
C	**V6.5**	Ported	V7.2
C++	**V6.5**	New from Intel	V7.2
Fortran 77		Not Ported	
Fortran 90	**V7.5**	Ported	V8.1
Cobol	**V2.8**	Ported	V2.8
Basic	**V1.5**	Ported	V1.5
Pascal	**V5.9**	Ported	V6.0
Java	**V1.4.2**	Implemented	V1.4.2-5
ADA 83		Not Ported	
ADA 95		New from ACT	
AMacro		IMacro created	
BLISS	**V1.01**	Ported	V1.01
Macro64		Not Ported	
IAS	**N/A**	Available	v7.0U (7.00.4160)
Dibol		Ported by Synergex	
Acucorp Cobol		Ported by Acucorp	
PL/I		Not Ported	

C

- HP C V7.2 for OpenVMS I64
- Language features and command line options are the same as HP C V6.5 for OpenVMS Alpha
- Compiler is installed using PRODUCT INSTALL
  - Alpha compiler installed using VMSINSTAL
- /ARCH and /OPTIMIZE=TUNE qualifiers are accepted but Alpha-only arguments and ignored
  - Allows existing command files to continue to work
- Inline assembly language code is not supported
- `#pragma linkage` maps the Alpha registers to their corresponding I64 registers
- #pragma linkage_alpha and #pragma linkage_ia64 have been added
  - Used to specify the platform specific register names to use

C

- Use /FLOAT= qualifier to use VAX floating point format
- /FLOAT=IEEE_FLOAT and
  IEEE_MODE= DENORM_RESULTS
  are the floating point defaults
- Compiler predefined macros:
  __ia64 and __ia64__
  - Defining the macro __ALPHA as a quick “hack” to deal with conditional code will cause problems with the CRTL and OpenVMS headers on OpenVMS I64. Do not attempt to use this trick

C++

- HP C++ V7.2 for OpenVMS I64
- This is new compiler technology that differs substantially from HPC++ and HPC for OpenVMS Alpha
  - Mostly source compatible with HP C++ V6.5 but there are some differences
- Compiler is installed using PRODUCT INSTALL
  - Alpha compiler installed using VMSINSTAL
- Inline assembly language code is not supported
- /STANDARD=CFRONT is not supported
- The object model and name mangling scheme are different than on Alpha

C++

- Command line differences
  - Comma lists are not supported
  - /INSTRUCTION_SET=NOFLOATING_POINT is not supported
  - /L_DOUBLE_SIZE=64 is not supported;
    /L_DOUBLE_SIZE=128 is used
  - /POINTER_SIZE=(LONG,64) is now supported
  - Use /WARN=ENABLE=QUALCHANGE and =QUALNA to identify/fix qualifier problems
  - /FLOAT=IEEE_FLOAT and
    IEEE_MODE= DENORM_RESULTS
    are the floating point defaults

C++

- New Features
  - cname header support
    - 18 <cname> headers from the C++ standard
  - /[NO]FIRST_INCLUDE qualifier
    - Corresponds to the Tru64 UNIX –FI switch
  - #pragma include_directory
  - New front end improves conformance to the C++ International Standard

C++

- Most existing Alpha builtins should continue to work but the compiler will issue a diagnostic message where a different builtin would be preferable
  - A significant number of __PAL builtins are implemented as system services
  - The compiler generates the code to call the appropriate system service
  - Builtins that take a retry count provoke a warning and are ignored
    - Due to absence of the load-locked/store-conditional sequences on Integrity systems
    - __LOCK_LONG_RETRY and __ACQUIRE_SEM_LONG_RETRY do still work the same
  - __CMP_STORE_LONG and __CMP_STORE_QUAD produce either a warning or an error depending on whether or not the source and destination addresses are identical
  - Consult builtins.h and pal_builtins.h for details

C++

- Template instantiation
  - Alpha had numerous models – I64 only uses COMDAT section groups
  - Similar to /TEMPLATE=LOCAL on Alpha except that the linker removes duplicate copies, resulting in a reduction in the image size
  - You’ll see little differences if you’re using /TEMPLATE=LOCAL or /TEMPLATE=IMPLICIT_LOCAL on Alpha
  - No repository is needed. Builds that manipulate objects in the repository will need to be changed

FORTRAN

- HP FORTRAN V8.1 for OpenVMS I64
- Language features and command line options are the same as HP FORTRAN for OpenVMS Alpha
- Use /FLOAT qualifier to use VAX floating point format
- /FLOAT=IEEE_FLOAT and
  IEEE_MODE= DENORM_RESULTS
  are the floating point defaults
- /OLD_F77 switch is no longer supported
  - FDML and CDD support from the F77 compiler has been added to this compiler
- /ARCH and /TUNE qualifiers are accepted and ignored
  - Allows existing command files to continue to work

The information contained herein is subject to change without notice

COBOL

- HP COBOL V2.8 for OpenVMS I64
- Language features and command line options are the same as HP COBOL V2.8 for OpenVMS Alpha
- In case you experience linker errors (exceeding short data), please contact HP

The information contained herein is subject to change without notice

Pascal

- HP Pascal V6.0 for OpenVMS I64
- Designed to be 100% source compatible with HP Pascal for OpenVMS Alpha
- Use /FLOAT=G_FLOAT or /FLOAT=D_FLOAT qualifiers or [FLOAT] module-level attribute to use VAX floating point format

BASIC

- HP BASIC V1.6 for OpenVMS I64
- Designed to be 100% source compatible with HP BASIC for OpenVMS Alpha
- Use /REAL_SIZE=(SINGLE | DOUBLE | GFLOAT) qualifier or the OPTION SIZE=REAL (SINGLE | DOUBLE | GFLOAT) statement to use VAX floating point format

The information contained herein is subject to change without notice

DIBOL

- Synergex Synergy/DE is DIBOL compiler on OpenVMS, both Alpha and I64
- Designed to be 100% source code compatible
- Millions of lines of code ported with no changes required

MACRO

- HP MACRO Compiler for OpenVMS I64
- The HP MACRO Compiler performs several transformations to allow most existing code to compile unmodified on OpenVMS I64
  - The compiler deals with differences in the calling standard and register usage
- Programs that use nonstandard return values or programs that use the JSB instruction to call routines written in other languages must use some new directives in the MACRO source code
- See the HP OpenVMS MACRO Compiler Porting and User’s Guide for porting details

Ada

- GNAT Pro 5.04 for OpenVMS on HP Integrity Servers
- Based on gcc technology
- Handles Ada 83, Ada 95, and many Ada 2005 language features
- Has optional HP Ada predefined library interface
- Supports 64-bit addresses
- “Source based” compilation model
  - eliminates the Aad 83 style program library
- Debug using OpenVMS debugger
- Available from and supported by AdaCore – http://www.adacore.com

The information contained herein is subject to change without notice

Binary Translator

- Translates Alpha OpenVMS binary images and libraries linked under all OpenVMS versions from 6.2 to current version
- Translates a VESTed image that was translated by DECmigrate from a VAX binary image
- Translates images written in C, C++, FORTRAN, COBOL, or Pascal (as of OpenVMS V8.3)
  - Does not translate applications written in BASIC, PL/1, or Ada
- Restrictions:
  - Alpha binary code
  - Only user-mode apps
  - No privileged instruction
  - No self-modifying code
  - No sys. Memory space reference
  - No user-written system services

Miscallenous Considerations

Infrastructure changes in OpenVMS V8.2

- We made changes to some system level data structures in OpenVMS V8.2 (Alpha and I64)
- Benefits
  - Laying the foundation for scalability and performance improvements in future releases of OpenVMS
- Impact to applications
  - **Non-privileged applications are not affected**
  - Applications that access the modified data structures in non-standard ways may need to be modified
    - Examples: hard-coded data structure sizes and assumptions about the relative locations of fields within a data structure

Infrastructure changes in OpenVMS V8.2

- Impact to applications (continued)
  - Some privileged applications (such as device drivers) will need to be recompiled and relinked
    - Privileged applications in this case are images linked against the system using the /SYSEXE qualifier and reference the changed data structures or related structures and routines
    - Attempting to execute or load such an image that has not been rebuilt will result in an error during image activation of SYSVERDIF – “System Version Mismatch”.

Major Porting Considerations

- New Calling Standard
  - Publicly available today at http://www.hp.com/products1/evolution/alpha_retaintrust/openvms/resources.html
  - Intel® calling standard with OpenVMS modifications
    - No frame pointer (FP)
    - Multiple stacks
    - only 4 preserved registers across calls
    - Register numbers you’re familiar with will change
  - All OpenVMS provided tools “know” about these changes
  - Most user applications are not affected
  - Your code that “knows” about the Alpha standard will almost certainly need to change

Major Porting Considerations

- Object file format
  - ELF/DWARF industry standards plus our extensions
    - ELF - Executable and Linkable Format, Itanium® architecture object code, images, etc.
    - DWARF - Debugging and traceback information (embedded in ELF).
  - All OpenVMS provided tools “know” about these changes
  - Most user applications are not affected
  - User written code that “knows” the object file format may have to change
  - Specifications are available on the WEB

The information contained herein is subject to change without notice

Major Porting Considerations

- Floating point data types
  - Itanium® architecture supports IEEE float only; Alpha supports IEEE and VAX Float
  - All compilers that currently support F, D, G, S, T, and X (S and T are native IEEE formats) will continue to do so on Itanium architecture
  - IEEE is the default
  - We have updated the appropriate Runtime Libraries to add IEEE interfaces where needed
  - White Paper with technical details about the differences between VAX Float and IEEE Float is available at http://www.hp.com/products1/evolution/alpha_retaintrust/openvms/resources.html

The information contained herein is subject to change without notice

Major Porting Considerations

Source Code that May Need to Change

- Architecture Specific code
  - All Alpha assembler code must be rewritten
- Conditionalized code
  - Build command files
    - `$ if .not. Alpha  ! Assumes VAX`
  - Application source code
    - `#ifndef (alpha)  // Assumes VAX`
    - C asm code

Major Porting Considerations

Source Code that May Need to Change

- SS$_HPARITH (high performance arithmetic trap) is replaced by SS$_FLTINV (floating point invalid) and SS$_FLTDIV (floating divide by zero)
  - To maintain common code use:

    ```
    if ((sigargs[1] == SS$_HPARITH) || (sigargs[1] ==
    SS$_FLTINV) || (sigargs[1] == SS$_FLTDIV))
    ```
- Mechanism Array data structure has been changed
  - Standard calling interfaces have not changed
- The Porting Guide contains all of the details

Major Porting Considerations

- Improperly declared functions and data
- C function declarations that points to objects that are not functions, may work on Alpha but these declarations will not work on IA64
  - Also seen with the Bliss compiler
- This problem may manifest itself in many ways
  - For whatever reason, the most common symptom is routine CLI$DCL_PARSE failing with CLI-E-INVTAB
  - In case of a failure the command table is usually defined as `int master_cmd();`
Change to `extern master_cmd;`
and change the way the parameter is passed to `cli$dcl_parse` from `master_cmd` to `&master_cmd`

Porting Profiles

C	35,090,958	
C++	2,878,485	
Fortran	8,133,031	
Basic	31,987,164	
Cobol	6,133,000	(+25-30 Mill.)
Pascal	3,822,742	
Java	409,500	
Macro32	4,090,493	
Macro64	14,561	
ADA	1,220,000	
PL/I	440,000	
Cache'	8,716,945	
Dibol	700,000	
BLISS	150,000	

Over 100,000,000 lines of code and still counting

What we learned

- FORTRAN 77 is not available on OpenVMS I64
  - Customers need to migrate from F77 to F90, preferably on Alpha prior to starting the port
- MACRO-32 (VAX MACRO)
  - Read the Macro-32 Porting Guide first
  - Pay attention to CALL LINKAGES
- Alpha Assembly (MACRO-64) code MUST be rewritten
- Strongly suggest that customers use supported, documented interfaces. Tricks will work but it will be difficult to port them

What we learned

- Generate .MAP and .LIS files on Alpha prior to starting the port
  - Useful to find APIs, match PSECT attributes, identify modules needed to build the application
- Don’t assume if an application runs on Alpha that it is correct
  - Some partners uncovered day-one bugs
- Increase quotas in SYSUAF on IPF to do compiles and links
  - BYTLM, FILLM, WSDEF, WSQUO, WSEXTENT, PGFLQUOT
  - Multiply by 5 then adjust

How can you make your port simple?

- Get current on OpenVMS Alpha
  - Recommend upgrading to V8.2 or V8.3
  - Upgrade to latest compilers and layered products
- Examine dependencies and check to make sure layered products and 3rd party products are available on OpenVMS I64
- If possible, build and run solution on “fresh” Alpha environment before porting to I64
- If application came from VAX to Alpha, check for architectural / conditional code

Tools and other Help

SEARCH_ALL.COM

SEARCHALL.COM

- Searches all files in a directory tree
- Looks for items of interest in porting from OpenVMS Alpha to OpenVMS I64
- Using it for OpenVMS VAX requires editing to add VAX specific keywords
- Looks for C, C++, Cobol, Pascal, Basic, Fortran, Macro, Map files
- Uses a "companion" file called searchall.lis
  - contains a list of key words and items that should be looked at
- Example from `searchall.lis`:
  - SS$_HPARITH
  - LIB$WAIT
  - __Alpha
  - __VAX
  - ....

OPENVMS TRANSITION MODULES

Transition Modules

- Web based tool
- Contain
  - HTML pages
  - Pointers to HP web pages and online documentation
  - Documents in PDF and HTM format
- Areas covered (= *Module*)
  - platform infrastructure
  - custom code applications
  - packaged applications (ISVs)
  - databases (Oracle, RDB)
- Focus is *PLANNING*

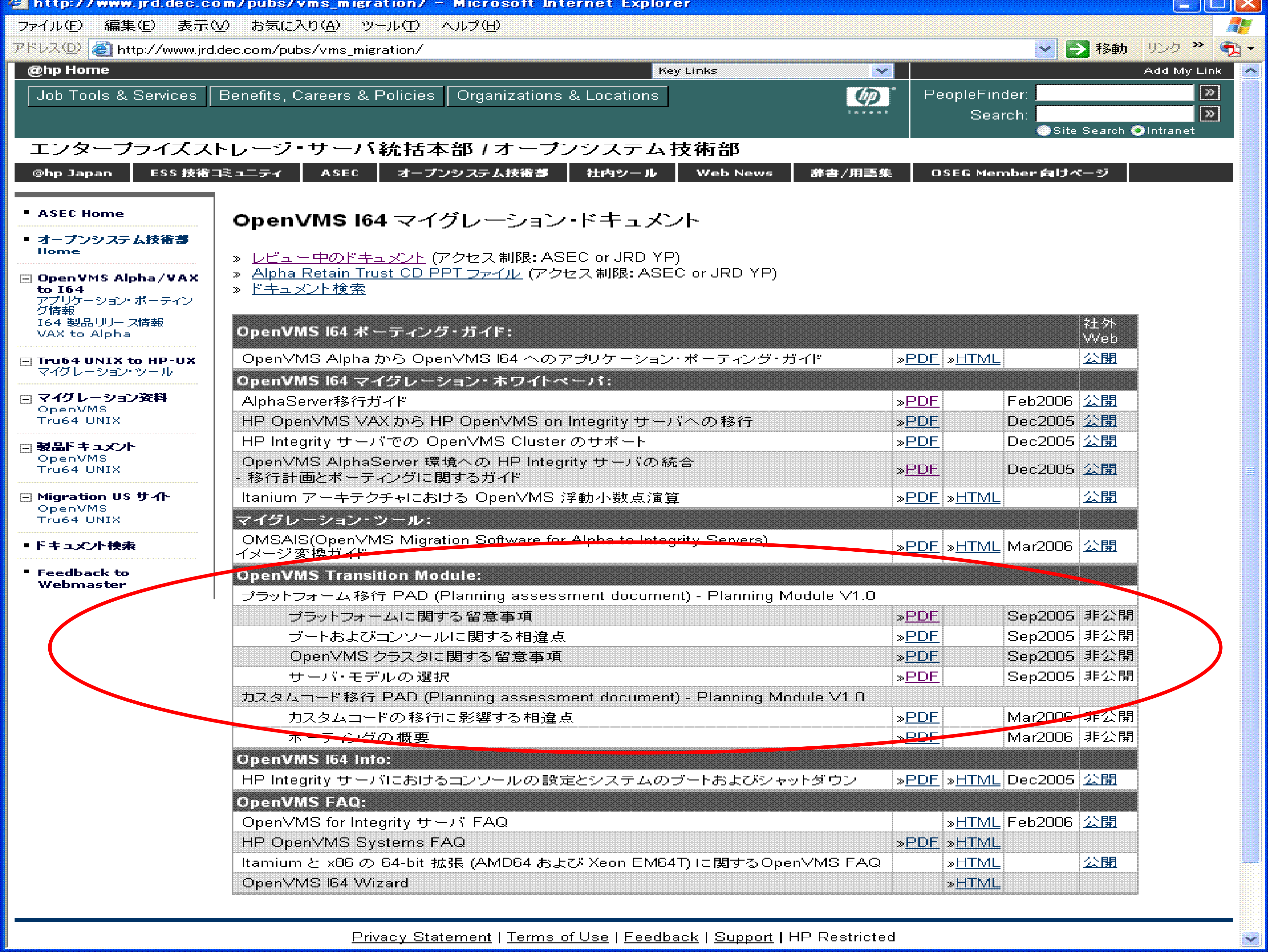
Individual module structure

- Each module has the following sections
- Planning assessment documents
  - Overview, analysis and recommendations of key functional areas of your platform infrastructure transition
- Supporting documentation & training
  - Guides, white papers, best practice documents, web pages, training, and webcasts that are directly related to platform infrastructure transition to HP Integrity Servers.
- Transition tools
  - HP tools to help you assess your platform infrastructure transition.
- Release notes & revision history
  - Provides module release notes and revisions.
- For more information
  - Access to contact information.
- Feedback
  - Mechanism for providing feedback about whether this module meets your transition planning needs.

Which Transition Modules are available?

- OpenVMS Alpha to OpenVMS I64 V1.1
  - Since July 2005 V1.0
  - Since May 2006 V1.1
  - ≈ 23MB
- OpenVMS Vax to OpenVMS I64 V1.0
  - Since May 2006 V1.0
  - ≈ 4MB

The information contained herein is subject to change without notice



エンタープライズストレージ・サーバ統括本部 / オープンシステム技術部

- ASEC Home
- オープンシステム技術部 Home

⊟ OpenVMS Alpha/VAX to I64
アプリケーション・ポーティング情報
I64 製品リリース情報
VAX to Alpha

⊟ Tru64 UNIX to HP-UX
マイグレーション・ツール

⊟ マイグレーション資料
OpenVMS
Tru64 UNIX

⊟ 製品ドキュメント
OpenVMS
Tru64 UNIX

⊟ Migration US サイト
OpenVMS
Tru64 UNIX

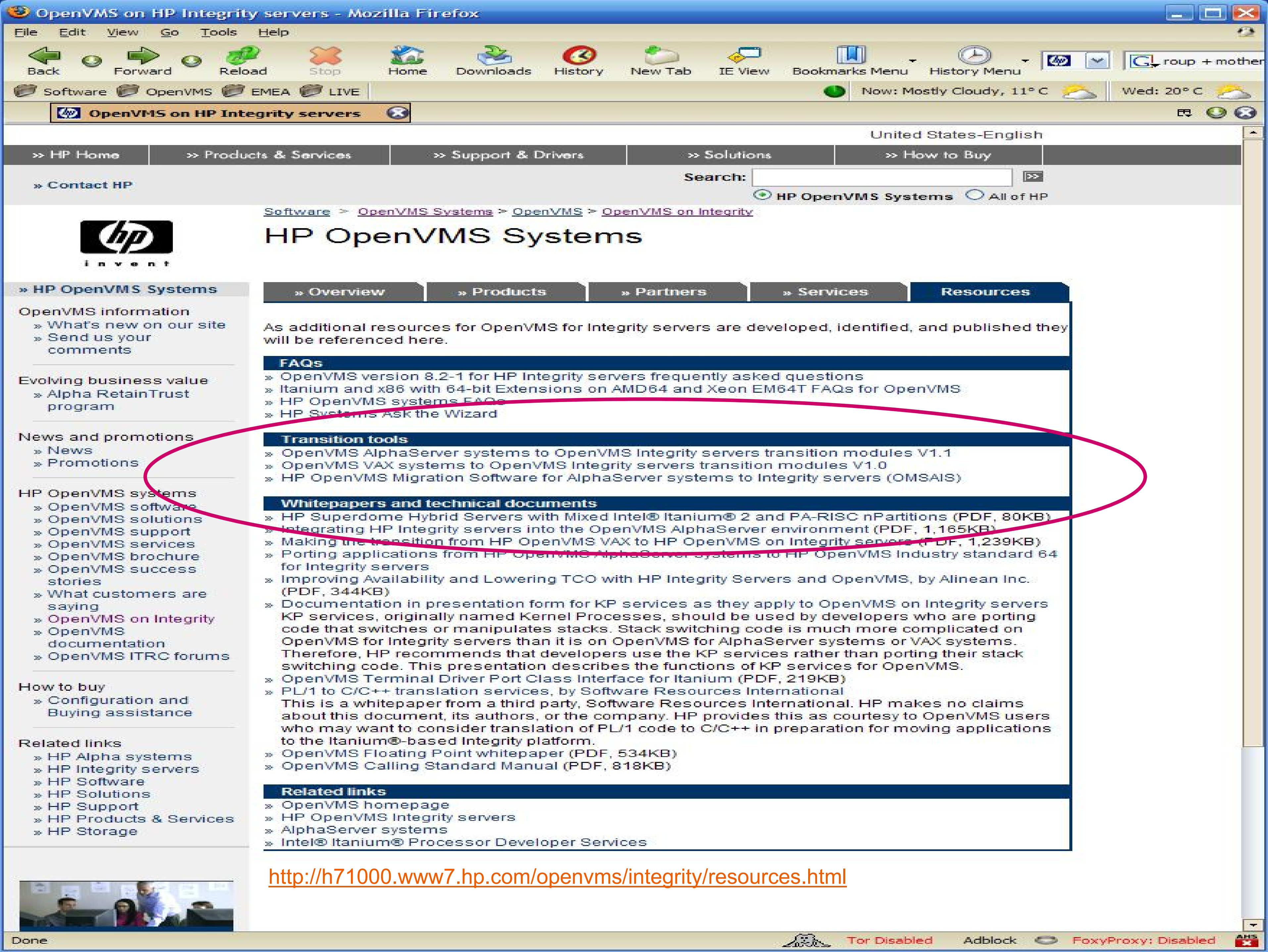
- ドキュメント検索
- Feedback to Webmaster

OpenVMS I64 マイグレーション・ドキュメント

» レビュー中のドキュメント (アクセス 制限: ASEC or JRD YP)
» Alpha Retain Trust CD PPT ファイル (アクセス 制限: ASEC or JRD YP)
» ドキュメント検索

OpenVMS I64 ポーティング・ガイド:				社外 Web
OpenVMS Alpha から OpenVMS I64 へのアプリケーション・ポーティング・ガイド	»PDF	»HTML		公開
OpenVMS I64 マイグレーション・ホワイトペーパ:				
AlphaServer移行ガイド	»PDF		Feb2006	公開
HP OpenVMS VAX から HP OpenVMS on Integrity サーバへの移行	»PDF		Dec2005	公開
HP Integrity サーバでの OpenVMS Cluster のサポート	»PDF		Dec2005	公開
OpenVMS AlphaServer 環境への HP Integrity サーバの統合 - 移行計画とポーティングに関するガイド	»PDF		Dec2005	公開
Itanium アーキテクチャにおける OpenVMS 浮動小数点演算	»PDF	»HTML		公開
マイグレーション・ツール:				
OMSAIS(OpenVMS Migration Software for Alpha to Integrity Servers) イメージ変換ガイド	»PDF	»HTML	Mar2006	公開
OpenVMS Transition Module:				
プラットフォーム移行 PAD (Planning assessment document) - Planning Module V1.0				
プラットフォームに関する留意事項	»PDF		Sep2005	非公開
ブートおよびコンソールに関する相違点	»PDF		Sep2005	非公開
OpenVMS クラスタに関する留意事項	»PDF		Sep2005	非公開
サーバ・モデルの選択	»PDF		Sep2005	非公開
カスタムコード移行 PAD (Planning assessment document) - Planning Module V1.0				
カスタムコードの移行に影響する相違点	»PDF		Mar2006	非公開
ポーティングの概要	»PDF		Mar2006	非公開
OpenVMS I64 Info:				
HP Integrity サーバにおけるコンソールの設定とシステムのブートおよびシャットダウン	»PDF	»HTML	Dec2005	公開
OpenVMS FAQ:				
OpenVMS for Integrity サーバ FAQ		»HTML	Feb2006	公開
HP OpenVMS Systems FAQ	»PDF	»HTML		
Itamium と x86 の 64-bit 拡張 (AMD64 および Xeon EM64T) に関するOpenVMS FAQ		»HTML		公開
OpenVMS I64 Wizard		»HTML		

Privacy Statement | Terms of Use | Feedback | Support | HP Restricted



United States-English

» HP Home | » Products & Services | » Support & Drivers | » Solutions | » How to Buy

» Contact HP

Search: ○ HP OpenVMS Systems ○ All of HP

Software > OpenVMS Systems > OpenVMS > OpenVMS on Integrity

HP OpenVMS Systems

» Overview | » Products | » Partners | » Services | Resources

» HP OpenVMS Systems

OpenVMS information
- » What's new on our site
- » Send us your comments

Evolving business value
- » Alpha RetainTrust program

News and promotions
- » News
- » Promotions

HP OpenVMS systems
- » OpenVMS software
- » OpenVMS solutions
- » OpenVMS support
- » OpenVMS services
- » OpenVMS brochure
- » OpenVMS success stories
- » What customers are saying
- » OpenVMS on Integrity
- » OpenVMS documentation
- » OpenVMS ITRC forums

How to buy
- » Configuration and Buying assistance

Related links
- » HP Alpha systems
- » HP Integrity servers
- » HP Software
- » HP Solutions
- » HP Support
- » HP Products & Services
- » HP Storage

As additional resources for OpenVMS for Integrity servers are developed, identified, and published they will be referenced here.

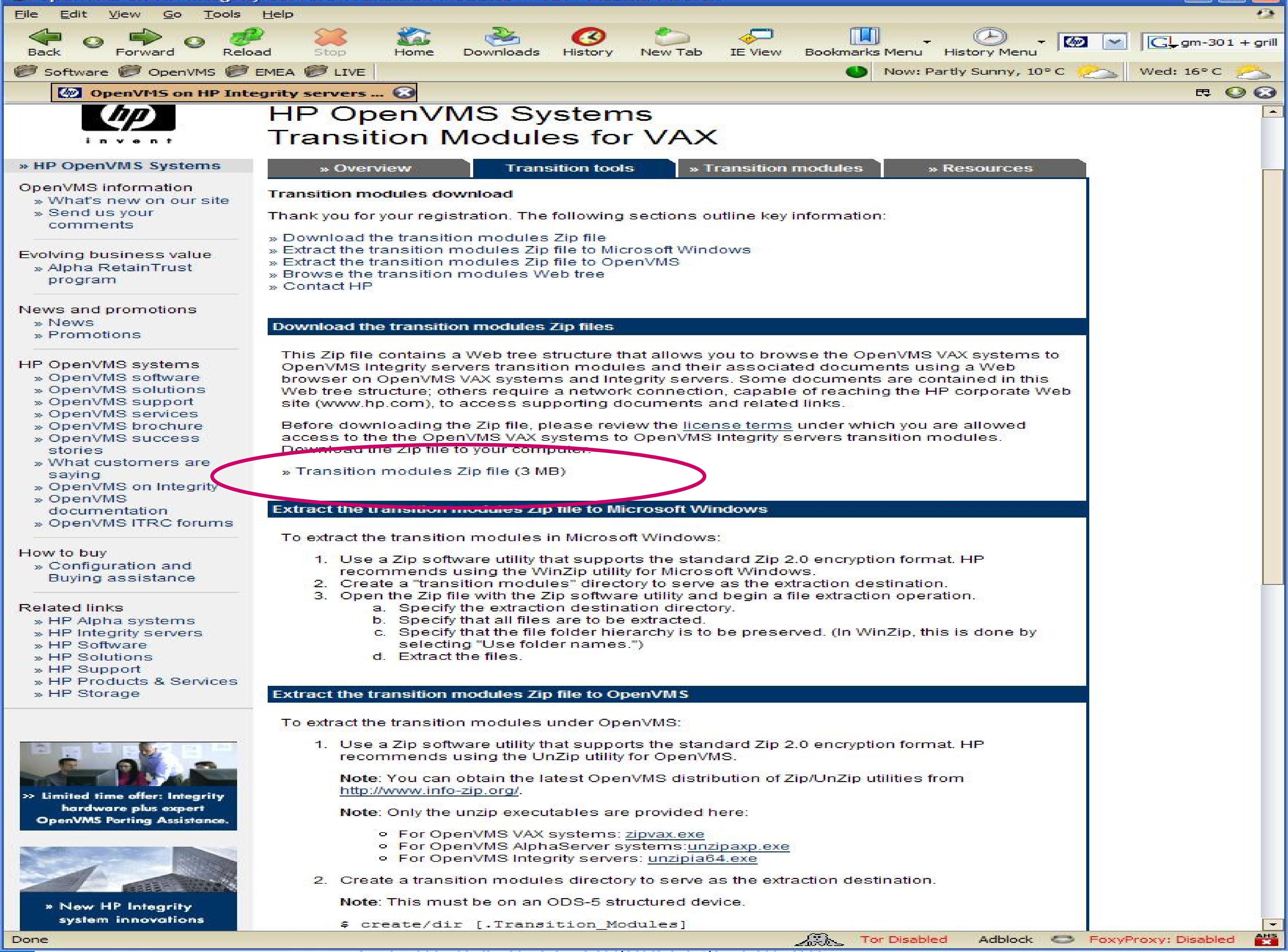
FAQs
- » OpenVMS version 8.2-1 for HP Integrity servers frequently asked questions
- » Itanium and x86 with 64-bit Extensions on AMD64 and Xeon EM64T FAQs for OpenVMS
- » HP OpenVMS systems FAQs
- » HP Systems Ask the Wizard

Transition tools
- » OpenVMS AlphaServer systems to OpenVMS Integrity servers transition modules V1.1
- » OpenVMS VAX systems to OpenVMS Integrity servers transition modules V1.0
- » HP OpenVMS Migration Software for AlphaServer systems to Integrity servers (OMSAIS)

Whitepapers and technical documents
- » HP Superdome Hybrid Servers with Mixed Intel® Itanium® 2 and PA-RISC nPartitions (PDF, 80KB)
- » Integrating HP Integrity servers into the OpenVMS AlphaServer environment (PDF, 1,165KB)
- » Making the transition from HP OpenVMS VAX to HP OpenVMS on Integrity servers (PDF, 1,239KB)
- » Porting applications from HP OpenVMS AlphaServer systems to HP OpenVMS Industry standard 64 for Integrity servers
- » Improving Availability and Lowering TCO with HP Integrity Servers and OpenVMS, by Alinean Inc. (PDF, 344KB)
- » Documentation in presentation form for KP services as they apply to OpenVMS on Integrity servers
  KP services, originally named Kernel Processes, should be used by developers who are porting code that switches or manipulates stacks. Stack switching code is much more complicated on OpenVMS for Integrity servers than it is on OpenVMS for AlphaServer systems or VAX systems. Therefore, HP recommends that developers use the KP services rather than porting their stack switching code. This presentation describes the functions of KP services for OpenVMS.
- » OpenVMS Terminal Driver Port Class Interface for Itanium (PDF, 219KB)
- » PL/1 to C/C++ translation services, by Software Resources International
  This is a whitepaper from a third party, Software Resources International. HP makes no claims about this document, its authors, or the company. HP provides this as courtesy to OpenVMS users who may want to consider translation of PL/1 code to C/C++ in preparation for moving applications to the Itanium®-based Integrity platform.
- » OpenVMS Floating Point whitepaper (PDF, 534KB)
- » OpenVMS Calling Standard Manual (PDF, 818KB)

Related links
- » OpenVMS homepage
- » HP OpenVMS Integrity servers
- » AlphaServer systems
- » Intel® Itanium® Processor Developer Services

http://h71000.www7.hp.com/openvms/integrity/resources.html



HP OpenVMS Systems Transition Modules for VAX

Transition modules download

Thank you for your registration. The following sections outline key information:

» Download the transition modules Zip file
» Extract the transition modules Zip file to Microsoft Windows
» Extract the transition modules Zip file to OpenVMS
» Browse the transition modules Web tree
» Contact HP

Download the transition modules Zip files

This Zip file contains a Web tree structure that allows you to browse the OpenVMS VAX systems to OpenVMS Integrity servers transition modules and their associated documents using a Web browser on OpenVMS VAX systems and Integrity servers. Some documents are contained in this Web tree structure; others require a network connection, capable of reaching the HP corporate Web site (www.hp.com), to access supporting documents and related links.

Before downloading the Zip file, please review the license terms under which you are allowed access to the the OpenVMS VAX systems to OpenVMS Integrity servers transition modules. Download the Zip file to your computer:

» Transition modules Zip file (3 MB)

Extract the transition modules Zip file to Microsoft Windows

To extract the transition modules in Microsoft Windows:

1. Use a Zip software utility that supports the standard Zip 2.0 encryption format. HP recommends using the WinZip utility for Microsoft Windows.
2. Create a "transition modules" directory to serve as the extraction destination.
3. Open the Zip file with the Zip software utility and begin a file extraction operation.
   a. Specify the extraction destination directory.
   b. Specify that all files are to be extracted.
   c. Specify that the file folder hierarchy is to be preserved. (In WinZip, this is done by selecting "Use folder names.")
   d. Extract the files.

Extract the transition modules Zip file to OpenVMS

To extract the transition modules under OpenVMS:

1. Use a Zip software utility that supports the standard Zip 2.0 encryption format. HP recommends using the UnZip utility for OpenVMS.

   Note: You can obtain the latest OpenVMS distribution of Zip/UnZip utilities from http://www.info-zip.org/.

   Note: Only the unzip executables are provided here:

   - For OpenVMS VAX systems: zipvax.exe
   - For OpenVMS AlphaServer systems:unzipaxp.exe
   - For OpenVMS Integrity servers: unzipia64.exe

2. Create a transition modules directory to serve as the extraction destination.

   Note: This must be on an ODS-5 structured device.

```
$ create/dir [.Transition_Modules]
```

- *How do they look?*

- *Tested with Internet Explorer 6 & 7, Firefox, Opera*

You don‘t have to buy them --- they are FREE!

- No needed

- Just download them !!

How can you provide feedback?

- Send an eMail to transition-modules@hp.com
- Send an eMail to thomas.siebold@hp.com

Questions ?

invent