The OpenVMS Frequently Asked Questions (FAQ)

ABSTRACT

This document contains answers to many common questions on the OpenVMS operating system, and covers OpenVMS, and OpenVMS running on Itanium, Alpha, and VAX hardware.

While this document is available in multiple formats, the text-format version is far and away generally the easiest format to search.

Revision/Update Information OpenVMS V8.3

Revision/Update Information: Revision Date/July 2006

FAQ

Contents

INDEX

TABLES

Preface

OpenVMS Frequently Asked Questions (FAQ)

This is the OpenVMS Frequently Asked Questions (FAQ) posting for the comp.os.vms and comp.sys.dec usenet newsgroups.

Suggestions and Updates Welcome!

> **Please send your suggestions for changes, additions, or corrections directly to FAQ Editor (hoff{atsign}hp{period}com)**

No Support Questions, Please

> **Please do *not* send technical questions to the FAQ Editor (hoff{atsign}hp{period}com) . The FAQ Editor (hoff{atsign}hp{period}com) is *not* in a position to answer general questions, nor to provide general product support. Rather, please post your questions to the appropriate newsgroup, or please contact your preferred hardware and/or software support organization(s) directly.**
>
> **Your understanding in this matter is greatly appreciated.**

If you believe any particular discussion worthy of inclusion in a future edition of the FAQ and particularly once you have or know both the question *and* the answer, please then send the entire discussion along to the FAQ Editor (hoff{atsign}hp{period}com) for potential inclusion.

Do note that the FAQ Editor (hoff{atsign}hp{period}com) takes URL link continuity seriously, and strives to maintain current links. This is, however, a large task, and it is only becoming more difficult. URL link continuity is also something that you can assist with, of course. If you should find a dead link or a stale URL pointer, please notify the FAQ Editor (hoff{atsign}hp{period}com) . If you have or can find the current, correct or updated URL, of course, that greatly assists the FAQ Editor (hoff{atsign}hp{period}com) in maintaining the most current set of links.

Thank you!

Post to Newsgroups With Care!

> **Please consider reading Section 1.3 before posting to the comp.os.vms or any other newsgroups. (What you post can come back to haunt you.)**

Contributors

Many people have contributed to this FAQ, directly and/or indirectly. In many cases, part or all of an answer included in the FAQ has been adapted from one or more postings from the comp.os.vms or other newsgroup—this though individual citations and specific quotations are generally not included in this FAQ. (The FAQ Editor (hoff{atsign}hp{period}com) wishes to thank to all of those folks who post answers to the newsgroups, and that have contributed to the contents of and to the general improvements of this FAQ.)

The following is an incomplete list of the folks that have contributed directly or indirectly to the contents of the OpenVMS FAQ. Additions, corrections, and updates to this list of contributors and to the contents of the FAQ itself are welcome, and all omissions and misspellings are unintentional. Thanks to each of the contributors here, and to all of the folks that have participated in the FAQ and in the newsgroups.

Jim Agnew, Paul Anderson, Jason Armistead, Zoltan Arpadffy, John AtoZ, Bailey, Gary Barclay, (The) Beav, Jeremy Begg, Colin Blake, Fabrice Blanquart, Ken Blaylock, Kenneth Block, Bol, Craig Berry, Mark Berryman, Jim Becker, Verell Boaen, Bol, Jim Brankin, Richard Brodie, Colin Butcher, Robert Alan Byer, Jeff Campbell, Scott Campbell, Antonio Carlini, Keith Cayemberg, Ken Chaney, Cristy, John Croll, David J. Dachtera, Robert Deininger, Dale Dellutri, Leo Demers, Sue Denham, Thomas Dickey, Eric Dittman, Mike Duffy, Jim Dunham, Eric, Glenn Everhart, Ken Fairfield, Yaacov Fenster, Gunther Froehlin, Harry Garonzik, Derek Garson, John Gillings, Andy Goldstein, Clair Grant, Hunter Goatley, Ruth Goldenberg, Vance Haemmerle, Ki Suk Hahn, SHarris, Rich Hearn, Fletcher Hearns, Phillip Helbig, Hein van den Heuvel, Stephen Hoffman, Eberhard Heuser-Hofmann, Sue Holt, Mark Hopkins, Horn, Jaf, Leif Jansson, Kevin Jenkins, Don R. Jones, Henry Juengst, inazu_k, Geoff Kingsmill, Fred Kleinsorge, Felix Kreisel, Veli Korkko, Ruslan R. Laishev, Peter ‘‘EPLAN’’ Langstoeger, Norm Lastovica, p_lee, Jerry Leichter, Richard Levitte, Tom Linden, Steve Lionel, Denis Maillard, Kerry Main, John E. Malmberg, David Mathog, Dale Miller, Ian Miller, Dick Munroe, Patrick Moreau, Morrison, otis, George Pagliarulo, Keith Parris, Plass, pvhp, David Rabahy, Stan Rabinowitz, Mike Raspuzzi, Steve Reece, Neil Rieck, Ian Ring, Aaron Sakovich, Warren Sander, Mark Schafer, Brian Schenkenberger, Atlant Schmidt, Wayne Sewell, Steven Shamlian, Sue Skonetski, Scott Snadow, Sue Sommer, Scott Squires, stfp, Dave Sweeney, Mike Thompson, Arne Vajhøj, Martin Vorländer, Peter Weaver, William Webb, Williams, Paul Winalski, Uwe Zinser.

Again, any omission from this list and any errors or misspellings are entirely unintentional. Please notify the FAQ Editor (hoff{atsign}hp{period}com) of any omissions, corrections, or updates that might be needed within this list. And thanks to each of these individuals for their assistance and for their contributions, and particularly for sharing their expertise.

FAQ Archives and FAQ Locations

The OpenVMS FAQ is archived in (at least) the following locations:

- http://www.hp.com/go/openvms/faq/
- ftp://rtfm.mit.edu/pub/usenet/comp.os.vms/
- news:comp.answers
- news:news.answers
- http://www.faqs.org/
- http://www.google.com/
- http://eisner.decus.org/vms/faq.htm
- http://www.hoffmanlabs.org

Other internet FAQs are generally available in these locations:

- news://comp.answers
- news://news.answers
- ftp://rtfm.mit.edu/pub/usenet/
- http://www.faqs.org/
- http://www.google.com/

FAQ File Formats and Production Information

This FAQ is available in ASCII text, Postscript, Bookreader, and HTML, and potentially in other file formats. Certain formats, such as the Adobe Acrobat PDF conversion from the Postscript format, are provided downstream from the FAQ text formatting, and are provided and maintained by others.

As the HTML-format FAQ is not a single monolithic (and accordingly large) HTML file, it is the most difficult format to search using the most common standard (platform-local) tools. Accordingly, the FAQ Editor (hoff{atsign}hp{period}com) recommends downloading, using and searching the ASCII text FAQ format.

The OpenVMS FAQ is maintained in DECdocument SDML format and all FAQ maintenance and the primary output file format conversions are performed on OpenVMS systems. For information on the DECdocument product, please contact Touch Technologies.

Old FAQ Section Notations Available in Index

For the folks that are familiar with older editions of this FAQ and specifically for those folks looking for the older section name keywords, please see the index entries located under *FAQ Sections (Old)*. These older section names include the prefixes DCL, DECW, DOC, FILES, MISC, MGMT, INTRO, SOFT, VMS, etc., followed by the section number.

The Fine Print

Corrections and suggestions are welcome. Should you believe you can provide better stewardship of this FAQ, or should you have strong feelings over the content, structure, organization or implementation of the FAQ, do realize you may be offered the editorship. The editor maintains this document as a free service to the OpenVMS community.

While the information contained here is believed correct, the use of any and all information acquired from this document is entirely at the risk of the user(s). If your system should crash or your data should become corrupted or the answer(s) found here should be found erroneous, that is solely your risk and your responsibility.

Though the editor of this FAQ is an employee of Hewlett-Packard Company (HP), this posting and the information contained within this FAQ are is not an official statement of HP, nor a commitment by HP. All information, comments or opinions included, cited or otherwise referenced here may or may not reflect the stated opinions of HP, of the editor, or of any other entity.

All trademarks, registered trademarks, servicemarks and registered servicemarks used or referenced here are the properties of their respective owner(s) or holders. Please route any corrections, requests or related updates to the FAQ Editor (hoff{atsign}hp{period}com)

OpenVMS Frequently Asked Questions (FAQ)

The master copy of this FAQ is posted at

- http://www.hp.com/go/openvms/faq

Translations of this FAQ into various languages may be available, please contact the maintainer(s) of any translated document for assistance or feedback involving the particular translated edition of this FAQ.

1 Introduction

If you are searching for something here,
please consider using the text-format FAQ.

1.1 What core OpenVMS URLs do you need to know?

If you have no other OpenVMS URLs bookmarked, you will want to be familiar with the URLs listed in Table 1–1:

Table 1–1 Core Websites

URL	Contents
Core OpenVMS Information, including roadmaps and release schedules	
	http://www.hp.com/go/openvms/
OpenVMS and Core Layered Product Documentation	
	http://www.hp.com/go/openvms/doc/
Core OpenVMS Support Search Engine URLs, the ITRC support and discussion forums, and the ITRC FTP-accessable Patch Area	
	http://www2.itrc.hp.com/service/james/CPQhome.do (formerly the NLSA Natural Language Search Assistant, and formerly AskQ.)
	http://www.itrc.hp.com/
	ftp://ftp.itrc.hp.com/openvms_patches/
The OpenVMS Freeware	
	http://www.hp.com/go/openvms/freeware/
The OpenVMS Hobbyist License Program	
	http://www.openvmshobbyist.org/
	http://h71000.www7.hp.com/openvms/license_software_programs/index.html

1.2 What are the OpenVMS Usenet Newsgroups?

There are a huge number of newsgroups available, some of which discuss topics of interest to folks using OpenVMS operating system. Key among these are the comp.os.vms newsgroup and the vmsnet.* heirarchy of newsgroups.

1.2.1 What is the scope of the comp.os.vms newsgroup?

The comp.os.vms newsgroup is the primary newsgroup for discussion of the HP OpenVMS operating system and the computer systems on which it runs. Questions about layered products which run on OpenVMS are also welcome, though many of them (in particular, language compilers and database systems) have more specific newsgroups. If a question has some relationship to OpenVMS, it belongs in comp.os.vms/.

1.2.2 What newsgroups carry VMS-related information?

The vmsnet.* hierarchy is operated by the Encompass users group (formerly known as DECUS), and contains various newsgroups of interest, including vmsnet.misc and vmsnet.alpha.

Sprechen Sie Deutsch? Besuchen Sie bitte den newsgroup de.comp.os.vms.

The comp.sys.dec newsgroup carries general discussions about various HP computer systems, and specifically systems that were acquired by HP as part of the acquisition of Compaq, systems which had previously been acquired by Compaq as part of the acquisition of Digital Equipment Corporation (DIGITAL).

Also of interest are resources and conferencing systems including Encompasserve (formerly known as DECUServe) and Ken Farmer’s OpenVMS.Org forums:

- telnet://eisner.decus.org/
- http://www.openvms.org/

1.2.3 What newsgroup archives are available?

Extensive archives of INFO-VAX mailings and comp.os.vms postings are archived and available at SRI International: http://mvb.saic.com/freeware/info-vax/

Google also has extensive newsgroup archives available at

- http://www.google.com/

1.2.4 What is the INFO-VAX mailing list?

INFO-VAX is a mailing list which is bidirectionally gatewayed to the comp.os.vms newsgroup. This means that postings to comp.os.vms get automatically sent to INFO-VAX subscribers and messages sent to the INFO-VAX list are automatically posted to comp.os.vms

INFO-VAX can be a useful way to participate in the newsgroup if you can't access the newsgroup directly through a newsreader. Note that INFO-VAX is a closed mailing list, which means that only subscribers may post to it. Please see Section 1.2.4.1 for information on how to subscribe, and how you can subscribe such that you can post from multiple addresses but still receive no more than one copy of the mailings.

Please remember that propogation delays can and do vary, meaning that mailings and postings may be delayed or even lost. It is quite possible that postings may not be delivered for several days, and that some postings will appear out of the expected order.

For details on the available archives, please see Section 1.2.3.

1.2.4.1 How do I subscribe to or unsubscribe from INFO-VAX?

The address for subscription requests is Info-VAX-Request[at]Mvb.Saic.Com. Subscription requests are handled automatically by a mail server. This mail server ignores the subject line and processes each line of the message as a command.

The syntax for subscribing and unsubscribing, as well as setting various parameters on your subscription, is:

Table 1–2 INFO-VAX Mail Server Commands

Command	Description
HELP	Acquire information on mail server
SUBSCRIBE INFO-VAX	Join to the mailing list
UNSUBSCRIBE INFO-VAX	Leave the mailing list
SET INFO-VAX DIGEST	Select periodic omnibus mail messages from the mail server, rather than multiple individual messages for each posting.
SET INFO-VAX NODIGEST	Separate mail delivery of each message posted
SET INFO-VAX NOMAIL	Establish an email address for posting, though not for receiving, mail.
SET INFO-VAX MAIL	Establish an email address for both posting and for receiving mail from the mailing list.

The mail server is not case-sensitive; case is irrelevant. Settings and policies can be retrieved from the mail server, typically using HELP.

The INFO-VAX moderator can be contacted directly at the email address Mark.Berryman[at]Mvb.Saic.Com.

1.3 What is [n]etiquette?

Before posting or emailing a question, please use the available local resources, such as the OpenVMS manuals, the HELP, and the resources and information in and referenced by this FAQ. Please use these first. Also please specifically read the release notes and (if appropriate) the cover letter for the product you are using. (The release notes are generally placed in SYS$HELP:.) Quite often, these simple steps will allow you to quickly find the answer to your own question—and more quickly than waiting for a response to question posted to a newsgroup, too. These steps will save you time, and will also help ensure you have a good reputation with the folks that might be included to answer one of your future questions, a question not covered in these resources. Put another way, if you do not want your questions to be ignored in the future—and please remember that the folks in the newsgroups do *not* have to answer your questions—you won't want to ‘‘annoy the natives’’ by asking a question that has already been answered far more times more than you might have realized, or a question whose answer is readily available had you made a small effort.

When posting, please consider the following suggestions:

- There is *no* particularly reliable way to recall, erase, delete, or otherwise hide a message once it is emailed or once posted. Once your message has reached an external email server or multiple news servers, the entire text is effectively a permanent fixture of the network. And using the available search engines, a fixture that is easy to locate and to correlate. (Do not assume that all tools or archives will honor the do-not-cache attributes, either—postings marked as such can be among the most interesting ones to cache, after all.)

  For details on some of the many available archives, please see Section 1.2.3.
- Include a valid e-mail address in the text of your posting or in a "signature" appended to the end. Reply-to addresses in headers often get garbled. Anonymous addresses can also simply be ignored, as fake addresses are regularly used by folks that are ‘‘trolling’’ and by folks that are spamming. (Though to avoid spam-harvesting of your email address, consider adding characters or a field into the address—but remember to include details around which characters or fields should be removed or altered if you decide to be particularly clever here.)
- If you are submitting a question, please be as specific as you can. Include relevant information such as processor type, product versions (OpenVMS and layered products that apply), error message(s), DCL command(s) used, and a short, reproducible example of problems. Say what you’ve tried so far, so that effort isn’t duplicated. Keep in mind that there’s not yet a telepathy protocol for the Internet. (The more detailed your description, the better that people can help you with your question.)
- If responding to a posting, include in your reply only as much of the original posting as is necessary to establish context. As a guideline, consider that if you’ve included more text than you’ve added, you’ve possibly included too much. Never include signatures and other irrelevant material.
- Please be polite. If the question isn’t worded the way you think is correct or doesn’t include the information you want, try to imagine what the problem might be if viewed from the poster’s perspective. Requests for additional detailed information are often better sent through mail rather than posted to the newsgroup.

- If you have a problem with HP (or any other vendor's) product, please use the appropriate support channel. Do not assume that newsgroup postings will get read, will be responded to by the appropriate developers, or will be later followed up upon.
- If you are posting from a web browser, news reader or if you are posting via email sent to INFO-VAX, please turn off MIME, vcard, attachments, and other mechanisms that assume anyone reading the post has the corresponding capability—use the text-only option of your web browser, news reader, or mailer. Usenet is traditionally a text-only medium, and many comp.os.vms participants will use tools that have this support disabled, or that do not have this support. If the message uses MIME or attachments or such, the text of your message will be buried in a large pile of gibberish, and some tools will send multiple copies of the text within a single posting.
- If you find that the postings of a particular user are uninteresting, annoying, or off-topic, most newsreaders include a filter or killfile mechanism, and many mail clients have similar filtering capabilities. *Please do not ''flame''—to email or to post vitriol – any individual that might annoy you, please enable and filter all of that user's postings.* Posting of vitriol and of ''flames'' *will* eventually come back to haunt you; netizens and the net itself have a very large and a very long memory. Similarly, readers that decide that your postings are not worthy of reading will similarly tend to filter or to killfile *all* of your postings. Please play nice, in other words.

Before posting your question to the comp.os.vms newsgroup or sending your message to the INFO-VAX list, also please take the time to review available etiquette information, such as that included in the following documents:

wideftp://rtfm.mit.edu/pub/usenet/
news.answers/usenet/primer/part1

wideftp://rtfm.mit.edu/pub/usenet/
news.answers/usenet/faq/part1

wideftp://rtfm.mit.edu/pub/usenet/
news.answers/usenet/emily-postnews/part1

wideftp://rtfm.mit.edu/pub/usenet/
news.answers/usenet/writing-style/part1

wideftp://rtfm.mit.edu/pub/usenet/
news.answers/usenet/posting-rules/part1

This information will document the etiquette of newsgroups, as well as providing you with the knowledge the vast amount of newsgroup-related information that is readily available to you, and where to find it...

Note

Please do not post security holes or system crashers

Rather, please report these problems directly to HP. Why? So that HP has a chance to resolve and distribute a fix before other customer sites can be affected.

Most folks in the newsgroups are honest and deserve to know about potential security problems, but a few folks can and will make nefarious use of this same information. Other sites will hopefully return the courtesy, and will not post information that will potentially compromise your site and your computer environment.

1.4 What OpenVMS user group(s) are available?

Encompass, the Enterprise Computing Association, is a user group comprised of information technology professionals that are interested in the Enterprise-oriented products, services, and technologies of Compaq and of the former DIGITAL. Encompass offers newsletters, the Encompass website, and offers various gatherings and related services, including symposia events and local users group meetings.

Encompass is a descendent of the organization known as DECUS, the Digital Equipment Computer Users Society.

For more information on Encompass, please visit the Encompass web site:

- http://www.encompassus.org/

The organization comprised of customers of Hewlett-Packard Company (HP) that is probably most analogous to the Encompass organization is Interex:

- http://www.interex.org/

Like Encompass, Interex offers various services and events of interest to folks that presently work with and/or that wish to learn about HP products and offerings. Please see the Interex website for details.

1.5 OpenVMS Support, Questions and Comments?

The following section includes contacts for OpenVMS Feedback, and information on how to obtain technical support information.

1.5.1 Corporate contacts for OpenVMS Business Issues?

The HP corporate contact for OpenVMS business issues is Ann McQuaid, the HP General Manager directly in charge of OpenVMS and OpenVMS Engineering, while feature requests and other related matters should be routed to MaryJane Vazquez, the OpenVMS Business Manager.

Ann and MaryJane will quite obviously respond best to cogently-worded OpenVMS corporate-level business issues or requests. With all due respect to all involved, neither Ann nor MaryJane are appropriate contacts for technical support matters nor for technical support requests, nor for any other non-corporate-related, non-business-related issues—these questions are best routed to the local or regional customer support center; to the support, technical and engineering teams.

To reach Ann or MaryJane via electronic mail, place a dot between the first and the surname, and append the expected HP.COM domain.

1.5.2 OpenVMS Ambassadors?

The OpenVMS Ambassadors are senior HP engineers with advanced technical knowledge and advanced training in OpenVMS, with detailed knowledge of current and future OpenVMS releases and product plans, and with contacts directly with the HP and ISV hardware and software engineering organizations developing OpenVMS and OpenVMS hardware platforms, as well as layered products and tools. Further, Ambassadors are experienced with integrating HP OpenVMS and application-specific products and ISV applications to solve specific business requirements.

OpenVMS Ambassadors are based throughout the world.

Your HP sales representative or HP reseller will be able connect you with your local OpenVMS Ambassador.

1.5.3 Contact for OpenVMS Marketing Issues and Questions?

Please see Section 3.4.

1.5.4 Contact URLs for OpenVMS Technical Issues?

For formal technical issues and technical support, please contact your software support organization, or your local HP Customer Support Center or HP Reseller. In North America, you can call 1-800-HP-INVENT.

For informal (free) support resources, see the newsgroups including comp.os.vms (news://comp.os.vms/ comp.os.vms), see the ITRC discussion forums, the James support database search engine (search assistant tool) (http://www2.itrc.hp.com/service/james/CPQhome.do), and see sections of this document including the platform support information in Section 14.4, sources of software and hardware support in Section 14.15, information on third-party devices and unsupported hardware in Section 14.25, and the many other sections of this document.

Please remember to review and to bookmark the following support URLs:

- http://www.hp.com/go/openvms/
- http://www2.itrc.hp.com/service/james/CPQhome.do
- http://www.hp.com/go/openvms/wizard/ (reference only; use ITRC forums for new questions)
- ftp://ftp.itrc.hp.com/openvms_patches/

2 General Information

**If you are searching for something here,
please consider using the text-format FAQ.**

2.1 What is OpenVMS? What is its history?

OpenVMS, originally called VMS (Virtual Memory System), was first conceived in 1976 as a new operating system for the then-new, 32-bit, virtual memory line of computers, eventually named VAX (Virtual Address eXtension).

The first VAX model, the 11/780, was code-named ‘‘Star’’, hence the code name for the VMS operating system, ‘‘Starlet’’, a name that remains to this day the name for the system library files (STARLET.OLB, etc.).

VMS version X0.5 was the first released to customers, in support of the hardware beta test of the VAX-11/780, in 1977. VAX/VMS Version V1.0 shipped in 1978, along with the first revenue-ship 11/780s.

OpenVMS was designed entirely within HP and specifically within the former Digital Equipment Corporation (DIGITAL). Two of the principal designers were Dave Cutler and Dick Hustvedt, though with a wide variety of other contributors. OpenVMS was conceived as a 32-bit, virtual memory successor to the RSX-11M operating system for the PDP-11. Many of the original designers and programmers of OpenVMS had worked previously on RSX-11M, and many concepts from RSX-11M were carried over to OpenVMS.

OpenVMS VAX is a 32-bit, multitasking, multiprocessing virtual memory operating system. Current implementations run on VAX systems from HP and other vendors, as well as on hardware emulators; for additional information on emulators, please see Section 13.12 and

OpenVMS Alpha is a 64-bit multitasking, multiprocessing virtual memory operating system. Current implementations run on Alpha systems from HP, and other vendors.

OpenVMS has also been ported to the Intel IA-64 architecture, and specifically to HP Integrity systems using microprocessors from the Intel Itanium Processor Family. This implementation of OpenVMS is officially known as ‘‘HP OpenVMS for Integrity Servers’’ and more commonly as ‘‘OpenVMS I64’’, and it operates in the native Itanium IA-64 architecture and 64-bit environment. OpenVMS I64 provides support for applications requiring 32- or 64-bit virtual addressing capabilities entirely within the native 64-bit Itanium execution environment. (For details on this and related terminology, please see Section 14.4.5.)

For more details on OpenVMS and its features, please read the OpenVMS Software Product Description at:

- http://h18000.www1.hp.com/info/spd/

  OpenVMS typically uses SPD 25.01.xx, SPD 41.87.xx, and SPD 82.35.xx.

Additional information on the general features of various OpenVMS releases, release dates, as well as the development project code names of specific releases, is available at:

- http://www.openvms.compaq.com/openvms/os/openvms-release-history.html

Additional historical information—as well as pictures and a variety of other trivia—is available in the VAX 20th anniversary book:

- http://www.openvms.compaq.com/openvms/20th/vmsbook.pdf

For information on the FreeVMS project, and on hobbyist and educational versions of OpenVMS, please see:

- http://www.systella.fr/~bertrand/FreeVMS/indexGB.html
- http://en.wikipedia.org/wiki/FreeVMS
- http://www.openvmshobbyist.org/
- http://www.openvmsedu.com/

Also please see the related software licensing topics Section 2.8.4, Section 2.8.1, and Section 2.15, and (for developers working on commercial applications for OpenVMS) Section 2.8.3.

2.2 What is the difference between VMS and OpenVMS?

VMS and OpenVMS are two names for the same operating system. Originally, the operating system was called VAX-11/VMS; it changed to VAX/VMS at around VAX/VMS V2.0. When the VMS operating system was ported to the Alpha platform, it was renamed OpenVMS, for both VAX and Alpha (and for the Itanium Processor Family), in part to signify the high degree of support for industry standards such as POSIX, which provides many features of UNIX systems.

For those versions with POSIX, an OpenVMS license allows you to install and run POSIX for OpenVMS at no additional charge; all you need is the media and documentation which can be found on the Consolidated Distribution and On-Line Documentation CD-ROMs. Support for the POSIX package on more recent OpenVMS releases is not available, various parts of POSIX such as calls from the API are being integrated more directly into OpenVMS. For more information on POSIX for VMS see question SOFT2

What became confusing is that the OpenVMS name was introduced first for OpenVMS AXP V1.0 causing the widespread misimpression that OpenVMS was for Alpha AXP only, while ‘‘regular VMS’’ was for VAX. In fact, the official name of the VAX operating system was changed as of V5.5, though the name did not start to be actually used in the product until V6.0.

2.3 What’s in a Name? Terminology and Products?

The proper names for OpenVMS on the various platforms are ‘‘OpenVMS VAX’’, ‘‘OpenVMS Alpha’’, and ‘‘OpenVMS I64’’. Use of ‘‘OpenVMS AXP’’ and of ‘‘VAX/VMS’’ are deprecated.

The VAX and Alpha terms are largely interchangeably used as the names of platforms, of processor or microprocessor implementations, and of the respective computing architectures.

Somewhat confusing to long-time OpenVMS users, Intel IA-32, IA-64, and EM64T, and AMD AMD64 are the names of various computing architectures and of architectural extensions. Only. These are not the names of any implementations, nor of any platforms.

Intel Itanium is the name of a family of microprocessor implementations of the Intel IA-64 architecture, as Intel Pentium and Xeon are the names of families of microprocessor implementations of Intel IA-32 and (potentially) of the EM64T extensions.

I64 is the generic name for the various HP Integrity platforms supported by HP OpenVMS for Integrity Servers (and more commonly as ‘‘OpenVMS I64’’); for the platforms supported by OpenVMS I64. (For additional related terminology, please see Section 14.4.5.)

2.3.1 How do I port from VMS to OpenVMS?

You already did. Wasn't that easy? Please see Section 2.2 for details.

2.4 Which is better, OpenVMS or UNIX?

This question comes up periodically, usually asked by new subscribers and new posters who are long-time UNIX or Linux users. Sometimes, the question is ignored totally; other times, it leads to a long series of repetitive messages that convince no one and usually carry little if any new information. Please do everyone a favor and avoid re-starting this perpetual, fruitless debate.

That said, OpenVMS and the better implementations of UNIX are all fine operating systems, each with its strengths and weaknesses. If you're in a position where you need to choose, select the one that best fits your own requirements, considering, for example, whether or not the layered products or specific OS features you want are available, and considering the expected cost-of-ownership over the lifetime of the system installation.

If you are asking this question, you are probably comparing OpenVMS to UNIX. It was once certainly true that OpenVMS and UNIX were quite different. In more recent times, there are tools and C APIs on OpenVMS that directly provide or that easily support porting UNIX programs and commands, and there are equivalent packages bringing various OpenVMS features and mechanisms to UNIX platforms.

If you seek UNIX tools on OpenVMS rather than the more philosophical discussion found in this section, please see the GNV package and other GNU discussions in Section 13.2.6, and please see the plethora of C calls currently available in the HP C Run-Time Library documentation, briefly discussed over in Section 13.2.1.

2.5 Is HP continuing funding and support for OpenVMS?

Yes.

Active development of new OpenVMS releases is underway, as well as the continuation of support.

Please see the following URLs for details, roadmaps, and related information:

- http://www.hp.com/go/openvms/

2.6 What OpenVMS distribution kits are available?

Various distributions are available.

For the most current information on the available part numbers and current products (OpenVMS distribution kits, media, documentation, etc) and the most current associated licensing information, please see the current OpenVMS Software Product Description (SPD) document, available at:

- http://h18000.www1.hp.com/info/spd/

  OpenVMS typically uses SPD 25.01.xx, SPD 41.87.xx, and SPD 82.35.xx.

The CD-ROMs listed in Table 2–1 contain just the OpenVMS Alpha operating system. The operating system distribution kits are bootable, and can be used to run BACKUP from the optical media, as well as performing an installation or upgrade.

Table 2–1 OpenVMS Alpha Media Kits

Part	Description
QA-MT1AG-H8	OpenVMS Alpha V6.2-1H3 hardware release CD-ROM; also requires QA-MT1AA-H8.6.2
QA-MT1AR-H8	OpenVMS Alpha V7.1-2 maintenance release CD-ROM
QA-MT1AT-H8	OpenVMS Alpha V7.2-1 maintenance release CD-ROM
QA-MT1AU-H8	OpenVMS Alpha V7.2-2 maintenance release CD-ROM
QA-MT3AA-H8	OpenVMS Alpha and VAX products and documentation on CD-ROM
QA-MT3AE-H8	OpenVMS Alpha and VAX documentation on CD-ROM

OpenVMS I64 is distributed on DVD-ROM media, and is bootable. OpenVMS I64 licensing is implemented on a per-processor-socket basis, with the classic license tiers based on the numbers of processor sockets that can be present. Further, three general product and licensing groupings are optionally available with OpenVMS I64, the Foundation Operating Environment (FOE), the Enterprise Operating Environment (EOE), and (as/when/if available) the Mission Critical Operating Environment (MCOE). Seperate per-product licenses are generally also available for various of the products within the Operating Environment groups.

Table 2–2 OpenVMS I64 Order Numbers

Part	Description
BA322AA#???	OpenVMS I64 FOE Product
BA323AA#???	OpenVMS I64 EOE Product
BA324AA#???	OpenVMS I64 MCOE Product

The product suffix required for the order numbers listed in Table 2–2 can be found in Table 2–3.

Table 2–3 OpenVMS I64 Media Suffix

Suffix	Description
A18	OpenVMS I64 FOE V8.2 DVD media
AJR	OE media kit on DVD media
0D1	Factory installation

The OpenVMS VAX, OpenVMS Alpha and OpenVMS I64 source listings sets referenced in Table 2–4 include the source listings of most of OpenVMS, and these machine-readable distributions are invaluable for any folks working directly with OpenVMS internals, as well as for folks interested in seeing examples of various OpenVMS programming interfaces.

Table 2–4 OpenVMS Source Listings Kits

Part	Description
QB-MT1AB-E8	OpenVMS Alpha Source Listings kit and license
QT-MT1AB-Q8	OpenVMS Alpha Source Listings Updates
BA422AA	OpenVMS I64 Source Listings kit and license
QB-001AB-E8	OpenVMS VAX Source Listings kit and license
QT-001AB-Q8	OpenVMS VAX Source Listings Updates
BA422AA	OpenVMS I64 source listings kit and license

Additional OpenVMS packages and technologies including NetBeans, XML, SOAP, UDDI, JDK, Perl, Tomcat, SSL and such are discussed within the OpenVMS e-Business Infrastructure Package SPD 80.58.xx.

Again, please see the OpenVMS SPD and the documents and parts referenced there for the most current information.

For information on non-commercial software distributions for use by and for OpenVMS Hobbyists, please see Section 2.8.1.

2.6.1 Where can I download OpenVMS and Layered Product Kits?

HP customers with commercial licenses and support contracts can download software product distribution kits from the following HP website:

- http://www1.sqp.com/

You can also find pointers to the Software Rollout Report and to the OpenVMS SPD listings via the above SQP website.

Information on obtaining and transfering licenses is available in Section 2.6 and Section 2.8.4, while information on the OpenVMS Hobbyist licensing program and on obtaining hobbyist product distribution kits is in Section 2.8.1.

2.7 In what language is OpenVMS written?

OpenVMS is written in a wide variety of languages.

In no particular order, OpenVMS components are implemented using Bliss, Macro, Ada, PLI, VAX and DEC C, Fortran, UIL, VAX and Alpha SDL, Pascal, MDL, DEC C++, DCL, Message, and Document. And this is certainly not a complete list. However, the rumor is NOT true that an attempt was made to write pieces of OpenVMS in every supported language so that the Run-Time Libraries could not be unbundled. (APL, BASIC, COBOL and RPG are just some of the languages NOT represented!)

There are a large variety of small and not-so-small tools and DCL command procedures that are used as part of the OpenVMS build, and a source code control system capable of maintaining over a hundred thousand source files across multiple parallel development projects, and overlapping releases.

2.8 Obtaining and Transfering OpenVMS licenses?

The following sections describe hobbyist and educational license programs, as well as information on commercial licenses and transfers.

For information on the available commercial OpenVMS licenses and for information on license transfers, please see Section 2.8.4. OpenVMS Hobbyist licenses are discussed in Section 2.8.1. For information on the licensing implementation, troubleshooting licensing problems, on the License Unit Requirements Table (LURT), and other related details, please see Section 5.39. For configuring and troubleshooting LMF, see Section 12.4.

2.8.1 Questions asked by Hobbyist OpenVMS licensees?

If you are a member of an HP-recognized user group (eg: Encompass, Enterex, DECUS), and are considering acquiring and using a VAX, Alpha or (soon) IA-64 system for hobbyist (non-commercial) use, (free) license product authorization keys (PAKs) for OpenVMS VAX, OpenVMS Alpha, and (reportedly) OpenVMS I64, and layered products are available.

In addition to the license keys, OpenVMS VAX and Alpha distribution CD-ROM distribution kits are available with OpenVMS, DECwindows Motif, DECnet and TCP/IP networking, compilers, and a variety of layered products. (A hobbyist distribution for OpenVMS I64 is expected.) (While the hobbyist CD-ROM distributions are intended for and tailored for OpenVMS Hobbyists, the contents and capabilities of the Hobbyist installation kits included within the OpenVMS Hobbyist distribution do not differ from the standard distribution installation kits. The products are chosen to reflect the most popular products and the space available on the media.)

If you have questions on what else is authorized by the license agreement and on what other distribution media is available to you, well, please read the applicable software license agreement(s).

For further information, please link to:

- http://www.openvmshobbyist.org/

On the OpenVMS Hobbyist license registration form at the above website (as of August 2005), you are offered the choice of the ‘‘OpenVMS VAX’’ license(s), the ‘‘OpenVMS Alpha’’ license(s), and the ‘‘Layered Products’’ licenses. You will want the operating system license for your particular

OpenVMS platform *and* you will want the ‘‘Layered Products’’ licenses. You will want to select and to acquire *two* sets of license PAKs.

For vendors wishing to license products specifically for hobbyist use (and to not issue hobbyist PAKs), the program provides hobbyists with the license PAK OPENVMS-HOBBYIST.

If you plan to use a hardware emulator (eg: VAX emulator) on a Microsoft Windows platform, make sure you have an OpenVMS distribution kit that can be installed and/or booted with the particular emulator package you plan to use. For additional information on emulators, please see Section 13.12 and particularly please see the emulator-related documentation.

2.8.1.1 Vendors offering Hobbyist Licenses

- GrayMatter Software
  http://www.graysoft.com/GSCHobbyPR.html
  http://www.graysoft.com/GSCHobbyPR.html
- Argent Software
  http://jams.argent-software.com/hobby.html
- Kednos
  http://www.kednos.com/
- LJK
  http://ljk.com
- Process Software
  http://www.process.com/openvms/hobbyist.html
- Raxco
  http://www.raxco.com/hobbyist/
- Software Resources International (SRI)
  http://www.softresint.com/charon-vax/Tools_and_tips.htm

Hobbyist license product additions, and any updates for products already listed here are welcome. Please contact the FAQ Editor (hoff{atsign}hp{period}com)

2.8.2 OpenVMS Educational and CSLG licenses?

For information on OpenVMS licenses for educational customers, please see the HP Campus Software License Grant (CSLG) license program and the OpenVMS Educational license program:

- http://www.openvmsedu.com/

2.8.3 What developer and partner licensing programs are available?

Commercial software developers can join the HP DSPP program, and can (potentially) receive discounts on various software product licenses and software distributions, as well as on hardware purchases.

- http://www.hp.com/go/dspp/

The DSPP program is the descendent of the DIGITAL ISVN and DIGITAL ASAP programs and the Compaq CSA program, and the analogous developer and partner programs at HP.

One of the benefits available is a (free) license product authorization key (PAK) that enables the generation of license PAKs—the so-called PAKGEN PAK generator license PAK. An example of this is available on the OpenVMS Freeware V8.0 distribution. (And no, you can't use the Freeware PAKGEN PAK to generate any license PAK you might want.)

Please see Section 2.8.3 for additional details on the DSPP program.

For information on the OpenbVMS Hobbyist and OpenVMS Educational license programs, please see Section 2.8.1.

2.8.4 How do I obtain or transfer an OpenVMS license?

To transfer a commercial OpenVMS license from one owner to another, or to purchase a commercial license, you can contact HP at regional sales office or reseller.

For information on the hobbyist license program, please see Section 2.8.1.

2.9 Does OpenVMS support the Euro currency symbol?

OpenVMS can generate the *%xA4* character code used for the Euro, and the DECwindows DECterm can display the glyph. Please check with the vendor of your terminal or terminal emulator for additional details.

2.10 OpenVMS Ports? Itanium? Ports to IA-32, EM64T or AMD64 systems?

OpenVMS has been ported to and is operational on four architectures: VAX, Alpha, IA-64, and IA-32. The first three have available native ports of OpenVMS, the fourth is available via emulation. VAX is the oldest architecture, and limited to 32-bit virtual and up to 34-bit physical addressing. The Alpha and IA-64 architectures are both 64-bit architectures, with 64-bit virtual addressing available. The available IA-32 emulation is provided for the OpenVMS VAX and other VAX operating systems, and provides a 32-bit VAX environment. For additional information on the emulation, please see Section 13.12.

As for (the lack of) a native port for IA-32, OpenVMS Engineering presently and continues to believe that there would be insufficient market (read: profit, customer interest) to justify the cost involved in a native port of OpenVMS to systems using the Intel IA-32 architecture. In addition to the direct costs involved in any port and in addition to the substantial effort involved in moving backwards from a 64-bit environment on Alpha and on IA-64 to a 32-bit platform (such as IA-32), and the exceedingly non-trivial device qualification costs and the costs in moving backwards into older PCI and I/O environments (IA-32 systems more than a few years old have equivalently aged I/O support and buses), each organization and each person maintaining a product or a package for OpenVMS will have to justify a port to ‘‘OpenVMS IA-32’’, ‘‘OpenVMS EM64T’’ or ‘‘OpenVMS AMD64’’, akin to the decisions and the effort involved in porting a product from OpenVMS VAX to OpenVMS Alpha, or the port to OpenVMS I64.

Platform ports of many of the various products can be easy, and many of the ports of applications using documented OpenVMS features are expected to require little more than a source rebuild. Other products can and do depend on platform-specific or undocumented features, and the associated ports can be more involved. Regardless, ports of operating systems are very large and involved projects. The prerequisite product requirements for an OpenVMS operating system port are also non-trivial, as well—compilers in particular are obviously required, and the suite of compilers provided must maintain a very high degree of source-level compatibility across the platforms. In the case of the HP Integrity port, OpenVMS I64 V8.0 used cross-compilers and cross-tools operating on OpenVMS Alpha systems, while V8.2 and later have various native compilers available.

The OpenVMS I64 port was centrally built using the existing OpenVMS Alpha environment and around the work and the knowledge from the OpenVMS Alpha port, and OpenVMS Engineering fully expects that customers and ISVs will use and will continue to use OpenVMS Alpha systems to assist with their own ports to OpenVMS I64. OpenVMS Engineering fully expects to see customers using mixed-architecture clusters and fully shared file systems, as well.

OpenVMS Engineering is well aware of the AMD AMD64 (64-bit) platform and processors. (At least one of the available VAX emulators can reportedly utilize parts of the AMD64 instruction set, please contact the VAX emulator vendor(s) or maintainer(s) for assistance and details on their products.) OpenVMS Engineering is also well aware of the Intel EM64T platform and processors. There are *no* plans to provide a native port of HP OpenVMS for any systems based on the AMD AMD64 nor Intel EM64T architectures.

As part of the work leading to the Itanium port, senior engineers had extensively evaluated the products and the architectures available across the high-end 64-bit computing space, and chose to target Itanium for 64-bit environments—this while under the Compaq organization. This included looking at IA-32. HP (a co-developer of Itanium with Intel) had seperately chosen to target Intel Itanium for its high-end computer products. Compaq then announced plans for the future of Alpha through EV7-series products and platforms, and HP (entirely seperately) announced plans for PA-RISC products and platforms. The Itanium target has been maintained consistently since the Itanium port was announced by Compaq, and has also been consistently maintained by HP and by the combined company. For those folks prefering to follow the schedules and the product deliveries, OpenVMS Engineering had OpenVMS I64 V8.0 ready (internally) ahead of schedule—and with more features available within the release than had been originally planned for the release. (For information on and for schedules of future OpenVMS releases, please see the roadmap that is available at the OpenVMS website.)

OpenVMS I64 itself does not require and does not plan to utilize the Itanium IA-32 32-bit environment for the operation of OpenVMS itself. OpenVMS I64 V8.0 and later run natively on the Itanium processor family, with no use of IA-32 instructions. While OpenVMS can and does support 32-bit OpenVMS applications and addressing on Itanium, this is done with sign-extension addressing techniques entirely analogous to what was done with 32-bit applications operating in the 64-bit Alpha environment. Both OpenVMS 32-bit and 64-bit applications operate within the native Itanium instruction set and run-time environment, and do not use the Itanium IA-32 environment.

But yes, a native IA-32 port or a native AMD AMD64 or Intel EM64T port of OpenVMS would certainly be nice to have—this, of course, following the traditional Linux preference for having a Linux port available for most (all?) computer architectures known, and even for certain high-end refrigerators and toasters, and similar appliance-like devices. (The downside of this all-encompassing approach: this requires near-infinite engineering and support costs from the various vendors involved, and the qualification efforts and costs of most everything—everywhere. Or reduced or eliminated testing and support efforts. Or an unfortunate combination of these two. These costs are huge, and the benefits derived from the work are comparatively small when given the comparable costs of more targeted (and thus supported and supportable) hardware configurations—the platform targets are and must be carefully selected and considered by each vendor. Put another way, there are *no* plans to provide a native port of HP OpenVMS for systems based on Intel IA-32 processors, nor for systems based on AMD AMD64 nor Intel EM64T architectures and processors.

All this material having been written, have you looked at the system configurations and pricing of the available HP Integrity Intel Itanium systems? Low-end computer hardware is clearly a commodity product, and the systems are priced, serviced, upgraded, and replaced accordingly. Intel Itanium is a commodity microprocessor presently used in platforms available from various hardware vendors, including (obviously) from HP. Further, Itanium is a microprocessor available from and supported by Intel, a semiconductor vendor known for exceedingly high-volume microprocessor fabrication process and production capabilities.

For information on supported platforms and processors, please see the OpenVMS Software Product Description (SPD) at:

- http://h18000.www1.hp.com/info/spd/

  OpenVMS typically uses SPD 25.01.xx, SPD 41.87.xx, and SPD 82.35.xx.

Please see Section 14.4.5 for Intel Itanium terminology.

2.11 Are there any network-accessible OpenVMS systems?

Yes, though various restrictions can and do apply.

- Hobbes
  Hobbes is a MicroVAX 3100 Model 40 for which free access and accounts are available to OpenVMS enthusiasts. This system has BASIC, Pascal, Fortran, and C compilers installed. If you would like an account on Hobbes, please see the FAQ at
  - http://www.hobbesthevax.com/
- OpenVMS Galaxy Test-Drive
  HP currently offers an OpenVMS Galaxy Test-Drive system, based on an AlphaServer 4100 series configured as two instances of the OpenVMS operating system. For details, please visit:
  - http://www.testdrive.hp.com/
- HP DSPP Test-Drive
  The HP DSPP program offers various test-drive systems, including an HP Integrity Itanium development system and an HP OpenVMS I64 installation on an HP Integrity rx2600 server. For details on the DSPP program and on the test-drive systems, please see section Section 2.8.3 and please visit:
  - http://www.testdrive.hp.com/
  - http://www.hp.com/dspp/

  The test-drive systems do require registration, though access to the systems is free.
- Encompasserve
  Encompasserve offers free access an OpenVMS Alpha system.
  - telnet://eisner.decus.org/
- OpenECS
  OpenECS offers free access to a VAX 6000 model 530 system. If interested, please visit:
  - http://vax6k.openecs.org/
- The Deathrow Cluster
  The maintainers of the Deathrow Cluster offer access to an OpenVMS VAX and an OpenVMS Alpha system, configured in a cluster.
  - telnet://deathrow.vistech.net

- The Preatorian Public OpenVMS Cluster
  The maintainers of the Deathrow Cluster offer access to an OpenVMS Alpha cluster. Details are at the website listed below:
  - http://www.preatorian.net

2.12 What version of OpenVMS do I need?

For information on supported platforms, please see the OpenVMS Software Product Description (SPD) for the particular OpenVMS version of interest.

- http://h18000.www1.hp.com/info/spd/
  OpenVMS typically uses SPD 25.01.xx, SPD 41.87.xx, and SPD 82.35.xx.

For a table of the minimum and (as applicable) maximum OpenVMS versions required for various platforms, please see the hardware support chart at HP OpenVMS website and (as available) the following (potentially volatile; intra-website) link:

- http://h71000.www7.hp.com/openvms/hw_supportchart.html

For information on the Multia, related Alpha single-board computers, or other officially unsupported systems, please see Section 14.4.1 and Section 14.4.2.1.

The following is a rule-of-thumb for Alpha platform support. The table Table 2–5 contains the earliest OpenVMS Alpha release with support for a particular series of Alpha microprocessors:

Table 2–5 OpenVMS Alpha Version Rule-Of-Thumb

Generic	Microprocessor Generation	OpenVMS Version	General Comments
EV4	21064	V1.0	few systems; most EV4 require later; upgrade available
EV5	21164	V6.2	subsequent upgrade available
EV56	21164A	V6.2-1H3	subsequent upgrade to V7.1 and later

Table 2–5 (Cont.) OpenVMS Alpha Version Rule-Of-Thumb

Generic	Microprocessor Generation	OpenVMS Version	General Comments
EV6	21264	V7.1-2	subsequent upgrade typically to V7.2-1 or later
EV67	21264A	V7.1-2	subsequent upgrade typically to V7.2-1 or later
EV68	21264B, C and D	V7.2-1	believed/probable; currently an unconfirmed expectation

Specific hardware present and various system configurations can require OpenVMS Alpha releases later than those referenced in Table 2–5.

2.13 How can I submit OpenVMS Freeware?

For the guidelines and submission info, please visit the URL:

- http://www.hp.com/go/openvms/freeware/

To order the current OpenVMS Freeware CD-ROM kit (shipping and handling charges apply), please request part number QA-6KZAA-H8.

2.14 Porting applications to OpenVMS?

Porting can range from simple to rather complex, and depends on the features used on the original platform.

This section covers generic porting, and porting among OpenVMS VAX OpenVMS Alpha, and OpenVMS I64. (Porting among OpenVMS VAX, OpenVMS Alpha and OpenVMS I64 is often quite simple and involves little more than rebuilding from source, though a few applications using features specific to the platform or the architecture, or using undocumented or unsupported interfaces can and likely will require some additional effort to port.)

Several manuals on porting from OpenVMS VAX to OpenVMS Alpha are available in the OpenVMS documentation set, including information on porting VAX Macro32 assembler code to the Macro32 compiler on OpenVMS Alpha, on management differences, on upgrading privileged code, and application migration:

- http://www.hp.com/go/openvms/doc/

Documentation on porting to OpenVMS I64 is available, as well.

Details on the C programming environment are available at:

- http://www.openvms.compaq.com/commercial/c/c_index.html

Details on porting VAX C to HP C are are available at:

- http://www.openvms.compaq.com/commercial/c/index_vax.htm

An OpenVMS Porting Library is available at:

- http://www.openvms.compaq.com/ebusiness/Technology.html

Information on the Enterprise Toolkit, a Visual-based development environment for developing applications for OpenVMS using a Microsoft platform, is available at:

- http://www.openvms.compaq.com/commercial/et/et_index.html

Details on DCE, CORBA, BridgeWorks, and COM/DCOM middleware is available at:

- http://www.openvms.compaq.com/commercial/middleware.html

Information on the COE standards is available at:

- http://diicoe.disa.mil/coe/

A wide variety of programming development tools and middleware are available as commercial products (eg: DECset, IBM WebSphere MQ—formerly MQseries), and various tools are also available as shareware or as Freeware. Please see other sections of this FAQ, and please see:

- http://www.hp.com/go/dspp_catalog

2.15 What resources are available to OpenVMS software developers?

The HP Developer and Software Product Partner (DSPP) program is open to and intended to support and to assist HP OpenVMS software partners, consultants, and service providers:

- http://www.hp.com/dspp/

DSPP provides members with various benefits, please see the website for details.

For those not familiar with the DSPP program or with its history, the DIGITAL Association of Software and Application Partners (ASAP) program and the DIGITAL Independent Software Vendors Network (ISVN) program were incorporated into the Compaq CSA program, and the CSA program has subsequently been incorporated into the HP DSPP program.

Please see Section 2.8.3 for additional details on the DSPP program.

2.16 memory management, resource management, process scheduling, etc?

So you have been instructed to write a school research paper on OpenVMS, and you need technical content on the OpenVMS Virtual Memory System, on any memory segmentation, on OpenVMS Resource Management, on the OpenVMS File System, on the OpenVMS user interface, etc.

Invariably, your professor/instructor/teacher will ask you a series of questions. Most commonly, the questions will request descriptions of one or more of the following items, and at varying levels of detail:

- process scheduling algorithm(s)
- Interprocess comunications
- Process or system synchronization constructs
- Memory management and/or virtual memory implementation
- RMS or XQP file structures
- Resource management
- History of HP OpenVMS
- History of Compaq and/or of Digital Equipment Corporation (DEC)

Any particular presentation or research paper, and particularly a scholastic presentation, can have many different potential target audiences, and very different presentation levels. Further, the usual underlying reason for scholastic presentations and scholastic research projects really has little to do with the subject matter, it is a task specifically intended to teach the student(s) (eg: you) how to perform the research. The instructor already knows most of (all of?) the information that you have been asked to collect.

For very technical details on OpenVMS and OpenVMS internals, the book you want is the Internals and Data Structures Manual (IDSM), available in your school or computing center library, and the IDSM can also be purchased. Additional technical details of the Alpha microprocessor are available in the Alpha Architecture Reference Manual documentation that is available for download. (Pointers to Alpha technical documentation are available in Section 14.6, and elsewhere.)

For higher-level (less technical) details, the OpenVMS documentation set is available on-line. The Programming Concepts and the File Systems manual are probably the best manuals to start with, depending on the particular level of detail the research requires.

And please understand the hesitation of various folks to provide you with a completely-written research report on your topic. Why? We might have to work with you after you graduate—you need to know how to perform at least basic research on your own, regardless of the topic.

2.17 Basic Units of Measurement?

OpenVMS and the underlying hardware use various units of measurement for disk and memory storage, and related abbreviations also typically exist. This section covers the most common units, and the associated abbreviations.

2.17.1 How many bytes are in a disk block?

A disk block is the minimum unit of disk storage allocation in OpenVMS.

Under OpenVMS VAX and OpenVMS Alpha, the disk volume block size is consistent, with each block containing 512 bytes.

The minimum disk allocation granularity actually permissible (in the ODS-2 and ODS-5 volume structures commonly used on OpenVMS) is determined on a per-volume basis, and is typically based on a combination of the total number blocks on the disk volume and the total size of the volume storage bitmap. The allocation granularity is known as the volume cluster factor—the cluster factor is the number of blocks in a disk cluster, and it is the smallest number of blocks that can be allocated on a particular disk volume.

Prior to OpenVMS V7.2, the maximum permissible size of the bitmap requires larger cluster factors as volume sizes increase. Starting with V7.2, the bitmap can be larger, and cluster factors as small as one block can be used.

The number of bytes in a file can be determined by multiplying the number of blocks allocated for the file times the number of bytes in a block. For sequential files (only), the FFB (XAB$W_FFB, in the File Header XAB) value can be used to find out how much of the last (XAB$L_EBK) block is used. FFB and EBK are meaningful only for sequential files, and only in a limited context—partial block allocations are not permitted. For other file formats, the EOF marker is not meaningful.

Disk allocations always occur only in units of the cluster factors, which can be from one block up to (potentially) clusters of eighteen blocks or more, depending on the volume cluster factor. (OpenVMS V7.2 and later optionally provide for a cluster factor of one up to volumes of approximately 137 gigabytes.)

OpenVMS assumes that the device driver and the underlying storage device will present the file system with addressable units of storage of 512 bytes in size, or the appearance of same. Various third-party CD-ROM devices, for instance, support only 2048 byte blocks, and such devices are incompatible with the standard OpenVMS device drivers.

To determine the number of bytes required for a file from DCL, one option uses the f$file_attributes item EOF, multiplied by the size of a block in bytes (512). This does not account for the unused space in the last block of a sequential file, but it also does not have to differentiate sequential files from other files.

2.17.2 How many bytes are in a memory page?

A memory page is the minimum unit of memory allocation in OpenVMS. With OpenVMS VAX, the memory page size matches the disk block size: it is always 512 bytes.

With OpenVMS Alpha, the memory page size is variable, and it can range from 8192 bytes (8 kilobytes) up to 64 kilobytes. The current system page size can be determined using the sys$getsyi or f$getsyi PAGE_SIZE item. Programs with hardcoded constants for the memory page size (or page alignment) should always assume a page size of 64 kilobytes.

On OpenVMS I64, the memory page size is also variable, ranging from 4096 bytes (4 kilobytes) up to 256 megabytes (MB) and potentially up to 4 gigabytes (GB). As with OpenVMS Alpha, sys$getsyi and f$getsyi and the PAGE_SIZE itemcode can and should be used to determine the current system page size. In general, OpenVMS I64 will use a page size of 8 kilobytes, or larger.

On OpenVMS Alpha and on OpenVMS I64, a 512 byte area of memory—equivalent in size to an OpenVMS VAX memory page—is often refered to as a ‘‘pagelet’’.

2.17.3 How do I convert? Disk Blocks? KB, MB, GB, TB?

The smallest granularity of disk storage addressing is called a disk block, or sometimes a disk sector. Groups of disk blocks are usually organized together into the smallest unit of storage that can be allocated, and this unit is called a disk cluster. The number of blocks in a cluster is the cluster factor, and is established when the disk volume is initialized.

Each individual disk block is composed of five hundred twelve (512) bytes, or one-half kilobyte. Each byte is comprised of eight bits. A bit represents the smallest unit of information, typically refered to as a one or a zero.

OpenVMS tends to uses base two notation for disk storage, while disk storage capacity specifications from most storage vendors will generally use base ten notation.

An OpenVMS disk block is 512 bytes in size; this is one-half kilobyte in base two notation.

The following table describes the prefix, the abbreviation, and the associated base ten (as used by marketing and by storage vendors) and base two (OpenVMS and various other operating systems) values.

```
          Base Ten                           Base Two
          --------------------------------   -------------------------
Kilobyte  (KB)  10**3                1000   2**10                1024
Megabyte  (MB)  10**6             1000000   2**20             1048576
Gigabyte  (GB)  10**9          1000000000   2**30          1073741824
Terabyte  (TB)  10**12      1000000000000   2**40       1099511627776
Petabyte  (PB)  10**15   1000000000000000   2**50    1125899906842624
Exabyte   (EB)  10**18 1000000000000000000   2**60 1152921504606846976
```

The base ten representation of the 2**40 value is 1099511627776, which is obviously rather ugly. When viewed as a base eight or base sixteen (octal or hexadecimal, respectively) value, the value is far nicer. Specifically, the value is 10000000000 and 40000000 when represented in octal and hexadecimal, respectively.

FAQ Notation

Within the OpenVMS FAQ, a thousand bits (either assuming base two or base ten, as determined by the context) is refered to as a kilobit, and is always represented by the appreviation Kb, while a thousand bytes is refered to as a kilobyte and is always abbreviated as KB. Similar notational usage also holds for Megabits (Mb) and Megabytes (MB), and for the various other units.

OpenVMS operating system references to system and storage are generally to the base-two version (eg: 1024, in the case of a kilobyte or kilobit) while storage hardware references and hardware specifications are generally to the base-ten version (eg: 1000).

To convert OpenVMS disk blocks to (base two) kilobytes (KB; 1024 bytes), simply divide by two. To convert blocks to (base two) megabytes, divide by 2048. Blocks to (base two) gigabytes (GB), divide by 2097152. These particular divisions can also be performed using bitshifts: to divide a value by two, shift the binary value rightward by one bit position.

To convert OpenVMS disk blocks to (base ten) kilobytes, divide by approximately 1.953125.

For those folks with an interest in odd applications for prefixes, and particularly for those folks also rummaging around deep within the OpenVMS operating system, a microfortnight is approximately one second.

3 Documentation

If you are searching for something here,
please consider using the text-format FAQ.

3.1 Where can I find online copies of OpenVMS manuals?

The HP OpenVMS and HP Layered Product documentation is copyrighted material.

HTML format on-line product documentation sets for specific HP OpenVMS products are presently available at:

- http://www.hp.com/go/openvms/doc/

Documentation is offered on separately orderable CD-ROM media through a subscription to the Consolidated On-Line Documentation (ConOLD) product (see Section 2.6.) ConOLD manuals are readable with BNU, a viewer that is supplied with the documentation distribution. BNU can display HTML, Bookreader, and documentation in other formats.

MGBOOK, a viewer for Bookreader-format documentation is available for character-cell terminals (eg. VTxxx) via the WKU VMS Freeware file server – see question Section 13.1 for details.

Information on the XPDF DECwindows PDF viewer for OpenVMS is available in Section 13.1, and XPDF kits are available on various Freeware distributions. An alternative on OpenVMS Alpha uses the Adobe Java PDF viewer, though this viewer is generally considered to be both slower and more resource-intensive when compared to the XPDF viewer.

3.2 What online information and websites are available?

On your OpenVMS system, the HELP command can provide a wealth of information, not only on DCL commands but on system services (HELP System_Services) and Run-Time Library routines (HELP RTL_Routines). The introduction displayed when you type the HELP command with no additional keywords provides further pointers.

OpenVMS Marketing runs a web server at http://www.hp.com/go/openvms/

Here, you will find product information, strategy documents, product roadmaps, the contents of the latest OpenVMS Freeware CD-ROM and more.

Table 3–1 OpenVMS Websites

URL	Sponsor
HP OpenVMS Marketing	
	http://www.hp.com/go/openvms/
Encompass DFWCUG	
	http://www.openvmshobbyist.org/
Arne Vajhøj	
	http://www.levitte.org/~ava/
Saiga Systems	
	http://www.saiga.com/
Wayne Sewell	
	http://www.tachysoft.com/
proGIS Software	
	http://www.progis.de/openvms.htm
Jeff Cameron	
	http://www.jcameron.com/vms/
David Mathog’s (quite useful) information about OpenVMS.	
	http://saf.bio.caltech.edu/soft_doc.html
Cracking	
“The Beave” Includes system cracking information that can be of interest to OpenVMS System Managers, and to OpenVMS Network and Security Managers. This information is available at the Deathrow cluster.	
	http://manson.vistech.net/ht_root/Hack-VMS-faq
Undocumented Features	
DECUS Deutschland	
	http://zinser.no-ip.info/www/eng/vms/qaa/undoc.htmlx
Arne Vajhøj	
	http://www.levitte.org/~ava/vms_tip.htmlx
The OpenVMS Freeware contains various examples of undocumented features and interfaces	

Table 3–1 (Cont.) OpenVMS Websites

URL	Sponsor
	http://www.hp.com/go/openvms/freeware/
Comparisons of UNIX and Linux shell commands and DCL Commands	
	http://www.mcsr.olemiss.edu/unixhelp/VMStoUNIX.html
	http://wwwvms.mppmu.mpg.de/vmsdoc/UNIX_VMS_CMD_XREF.HTML
Comparisons of emacs and OpenVMS text editor commands	
	http://www.unh.edu/cis/docs/vms-to-unix/Emacs/cheat-sheet.html
Bibliographies	
	http://www.openvms.org/pages.php?page=Books
	http://www.levitte.org/~ava/vms_book.htmlx
Introductory	
	Please see Table 3–2 for listings of introductory web sites and related materials.
Programming	
An OpenVMS Programming FAQ	
	http://www.pdv-systeme.de/users/martinv/VMS_Programming_FAQ.html
Networking	
Tutorial information and tips for connecting OpenVMS systems to the Internet	
	http://www.tmesis.com/internet/
Documentation and Specifications for DECnet Phase IV, DECnet task-to-task DCL examples, and a whole lot more.	
	http://www.hp.com/go/openvms/wizard/
HP OpenVMS Documentation	
	Please see Table 3–2 for listings of documentation web sites and related materials.
System Performance	
	See Section 14.2.
Patch (ECO) Kits	
	For the HP Services FTP server hosting Various contract-access and non-contract access ECO (patch) kits, see section Section 5.17.
Catalogs and Pricing	
HP Product QuickSpecs and product information	
	http://www.hp.com/go/productbulletin/

Table 3–1 (Cont.) OpenVMS Websites

URL	Sponsor
http://www.compaq.com/products/quickspecs/soc_archives/SOC_Archives.html	The HP Systems and Options Catalog (SOC) archive
Hardware and Software Archives	
http://vax.sevensages.org/index.html	The VAXarchive, including hardware and software information
http://www3.sympatico.ca/n.rieck/docs/alpha_diary.html	A VAX to Alpha upgrade diary
http://www.montagar.com/~patj/dec/hcps.htm	Scanned versions of old DIGITAL manuals from DFWCUG
http://www.compaq.com/support/techpubs/qrg/index.html	A wide variety of HP VAX, Alpha, platform and other product documentation. Some introductory, some technical.
http://dtrwiz.home.netcom.com/	dtrwiz's Datatrieve website

3.3 How do I extract the contents of a HELP topic to a text file?

To extract all the text of a HELP topic (and its subtopics) to a text file for perusal with a text editor, printing out, etc., use the following command:

```
$ HELP/OUT=filename.txt help-topic [help-subtopic]
```

If the help text you want is not in the standard help library (for example, it's help for a utility such as MAIL that has its own help library), add /LIBRARY=libname after the HELP verb. To see the names of help library files, do a directory of SYS$HELP:*.HLB.

3.4 Does OpenVMS Marketing have an e-mail address?

Yes - if you can't get the answers to marketing questions elsewhere, if you have comments or complaints about OpenVMS, send mail to openvms-info{atsign}hp.com. This address is *not* a support channel, and is solely intended to provide informal method to communicate directly with members of OpenVMS Marketing.

3.5 Where can I learn about OpenVMS executive internals?

The OpenVMS Internals and Data Structure manual (IDSM) explains how the OpenVMS executive works. The book covers the operating system kernel: process management; memory management; the I/O subsystem; and the mechanisms that transfer control to, from, and among these. It gives an overview of a particular area of the system, followed by descriptions of the data structures related to that area and details of the code that implements the area.

The first edition of the OpenVMS Alpha internals book describes Version 1.5. Although there have been several releases of OpenVMS Alpha since Version 1.5 (V6.1, V6.2, V7.0, V7.1, etc) and many details in the book are no longer accurate, it continues to provide a strong conceptual description of OpenVMS internals.

This book has been split into five pieces, each to be updated separately. The first such volume, published in early 1997, was "OpenVMS Alpha Internals and Data Structures: Scheduling and Process Control," which covers the Version 7.0 implementation of true multithreading and the changed scheduling model it implies.

The internals books are available through Digital Press, see Section 3.6

3.6 Where can new users find tutorial information about OpenVMS?

First, see if your local site has information on this topic. Each site can have site-specific features and configuration. Some sites will have site-specific new user's documentation, covering various site-specific things that are difficult or impossible for the general OpenVMS documentation to cover.

3.6.1 Tutorial Websites?

Various websites with OpenVMS information are available; Table 3–2 contains some suggested URLs.

Table 3–2 OpenVMS Tutorial and Documentation Websites

URL	Sponsor
Introductory	

Table 3–2 (Cont.) OpenVMS Tutorial and Documentation Websites

URL	Sponsor
http://www.levitte.org/~ava/vms_faq.htmlx http://saf.bio.caltech.edu/vms_sheet.html http://seqaxp.bio.caltech.edu/www/vms_beginners_faq.html	Various introductory materials
http://www.montagar.com/openvms_class/	Members of the Encompass DFWCUG maintain a website with many materials available, including an Overview of OpenVMS, an Introduction to DCL and the TPU Editor, Advanced DCL Command Procedures, OpenVMS Operations: Batch, Print, Tape, an Introduction to OpenVMS Management, to OpenVMS User Management, to OpenVMS Network Management, and to OpenVMS Cluster Management. These training materials have been presented at various DECUS symposia.
http://www.mcsr.olemiss.edu/unixhelp/VMStoUNIX.html	A comparison table of various command-level tasks, with information on the UNIX and Linux shell command(s), and on the OpenVMS DCL command(s).
HP OpenVMS Documentation	
http://www.hp.com/go/openvms/doc/	Various introductory guides as well as more advanced manuals are available in the OpenVMS and layered product documentation set.
HP OpenVMS Training	
http://www.compaq.com/training/home.html http://www.openvms.compaq.com/wbt/index.html	HP offers training information and Technical Resource Kits (TRKs) and other Training for OpenVMS. An OpenVMS certification (testing) program is also available.
http://www.jcameron.com/vms/	An OpenVMS Quiz
http://www.CCSScorp.com/	CCSS Interactive Learning has OpenVMS training materials
http://www.acersoft.com/	AcerSoft Training information, and Shannon Knows Punditry

Table 3–2 (Cont.) OpenVMS Tutorial and Documentation Websites

URL	Sponsor
http://www.mindiq.com/	MindIQ training information
http://www.quadratrix.be/	Quadratrix; OpenVMS training, products and services; affiliated with Global Knowledge and KeyJob

3.6.2 Books and Tutorials?

Some of the OpenVMS books that are now or that have been available from the Elsevier Digital Press imprint

- http://www.digitalpressbooks.com/

are listed in Table 3–3:

Table 3–3 DP Books

Title and Author	ISBN
Getting Started with OpenVMS Michael D. Duffy	1-55558-279-6
Getting Started with OpenVMS System Management, 2nd Edition David Donald Miller, et al	1-55558-243-5
Introduction to OpenVMS, 5th Edition Lesley Ogilvie Rice	1-55558-194-3
Introduction to OpenVMS David W Bynon	1-878956-61-2
OpenVMS Alpha Internals: Scheduling and Process Control	1-55558-156-0
OpenVMS AXP Internals and Data Structures: Version 1.5	1-55558-120-X
OpenVMS System Management Guide Baldwin, et al	1-55558-143-9
The OpenVMS User's Guide, Second Edition Patrick Holmay	1-55558-203-6

Table 3–3 (Cont.) DP Books

Title and Author	ISBN
Using DECwindows Motif for OpenVMS Margie Sherlock	1-55558-114-5
VAX/VMS Internals and Data Structures: Version 5.2	1-55558-059-9
Writing Real Programs in DCL, Second Edition Hoffman and Anagnostopoulos	1-55558-191-9
Writing OpenVMS Alpha Device Drivers in C Sherlock and Szubowicz	1-55558-133-1

Within the above table, no attempt is made to track which books are currently in print, or are currently out of print.

For various featured OpenVMS books, also please see the books link at the OpenVMS website:

- http://www.hp.com/go/openvms

For a bibliography of various OpenVMS books, please see:

- http://www.levitte.org/~ava/vms_book.htmlx

3.7 What OpenVMS mailing lists and forums are available?

Various OpenVMS mailing lists are available, with some of the available lists detailed in Table 3–4, as are the various discussion forums in Table 3–5.

Table 3–4 OpenVMS Mailing Lists

Subscription	Interest Area
OpenVMS Freeware archive announcement list	FSupdate@goatley.com FSupdate-request@goatley.com[1]
Two-way echo of vmsnet.internals	VMSnet-Internals@goatley.com VMSnet-Internals-request@goatley.com[1]

---

[1] This is the subscription address. Usually, you will want to send a mail message with no subject line, and a SUBSCRIBE or HELP command in the body of the mail message.

Table 3–4 (Cont.) OpenVMS Mailing Lists

Subscription	Interest Area
OpenVMS Alpha Internals discussions	Alpha-IDS@goatley.com Alpha-IDS-request@goatley.com[1]
BLISS discussions	BLISSters@goatley.com BLISSters-request@goatley.com[1]
Process Software MultiNet mailing list (news gateway)	Info-MultiNet@process.com Info-MultiNet-request@process.com[1]
Process Software TCPware mailing list (news gateway)	Info-TCPware@process.com Info-TCPware-request@process.com[1]
Process Software PMDF mailing list (news gateway)	Info-PMDF@process.com Info-PMDF-request@process.com[1]
The Software Resources International (SRI) CHARON-VAX VAX emulator package	CHARON-VAX-Users@process.com CHARON-VAX-Users-request@process.com[1]
Info-Zip's Zip & UnZip discussion list	Info-Zip@wku.edu Info-Zip-Request@wku.edu[1]
RADIUS-VMS, a RADIUS server for OpenVMS discussion forum	radius-vms@dls.net radius-vms-request@dls.net[1]
Internet Service Providers (ISPs) running OpenVMS	vms-isps@dls.net vms-isps-request@dls.net[1]
Users of Mark Daniel's WASD web server for OpenVMS VAX and Alpha exists. Information about this list server and details on how to subscribe to the list are available at the referenced website.	http://wasd.vsm.com.au/
VMS Forum	http://www.neurophys.wisc.edu/comp/ava/vms_forum.htmlx

---

[1] This is the subscription address. Usually, you will want to send a mail message with no subject line, and a SUBSCRIBE or HELP command in the body of the mail message.

Various OpenVMS discussion forums are available, with some of the available lists detailed in Table 3–5.

Table 3–5 OpenVMS Discussion Forums

Description	Forum Location
The comp.os.vms newsgroup	news:comp.os.vms
HP ITRC Forums	http://forums.itrc.hp.com/service/forums/familyhome.do?familyId=288
Hunter Goatley: The HG OpenVMS Message Board	http://www.goatley.com/scripts/vmsboard/view.com
Computing.Net: OpenVMS Message Area	http://www.computing.net/openvms/wwwboard/wwwboard.html
Tek-Tips: DEC (HP/Compaq): OpenVMS Forum	http://www.tek-tips.com/gthreadminder.cfm/lev2/3/lev3/19/pid/951
OpenVMS.Org forums	http://www.openvms.org/
OpenVMSHobbyist.Org forums	http://www.openvmshobbyist.org/
Encompasserve (DECUSserve) Notes Conferences	telnet://www.encompasserve.org/

3.8 What is this Ask The Wizard website I've heard about?

The HP OpenVMS Ask The Wizard (ATW) website was an informal area discussing OpenVMS, containing questions and answers on a wide variety of topics.

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

To access a cited topic directly, use the URL filename WIZ_topic-number.HTML, or use the topic search engine. Cited topics are shown in parentheses, and act as unique topic addresses. These should not be confused with the relative topic numbers shown at the site. For example, the topic (1020) can be accessed directly using the URL filename wiz_1020.html, at the web site that the following URL resolves into:

- http://www.hp.com/go/openvms/wizard/

A zip archive (named wizard.zip) containing all of the available topics and questions can be downloaded from the above URL. The wizard.zip zip archive is completely regenerated when/if existing topics posted out to the ATW website are updated. Copies of this wizard.zip archive also generally ship out on the OpenVMS Freeware, as well.

New (informal) questions and discussions are now being directed away from the ATW area to the ITRC area, and specifically to the ITRC discussion forums:

- http://www.itrc.hp.com/

3.9 Where can I find the latest C run-time library manuals?

The C run-time library (RTL) reference documentation has been moved from the C language documentation set to the OpenVMS documentation set. For the most recent version of the C RTL documentation and the OpenVMS standard C library, please see the OpenVMS manuals.

In addition to the user-mode C RTL, there is a second kernel-mode RTL accessable to drivers on OpenVMS Alpha and OpenVMS I64. For details on this second library and on the duplicate symbol errors that can be triggered when this library is referenced during an incorrectly-specified LINK command, please see Section 10.22.1. For general information on this kernel RTL, see the Digital Press book Writing OpenVMS Device Drivers in C. For details, please see the associated OpenVMS source listings distribution.

- http://www.hp.com/go/openvms/doc/

4 Time and Timekeeping

If you are searching for something here,
please consider using the text-format FAQ.

This chapter discusses time, timekeeping, system time synchronization, clock skew and clock drift, implications of using SUBMIT/AFTER=TOMORROW, and other time-related topics.

4.1 A brief history of OpenVMS Timekeeping, please?

Why does OpenVMS regards November 17, 1858 as the beginning of time...

The modified Julian date adopted by the Smithsonian Astrophysical Observatory (SAO) for satellite tracking is Julian Day 2400000.5, which turns out to be midnight on November 17, 1858.

SAO started tracking satellites with an 8K (nonvirtual) 36-bit IBM 704 in 1957 when Sputnik went into orbit. The Julian day was 2435839 on January 1, 1957. This is 11225377 octal, which was too big to fit into an 18-bit field. With only 8K of memory, the 14 bits left over by keeping the Julian date in its own 36-bit word would have been wasted. SAO also needed the fraction of the current day (for which 18 bits gave enough accuracy), so it was decided to keep the number of days in the left 18 bits and the fraction of a day in the right 18 bits of one word.

Eighteen bits allows the truncated Julian Day (the SAO day) to grow as large as 262143, which from November 17, 1858, allowed for 7 centuries. Possibly, the date could only grow as large as 131071 (using 17 bits), but this still covers 3 centuries and leaves the possibility of representing negative time. The 1858 date preceded the oldest star catalogue in use at SAO, which also avoided having to use negative time in any of the satellite tracking calculations.

The original Julian Day (JD) is used by astronomers and expressed in days since noon January 1, 4713 B.C. This measure of time was introduced by Joseph Scaliger in the 16th century. It is named in honor of his father, Julius Caesar Scaliger (note that this Julian Day is different from the Julian calendar that is named for the Roman Emperor Julius Caesar!).

Why 4713 BC? Scaliger traced three time cycles and found that they were all in the first year of their cyle in 4713 B.C. The three cycles are 15, 19, and 28 years long. By multiplying these three numbers (15 * 19 * 28 = 7980), he was able to represent any date from 4713 B.C. through 3267 A.D.

The starting year was before any historical event known to him. In fact, the Jewish calendar marks the start of the world as 3761 B.C. Today his numbering scheme is still used by astronomers to avoid the difficulties of converting the months of different calendars in use during different eras.

The following web sites:

- http://www.openvms.compaq.com/openvms/products/year-2000/leap.html
- http://www.eecis.udel.edu/~ntp/
- http://www.nist.gov/
- http://www.boulder.nist.gov/timefreq/
- http://www.tondering.dk/claus/calendar.html
- http://es.rice.edu/ES/humsoc/Galileo/Things/gregorian_calendar.html
- http://www.timeanddate.com

are all good time-related resources, with some general and some specific to OpenVMS.

4.1.1 Details of the OpenVMS system time-keeping?

4.1.1.1 VAX hardware time-keeping details...

4.1.1.1.1 TOY clock

This is battery backed up hardware timing circuitry used to keep the correct time of year during rebooting, power failures, and system shutdown. This clock only keeps track of months, days, and time. The time is kept relative to January 1st, at 00:00:00.00 of the year the clock was initiailized.
The VAX Time-Of-Year (TOY) clock (used to save the time over a reboot or power failure) is specified as having an accuracy of 0.0025%. This is a drift of roughly 65 seconds per month.

The VAX Interval Time is used to keep the running time, and this has a specified accuracy of .01%. This is a drift of approximately 8.64 seconds per day.

Any high-IPL activity can interfere with the IPL 22 or IPL 24 (this depends on the VAX implementation) clock interrupts—activities such as extensive device driver interrupts or memory errors are known to slow the clock.

4.1.1.1.2 EXE$GQ_SYSTIME

This is the OpenVMS VAX system time cell. This cell contains the number of 100ns intervals since a known reference. This cell is incremented by 100000 every 10ms by an hardware interval timer.

4.1.1.1.3 EXE$GQ_TODCBASE

This cell contains the time and date the system time was last adjusted by EXE$SETTIME. It uses the same format as EXE$GQ_SYSTIME. On adjustment of the system time a copy of EXE$GQ_SYSTIME is stored in this cell in both memory and on disk. This cell is used to get the year for the system time.

4.1.1.1.4 EXE$GL_TODR

This cell contains the time and date the system time was last adjusted by EXE$SETTIME. It uses the same format as the time of year clock. On adjustment of the system time this cell gets saved back to both memory and disk. The contents of this cell are used to test the validity of the TOY clock.
The system parameters SETTIME and TIMEPROMPTWAIT determine how the system time will be set.

- IF SETTIME = 0 and the TOY clock is valid
  THEN the contents of the TOY clock are compared to those of EXE$GL_TODR. IF the TOY clock is more than a day behind EXE$GL_TODR
    THEN the TOY clock is presumed invalid.
  - IF the TOY clock is within a day of EXE$GL_TODR
    THEN the system time is calculated as follows:
  - EXE$GQ_SYSTIME = EXE$GQ_TODCBASE + ((TOY_CLOCK - EXE$GL_TODR) * 100000)

- IF SETTIME = 1 or the TOY clock is invalid
  THEN the value of TIMEPROMPTWAIT determines how to reset the time of year. IF TIMEPROMPTWAIT > 0
    THEN the user is prompted for the time and date, for a length of time equal to TIMEPROMPTWAIT microfortnights.
  - IF TIMEPROMPTWAIT = 0
    THEN the time of year is the value of EXE$GL_TODR + 10ms.
  - IF TIMEPROMPTWAIT < 0
    to proceed until they do so.
  - THEN the user is prompted for the time and date and unable

When booting a CD-ROM containing an OpenVMS VAX system, the system will typically be deliberately configured prompt the user to input the time – this is necessary in order to boot with the correct time.

If either TIMEPROMPTWAIT or SETTIME are set to zero, OpenVMS VAX will use the TOY clock to get the time of year, and the year will be fetched from the distribution medium. The value of the year on the distribution medium (saved within the SYS.EXE image) will most likely be that of when the kit was mastered, and cannot be changed. Unless the current year happens to be the same year as that on the distribution, most likely the year will be incorrect. (Further, with the calculation of Leap Year also being dependent on the current year, there is a possibility that the date could be incorrect, as well.)

4.1.1.2 Alpha hardware time-keeping details...

4.1.1.2.1 Battery-Backed Watch (BB_WATCH) Chip

This is battery backed up hardware timing circuitry used to keep the correct time of year during rebooting, power failures, and system shutdown. This clock keeps track of date and time in 24 hour binary format.
The BB_WATCH time is used to initialize the running system time during bootstrap, and the BB_WATCH time is read when the SET TIME command is issued with no parameters; when the running system time is reset to the value stored in the BB_WATCH. The running system time is written into the BB_WATCH when the SET TIME command is issued with a parameter.

The specification for maximum clock drift in the Alpha hardware clock is 50 parts per million (ppm), that is less than ±0.000050 seconds of drift per second, less than ±0.000050 days of drift per day, or less than ±0.000050 years of drift per year, etc. (eg: An error of one second over a day-long interval is roughly 11ppm, or 1000000/(24*60*60).) Put another way, this is .005%, which is around 130 seconds per month or 26 minutes per year.

The software-maintained system time can drift more than this, primarily due to other system activity. Typical causes of drift include extensive high-IPL code (soft memory errors, heavy activity at device IPLs, etc) that are causing the processing of the clock interrupts to be blocked.

4.1.1.2.2 EXE$GQ_SYSTIME

This is the OpenVMS Alpha system time cell. This cell contains the number of 100ns intervals since November 17, 1858 00:00:00.00. This cell is incremented by 100000 every 10ms by an hardware interval timer.

4.1.1.2.3 EXE$GQ_SAVED_HWCLOCK

This cell is used by OpenVMS Alpha to keep track of the last time and date that EXE$GQ_SYSTIME was adjusted. It keeps the same time format as EXE$GQ_SYSTIME. The value in this cell gets updated in memory and on disk, every time EXE$GQ_SYSTIME gets adjusted.

- The system parameters SETTIME and TIMEPROMPTWAIT determine how the system time will be set.
- If SETTIME = 0
  then EXE$INIT_HWCLOCK reads the hardware clock to set the system time.
  - IF TIMEPROMPTWAIT > 0
    THEN the value of TIMEPROMPTWAIT determines how long the user is prompted to enter the time and date. If time expires and no time has been entered the system acts as if TIMEPROMPTWAIT = 0.
  - IF TIMEPROMPTWAIT = 0
    THEN the system time is calculated from the contents of EXE$GQ_SAVED_HWCLOCK + 1.
  - IF TIMEPROMPTWAIT < 0
    THEN the user is prompted for the time and date and unable to continue until the information is entered.

Unlike the VAX, the Alpha hardware clock tracks the full date and time, not just the time of year. This means it is possible to boot from the CD-ROM media without entering the time at the CD-ROM bootstrap. (This provided that the time and date have been initialized, of course.)

IA-64 (Itanium) hardware time-keeping details to be added...

4.1.1.3 Why does VAX need a SET TIME at least once a year?

Because the VAX Time Of Year (TOY) has a resolution of 497 days, the VAX system time is stored using both the TOY and the OpenVMS VAX system image SYS.EXE. Because of the use of the combination of the TOY and SYS.EXE, you need to issue a SET TIME command (with the time parameter specified) at least once between January 1st and about April 11th of each year, and whenever you change system images (due to booting another OpenVMS VAX system, booting the standalone BACKUP image, an ECO that replaces SYS.EXE, etc).

The SET TIME command (with the current time as a parameter) is automatically issued during various standard OpenVMS procedures such as SHUTDOWN, and it can also obviously be issued directly by a suitably privileged user. Issuing the SET TIME command (with a parameter) resets the value stored in the TOY, and (if necessary) also updates the portion of the time (the current year) saved in the SYS.EXE system image.

This VAX TOY limit is the reason why OpenVMS VAX installation kits and standalone BACKUP explicitly prompt for the time during bootstrap, and why the time value can "get weird" if the system crashes outside the 497 day window (if no SET TIME was issued to update the saved values), and why the time value can "get weird" if a different SYS$SYSTEM:SYS.EXE is used (alternate system disk, standalone BACKUP, etc).

4.1.2 How does OpenVMS VAX maintain system time?

VAX systems maintain an interval clock, and a hardware clock.

The VAX hardware clock is called the TOY ("Time Of Year") clock. The register associated with the clock is called the TODR ("Time Of Day Register").

The TOY clock—as used—stores time relative to January first of the current year, starting at at 00:00:00.00. It is a 100 Hz, 32-bit counter, incremented every 10ms, and thus has a capacity of circa 497 days.

OpenVMS (on the VAX platform) stores system date information—and in particular, the current year—in the system image, SYS$SYSTEM:SYS.EXE.

The TOY is used, in conjunction with the base date that is stored and retrieved from the system image, to initialize the interval clock value that is stored in EXE$GQ_SYSTIME.

Once the interval clock is loaded into the running system as part of the system bootstrap, the system does not typically reference the TOY again, unless a SET TIME (with no parameters) is issued. The interval clock value is updated by a periodic IPL22 or IPL24 (depending on the specific implementation) interrupt. (When these interrupts are blocked as a result of the activity of higher-IPL code—such as extensive driver interrupt activity or a hardware error or a correctable (soft) memory error—the clock will ‘‘loose’’ time, and the time value reported to the user with appear to have slowed down.)

When SET TIME is issued with no parameters, the TOY clock is loaded into the system clock; the running system clock is set to the time stored in the TOY clock. This assumes the TOY clock is more accurate than the system clock, as is normally the case.

On most (all?) VAX systems, the battery that is associated with the TOY clock can be disconnected and replaced if (when) it fails—TOY clock failures are quite commonly caused by a failed nickel-cadmium (NiCd) or lithium battery, or by a failed Dallas chip.

4.2 Keeping the OpenVMS system time synchronized?

To help keep more accurate system time or to keep your system clocks synchronized, TCP/IP Services NTP, DECnet-Plus DTSS (sometimes known as DECdtss), DCE DTS, and other techniques are commonly used. If you do not or cannot have IP access to one of the available time-base servers on the Internet, then you could use dial-up access to NIST or other authoritative site, or you can use a direct connection to a local authorative clock.

There exists code around that processes the digital (ie: binary) format time that is available via a modem call into the NIST clock (the Automated Computer Telephone Service (ACTS) service), and code that grabs the time off a GPS receiver digital link, or a receiver (effectively a radio and a codec) that processes the time signals from radio stations WWV, WWVH, WWVB, or similar.

Processing the serial or hardware time protocols often involves little more than reading from an EIA232 (RS232) serial line from the receiver, something that is possible from most any language. Information on correctly drifting the OpenVMS system clock to match the time-base time is available within the logic of at least one OpenVMS Freeware package. (See Section 4.3 for a few potential hardware options.)

One example of acquring a time-base through local integrated hardware involves the IRIG time format (IRIG-A, -B, -G), a binary signal containing the current time in hours, minutes, seconds and days since the start of the current year. IRIG can also contain the time of day as the number of seconds since midnight. HP Custom Systems and third-party vendors have variously offered IRIG-based reader/generator modules for OpenVMS systems.

One of the easiest approaches is a network-based GPS or other similar receiver. Basically, this is a network server box that provides an NTP server with the necessary hardware for external synchronization. In addition to the antenna and the receiver and processing components, these devices provide a network interface (NIC) and support for an NTP time server, and applications including the NTP support within TCP/IP Services and within various third-party IP stacks can then be used to synchronize with the the NTP information provided by time-base receivers. No other host software is required, and no host configuration steps and no host software beyond NTP are required. (See Section 4.3 for a few potential hardware options.)

Differing time servers (DECnet-Plus DTSS, DCE DTS, NTP, etc) do not coexist particularly well, particularly if you try to use all these together on the same node. Please pick and use just one. (If needed, you can sometimes configure one package to acquire its timebase from another protocol, but one and only one time server package should have direct control over the management of and drifting of the local OpenVMS system time. In the specific case of DECnet-Plus DTSS, older product versions and versions V7.3 and later provide a provider module, a module which permits DTSS to acquire its time from NTP. For details on this, please see the comments in the module DTSS$NTP_PROVIDER.C.)

Unlike DECnet-Plus, TCP/IP Services NTP is not capable of connecting to a time-base other than the network time base or the local system clock. Third-party and open source NTP implementations are available for OpenVMS, as well.

Useful URLs:

- http://www.boulder.nist.gov/timefreq/service/nts.htm

- http://www.boulder.nist.gov/timefreq/service/acts.htm
- http://www.boulder.nist.gov/timefreq/
- http://www.time.gov/

4.3 External time-base hardware?

Here are a few possibilities for providers of a GPS-based receiver with an embedded NTP server, strictly culled from the first few pages of a Google search. Availability, pricing, OpenVMS compatibility and other factors are not known.

- http://www.galleon.eu.com/
- http://www.meinberg.de/english/
- http://www.ntp-servers.com/

For a direct-connected (local, non-IP, non-NTP) link, there are serial options available. Google finds Spectracom Corporation has a NetClock that could be used here, based on a quick look—I do not know if there is OpenVMS host software, but that would be possible to write for the ASCII data stream that the device supports. (Such coding requires knowledge of serial I/O, character processing, and knowledge of the clock drift API mechanisms in OpenVMS—there exists Freeware tools that could be used to learn how to tie into the clock drifting mechanisms of OpenVMS.)

- http://www.spectracomcorp.com/

http://www.spectracomcorp.com/

Information on, and experiences or recommendations for or against these or other similar devices is welcome.

4.3.1 Why do my cluster batch jobs start early?

Your system time is skewed across your cluster members, and the cluster member performing the queue management tasks has a system time set later than the system time of the member running the batch job.

This behaviour is most noticable when using SUBMIT/AFTER=TOMORROW and similar constructs, and use of /AFTER="TOMMOROW+00:01:00" or such is often recommended as a way to avoid this. The combination time value specified should be larger than the maximum expected time skew. In the example shown, the maximum cluster clock skew is assumed less than 1:00.

You can also maintain your system times in better synchronization, with available tools described in Section 4.2 and elsewhere.

4.3.2 Why does my OpenVMS system time drift?

Memory errors, hardware problems, or most anything operating at or above IPL 22 or IPL 24 (clock IPL is system family dependent; code executing at or above the clock IPL will block the processing of clock interrupts), can cause the loss of system time. Clock drift can also be caused by normal (thermal) clock variations and even by the expected level of clock drift.

When clock interrupts are blocked as a result of the activity of high-IPL code—such as extensive driver interrupt activity or a hardware error or a correctable (soft) memory error—the clock will ‘‘loose’’ time, and the time value reported to the user with appear to have slowed down. Correctable memory errors can be a common cause of system time loss, in other words. Heavy PCI bus traffic can also cause lost time.

One bug in this area involved the behaviour of certain graphics controllers including the ELSA GLoria Synergy PBXGK-BB; the PowerStorm 3D10T effectively stalling the PCI bus. See Section 5.16 for details on the ELSA GLoria Synergy controller, and make certain you have the current GRAPHICS ECO kit installed.

Clock drift can also be (deliberately) caused by the activity of the DTSS or NTP packages.

Also see Section 4.1.1.2.1, Section 4.1.1, and Section 4.3.4.

4.3.3 Resetting the system time into the past?

You can resynchronize system time using DCL commands such as SET TIME and SET TIME/CLUSTER, but these commands can and obviously will cause the current system time to be set backwards when the specified time predates the current system time. This time-resetting operation can cause application problems, and can adversely effect applications using absolute timers, applications that assume time values will always be unique and ascending values, and applications.

Setting the time backwards by values of even an hour has caused various run-time problems for applications and layered products. For this reason, this technique was not considered supported during the Year 2000 (Y2K) testing; a system or cluster reboot was strongly recommended as the correct means to avoid these problems.

Application programmers are encouraged to use the time-related and TDF-related events that are available with the $set_system_event system service, and/or to use UTC or similar time, as these techniques can permit the application to better survive retrograde clock events. (There is an ECO to repair problems seen in the DECnet-Plus support for generating TDF events from DTSS, and this applies to V7.3 (expected to be in ECO4 and later) V7.3-1 (expected to be in ECO3 and later) and V7.3-2 (expected to be in ECO1 and later). Apply the most current DECnet-Plus ECO kits for these OpenVMS releases, for best TDF event support from DECnet-Plus.)

See Section 4.3.4 and Section 4.3.1.

4.3.4 How can I drift the OpenVMS system time?

With DECdts and TCP/IP Services NTP, the system time value is "drifted" (rather than changed), to avoid the obvious problems that would arise with "negative time changes". The same basic clock drifting technique is used by most (all?) time servers operating on OpenVMS, typically using the support for this provided directly within OpenVMS.

An example of the technique used (on OpenVMS VAX) to drift the system time is the SETCLOCK tool on the OpenVMS Freeware.

For information on the use of the EXE$GL_TIMEADJUST and EXE$GL_TICKLENGTH cells on OpenVMS Alpha, see *OpenVMS AXP Internal and Data Structures*, located on page 348.

For those areas which switch between daylight saving time (DST) and standard time, the time value is *not* drifted. The time is adjusted by the entire interval. This procedure is inherent in the definition of the switch between DST and standard time. (Do look at either not switching to daylight time, or (better) using UTC as your time-base, if this change-over is not feasible for your environment.)

See Section 4.3.4 and Section 4.3.3.

4.3.5 How can I configure TCP/IP Services NTP as a time provider?

An NTP time provider provides its idea of the current time to NTP clients via the NTP protocol. Most systems are NTP clients, but...

NTP has a heirarchy of layers, called strata. The further away from the actual NTP time source (Internet time servers are at stratum 1), the lower the strata (and the larger the number assigned the statum).

NTP explicity configured at stratum one provides time to NTP operating at lower strata, and the provided time is acquired based on the local system time or via some locally-accessible external time source.

NTP at other (lower) strata both receive time from higher strata and can provide time to lower strata, and automatically adjust the local stratum. The highest stratum is one, and the lowest available stratum is fifteen.

The TCP/IP Services NTP package can operate at any stratum, and can be configured as a peer, as a client, or as a broadcast server. NTP can also provide time to a DECnet-Plus DTSS network, see Section 4.2.

With TCP/IP Services V5.0 and later, the only supported reference clock is the LCL (local system clock). If your system has an excellent clock or if the system time is being controlled by some other time service or peripheral (such as DTSS services, GPS services, a cesium clock, a GPIB controller or other similar time-related peripheral), you can configure NTP to use the system clock as its reference source. This will mimic the master-clock functionality, and will configre NTP as a stratum 1 time server. To do this, enter the following commands in TCPIP$NTP.CONF:

```
server 127.127.1.0 prefer
fudge 127.127.1.0 stratum 0
```

For local-master functionality, the commands are very similiar. Use:

```
server 127.127.1.0
fudge 127.127.1.0 stratum 8
```

The difference between these two is the stratum, and the omission of the prefer keyword. Specifying a higher stratum allows the node to act as a backup NTP server, or potentially as the sole time server on an isolated network. The server will become active only when all other normal synchronization sources are unavailable. The use of "prefer" causes NTP to always use the specified clock as the time synchronization source.

With the TCP/IP Services versions prior to V5.0, the NTP management is rather more primitive. To configure the local OpenVMS system from an NTP client to an NTP server (on TCP/IP Services versions prior to V5.0), add the following line to the sys$specific:[ucx$ntp]ucx$ntp.conf file:

```
master-clock 1
```

Also, for TCP/IP Services prior to V5.0, see the NTP template file:

```
SYS$SPECIFIC:[UCX$NTP]UCX$NTP.TEMPLATE
```

Note that NTP does *not* provide for a Daylight Saving Time (DST) switch-over, that switch must arise from the timezone rules on the local system and/or from the SYS$EXAMPLES:DAYLIGHT_SAVINGS procedure. (Further, there is a known bug in SYS$EXAMPLES:DAYLIGHT_SAVINGS.COM in V7.3, please obtain the available ECO kit.)

For current TCP/IP Services and related OpenVMS documentation, please see:

- http://www.hp.com/go/openvms/doc/

4.4 Managing Timezones, Timekeeping, UTC, and Daylight Saving Time?

You will want to use the command procedure:

- SYS$MANAGER:UTC$TIME_SETUP.COM

to configure the OpenVMS Timezone Differential Factor (TDF) on OpenVMS V6.0 and later. Select the BOTH option. This configures the OpenVMS TDF settings, though it may or may not configure the TDF and the timezone rules needed or used by other software packages. Please do NOT directly invoke the following command procedures:

- SYS$MANAGER:UTC$CONFIGURE_TDF.COM ! do not directly use
- SYS$MANAGER:UTC$TIMEZONE_SETUP.COM ! do not directly use

TCP/IP Services V5.0 and later use the OpenVMS TDF, UTC, and timezone support. Earlier versions use a TDF mechanism and timezone database that is internal to the TCP/IP Services package. Also on the earlier versions, the TDF must be manually configured within TCP/IP Services, in addition to the OpenVMS configuration of the TDF.

DECnet-Plus in V7.3 and later uses the OpenVMS TDF, UTC, and timezone support, and displays its timezone prompts using UTC$TIME_SETUP.COM. Earlier versions use a TDF TDF mechanism, timezone database, and automatic switch-over that is internal to the DECnet-Plus package. Also on earlier versions, the TDF must be configured within the DECnet-Plus DECdtss package, in addition to the OpenVMS configuration of the TDF.

Application code using HP C (formerly Compaq C, formerly DEC C) will use the OpenVMS UTC and TDF mechanisms when the C code is compiled on OpenVMS V7.0 and later (and when the macro _VMS_V6_SOURCE is NOT defined). HP C does NOT use the OpenVMS UTC and TDF mechanisms when the C code is compiled on OpenVMS releases prior to V7.0, or when the preprocessor declaration _VMS_V6_SOURCE is declared.

DCE DTS TDF management details to be determined.

In OpenVMS Alpha V6 releases (V6.1, V6.2, V6.2-1Hx, etc), the TDF value is written to SYS$BASE_IMAGE.EXE. With OpenVMS Alpha V7.0 and later and with OpenVMS VAX V6.0 and later, SYS$SYSTEM:SYS$TIMEZONE.DAT contains the TDF. This means that OpenVMS Alpha systems will need to have the TDF value reset manually—usually within SYSTARTUP_VMS.COM—on reboots prior to V7.0.

During OpenVMS Bootstrap, the SYSINIT module reads SYS$TIMEZONE.DAT to acquire the TDF for use in the system global cell EXE$GQ_TDF. This is done to ensure that the system boots with a valid TDF (a value which may be zero). The UTC system services get the TDF from this cell. These services, as well as the HP C RTL, must have a valid TDF. (Prior to OpenVMS V7.3, if either DECnet-Plus or DECnet/VAX Extensions is configured and run, the image DTSS$SET_TIMEZONE.EXE is invoked and can override the TDF and timezone rule settings from SYSINIT or from UTC$TIME_SETUP.COM— this image runs even if DTSS is disabled. If the settings do not match (due to inconsistencies in timezone specification in UTC$TIME_SETUP.COM and NET$CONFIGURE.COM), DTSS will reset the values to match its definitions.)

Prior to OpenVMS V7.3, daylight saving time (DST) switchover is handled automatically only when DCE DTS or DECnet-Plus DTSS is in use. In V7.3, OpenVMS can be configured to automatically switch over to daylight time, and also generates an event that interested applications can use to detect the switch-over between standard time and daylight time.

The manual switchover between daylight time and standard time is correctly accomplished via the SYS$EXAMPLES:DAYLIGHT_SAVINGS.COM command procedure procedure.

Note

NTP (alone) does *NOT* provide automatic switch-over.

Note

The DST switch-over does *NOT* drift the time value; the switch-over applies the entire difference as a unit, as is standard and expected practice. (Do look at either not switching to daylight time, or (better) using UTC as your time-base, if this one-hour change is not feasible within

your environment.) (For information associated with drifting the systen time, please see Section 4.3.4.)

If you switch the TDF or DST setting, you will also want to restart or reconfigure any time-sensitive applications (those not using the time differential factor (TDF) change event available in V7.3 and later). Examples of these applications can include the need to restart the NFS client and NTP. (In the case of NTP, will want to try to ‘‘drift’’ the time (see Section 4.2 and see Section 4.3.4), and will find that the DST switch-over will exceed the NTP-defined maximum threshold allowed for drifting. Hence the NTP restart is presently required.)

4.4.1 Creating, Updating and Managing Timezone Definitions?

One issue with the UTC implementation on OpenVMS is the behaviour of C functions and other programs that use SYS$TIMEZONE_RULE; the OpenVMS mechanism assumes all control over the timezone and the daylight time switchover. This allows calculation of the time by the C library and various applications.

This can be incompatible with a system or application that requires manual modifications to the DST or TDF settings, or that requires a local or customized timezone definition. For such a site to ensure the timekeeping is correct, the site must provide procedure that sets the local time and the TDF when the SYS$TIMEZONE_RULE says to do it.

If a site requires a non-standard time switch-over, as in coordinating with a shift change or due to changes in the local or regional timezone rules, the site will need to use the zic compiler to create a custom timezone rule.

Additionally, applications may need to have special actions taken or actions queued just before the time change takes effect. If the application source code is available, one of the best ways to handle this is via the TDF and time-change notification events available via the OpenVMS sys$set_system_event system service.

For information on zic and related tools used to manage the OpenVMS Timezone database, please see the HP C Run-time Library Utilities Reference Manual—though the title would imply otherwise, this particular manual is part of the OpenVMS documentation set, and not part of the HP C (formerly Compaq C, formerly DEC C) documentation set.

For related information, see Section 4.4.1.1.

4.4.1.1 Customizing or Updating your TDF (Timezone) Setting?

Individual, local, and regional differences on the use (or the lack of use) of Daylight Saving Time (DST) are quite common, as are occasional regulatory changes to the particular applicable regional DST settings. (eg: The United States Government is expecting to change its DST rules starting in March of 2007; please see Section 4.4.1.2 for details.)

DST-related ECO kits are now available for various of the supported releases, please see the ECO FTP site for details.

If you need or wish to add, modify or remove DST rules for your area, or otherwise alter the rules for your local area, you will probably end up creating a variation to an existing timezone rule, or potentially simply downloading a new set of DST rules. This requirement can arise, for instance, if your local region changes its timezone rules, or if you are using an OpenVMS release that does not have an ECO kit for the US DST rules available.)

The necessary zone line to add for support of the hypothetical new WhereEverLand timezone will probably look something like this:

```
# Zone  NAME            GMTOFF  RULES/SAVE      FORMAT  [UNTIL]
Zone    WhereEver       2:00    -               WhereEver
```

The OpenVMS source files for the timezone rules are stored here:

```
SYS$COMMON:[SYS$ZONEINFO.SYSTEM.SOURCES]
```

You’ll then want to use the zic compiler to compile your own new timezone definition, or to compile a new set of timezone definitions that have been freshly downloaded from a published source.

The zic compiler is documented in the OpenVMS Documentation Set, and specifically in the HP C Run-Time Library Reference Manual. (Despite the name of this manual, it is part of the OpenVMS documentation set and not of the C manuals.)

Once you have created and compiled a new timezone rule (or have downloaded and have compiled a whole new set of timezone rules), use the SYS$MANAGER:UTC$TIME_SETUP.COM to select the new timezone if necessary—with V7.3 and later, this tool will automically notice the new timezone and will offer it, on earlier releases, you may/will have to hack the code of the tool somewhat to allow it to present the new timezone rule. (If an existing timezone rule is simply changing, you don’t need this re-selection step.)

Note

As mentioned in Section 4.4.2, please don't modify or redefine the TZ logical name (found on older configurations), or the SYS$TIMEZONE_NAME logical name, or any other time- or timezone-related logical names directly yourself. Rather, please use the zic compiler and/or the UTC$TIME_SETUP.COM procedure.

For various published timezone rules or updated to same, see the tar.gz files (these are gzipped tar archives) available at:

- ftp://elsie.nci.nih.gov/pub/

These are gzipped tar archives, and are the pubished source used for the OpenVMS timezone rules on OpenVMS V7.3 and later, and within the predecessor C run-time environment timezone support used on older OpenVMS releases. You'll need to first gunzip and then use vmstar to unpack and access the contents of the archives.

The published timezone rules include the effective date ranges for the individual rules, so you can reload your rules prior to a particular set of new rules becoming effective. The effective dates for the particular timezone rules are additionally necessary to allow the appropriate translation of older dates and times within the appropriate historical context of the particular date and time value.

For related information, see Section 4.4.1.

4.4.1.2 US Daylight Time Changes Starting 1-Mar-2007?

The United States Federal Government is presently expecting to change its DST rules starting in March, 2007. (The change-over date and the planned change itself has not come to pass as of this writing, hence the phrasing used.)

As amended, US daylight time will be increased to be effective from the second Sunday in March through the first Sunday of November. Other countries, US local political geographies and businesses may or may not follow suite and implement these changes, obviously.

For further regulatory details, see the US Uniform Time Act of 1966 (15 U.S.C 260a(a)), as amended by the Energy Policy Act of 2005.

For details on how to create, customize or to download new rules and to update your local timezone rules, please see Section 4.4.1.1.

4.4.2 Timezones and Time-related Logical Names?

Various logical names are used to manage time and timezones, and you should avoid direct modification of these logical names as the implementations are subtle and quick to change. As discussed in section Section 4.4.3, you will want to use the following command procedure to maintain the time and the timezone:

- SYS$MANAGER:UTC$TIME_SETUP.COM

If you want to venture into uncharted territories and modify the TDF used within older releases of TCP/IP Services—within releases prior V5.0—you can attempt to use the following undocumented commands:

```
SET TIME/DIFF=[positive or negative TDF integer]
GENERATE TIME
```

to reset the value of the logical name UCX$TDF.

Prior to OpenVMS V7.3, the command:

```
$ SETTZ :== $SYS$SYSTEM:DTSS$SET_TIMEZONE
$ SETTZ MODIFY
```

can be used to modify the settings of the SYS$TIMEZONE_DAYLIGHT_SAVING, SYS$TIMEZONE_DIFFERENTIAL, and SYS$TIMEZONE_NAME system logical names based on the SYS$TIMEZONE_RULE.

The following are other TDF-related logical names used/available on OpenVMS systems, with typical daylight time and standard time settings for the US Eastern Time (ET) timezone.

```
$daylight_time:
$ DEFINE/SYSTEM/EXECUTIVE MAIL$TIMEZONE EDT
$ DEFINE/SYSTEM/EXECUTIVE NOTES$TIMEZONE "-0400 EDT"
$ DEFINE/SYSTEM/EXECUTIVE LISP$DAYLIGHT_SAVING_TIME_P true  ! Not 'EDT'
$ DEFINE/SYSTEM/EXECUTIVE LISP$TIME_ZONE 05   ! Constant
$
$standard_time:
$ DEFINE/SYSTEM/EXECUTIVE MAIL$TIMEZONE EST
$ DEFINE/SYSTEM/EXECUTIVE NOTES$TIMEZONE "-0500 EST"
$ DEFINE/SYSTEM/EXECUTIVE LISP$DAYLIGHT_SAVING_TIME_P false ! Not 'EST'
$ DEFINE/SYSTEM/EXECUTIVE LISP$TIME_ZONE 05   ! Constant
$
$ DEFINE/SYSTEM/EXECUTIVE UCX$NFS_TIME_DIFFERENTIAL -
    'f$integer(f$element(0," ",f$logical("notes$timezone"))/-100)'
```

For information on modifying these timezone logical names and on managing the timezone rules, see Section 4.4.1.

4.4.3 How to troubleshoot TDF problems on OpenVMS?

This is an OpenVMS Alpha system prior to V7.0 and the startup is not invoking the procedure:

```
SYS$MANAGER:UTC$TIME_SETUP.COM
```

This is an OpenVMS system prior to V6.0, where there is no OpenVMS TDF nor UTC available.

The version of the application does not use the OpenVMS TDF. This includes TCP/IP Services prior to V5.0, applications using HP C built on or targeting OpenVMS prior to V7.0, and systems using the DECnet-Plus DTSS mechanisms prior to the release associated with OpenVMS V7.3. (DCE DTS TDF management details to be determined.)

If you should find either of the following two timezone-related database files located in SYS$SPECIFIC:[SYSEXE]:

- SYS$SPECIFIC:[SYSEXE]SYS$TIMEZONE.DAT
- SYS$SPECIFIC:[SYSEXE]SYS$TIMEZONE_SRC.DAT

These two files are in an erroneous location and must be recreated in the correct directory:

```
SYS$COMMON:[SYSEXE]
```

If the DCL command:

```
$ DIRECTORY SYS$SYSTEM:SYS$TIMEZONE*.DAT
```

shows these files in SYS$SPECIFIC:[SYSEXE], then delete them and use SYS$MANAGER:UTC$TIME_SETUP.COM to recreate them.

On OpenVMS versions prior to V7.3, if the file:

```
$ SYS$STARTUP:DTSS$UTC_STARTUP.COM
```

is present on your system, then you may need to invoke:

```
$ @SYS$UPDATE:DTSS$INSTALL_TIMEZONE_RULE.COM
```

to recreate the timezone files correctly. Invoke this command immediately after [re]executing SYS$MANAGER:UTC$TIME_SETUP.COM.)

If SYS$UPDATE:DTSS$INSTALL_TIMEZONE_RULE.COM is not present on your system, then you may need to execute the following commands:

```
$ DELETE SYS$STARTUP:DTSS$UTC_STARTUP.COM
$ DEASSIGN/SYSTEM/EXEC SYS$TIMEZONE_RULE.
```

If your system time is being reported as being off by one hour (or whatever the local DST change), please see sections Section 4.7, Section 4.4 and Section 10.22.1.

4.5 Why does the SET TIME command fail? Help managing DTSS?

If you try to set the system time with the SET TIME command, and see one of the following messages:

```
%SET-E-NOTSET, error modifying time
-SYSTEM-F-IVSSRQ, invalid system service request
```

```
%SET-E-NOTSET, error modifying time
-SYSTEM-E-TIMENOTSET, time service enabled;
  enter a time service command to update the time
```

This occurs if the time on the local system is controlled by a time service software, for example the distributed time service software (DTSS) provided as part of the DECnet-Plus installation. The DTSS software communicates with one or more time servers to obtain the current time. It entirely controls the local system time (for DECnet-Plus, there is a process named DTSS$CLERK for this); therefore, the usage of the SET TIME command (and the underlying $SETTIM system service) is disabled.

The first message is displayed on systems running DECnet-Plus V6.1 and earlier. On systems with newer DECnet-Plus software, the second (and more informative) message is given.

You shouldn't have to change the time manually - you should be doing this through the time server - but if you insist... you'll have to shutdown DTSS:

```
$ RUN SYS$SYSTEM:NCL
DISABLE DTSS
DELETE DTSS
```

This will shutdown DTSS$CLERK. You may then change the system time as usual. To restart the DTSS software, type

```
$ @SYS$STARTUP:DTSS$STARTUP
```

You will need a number of privileges to issue this command, and you must also be granted the NET$MANAGE identifer to shutdown and to restart DTSS.

If you wish to ‘‘permanently’’ disable DTSS on a system running DECnet-Plus, the above NCL sequence must be performed each time the system is bootstrapped. (On DECnet-Plus V7.3 and later, you can define the logical name NET$DISABLE_DTSS to disable the DTSS startup. This logical name must be defined in the command procedure SYLOGICALS.COM, as this logical name must be present and defined sufficiently early in the OpenVMS system bootstrap sequence for it to function.)

If DTSS is running and no time servers are configured, you can (and will) see the following messages at regular intervals:

```
%%%%%%%%%%%  OPCOM   2-SEP-1999 19:41:20.29  %%%%%%%%%%%
Message from user SYSTEM on UNHEDI
Event: Too Few Servers Detected from: Node LOCAL:.mynode DTSS,
        at: 1999-09-02-19:41:20.296-04:00Iinf
        Number Detected=0,
        Number Required=1
        eventUid   5FA70F4F-616E-11D3-A80E-08002BBEDB0F
        entityUid  DE9E97DE-6135-11D3-8004-AA000400BD1B
        streamUid  D6513A46-6135-11D3-8003-AA000400BD1B
```

You can either configure the appropriate number of time servers, or you can disable DTSS, or you can ignore it and (if OPCOM is set to write to the log via via the logical names in SYLOGICALS.COM/SYLOGICALS.TEMPLATE) clean out OPERATOR.LOG regularly.

You can also simply disable the display of these messages:

```
$ run sys$system:ncl
block event dispatcher outbound stream local_stream -
    global filter -
    ((Node, DTSS), Too Few Servers Detected)
```

If you wish to disable the automatic TDF adjustment for the daylight time switch-over (on OpenVMS versions prior to V7.3), you can use the command:

```
$ run sys$system:ncl
set dtss automatic TDF change = false
```

or alternatively, you can set the local timezone to one that does not include the automatic daylight time change-over.

OpenVMS V7.3 and later simplify time and timezone management.

4.6 Setting time on AlphaServer ES47, ES80, GS1280 console?

To set the base system time on an member of the AlphaServer ES47, AlphaServer ES80 or AlphaServer GS1280 series system family, you must access the Platform Management Utility (PMU). The PMU is implemented within this family of related AlphaServer systems, and is part of a layer providing services beyond those of the traditional Alpha SRM console layer, and within a layer architecturally implemented beneath the SRM console. In particular, the PMU and related management components are used to provide services across multiple vPars or nPars partitions. In particular, the SRM obtains and manages the local system time on these systems as a delta time offset from the underlying base system time. Neither the SRM console nor OpenVMS directly accesses nor alters the underlying base system time nor other information maintained within the PMU layer.

The PMU uses the System Management components, centrally including the Backplane Manager (MBM) module found in each drawer, user interface, PCI and CPU management components, and the interconnections among these provided by the private system management LAN. When the system has power applied and the main breakers are on, the MBMs are active.

The PMU offers a command line interface for a serial communications or telnet connection and allows command and control of the MBM, and of the server. The PMU and the MBM system management components are responsible for the following tasks:

- Show the system configuration and provide the basic debugging capability
- Initiate the firmware update or load the test firmware version
- Power on or off, halt, or reset the system or partition
- The system partitioning and cabling functions
- Displays of the health of hardware environment, including such constructs as fans, power supplies and environmental and temperature values.
- Remote server management tasks
- The connection to the virtual SRM console
- Set and show the base system time.

You can use the MBM commands SHOW TIME and SET TIME to view and to manipulate the base system time. The delta time value for the primary MBM will be indicated, and it is this value in conjunction with the base time that is used to generate the time available to OpenVMS via the SRM console. If you issue a SET TIME=time command from OpenVMS, the delta time will change, but not the MBM base system time. If you change the MBM base system time, the calculated time available to OpenVMS via the SRM console(s) will change. (Resetting the base time thus involves changing the base system time, and then issuing SET TIME=time command(s) to each of the OpenVMS vPars or nPars environments to adjust the respective delta time values.) Rebooting, resetting or issuing an MBM SET TIME will reset the system time.

Typically, you will want to establish the MBM time value once, and probably setting it to UTC or such, and you will then want to boot each partition conversationally, setting the SETTIME system parameter to force the entry of the time within each booting system environment. Once the MBM time value has been set once, you will typically not want to alter it again. You will typically want to manage and modify only the time values within each partition.

The time and data values stored in the primary MBM and replicated in the zero or more secondary MBMs that might be present within the system are coordinated.

To enter the PMU from the SRM console, and to exit back to SRM:

```
MBM - (PMU, Platform Management Utility)

  From SRM P00> enter {Esc} {Esc} MBM
  CTRL/[ CTRL/[ MBM           (MBM must be uppercase)
  MBM> connect                (to exit to SRM)
```

The CTRL/[is the escape character. Use the cited key sequences to enter the PMU. You can also access the PMU through a modem, or from a terminal or terminal emulator or terminal server connected to the server management LAN. Having the server management LAN bridged to an untrusted LAN can be unwise, however, and with risks analogous to those of configuring a traditional VAX or Alpha console serial line to an open terminal server or to a dial-in modem.

See the AlphaServer GS1280 documentation for additional information.

4.7 UTC vs GMT vs vs UT1/UT1/UT2 TDF? What are these acronyms?

The results of an international compromise—though some would say an international attempt to increase confusion—UTC is refered to as ‘‘Coordinated Universal Time’’ (though not as CUT) in English and as ‘‘Temps Universel Coordinné’’ (though not as TUC) in French. (No particular information exists to explain why UTC was chosen over the equally nonsensical TCU, according to Ulysses T. Clockmeister, one of the diplomats that helped establish the international compromise.)

Universal Time UT0 is solar time, UT1 is solar time corrected for a wobble in the Earth’s orbit, and UT2 is UT1 corrected for seasonal rotational variations in rotation due to the Earth’s solar orbit.

GMT—Greenwich Mean Time—is UT1. GMT is the time at the classic site of the since-disbanded Royal Greenwich Observatory; at the most widely-known tourist attraction of Greenwich, England.

UTC is based on an average across multiple atomic clocks, and is kept within 0.9 seconds of GMT, through the insertion (or removal) of seconds. In other words, UTC matches GMT plus or minus up to 0.9 seconds, but UTC is not GMT.

TDF is the Timezone Differential Factor, the interval of time between the local time and UTC. Areas that celebrate daylight saving time (DST) will see periodic changes to the TDF value, when the switch-over between daylight time and standard time occurs. The switch-over itself is entirely left to local governmental folks, and can and has varied by political entity and politics, and the switch-over has varied over the years even at the same location.

If your local OpenVMS system time is off by one hour (or whatever the local DST change) for some or all applications, you probably need to reset your local TDF. (For related details, please see sections Section 4.4 and Section 10.22.1.)

Further discussions of history and politics, the Royal Observers’ outbuildings, and the compromise that left the English with the Time Standard (the Prime Meridian) and the French with the standards for Distance and Weight (the Metric System) are left to other sources. Some of these other sources include the following URLs:

- ftp://elsie.nci.nih.gov/pub/
- http://physics.nist.gov/GenInt/Time/time.html
- http://nist.time.gov/

4.8 Using w32time or an SNTP as a time provider?

No standards-compliant NTP or SNTP server is reportedly capable of synchronizing with the Microsoft Windows w32time services.

Further, NTP clients are not generally capable of synchronizing with an SNTP server.

Open Source (Free) NTP servers (qv: OpenNTP) are available for Microsoft Windows platforms, and TCP/IP Services and third-party packages all provide NTP servers for OpenVMS, and NTP and SNTP clients can synchronize with these srvers.

5 System Management Information

If you are searching for something here,
please consider using the text-format FAQ.

5.1 What is an installed image?

The term ‘‘install’’ has two distinct meanings in OpenVMS. The first relates to ‘‘installing a product’’, which is done with either the SYS$UPDATE:VMSINSTAL.COM command procedure or the POLYCENTER Software Installation (PCSI) utility (PRODUCT command). The second meaning relates to the use of the INSTALL utility, which is what concerns us here.

The INSTALL utility is used to identify to OpenVMS a specific copy of an image, either executable or shareable, which is to be given some set of enhanced properties. For example, when you issue the SET PASSWORD command, the image SYS$SYSTEM:SETP0.EXE is run. That image needs to have elevated privileges to perform its function.

The other important attribute is /SHARED. This means that shareable parts of the image (typically read-only code and data) are loaded into memory only once and are shared among all users on a system. Executable images can be installed /SHARED as well as shareable library images. (The term ‘‘shareable’’ has dual meanings here, too. See the *OpenVMS Programming Concepts Manual* for further details.)

It’s important to note that there is no such thing as ‘‘installing a shareable image with privileges’’. The INSTALL utility will let you do it, but the privileges you specify will be ignored. To have a callable routine run with enhanced privileges that are not available to its caller, you must construct your routines as ‘‘user-written system services’’ (UWSS) and install the shareable image with the /PROTECT qualifier. See the *OpenVMS Programming Concepts Manual* for more information on user-written system services. Note also that in many cases the need to grant privileges to an image can be replaced with the use of the ‘‘Protected Subsystems’’ feature that grants a rights identifier to an image. See the *OpenVMS Guide to System Security* for information on Protected Subsystems.

5.2 Are there any known viruses for OpenVMS?

Viruses and worms are common on personal computers because the operating systems involved, such as the Microsoft MS-DOS, Windows 95, Windows 98 and Windows ME variants, do not particularly protect the operating system or the file system against hostile action by programs. Microsoft Windows NT, Windows 2000 and Windows XP do implement protections for specific configurations and do implement memory protection models, but many users of these systems choose to operate with full adminstrator access and thus the available protections are entirely defeated and entirely not relevent, and any program that can activate itself or can cause the user to activate the code can subvert the operating system and take over the hardware, at which point the malicious code can do most anything it wishes, including hiding copies of itself in other programs or in the file system, redistributing itself via mail, IM, or network connections, or can be used as a zombie in staging attacks on other systems.

This is less likely with multi-user systems such as OpenVMS, Unix, Linux, MVS and other platforms for various reasons. First, the operating system runs in a privileged mode in memory that is protected against modification by normal user programs. Any program cannot simply take over the hardware as it can on operating systems without security and particularly without memory page protections. Secondly, multi-user systems can be set up so that non-privileged programs cannot modify system programs and files on disk, and this is normal for most installations. Both of these protection schemes mean that traditional viral infections don’t work on these OSes. Third, typical applications and configurations tend to prevent the uncontrolled execution of untrusted code as part of received mail messages or web access; one of the central vulnerabilities of the Microsoft Windows platform involves its intentionally easy ability to dynamically (and transparently) activate code and macros that are embedded within mail messages and within data files.

It is possible for OpenVMS and other multi-user systems to become infected by viruses or worms, but to do so, the program containing the virus must be run from a user account that has amplified privileges. So long as the system administrator is careful that only trusted applications are run from such accounts (and this is generally the case) and so long as there are no OpenVMS system security breaches (due to malicious operator activity, OpenVMS errors, or errors within trusted and privileged product packages) there is no of modifications to the operating system or other protected files from the virus or the worm.

The FAQ maintainer is aware of a few (and very old) DECnet worms that have affected OpenVMS systems on DECnet networks (‘‘WANK’’ was one), but is aware of no OpenVMS viruses that are loose in the field.

To protect against viruses and other attempts at system interference or misuse, please follow the security recommendations in the *OpenVMS Guide to System Security*. Additionally, you will want to keep your OpenVMS ECOs current and you will want to apply all mandatory ECO kits and any security MUPs for OpenVMS and OpenVMS products, and you will want to keep to OpenVMS releases with Prior Version Support (PVS) or with Current Version Support. (This is obviously a general system maintenance recommendation, in addition to being a good system security recommendation—new security features and capabilities are implemented in more recent OpenVMS releases, for instance. Details on PVS releases are available over in Section 5.10.6.) You may also want to consider optional software products which can monitor your system for intrusion or infection attempts. Computer Associates (CA) offers various products in this area, as to other vendors.

Rocksoft offers the Veracity data integrity tool (for info, send mail to demo@rocksoft.com). MD5 tools are also available; see Section 5.30 for details on MD5.

Tools to scan OpenVMS file systems for Microsoft Windows infections are and have been available, including a commercial package from Sophos , and a port of the open source Clam Antivirus scanner at http://www.clamav.net/ and with an OpenVMS port at http://fafner.dyndns.org/~alexey/clamav.zip.

These scanning tools are particularly useful for systems running Samba or Advanced Server (PATHWORKS), as these servers tend to have a higher population of files intended for Microsoft Windows systems users, and as common virus and worm attacks can find and infect files on the file shares that these products can provide. These infections do not target OpenVMS itself, though the OpenVMS server (and any other platform and any other server capable of storing files for Windows systems) can silently host files containing common Microsoft Windows infections.

5.3 Sources of OpenVMS security information?

Where can I get information on OpenVMS system security?

- http://www.hp.com/go/openvms/doc
- http://www.blacksheepnetworks.com/security/resources/openvms/

5.4 How do I mount an ISO-9660 CD on OpenVMS?

ISO-9660 support was added in the following releases:

- OpenVMS VAX V6.0
- OpenVMS AXP V1.5

An add-on ISO-9660 kit was also available for OpenVMS VAX V5.5, V5.5-1, V5.5-2, and V5.5-2H4. This requires the installation of the F11CD kit from the InfoServer CD, from the Consolidated Distribution CD under the InfoServer area, or the F11CD ECO kit. (Upgrades to V6 and later are strongly recommended.)

By default, OpenVMS senses the specific type of media. If you are working with dual-format media—media that uses both the ODS-2 and ISO-9660 formats on the same CD-ROM—then MOUNT will first detect and then default to the ODS-2 format. If you wish to override this and explicitly mount the media using ISO-9660, use the command:

```
$ MOUNT/MEDIA_FORMAT=CDROM  device-name[:] [volume-label]
```

In most circumstances, you will not need nor will you want to include an explicit /MEDIA_FORMAT specification. For further information, please refer to the OpenVMS MOUNT Utility Manual. Particularly note the information on the MOUNT /MEDIA_FORMAT and /UNDEFINED_FAT qualifiers.

The MOUNT /UNDEFINED_FAT qualifier is of interest because ISO-9660 media can be mastered on a wide variety of operating system platforms, and these platforms do not necessarily support the semantics needed for files containing predefined record formats. The /UNDEFINED_FAT allows you to specify the default attributes for files accessed from volumes using the ISO-9660 format.

An example which works for most CD-ROMs is:

```
$ MOUNT/MEDIA_FORMAT=CDROM/UNDEFINED_FAT=STREAM:2048 DUA0: FREEWARE
```

This particular MOUNT command forces access to the CD-ROM media using the ISO-9660 volume structure, and the use of the MOUNT /UNDEFINED_FAT qualifier causes any file whose file attributes are ‘‘undefined’’ to be returned with ‘‘stream’’ attributes with a maximum record length 2048.

On OpenVMS, the ISO-9660 format is (internally) considered to be the ODS-3 file structure, while the High Sierra extensions to the standard are considered to be the ODS-4 file structure. The Rock Ridge extensions are not currently available on OpenVMS.

For details on ODS-1 and ODS-2 file specifications, see Kirby McCoy's *VMS File System Internals Manual* (published by Digital Press, but potentially out of print), and see:

- http://pdp-11.trailing-edge.com/www/ods1.txt
- Look for the Freeware V5.0 directory ODS2 at http://www.hp.com/go/openvms/freeware/

5.5 How do I extract the contents of a PCSI kit?

A growing number of OpenVMS products are being provided in PCSI (POLYCENTER Software Installation) kits which are installed using the PRODUCT INSTALL command. These are alternatives to or replacement for VMSINSTAL kits which were BACKUP savesets. PCSI kits are not BACKUP savesets and are structured differently from VMSINSTAL kits.

If you want to extract product files from a PCSI kit, create a directory into which the kit should be expanded and use the following command:

```
$ PRODUCT COPY prodname /SOURCE=[where-the-kit-is] -
    /DEST=[destination-directory] /FORMAT=REFERENCE
```

A PCSI kit file has a file specification of the following form:

```
DEC-VAXVMS-FORTRAN-V0603-141-1.PCSI
```

In this example, ''FORTRAN'' is the ''prodname''. PCSI will expand the kit files into the directory you specify and subdirectories beneath such as [SYSEXE], [SYSLIB], etc., reflecting the eventual destination of files found there. Most of the actual product files (images, etc.) will be in the subdirectories. In the top-level directory will be a file with the file type PCSI$DESCRIPTION that specifies where various files should go. For more details, see the *POLYCENTER Software Installation Developer's Guide for OpenVMS*, which can be found in the OpenVMS documentation on the Consolidated Online Documentation CD-ROM.

5.6 Emergency (Conversational) System Startup?

If you need to perform system management operations on an OpenVMS system and cannot access the system through normal means—the password on the SYSTEM username was forgetten and no other privileged usernames are available, or one or more core system product authorization key (PAK) software licenses are unavailable or expired—then you must perform a conversational (emergency) bootstrap.

Here are the steps:

1. Halt the system. Exactly how this is done depends on the specific system model: Depending on the model, this can involve pressing the HALT button, entering CTRL/P on the console, or pressing the BREAK key on the console.
2. At the console prompt, use a console command to boot into the SYSBOOT utility. (SYSBOOT allows conversational changes to system parameters.) (The console syntax for the conversational bootstrap varies by system model and by system architecture—this typically involves specifying a flag with the lowest bit set. See Section 14.3.5 for related details.) For example:

   On VAX, use one of the following three commands depending on the particular model of VAX system involved:

   ```
   B/R5:1
   B/1
   @GENBOO
   ```

   On Alpha:

   ```
   b -flags 0,1
   ```

   If your system has a non-zero system root (such as root SYSE, shown here), you will have to use a console command such as the following:

   On VAX:

   ```
   B/E0000001
   B/R5:E0000001
   @<console media procedure name varies widely>
   ```

   On Alpha:

   ```
   b -flags e,1
   ```

On the IA-64 architecture systems, you can establish and manage an EFI boot alias for a conversational bootstrap as discussed in Section 14.3.5.1 and in Section 14.3.10, or you can use VMS_LOADER.EFI interactively as shown here. Of the core mechanisms discussed in Section 14.3.5.1, the following uses an EFI Shell command to perform a conversational bootstrap of root SYSE via the partition device fsn:. There are alternative mechanisms available.

```
fsn:\efi\vms\vms_loader.efi -flags e,1
```

If your Alpha system has a hardware password (various systems support a password that prevents unauthorized access to the console), you will need to know theis password and will need to enter it using the LOGIN or similar command at the console. If you get an ‘‘Inv Cmd’’ error trying to perform a conversational bootstrap, and you do not have the hardware console password for the console LOGIN command, you are stuck—you will need to call for hardware service for assistance in resetting the hardware console password. The implementation and the syntax used for the console password mechanism does vary by implementation.

3 Once at the SYSBOOT prompt, request that OpenVMS read the system startup commands directly from the system console, that the window system (if any) not be started, and that OpenVMS not record these particular parameter changes for subsequent system reboots:

```
SET/STARTUP OPA0:
SET WINDOW_SYSTEM 0
SET WRITESYSPARAMS 0
CONTINUE
```

4 At the $ prompt, the system will now be accepting startup commands directly from the console. Type the following two DCL commands:

```
$ SPAWN
$ @SYS$SYSTEM:STARTUP
```

5 You should now see the dollar ($) prompt of DCL.

The result of these two commands will be the normal system startup, but you will be left logged in on the console, running under a fully privileged username. Without the use of the SPAWN command, you would be logged out when the startup completes.

Perform the task(s) required, such as resetting the password on the SYSTEM username as described in Section 5.6.1 or registering one or more license product authorization keys (PAKs) as described in Section 5.6.2.

6 Once you log out of this session, the system will complete the startup and can be used normally. You can choose to reboot the system, but that is not necessary.

Some system managers will suggest a method using the UAFALTERNATE system parameter rather than the SET/STARTUP OPA0: command shown. This approach is not always available and is accordingly less commonly recommended, as there can easily be an alternate user authorization database (SYS$SYSTEM:SYSUAFALT.DAT) configured on the system. With a system manager that has configured an alternate SYSUAFALT.DAT file, the UAFALTERNATE method will fail—well, assuming you do not know the password of a privileged username stored within SYSUAFALT.DAT, of course.

The UAFALTERNATE system parameter is used to trigger what is sometimes known as the console backdoor. The OPA0: system console is critical to system operations and system security, and will allow access when the SYSUAF system authorization database is unavailable or corrupted, when core product license PAKs are not registered, expired or disabled (NOLICENSE errors), or in various other cases of system failures. All this is in addition to the role of the console in the display of certain system-critical event messages. *Access to the OPA0: console has a security exposure that is equivalent to direct access to the system hardware.*

When LOGINOUT detects an error (such as a SYSUAF corruption, by a missing SYSUAF, missing product licenses, or other trigger), it will prevent access to the OpenVMS system from all terminals except the system console. The OPA0: system console will be allowed access, and the resulting process will be fully privileged. Resetting the UAFALTERNATE system parameter—in the absence of an alternate SYSUAF system authorization database—will cause the console backdoor to be opened simply because LOGINOUT cannot locate SYS$SYSTEM:SYSUAFALT.DAT. When the authorization database cannot be located, access will be granted from the console only.

For further information on emergency startup and shutdown, as well as for the official OpenVMS documentation on how to change the SYSTEM password from the console in an emergency, please see the *OpenVMS System Manager's Manual* in the OpenVMS documentation set.

For information and recommendations on setting up OpenVMS system security, please see the NCSC Class C2 appendix of the Guide to OpenVMS System Security manual, also in the OpenVMS documentation set.

You can also use the conversational bootstrap technique shown earlier (the steps until SET/STARTUP) to alter various system parameters, as well. At the SYSBOOT prompt, you can enter new parameters values:

```
SHOW MAXPROCESSCNT
SET . 64
CONTINUE
```

The ⌈.⌋ is a shorthand notation used for the last parameter examined within SYSGEN and SYSBOOT.

5.6.1 I've forgotten the SYSTEM password - what can I do?

If you have forgotten or do not have the password for the SYSTEM username, you must perform the conversational bootstrap as described in Section 5.6, and must enter the following commands once you have reached the dollar ($) prompt:

```
$ SET DEFAULT SYS$SYSTEM:  ! or wherever your SYSUAF.DAT resides
$ RUN SYS$SYSTEM:AUTHORIZE
MODIFY SYSTEM /PASSWORD=newpassword
EXIT
```

You have now reset the password on the SYSTEM username.

5.6.2 My product licenses have expired - what can I do?

If you have a system with no licenses for OpenVMS or for OpenVMS users and thus cannot log into the OpenVMS system normally, you should be able to log into the console serial terminal—this is the terminal device known as OPA0:—and perform the commands necessary.

For systems that are not configured with an accessable console serial terminal—as can be the case with how some DECwindows workstations are configured—you must log in over the network or from a local serial connection. If you cannot log in over a network connection (SET HOST, telnet, etc) or from another local serial terminal connection, you will have to halt the system and perform a conversational bootstrap as described in Section 5.6. You must then enter licensing-related commands once the conversational bootstrap has reached the dollar ($) prompt.

Use the following DCL command to invoke a menu that allows you to manage and to register new or replacement license PAKs:

```
$ @SYS$UPDATE:VMSLICENSE
```

You have now registered the license PAKs. Direct use of the DCL commands LICENSE and SHOW LICENSE and such is also obviously available.

If you wish to connect a serial console on your DECwindows workstation, please see Section 14.3.3.3, Section 14.3.6, Section 11.10, and Section 14.17.

For information on troubleshooting DECwindows, please see Section 11.5.

5.7 How do I change the node name of an OpenVMS System?

The first step is to get a BACKUP of the system disk before making any changes—use the system disk backup procedures as documented in the OpenVMS System Management Manual, making sure to use the procedures and commands appropriate for the system disk.

Changing the node name involves a number of steps—the node name tends to be imbedded in a number of different data files around the system.

- Update the SCSNODE in MODPARAMS.DAT, and then run AUTOGEN as far as the SETPARAMS phase. (Do not reboot yet.)
- Modify the DECnet node name. (NETCONFIG is the DECnet Phase IV tool, and NET$CONFIGURE is the DECnet-Plus tool.)
- Modify the host node name on the various queues in the queue database. (each queue has a host name, and it defaults to the SCS node name of the queue's host system. See the command INIT/QUEUE/ON=node for information.) Site-specific startup command procedures can explicitly specify the (local or even the current) node on the /ON parameter in an INIT/QUEUE/START/ON= command.
- Modify the node name saved in any application databases, or any local node-conditional operations present in the site-specific system startup, etc. (SEARCH for the node name, specifying all types of files.)
- Use the AUTHORIZE utility command RENAME/IDENTIFIER to rename the SYS$NODE_oldnodename rightslist identifier to match the new node name. (Do not change the binary value of this identifier, and do not delete the identifier.)

If you have erroneously deleted or duplicate the identifier, you can locate existing references to the binary identifier value using the Freeware DFU package, and specifically the commands SEARCH/ACE and /OWNER. You must (re)create the correctly-named identifier using the binary value that is often stored in various Access Control List Entry (ACE) structures and object owner fields associated with files and objects present in the OpenVMS system.

- Reset any license PAKs that are restricted to the old node name to the new node name.
- If the node name is part of a disk volume label, see Section 5.13.
- Reboot the node or—if in a VMScluster—reboot the whole VMScluster. (This tends to catch any errors immediately.)
- Modify the IP node name. (The TCP/IP Services tool is UCX$CONFIG prior to V5.0, and is TCPIP$CONFIG in V5.0 and later releases.) Note that TCP/IP Services ties the IP host name to the current SCSNODE value within its UCX$CONFIGURATION.DAT or TCPIP$CONFIGURATION.DAT database. Thus if SCSNODE is changed, the IP host name reconfiguration must occur, and the required reconfiguration can occur only after a system reboot. Accordingly, it is best to perform the TCP/IP Services host name reconfiguration step after the reboot.

There are likely a few other areas where the nodename will be stored. Local procedures and data files are one such example, and various sites will have the system name loaded in the operator control panel via the OCP_TEXT console environment variable available at the SRM prompt on some Alpha systems is another.

If the system is configured in a VMScluster and you change *either* the SCSNODE or the SCSSYSTEMID—but *not* both values—then you will have to reboot the entire VMScluster. (The VMScluster remembers the mapping between these two values, and will assume that a configuration problem has occured if a mismatched pair appears, and will refuse to let a node with a mismatched pair join the VMScluster.)

To calculate the correct SCSSYSTEMID value, multiply the DECnet Phase IV area number by 1024, and add the DECnet Phase IV node number. For example, the SCSSYSTEMID value for a DECnet node with address 19.22 is 19478. ((19 * 1024) + 22 = 19478)

This may well have missed one or two configuration tools (or more!) that are needed at your site—the node name tends to get stored all over the place, in layered products, and in local software...

Also see Section 15.6.3 and Section 15.6.4.

5.8 Why doesn’t OpenVMS see the new memory I just added?

When adding memory to an OpenVMS system, you should check for an existing definition of the PHYSICALPAGES (OpenVMS VAX) or PHYSICAL_MEMORY (OpenVMS Alpha) parameter in the SYS$SYSTEM:MODPARAMS.DAT parameter database, use a text editor to reset the value in the file to the new correct value as required, and then perform the following command:

```
$ @SYS$UPDATE:AUTOGEN GETDATA REBOOT FEEDBACK
```

This AUTOGEN command will reset various system parameters based on recent system usage (FEEDBACK), and it will reset the value for the PHYSICALPAGES parameter to the new value. It will also reboot the OpenVMS system.

PHYSICALPAGES and PHYSICAL_MEMORY can also be used to deliberately lower the amount of memory available for use by OpenVMS. This ability can be useful in a few specific circumstances, such as testing the behaviour of an application in a system environment with a particular (lower) amount of system memory available.

PHYSICALPAGES and PHYSICAL_MEMORY can be set to -1 (on OpenVMS Alpha) or (better and simpler) the entry can be removed from the MODPARAMS.DAT file, to indicate that all available memory should be used.

5.9 How do I change the text in a user’s UIC identifier?

The text translations of the numeric User Identification Code (UIC) are based on identifiers present in the OpenVMS rightslist. Documentation on this area is included in the _Guide to OpenVMS System Security_ manual.

To control the identifiers shown for a user’s UIC, you use AUTHORIZE. Each user has an associated group identifier, and an identifier specific to the user. And each user should have a unique UIC.

To alter the text of a user or group identifier, use commands such as:

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> rename/ident oldgroupid newgroupid
UAF> rename/ident olduserid  newuserid
```

If you should find yourself missing an identifier for a particular user, you can add one for the user's UIC using a command such as:

```
UAF> add/ident/value=uic=[group,user] newuserid
```

The UIC user identifier text is assigned when the username is created, and is the text of the username. The UIC group group identifier is assigned when the first username is created in the UIC group, and the text is based on the account name specified for the first user created in the group. The value of this identifier is [groupnumber, 177777]. To add a missing group identifier, use an asterisk as follows:

```
UAF> add/ident/value=uic=[group,*] newgroupid
```

You may find cases where an identifier is missing from time to time, as there are cases where the creation of a UIC group name identifier might conflict with an existing username, or a user identifier might conflict with an existing group identifier. When these conflicts arise, the AUTHORIZE utility will not create the conflicting group and/or user identifier when the username is created.

You can can add and remove user-specified identifiers, but you should avoid changing the numeric values associated with any existing identifiers. You should also avoid reusing UICs or identifiers when you add new users, as any existing identifiers that might be present on objects in the system from the old user will grant the same access to the new user. Please see the security manual for details.

5.10 What are the OpenVMS version upgrade paths?

5.10.1 OpenVMS Alpha Upgrade (or Update) Paths

Note

Upgrade path information here has occasionally been found to be wrong. Information here does not reflect cluster rolling upgrade requirements; see Section 5.10.4 for related rolling upgrade information; versions permissible for rolling upgrades can be and often are more constrained. When upgrade information here conflicts with the official documentation, please assume that the information here is wrong. Corrections and updates to this material are welcome.

```
From V1.0,
    you can upgrade to V1.5.
From V1.5, or V1.5-1H1,
    you can upgrade to V6.1.
From V6.1,
    you can upgrade to V6.2.
From V6.1, or V6.2,
    you can upgrade to V7.0.
From V6.1, V6.2, V6.2-1H(1,2,3), or V7.0,
    you can upgrade to V7.1.
From V6.2,
    you can update to V6.2-1H1, V6.2-1H2, or V6.2-1H3.
From V6.2, V6.2-1H(1,2,3), V7.1, V7.1-1H(1,2), or V7.2,
    to V7.2-1.
From V6.2, ... or V7.2,
    to V7.2-1H1, to 7.3.
From V7.1, you can update to V7.1-1H(1,2), ...
    to V7.2-1H1, to 7.3.
From 7.2, 7.2-1, 7.2-1H1, 7.2-2, 7.3 or 7.3-1,
    you can upgrade to V7.3-2
From V7.3, V7.2-2, V7.2-1H1, V7.2-1, and V7.1-2,
    you can upgrade to V7.3-1
From V7.3-1,
    you can upgrade to V7.3-2 or to V8.2.
From V7.3-1 or V7.3-2,
    you can upgrade to V8.2.
From V7.3-2 or V8.2,
    you can upgrade to V8.3.
```

Some typical OpenVMS Alpha upgrade (or update) paths are:

```
V1.0 -> V1.5 -> V6.1 -> (V6.2, V7.0, V7.1, V7.2, V7.3)
V1.5-1H1 -> V6.1 -> (V6.2, V7.0, V7.1, V7.2, V7.3)
V6.2 -> V6.2-1H3
V6.2 -> V7.2-1
V6.2 -> V7.3
V6.2-1H(1,2,3) -> V7.1
V6.2-1H(1,2,3) -> V7.2-1
V6.2 through 7.1-1H2 inclusive -> V7.3
V7.1 -> V7.1-2
V7.1 -> V7.2-1
V7.1-1H(1,2) -> V7.1-2
V7.1-1H(1,2) -> V7.2-1
V7.1-2 -> V7.3-1
V7.2 -> V7.2-1H1
V7.2 -> V7.3 -> V7.3-1
V7.2-1 -> (V7.3, V7.3-1)
V7.2-2 -> (V7.3, V7.3-1, V7.3-2)
V7.3 -> (V7.3-1, V7.3-2)
V7.3-1 -> (V7.3-2, V8.2)
V7.3-2 -> (V8.2, V8.3)
```

Note that OpenVMS Alpha V7.0 does not include support for hardware and/or configurations first supported in OpenVMS Alpha V6.2-1H1, V6.2-1H2, or V6.2-1H3; one must upgrade to OpenVMS Alpha V7.1, or later.

One cannot update directly to a V6.2-1Hx Limited Hardware Release (LHR) from any release prior to the baseline V6.2 release. The same prohibition holds for performing updates directly to V7.1-1Hx from any release prior to V7.1—this is not supported, and does not produce the expected results. The LHR kits can, however, be directly booted and can be directly installed, without regard to any operating system that might be present on the target disk.

Users of OpenVMS Alpha V7.1-1H1, V7.1-1H2, V7.2-1H1 or other hardware are encouraged to upgrade to the next available non-hardware-release, and should preferably upgrade to the current or to a supported OpenVMS Alpha release.

OpenVMS Alpha updates for LHRs (through V7.1-1Hx) require the use of VMSINSTAL for the update. These LHR releases use PCSI for the installation, but not for the update. Non-LHR releases use PCSI for installs and upgrades.

OpenVMS Alpha V7.1-2 and later use PCSI for LHRs and for OpenVMS upgrades and for all OpenVMS ECO kit installations; V7.1-2 and later use upgrades and not updates. VMSINSTAL OpenVMS ECO kits (updates) are not used on OpenVMS Alpha V7.1-2 and later; prior to V7.1-2, VMSINSTAL-based ECO (update) kits are used for OpenVMS.

5.10.2 OpenVMS I64 Upgrade Paths

Note

Upgrade path information here has occasionally been found to be wrong. Information here does not reflect cluster rolling upgrade requirements; see Section 5.10.4 for related rolling upgrade information; versions permissible for rolling upgrades can be and often are more constrained. When upgrade information here conflicts with the official documentation, please assume that the information here is wrong. Corrections and updates to this material are welcome.

```
From V8.2,
    you can upgrade to V8.2-1 or V8.3
From V8.2-1,
    you can upgrade to V8.3
```

Some typical OpenVMS I64 upgrade (or update) paths are:

```
V8.2 -> V8.2-1 -> V8.3
```

OpenVMS I64 V8.2 is the first production release. OpenVMS I64 V8.0 and V8.1 were intended for early adopters of OpenVMS on Integrity servers, and are not considered to be production releases.

To utilize OpenVMS I64 V8.2, you must perform a full installation of V8.2. No supported upgrade path to V8.2 is available from previous releases; there is no upgrade from OpenVMS I64 E8.2, nor from the earlier V8.1 or V8.0 releases.

5.10.3 OpenVMS VAX Release Upgrade Paths

Note

Upgrade path information here has occasionally been found to be wrong. Information here does not reflect cluster rolling upgrade requirements; see Section 5.10.4 for related rolling upgrade information; versions permissible for rolling upgrades can be and often are more constrained. When upgrade information here conflicts with the official documentation, please assume that the information here is wrong. Corrections and updates to this material are welcome.

```
From V5.0 through V5.4-3 inclusive, one can upgrade to V5.5.
From V5.5, V5.5-1, or V5.5-2HW, one can upgrade to V5.5-2.
From V5.5, V5.5-1, or V5.5-2, one can upgrade to V6.0.
From V5.5-2, V5.5-2H4, or V6.0, one can upgrade to V6.1.
From V6.0, or V6.1, one can upgrade to V6.2.
From V6.1, or V6.2, one can upgrade to V7.0.
From V6.1, V6.2, or V7.0, one can upgrade to V7.1.
From V6.1, one can upgrade to V7.3 (with VAXBACK ECO for V6.1).
```

Some typical OpenVMS VAX upgrade paths are:

```
V5.x -> V5.5 -> V6.0 -> V6.2 -> (V7.1, V7.2, V7.3)
V5.5-2HW -> V5.5-2
V5.5-2, or V5.5-2H4 -> V6.1 -> (V6.2, V7.0, or V7.1)
V6.1 -> V6.1 with VAXBACK ECO -> (V7.2, V7.3)
V6.2 -> V7.2
V6.2 -> V7.3
```

Note that OpenVMS VAX V6.0 does not include support for hardware and/or configurations first added in OpenVMS VAX V5.5-2H4, one must upgrade to OpenVMS VAX V6.1.

Note that OpenVMS VAX V5.5-2HW is a pre-release version of V5.5-2. Any system running it should be upgraded to V5.5-2, or later.

If you attempt a direct upgrade from OpenVMS VAX V6.1 to V7.2 or later without having first applied the VAXBACK ECO kit to your V6.1 system, you will receive an error message:

```
%BACKUP-E-INVRECTYP, invalid record type in save set
```

and the upgrade will fail. Acquire and apply the VAXBACK ECO kit for OpenVMS VAX V6.1. OpenVMS VAX V6.2 and later do not require an application of an ECO for an upgrade to V7.2 and later.

5.10.4 OpenVMS Cluster Rolling Upgrade Paths

Rolling Upgrades permit the OpenVMS Cluster and the applications to remain available while individual systems are being upgraded to a new OpenVMS release.

Rolling Upgrades require multiple system disks.

OpenVMS Cluster Rolling Upgrades for OpenVMS Alpha, OpenVMS I64 and OpenVMS VAX may (will) have architecture-specific, or additional upgrade requirements or prerequisites, and have requirements around which versions and architectures of OpenVMS can coexist within a OpenVMS Cluster than what are listed here.

For specific details on Rolling Upgrades, please see the OpenVMS Upgrade and Installation Manual for the particular release, and the OpenVMS Software Product Descriptions for OpenVMS and for OpenVMS Cluster software:

- http://h18000.www1.hp.com/info/spd/

  OpenVMS typically uses SPD 25.01.xx, SPD 41.87.xx, and SPD 82.35.xx.

for further details on the Rolling Upgrade, and for support information.

5.10.5 OpenVMS VAX Manual Organization

The documentation for older releases of OpenVMS VAX was comprised of various platform-specific manuals, manuals that include instructions that are specific to installing and upgrading on the particular VAX platform. These older manuals can be useful for learning platform- or console-specific operations or requirements for the particular (and older) VAX platform.

There is far less console command syntax, and console storage media variability, among the more recent Alpha and Integrity processors. The newer platform operator and management interfaces are far more consistent across the platform lines.

5.10.6 OpenVMS Product Version and Support Information

For information on Prior Version Support (PVS) and Mature Product Support (including information on support end dates for OpenVMS and various layered products), please see the support resources link available at the main OpenVMS website or the services links available at the main services website:

- http://www.hp.com/go/openvms/
- http://www.hp.com/go/services

And see the following links, with the caveat that the direct ‘‘/hps’’ links shown here may become stale:

- http://www.hp.com/hps/os/os_pvs.html
- http://www.hp.com/hps/os/os_ovms.html

For information on the supported and required versions of layered products, and the minimum required layered product versions for various configurations, please see the Software Rollout Report (SWROLL), available at:

- http://h71000.www7.hp.com/openvms/os/swroll/

For additional related information, see Section 2.6.1.

For information on the release history of OpenVMS, including information on the code names of various releases and the major features:

- http://www.openvms.compaq.com/openvms/os/openvms-release-history.html

Additional release history information, as well as a variety of other trivia, is available in the VAX 20th anniversary book:

- http://www.openvms.compaq.com/openvms/20th/vmsbook.pdf

5.10.7 OpenVMS Alpha and I64 Upgrade Terminology

OpenVMS Alpha and OpenVMS I64 use the POLYCENTER Software Product Install Utility, occasionly refered to as SPIU and rather more commonly known as PCSI. PCSI is a component of the OpenVMS operating system, and is available on OpenVMS VAX, OpenVMS Alpha, and OpenVMS I64.

The following terms apply to OpenVMS Alpha and to OpenVMS I64 Upgrades and Installations using PCSI:

- UPDATE: Typically used for Limited Hardware Releases (LHR) releases. Performed via VMSINSTAL. Applies only to the OpenVMS release that the LHR is based on, or to an intermediate LHR. (eg: V7.1-1H2 applies only to V7.1-1H1 and to V7.1, not to any other releases.) LHRs within a series are cumulative, containing all files and features of previous LHRs in the same series.

  VMSINSTAL-based Updates and VMSINSTAL-based ECO kits are not generally used to upgrade OpenVMS on releases of OpenVMS Alpha V7.1-2 and later, nor are thse used on OpenVMS I64; only PCSI-based Upgrades and Installs are used. VMSINSTAL remains available for other uses and other products; for upgrades and installations of products other than OpenVMS itself.

- UPGRADE: Performed via PCSI. Upgrades can typically be applied directly to a release-specific range of earlier OpenVMS releases. The product release documentation specifies the prior OpenVMS releases; if your release is not one of the specified releases, you will have to perform one or more additional upgrades (through intermediate OpenVMS releases) to reach one of the prerequisite releases.

- INSTALL: Performed via PCSI. With an installation, no existing version of the operating system is assumed present, nor are any files from any copy of the operating system might be present preserved, and the entire contents of the target disk are destroyed via a disk initialization.

- PRESERVE: Performed via PCSI. Otherwise similar to an installation, this option skips the disk reinitialization. User files on the target disk are preserved. Any existing operating system files on the target disk are clobbered.
- LHR: Limited Hardware Release. LHRs are specific to and are targeted at new hardware configurations, and are not shipped to customers with support contracts. At least one LHR kit must be specifically acquired when purchasing new hardware, new hardware that is not (yet) supported by any mainline (non-LHR) release. LHRs have an ‘‘H’’ in the OpenVMS version string, indicating a ‘‘Hardware’’ release.

  You will not generally want to continue using an LHR once a subsequent OpenVMS release is available; you will want to upgrade off the LHR at your earliest convenience.

For minimum OpenVMS versions for various platforms, see Section 2.12.

5.11 Why do I have a negative number in the pagefile reservable pages?

Seeing a negative number in the reservable pages portion of the SHOW MEMORY/FULL command can be normal and expected, and is (even) documented behaviour. A pagefile with a negative number of reservable pages is overcommitted, which is generally goodness assuming that every process with reserved pages does not try to occupy all of the reserved pagefile space at the same time.

To understand how the pagefile reservation process works, think about how a traditional bank operates when accepting customer deposits and making loans. It’s the same idea with the pagefile space. There is less money in the bank vault than the total deposits, because much of the money has been loaned out to other customers of the bank. And the behaviour parallels that of the pagefile down to the problems that a ‘‘run on the bank’’ can cause for banking customers. (Though there is no deposit insurance available for pagefile users.)

If all of the running applications try to use the reserved space, the system manager will need to enlarge the pagefile or add one or more additional pagefules.

To determine if the pagefile is excessively overcommitted, watch for ‘‘double overcommitment’’—when the reservable space approaches the negatation of the available total space—and watch that the total amount of free space available in the pagefile remains adequate. If either of these situations arises, additional pagefile storage is required.

Additional pagefile information: Additional pagefiles can typically be created and connected on a running OpenVMS system. New processes and new applications will tend to use the new pagefile, and existing applications can be restarted to migrate out of the more congested pagefiles. Pagefiles are generally named PAGEFILE.SYS, and multiple pagefiles are generally configured on separate disk spindles to spread the paging I/O load across the available disk storage. When multiple pagefiles are present on recent OpenVMS versions, each pagefile file should be configured to be approximately the same total size as the other pagefiles.

For additional information on pagefile operations and related commands, see the system management and performance management manuals in the OpenVMS documentation set.

With OpenVMS V7.3 and later, the displays have been changed and these negative values are no longer visible.

5.12 Do I have to update layered products when updating OpenVMS?

The Software Public Rollout Reports for OpenVMS list the current and future availability of HP software products shipping on the OpenVMS Software Products Library kits (CDROM consolidations) for OpenVMS Alpha and/or OpenVMS VAX. Specifically, the required minimum versions for product support are listed.

Comprehensive Public Rollout Information, listing previous product versions as well as currently shipping versions, has been compiled into a separate set of reports. The product information is grouped to show Operating System support.

You may or may not be able to use older versions of local applications, third-party products, and various HP OpenVMS layered products with more recent versions of OpenVMS. User-mode code is expected to be upward compatible. Code executing in a privileged processor mode—typically either executive or kernel mode—may or may not be compatible with more recent OpenVMS versions.

These Software Rollout (SWROLL) Reports are updated regularly. Please see:

- http://h71000.www7.hp.com/openvms/os/swroll/

For related information, see Section 2.6.1.

5.13 How do I change the volume label of a disk?

Dismount the disk, and mount it privately. If the disk is mounted by more than one node in an OpenVMS Cluster, dismount it from all other nodes. If this disk is an OpenVMS system disk, shut down all other nodes that are bootstrapped from this disk.

Issue the SET VOLUME/LABEL command, specifying the new label.

On OpenVMS V6.0 and later, issue the following PCSI command to reset the label information stored within the PCSI database to reflect the new disk volume label:

```
$ PRODUCT REGISTER VOLUME old-label device
```

Locate any references in the system startup (typically including the disk MOUNT commands) and any DISK$label references in application files, and change the references appropriately.

If this is a system disk (for the host or for a satellite), also check the DECnet MOP or LANCP boot database, as well as any references to the disk created by CLUSTER_CONFIG*.COM.

If Compaq Analyze is in use, check the system startup procedures for the Compaq Analyze tool. Certain versions of Compaq Analyze will record specific disk volume labels within the startup procedures.

Remount the disk appropriately.

5.14 How can I set up a shared directory?

To set up a shared directory—where all files created in the directory are accessible to the members of specified group of users—you can use an access control list (ACL) and an identifier.

The following also shows how to set up a resource identifier, which further allows the disk resources to be charged to the specified identifier rather than each individual user. (If you don't want this, then omit the attributes option on the identifier creation and omit the entry added in the disk quota database.

Add an identifier using the AUTHORIZE utility:

```
ADD/IDENTIFER/ATTRIBUTES=RESOURCE groupidentifier
```

Grant the identifier to each user in the group using AUTHORIZE:

```
GRANT/IDENTIFIER groupidentifier username
```

If disk quotas are in use, add an entry via SYSMAN for each disk:

```
DISKQUOTA ADD groupidentifier -
  /PERMQUOTA=pq/OVERDRAFT=od/DEVICE=ddcu:
```

Set the shared directory to have an ACL similar to the following using the SET SECURITY (V6.0 and later) or SET ACL (versions prior to V6.0) command:

```
(DEFAULT_PROTECTION,S:RWED,O:RWED,G,W)
(IDENTIFIER=groupidentifier,OPTIONS=DEFAULT,-
  ACCESS=READ+WRITE+EXECUTE+DELETE)
(IDENTIFIER=groupidentifier, -
  ACCESS=READ+WRITE+EXECUTE+DELETE)
(CREATOR,ACCESS=READ+WRITE+ACCESS+DELETE)
```

If there are files already resident in the directory, set their protections similarly. (The OPTIONS=DEFAULT, DEFAULT_PROTECTION, and CREATOR ACEs apply to directories.)

The default protection mask is used to establish the default file protection mask, this mask does not prevent the users holding the specified groupidentifier from accessing the file(s), as they can access the file via the explicit identifier granting access that is present in the ACL.

For further information, see the *OpenVMS Guide to System Security Manual*, specifically the sections on ACLs and identifiers, and resource identifiers.

5.15 Why do I get extra blank pages on my HP Printer?

For information on configuring telnet print symbiont, on device control libraries such as SYSDEVCTL.TLB, and for ways of dealing with the extra blank pages that can arise on various HP printers, please see the OpenVMS Ask The Wizard area, starting particularly with topic (1020):

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

There are a variety of discussions of this and of related printing topics in the Ask The Wizard area, in addition to topic (1020).

Also see Section 5.34.

5.16 Drivers and Configuration of New Graphics Controllers?

This section contains information on various graphics controllers supported by OpenVMS Alpha, and specifically information on where and how to obtain device drivers for specific early OpenVMS releases— device drivers for controllers are integrated into and shipped with OpenVMS Alpha, but versions of these device drivers are sometimes made available for specific earlier OpenVMS releases.

5.16.1 The ELSA GLoria Synergy

On OpenVMS Alpha V7.1-2, V7.2, and V7.2-1, acquire the appropriate GRAPHICS PCSI kit, and all prerequisite OpenVMS ECO kits:

- VMS712_GRAPHICS-V0300 or later
- VMS72_GRAPHICS-V0100 or later
- VMS712_GRAPHICS-V0300 or later

The ELSA GLoria Synergy is the PBXGK-BB; the PowerStorm 3D10T. Please ensure you have the most current ECOs for this and other graphics controllers installed; check for and install the current GRAPHICS kit. (See Section 4.3.2 for some unexpectedly related details.)

On OpenVMS Alpha V7.2-1, the files necessary for this graphics controller are located in the distribution CD-ROM directory:

```
DISK$ALPHA0721:[ELSA.KIT]
```

Also check for any available (later) ECO kits.

An earlier kit (ALP4D20T01_071) (for V7.1, V7.1-1H1, and V7.1-1H2) was once available, but has been superceded and is not recommended. Use of V7.1-2 or later (and use of one the above GRAPHICS kits as required) is typically the best approach.

OpenVMS V7.2-2 and later mainline releases directly support the controller.

Additional information is available in topics (3419) and (5448) in the Ask The Wizard area:

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

Support for the ELSA GLoria Synergy is integrated into all current OpenVMS Alpha releases.

5.16.2 PowerStorm 300, PowerStorm 350

The PowerStorm 300 is the PBXGD-AC, while the PowerStorm 350 is the PBXGD-AE.

For support of the PowerStorm 300 and PowerStorm 350 graphics controllers, acquire and install the following available ECO kits:

For OpenVMS Alpha V7.1-2:

- DEC-AXPVMS-VMS712_P350-V0100–4 or later
- DEC-AXPVMS-VMS712_GRAPHICS-V0300–4 or later

For OpenVMS Alpha V7.2-1:

- DEC-AXPVMS-VMS721_P350-V0100–4 or later
- DEC-AXPVMS-VMS721_GRAPHICS-V0300–4 or later

Support for the PowerStorm 300 and PowerStorm 350 series graphics controllers is integrated into current OpenVMS Alpha releases.

5.16.3 PowerStorm 3D30, PowerStorm 4D20

PowerStorm 3D30 (PBXGB-AA), PowerStorm 4D20 (PBXGB-CA) information is available in Ask The Wizard topics including topic (2041):

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

5.16.4 Radeon 7500

Install the current GRAPHICS ECO kit for OpenVMS Alpha V7.2-2 or V7.3-1 for support of the Radeon 7500 series PCI and AGP graphics controllers.

Support for this controller (without an ECO kit) is first integrated into and available in OpenVMS Alpha V7.3-2. (Please do always install the most current GRAPHICS ECO kit whenever one is available, however.)

5.17 How can I acquire OpenVMS patches, fixes, and ECOs?

You can acquire and download kits containing OpenVMS fixes (ECOs) for various releases, as well as related support information, via the ITRC support center:

- http://www.itrc.hp.com/
- ftp://ftp.itrc.hp.com/openvms_patches/

Some systems with Internet firewalls may/will have to use passive mode FTP to access the above sites. Assuming recent/current versions of the TCP/IP Services package, the DCL FTP command necessary is:

```
$ DIRECTORY/FTP/ANONYMOUS/PASSIVE ftp.itrc.hp.com::
```

You can subscribe to an email notification list at the ITRC site.

For a list of OpenVMS ECO kits recently released, you can use:

- http://Eisner.DECUS.org/conferences/OpenVMS-patches_new_1.HTML

Examples and ECO kit installation instructions are included in the cover letter. For ECO kit email notifications, lists of available ECO kits, cover letters and other associated documentation, look in:

- http://www.itrc.hp.com/
- ftp://ftp.itrc.hp.com/openvms_patches/

For additional information, please see Section 5.17.

Do NOT attempt to install a VMSINSTAL-based OpenVMS ECO kit on OpenVMS Alpha V7.1-2 and later. While VMSINSTAL itself remains available, it is not used for OpenVMS Alpha ECO kits starting in OpenVMS Alpha V7.1-2. OpenVMS Alpha V7.1-2 and later use PCSI for OpenVMS ECO kits.

See Section 5.30 for information on ECO kit checksums.

5.18 How do I move the queue manager database?

To move the location of the queue database, the SYS$QUEUE_MANAGER.QMAN$QUEUES and SYS$QUEUE_MANAGER.QMAN$JOURNAL files, to a disk that is fast(er), has plenty of free space, and that is not heavily used. If the queue database is on a (busy) OpenVMS system disk, you can and probably should move it off the system disk to another disk spindle.

To move the queue database:

1 Checkpoint the journal file. This reduces the file size to the in-memory database size. This will cause the noted delay.

```
$ RUN SYS$SYSTEM:JBC$COMMAND
JBC$COMMAND> DIAG 0 7
```

2 Stop the queue manager

```
$ STOP/QUEUE/MANAGER/CLUSTER
```

3 Backup the .QMAN$QUEUES and .QMAN$JOURNAL files from the present location for safety.

```
$ backup SYS$COMMON:[SYSEXE]SYS$QUEUE_MANAGER.QMAN$*  DISK:[DIR]
```

4 Create a new directory for the queue database. Insure that this disk is accessible to all nodes that can run the queue manager. If the /ON list for the queue manager is ‘‘/ON=(*)’’, the disk must be available to all nodes in the cluster

```
$ CREATE/DIR fast_disk:[qman]
```

5 Copy the .QMAN$QUEUES and .QMAN$JOURNAL files to the new directory

```
$ copy SYS$COMMON:[SYSEXE]SYS$QUEUE_MANAGER.QMAN$*  fast_disk:[qman]
```

6 Delete the old queue database.

```
$ DELETE SYS$COMMON:[SYSEXE]SYS$QUEUE_MANAGER.QMAN$*;*
```

7 Restart the queue manager pointing to the new location

```
$ START/QUEUE/MANAGER fast_disk:[qman]
```

5.19 How do I delete an undeletable/unstoppable (RWAST) process?

‘‘Undeleteable’’ jobs are usually ‘‘undeleteable’’ for a reason—this can track back to insufficient process quotas, to a kernel-mode error in OpenVMS or a third-party device driver, or to other odd problems.

These undeletable jobs typically become of interest because they are holding onto a particular resource (eg: tape drive, disk drive, communications widget) that you need to use... If the particular device supports firmware, ensure that the device firmware is current – TQK50 controllers are known for this when working with old firmware. (That, and the infamous ‘‘MUA4224’’ firmware bug.) If this device has a driver ECO kit available, acquire and apply it... If the particular relevant host component has an ECO, acquire and apply it.

Useful tools include SDA (to see what might be going on) and DECamds (which increase and thus potentially fix quota-related problems). (nb: Applications with quota leaks will obviously not stay fixed.)

If the stuck application is BACKUP, ensure you have the current BACKUP ECO and are directly following the V7.1 or (better) V7.2 or later process quota recommendations for operator BACKUP accounts. Quota details are in the *OpenVMS System Manager’s Manual*.

If the firmware and ECO levels are current, the best approach is to take a system crashdump, and pass a copy of the dump file along to whomever is maintaining the device driver for the particular device/widget/driver involved, with any details on how you got into this situation. (The reboot involved with taking the crashdump will obviously clear the problem.)

There was some kernel-mode code (typically for OpenVMS VAX) that can reset the device ownership field, but that is rather obviously only an interim solution—the real fix is avoiding the loss of the IRP, the process quota leak, or whatever else is ‘‘jamming up’’ this particular process...

5.20 How do I reset the error count(s)?

The system reboot is the only supported approach prior to V7.3-2, but a reboot is obviously undesirable in various situations—there is presently no supported mechanism to reset error counts once the error(s) have been logged on these older releases. On V7.3-2 and later, you can use the DCL command:

```
$ SET DEVICE/RESET=ERROR_COUNT
```

As for an unsupported approach—and be aware of the potential for triggering a system crash, you need to determine the system address of the error count field. For a device, this is at an offset within the device's UCB structure. On VAX, the field is at an offset symbolically defined as UCB$W_ERRCNT. On Alpha, this field's offset is symbolically defined as UCB$L_ERRCNT. The former is a word in size; the latter is a longword.

You now need to locate the system address of the UCB$%_ERRCNT field of the device you wish to reset. Enter SDA. In the following, you will see designations in {} separated by a /. The first item in braces is to be used on the VAX and the second item should be used on an Alpha. (ie. {VAX/Alpha})

```
$ ANALYZE/SYSTEM
SDA>  READ SYS${SYSTEM/LOADABLE_IMAGES}:SYSDEF.STB
SDA>  ! SHOW DEVICE the device with the error(s)
SDA>  SHOW DEVICE <ddnc:>
SDA>  EVALUATE UCB+UCB${W/L}_ERRCNT
Hex = hhhhhhhh   Decimal = -dddddddddd         UCB+offset
```

Record the hexadecimal value 'hhhhhhhh' returned.

You can now exit from SDA and $ RUN SYS$SHARE:DELTA or do what I prefer to do, issue the following:

```
SDA> SPAWN RUN SYS$SHARE:DELTA
```

On both VAX and Alpha, the DELTA debugger will be invoked and will ident- ify itself. On Alpha, there will be an Alpha instruction decoded. For those unfamiliar with DELTA, it does not have a prompt and only one error message—Eh? (Well, for sake of argument, there might be another error produced on the console if you're not careful. This second error is more commonly known as a system crash.)

If you are on a VAX, enter the command: [W

If you are on Alpha, enter the command: [L

These set the prevailing mode to word and longword respectively. Remember the UCB${W/L)_ERRCNT differences?

Now issue the command 1;M

DELTA will respond with 00000001

You are now poised to ZAP the error count field. To do so you need to enter the system address and view its contents. The format of the command to do this is of the form:

```
IPID:hhhhhhhh/
```

For an IPID, use the IPID of the SWAPPER process. It is always: 00010001

Thus, to ZAP the error count, you would enter:

```
00010001:hhhhhhhh/
```

When you enter the / SDA will return the content of the address hhhhhhhh. This should be the error count (in hexadecimal) of the device in question. If it is not, you did something wrong and I'd suggest you type a carriage return and then enter the command EXIT to get out of DELTA. Regroup and see where your session went awry.

If you entered your address correctly and the error count was returned as in the following example, you can proceed.

```
00010001:80D9C6C8/0001   ! output on VAX, 1 error
```

```
00010001:80D9C6C8/00000001   ! output on Alpha, 1 error
```

You can now ZAP the error count by entering a zero and typing a carriage return. For example:

```
00010001:80D9C6C8/0001 0[return]   ! output on VAX. 1 error
```

```
00010001:80D9C6C8/00000001 0[return]   ! output on Alpha, 1 error
```

Now type the command EXIT and a carriage return.

Alternatively, reboot the system.

5.21 How do I find out if the tape drive supports compression?

For various SCSI-based MK-class magnetic tape devices:

```
$ Devdepend2 = F$GETDVI("$n$MKcxxx:","DEVDEPEND2")
$ Comp_sup = %X00200000
$ Comp_ena = %X00400000
$ IF (Devdepend2.AND.Comp_sup).EQ.Comp_sup THEN -
    WRITE SYS$OUTPUT "Compression supported"
$ IF (Devdepend2.AND.Comp_ena).EQ.Comp_ena THEN -
    WRITE SYS$OUTPUT "Compression enabled"
```

5.22 Can I copy SYSUAF to another version? To VAX? To Alpha?

The format of the SYSUAF.DAT, RIGHTSLIST, and associated files are upward-compatible, and compatible across OpenVMS VAX, OpenVMS Alpha and OpenVMS I64 systems. (This compatibility is a a basic requirement of mixed-version OpenVMS Cluster configurations and OpenVMS upgrades—for specific support information, please see the OpenVMS Cluster rolling upgrade and mixed-version requirements.) That said, it's the contents of the SYSUAF and RIGHTSLIST files that will make this more interesting.

The same basic steps necessary for moving RIGHTSLIST and SYSUAF files to another node are rather similar to the steps involved in merging these files in an OpenVMS Cluster—see the appendix of the OpenVMS Cluster documentation for details of merging files. (You might not be merging the contents of two (or more) files, but you are effectively merging the contents of the files into the target system environment.)

Considerations:

- applications often hold SYSUAF or RIGHTSLIST open, meaning a system reboot is often the best way to activate new files.
- the meanings of the RESTRICTED and CAPTIVE flags settings on the UAF entries have changed over time.
- the new NET$PROXY.DAT file that is initially created based on the contents of the NETPROXY.DAT during the OpenVMS VAX V6.1 upgrade and during the OpenVMS Alpha V6.2 upgrade. This file is maintained in parallel with NETPROXY.DAT.
- the RIGHTSLIST identifier values and UIC values that end up scattered around the target system must be rationalized with the contents of the new RIGHTSLIST and SYSUAF files.

The lattermost case—resolving the identifier values—is often the most interesting and difficult part. If you find that an identifier value (or identifier name) from the source RIGHTSLIST collides with that of an identifier existing on the target system, you must first determine if the two identifiers perform the same function. In most cases, they will not. As such, you will have to find and chance all references to the identifier value(s) (or name(s)) to resolve the ‘‘collision’’.

If you encounter a collision, changing both of the identifier binary values (or names) involved in the collision to new and unique values can prevent security problems if you should miss a couple of identifiers embedded somewhere on the target system during the whole conversion process—rather than the wrong alphanumeric value for the identifier being displayed, you'll simply see the binary format for the identifier displayed, and no particular access will be granted. And any DCL commands or such that reference the old alphanumeric name will fail, rather than silently (and potentially erroneously) succeeding.

Similar requirements exist for UIC values, as these too tend to be scattered all over the system environment. Like the binary identifier values, you will find UIC values associated with disks, ACLs, queues, and various other structures.

For a list of the various files shared in an OpenVMS Cluster and that can be involved when relocating an environment from one node to another (or merging environments into an OpenVMS Cluster), please see the SYLOGICALS.TEMPLATE file included in OpenVMS V7.2 and later releases.

Procedures to extract the contents of a (potentially corrupt) queue database are provided on the OpenVMS Freeware (V5) and can be used to combine two queue databases together while shuffling files between OpenVMS Cluster hosts.

For related discussions of splitting a cluster into two or for removing a node from cluster (political divorce, etc), see topics (203), (767), (915) and others in the Ask The Wizard area:

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

5.23 How do I delete (timeout) idle processes?

There is no such command integrated within OpenVMS, though there are (optional) timers available within certain terminal servers and similar devices, and there is an integrated time-of-day mechanism that provides control over when a user can access OpenVMS.

As for available tools, there are DECUS, freeware, and third-party tools known variously as ‘‘idle process killers’’ (IPK) or ‘‘terminal timeout’’ programs, as well as various other names. Examples include: Saiga Systems Hitman, Watchdog, MadGoat Watcher (via the MadGoat site or the OpenVMS Freeware), Kblock, the Networking Dynamics tool known as Assassin, and the Zap tool. Also available is the XLNperformance system management utility, from XLNsystems.

A related package (for DECwindows sessions) is xtermlock.

If the forgetful users are in an application menu environment, the menu can potentially be extended to provide this capability.

5.24 Do I need a PAK for the DECevent (HP Analyze) tool?

DECevent and HP (Compaq) Analyze are available to customers with support contracts. The PAK is required only for the advanced functions of DECevent, the basic bits-to-text translation of the error log does not require a license PAK. Ignore the prompt, in other words. (The PAK should be available to you if you have a hardware support contract or warrantee, and the PAK enables the use of the advanced error analysis and notification capabilities within DECevent.)

Please see the following website for details and downloads: Analyze)

- http://www.compaq.com/support/svctools/

Also see the tool that is available on V7.3-2 and later.

```
$ ANALYZE/ERROR/ELV
```

5.25 INITIALIZE ACCVIO and ANSI tape label support?

A change was made (back in 1988) to (as it was then known) VAX/VMS V5.1-1 that added support for the then-new ANSI X3.27-1987 magnetic tape label standard. Prior to the ANSI X3.27-1987 standard, the date field in the ANSI HDR1 record permits dates only as far as the end of Year 1999. With ANSI X3.27-1987, dates through Year 1999 and dates from Years 2000 to 2099 are permitted.

Versions of INIT.EXE and MTAACP.EXE from VAX/VMS releases prior to V5.1-1 will potentially have problems properly processing ANSI magnetic tapes when Y2K and later dates are involved—the DCL INITIALIZE command is known to encounter access violation (ACCVIO) errors.

The available solutions include upgrades, or setting the date back. Direct initialization of the tape with the new headers (via $qio) is also clearly possible, though the limitation within the old MTAACP.EXE magtape ACP image is not nearly so easy to bypass.

5.26 How do I recover from INSVIRMEM errors?

Prior to OpenVMS Alpha V7.0 and on all OpenVMS VAX releases, VIRTUALPAGECNT and PGFLQUOTA limit the amount of virtual address space that is available to each process.

Further limiting the amount of address space is the size of system space (S0 and S1 space). On OpenVMS Alpha versions prior to V7.0 and on all OpenVMS VAX releases, VIRTUALPAGECNT and MAXPROCESSCNT together determine the size of the page table data structures that occupy large tracts of system space. When no system virtual address space is available for the stuff that needs it—this includes the page tables, non-paged pool, and various other structures—then the values of VIRTUALPAGECNT and MAXPROCESSCNT cannot be increased.

In OpenVMS Alpha V7.0 and later, the page table data structures have been moved out of S0 and S1 space and into page table space. In OpenVMS Alpha V7.2 and later, certain large data structures found in non-paged pool (eg: lock management structures) have been moved into 64-bit space, thus freeing up room in non-paged pool and in S0 and S1 space (where non-paged pool resides) while also permitting much larger data structures.

5.27 How can I prevent a serial terminal line from initiating a login?

In SYSTARTUP_VMS.COM, issue the command:

```
$ SET TERMINAL/NOTYPEAHEAD/PERMANENT ddcu:
```

This will prevent any unsolicited terminal input on ddcu:, and this unsolicited input is what triggers JOB_CONTROL to start up LOGINOUT on the terminal. Once LOGINOUT starts up on the serial line, you can see interesting behaviour (eg: audits, process creations, etc) as LOGINOUT tries to ‘‘chat’’ with whatever device is hooked onto the remote end of the serial terminal line.

5.28 How does PCSI use the image BUILD_IDENT field?

The (undocumented) build ident field in an OpenVMS Alpha image header is 16 bytes long, and is used as a counted string of 0-15 characters (ie, as an .ASCIC string, a string with the character count in byte 0) and was originally introduced to provide information for use by VMSINSTAL patch kits to determine whether an image should be replaced or not.

Starting with OpenVMS Alpha V7.1-2, OpenVMS Engineering uses the PCSI utility to package and install ECO kits for OpenVMS. PCSI uses the generation attribute (a 32-bit unsigned integer) specified for files in the product description file (PDF) of a PCSI kit as the basis for performing file conflict detection and resolution. When a product is installed, PCSI modifies the build ident field of Alpha image headers to store an encoded form of the generation number. It also looks at the build ident field of previously installed images to obtain the generation information for those files as input to the file conflict processing algorithm. (Only images have this field, obviously.)

PCSI interprets the build ident field of a previously installed image as follows:

- if the string length is 15, the 5th character is a hyphen, and the last ten characters are a ten digit number with leading zeros, then the last ten characters are treated as a valid generation number.
- for V7.1-2 through V7.2-1, inclusive, if the above test fails, the information is obtained from the PCSI product database.
- in releases after V7.2-1 and with current PCSI ECO kits, if the above test fails, an invalid generation number is treated as 0000000000 so that the ECO kit will simply replace the image rather than assuming the PCSI database is in error.

So, what will you see in the image identification displayed via the ANALYZE/IMAGE command?

For an image that has been built as part of an OpenVMS Engineering system build, you will generally see a build ID string in the format ‘‘X6TE-SSB-0000’’—X6TE is the build number for the OpenVMS Alpha V7.2-1 release. This id format is used within the OpenVMS system build, and can generally only be seen associated with images that have not yet been processed via PCSI.

During the installation of V7.2-1, PCSI will modify the image header to have a build ident string of ‘‘X6TE-0050120000’’. During installation of an ECO kit containing this image with a generation number of 50130052, for example, PCSI would determine that 50130052 is greater than 50120000, and will replace the existing image on the target disk with the version of the image included in the ECO kit.

Ranges of PCSI generation numbers for various OpenVMS releases are included in Table 5–1. The use of xxxx indicates a range of generations is available, from 0000 to 9999, inclusive. *The format of, the particular operation of, and the assignment of PCSI generation numbers is subject to change without notice.*

Table 5–1 PCSI Generation Number

Generation Number	Generation Source
0040100000	V7.1-2
004011xxxx	V7.1-2 ECOs
0050100000	V7.2
005011xxxx	V7.2 ECOs
0050120000	V7.2-1
005013xxxx	V7.2-1 ECOs
0050140000	V7.2-1H1
005015xxxx	V7.2-1H1 ECOs
0050160000	V7.2-2
005017xxxx	V7.2-2 ECOs
0060000000	V7.3
006001xxxx	V7.3 ECOs
0060020000	V7.3-1
006003xxxx	V7.3-1 ECOs
0060100000	V7.3-2
006011xxxx	V7.3-2 ECOs
0070040000	V8.2
007005xxxx	V8.2 ECOs
0070060000	V8.2-1
007007xxxx	V8.2-1 ECOs

5.29 How can I tell what software (and version) is installed?

There is unfortunately no consistent nor single way to make this determination—this is one of the reasons that a move to PCSI installations is underway.

On OpenVMS Alpha, you can use VMSINSTAL.HISTORY and PRODUCT SHOW PRODUCT to determine what packages have been installed via the VMSINSTAL and PCSI tools, respectively.

To see which OpenVMS Alpha ECO kits have been applied, look in VMSINSTAL.HISTORY on OpenVMS Alpha prior to V7.1-2, and use PRODUCT SHOW PRODUCT/FULL on OpenVMS Alpha V7.1-2 and later.

On OpenVMS VAX, you can use PRODUCT SHOW PRODUCT and (for software that is installed via VMSINSTAL on V7.3 and later) in VMSINSTAL.HISTORY.

For products installed on OpenVMS VAX prior to V7.3 using VMSINSTAL, there is no reliable way to determine what products have been installed. If the product provides a RELEASE_NOTES file (as many do), you can look for the list of these files via DIRECTORY SYS$HELP:*.RELEASE_NOTES. Again, this approach is NOT reliable: some kits do not provide release notes, some system managers will install only the release notes, some system managers will delete release notes, and release notes for multiple versions can be present.

On most packages, you can generally use ANALYZE/IMAGE on one of the core images, looking at the image identification area. Some of the product-specific mechanisms available are:

- DQS DQS$VERSION logical name
- C CC/VERSION
- C++ CXX/VERSION
- TCP/IP TCPIP SHOW VERSION command

5.30 What file checksum tools are available for OpenVMS?

The undocumented (prior to V8.2) DCL command CHECKSUM is the usual means, and provides a rather simple-minded checksum suitable to detect basic file corruptions. Starting with V8.2, additional algorithms beyond the classic XOR scheme are available.

One of the most common schemes beyond the CHECKSUM XOR scheme is MD5, and information and a source code example are available via the MD5 RFC. Pre-built versions of MD5 are available at the OpenVMS Freeware website (http://www.hp.com/go/openvms/freeware/), and on the OpenVMS Freeware V8.0 distribution. Also see the CHECKSUM/ALGORITHM=MD5 command available on OpenVMS V8.2 and later.

The OpenVMS Alpha ECO (patch) kit checksums available at the ECO website are determined using the following DCL command sequence:

```
$ CHECKSUM kitname.pcsi-dcx_axpexe
$ SHOW SYMBOL CHECKSUM$CHECKSUM
```

See Section 5.17 for information on acquiring OpenVMS ECO (patch) kits.

5.31 What (and where) is the OpenVMS Management Station?

For information and current kits for the OpenVMS Management Station (OMS), a PC-based tool that permits you to manage an OpenVMS system, please see:

- http://www.openvms.compaq.com/openvms/products/argus/

5.32 How to determine current disk fragmentation level?

The HP OpenVMS Disk File Optimizer (DFO) defragmentation package provides a fragmentation monitoring tool, and a DFO product authorization key (PAK) is not required for the fragmentation reporting tool:

```
$ DEFRAG SHOW/VOLUME ddcu:
```

The DFU tool available on the OpenVMS Freeware can generate a report on the disk fragmentation:

```
DFU> REPORT ddcu:
```

5.33 SYSBOOT-I-FILENOTLOC, Unable to locate SYS$CPU_ROUTINES?

A message at the OpenVMS Alpha bootstrap such as the following:

```
%SYSBOOT-I-FILENOTLOC, Unable to locate
  SYS$CPU_ROUTINES_1C02.EXE
%SYSBOOT-E-LDFAIL, failed to load execlet,
  status = 00000910
```

indicates that the particular OpenVMS Alpha release does not contain support for the target platform. In this case, OpenVMS does not recognize Alpha family 1C member 02 as a supported platform. A later version of OpenVMS might support the platform, or there might be no support on any release. Ensure that you have the most current firmware, and review the minimum version requirements for the platform.

The execlet load failure and other similar bootstrap status values can often be decoded using either of the following techniques:

```
$ exit %x910
%SYSTEM-W-NOSUCHFILE, no such file
$
```

```
$ x = f$message(%x910)
$ show symbol x
  X = "%SYSTEM-W-NOSUCHFILE, no such file"
$
```

Also see Section 14.4.4.1.

5.34 How can I customize the DCPS device control for a new printer?

To customize DCPS for an otherwise unsupported printer, you can try the following sequence:

- Extract the most closely-associated setup modules from the existing device control library, DCPS$DEVCTL.TLB. (For instance, you can probably extract and use the HP LaserJet 4000 series definitions for the HP LaserJet 4050 series. Each printer will vary, please consult the printer documentation for specifics and requirements.)
- rename each extracted setup module to a corresponding:

  ```
  LPS$$UNRECOGNIZED_*
  ```

- Insert all of the above-renamed setup modules into a newly-created device control library specific to the new printer:

  ```
  $ LIBRARY/TEXT/CREATE -
      SYS$COMMON:[SYSLIB]HP4050_DEVCTL.TLB
      LPS$$UNRECOGNIZED*
  ```

  The above assumes the filename HP4050_DEVCTL.TLB, alter as required.

- Set up your DCPS startup procedures to include a search-list logical name such as:

  ```
  $ DEFINE/SYSTEM/EXECUTIVE DCPS_HP4050_LIB  -
      SYS$LIBRARY:HP4050_DEVCTL.TLB, -
      SYS$LIBRARY:DCPS$DEVCTL.TLB
  ```

- Supply DCPS_HP4050_LIB as the library parameter in the queue startup for this printer, this is the P3 parameter to the command procedure SYS$STARTUP:DCPS$EXECUTION_QUEUE.COM.
- The HP4050_DEVCTL library may/will need to be recreated and modules re-edited and replaced with each DCPS upgrade, particularly if any modules are updated in the original library. You will also want to determine if the upgraded version of DCPS directly supports the particular printer.
- To customize the processing of file extensions within DCPS (to enable or disable graybar output, for instance), use the information available in:

  ```
  SYS$LIBRARY:DCPS$FILE_EXTENSION_DATA_TYPE.DAT_DEFAULT
  ```

to create your own site-specific:

```
SYS$LIBRARY:DCPS$FILE_EXTENSION_DATA_TYPE.DAT
```

Also see Section 5.15.

5.35 Why do $GETDEV MOUNTCNT and SHOW DEVICE mount counts differ?

MOUNTCNT returns the local mount count, while SHOW DEVICE returns the cluster-wide mount count.

5.36 What software is needed for Postscript printers?

The NorthLake PrintKit (www.nls.com) and DECprint Supervisor (DCPS) are common choices for support of Postscript printers on OpenVMS.

- http://www.nls.com/
- http://www.openvms.compaq.com/openvms/Print/print_sw_prods.html

You may also require the installation of an IP transport stack.

Also please see Section 15.2.2 and Section 15.2.3.

5.37 How do I remove a PCSI-installed patch (ECO) kit?

You cannot PRODUCT REMOVE a PCSI patch (ECO) kit.

In order to remove an ECO kit, PCSI would have to have copies of all the other version of the files from all other patches and products that previously were installed. This can clearly involve a large number of files and a large archive of old file versions and a substantial quantity of disk space. While removal is clearly theoretically possible, it is not currently implemented.

The following is the supported mechanism to remove a PCSI patch kit.

1 Execute a PRODUCT SHOW PRODUCT product-name. /FULL command. The ‘‘maintenance’’ column (132 column width) shows the patches that have been installed. Keep a copy of this listing.

2 Acquire kits for all of the maintenance kits listed.

3 Re-install the prior FULL version of the product. This will remove all patch kits, setting to product back to ‘‘original’’ condition.

4 Re-install all the patches in the list from step 1, except those patches which you have determined you do not want.

The above information also applies to PCSI PARTIAL kits.

5.38 SYSINIT-E, error mounting system device, status=0072832C

This message can arise during an OpenVMS system bootstrap...

```
%MOUNT-F-DIFVOLMNT, different volume already mounted on this device
```

For details and further information, use the DCL command:

```
$ HELP/MESSAGE /STATUS=%X72832C
```

5.39 Resolving License PAK Problems?

The PAK release date, the PAK termination date, and the PAK version are the usual culprits when a license product authorization key (PAK) check failure occurs.

The PAK termination date is the date when the license PAK will expire.

The PAK release date is the date of the most recent release date of the software package that will be permitted by the particular license PAK. (The release date check is analogous to a product version check.) The PAK version indicates the most recent product version that is permitted by the license.

Having multiple license PAKs registered (and active) can also cause problems if an expired PAK gets loaded. You will want to DISABLE license PAKs you do not wish to have loaded.

Other problems include a failure to register each PAK in all license databases throughout a multiple-system-disk cluster, with a consistent set of /INCLUDE lists specified across each of the duplicated PAKs.

Additionally, you could have an invalid LMF$LICENSE logical name defined. (If no LMF$LICENSE logical name is defined, the standard license database named SYS$SYSTEM:LMF$LICENSE.LDB will be used.)

You can display license failures by defining the following logical name:

```
$ DEFINE/SYS/EXEC LMF$DISPLAY_OPCOM_MESSAGE TRUE
```

Enable your terminal as a license operator (REPLY/ENABLE=LICENSE), define the LMF$DISPLAY_OPCOM_MESSAGE logical name, and then try the failing operation again. You should see one or more OPCOM messages displayed.

If you have the LMF$DISPLAY_OPCOM_MESSAGE logical name defined, you can (will?) see spurious license check failures—various products will check for multiple licenses, and a few products will check for PAKs that either have not yet been or will not be issued. Once you figure out which license has failed, you will want to deassign this logical name.

Note

That there are no license check failures does *not* indicate that the particular product or operation or use is permissible per applicable licensing agreements. Please consult the applicable agreement(s) for licensing-related information and requirements.

To register a license PAK on a DECwindows system when DECwindows cannot start (because of an expired license or other licensing problem), follow the steps outlined in section Section 5.6 up through the use of the AUTHORIZE command. In place of the AUTHORIZE command, use the console to register the license PAKs. Also see Section 12.4 for licensing and troubleshooting information.

For information on licensing and on the numbers of license units required for various products and various platforms, the License Unit Requirements Table (LURT) is available at:

- http://www.compaq.com/products/software/info/

5.40 Changing the OpenVMS Version Number?

Fool your friends, baffle your enemies, run the OpenVMS version of your choice! Instantly back-port all those useful new OpenVMS features to an ancient OpenVMS version!

On OpenVMS Alpha systems:

```
$ SET DEFAULT SYS$COMMON:[SYS$LDR]
$ RUN SYSVER
REPLACE V9.9
WRITE
$ EXIT
```

How does SYSVER work? SYSVER looks for SYS$BASE_IMAGE.EXE within the default directory, and updates the version string stored within. And yes, obviously, SYS$BASE_IMAGE.EXE resides in the directory SYS$COMMON:[SYS$LDR].

On OpenVMS VAX systems:

```
$ set default SYS$COMMON:[SYS$LDR]
$ copy SYS.EXE SYS.EXE_IN-CASE-I-FAIL
$ patch SYS.EXE
define sys$gq_version=800044b8
set mode ascii
!examine sys$gq_version
!examine sys$gq_version+4
deposit sys$gq_version   = "V9.9"
deposit sys$gq_version+4 = "    "
update
exit
$ Exit
```

Then reboot the system at your leisure.

5.41 How to prevent users from choosing obvious passwords?

To prevent users from selecting obvious passwords on OpenVMS, you will want to use the reserved password (password screening) mechanism. Effectively, you merge your list of reserved passwords into the existing reserved words database maintained by OpenVMS. (You can also then require all users to reset their passwords—via the pre-expired password mechanism—thus forcing users to select new passwords.) For details on the password screening mechanism, of the reserved password database (VMS$PASSWORD_DICTIONARY.DATA), and details of how to merge your list of prohibited passwords into the database, please see the associated chapter in the OpenVMS security manual. For details of the password expiration mechanism, see the AUTHORIZE command qualifier /PWDEXPIRED.

You can also implement a site-specific password filter with the information provided in the back of the OpenVMS Programming Concepts manual. The password filter permits you to establish particular and site-specific password requirements. For details, please see the system parameter LOAD_PWD_POLICY and the programming concepts manual, and see the examples in SYS$EXAMPLES:. (Examples and documentation on V7.3 and later reflect both platforms, the examples are found only on OpenVMS VAX kits on earlier releases. The capabilities have existed on both the VAX and Alpha platforms for some time now.)

To verify current passwords, you can also use a technique known to system crackers as the ‘‘dictionary attack’’ – the mechanism that makes this attack somewhat more difficult on OpenVMS is the hashing scheme used on OpenVMS, and the file protections used for the SYSUAF authorization database. Given a dictionary of words and the unprotected contents of the SYSUAF file, a search for obvious passwords can be performed. Interestingly, a ‘‘dictionary attack’’ also has the unfortunate side-effect of exposing the password to the user—while this is clearly the goal of a system cracker, authorized privileged and non-privileged system users should not know nor have access to the (cleartext) passwords of other users.

Accordingly, OpenVMS does not store the cleartest password. Further, OpenVMS uses a password hashing algorithm, not an encryption algorithm. This means that storage of a cleartext password is deliberated avoided, and the cleartext value is deliberately very difficult to obtain. The hash is based on a Purdy Polynomial, and the hash itself includes user-specific values in addition to the password, values that make the results of the password hash unique to each user.

Regardless of the use of a password hashing scheme, if a copy of your password file should become available to a system cracker, you will want to force all users to use new passwords immediately.

If you should require a user to verify a password, use the username, the user's salt value (this value is acquired via $getuai) and the user's specified cleartext password, and compare the resulting hashed value (using a call to $hash_password) against the saved hashed password value (this value also acquired via $getqui). For reasons of security, avoid saving a cleartext password value in any data files, and do not maintain the cleartext password in memory longer than required. (Use of sys$acm on V7.3-1 and later is recommended.)

Kerberos authentication (client and server) is available on OpenVMS V7.3 and later. Integration of Kerberos support into various Compaq and into third-party products is expected.

External authentication is available in V7.3-1 and later, with support for user-written external authentication in V7.3-2 and later.

If you are simply looking for OpenVMS access and the SYSTEM and all other privileged passwords are forgotten or otherwise unavailable, please see section Section 5.6 and/or the OpenVMS documentation set.

Also please see the NCSC C2 guidelines in the OpenVMS security manual.

5.42 Please help me with the OpenVMS BACKUP utility?

5.42.1 Why isn't BACKUP/SINCE=BACKUP working?

If you are seeing more files backed up than previously, you are seeing the result of a change that was made to ensure BACKUP can perform an incrementation restoration of the files. In particular, if a directory file modification date changes, all files underneath it are included in the BACKUP, in order to permit incremental restoration should a directory file get renamed.

5.42.1.1 Why has OpenVMS gone through the agony of this change?

When a directory is renamed, the modified date is changed. When the restoration needs to restore the directory and its contents, and the restoration should not result in the restoration of the older directory name when a series of incremental BACKUPs are restored. Thus an incremental BACKUP operation needs to pick up all of the changes.

Consider performing an incremental restoration, to test the procedures. This testing was how OpenVMS Engineering found out about the problem that was latent with the old BACKUP selection scheme—the old incremental BACKUP scheme would have missed restoring any files under a renamed directory. Hence the change to the selection mechanisms mentioned in Section 5.42.1.

5.42.1.2 Can you get the old BACKUP behaviour back?

Yes, please see the /NOINCREMENTAL qualifier available on recent OpenVMS versions (and ECO kits). Use of this qualifier informs BACKUP that you are aware of the limitations of the old BACKUP behaviour around incremental disk restorations.

5.42.2 What can I do to improve BACKUP performance?

Use the documented commands in the manual for performing incremental BACKUPs. Use the documented incremental procedures. Don't try to use incremental commands in a non-incremental context.

Also consider understanding and then using /NOALIAS, which will likely be a bigger win than will anything to do with the incremental BACKUPs, particularly on system disks and any other disks with directory aliases.

See the OpenVMS documentation for additional details.

Ignoring hardware performance and process quotas, the performance of BACKUP during a disk saveset creation is typically limited by three factors:

1. Default extend size

   The default behavior can have poor performance, as the extend operation can involve extensive additional processing and I/O operations. Consider changing the default extend value on the volume, or change the extend for the process:

   ```
   $ set rms/extend=65000
   ```

2. Output IO size

   The default IO size for writing an RMS sequential file is 32 blocks, an increase from the value of 16 blocks used on earlier versions. Setting this to the maximum of 127 can reduce the number of IOs by almost a factor of 4:

   ```
   $ set rms/block=127
   ```

Note that the performance might be better on some controllers if the block count is a multiple of 4 - e.g. 124

3 Synchronous writes to the saveset

Starting with OpenVMS V7.3, you can now persuade RMS to turn on write-behind for sequential files opened unshared. (Please see the V7.3 release notes or more recent documentation for details.) Enabling the write-behind operations involves setting the dynamic system parameter RMS_SEQFILE_WBH to 1. This parameter is dynamic, and it can be enabled and disabled without a reboot, and changes in its setting can and will directly effect the running system. In order to get the full benefit from write-behind operations, you also need to increase the RMS local buffer count from the default of 2 to a larger number. Raising the value to 10 is probably a reasonable first estimate for this value.

```
$ run sys$system:sysman
PARAMETERS USE ACTIVE
PARAMETERS SET RMS_SEQFILE_WBH 1
PARAMETERS WRITE ACTIVE
EXIT
$ SET RMS/BUFFER=10/EXTEND=65000/BLOCK=127
$ BACKUP source-specification ddcu:[dir]saveset.bck/SAVE
```

5.42.3 Why is BACKUP not working as expected?

First, please take the time to review the BACKUP documentation, and particularly the BACKUP command examples. Then please download and install the most current BACKUP eco kit. Finally, please please set the process quotas per the System Management documentation. These steps tend to resolve most problems seen.

BACKUP has a very complex interface, and there are numerous command examples and extensive user documentation available. For a simpler user interface for BACKUP, please see the documentation for the BACKUP$MANAGER tool.

As for recent BACKUP changes, oddities, bugs, etc:

- A change made in OpenVMS V6.2 WILL cause more files to be included into a file-based BACKUP saveset using /SINCE=BACKUP as all files underneath any directory with a sufficiently recent (selected) date will be included in the saveset. This change was deliberate and intentional, and was mandated by the need to provide a functional incremental restoration.

Without the inclusion of these apparently-extra files, an incremental saveset can NOT be reliably restored.

- As part of the OpenVMS V6.2 change, the /SINCE command—without the specification of the =BACKUP keyword—selected more files than it should have. This is a bug. This bug has been remedied in the OpenVMS BACKUP source code and in some of (all of?) the BACKUP ECO kits.

When working with BACKUP, you will want to:

- Ensure you have your process quotas set per the recommendations in the OpenVMS System Management documentation. Deviation from these values can and will lead to access violation (ACCVIO) and other untoward behaviour.
- Get the current BACKUP ECO kit and install it BEFORE you attempt to troubleshoot any problems.
- Learn about the /NOINCREMENTAL (new) and /NOALIAS (V6.2 and later) command qualifiers. The former qualifier returns to the pre-V6.2 behaviour of the /SINCE file selection mechanism, while the latter (specified with /IMAGE) reduces the replication of files on system disks and other disks with file alias and directory alias entries. Both of these can reduce the numbers of files that will be selected and thus included into the saveset.

  Learn what /IGNORE=INTERLOCK means. This command probably does not provide what you think it does—those file system interlocks that this command is ignoring were implemented for a reason, after all. Ignoring these interlocks can lead to missed data and potentially to corruptions to individual files stored within the output saveset, corruptions that may or may not be reported. For details on this BACKUP command qualifier, please see the Ask The Wizard topic (2467).

When working with the BACKUP callable API:

- Build your applications with the most current BACKUP API available. Changes made to the V7.1-2 and V7.2 API were incompatible with the V7.1 and V7.2-1 and later APIs, and this incompatibility was repaired via a BACKUP ECO kit. Do NOT build your application with the versions of the BACKUP API that shipped with V7.1-2 and V7.2, as these are incompatible with the BACKUP API constants that were used on other versions.

5.42.4 How do I fix a corrupt BACKUP saveset?

BACKUP savesets can be corrupted by FTP file transfers and by tools such as zip (particularly when the zip tool has not been asked to save and restore OpenVMS file attributes or when it does not support OpenVMS file attributes; use the zip ‘‘-V’’ option), as well as via other means of corruptions.

If you have problems (eg: NOTSAVESET errors) with the BACKUP savesets after unzipping them or after an FTP file transfer, you can try restoring the appropriate saveset attributes using the tool:

```
$ BACKUP/LIST saveset.bck/SAVE
Listing of save set(s)

%BACKUP-F-NOTSAVESET, saveset.bck/SAVE is not a BACKUP save set
$ @SRH:[UTIL]RESET_BACKUP_SAVESET_FILE_ATTRIBUTES.COM saveset.bck
$ BACKUP/LIST saveset.bck/SAVE
Listing of save set(s)

Save set:          saveset.bck
Written by:        username
...
```

This tool is available on the OpenVMS Freeware (in the [000TOOLS] directory). The Freeware is available at various sites—see the Freeware location listings elsewhere in the FAQ—and other similar tools are also available from various sources.

In various cases, a SET FILE/ATTRIBUTES command can also be used. As the parameters of this command must be varied as the target BACKUP saveset attributes vary, this approach is not recommended.

Also see the ‘‘SITE VMS’’, /FDL, and various other file-attributes options available in various FTP tools. (Not all available FTP tools support any or all of these options.)

Browser downloads (via FTP) and incorrect (binary or ascii FTP transfer modes) are notorious for causing RMS file corruptions and particularly BACKUP saveset corruptions. You can sometimes help encourage the browser to select the correct FTP transfer type code (via RFC1738):

- ftp://host/urlname.ext;type=i ! request ftp image/binary transfer
- ftp://host/urlname.ext;type=a ! request ftp ascii/text transfer

You can also often configure the particular web browser to choose the appropriate transfer mode by default, based on the particular file extensions, using a customization menu available in most web browsers. You can select that the specific file extentions involved use the FTP binary transfer mode, which will reduce the number of corruptions seen.

5.42.5 How do I write a BACKUP saveset to a remote tape?

How to do this correctly was described at DECUS long ago. On the OpenVMS host with the tape drive, create the following SAVE-SET.FDL file:

```
RECORD
        FORMAT                  fixed
        SIZE                    8192
```

Then create BACKUP_SERVER.COM:

```
$ !
$ ! BACKUP_SERVER.COM - provide remote tape service for BACKUP.
$ !
$ set noon
$ set rms/network=16
$ allocate mka500 tapedev
$ mount/nounload/over:id/block=8192/assist tapedev
$ convert/fdl=SAVE-SET sys$net tapedev:save-set.
$ dismount/unload tapedev
$ stop/id=0
```

On the node where you want to do the backup, use the DCL command:

```
$ backup -
    srcfilespec -
    node"user pwd"::"task=backup_server"/block=8192/save
```

One area which does not function here is the volume switch; multi-reel or multi-cartridge savesets. Since the tape is being written through DECnet and RMS and the magtape ACP, BACKUP won't see the media switch and will split an XOR group across the reel boundary. BACKUP might well be willing to read such a multi-reel or multi-cartridge saveset (directly, not over the net) as the XOR blocks are effectively ignored until and unless needed for error recovery operations. BACKUP likely will not be able to perform an XOR-based recovery across reel or cartridge boundaries.

Unfortunately BACKUP can't read tapes over the network because the RMS file attributes on a network task access look wrong; the attributes reported include variable length records.

5.42.6 How to perform a DoD security disk erasure?

Sometimes refered to as disk, tape, or media declassification, as formatting, as pattern erasure, or occasionally by the generic reference of data remanence. Various references to the US Deparment of Defence (DoD) or NCSC ‘‘Rainbow Books’’ documentation are also seen in this context.

While this erasure task might initially appear quite easy, basic characteristics of the storage media and of the device error recovery and bad block handling can make this effort far more difficult than it might initially appear.

Obviously, data security and sensitivity, the costs of exposure, applicable legal or administrative requirements (DoD, HIPPA or otherwise), and the intrinsic value of the data involved are all central factors in this discussion and in the decision of the appropriate resolution, as is the value of the storage hardware involved.

With data of greater value or with data exposure (sometimes far) more costly than the residual value of the disk storage involved, the physical destruction of the platters may well be the most expedient, economical, and appropriate approach. The unintended exposure of a bad block containing customer healthcare data or of credit card numbers can quite be costly, of course, both in terms of the direct loss, and the longer-term and indirect costs of such exposures.

Other potential options include the Freeware RZDISK package, the OpenVMS INITIALIZE/ERASE command (and potentially in conjunction with the $erapat system service) and OpenVMS Ask The Wizard (ATW) topics including (841), (3926), (4286), (4598), and (7320). For additional information on sys$erapat, see the OpenVMS Programming Concepts manual and the OpenVMS VAX examples module SYS$EXAMPLES:DOD_ERAPAT.MAR. Some disk controllers and even a few disks contain support for data erasure. Some DSSI Disk ISEs, for instance.

For the prevention of casual disk data exposures, a generic INITIALIZE/ERASE operation is probably sufficient. This is not completely reliable, particularly if the data is valuable, or if legal, administrative or contractual restrictions are stringent—there may well be revectored blocks that are not overwritten or not completely overwritten by this erasure, as discussed above, and these blocks can obviously contain at least part of most any data that was stored on the disk – but this basic disk overwrite operation is likely sufficient to prevent the typical information disclosures.

You will want to consult with your site security officer, your corporate security or legal office, with HP Services or your prefered service organization, or with a firm that specializes in erasure or data declassification tasks. HP Services does traditionally offer a secure disk declassification service.

5.42.7 How to enable telnet virtual terminals?

To enable virtual terminal support for telnet and rlogin devices, add the following logical name definitions into SYLOGICALS.COM:

```
$ DEFINE/SYSTEM/EXECUTIVE TCPIP$RLOGIN_VTA TRUE
$ DEFINE/SYSTEM/EXECUTIVE TCPIP$TELNET_VTA TRUE
```

See SYS$STARTUP:SYLOGICALS.TEMPLATE for details on the typical contents of SYLOGICALS.COM.

In SYSTARTUP_VMS.COM, ensure that a command similar to the following is invoked:

```
$ SYSMAN IO CONNECT VTA0/NOADAPTER/DRIVER=SYS$LOADABLE_IMAGES:SYS$TTDRIVER.EXE
```

In MODPARAMS.DAT, add the following line or (if already present) mask the specified hexidecimal value into an existing TTY_DEFCHAR2, and perform a subsequent AUTOGEN with an eventual reboot:

```
TTY_DEFCHAR2 = %X20000
```

This value is TT2$M_DISCONNECT.

On older TCP/IP Services—versions prior to V5.0—you will have to perform the following UCX command:

```
$ UCX
UCX> SET CONF COMM/REMOTE=VIRTUAL
```

5.42.7.1 Volume Shadowing MiniCopy vs MiniMerge?

MiniMerge support has been available for many years with OpenVMS host-based volume shadowing, so long as you had MSCP controllers (eg: HSC, HSJ, or HSD) which supported the Volume Shadowing Assist known as ‘‘Write History Logging’’.

If you are interested in mini-merge and similar technologies, please see the Fibre Channel webpage and the information available there:

- http://www.openvms.compaq.com/openvms/fibre/

Mini-Merge support was originally intended to be controller-based and was expected with HSG80 series storage controllers and was expected to require ACS 8.7 and OpenVMS Alpha V7.3-1.

Host-based Mini-Merge (HBMM) is now available for specific OpenVMS releases via a shadowing ECO kit, and is also present in OpenVMS V8.2 and later. HBMM applies to the HSG80 series and—like host-based volume shadowing—to most other (all other?) supported storage devices.

The following sections describe both Mini-Copy and Mini-Merge, and can provide a basis for discussions.

5.42.7.1.1 Mini-Copy?

A Shadowing Full Copy occurs when you add a disk to an existing shadowset using a MOUNT command; the entire contents of the disk are effectively copied to the new member (using an algorithm that goes through in 127-block increments and reads one member, compares with the target disk, and if the data differs, writes the data to the target disk and loops back to the read step, until the data is equal for that 127-block section). (This is one of the reasons why the traditional recommendation for adding new volumes to a shadowset was to use a BACKUP/PHYSICAL copy of an existing shadowset volume, simply because the reads then usually matched and thus shadowing usually avoided the need for the writes.)

If you warn OpenVMS ahead of time (at dismount time) that you're planning to remove a disk from a shadowset but re-add it later, OpenVMS will keep a bitmap tracking what areas of the disk have been modified while the disk was out of the shadowset, and when you re-add it later with a MOUNT command OpenVMS only has to update the areas of the returned disk that the bit-map indicates are now out-of-date. OpenVMS does this with a read source / write target algorithm, which is much faster than the shenanigans the Full Copy does, so even if all of the disk has changed, a Mini-Copy is faster than a Full Copy.

5.42.7.1.2 Mini-Merge?

A Shadowing Merge is initiated when an OpenVMS node in the cluster (which had a shadowset mounted) crashes or otherwise leaves unexpectedly, without dismounting the shadowset first. In this case, OpenVMS must ensure that the data is identical, since Shadowing guarantees that the data on the disks in a shadowset will be identical. In a regular Merge operation, Shadowing uses an algorithm similar to the Full Copy algorithm (except that it can choose either of the members' contents as the source data, since both are considered equally valid), and scans the entire disk. Also, to make things worse, for any read operations in the area ahead of what has been merged, Shadowing will first merge the area containing the read data, then allow the read to occur.

A Merge can be very time-consuming and very I/O intensive. If a node crashes, the surviving nodes can query to determine what exact areas of the disk the departed node was writing to just before the crash, and thus Shadowing only needs to merge just those few areas, so this tends to take seconds, as opposed to potentially requiring many minutes or even hours for a regular full Merge.

5.43 Please explain DELETE/ERASE and File Locks?

DELETE/ERASE holds the file lock and also holds a lock on the parent directory for the duration of the erasure. This locking can obviously cause an access conflict on either the file or on the directory—it might well pay to rename files into a temporary directory location before issuing the DELETE/ERASE, particularly for large files and/or for systems with multiple overwrite erase patterns in use; for any systems where the DELETE/ERASE erasure operation will take a while.

5.44 Managing File Versions?

Some applications will automatically roll file version numbers over, and some will require manual intervention. Some will continue to operate without the ability to update the version, and some will be unable to continue. Some sites will specifically (attempt to) create a file with a version of ;32767 to prevent the creation of additional files, too.

To monitor and resolve file versions, you can use commands including:

```
$ SET FILE/VERSION_LIMIT=n filename
$ SET DIRECTORY/VERSION_LIMIT=n [directory]
```

And you can also monitor file version numbers, and can report problems with ever-increasing file versions to the organization(s) supporting the application(s) generating files with ever-increasing version numbers for details on potential problems, and for any recommendations on resetting the version numbers for the particular product or package. If required, of course.

The following pair of DCL commands—though obviously subject to timing windows— can be used to rename all the versions of a file back down to a contiguous sequence of versions starting at 1:

```
$ RENAME file.typ;*   RENAME.TMP;
$ RENAME RENAME.TMP;* file.typ;
```

The key to the success of this RENAME sequence is the specification of (only) the trailing semicolon on the second parameter of each of the RENAME commands.

You may also see the numbers of files reduced with DELETE commands, with multiple directories, or with PURGE commands such as the following examples:

```
$ PURGE/BEFORE="-2-"
$ PURGE/BEFORE="TODAY-2-"
$ PURGE/KEEP=10"
```

You can use DFU (Freeware) to quickly and efficiently scan for all files with large(r) version numbers:

```
DFU SEARCH/VERSION=MINIMUM=nnnn
```

If you are creating or supporting an application, selecting temporary or log file filenames from among a set of filenames—selecting filenames based on time, on process id, on the day of week, week number, or month, on the f$unique lexical (V7.3-2 and later), etc—is often useful, as this approach more easily permits on-line adjustments to the highest file versions and easily permits on-line version compression using techniques shown above. With differing filenames, you are less likely to encounter errors resulting from files that are currently locked. You can also detect the impending version number limit within the application, and can clean up older versions and roll the next file version creation to ;1 or such.

Also see Section 9.4.

5.45 Host-based Volume Shadowing and RAID?

Host-based Volume Shadowing (HBVS) is Disk Mirroring is RAID Level 1.

HBVS is capable of shadowing devices of different geometries, of different block counts (with dissimilar device shadowing; allowing for mixtures of hardware) and—with dynamic volume expansion—of growing volumes on the fly, and HBVS is capable of shadowing/mirroring/raid-1 operations across cluster configurations up to the full span—please see the Cluster SPD for the current supported span; the supported span limit is currently multiple hundreds of kilometers—of a cluster. HBVS can be layered onto controller (hardware) RAID, as well.

For information on host-based striping, please see the StorageWorks RAID product.

6 Information on Mail

**If you are searching for something here,
please consider using the text-format FAQ.**

6.1 MAIL keeps saying I have new messages, but I don't. What do I do?

if you see the *%MAIL-W-NONEWMAIL, no new messages* error reported when MAIL indicates you have messages, then the NEWMAIL count has become skewed from reality.

The count of new mail messages is kept separately from your mail folder, and is stored in VMSMAIL_PROFILE.DATA. It sometimes happens that this count differs from what is stored in your mail folder. If this arises, invoke MAIL and repeatedly enter the READ/NEW command (or press the keypad hyphen key on an LK-compatible keyboard) until you see no new mail messages. Then enter the command one more time. This will resynchronize the counters.

If you are operating in a cluster and find your mail counts inconsistent across cluster members, your customer is likely missing a definition of the VMSMAIL_PROFILE logical name—and is probably also missing definitions of other logical names associated with other shared files—or has one or more inconsistent definitions of this and likely of other logical names.

For details on the configuration data files that must be shared within a cluster, please see SYS$STARTUP:SYLOGICALS.TEMPLATE on V7.2 and later.

6.2 How do I send or read attachments in VMS MAIL?

Is there any way to send or read mail with files as attachments from VMS?

Not directly with the OpenVMS MAIL facility, but there are several other options:

- Install PINE, available commercially from Innosoft or free from Andy Harper. With PINE you can both send and receive MIME messages, if you have the appropriate viewers available.
  - http://www.process.com/

  - http://www.agh.cc.kcl.ac.uk/files/vms/pine-vms/
  - ftp://ftp2.kcl.ac.uk/pub/vms/pine-vms/
- If you are working from an X Windows display, you can use the OpenVMS version of Netscape Navigator or Mozilla. The mail download protocol chosen to access the mail server from the Navigator or Mozilla mail client can be POP or IMAP, with the former causing messages to be downloaded while the latter permits messages to be retained on the mail server. Most folks prefer IMAP for this reason.
- MPACK/MUNPACK. To send a MIME mail, construct the message with attachments manually using MPACK. You cannot send the resulting file directly through MAIL because an extra blank header line will be inserted between your message and the OpenVMS MAIL headers, which will cause the message to appear as plain text in most mail programs. Some TCP/IP stacks provide a work around for this problem, and if that doesn't work, you should generally be able to force the message directly into the SMTP port of your mail machine. Examples of both methods are in:
  - http://saf.bio.caltech.edu/pub/software/openvms/mmail.com

  To read a MIME mail message, open it in MAIL, extract it to a file, then use MUNPACK to break out and decode the attachments.

  MPACK/MUNPACK tools are also available on OpenVMS Freeware V5.0.
- With OpenVMS V7.2 and later, use the MIME tool supplied with OpenVMS.

6.3 How can I block SMTP mail relay spam?

Use the anti-spam capabilities present in the TCP/IP Services V5.1 and later SMTP servers.

Use a firewall.

On earlier TCP/IP Services releases, some simple DCL can reportedly prevent relay SMTP spam. Use the UCX command SHOW SERVICE SMTP/FULL to find the directory containing the UCX$SMTP_RECV_STARTUP.COM file, and insert the following DCL:

```
$       !
$       ! Block spam.
$       !
$       MY_ADDRESS_LONG[0,32]=F$INTEGER(F$TRNLNM("SYS$REM_NODE")-"::")
$       MY_ADDRESS=F$FAO("!UB.!UB.!UB.!UB",F$CVUI(0,8,MY_ADDRESS_LONG),-
         F$CVUI(8,8,MY_ADDRESS_LONG),F$CVUI(16,8,MY_ADDRESS_LONG),-
         F$CVUI(24,8,MY_ADDRESS_LONG))’"
$       MY_ADDRESS_REVERSE=F$FAO("!UB.!UB.!UB.!UB",-
         F$CVUI(24,8,MY_ADDRESS_LONG),F$CVUI(16,8,MY_ADDRESS_LONG),-
         F$CVUI(8,8,MY_ADDRESS_LONG),F$CVUI(0,8,MY_ADDRESS_LONG))’"
$       WRITE SYS$OUTPUT F$TIME()+" "+F$TRNLNM("SYS$REM_NODE")+MY_ADDRESS
$       UCX SHOW HOST ’MY_ADDRESS_REVERSE’.INPUTS.ORBS.ORG
$       IF $STATUS.EQ.1
$       THEN
$         WRITE SYS$OUTPUT "SPAM from relay rejected"
$         EXIT
$       ENDIF
$       UCX SHOW HOST ’MY_ADDRESS_REVERSE’.SPAMSOURCES.ORBS.ORG
$       IF $STATUS.EQ.1
$       THEN
$         WRITE SYS$OUTPUT "SPAM source relay rejected"
$         EXIT
$       ENDIF
$       !
$       ! Run receiver.
$       !
$       run sys$system:ucx$smtp_receiver.exe
$       goto exit
```

7 Information on Utilities

If you are searching for something here, please consider using the text-format FAQ.

7.1 How do I play an audio CD on my workstation?

If you've installed the DECwindows examples, you'll find DECW$CDPLAYER.C, .DAT, .EXE, .UIL, and .UID. Copy the .UID and .DAT files to DECW$USER_DEFAULTS: (typically SYS$LOGIN:), define the logical name DECW$CD_PLAYER to be the device name of your CD-ROM drive (eg. DKA400:), give yourself PHY_IO and DIAGNOSE privileges, and run the .EXE. (These privileges are required, as the access to the CD-related extensions will require the use of the privilege-protected IO$_DIAGNOSE I/O function code.) You can also install the image with these privileges. See the source for additional details - note that the comments regarding the need for SYSGEN CONNECT are no longer applicable (at least as of VMS V5.5-2).

There's also SYS$EXAMPLES:CDROM_AUDIO.C and .EXE, a non-Motif program, available on OpenVMS VAX, and DECW$EXAMPLES:DECW$CDPLAYER.* on OpenVMS VAX and OpenVMS Alpha.

The standard OpenVMS ATA (IDE) SYS$DQDRIVER device driver does not support the necessary does not support the necessary IO$_DIAGNOSE function code that is required for access to audio CD media commands (on OpenVMS versions prior to V7.3), but an updated SYS$DQDRIVER device driver (source code and all) with this capability and with the source code of an updated DECW$CDPLAYER CD audio player is available on the OpenVMS Freeware website (www.hp.com/go/openvms/freeware/, look for the directory /dqdriver/), and these updates are also included on OpenVMS Freeware V5.0, and OpenVMS ECO kits containing newer versions of the driver are available. Freeware V6.0 has a version of DQDRIVER that is newer than that of the OpenVMS Alpha V7.3-2 release, with additional capabilities and with improved error diagnostics.

OpenVMS Alpha V7.3 and later include a version of SYS$DQDRIVER with the necessary IO$_DIAGNOSE support.

7.2 How do I access a Microsoft Windows floppy disk from OpenVMS?

The HP Advanced Server (formerly known as PATHWORKS) for OpenVMS product includes an unsupported and undocumented utility called PCDISK, and this tool can read and write various Microsoft MS-DOS and Microsoft Windows FAT-format diskettes, and can usually access FAT-format volumes written by other operating systems.

ProGIS in Germany sells a product called VMove which supports DOS files on many different device types. For more information, send mail to info@progis.de.

Engineering Software has a product called VAKSAT which will read, write, and erase files on MS-DOS FAT diskettes. Available for both VAX and Alpha. Contact ed@cityscape.co.uk for more information.

MadGoat PC Exchange (PCX) is a utility for copying files to and from MS-DOS and Microsoft Windows (FAT) format diskettes under OpenVMS, using an RX23 (3.5"), RX26 (3.5"), or RX33 (5.25") diskette drive. For 3.5" diskettes, high-density disks can be read or written; double-density disks are read-only. Only high-density disks are supported on the RX33.

- http://www.madgoat.com/

The Freeware package WINFX is available on Freeware V6.0, and can read the FAT volume structure.

- http://www.hp.com/go/openvms/freeware/

7.3 How do I play sound files on an AlphaStation? DECsound doesn't work.

Various of the more recent AlphaStation systems use a different sound board (Microsoft Sound System) than the earlier DEC 3000 series systems, and DECsound, as supplied by DECwindows Motif, doesn't support this board nor this interface. HP offers an optional product, Multimedia Services (MMOV) for OpenVMS:

- http://h18000.www1.hp.com/info/spd/
  OpenVMS typically uses SPD 25.01.xx, SPD 41.87.xx, and SPD 82.35.xx.

which provides a replacement for DECsound for this card as well as many other features (an AVI and MPEG player, video capture support, etc.)

Ensoniq sound support is also available.

7.4 How do I read IBM EBCDIC tapes on OpenVMS?

Most (all?) IBM EBCDIC-based systems can read and write ANSI-labeled ASCII magtapes. Fixed-length records (MOUNT /FOREIGN /BLOCKSIZE=512 /RECORDSIZE=512, for one-block records) and the DCL COPY command can be used to transfer fixed-record-size text files out onto tape media, or to read from fixed-record tape media. Please consult the IBM documentation for the details and command syntax needed when reading and writing ANSI media using IBM JCL or other applicable IBM command language.

There exists various freeware around (TAPECOPY, ETAPE, TCOPY, MTEXCH) that can read and write EBCDIC tapes. Visit the Encompasserve (DECUS) website software archives search engine and search for ‘‘EBCDIC’’ for details.

- http://www.encompassus.org

OpenVMS does not include an integrated tool for EBCDIC tape processing, but does provide a character conversion API useful within application programs.

One source for ETAPE is:

- http://www.ualr.edu/ftp/vms/ETAPE_SRC/

The OpenVMS Freeware V5.0 distribution included this ETAPE tool, as well.

7.5 How can I patch an OpenVMS Alpha image?

Using the OpenVMS Freeware tool ZAP:

- Look for the RMS_TOOLS directory on Freeware V5.0: http://www.hp.com/go/openvms/freeware/

tell ZAP to read a block (bucket) of information based on the virtual block number (VBN), using X for hexadecimal. Dump yourself into the OpenVMS debugger with R2 pointing into the buffer, EXAMINE/INSTRUCTION as needed, alter the buffer as required, GO to get out of the debugger and back into ZAP, and use the ZAP W command to write the updated block.

8 DCL Details

If you are searching for something here,
please consider using the text-format FAQ.

8.1 DCL Symbols and OpenVMS Logical Names?

DCL symbols are programming-style variables implemented within the DCL command interpreter, and these are used both for programming and to provide command verb synonyms. Symbols are local to the command interpreter operating within a particular process, and are not shared. Lists of symbols can be copied into subprocesses during a subprocess creation operation, but these symbols are neither copied back into the parent process when the subprocess exits, nor are symbols ever shared across processes.

Symbols can be specified in and utilized in basic mathematical operations, and bit-level operations are available with the f$cvsi and f$cvui bit extraction lexical functions, and with the square-brackets notation for bit insertion (see Section 8.13 for an example), and with bitwise operators. Symbols are of two basic types, STRING and INTEGER, and these (or an undefined symbol) can be differentiated with the f$type lexical function. DCL symbols can also be used as a mechanism to abbreviate a DCL command verb, or an easy way to invoke a DCL command procedure.

Symbols can have local or global scope within a process, and scope is affected by nested procedure calls and DCL constructs such as CALL and SET SCOPE, but such discussions are beyond the scope of this section.

OpenVMS Logical names can store device names, device and directory specifications, rooted or searchlist specifications, and full filenames. Logical names can also store arbitrary data, but there are no native mathematical or bitwise operators available. Analogous to DCL symbols, process-local logical names can be copied into subprocesses during a subprocess creation operation, but these process-local logical names are neither copied back into the parent process when the subprocess exits, nor are these logical names ever shared.

Logical names are implemented deep within the OpenVMS executive, and are organized into logical name tables. Logical names can be stored in tables private to a process(LNM$PROCESS, the process-local logical name table) , that can be shared among processes in the same job tree (LNM$JOB, the job logical name table) or in logical name tables that are shared among larger groups of processes (eg: LNM$GROUP, the UIC group logical name table and LNM$SYSTEM, the system-wide logical name table). Logical names are centrally intended to provide various I/O-related capabilities, including device independence and configuration customization—correctly-written application programs can use logical names to avoid embedding specific device or device and directory specifications, and to allow filename and configuration customizations.

One of the most powerful capabilities of logical names beyond the device independence provided involves the defaulting capabilities; you can use RMS parsing (directly, or with mechanisms such as the f$parse lexical function) to provide a filename and a default filename. To provide the mechanism that allows SYSUAF to be located in an arbitrary position or even an arbitrary filename, a construct similar to the following is used:

```
$ UAF = F$PARSE("SYSUAF","SYS$SYSTEM:.DAT")
```

This design allows the logical name SYSUAF to be optionally defined, and – when present—to specify the particular location and name of the file. Portions of the full file specification that are omitted are retrieved using the default translation of SYS$SYSTEM: and the file type of .DAT.

Logical names also have assigned processor modes, as some translations must be trustworthy. In the example above, only trusted and privileged system users should be able to redirect the SYSUAF authorization database, so any definition of the SYSUAF logical name must be made in EXECUTIVE mode in a trusted logical name table.

As for common OpenVMS terminology, logical names are ‘‘defined’’ and the associated processing is refered to as ‘‘translation’’, while symbols are ‘‘equated’’ and the associated processing is refered to as ‘‘substitution’’. ‘‘Lexical functions’’ are processing routines built into DCL, and typically prefixed with f$. Many of the lexical functions are built upon correspondingly-named system services, though not all.

Symbol substitution occurs only when the DCL command interpreter is reading and processing the command input; for information on DCL symbol substitution, see Section 8.10. For program access, see the RTL routines lib$set_symbol and lib$get_symbol.)

For information on logical name translation, please see f$trnlnm lexical function and the DCL commands DEFINE and DEASSIGN, as well as underlying system services such as sys$trnlnm. Logical name translation occurs when requested, or as the file or I/O operation is started.

Please see the OpenVMS User's Guide in the OpenVMS documentation set for a far more detailed description of these constructs.

For related materials, please see Section 8.10 and Section 8.11.

8.2 How do I run a program with arguments?

The RUN command does not accept arguments. To pass arguments to a program, you must use what is called a ‘‘foreign command’’, and either an explicit command as shown here, or an automatic foreign command. For example:

```
$ unzip :== $disk:[dir]unzip.exe
$ unzip -?
```

The leading $ in the equivilence name for the symbol definition is what makes the DCL symbol a foreign command. If the device and directory are omitted, SYS$SYSTEM: is assumed.

Under OpenVMS V6.2 and later, DCL supports automatic foreign command definition via the logical name DCL$PATH. An example of a definition of this logical name is:

```
$ DEFINE DCL$PATH SYS$DISK:[],ddcu:[mytooldir],SYS$SYSTEM:
```

DCL will first look for a command in the DCL command table, and if no match is found and if DCL$PATH is defined, it will then look for command procedures and executable images with filenames matching the command specified, in the directories specified via DCL$PATH. The first match found is invoked, and under OpenVMS, the DCL$PATH support will cause a command procedure to be activated in preference to an executable image.

For more information on foreign commands or on automatic foreign command support, see the *OpenVMS User's Manual*.

See also Section 10.3.

If you want to create a detached process that takes arguments from a command line, it must be run under the control of a command line interpreter (CLI) (typically DCL). This is done by placing the command line in a file, specifying SYS$SYSTEM:LOGINOUT.EXE as the image to run and the command file as the input. For example:

```
$ OPEN/WRITE CMD TEMP_INPUT.COM
$ WRITE CMD "$ MYCOMMAND arguments"
$ CLOSE CMD
$ RUN/DETACHED SYS$SYSTEM:LOGINOUT /INPUT=TEMP_INPUT.COM
```

Various OpenVMS library calls (such as lib$spawn( ), cli$dcl_parse( ), and the C library system( ) call) require access to a command line interpreter such as DCL to perform requested actions, and will not operate if a CLI is not available.

When a CLI is not available, these calls typically return the error status SS$_NOCLI. And as mentioned above, invoke the image LOGINOUT to cause a CLI (such as DCL) to be mapped into and made available in the context of the target process.

For examples of how TCP/IP Services sets up its foreign commands (which includes tools such as uuencode and uudecode), please see the DCL command procedure SYS$STARTUP:TCPIP$DEFINE_COMMANDS.COM.

Also see Section 8.12.

8.3 How can I clear the screen in DCL?

The simplest way is the TYPE/PAGE NLA0: command.

You can set up a symbol to clear the screen in your LOGIN.COM:

```
$ CLS :== TYPE/PAGE NLA0:
```

8.4 Using REPLY/LOG from DCL? Disabling Console OPCOMs?

Your terminal must be enabled as an operator terminal before the REPLY/LOG command can be used, but a DCL procedure (batch command file, system startup, etc) does not have an associated terminal. To make this work, use the following sequence to enable the OPA0: console as the operator terminal, then the REPLY/LOG command will be accepted:

```
$ DEFINE/USER SYS$COMMAND _OPA0:
$ REPLY/LOG
$ DEFINE/USER SYS$COMMAND _OPA0:
$ REPLY/ENABLE
```

To disable the system console terminal (OPA0:) as an operator terminal, use the following command:

```
$ DEFINE/USER SYS$COMMAND _OPA0:
$ REPLY/DISABLE
```

Also see SYLOGICALS.COM (and SYLOGICALS.TEMPLATE) for information on configuring the behaviour of OPCOM, including the (default) use of the system console (OPA0:) as an operator terminial and the specific contents and behaviour of the system operator log file OPERATOR.LOG.

8.5 How do I generate a random number in DCL?

With V7.3-2 and later, f$unique can be useful here. Alternatively, here is a pseudo-random number generator, just do a GOSUB RAND and the global symbol RANDOM will contain a randomly generated number. You can feed the generator a ceiling value (__CEIL) or a new seed (__SEED).

```
$! RAND - returns a positive random number ("RANDOM") between 0 and
$!        __CEIL - 1.
$! sharris-at-sdsdmvax.fb3.noaa.gov
$ RAND:
$
$ IF F$TYPE(__SEED) .EQS. ""
$ THEN
$     ! seed the random number generator, ...
$     __NOW = F$CVTIME()
$     __HOUR = ’F$EXTRACT(11,2,__NOW)’
$     __MINUTE = ’F$EXTRACT(14,2,__NOW)’
$     __SECOND = ’F$EXTRACT(17,2,__NOW)’
$     __TICK = ’F$EXTRACT(20,2,__NOW)’
$
$     __SEED == __TICK + (100 * __SECOND) + (6000 * __MINUTE) + -
        (360000 * __HOUR)
$     ! the generator tends to do better with a large, odd seed, ...
$     __SEED == (__SEED .OR. 1)
$     ! clean up, ...
$     DELETEX/SYMBOL __NOW
$     DELETEX/SYMBOL __HOUR
$     DELETEX/SYMBOL __MINUTE
$     DELETEX/SYMBOL __SECOND
$     DELETEX/SYMBOL __TICK
$ ENDIF
$
$ IF F$TYPE(__CEIL) .EQS. "" THEN __CEIL = %X3FFFFFFF
$
$ __SEED == __SEED * 69069 + 1
$
$ RANDOM == (__SEED.AND.%X3FFFFFFF)/(%X40000000/__CEIL)
$
$ RETURN
```

8.6 What does the MCR command do?

The MCR is an artifact of RSX compatibility mode, the operating system from which OpenVMS is descended. MCR is the Monitor Console Routine, and the command is intended to activate RSX compatibility mode utilities. When used on OpenVMS, the command is most commonly used to run the specified image and—because the tool detects the image is not a compatibility-mode image—it acts as a form of RUN command with the default file specification of SYS$SYSTEM:.EXE. MCR passes any (optional) command line arguments in a fashion similar to a foreign command. In other words:

```
$ MCR FOO BAR
```

is equivalent to:

```
 $ FOO :== $FOO
 $ FOO BAR
```

MCR is not documented. Use of a foreign command or the DCL$PATH mechanism is preferred. For details on this, see Section 8.2.

8.7 How do I change the OpenVMS system prompt?

You can use the SET PROMPT command for this purpose. SET PROMPT sets the DCL prompt to the specified string.

When you want to display variable information, you will need to establish a tie-in that provides the information to the SET PROMPT command as required.

If you wish to display the default directory for instance, you will have to establish a tie between the SET DEFAULT command and the SET PROMPT commands, as there is no direct way to get the default directory as the DCL prompt. You can easily acquire or create a set of DCL command procedures that perform the SET DEFAULT and SET PROMPT for you. These DCL command procedures often use a command such as:

```
$ set prompt='f$environment("default")'
```

More advanced users could implement a system service or other intercept, and use these tools to intercept the directory change and reset the prompt accordingly. (This approach likely involves some kernel-mode programming, and requires write access to various undocumented OpenVMS data structures.)

There are related tools available from various sources, including the following web sites:

- ftp://ftp.hhs.dk/pub/vms/setpmt/
- ftp://ftp.tmesis.com/sys_service_hook.src
- James F. Duff has also made available a Macro32 tool known as TIME_PROMPT, a tool that sets the prompt to the current system time.
- Many folks have contributed DCL procedures to perform this task. Visit the newsgroup archives for information and examples.

8.8 Can I do DECnet task-to-task communication with DCL?

Yes, you can do this with DCL.

The OpenVMS DECnet documentation shows various simple examples using the task object and the TYPE command to trigger the execution of a DCL command procedure on a remote node. An example DCL command procedure that is rather more advanced than using the TYPE command as a trigger is included in the Ask The Wizard area:

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

DCL does not include support asynchronous I/O, thus a predetermined protocol or a predetermined "turn-around" command sequence must be implemented in order to avoid protocol deadlocks—cases where both tasks are trying to write or both tasks are trying to read. The task that is writing messages to the network must write (or write and read) a predetermined sequence of messages, or it must write a message that tells the reader that it can now start writing messages. (This is the essence of a basic half-duplex network protocol scheme.)

8.9 How can I get the width setting of a terminal?

```
$ width = f$getdvi(terminal,"DEVBUFSIZ")
```

8.10 Why doesn't DCL symbol substitution work?

The DCL symbol substitution processing occurs only at the DCL prompt, not within data and not within files. If you wish to perform symbol substitution in this environment, you typically write a small file containing the command(s) and data to be invoked—potentially only the data—and you then invoke the created procedure or reference the specified data.

In this case, use of a file containing nolinemode commands or other techniques might be useful—you will want to ensure that the text editor you use does not attempt to use screen mode or similar, as this is not generally considered adventageous within a command procedure.

Tools such as FTP have alternatives: COPY/FTP.

DCL symbol substitution occurs in two passes, using the ampersand and the apostrophe. In most cases, only the apostrophe is necessary. In a few cases—such as the DCL PIPE command—you will may need to use the ampersand to get the substitution to work. The following example uses ampersand substitution to transfer the contents of the header into a logical name:

```
$ PIPE CC/VERSION | (READ SYS$PIPE hdr ; DEFINE/JOB/NOLOG hdr &hdr )
```

A logical name (in the job logical name table; shared by all processes in the current job) was used as DCL symbols cannot be returned back out from a DCL PIPE or other spawned subprocess.

For related materials, please see Section 8.1 and Section 8.11.

8.11 How can I substitute symbols in a PIPE?

Use DCL ampersand substitution, and not apostrophe substitution.

```
$ pipe show system | search sys$input opcom | (read sys$input pid ;
    pid=f$element(0," ",pid) ; define/system opcom_pid &pid)
$ show log opcom_pid
    "OPCOM_PID" = "0000020B" (LNM$SYSTEM_TABLE)
```

8.12 Use of RUN/DETACH, LOGINOUT, and logical names?

With a command to create a detached process such as:

```
$ RUN/DETACHED SYS$SYSTEM:LOGINOUT /INPUT=TEMP_INPUT.COM
```

If you are trying to use a logical name as the /INPUT, /OUTPUT or /ERROR on a RUN/DETACH command, then you must translate the logical name specifications to physical references before passing them, or the definitions must reside in a logical name table that is visible to the newly-created process.

Also note that LOGINOUT only creates the SYS$LOGIN, SYS$LOGIN_DEVICE, and SYS$SCRATCH logical names if it is processing a login that is based on the contents of a SYSUAF record—without access to the associated SYSUAF record, this information is not available to LOGINOUT. (If you want to see these particular logical names created, then please specify the /AUTHORIZE qualifier on the RUN/DETACHED command.)

If you do not specify LOGINOUT as the image, then there is no easy way to get these logical names. Also, any logical names that are used in the target image file specification must also be in a logical name table accessible (by default) by the newly-created detached process. Shared tables include the group (if the process is in the same UIC group) and the system table. (If the target process is to be in another UIC group, a suitablly privileged user or application can create the necessary logical name(s) directly in the other group logical name table.)

When in doubt, create a short DCL command file as input, and use a SHOW LOGICAL and similar commands to examine the context. (And use physical device and directory references on the RUN/DETACH of the LOGINOUT image, when specifying this command file as /INPUT.) Also remember to check both security auditing and system accounting when troubleshooting problems with the RUN/DETACH.

Also see Section 8.2.

8.13 How to use escape and control characters in DCL?

To write a message and then the bell character, use:

```
$ bell[0,7] = 7
$ write sys$output "Hello''bell'"
```

To write blinking text, use:

```
$ esc[0,7] = 27
$ text = "Blinking Text"
$ write sys$output "’’esc’[5m’’text’’’esc’[m"
```

Also see sections Section 11.6, Section 12.1.

9 Files

If you are searching for something here,
please consider using the text-format FAQ.

9.1 How can I undelete a file?

OpenVMS doesn't have an ‘‘undelete’’ function. However, if you are quick to write-protect the disk or if you can guarantee that no new files get created or existing files extended, your data is still on the disk and it may be possible to retrieve it. The FLORIAN tool available from various websites can potentially recover the file, see question Section 13.1 for pointers. Other alternatives here include the DFU tool, available on the OpenVMS Freeware CD-ROM distribution.

If you are setting up a user environment for yourself or for others, it is quite easy to use DCL to intercept the DELETE command, using a symbol:

```
$ DEL*ETE :== @SYS$LOGIN:MYDELETE.COM
```

The DELETE symbol will cause the procedure to be invoked whenever the user enters the DELETE command, and it can copy the file(s) to a ‘‘trashcan’’ subdirectory before issuing a ‘‘real’’ DELETE on the files. Other procedures can retrieve the file(s) from the ‘‘trashcan’’ subdirectory, and can (and should) clean out the ‘‘trashcan’’ as appropriate. (Realize that this DELETE symbol can interfere with DELETE/GLOBAL and other similar DCL commands.)

9.2 Why does SHOW QUOTA give a different answer than DIR/SIZE?

DIRECTORY/SIZE doesn't take into account the size of file headers which are charged to your quota. Also, unless you use DIRECTORY/SIZE:ALL, you will see only the ‘‘used’’ size of the file, not the allocated size which is what gets charged against your quota. Also, you may have files in other directories.

```
$ DIRECTORY/SIZE=ALL/GRAND [username...]
Grand total of D1 directories, F1 files, B1/B2 blocks.
$ DIRECTORY/SIZZ=ALL/GRAND [-]username.DIR
Grand total of 1 directory, 1 file, B3/B4 blocks.
$ SHOW QUOTA
User [username] has B5 blocks used, B6 available
of B7 authorized and permitted overdraft of B8 blocks on disk
```

If the user has no files in other directories and all file-headers are only 1 block, then the following should apply:

```
  B5=B2+B4+F1+1
```

If the diskquota has drifted out of synchronization, then the system-manager can force a quota rebuild—due to various factors, the quota file can potentially drift from the actual use over time, and a periodic rebuild can be performed at appropriate intervals.

Also be aware that the DIRECTORY/SIZE command can report larger values than might otherwise be expected when used to evaluate files and/or directories that are alias links—such as the system roots on OpenVMS system disks—as the command reports a total that is cumulative over all of the files and directories examined, without regard for which ones might be alias entries and which are not. (In other words, a DIRECTORY/SIZE of an entire OpenVMS system disk will report a disk useage value larger than the (usually more accurate) value reported by the SHOW DEVICE command. This as a result of the alias entries linking each SYS$SYSDEVICE:[SYSCOMMON]SYS*.DIR directory file and the SYS$SYSDEVICE:[000000]VMS$COMMON.DIR file together.)

9.3 How do I make sure that my data is safely written to disk?

If your application must absolutely guarantee that data is available, no matter what, there’s really no substitute for RMS Journaling and host- or controller-based shadowing. However, you can achieve a good degree of data integrity by issuing a SYS$FLUSH RMS call at appropriate times (if you’re using RMS, that is.) If you’re using a high-level language’s I/O system, check that language’s documentation to see if you can access the RMS control blocks for the open file. In C you can use fflush followed by fsync.

For details on disk bad block handling on MSCP and on SCSI disk devices, please see Ask The Wizard (ATW) topic (6926).

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

9.4 What are the limits on file specifications and directories?

A file specification has an aggregate maximum size of 255 characters (NAM$C_MAXRSS) at present, assuming ODS-2 limits and traditional DCL process parsing settings (SET PROCESS/PARSE_STYLE). The node and device specification may be up to 255 characters each—file name and file types may be up to 39 characters each. File versions are from 1 through 32767, though 0 (latest version), -0 (oldest version) and -n (n'th previous version) can be used in most contexts. A file specification may not have more than 8 directories and subdirectories or—with a rooted directory, two sets of eight are possible—and while it is possible to create subdirectories of greater depth, accessing them under ODS-2 is somewhat problematic in most cases, and thus should be avoided.

Under ODS-5 with extended DCL parsing (SET PROCESS/PARSE_STYLE), the filename length limits are up around 4,095 (NAML$C_MAXRSS) characters, and directories can be around 255 levels deep.

Application developers should use OpenVMS-supplied routines for parsing file specifications - this ensures that changes in what is allowable will not tend to break your application. Consider that various parts of the file specification may contain quoted strings with embedded spaces and other punctuation! Some routines of interest are SYS$FILESCAN, SYS$PARSE and LIB$TRIM_FILESPEC. For further information, see the OpenVMS Guide to File Applications.

Performance of larger directory files improves (greatly) with OpenVMS V7.2 and later—operations on directory files of 128 blocks and larger were rather slower on earlier OpenVMS releases due to the smaller size of the directory cache and due to the directory I/O processing logic.

For fastest directory deletions, consider a reverse deletion—delete from the last file in the directory to the first. This reversal speeds the deletion operation by avoiding unnecessary directory I/O operations as the files are deleted. Tools such as the Freeware DFU can be used for this purpose, as can various available reverse-DELETE DCL command procedures.

Also see Section 5.44.

9.5 What is the largest disk volume size OpenVMS can access?

One Terabyte (TB; 2**31 blocks of 2**9 bytes; 0x07FFFFFFF blocks). 255 volumes in a volume set. The largest contiguous allocation possible for any particular file is 0x03FFFFFFF blocks.

Prior to the release of V6.0, the OpenVMS file system was limited to disk volumes of 8.38 GB (2**24 blocks, 16777216 blocks) or less.

On some systems, there are restrictions in the console program that limit the size of the OpenVMS system disk. Note that data disks are not affected by console program limits. For example, all members of the VAXstation 3100 series are limited to a system disk to 1.073 GB or less due to the console, though larger data disks are possible. This limit due to the SCSI drivers used by and built into the console ROM to read the OpenVMS bootstrap files, and these same drivers are also used by OpenVMS to write the system crashdump.

There are numerous discussions of this VAXstation 3100 in the comp.os.vms newsgroup archives. Please use Google newsgroup search to search the archives for further details, for discussions of the workarounds, and for details of the potential for a simple failed bootstrap and particularly for discussions of the potential for severe system disk corruptions on crashes.

Some SCSI disks with capacities larger than 8.58 gigabytes (GB) will require the use of an OpenVMS ECO kit (eg: ALPSCSI04_062 or later; see Section 14.25 for details) for new SCSI device drivers. Failure to use this ECO can cause "rounding errors" on the SCSI disk device capacity—OpenVMS will not use nor display the full capacity of the drive—and "%sysinit-e-error mounting system device status equals 000008C4" (8C4 -> "%SYSTEM-?-FILESTRUCT, unsupported file structure level") errors during bootstrap. (One workaround for the bootstrap when the bitmap is located far into the disk is the use of INIT/INDEX=BEGIN.) The problem here involves the particular extensions and fields used for larger capacity disks within the SCSI specifications and within the various intepretations of same.

For ATA (IDE) disk drives:

- Versions of SYS$DQDRIVER *BEFORE* X-15 topped out at 8.455 GB.

  Fixed drivers (equal or greater than "X-15") were shipped in:

  - OpenVMS Alpha V7.2-1, and later
  - V7.2 UPDATE V1.0 ECO, and later

  - V7.1-2 UPDATE V1.0 ECO, and later
  - V7.1-2 UPDATE V3.0 ECO, and later
- The newer SYS$DQDRIVER driver operates to disks up to 33 GB without (known) problems, and effectively works with rather larger disks (up to circa 137 GB) but is known to report an incorrect number of "cylinders" with disks above 33 GB.

See Section 14.4.4.2 for additional ATA SYS$DQDRIVER information.

Be aware that a known restriction in certain older versions of the Alpha SRM Console prevents booting most ATA (IDE) drives larger than 8.455 GB, depending on exactly where the various files are located on the volume. Updated SRM consoles for systems with SRM and ATA (IDE) drive support are (will be) available. (OpenVMS Engineering has successfully bootstrapped 20GB ATA (IDE) disks using the appropriate SRM console version.)

Note

All disk-related listed in this section are stated in units of "disk (base ten) gigabytes" (1 GB = 10^9 bytes) and *not* in units of "software (base two) gigabytes" (1 GB = 2^30; 1 GB = 1073741824.) bytes. Please see Section 14.25 for details of the nomenclature and of the units.

Be aware that larger disks that are using an extension of SCSI-2— disks that are using a mode page field that the SCSI-2 specifications normally reserved for tape devices—to permit a larger disk volume size will require a SCSI driver update for OpenVMS, and this change is part of V7.1-2 and later, and also part of ALPSCSI07_062 and later. (These larger disks disks will typically report a DRVERR, or will see the volume size "rounded down".) SCSI disks larger than 16777216 blocks cira 8.455 GB (base ten); 8GB (base two) require this ECO, or require the use of OpenVMS Alpha V7.1-2 or later.

Applications written in C can be limited to file sizes of two gigabytes and less, as a result of the use of longword values within C file operations, and specifically off_t. This restriction is lifted in OpenVMS V7.3-1 and later, and with the application of the C ECO kits available for specific earlier releases. The use of a longword for off_t restricts applications using native C I/O to file sizes of two gigabytes or less, or these applications must use native RMS or XQP calls for specific operations.

Also see Section 14.13, Section 14.25.

9.6 What is the maximum file size, and the RMS record size limit?

RMS can store individual files of a size up to the maximum supported volume size. Under OpenVMS V6.0 and later, the volume size and the RMS maximum file size limit is 2**31 * 512 bytes—one terabyte (1 TB).

‘‘Use a volume set to provide a large, homogeneous public file space. You must use a volume set to create files that are larger than a single physical disk volume. (The file system attempts to balance the load on the volume sets, for example, by creating new files on the volume that is the least full at the time.)’’

‘‘You can add volumes to an existing volume set at any time. The maximum number of volumes in a volume set is 255.’’ Further, with a 255 member bound-volume set, the theoretical maximum limit of files is 4,261,478,145 files, less the directories and reserved files.

The RMS formats—sequential, relative, and indexed—are limited by the one terabyte maximum volume size. RMS relative files are further limited to a number of records that will fit in 32 bits—4 billion records. Sequential and indexed formats do not have a record limit.

Also see Section 2.17.1, Section 14.25.

9.7 How do I write CD-Recordable or DVD media on OpenVMS?

How to create CD-R, CD-RW, DVD-R, DVD+R, DVD-RW, or DVD+RW media on OpenVMS?

For information on CD and DVD optical media drives on OpenVMS, please see Section 14.29. For information on the creation of OpenVMS media and of OpenVMS bootable media, a full step-by-step sequence is documented in the OpenVMS Ask The Wizard topic (9820). An abbreviated version of the sequence is included here.

Recording (writing) of CD and DVD optical media requires a recording or media mastering application or tool, and both commercial and non-commercial options are available. For OpenVMS V8.3 and later, see the COPY/RECORDABLE_MEDIA command available within OpenVMS itself. Alternatively, please see CDRECORD (both non-DVD and DVD versions are available, and at least one commercial version is available), and also see DVDwrite (commercial) or DVDRECORD (open source). A port of CDRECORD is present in OpenVMS V7.3-1 and later.

- Acquire a comparatively recent SCSI-based or ATAPI (IDE) CD-R or DVD-R/RW or DVD+R/RW drive. *Older drives can be very* *problematic, while newer drives are readily available, and are cheap and very fast, and tend to have better compliance with current standards. Use of older drives is not recommended.* Related device requirements information is available in Section 14.29.

- Get the most recent LDDRIVER available on the Freeware, or activate and use the LD version latent in OpenVMS Alpha V7.3-1 and V7.3-2 by loading the LD command verb (look within SYS$MANAGER:CDRECORD.COM for related details), or use the integrated LD found in OpenVMS V8.2 and later.

  In particular, you will want to use the current ECO kit for LDDRIVER (as available), or the version of LD distributed with V8.2. The OpenVMS V8.2 version of LDDRIVER was also kitted on Freeware V7.0 as LD071.

  If you are not running OpenVMS V8.2, the specified LD071 kit or later, or a current ECO with the update, you will want to upgrade, or you will want to use the DCL command:

  ```
  SET FILE/CACHING_ATTRIBUTES=NO_CACHING
  ```

on the LD partition file. This is a workaround for an incompatibility found between older LDDRIVER versions and the XFC caching support.

  As an alternative to LD and LDDRIVER, you can acquire and load the VD64 package from the Freeware.

- Use the COPY/RECORDABLE_MEDIA command available within OpenVMS V8.3 and later, or use the version of CDRECORD built in V7.3-1 or later, or obtain and utilize one of the open source or commercial versions of the CDRTOOLS/CDRECORD or DVDRTOOLS/CDRECORD tools, or other similar recording tools. Freeware V6.0 distribution. (http://www.hp.com/go/openvms/freeware/) An OpenVMS port of the cmcd CD audio ripper is also reportedly available. http://www.amb.org/xmcd/

  Versions of CDRECORD (non-DVD) are latent in OpenVMS Alpha V7.3-1 and later. Commercial versions of CDDRECORD—with DVD capabilities—are also available for various platforms, and particularly a variant of CDRECORD known as CDRECORD-ProDVD. COPY/RECORDABLE_MEDIA is built into OpenVMS V8.3 and later.

  Beware the tool chosen: some versions and configurations of CDRECORD can record DVD media, as can the DVDRECORD package, as can the commercial DVDwrite package. Many versions of CDRECORD cannot record DVD media, including the version of CDRECORD latent within OpenVMS and the version found on Freeware V6.0; these versions cannot record DVD media.

- Build the contents of the disk on the LD or VD64 device partition.
- Use the chosen recording tool to record the contents of the LD or VD64 partition directly onto the optical medium.

Alternatively, consider the following command on OpenVMS Alpha V7.3-1 and later:

```
@SYS$MANAGER:CDRECORD.COM HELP
```

While folks have had success getting PC-based CD-R/RW or DVD-R/RW or DVD+R/RW tools to work with OpenVMS partitions, it is far easier and more reliable to use the OpenVMS-based versions of these tools and directly-attached devices. If you use a Windows-based tool, you will want to specifically select its raw mode, image mode, or block-copy mode, depending on the terminology within the particular tool. The transfer mode and selections is variously refered to as a disk-at-once (DAO) 2048-byte block ISO Mode 1 raw/image/block data disk recording mode.

More details: Creation of CD recordable or DVD recordable media under OpenVMS typically involves one of two approaches: the use of the optional CD-R (‘Scribe’) capabilities available for the InfoServer or other ‘‘offline’’ hardware packages (PC-based packages will be included in this), or the use of a host-based package such as the CDRECORD or COPY/RECORDABLE_MEDIA (V8.3 and later) or other utilities, including OpenVMS ports of common open-source tools made available by Dr. Eberhard Heuser-Hofmann and various others. Commercial packages and options are also available. Dr. Heuser-Hofmann has DVDwrite , a commercial package which can record DVD media. (http://home.tiscali.de/dvd4openvms)

OpenVMS can read ODS-2, ODS-5, and ISO-9660 format CD-ROMs and DVD-ROMs directly. (If you are very careful, you can create a dual-format CD-R; a CD-R with both ODS-2 and ISO-9660 or both ODS-5 and ISO-9660 or both.)

OpenVMS does not support ISO-9660:1999, nor the Joliet or Rock Ridge extensions to ISO-9660, nor can OpenVMS decrypt copy-protected video DVDs.

InfoServer hardware configurations are no longer available from HP, but may potentially be acquired through other means; as used equipment. InfoServer support also has very specific CD-R recording device prerequisites, and these recording devices are no longer generally available.

Packages related to the use of DVD archiving are also available, see the multi-volume capabilities of the DVDarchive/restore Freeware.

`http://www.geocities.com/SiliconValley/Lakes/9999/vmscdwri.html`

Additional information is available at the following sites:

- http://www.djesys.com/vms/cdrom.html
- http://www.cd-info.com/CDIC/Technology/CD-R/vms.html
- http://www.faqs.org/faqs/cdrom/cd-recordable/part1/preamble.html
- http://www.tmesis.com/CDrom/
- http://www.tditx.com/~odsiso/

U.S. Design offers a package that includes the tools necessary to create a CD or DVD-R with either ISO-9660 or ODS-2 format, for standalone CD-R/RW, DVD-R, or DVD+R/RW drives, for recent OpenVMS versions. Details are available at:

- http://www.usdesign.com/

Also see Section 9.7.2 for details on access to recorded media on older CD-ROM drives.

9.7.1 CD and DVD notation, terminology?

CD-ROM is pre-recorded Compact Disk media, and is the original and oldest CD format. The original CD media was physically stamped, a recording process that is now largely reserved to the highest-volume media reproduction requirements.

CD-R is CD Recordable, a write-once storage medium that can be read by all but the oldest of CD drives; a format which can be read and often even recorded by most CD-RW drives.

CD-RW is CD ReWritable, a format which is readable by many CD drives and by most CD-R drives, and with media that can be recorded and re-recorded by CD-RW drives.

CD media recording speeds are listed as multiples of 150 kilobytes per second, so a 10X drive records at 1500 kilobytes (1.5 megabytes) per second. 600 MB (70 minutes) and 700 MB (80 minutes) recording capacities are both widely available. The minutes designation is derived from the traditional audio-format recording capacity of the particular media.

DVD-R/RW is the older of two common Digital Versatile Disk recording formats, and the DVD-R Recordable or DVD-RW ReWritable media can be read by many DVD drives. As with CD-R formats in older CD drives, older DVD and particularly first-generation DVD players may have problems reading this media format.

DVD+R/RW is the newer of the two common Digital Versatile Disk recording formats, and the DVD+R Recordable or DVD+RW ReWritable media can be read by many DVD drives. Akin to DVD-R/RW media, older and particularly first-generation DVD drives can have problems reading this media format.

The DVD Plus-series drives and media tend to record faster than Minus drives, as (as of this writing) the Plus (+) drives do not require an initial media formatting pass and the Minus (-) drives do. While the appropriate Plus (+) or Minus (-) DVD raw media must be chosen for the particular DVD recorder (and DVD recording drives that are compatible with and capable of using both Plus and Minus media are available), the resulting recorded media is generally readable (playable) in all recent DVD drives and DVD players, regardless of type. (Compatibility is best within the same media-series devices of course, but be certain to verify compatibility across devices regardless of the particular device or particular recording media chosen.)

Presently Plus (+) media is slightly more expensive than Minus (-), but with the prices of all CD and all DVD media continuing to consistently fall, the differences in DVD media costs are becoming irrelevent for all but the production of huge volumes of DVD media.

The rated DVD recording speeds are in multiples of 1353 kilobytes per second, thus a DVD 1X drive is roughly equivalent to a CD 9X drive in I/O requirements and transfer speed.

DVD drive recording speed can and does vary. DVD disk drive recording speed is limited by the rated recording speed of the media used, so the slower (and cheaper) DVD media will not record any more quickly in a faster drive. A 2.4X DVD drive loaded with 1X media will record at 1X.

9.7.2 Use of RRD42 and other older (embossed-media) CD drives?

The RRD42 series SCSI CD-ROM drive is sufficiently old that it can have problems processing CD-R and CD-RW media. Other very old CD drives can have equivalent media compatibility problems when attempting to read (much) newer CD media and newer CD media technologies. These older CD drives are generally intended for use with the so-called embossed media, rather than with non-embossed recorded (recordable) media now in common circulation.

Please consider using a slightly-less-ancient CD-ROM or CD-R or CD-RW drive when working with non-embossed recorded CD media.

To paraphrase one knowledgable—though deliberately nameless—storage engineer, ‘‘The RRD42 drive is just past the drooling idiot stage’’.

9.7.3 Creating Bootable OpenVMS I64 CD or DVD Media? SYS$SETBOOT?

If you are creating a bootable CD or DVD media for use with OpenVMS I64, you will want to specify the SYS$SETBOOT block size of 2048, and you will also want a disk cluster factor that is a multiple of four via INITIALIZE/CLUSTER=4 (or 8, or...), or you will want to ensure that SYS$EFI.SYS and SYS$DIAGNOSTICS.SYS are aligned to a multiple of four blocks; to a 2048 byte boundary. This alignment and this blocking is only necessary for OpenVMS I64, and only when creating optical media OpenVMS I64 for bootstraps.

The default 512-byte block setting used by SYS$SETBOOT is the correct and expected value for traditional disk bootstraps on OpenVMS I64 systems.

Once the boot files are loaded, OpenVMS I64 operates with 512-byte blocks; as is the case with ATAPI disks on OpenVMS Alpha, all application code will only see 512-byte blocks on optical media on OpenVMS I64.

OpenVMS I64 V8.2 and later are expected to have a version of SYS$SETBOOT that will flag a misaligned SYS$EFI.SYS and (if present) a misaligned SYS$DIAGNOSTICS.SYS file.

For information on SYS$SETBOOT and the SET BOOTBLOCK command, please see Section 14.3.9 and see the OpenVMS documentation. The purpose and intent of the SYS$SETBOOT.EXE image and the SET BOOTBLOCK command is analogous to the WRITEBOOT.EXE image on existing OpenVMS VAX and OpenVMS Alpha systems.

For information on CD and DVD optical media drives on OpenVMS, please see Section 14.29. For additional related information on creating bootable OpenVMS media, please see Ask The Wizard topic (9820).

9.8 What I/O transfer size limits exist in OpenVMS?

The maximum transfer size is an attribute of the particular I/O device, controller and driver combination; there is no inherent limit imposed by OpenVMS (other than the fact that, today, byte counts and LBNs are generally limited to 32 bits).

The maximum size of a device I/O request is limited by the value in UCB$L_MAXBCNT, which is set by the device driver based on various factors. (Also check the setting of the MAXBUF system parameter for buffered I/O transfers, and check the process quotas.)

Currently, SCSI drivers limit I/O transfers to FE00(16) bytes, 65024 bytes (decimal). The reasons for this transfer size limitation are largely historical. Similarly, DSSI devices are limited to the same value, this for hardware-specific reasons. Transfers to HSC and HSJ device controllers via the CI are limited to 1,048,576 bytes. Client MSCP-served devices are limited to 65535 bytes—to help ensure that the I/O fragmentation processing happens on the client and not on the server system.

Parts of the OpenVMS I/O subsystem are optimized for data transfers less than 64KB, because (obviously) most I/O operations are (substantially) less than that. OpenVMS can handle larger transfers, if the driver and the device can handle it.

Also see Section 9.4, Section 9.5.

9.9 Can I use ODBC to connect to OpenVMS database files?

Yes, you can use various available third-party packages that permit remote ODBC clients to access RMS files and various commercial databases via the network.

For RMS, consider acquiring one of the packages available from EasySoft, Attunity Connect (formerly known as ISG Navigator), Oracle (DB Integrator), SolutionsIQ, OpenLink Software (OpenLink Universal Data Access), and Synergex.

The unixODBC package available at http://www.unixodbc.org has variously been found to operate on OpenVMS, as well.

For specific commercial databases (other than RMS, of course), contact the database vendor directly for assistance.

9.10 If my disks are shown as VIOC Compatible, am I using XFC?

Yes, you are using XFC caching.

Disks that are using XFC caching use communication and coordination protocols that are compatible with the older VIOC caching implementation. With the initial implementation of XFC on OpenVMS, you can use the command SHOW MEMORY/CACHE to see no disks reported in full XFC mode; all disks shown will be listed in ‘‘VIOC Compatable Mode’’.

If you have the OpenVMS system parameter VCC_FLAGS set to 2 and are using OpenVMS Alpha V7.3-1 or later, or are using OpenVMS Alpha V7.3 with the VMS73_XFC V2.0 ECO kit or later or with the UPDATE kits, you are using XFC.

Another confusion: the XFC product version is and remains V1.0 in all released configurations, please do not confuse the internal XFC product version (displayed by various commands) with the version number associated with the various ECO kit(s). XFC V1.0 does not permit volumes to enter full XFC caching, as displayed by the ‘‘Vols in Full XFC mode’’ portion of the DCL command SHOW MEMORY/CACHE output.

9.11 RMS Sequential Files and Platform Portability?

When working with mixed platforms, you will want to become familiar with the various RMS sequential record formats, including Variable with Fixed Control (VFC), stream, stream LF, and stream CR, among other record formats.

Switching formats uses CONVERT/FDL or SET FILE/ATTRIBUTES. The former converts files, the latter resets attributes. Text editors tend to select attributes when creating new files that may or may not meet requirements. If the default attributes do not match your requirements, create a stub file, SET FILE/ATTR, then edit the existing file. (Most editors will preserve attributes on an existing file.)

When working with Windows, stream is usually the best choice for sequential file operations. Stream LF is most commonly used with UNIX and C applications. Windows and UNIX tend not to be able to directly read files of "unexpected" sequential RMS record formats.

VFC is a common OpenVMS format, encoding the record length into the record. It is this extra data that can cause corruption-like problems when viewed without RMS; either directly via $qio or via the file system API on other operating system platforms. You will want to look at the low-level record formats, and at the RMS and the Files and Applications documentation in the OpenVMS manuals.

If transfering through other platforms, use of a current version of Zip (with the "-Vv" or "-V" option) and unzip, or use of a BACKUP saveset will contain and maintain the RMS file and record attributes. (For BACKUP and its own attributes requirements, see the restoration tool.)

9.12 How to read locked files?

Files can be locked by applications, and various approaches including CONVERT/SHARE and DUMP/ALLOCATED can be used, as can the following command sequence:

```
$ open/read/share=write x lockedfile.txt
$ type x
```

If you can rebuild the application from source, details related to file sharing are in Section 10.17.

10 OpenVMS Programming Information

**If you are searching for something here,
please consider using the text-format FAQ.**

10.1 Modular Programming, Facility Prefixes and Symbol Naming?

Please first review the OpenVMS Programming Concepts Manual and the Guide to Modular Programming manuals. Both are available within the OpenVMS documentation set, and provide details of the expected norms for OpenVMS programs.

- Learn about the facility prefix, and use a the appropriate prefix uniformly throughout all external symbols, all logical names, and all files located in shared directories. The prefix and the use of the dollar sign ($) and the underscore (_) help avoid collisions with other products. Use of the dollar sign is reserved to registered products.
- Please consider use of tools such as the Freeware SDL package, and the GNM package. These permit you to generate include files and message documentation akin to that of OpenVMS, providing users of your product with a familiar environment.
- For product installations, consider use of the PCSI installation utility, and provide a product-specific configuration DCL command procedure (usually SYS$MANAGER:prefix$CONFIG.COM) if configuration is required.
- The product startup file is usually named SYS$STARTUP:prefix$STARTUP.COM, and the shutdown file (if needed) is usually SYS$STARTUP:prefix$SHUTDOWN.COM.

OpenVMS provides a registry for facility prefixes and for MESSAGE message compiler codes. To request a prefix and a message facility code for a product you distributinng to other customer sites, send your request in a mail message addressed to product[-at-sign-]hylndr.sqp.zko.dec.com, requesting the submission form and details of the registration process.

Note

Please do not request facility prefixes for products that local to your business, your site, or your system. Facility prefixes and message codes and the facility registration

> **process are intended solely for HP products and Partner Products (and yes, even OpenVMS Freeware packages) that will be distributed across multiple OpenVMS customer sites.**

For a list of common coding bugs, please see the remainder of this section of the FAQ and specifically Section 10.22, please also see the Ask The Wizard topic (1661), and for information on debugging an OpenVMS application, please see topic (7552).

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

10.2 Can I have a source code example of calling...?

Please use the available SEARCH command on OpenVMS, and please remember to search the available resources, including the support databases and the newsgroup archives. Please also realize that most OpenVMS system services use similar calling sequences, meaning that an example of calling sys$getjpi can be used as an example for sys$getsyi and sys$getdvi. Students: please do not expect folks to write your homework for you. As for search resources:

- SEARCH SYS$EXAMPLES:*.* target
- SEARCH TCPIP$EXAMPLES:*.* target
- http://www.hp.com/go/openvms/freeware/
- http://www.google.com/

OpenVMS programming documentation, including the numerous example programs found in recent versions of the *OpenVMS Programming Concepts* manual, is available:

- http://www.hp.com/go/openvms/doc/

As for details of argument passing, most OpenVMS system services and RTL routines pass string arguments by descriptor. Languages which support native string data types create descriptors automatically; those which do not (eg., C) require that you set them up explicitly. For further details on using descriptors and particularly for using descriptors from C, please see Section 10.13.

There is extensive information available on how to call OpenVMS system services and OpenVMS Run-Time Library routines, including examples in numerous languages. Among the best available references are:

- Your language's User Manual
- *OpenVMS Programming Environment Manual*
- *OpenVMS Programming Concepts Manual*
- *OpenVMS Programming Interfaces: Calling a System Routine*
- *OpenVMS Calling Standard*

In addition, you can also locate answers, source code examples and related discussions in the James Search Assistant (formerly NLSA (Natural Language Search Assistant), itself formerly known as Ask Compaq (AskQ)) database:

- http://www2.itrc.hp.com/service/james/CPQhome.do

In this area, you will find the source code of programming examples for calls to many of the OpenVMS system services (and from various programming languages), including calls to core services sys$getjpi[w], sys$getsyi[w] and sys$qio[w], as well as source code examples for calls to many other system services and run-time library routines, and examples of one of the more difficult calling interfaces found on OpenVMS systems, that of the smg$create_menu routine.

Arne Vajhøj has put together a collection of OpenVMS example programs. It can be found at:

- ftp://ftp.hhs.dk/pub/vms/

Additional information and examples for OpenVMS are available via:

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference. and via:

- http://www.hp.com/go/openvms/freeware/

10.3 How do I get the arguments from the command line?

If you’re writing a program and want to accept arguments from a foreign command, you can use LIB$GET_FOREIGN to get the command line and parse it yourself, or if you’re programming in C, use the normal argc/argv method.

To write an application which uses the normal DCL verb/qualifier/parameter syntax for invocation, see the description of the CLI$ routines in the OpenVMS Callable Utility Routines Reference Manual.

It is possible to write an application which can be used both ways; if a DCL verb isn’t used to invoke the image, the application parses the command line itself. One way to do this is to call CLI$GET_VALUE for a required parameter. If it is not present (or you get an error), call LIB$GET_FOREIGN to get the command line and do the manual parse.

See also Section 8.2.

10.4 How do I get a formatted error message in a variable?

Use the SYS$PUTMSG system service with an action routine that stores the message line(s) in the variable of your choice. Be sure the action routine returns a "false" (low bit clear) function value so that SYS$PUTMSG doesn’t then try to display the message (unless you want it to.) See the description of $PUTMSG in the System Services Reference Manual for an example of using an action routine.

10.5 How do I link against SYS$SYSTEM:SYS.STB on an Alpha system?

LINK/SYSEXE is the OpenVMS Alpha equivalent of linking against SYS.STB. This links against the base image: SYS$BASE_IMAGE.EXE

Also see Section 10.11, particularly for pointers to the details on shareable images and shareable image creation, and see Section 10.22 for details of inner-mode floating point requirements, of data alignment, requirements for use of /NOSYSLIB, and other related inner-mode programming details, and see Section 10.11 for image-related information.

10.6 How do I do a SET DEFAULT from inside a program?

The problem is that SYS$SETDDIR only changes the default directory - NOT the default disk. The default disk is determined by the logical SYS$DISK. If you want to change the default disk within a program, then call LIB$SET_LOGICAL to change the logical SYS$DISK. You will need to call both LIB$SET_LOGICAL and SYS$SETDDIR to change both default disk and the default directory!

10.7 How do I turn my Fortran COMMON into a shareable image on Alpha?

You need to add SYMBOL_VECTOR=(<common-name>=PSECT) to your options file. On OpenVMS VAX all OVR/REL/GBL psects were automatically exported into the shareable image's Global Symbol Table. On OpenVMS Alpha you have to tell the linker that you want this done by means of the PSECT keyword in the SYMBOL_VECTOR options file statement.

This has several advantages over OpenVMS VAX. First, you don't have to worry about the address of the psect when you try to create a new, upwardly compatible version of the shareable image. Second, you can control which psects, if any, are made visible outside the shareable image.

By default, COMMON PSECTs in HP Fortran for OpenVMS Alpha (as well as most other OpenVMS Alpha compilers) are NOSHR. On VAX, the default was SHR which required you to change the attribute to NOSHR if you wanted your COMMON to be in a shareable image but not write-shared by all processes on the system. If you do want write-sharing, use:

```
CDEC$ PSECT common-name=SHR
```

in the Fortran source code (the CDEC$ must be begin in column 1) or a linker options file PSECT_ATTR statement to set the COMMON PSECT attribute to SHR.

For further information, see the Linker manual.

10.8 How do I convert between IEEE and VAX floating data?

In OpenVMS V6.1 and later, the routine CVT$CONVERT_FLOAT is documented in the LIB$ Run-Time Library Reference Manual, and can perform floating point conversions between any two of the following floating datatypes: VAX (F,D,G,H), little-endian IEEE (single, double, quad), big-endian IEEE (single, double, quad), CRAY and IBM System\370, etc.

HP Fortran (all OpenVMS platforms) has a feature which will perform automatic conversion of unformatted data during input or output. See the HP Fortran documentation for information on "non-native data in I/O" and the CONVERT= OPEN statement keyword.

There are floating-point conversion source code packages available for various platforms.

For further floating-point related information, see:

- http://www.hhs.dk/anonymous/pub/vms/collection/ieee.zip

10.9 How do I get the argument count in a Fortran routine?

On VAX, many programmers would use a MACRO routine which accessed the AP register of the caller to get the address of the argument list and hence the argument count. This was not guaranteed to work on VAX, but usually did. However, it doesn't work at all on OpenVMS Alpha, as there is no AP register. On Alpha systems, you must use a language's built-in function to retrieve the argument count, if any. In Fortran this is IARGCOUNT, which is also available in DEC Fortran on OpenVMS VAX.

Note that omitting arguments to Fortran routines is non-standard and is unsupported. It will work in many cases - read the DEC Fortran release notes for additional information.

10.10 How do I get a unique system ID for licensing purposes?

Many software developers desire to use a unique hardware ID to ‘‘lock’’ a given copy of their product to a specific system. Most VAX and Alpha systems do not have a unique hardware-set ‘‘system ID’’ that can be used for this purpose. HP OpenVMS products do not use hardware IDs in the licensing methods, as many users consider a hardware-based licensing scheme to be negative attribute when considering software purchases.

HP OpenVMS uses a software-based system called the License Management Facility (LMF). This provides for software keys (Product Authorization Keys or PAKS) which support capacity and user-based license checking. HP offers an LMF PAK Generator to DSPP (formerly CSA) members—see Section 2.8.3 for general details on the DSPP offerings, and on the DSPP-provided Freeware PAKGEN PAK that is available for those that are interested in details. (Sorry, no; you can't use the Freeware PAKGEN PAK to generate any PAK you might want. Folks at HP also thought of that one, too.)

For information on licensing, please see Section 12.4.

However, if a hardware-based method is required, the most common method is based on an Ethernet adaptor hardware address. Sample source code for implementing this is available at:

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

10.11 What is an executable, shareable, system or UWSS image?

Executable code in OpenVMS typically resides in an image—an image is a file—the file extension is typically .EXE—that contains this code. Common types of images include executable images, shareable images, system images, and protected (UWSS) images.

Executable images are programs that can be directly executed. These images can grant enhanced privileges, with an INSTALL of the image with /PRIVILEGE, or can grant enhanced access with the specification of a subsystem identifier on the ACL associated with the image.

Shareable images contain code executed indirectly, these images are referenced from executable images and/or from other shareable images. These images can not grant enhanced privileges, even with the use of INSTALL with /PRIVILEGE or a subsystem identifier. These shareable images can be dynamically activated (a LINK that occurs at run-time) via the LIB$FIND_IMAGE_SYMBOL run-time library (RTL) routine. (See ‘protected images’ for information on ‘privileged shareable images’.)

System images are intended to run directly on the VAX or Alpha hardware—these are normally used for the kernel code that comprises an operating system.

Protected images—also refered to as User-Written System Services (UWSS), or as privileged shareable images—are similiar in some ways to a standard shareable images, but these images include a ‘change mode’ handler, and execute in an ‘inner’ processor mode (privileged mode; executive or kernel), and code executing in inner modes has implicit SETPRV privilege. Must be INSTALLed with /PROTECT. Note that inner-mode code has restrictions around calling library routines, around calling various system services, and around calling code located in other protected or shareable images.

Loadable images and device drivers are images that can be used to add code into the OpenVMS kernel. Pseudo-device drivers are a particularly convenient way to add executable code, with associated driver-defined data structures, into the kernel. The pseudo-device driver includes the UCB and DDB data structures, and a calling interface with support for both privileged and unprivileged access to the driver code via sys$qio[w] calls.

A cookbook approach to creating OpenVMS shareable images is available at the URL:

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

10.12 How do I do a file copy from a program?

There are several options available for copying files from within a program. Obvious choices include using lib$spawn( ), system( ), sys$sndjbc( ) or sys$creprc( ) to invoke a DCL COPY command. Other common alternatives include using the callable convert routines and the BACKUP application programming interface (V7.1 and later).

10.13 What is a descriptor?

A descriptor is a data structure that describes a string or an array. Each descriptor contains information that describes the type of the data being referenced, the size of the data, and the address of the data. It also includes a description of the storage used for the data, typically static or dynamic. Descriptors are passed by reference.

The following are examples of creating and using descriptors in C, with the use of the angle brackets normally expected by the C include statements deliberately altered in deference to HTML:

```
#include {descrip.h}
#include {lib$routines.h}
#include {stsdef.h}
int RetStat;
char TxtBuf[TXTSIZ]
struct dsc$descriptor StaticDsc =
  { 0, DSC$K_DTYPE_T, DSC$K_CLASS_S, NULL };
struct dsc$descriptor DynDsc =
  { 0, DSC$K_DTYPE_T, DSC$K_CLASS_D, NULL };
int DynDscLen = 255;
$DESCRIPTOR( ConstDsc, "This is a string" );

/* finish setting up a static descriptor */
StaticDsc.dsc$w_length      = TXTSIZ;
StaticDsc.dsc$a_pointer     = (void *) TxtBuf;

/* finish setting up a dynamic descriptor */
RetStat = lib$sget1_dd( &DynDscLen, &DynDsc );
if ( !$VMS_STATUS_SUCCESS( RetStat ) )
  return RetStat;

/* release the dynamic storage */
RetStat = lib$sfree1_dd( &DynDsc );
if (!$VMS_STATUS_SUCCESS( RetStat ))
  return RetStat;
```

Static descriptors reference storage entirely under application program control, and the contents of the descriptor data structure can be modified as required (by the application). OpenVMS routines do not modify the contents of a static descriptor, nor do they alter the address or length values stored in the static descriptor. (The term "static" refers to the descriptor data structure, and not necessarily to the storage referenced by the descriptor.)

Dynamic descriptors reference storage under the control of the run-time library, and the contents of a dynamic descriptor data structure—once initialized—can only be modified under control of run-time library routines. The dynamic storage referenced by the dynamic descriptor is allocated and maintained by the run-time library routines. Various OpenVMS routines do alter the contents of the descriptor data structure, changing the value for the amount and the address of the storage associated with the dynamic descriptor, as required. Routines can obviously access and alter the contents of the storage referenced by the descriptor.

OpenVMS languages that include support for strings or arrays are expected to use descriptors for the particular structure. Most OpenVMS languages, such as Fortran and BASIC, use descriptors entirely transparently. Some, like DEC C, require the programmer to explicitly create and maintain the descriptor.

For further information on string descriptors, see the *OpenVMS Programming Concepts* manual, part of the OpenVMS documentation set.

Fortran defaults to passing integers by reference and characters by descriptor. The following sites discuss mixing Fortran and C source code in the same application:

- http://www.hhs.dk/anonymous/pub/vms/misc/FORTRAN_C_CALL.COM
- ftp://ftp.hhs.dk/pub/vms/misc/FORTRAN_C_CALL.COM

10.14 How do I create a process under another username?

Many server processes can operate within the context of the target user using privileges, using calls such as sys$chkpro and (more commonly in this context) sys$check_access as needed to determine if access would be permitted for the specified user within the current security model.

With OpenVMS V6.2 and later, the persona system services (SYS$PERSONA_*) can be used to assume the persona of the specified user—these allow the server to operate as the specified user, in a controlled fashion. The persona services can be used as a "wrapper" around a sys$creprc process creation call, as well—this will create a seperate process entirely under the assumed persona.

Information on the persona system services is included in the OpenVMS V6.2 new features documentation, and in the OpenVMS V7.1 and later system services documentation. These system services exist and are supported in OpenVMS V6.2 and later releases.

Typical mechanisms for creating a process under another username include:

- personna services around a sys$creprc call. See above.
- via DECnet task-to-task, using explicit specification of username and password, or using a DECnet proxy. This creates a network-mode job under the target user. The network-mode job might do little more than a RUN/DETACH of an image passed in via task-to-task—task-to-task communications are fully available using strictly DCL-to-DCL processing, or using a compiled language and DCL, etc.)

- SUBMIT/USER, or the username argument on the sys$sndjbc call. This creates a batch-mode job under the specified username. The batch-mode job might do little more than a RUN/DETACH of an image passed in via a parameter.
- the UIC argument on the sys$creprc call. This mimics the UIC of the target user, and is certainly not the prefered mechanism for this task.
- Via pseudo-terminals...

There are likely a few other mechanisms around... There are various tools available from DECUS and other sources that allow various forms of user impersonation, as well. These tools will require version-dependent kernel code and enhanced privileges for some of (or all of) their operations.

10.15 Why do lib$spawn, lib$set_symbol fail in detached processes?

The processing within run-time library (RTL) calls such as lib$attach, lib$disable_ctrl, lib$do_command, lib$enable_ctrl, libget_symbol, librun_program, libset_symbol, libset_logical, and lib$spawn, is dependent on and requires the presence of a command language interpreter (CLI), such as DCL. Without a CLI present in the current process, these calls will fail with a "NOCLI, no CLI present to perform function" error.

Detached processes typically do not have a CLI present.

In place of lib$spawn, sys$creprc can often be used. The context of the parent process (symbols and logical names) will not be propogated into the subprocess when sys$creprc is used, though when there is no CLI present in the process this (lack of) propogation is moot.

To create a detached process with a CLI, you must specify LOGINOUT as the target image as discussed elsewhere in the FAQ, or only use these calls (and any other calls requiring a CLI) from images that are running in an "interactive", "batch", or "other" mode process.

Also note that the lib$spawn and the C system call will fail in a CAPTIVE login environment. The lib$spawn call can be gotten to work in this environment with the specification of the TRUSTED flag.

10.16 Where can I obtain Bliss, and the libraries and supporting files?

The Bliss language compilers and documentation are available on the OpenVMS Freeware distributions.

Bliss language source code that contains the following statement:

```
LIBRARY 'SYS$LIBRARY:STARLET.L32';
```

or similar requires the presence of the Bliss libraries. These libraries are created on the target system using the Bliss require files, and are built using the following Bliss commands:

STARLET.L32 contains the public interfaces to OpenVMS:

```
$ BLISS /LIBRARY=SYS$COMMON:[SYSLIB]STARLET.L32 -
    SYS$LIBRARY:STARLET.REQ
```

LIB.L32 contains both the public and private interfaces to OpenVMS:

```
$ BLISS /LIBRARY=SYS$COMMON:[SYSLIB]LIB.L32 -
    SYS$LIBRARY:LIB.REQ+SYS$LIBRARY:STARLET.REQ
```

The equivilent files for Bliss64 are created with:

```
$ BLISS/A64/LIBRARY=SYS$COMMON:[SYSLIB]LIB.L64 -
    SYS$LIBRARY:LIB.R64+STARLET.REQ+STARLET.R64
$ BLISS/A64/LIBRARY=SYS$COMMON:[SYSLIB]STARLET.L64 -
    SYS$LIBRARY:STARLET.R64
```

Some Bliss code may also require the OpenVMS VAX architecture flags. The following is the equivilent of the Alpha ARCH_DEFS.REQ module:

```
!
! This is the OpenVMS VAX version of ARCH_DEFS.REQ, and
! contains the architectural definitions for conditionally
! compiling OpenVMS Bliss sources for use on VAX systems.
! (If you should encounter compilation errors here, please
! seriously consider upgrading your Bliss compiler.)
!
MACRO VAXPAGE = 1%;
MACRO BIGPAGE = 0%;
!
MACRO VAX =                     ! = 1 if compiled BLISS/VAX
        %BLISS(BLISS32V)%;      ! = 0 if not compiled BLISS/VAX

MACRO EVAX =                    ! = 1 if compiled BLISS/E* (Obsolete, old name)
        (%BLISS(BLISS32E) OR %BLISS(BLISS64E))%; ! = 0 if compiled /VAX /Inn

MACRO ALPHA =                   ! = 1 if compiled BLISS/E* (New arch name)
        (%BLISS(BLISS32E) OR %BLISS(BLISS64E))%; ! = 0 if compiled /VAX /Inn

MACRO IA64 =                    ! = 1 if compiled BLISS/I* (New arch name)
        (%BLISS(BLISS32I) OR %BLISS(BLISS64I))%; ! = 0 if compiled /VAX or /Ann
```

```
MACRO ADDRESSBITS =
        %BPADDR%;              ! = 32 or 64 based on compiler used
```

Some Bliss code may require the definition files for the OpenVMS older LIBRTL routine lib$tparse, or the newer lib$table_parse call:

```
$ BLISS /LIBRARY=SYS$COMMON:[SYSLIB]TPAMAC.L32 -
    SYS$LIBRARY:TPAMAC.REQ
```

10.17 How can I open a file for shared access?

When creating a file, it is often useful to allow other applications and utilities—such as TYPE—to share read access to the file. This permits you to examine the contents of a log file, for instance.

A C source example that demonstrates how to do this is available in topic (2867) in the OpenVMS Ask The Wizard area:

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

Depending on the environment, you may need to use C calls such as fsync and fflush, and—in specific cases—the setvbuf(_IONBF) call.

For related materials around commands that can potentially be used to override file locking—to view the file contents as currently written to disk—see Section 9.12.

10.18 How can I have common sources for messages, constants?

Use the GNM tools on the OpenVMS Freeware to have common sources for MSG (message) files and SDML (Document) documentation files. Use the DOCUMENT command to convert the SDML documentation into the necessary formats (Text, Postscript, HTML, etc). Use the MESSAGE/SDL tool (latent in OpenVMS) to create an SDL file based on the messages. Then use the SDL tool (available on the OpenVMS Freeware) to convert the SDL file into language-specific definitions. (There is also a converter around to convert SDL into SDML, if you want to get pictures of the data structures for your documentation.)

10.19 How do I activate the OpenVMS Debugger from an application?

```
#include {lib$routines.h}
#include {ssdef.h}
#include {string.h}

main()
    {
    char ascic_debug_commands[128];
    char *dbgcmd = "*show calls;go;exit";

    strcpy( ascic_debug_commands, dbgcmd );
    ascic_debug_commands[0] = (char) strlen( dbgcmd ) - 1;

    lib$signal(SS$_DEBUG,1,ascic_debug_commands);

    return 1;
    }
```

Also see Section 10.28 for another related discussion of the OpenVMS Debugger, and of a technique that uses the SS$_DEBUG signal.

10.20 Dealing with Endian-ness?

OpenVMS VAX, OpenVMS Alpha and OpenVMS I64 (as well as all Microsoft Windows implementations) all support and all use the little-endian byte ordering. Certain Alpha microprocessors and certain Intel Itanium processors can be configured to operate in big-endian and potentially in bi-endian mode. HP-UX typically operates big-endian.

With little-endian byte order, the least significant byte is always the first byte; the byte at the lowest address. With big-endian byte ordering, the byte storage order in memory is dependent on the size of the data (byte, word, longword) that is being referenced.

Endian-ness is a problem has been solved many times before. Some of the typical solutions include htonl/htons and ntohl/ntohs in the standard C library and the TCP/IP Services XDR (eXternal Data Representation) libraries. One of the more recently introduced network formats, and one that is seeing extensive press and marketing coverage, is XML.

10.21 How to resolve LINK-I-DATMISCH errors?

The message LINK-I-DATMISCH is informational, and indicates that the version of the specified shareable image found in the system shareable image directory does not match the version of the shareable image that was originally loaded into IMAGELIB.OLB, one of the OpenVMS libraries typically searched by the LINKER.

From a privileged username, you can usually completely repair this via the following DCL command:

```
$ LIB/REPLACE/SHARE SYS$LIBRARY:IMAGELIB.OLB SYS$SHARE:LIBRTL.EXE
```

This command assumes that the shareable image that was found in the SYS$SHARE: area is valid and upward-compatiable, and that the image has simply replaced an older version without also updating IMAGELIB.

10.22 HP C and other OpenVMS C Programming Considerations?

VAX C V3.2 was released for OpenVMS VAX systems in 1991. DEC C V4.0 replaced VAX C V3.2 in 1993 as the HP C compiler for OpenVMS VAX systems. HP C is the ANSI C compiler for OpenVMS Alpha systems. VAX C predates the ANSI C standards, and has various areas that are not compliant with ANSI C requirements. HP C is an ANSI C compiler, and can also compile most VAX C code when /STANDARD=VAXC is specified. Versions of this compiler between V3.2 and V6.5 (exclusive) were known as DEC C, DIGITAL C, and Compaq C.

Both compilers can be installed at the same time on the same OpenVMS VAX system, allowing a migration from VAX C to DEC C, and allowing the same DEC C code to be used on OpenVMS VAX and OpenVMS Alpha.

The system manager can choose the system default C compiler when HP C is installed on a system with VAX C, and a C programmer can explicitly select the required compiler for a any particular compilation.

A current "C" license PAK allows access to both VAX C and HP C on the same OpenVMS VAX system.

Various HP C versions can be installed on OpenVMS VAX V5.5-2 and later. OpenVMS VAX releases such as V5.5-2 and V6.0 will require the installation of a HP C RTL kit, a kit that is included with the HP C compiler. OpenVMS VAX versions V6.1 and later do not require a seperate RTL kit, but HP C RTL ECO kits are available to resolve problems found with the C RTL on various OpenVMS releases.

With HP C, for automatic resolution of the standard C library routines by the LINKER utility, use the /PREFIX qualifier, such as /PREFIX=ALL_ENTRIES. If a particular application program replaces an existing C library routine, use /PREFIX=(ALL_ENTRIES,EXCEPT=(...)). (VAX C required explicit specification of an RTL shareable image or C object library during the link.)

When the /PREFIX is requested, the compiler generates a "decc$" prefix on the specified symbols. This prefix allows the LINKER to resolve the external symbols against the symbols present in the DECC$SHR library. The DECC$SHR library is included in the IMAGELIB.OLB shareable image library, and IMAGELIB is searched by default when any program (written in any language) is LINKed. Because the standard C library routine names are very likely to match application routines written in other languages, a prefix "decc$" is added to the C symbol names to assure their uniqueness; to prevent symbol naming conflicts. C programs, however, can sometimes have private libraries for various purposes, and the external routines share the same names as the library routines. (This is not recommended, but there are applications around that use this technique.) Thus the need to explicity specify whether or not the "decc$" prefix should be prepended to the external symbol names by the compiler.

The qualifiers, and most (all?) with associated pragmas, that may be of interest when migrating VAX C code to HP C include:

- Failure to specify the prefixing qualifier (on certain and usually older versions of C) can cause the compiler to not add the prefixes for the names of the C library routines into the references placed in the object module, which can in turn cause problems resolving the external symbols in the library when the object code is linked:

```
/PREFIX=ALL_ENTRIES
```

- Some VAX C programs erroneously write to the string literals. By default, HP C does not allow the constants to change.

```
/ASSUME=WRITABLE_STRING_LITERALS
```

- Enables sharing (‘‘shr’’) of globals and of extern variables. HP C sets externs as non-shareable (‘‘noshr’’), VAX C as ‘‘shr’’.

```
/SHARE_GLOBALS
```

- VAX C assumes common block model for external linkages.

```
/EXTERN_MODE=COMMON_BLOCK
```

- Refers to the padding placed between member elements within a struct. Disabling member alignment packs the data more tightly into memory, but this packaging has performance implications, both on OpenVMS VAX and particularly on OpenVMS Alpha systems.

```
/[NO]MEMBER_ALIGNMENT
```

- Enable all manner of useful compiler diagnostics:

```
/WARN=ENABLE=(LEVEL4,QUESTCODE)/STANDARD=PORT/ACCEPT=NOVAXC_KEYWORDS
```

You can disable extraneous diagnostics with the following:

```
#ifdef __DECC
#pragma message save
#pragma message disable /* insert message tag here */
#endif
```

Permit structure members to be naturally aligned whenever possible, and avoid using /NOMEMBER_ALIGNMENT. If you need to disable member alignment, use the equivilent #pragma to designate the specific structures. The alignment of structure members normally only comes into play with specific unaligned data structures—such as the sys$creprc quota itemlist—and with data structures that are using data that was organized by a system using byte or other non-member alignment.

Versions of HP C such as V6.0 include the capability to extract the contents of the standard header libraries into directories such as SYS$SYSROOT:[DECC$LIB...], and provide various logical names that can be defined to control library searches. With HP C versions such as V6.0, the default operations of the compiler match the expectations of most OpenVMS programmers, without requiring any definitions of site-specific library-related logical names. (And logical names left from older DEC C versions can sometimes cause the compiler troubles locating header files.)

HP C V5.6 and later include a backport library, a mechanism by which HP C running on older OpenVMS releases can gain access to newer RTL routines added to the RTL in later OpenVMS releases—the language RTLs ship with OpenVMS itself, and not with the compilers.

Example C code is available in SYS$EXAMPLES:, in DECW$EXAMPLES (when the DECwindows examples are installed), in TCPIP$SERVICES (or on older releases, UCX$EXAMPLES) when HP TCP/IP Services is installed), on the Freeware CD-ROMs, and at web sites such as

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

10.22.1 Other common C issues

The localtime( ) function and various other functions maintain the number of years since 1900 in the "struct tm" structure member tm_year. This field will contain a value of 100 in the year 2000, 101 for 2001, etc., and the yearly incrementation of this field is expected to continue.

The C epoch typically uses a longword (known as time_t) to contain the number of seconds since midnight on 1-Jan-1970. At the current rate of consumption of seconds, this longword is expected to overflow (when interpreted as a signed longword) circa 03:14:07 on 19-Jan-2038 (GMT), as this time is circa 0x7FFFFFFF seconds since the C base date. (The most common solution is to ensure that time_t is an unsigned.)

If C does not correctly handle the display of the local system time, then check the UTC configuration on OpenVMS—the most common symptom of this is a skew of one hour (or whatever the local daylight saving time change might be). This skew can be caused by incorrect handling of the "is_dst" setting in the application program, or by an incorrect OpenVMS UTC configuration on the local system. (See section Section 4.4.)

Floating point is prohibited in OpenVMS Alpha inner-mode (privileged) code, and in any process or other execution context that does not have floating point enabled. C programmers developing and working with OpenVMS Alpha high-IPL kernel-mode code such as device drivers will want to become familiar with the floating point processing available in the environment, and with the C compiler qualifier /INSTRUCTION_SET=[NO]FLOATING_POINT. Device drivers and other similar kernel-mode C code must be compiled with /INSTRUCTION_SET=FLOATING_POINT and /EXTERN_MODEL=STRICT_REFDEF.

Additionally, the SYS$LIBRARY:SYS$LIB_C.TLB/LIBRARY parameter will be needed to be appended to the module specification or declared via the C compiler's include library logical name mechanism when the C compiler is resolving kernel-mode data structures and definitions. This library contains OpenVMS kernel-mode and other system declaractions, and particularly a mixture of undocumented definitions and declarations, and particularly definitions and declarations that are subject to change (and that can accordingly lead to requirements for the recompilation of application code).

In addition to the user-mode C Run-Time Library (RTL) mentioned in the OpenVMS C RTL documentation and referenced over in Section 3.9, there is a second and parallel kernel-mode RTL accessable to device drivers and other kernel code on OpenVMS Alpha and OpenVMS I64. The most common time this second C library is noticed is when C code is (erroneously) linked with /SYSEXE/SYSLIB, and duplicate symbol errors typically then arise. As code running in supervisor-, executive- or kernel-mode context cannot call out a user-mode RTL or other user-mode library, you will want to respecify the command as LINK /SYSEXE/NOSYSLIB. This will eliminate the duplicate symbol errors, since only the kernel-mode library will be referenced, and it will also avoid calling out into the user-mode libraries.

When sharing variables with other languages, here is some example HP C code...

```
...
#pragma extern_model save
 #pragma extern_model strict_refdef
 extern int   VMS$GL_FLAVOR;
 #pragma extern_model restore
 ...
```

and here is some associated example Bliss code...

```
...
EXTERNAL
   VMS$GL_FLAVOR,
....
```

10.22.2 Other common C++ issues

HP C++ (a separate compiler from HP C) provides both symbol mangling and symbol decoration. Some of the details of working with longer symbol names and the resulting symbol name mangling in mixed language environments are listed in the shareable image cookbook, and in the C++ documentation. Symbol name decoration permits the overloading of functions (by adding characters to the external symbol for the function to indicate the function return type and the argument data types involved), and mixed-language external references can and often do need to disable this decoration via the extern "C" declaration mechanism:

```
extern "C"
  {
  extern int ExternSymbol(void *);
  extern int OtherExternSymbol(void *);
  }
```

Also see Section 14.7 for information on /ARCHITECTURE and /OPTIMIZE=TUNE.

See Section 10.15 for information on the C system and the lib$spawn call in CAPTIVE environments.

Constructs such as the order of incrementation or decrementation and the order of argument processing within an argument list are all implementation-defined. This means that C coding constructs such as:

```
i = i++;
a[i] = i++;
foo( i, i++, --i);
```

are undefined and can have (adverse) implications when porting the C code to another C compiler or to another platform. In general, any combination of ++, –, =, +=, -=, *=, etc operators that will cause the same value to be modified multiple times (between what the ANSI/ISO C standard calls "sequence points") produce undefined and implementation-specific results.

Within C, the following are the "sequence points": the ";" at the end of a C statment, the ||, &&, ?:, and comma operators, and a call to a function. Note specifically that = is NOT a sequence point, and that the individual arguments contained within a function argument list can be processed from right to left, from left to right, or at any random whim.

HP C for OpenVMS VAX (formerly DEC C) and VAX C do differ in the related processing.

So you are looking for OpenVMS-specific definitions (include files)?

UCBDEF.H, PCBDEF.H and other OpenVMS-specific definitions—these are considered part of OpenVMS and not part of the C compiler kit—are available on all recent OpenVMS Alpha releases.

To reference the version-dependent symbol library sys$share:sys$lib_c.tlb, use a command similar to the following for compilation:

```
$ CC sourcea+SYS$LIBRARY:SYS$LIB_C/LIB
```

You can also define DECC$TEXT_LIBRARY to reference the library.

You will want to review the Programming Concepts manual, and specifically take a quick look at Chapter 21.

And some general background: the STARLET definitions (and thus the sys$starlet_c.tlb library) contain the symbols and the definitions that are independent of the OpenVMS version. The LIB definitions (and thus sys$lib_c) contain symbols and definitions that can be dependent on the OpenVMS version. You won't need to rebuild your code after an OpenVMS upgrade if you have included definitions from STARLET. The same cannot be said for some of the definitions in LIB—you might need to rebuild your code. (The UCB structure can and has changed from release to release, for instance.)

Recent versions of C automatically search sys$starlet_c.tlb. Explicit specification of sys$lib_c.tlb is required.

Also see the Ask The Wizard website topics (2486), (3803), and (1661):

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

See Section 9.5 for information on the C off_t limitations, resolved in OpenVMS V7.3-1 and later and in ECO kits available for specific OpenVMS releases. The use of a longword for off_t restricts applications using native C I/O to file sizes of two gigabytes or less, or these applications must use native RMS or XQP calls for specific operations.

10.23 Status of Programming Tools on OpenVMS VAX?

DECthreads V7.3 and the HP C compiler (also known as Compaq C and DEC C) V6.4 are presently expected to be the last updates and the last releases of these development packages for use on OpenVMS VAX. The run-time support for both DECthreads (CMA$RTL) and for C (DECC$CRTL) will continue to be maintained, and will continue to be available on OpenVMS VAX. The VAX C V3.2 compiler is the final VAX C compiler release for OpenVMS VAX, and the VAX C Run-Time Library (VAXCRTL) will also continue to be available.

New development and new features and product enhancements continue for the OpenVMS Alpha and the OpenVMS IA-64 DECthreads and C compilers.

10.24 Choosing a Version Number for Application Code?

One of the common rules-of-thumb used for choosing a displayed version number string for a new version of a layered product or an application, its implications, and its expected effects on client applications and users, follows:

- No functional and no application-visible changes, bugfixes only—the edit number is incremented. These tend to be very small, very isolated, or ECO-level changes. These can also be distributions for specific hardware configurations or platforms, as is the case with an OpenVMS Limited Hardware Release (LHR). Application rebuilds are not expected, and there is an assumption that general user-provided application-related regression testing will not be required.
- Minimal functional and very few user-visible changes—the maintenance number is incremented. These tend to be very small or even ECO-level changes, though somewhat larger than an edit-level change. Application rebuilds are not expected, and there is an assumption that user-provided application-related regression testing will not be required.
- Various small and upward-compatible functional changes—the minor version number is incremented. The changes are user-visible, and are intended to be user-visible. Application rebuilds are not expected. Some application programmers may choose to perform regression tests.
- Large and/or potentially incompatible changes—the major version number is incremented. Some applications might need to be rebuilt. Various application programmers will choose to perform regression tests of their respective applications.

For additional version-numbering materials and for information on assigning module generation numbers, please see the OpenVMS (POLYCENTER) Software Product Installation Utility—variously refered to by acronyms including PCSI and SPIA—reference manual available within the OpenVMS documentation set.

Of course, all of this is obviously subject to interpretation, particularly around the distinction between large and small changes and such. The scale of the application is also a factor, as larger and more complex applications will tend toward smaller increments and will tend to see the maintenance number incremented, while new releases of smaller applications will tend to see the minor version incremented somewhat more frequently.

The goal of all this is to provide a guide to relative scale of changes and the associated effort involved in an upgrade for the user and/or for the application programmer.

10.25 Selecting a Process Dump Directory?

You can customize the device and directory for the process dump by defining the logical names SYS$PROCDMP and SYS$PROTECTED_PROCDMP. The former is for non-privileged dumps, while the latter is the location where privileged image dumps are written, and preferably an area protected against untrusted access. For example:

```
$ define SYS$PROCDMP SYS$ERRORLOG:
$ define /exec SYS$PROTECTED_PROCDMP SYS$ERRORLOG:
```

The abouve presumes that the SYS$ERRORLOG logical name points to a valid location.

There is presently no means to change the name of the generated dump file from IMAGENAME.DMP to something else. Accordingly, you will want to use different target directories for this purpose, particularly if there is more than one application or process potentially writing process dumps.

10.26 Access to Itanium Assembler?

If you are interested in accessing the native Intel Itanium assembler within the OpenVMS I64 GNV environment—and since the iasi64 assembler is a Unix program and GNV is a Unix environment for OpenVMS I64—you can simply copy iasi64.ext into your gnu:[bin] directory in place of ‘‘as.’’, and of ‘‘AS.EXE’’.

Alternately and probably also better, you can write an ‘‘as.’’ script to invoke the iasi64.exe image from its particular prefered location on the local system.

A typical ‘‘as.’’ script looks like this:

```
path/iasi64.exe $1 $2 $3 $4 $5
```

10.27 Kernel-mode coding restrictions?

Floating point is prohibited in OpenVMS Alpha inner-mode (privileged) code, and within any process or other execution context that does not have floating point enabled and available.

Programmers developing and working with OpenVMS Alpha high-IPL kernel-mode code, such as device drivers, will further want to become familiar with the floating-point processing and the instruction set emulation available in the particular target environment (if any). When working with C, inner-mode programmers will want to become familiar with the C compiler qualifier /INSTRUCTION_SET=[NO]FLOATING_POINT.

Device drivers and other similar kernel-mode C code must be compiled with /INSTRUCTION_SET=FLOATING_POINT and /EXTERN_MODEL=STRICT_REFDEF.

Additionally, inner-mode code cannot call out to the user-mode language run-time libraries nor to any of the OpenVMS system run-time libraries. In particular, this prohibition prevents pages of inner-mode-protected memory from being allocated and interspersed within the user-mode heap or other such user-mode data structures.

The prohibtion on user libraries also generally means that such code must be linked with LINK /NOSYSLIB, and quite probably also with /SYSEXE. The former causes the linker to avoid searching the system shareable image libraries (via IMAGELIB.OLB), while the latter brings in symbols typically only known to or otherwise accessable from the OpenVMS executuve.

To include kernel-mode C programming definitions, macros and system constants within a C compilation, include SYS$LIBRARY:SYS$LIB_C.TLB/LIBRARY on the C compilation. (Constructs defined within the system macro library LIB.MLB or within its C equivalent SYS$LIB_C.TLB tend to be version-dependent, or undocumented, or both.) As an example of the compilation, the following is a typical C device driver compilation command:

```
$   CC /STANDARD=RELAXED_ANSI89/INSTRUCTION=NOFLOATING_POINT/EXTERN=STRICT -
        'DEBUG_CC_DQ_OPT' 'ARCH_CC_OPT' 'CHECK_CC_OPT' 'SHOW_CC_OPT' -
        /LIS=LIS$:xxDRIVER/MACHINE_CODE/OBJ=OBJ$:xxDRIVER -
        SRC$:xxDRIVER.C+SYS$LIBRARY:SYS$LIB_C.TLB/LIBRARY
```

Additionally, code running in executive mode in an AST or in kernel mode cannot call RMS services, or routines which directly or indirectly call RMS.

For related kernel-mode programming materials and driver documentation, please see the *Writing OpenVMS Alpha Device Driversin C* book, ISBN 1-55558-133-1.

10.28 Decoding an Access Violation (ACCVIO) Error?

To decode the virtual addresses returned by an access violation or by another similar OpenVMS display, you need to have created and retained a listings file—preferably one with machine code generation enabled—and a full link map.

Starting with the virtual address reported by the error, use the link map to find the module that contributed the code that contains the virtual address range. Calculate the offset from the base of the range, by subtracting the base of the range from teh failing virtual address. Then use the compiler listings for the particular component that contributed the code to locate the offset of the failing instruction.

If the map and listings information was not maintained, working backwards is far more difficult—you are left to use the binary instruction data around the failure to locate the associated source code, and this process is far more involved. This usually involves matching up blocks of decoded instructions around the failing code, or the direct analog involving matching up ranges of decoded instructions. Keep the maps and listing files around, in other words.

Rather easier than an approach based on virtual address arithmetic and far easier than working backwards from the instruction stream is to use integrated debugging—this inclusion is arguably an essential component of any non-trivial application—and to use the OpenVMS Debugger.

The OpenVMS Debugger in particular can be used to examine the source code, to examine the stack, and can even be programmed to wait patiently for the incidence of a particular value or failure or condition, and this is far easier than working backwards from the instruction stream is to use integrated debugging—this inclusion is arguably an essential component of any non-trivial application—and to use the OpenVMS Debugger. The debugger can also be activated from within a signal handler, and commands to generate a traceback can be generated directly, or through the invocation of a procedure containing a series of debugger commands.

Details on the debugger are in the OpenVMS Debugger Manual, and also see the discussion of dyanmically activating the Debugger in Section 10.19.

10.29 Generating an AUTODIN-II CRC32?

The following code can be used to generate an AUTODIN-II 32-bit Cyclic Redundency Check (CRC32) value from an input string descriptor, similar to that used by the HP C compiler for its /NAMES=SHORTENED mechanism, and by various other applications requiring a CRC32.

The routine uses the OpenVMS library routine lib$crc_table to generate a sixteen longword array of data from the specified encoded polynomial coefficient (AUTODIN-II, in this case), and then lib$crc to generate the CRC32 value from the array and the input data.

```
static int CreateCRC32( struct dsc$descriptor *InputDataDesc )
  {
  uint32 AUTODIN2;
  uint32 Seed = ~0UL;
  uint32 Coefficient = 0x0EDB88320UL;
  uint32 CRCArray[16];

  lib$establish( lib$sig_to_ret );

  lib$crc_table( (void *) &Coefficient, (void *) CRCArray );
  AUTODIN2 = lib$crc( (void *) CRCArray, (void *) &Seed, InputDataDesc );
  AUTODIN2 ^= Seed;

  return AUTODIN2;
  }
```

10.30 Enabling built-in tracing?

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SYS_LOAD ADD TR$DEBUG TR$DEBUG/LOAD_STEP=INIT/LOG
SYSMAN> Exit
$ @SYS$UPDATE:VMS$SYSTEM_IMAGES.COM
```

To stop it from loading early in boot

```
$  RUN SYS$SYSTEM:SYSMAN
SYSMAN> SYS_LOAD REMOVE TR$DEBUG TR$DEBUG/LOG
SYSMAN> Exit
$ @SYS$UPDATE:VMS$SYSTEM_IMAGES.COM
```

The first occurance of the name TR$DEBUG within the command is considered the ‘‘product’’ and the second is considered the ‘‘image’’ that should exist within SYS$LOADABLE_IMAGES.

When TR$DEBUG loads in the init phase, it will automatically turn on tracing.

Also see the SDA TR extension.

11

DECwindows

If you are searching for something here,
please consider using the text-format FAQ.

11.1 How do I let someone else display something on my workstation?

On a workstation, you will want to use the ‘‘Customize’’ menu of the session manager utility and select ‘‘Security’’. When the pop-up box appears, you can select the host node, username, and tranport that will allow you to launch an application that targets the workstation display.

If this does not provide you with access to the display, You need a checklist of sorts:

- Make sure that you’ve specified the X-windows ‘‘display’’ correctly on the remote host. For a DECnet transport, the specification uses two colons, while the TCP/IP transport typically uses one. The X Windows server and the X Windows screen follow the host specification, delimited by a period. For example:

Table 11–1 X Windows Display Commands

Shell	Command
csh	`# setenv DISPLAY vms.domain:0.0`
sh and ksh	`# $ DISPLAY=vms.domain:0.0 ; export DISPLAY`
DCL	`$ SET DISPLAY/CREATE/NODE=vms.domain -` `/TRANSPORT=TCPIP/SERVER=server/SCREEN=screen`

- If you have verified the command is correct and things are still not working, ensure the Security settings on the OpenVMS host side will allow the incoming connection: Pull down the ‘‘Options’’ menu in the Session Manager, and select ‘‘Security...’’. If you do not find your host and username and transport listed among the authorized users, you will need to add an entry.

  - There are various transports available, including LOCAL, DECNET, LAT, and TCPIP. You must Select the transport appropriate to the incoming connection.
  - If the transport is ‘‘DECnet’’, do *NOT* add the double colon (::) to the node name.
  - If the transport is ‘‘TCPIP’’, ‘‘Username’’ *must* be an asterisk (*). Why? Because unlike DECnet, the TCP/IP protocol does not provide the remote username information in the incoming connection.
  - If the connection is ‘‘TCPIP’’, it is best to use a full domain name (e.g. Node.Subd.Domain). However, you may have to use the IP address itself, if your host does not have a way to resolve the address via DNS. If you have the luxury of fixed addresses (eg: you are *not* using DHCP), then it can be helpful to add two entries for each TCP/IP host, one that specifies the host name and one that specifies the host address.
  - There are various TCP/IP packages for OpenVMS, and you must use syntax appropriate to the transport installed.
  - If a TCP/IP connection is still not working, ensure that the transport you want has been activated for use with DECwindows. See Section 11.14 for details of configuring TCP/IP as a transport.

- There is a log file created in SYS$MANAGER: which can tell you which transports are loaded, and also tell you what connect attempts were rejected, including showing what the presented credentials were. This file is SYS$MANAGER:DECW$SERVER_0_ERROR.LOG, although the 0 could be another number if you have multiple servers on the workstation. I have found this file to be very useful for tracking down what needs to be put in the Session Manager Security entries.

11.2 How do I create a display on another workstation?

To create a display from an OpenVMS host to a remote X Windows display, use one of the following DCL commands:

```
$ SET DISPLAY /CREATE /TRANSPORT=net_transport /NODE=remote_node
$ SET DISPLAY /CREATE /TRANSPORT=LAT /NODE=remote_node
$ SET DISPLAY /CREATE /TRANSPORT=DECnet /NODE=remote_node
$ SET DISPLAY /CREATE /TRANSPORT=TCPIP /NODE=remote_node
```

Note that LAT is typically used only for the VXT series X Windows terminals, but it can also be used from OpenVMS to OpenVMS systems on various OpenVMS releases, such as on OpenVMS Alpha V6.1 and later. For details on configuring the TCP/IP transport, see Section 11.14.

If you are interested in X Windows terminals and have an older VAXstation system around, please see the EWS package on Freeware V5.0.

11.3 How can I get the information from SHOW DISPLAY into a symbol?

Use the undocumented SHOW DISPLAY/SYMBOL, and then reference the symbols DECW$DISPLAY_NODE, DECW$DISPLAY_SCREEN, DECW$DISPLAY_SERVER and/or DECW$DISPLAY_TRANSPORT.

An example of calling the underlying (and also undocumented) sys$qio programming interface for the WSDRIVER (WSAn:) is available at:

`http://www.hp.com/go/openvms/freeware/`

Look in the Freeware V4.0 directory /srh_examples/DECUS_UNDOC_CLINIC/.

11.4 How do I get a log of a DECterm session?

If you are working from a DECwindows DECterm terminal emulator, you can use the AutoPrint feature. Choose the ‘‘Printer...’’ menu item from the ‘‘Options’’ menu, set the printing destination to the name of the file you want, and set ‘‘Auto Print Mode’’. You are now free to continue.

It should be noted that all of the characters and escape sequences are captured, but if you display the resulting log file on a DECterm, then you will see exactly what was originally displayed.

You can also use the ‘‘Print Screen’’ screen capture available in the DECwindows session manager menus, if you simply wish to snapshot a particular portion of the X Windows display.

If you are using the Freeware VTstar terminal emulator package, you will find a similar logging mechanism is available in the menus.

11.5 Why is DECwindows Motif not starting?

First check to see if there is a graphics device, usually a G* device. (eg: On a DEC 2000 model 300, use the command SHOW DEVICE GQ) If you do not find a graphics device:

- OpenVMS has failed to find the appropriate IRQ information for an EISA graphics card (on the DEC 2000 series) such as the HP (Compaq) QVision, and did not autoconfigure it. Run the correct ECU (for Tru64 UNIX and OpenVMS) and reboot. This is necessary only on EISA-based systems.
- You have an EISA-based system (such as the DEC 2000 model 300) and do not have a HP (Compaq) QVision video card. This EISA graphics card should have Compaq printed on it, and identifies itself as a CPQ3011 or a CPQ3111. If it is not one of these two EISA devices, then OpenVMS does not support it. (There are no other supported EISA graphics controllers, and EISA graphics are normally used with DECwindows only on the DEC 2000 series systems.)
- You have a PCI-based system, and do not have a supported graphics controller—examples of supported controllers include the following:
  - Radeon 7500
  - PowerStorm 3D30, PowerStorm 4D20
  - 3DLabs Oxygen VX1

  See Section 5.16 for further information on some of these graphics controllers.
- You have booted the system minimally, or have otherwise disabled the device autoconfiguration process.

If there is a G* graphics device present:

- There may have been a severe error in the DECwindows startup. Type the contents of SYS$MANAGER:DECW$SERVER_0_ERROR.LOG for any information on errors starting the server.
- The system parameter WINDOW_SYSTEM is not set to 1. While this was a common way for system managers to disable the DECwindows server startup, it is not particularly reliable as DECwindows can now ‘‘correct’’ this setting.

If you really do not want an OpenVMS system with workstation hardware to bootstrap and configure itself as a workstation, add the following definition to SYLOGICALS.COM:

```
$ DEFINE/SYSTEM/EXEC DECW$IGNORE_WORKSTATION TRUE
```

- You may not have a valid DECwindows Motif license loaded. To check for the two most common types of Motif product authorization keys (PAKs), use the following DCL commands:

  ```
  $ LICENSE LIST DW-MOTIF/FULL
  $ LICENSE LIST NET-APP-SUP*/FULL
  ```

  and examine the information displayed. Make sure that one of these licenses is present, valid and active.

  For information on registering software license product authorization keys (PAKs) when you cannot log into the system directly, please see Section 5.6.2.

- Check that the DECW$PRIVATE_SERVER_SETUP.COM is correct for the graphics controller in use. For instance:

  The following is from the 9FX Vision 330 Owners Guide, EK-V330G-OG pg 2-9. Place the following in DECW$PRIVATE_SERVER_SETUP.COM, creatibng .COM from .TEMPLATE if necessary. Locate the DECW$PRIVATE_SERVER_SETUP.COM file in SYS$SPECIFIC:[SYSMGR] or in SYS$COMMON:[SYSMGR] as appropriate; the former file is used for this system within a cluster configuration, and the latter is used for all systems that do not also have a local copy of this file in SYS$SPECIFIC:[SYSMGR].

  ```
  $ DECW$XSIZE_IN_PIXELS == xvalue
  $ DECW$YSIZE_IN_PIXELS == yvalue
  $ DEFINE/SYSTEM DECW$SERVER_REFRESH_RATE rate_in_Hz
  ```

  Also see Section 11.11. Details of the PowerStorm 3D30 and 4D20 settings are available in the OpenVMS Ask The Wizard area.

11.6 How do I set the title on a DECterm window?

If you are creating a new DECterm window, check

```
$ HELP CREATE /TERMINAL /WINDOW_ATTRIBUTES
```

If you want to change the title of an existing window, use the following control sequences, where esc is the ANSI escape code, value decimal 27, and ‘‘text label’’ is what you want to display:

To set the DECterm title, send the escape character, then the characters ‘‘]21;’’, then the text label string, and then an escape character followed by a backslash character.

To set the icon label, send the escape character, then the characters ‘‘]2L;’’, then the icon label string, and then an escape character followed by a backslash character.

To set both the DECterm title and icon to the full device name, you can use the following DCL commands:

```
$  esc[0,7] = 27
$  fulldevnam = F$Edit(F$GetDVI("TT","FULLDEVNAM"),"UPCASE,COLLAPSE")
$  write sys$output esc+ "]21;" + fulldevnam + esc + "\"
$  write sys$output esc+ "]2L;" + fulldevnam + esc + "\"
```

You can also change the title and the icon using the Options-Window... menu.

Also see Section 12.1 and Section 8.13.

11.7 How do I customize DECwindows, including the login screen?

To customize various DECwindows Motif characteristics including the defaults used by the SET DISPLAY command, the DECwindows login screen background logo used (the default is the DIGITAL, Compaq, or HP logo), various keymaps (also see Section 11.7.2 and Section 11.7.1), the FileView defaults, session manager defaults, the DECwindows login processing, DECwindows log file processing, and various other DECwindows attributes, see the example file:

```
$ SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.TEMPLATE
```

This example template file is typically copied over to the filename SYS$COMMON:[SYSMGR]DECW$PRIVATE_APPS_SETUP.COM and then modified to meet site-specific requirements.

Additionally, various X tools such as xsetroot, bitmap and xrdb—some these can be useful in customizing the appearance of an application or of the DECwindows Motif display—are provided in the DECW$UTILS: area.

When using DECwindows V1.2-4 and later on OpenVMS Alpha, the default desktop is the Common Desktop Environment (CDE). You can select your preferred desktop (CDE or DECwindows Motif) when logging in, or you can change the default to the DECwindows Motif desktop using the DCL symbol decw$start_new_desktop in the DECwindows private application setup command procedure. See

SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.TEMPLATE for further details, and how to create DECW$PRIVATE_APPS_SETUP.COM.

Note that with DECwindows CDE, the root window is no longer visible by default. The root window is hidden behind the ‘‘backdrop’’ window of the current CDE workspace. To make the root window visible, use the CDE style manager selection ‘‘backdrop none’’, and use information such as that in the OpenVMS FAQ to set the root window.

To add a new backdrop to the DECwindows CDE environment, the backdrop must first be in or be converted into X11 pixmap format. (This conversion is often possible using tools such as xv.) Then (if necessary) create the default backdrop directory SYS$COMMON:[CDE$DEFAULTS.USER.BACKDROPS]. Place the X11 pixmap file containing the desired image into the backdrops directory, ensure that it has a filename extension of .PM. (The xv default filename extension for the X11 pixmap file is .XPM, while CDE expects only to see files with .PM.) Now invoke the CDE style manager and select a new backdrop. You will find your image will be placed at the end of the list of backdrops available.

If you require a message be included on the initial display—where the start session display and the logo appears—you can use either of the following approaches:

- The simplest approach requires OpenVMS V7.3-2 or later, and the corresponding DECwindows V1.3-1 kit or later. You will want to create a file named SYS$COMMON:[SYSMGR]DECW$GREET.TXT, and this will be displayed in a popup—with an OK button—when the login box is displayed. This is intended specifically for applications requiring such a display.
- The second approach involves copying the file XRESOURCES.DAT from

```
SYS$SYSDEVICE:[VMS$COMMON.CDE$DEFAULTS.SYSTEM.CONFIG.C]
```

into the directory

```
SYS$SYSDEVICE:[VMS$COMMON.CDE$DEFAULTS.USER.CONFIG.C]
```

and editing the copy. Specifically, look for the following:

```
Dtlogin*greeting.labelString:
```

The line is normally commented out, and by default contains the string:

```
Welcome to %localhost%
```

You can change this text to something akin to the following:

```
Dtlogin*greeting.labelString:  Welcome to Heck \n\
This is a Trusted System owned by the Rulers of the planet Zark\n\
\n\
We Come In Peace\n\
\n
If you want Privacy, you've come to the wrong place\n\
\n
```

The lines of text will be centered for you.

In most DECwindows versions, you will be able to onbtain only about eight (8) lines of text. Changes have been implemented in DECwindows V1.3 and later that permit up to about twenty-five (25) lines of text.

The login logo is stored as an XPM bitmap image in the text file SYS$SYSROOT:[SYSCOMMON.CDE$DEFAULTS.SYSTEM.APPCONFIG.ICONS.C]DECDTLOGO.PM, and it can be changed. Copy the file to SYS$SYSROOT:[SYSCOMMON.CDE$DEFAULTS.USER.APPCONFI as DECwindows upgrades can replace the system version of this file.

On DECwindows V1.3-1 and later (and possibly on V1.3), both DECwindows CDE and DECwindows Motif displays use this logo file. On older releases, only the DECwindows CDE displays used this logo file, while the logo used for the Motif login display was hard-coded into the package and the only available override is the DECW$LOGINLOGO command procedure mechanism within the customized, site-specific DECW$PRIVATE_APPS_SETUP.COM file.

Look at the contents of the DECDTLOGO.PM file and at other *.XPM files and tools for additional details.

11.7.1 How do I customize DECwindows keymapping?

Various keymaps can be implemented on OpenVMS and other X Windows systems, allowing the implementation of a Dvorak-style or other alternate keymappings. For details, see the available X Windows documentation (this is the documentation associated with X Windows itself, and not the product documentation for the OpenVMS operating system nor for the DECwindows X Windows implementation) and see the DECwindows *.DECW$KEYMAP (text-format) files found in the DECwindows DECW$KEYMAP: directory.

For other keymapping information, see Section 11.7.2.

11.7.2 Why does the DELETE key delete forward instead of backward?

Some operating systems use Ctrl/H as Delete; as the character back-delete key. Most OpenVMS applications, VT-series terminals (check the terminal settings, many can have this configured within the terminal set-up menus), and DECterm terminals all use Delete as the character back-delete, and use Ctrl/H as a request to move to the beginning of the command line.

The Integrity EFI console does use Ctrl/H as Delete; as the character back-delete key.

Also see the SET TERMINAL/BACKSPACE command available on OpenVMS V8.2 and later.

This behaviour involves the Motif virtual key bindings. When a Motif application starts, it looks at the vendor string returned in the display connection information and attempts to match the string to a table of virtual bindings.

You can override the default bindings in your decw$xdefaults.dat file. Here is the entry you would make to get the default OpenVMS bindings.

```
*defaultVirtualBindings:\
 osfCancel :  F11 \n\
 osfLeft :  Left \n\
 osfUp  :  Up \n\
 osfRight :  Right \n\
 osfDown :  Down \n\
 osfEndLine :Alt  Right \n\
 osfBeginLine :Alt  Left \n\
 osfPageUp :  Prior \n\
 osfPageDown :  Next \n\
 osfDelete :Shift  Delete \n\
 osfUndo :Alt  Delete \n\
 osfBackSpace :  Delete \n\
 osfAddMode :Shift  F8 \n\
 osfHelp :  Help \n\
 osfMenu :  F4 \n\
 osfMenuBar :  F10 \n\
 osfSelect :  Select \n\
 osfActivate :  KP_Enter \n\
 osfCopy :Shift  DRemove \n\
 osfCut  :  DRemove \n\
 osfPaste :  Insert
```

To merge:

```
$ xrdb :== $decw$utils:xrdb.exe
$ xrdb −nocpp −merge decw$xdefaults.dat
```

Also note that the DECW$UTILS:DECW$DEFINE_UTILS.COM procedure can be used to establish the xrdb and other symbols.

Also see the DECxterm directory of Freeware V5.0 for details on connecting to OpenVMS from various UNIX platforms.

For other keymapping information, see Section 11.7.1.

11.8 Why doesn't XtAppAddInput( ) work on OpenVMS?

Yes, XtAppAddInput( ) does work on OpenVMS. The MIT definition of the X Windows call XtAppAddInput( ) includes platform-specific arguments.

On platforms where C is the typically the primary programming language for the platform, the file descriptor mask is one of the arguments to the XtAppAddInput( ) call.

On OpenVMS, the platform-specific arguments to this call include an event flag and an IOSB, as these are the traditional OpenVMS constructs used to synchronize the completion of asynchronous operations. While it would be easier to port non-OpenVMS C code that calls XtAppAddInput( ) over to OpenVMS if the arguments included the C file descriptor, this would make the call unusable from other OpenVMS languages, and would make it extremely difficult to use OpenVMS features such as ASTs and sys$qio calls.

One restriction on the event flag: the event flag chosen must be from event flag cluster zero. When using the traditional lib$get_ef and lib$free_ef calls to allocate and deallocate event flags, you must first explicitly call lib$free_ef to free up some event flags in event flag cluster zero. Please see the event flag documentation for specific details on these calls and for specific event flags that can be freed in event flag cluster zero.

Here is some example code that covers calling this routine on OpenVMS:

```
    m->InputID = XtAppAddInput(
        m->AppCtx,
        m->InputEF,
        m->InputIosb,
        the_callback, 1 );
    if ( !((int) m->InputID ))
        {
        XtAppErrorMsg(
            m->AppCtx,
            "invalidDevice",
            "XtAppAddInput",
            "XtToolkitError",
            "Can't Access Device",
            (String *) NULL,
            (Cardinal *) NULL );
        ...
```

11.9 Why do the keyboard arrow keys move the DECwindows cursor?

Congratulations, you have just stumbled into ‘‘dead rodent’’ mode. This DECwindows environment—where the keyboard arrow keys move the mouse cursor and where the [SELECT], [PREV], and [NEXT] keys emulate the three mouse buttons—allows rudimentary system operations when the mouse is among the casualties.

To enter or exit ‘‘dead rodent’’ mode, enter the following: CTRL/SHIFT/F3

11.10 Why does half my DECwindows display blank?

This is likely a result of receiving an OPCOM or other console message on a system that shares the system console with the DECwindows graphics workstation display.

You can toggle off the console display window using CTRL/F2 and you can enable a serial console per Section 14.3.6 or Section 14.3.3.3.

Also see the console message window application available with recent DECwindows versions—DECwindows versions V1.2-3 and later will enable this window by default. For details on this console message window, see the DECW$CONSOLE_SELECTION option in SYS$STARTUP:DECW$PRIVATE_APPS_SETUP.TEMPLATE.

On older releases, you can disable output using the following:

```
$ SET TERMINAL/PERMANENT/NOBROADCAST OPA0:
$ DEFINE/USER SYS$COMMAND OPA0:
$ REPLY/DISABLE
```

Also see Section 14.3.3.2, Section 14.17, and Also see Section 8.4,

11.11 %DECW-W-NODEVICE, No graphics device found on this system?

To resolve the following error:

```
%DECW-W-NODEVICE, No graphics device found on this system
-DECW-I-NODECW, DECwindows graphics drivers will not be loaded
```

- Ensure that the system parameter WINDOW_SYSTEM is set to 1. If it is not set to a value of 1, issue the commands:

  ```
  $ run sys$system:sysgen
  USE CURRENT
  SET WINDOW_SYSTEM 1
  WRITE ACTIVE
  WRITE CURRENT
  EXIT
  ```

  Then reboot the system.

- On OpenVMS Alpha, ensure the SYSMAN IO PREFIX LIST is set correctly, and specifically ensure the DECW$ prefix is included in the existing list. If it is not, you will need to add it:

```
$ run sys$system:sysman
IO SHOW PREFIX
IO SET PREFIX=(DECW$,*)    * = list returned by the show command
IO AUTO/LOG
EXIT
```

- Ensure that the image SYS$SHARE:DECW$ICBM.EXE is installed in memory. If it is not installed, then install it:

```
$ INSTALL LIST/FULL SYS$SHARE:DECW$ICBM
$ INSTALL REPLACE SYS$SHARE:DECW$ICBM
$ EDIT SYS$MANAGER:SYCONFIG.COM

$! The following line was added to install
$! support for the Mach64 Graphics Card
$!
$ INSTALL REPLACE SYS$SHARE:DECW$ICBM
$ ^Z
```

  Then reboot the system.

  The ICBM mechanism is not used on and not needed by more recent DECwindows versions.

- If the system still complains ‘‘%DECW-W-NODEVICE, No graphics device found on this system’’, then:
    - Boot the system as normal
    - Login as SYSTEM.
    - Create the file SYS$COMMON:[SYSMGR]DECW$USER_AUTOCONFIG.DAT. Protection must permit world read access.
    - Add the following string on the very first line:

```
CLEAR_PFLAG = ISA_4BYTE
```

    - Save the file
    - Set the file protections

```
$ SET PROTECTION=W:RE SYS$MANAGER:DECW$USER_AUTOCONFIG.DAT
```

    - Reboot the system

Also see Section 11.5.

11.12 How can I reset the warning bell volume?

With DECwindows CDE drivers and ECOs starting with ECOs for the DECwindows keyboard driver SYS$IKBDRIVER.EXE in OpenVMS Alpha V7.1-2 and V7.2-1 and with the SYS$IKBDRIVER.EXE included in OpenVMS V7.2-1H1 and later, the DECwindows CDE controls will now correctly manage the setting of the warning bell volume.

Unfortunately, the equivalent controls in the older DECwindows Motif interface are not compatible and can no longer manage the warning bell volume.

If you need to manage the volume with DECwindows Motif, consider using the following approach:

```
$ @decw$utils:decw$define_utils
$ xset b 1 100 100
```

The numerics are the volume, pitch, and duration, respectively.

Why? When OpenVMS first started supporting the PC-style keyboards, the X Windows Server and the keyboard driver interface did not support the pitch and duration, and neither did DECwindows Motif. The DECwindows keyboard driver was accordingly changed to use the volume from the keyclick setting (keyclick is not available in a PC-style keyboard) and the bell volume setting to control the pitch and duration.

DECwindows CDE does provide sliders for setting pitch and duration, so the keyboard driver and X Windows Server were modified to provide all of the information, and now the DECwindows CDE sliders work. This change is unfortunately incompatible with the old scheme used on the pre-CDE desktops, and the volume controls are now incompatible with the current keyboard drivers. Hence the use of xset.

11.13 How can I alter the DECwindows CDE backdrop?

To select a separate backdrop to be displayed on each screen using DECwindows CDE:

- Click on the Application Manager. This is the drawer icon on the CDE toolbar.
- Click on Desktop Tools
- Click on Set Default Screen and select the required screen
- Click on the Style Manager. This is the one containing the mouse and ttt on the CDE toolbar
- Now change the background.

11.14 How can I enable the DECwindows TCP/IP Transport

To configure the TCP/IP transport for DECwindows, first ensure that a TCP/IP package is installed and configured. Then set the DCL symbol DECW$SERVER_TRANSPORTS in SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM to the appropriate local value, based on the comments in that file. If you do not have a copy of SYS$STARTUP:DECW$PRIVATE_SERVER_SETUP.COM, the use the following COPY command to create this file based on the provided template file:

```
$ COPY SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.TEMPLATE -
$_ SYS$COMMON:[SYSMGR]DECW$PRIVATE_SERVER_SETUP.COM
```

11.15 Can I use DECwindows 1.2-* on OpenVMS V7.3-2 or later?

The short answer is no.

OpenVMS Alpha V7.3-2 only supports DECwindows Motif V1.3 and later. If you require DECwindows V1.2-6 or earlier, then you are limited to operations on OpenVMS Alpha V7.3-1 and earlier releases.

The central technical reason involves depdendencies among the parts of the X11 subsystem that are delivered with the base OpenVMS operating system including the X Windows display server and the transport images, and the parts of the DECwindows product that are delivered within the DECwindows installation kits including the client libraries and the DECwindows applications.

DECwindows V1.3 and later made substantial changes to the transport layer, and these required corresponding changes to both the associated client and server code. OpenVMS Alpha V7.3-2 includes the server and transport with the V1.3 modifications. These changes were in support of the upgrade of Xlib from X11R5 to X11R6.6, and transport-level changes associated with support of the Kerberos and LBX features.

If you attempt to load DECwindows V1.2-6 images onto an OpenVMS Alpha V7.3-2 or later system, the DECwindows libraries will not function with with system images and will particularly not function with the transport layer.

11.16 How to add Fonts into DECwindows?

The following assumes DECwindows V1.3-1 and OpenVMS Alpha V7.3-2 and later unless stated otherwise, and can permit fonts of various formats to be added into the DECwindows environment.

The recommended location for user font files is to place them in the directories which are reserved for this purpose, typically located below the SYS$COMMON:[SYSFONT.DECW] directory.

```
SYS$COMMON:[SYSFONT.DECW.USER_100DPI]
SYS$COMMON:[SYSFONT.DECW.USER_75DPI]
```

The above are recommended for PCF files of 100 Dots Per Inch (DPI) and of 75 DPI resolution, respectively.

```
SYS$COMMON:[SYSFONT.DECW.USER_COMMON]
```

The above is recommended for other PCF files, such as terminal (character cell) fonts, and fonts used by specific applications.

```
SYS$COMMON:[SYSFONT.DECW.USER_CURSOR16]
SYS$COMMON:[SYSFONT.DECW.USER_CURSOR32]
```

The above are recommended for cursors.

```
SYS$COMMON:[SYSFONT.DECW.USER_SPEEDO]
```

SPEEDO is recommended for SPD files.

`SYS$COMMON:[SYSFONT.DECW.USER_TRUETYPE]`

USER_TRUETYPE is recommended for TrueType (TTF) fonts. Fonts placed in this directory should be in the ‘‘Windows / Linux’’ format.

The directory will contain the font files themselves, and a data file that describes each font in the directory. This file is named DECW$FONT_DIRECTORY.DAT or DECW$FONT_DIRECTORY_extension.DAT, where ‘‘extension’’ is replaced by the type of font (100DPI, SPEEDO, TRUETYPE, TYPE1, etc.)

Make sure that the file protection on the font files is set to allow world access to the fonts.

For example: to add TrueType fonts to DECwindows, place the font files in SYS$COMMON:[SYSFONT.DECW.USER_TRUETYPE]

A directory listing might look like this:

```
Directory SYS$COMMON:[SYSFONT.DECW.USER_TRUETYPE]

ARKOI8N.TTF;1                             46KB/48KB        5-MAR-1995 04:00:00.00
backstage.ttf;1                           55KB/56KB       19-JUL-2004 09:42:20.92
IDAutomationHC39M_Free.ttf;1              27KB/32KB       29-JUL-2003 11:25:48.00
...
texsi.ttf;1                              133KB/136KB      25-MAY-2003 15:31:11.00
texw.ttf;1                               150KB/152KB      25-MAY-2003 15:32:33.00

Total of 37 files, 3.09MB/3.23MB
```

The case of the filename is not important.

TrueType fonts should be in Stream_LF file format.

To generate the appropriate DECW$FONT_DIRECTORY.DAT file for most font formats, issue the command:

```
$ FONTCOMPILER /DIRECTORY
```

The above may or may not operate with TrueType files, and you will likely have to generate the DECW$FONT_DIRECTORY_TRUETYPE.DAT file manually. A sample file follows:

```
37
BACKSTAGE.ttf -Grfonts-Backstage-bold-r-normal--0-0-0-0-p-0-iso8859-1
IDAutomationHC39M_Free.ttf -IDAutomation-HC39M-medium-r-normal--0-0-0-0-m-0-misc-Barcode39
SUSESerif-Bold.ttf -Suse-Suse-bold-r-normal--0-0-0-0-p-0-iso8859-1
SUSESerif-Roman.ttf -Suse-Suse-medium-r-normal--0-0-0-0-p-0-iso8859-1
SUSESans-Bold.ttf -Suse-Suse-bold-r-normal-sans-0-0-0-0-p-0-iso8859-1
SUSESans-BoldOblique.ttf -Suse-Suse-bold-o-normal-sans-0-0-0-0-p-0-iso8859-1
SUSESans-Oblique.ttf -Suse-Suse-medium-o-normal-sans-0-0-0-0-p-0-iso8859-1
SUSESans-Roman.ttf -Suse-Suse-medium-r-normal-sans-0-0-0-0-p-0-iso8859-1
SUSESansMono-Bold.ttf -Suse-Suse Mono-bold-r-normal-sans-0-0-0-0-m-0-iso8859-1
...
MCTIMEBI.TTF -UOregon-MAC C Times-bold-i-normal--0-0-0-0-p-0-macedonian-0
MCTIMEI.TTF -UOregon-MAC C Times-medium-i-normal--0-0-0-0-p-0-macedonian-0
```

The first line of this data file is the number of font file entries which follow. Each entry consists of the font file name, and a font description. There are fourteen fields in the description, separated by hyphens (dashes, ‘‘-’’). Fields may contain embedded spaces. The fields are

- Foundry: the name of the company or person which produced the font.
- Family: the name of the Typeface (what most people will call the ‘‘font’’).
- Weight: How ‘‘heavy’’ the type appears. Normal fonts are ‘‘medium’’ or ‘‘regular’’, variations include ‘‘bold’’, ‘‘demi’’, ‘‘light’’, etc.
- Slant: ‘‘r’’ for regular, ‘‘i’’ for italic, or ‘‘o’’ for oblique.
- Width: ‘‘normal’’, ‘‘wide’’, ‘‘narrow’’, ‘‘condensed’’, etc.
- Style: normally left empty, it can also identify variations on a basic family such as ‘‘sans’’ (sans serifs; without the serif, the ending and usually pointed portion of the stroke). Fonts of different styles can be grouped in the same family.
- Sizes: the next four fields identify the size and scale of individual characters for fonts that have fixed point sizes. For fonts which scale (such as TrueType), the four fields are all zero.
- Spacing: ‘‘p’’ for proportional, ‘‘m’’ for monospaced, or ‘‘c’’ for character cell.

  Note: although DECwindows can identify different spacings within a family, the author has found that mixing monospaced and proportional fonts in the same family may cause some proportional font options to not appear in a font selection menu within Notepad (only). (A fix for this is expected in DECwindows V1.5 and later.)
- The next field is always zero for TrueType fonts.

- Character Set: the last two fields identify the name and version number of the character set represented within the font. For many applications, these fields are informational only.

The next step is to update the list of fonts known to DECwindows, using the xset utility.

```
$ mc decw$utils:xset fp rehash
```

It is also possible to reset the font list to the default:

```
$ mc decw$utils:xset fp default
```

This is useful if you need to recover from errors.

The Notepad utility, normally available through the ‘‘Applications’’ menu in Session Manager, is a convenient way to see if the font is available. Start the application, select ‘‘Options’’, then select ‘‘Font...’’. In the ‘‘Family (Foundry)’’ window, you will see the list of fonts available. User-added TrueType fonts will normally be at the end of this list. Select the desired font family, then select the Size (dpi) (which will always be 0(0) for TrueType fonts), and the various font options (Weight, Slant, Width, etc.) should appear in the next window. You should then be able to select the desired font and click OK or Apply to use it, or Cancel to exit without changing the font.

If you don’t see all of the fonts you added, check to see that the number at the beginning of the DECW$FONT_DIRECTORY*.DAT file is correct, that the files are set to world (or appropriate) access, and that TrueType fonts are in Stream_LF format.

Some applications require entering a full font name, which will look like the font description entry.

Please keep in mind that not all applications can use every font which may be available on your system. For example, DECterm is designed to use families of fonts specifically designed for character cell applications. Other fonts (specifically TrueType) may work erratically, and may result in an unusable display. It is best to use only monospaced fonts specifically intended for DECterm with DECterm.

The SYS$COMMON:[SYSFONT.DECW.USER_TRUETYPE] doesn’t exist on OpenVMS VAX V7.3 with DECwindows V1.2-6, but the procedure above does appear to work if the directory is created and the instructions above are followed.

12 Miscellaneous Information

If you are searching for something here,
please consider using the text-format FAQ.

12.1 Where can I find information on escape and control sequences?

Information on escape and control sequences can be found in the *OpenVMS I/O User's Reference Manual*, in the chapter on the terminal driver. The chapter also includes details on the general format and content of these sequences.

Specific details on the escape and control sequences supported by a particular serial device are typically found in the documentation provided with the specific device. Information on the sequences supported by DECwindows DECterm terminal emulator are included in the DECwindows documentation.

Examples of common escape and control sequences—those typically used by the OpenVMS screen management package—can be found in the OpenVMS system file SYS$SYSTEM:SMGTERMS.TXT. (This file can be queried under program control using SMG$GET_TERM_DATA, and you don't need to use all of SMG to use this call.)

The following refers to the function keys on the LK-series keyboards found on the VT-series terminals such as the VT220 and VT510, and the LK-series keyboards found on the OpenVMS workstations, and the keyboards found on compatible terminals. (Though note that the keyboard itself does *not* generate the sequence, the terminal or terminal emulator generates the sequence in response to user input.) In the following, {CSI} is decimal code 155 and can be replaced by the sequence "{ESC}[" (without the quotes) particularly for seven-bit operations, SS3 is decimal code 143 and can be replaced by "{ESC}O" particularly for seven-bit operations. Older VT1xx series terminals and any other terminals operating with seven-bit characters should not be sent eight-bit operators such as {CSI} and {SS3}.

```
PF1={SS3}P PF2={SS3}Q PF3={SS3}R PF4={SS3}S
KP0={SS3}p KP1={SS3}q KP2={SS3}r KP3={SS3}s KP4={SS3}t KP5={SS3}u
KP6={SS3}v KP7={SS3}w KP8={SS3}x KP9={SS3}y KPCOMMA={SS3}l KPMINUS={SS3}m
KPPERIOD={SS3}n ENTER={SS3}M DNARROW={CSI}B UPARROW={CSI}A LFARROW={CSI}D
RTARROW={CSI}C FIND={CSI}1~ INSERT={CSI}2~ REMOVE={CSI}3~ SELECT={CSI}4~
PREV={CSI}5~ NEXT={CSI}6~ F6={CSI}17~ F7={CSI}18~ F8={CSI}19~ F9={CSI}20~
F10={CSI}21~ F11={CSI}23~ F12={CSI}24~ F13={CSI}25~ F14={CSI}26~
HELP={CSI}28~ DO={CSI}29~ F17={CSI}31~ F18={CSI}32~ F19={CSI}33~ F20={CSI}34~
```

An example of working with escape sequences (in DCL) follows:

```
$ esc5m = "*[5m"
$ esc5m[0,8] = 27
$ esc0m = "*[0m"
$ esc0m[0,8] = 27
$ write sys$output esc5m + "blinking text" + esc0m
```

Documentation on an ANSI terminal relatively similar to the VT525 series is available at:

- ftp://ftp.boundless.com/pub/text/adds/docs/260_prog/
- ftp://ftp.boundless.com/pub/text/adds/docs/260_user/

Also see the various documentation and manuals available at:

- http://www.vt100.net/

Information on the ReGIS graphics character set is available at:

- http://www.cs.utk.edu/~shuford/terminal/dec_regis_news.txt

Also:

- http://www.boundless.com/Text_Terminals/VT/

Also see Section 11.6, Section 8.13.

12.2 Does DECprint (DCPS) work with the LRA0 parallel port?

No.

The parallel printing port LRA0: found on many OpenVMS Alpha systems is capable of some bidirectional communications, with enough for basic operations with most parallel printers.

DECprint (DCPS) requires more than just the simple handshaking provided by the LRA0: port, therefore DCPS does not work with the LRA0: port.

12.3 How do I check for free space on a (BACKUP) tape?

You cannot know for certain, though you can certainly estimate the remaining capacity.

Tape media is different than disk media, as disks have a known and pre-determined fixed capacity. Modern disks also appear logically perfect, based on bad block revectoring support and the extra blocks hidden within the disk structure for these bad block replacements.

The capacity of tape media is not nearly as pre-determined, and the capacity can vary across different tape media (slightly different media lengths or different foil markers or other variations, for instance) and even on the same media over time (as bad spots in the media arise). Tapes can vary the amount of recording media required, depending on the remaining length of the tape, the numbers of correctable and uncorrectable media errors that might occur, the numbers and sizes of the inter-record gaps and related tape structure overhead, the particular media error recovery chosen, the tape density, the efficiently of any data compression in use, and the storage overhead required by BACKUP, tar, and other similar commands.

BACKUP using with the default settings results in approximately 15% overhead, in terms of saveset size. (eg: Assuming a 500 KB input, the total size would be 575 KB.)

Assuming no compression:
4 GB media / 575 KB saveset = 7294 savesets

Assuming 1:2 compression:
8 GB media / 575 KB saveset = 14588 savesets

Note

There are no inter-record gaps on DAT tapes. When determining media capacity, you have to consider these gaps with nine-track magtape media and other formats with gaps. This is not the case with DAT (DDS), as the format has no recording gaps. However, the block structure underneath the variable length record recording is based on a block size of circa 124 KB. Further, writing doubles filemarks and such can cause a loss of up to the underlying block size. Thus even though there are no inter-record gaps on DAT, larger savesets are still usually best.

The compression algorithms used on various devices are generally not documented—further, there is no way to calculate the effective data compression ratio, the tape mark overhead, and similar given just the data to be stored on tape—short of actually trying it, of course.

A typical compression ratio found with "everyday" data is somewhere around 1:1.8 to 1:2.

Note

OpenVMS often uses the term COMPACTION for compression control, as in the qualifier /MEDIA_FORMAT=COMPACTION.

12.4 Correctly using license PAKs and LMF?

If you have multiple LMF$LICENSE.LDB databases in your OpenVMS Cluster, then each and every PAK must be installed in each and every license database present in an OpenVMS Cluster. Even if you use /EXCLUDE or /INCLUDE, you need to have a consistent set of PAKs registered across all licensing databases present in the OpenVMS Cluster.

If your software license permits it, you can use the following two commands to transfer license PAKs:

```
$ LICENSE COPY...
$ LICENSE ISSUE/PROCEDURE/OUTPUT=file product,...
```

To display the particular license(s) required (such as when you receive a NOLICENSE error), use the following DCL sequence:

```
$ SET PROCESS/PRIVILEGE=ALL
$ REPLY/ENABLE
$ DEFINE/SYSTEM/EXECUTIVE LMF$DISPLAY_OPCOM_MESSAGE
```

This logical name will cause all license failures to generate OPCOM messages, and this will hopefully show which license(s) you need— there may well also be additional license failures displayed, as various products can check for and can be enabled by multiple license PAKs. You will want to deassign this logical name when done.

Some of the more common license PAKs:

```
DECnet Phase IV:   DVNETRTG, DVNETEND, DVNETEXT, or NET-APP-SUP*
DECnet-Plus:       DVNETRTG, DVNETEND, DVNETEXT, or NET-APP-SUP*
TCP/IP Services:   UCX, or NET-APP-SUP*
OpenVMS Alpha:     OPENVMS-ALPHA and OPENVMS-ALPHA-USER
OpenVMS VAX:       VAX-VMS
OpenVMS Galaxy:    OPENVMS-GALAXY
Cluster (Alpha):   VMSCLUSTER, NET-APP-SUP*
Cluster (VAX):     VAXCLUSTER, NET-APP-SUP*
```

Various NET-APP-SUP (NAS) license packages are available, each with differing collections of products authorized. See the various NAS Software Product Description (SPD) documents for specific details.

- http://h18000.www1.hp.com/info/spd/

  OpenVMS typically uses SPD 25.01.xx, SPD 41.87.xx, and SPD 82.35.xx.

To determine which license PAK is failing (via a license check failure OPCOM message), use the command:

```
$ DEFINE/SYSTEM/EXECUTIVE LMF$DISPLAY_OPCOM_MESSAGE TRUE
```

Realize that defining this logical name will cause license checks that are otherwise hidden (unimplemented, latent, or part of a check for any of a series of licenses) to become visible. In other words, expect to see zero or more spurious license check calls when you define this, in addition to the check for the particular license.

For information on PAKGEN and on generating license PAKs, please see Section 10.10. For information on obtaining commercial and hobbyist licenses, and for additional adminstrative information on LMF, please see Section 2.8.4 and Section 2.8.1.

13 Finding and Using Software

If you are searching for something here,
please consider using the text-format FAQ.

For information on programming on OpenVMS, see Chapter 10.

13.1 Where can I find freeware/shareware/software for OpenVMS?

- Details on many commercial OpenVMS products are available in the catalog located at:
  - http://www.hp.com/go/dspp_catalog
- The OpenVMS Freeware is is distributed with OpenVMS, and is also available for download at various sites, including the following:
  - http://www.hp.com/go/openvms/freeware/
  - ftp://ftp.montagar.com/
  - ftp://mvb.saic.com/freewarev40/

  and at various other sites. The website also includes various updates and new packages that become available after the CD-ROM distributions are created.

  Submissions to the OpenVMS Freeware can be made via:

  - http://www.hp.com/go/openvms/freeware/

  To acquire the OpenVMS Freeware CD-ROM distribution, you can order an OpenVMS distribution from HP (the Freeware is included)(see the OpenVMS SPD for part numbers), or you can specifically order a Freeware distribution from HP under part number:

  - QA-6KZAA-H8

  The Freeware CD-ROM set contains a large assortment of freeware, and is a good starting point if looking for utilities. Many of the packages listed below are also on the Freeware CD. Some of the most oft-requested OpenVMS tools on the Freeware CD include ZIP and UNZIP and GZIP (please see Section 13.11, MMK (make), PINE, PERL, TAR, UUENCODE and UUDECODE. Many other tools are available on the Freeware.

- The UUENCODE and UUDECODE tools and various other tools are also available as part of TCP/IP Services package. (Use the DCL command procedure SYS$STARTUP:TCPIP$DEFINE_COMMANDS.COM available on V5.0 and later to set up the necessary DCL foreign command symbols used for these and for various other tools provided by TCP/IP Services.)
- OpenVMS software (formerly at Western Kentucky University (WKU) is now available via Madgoat and via Process Software archives:
  - http://www.process.com/openvms/index.html
  - http://www.madgoat.com/
- The FILESERV packages are also available via anonymous FTP from:
  - ftp://ftp.process.com/vms-freeware/
  - ftp.process.com, under [.WKU.VMS.FILESERV].
  - ftp.vms.stacken.kth.se, under [.MIRRORS.WKU.VMS.FILESERV].
  - ftp.ctrl-c.liu.se, under [.WKU.VMS.FILESERV].
  - ftp.riken.go.jp
  - ftp.vsm.com.au, under kits and kits/decwindows.
  - ftp.vsm.com.au, via the WWW instead of FTP.

  The packages are also available via e-mail from FILESERV@PROCESS.COM. Send the commands HELP and DIR ALL in the body of a mail message to this email address for further information on retrieving the files.
- Arne Vajhøj operates an OpenVMS website, with software and other pointers, at:
  - http://www.levitte.org/~ava/
- Jouk maintains a summary of various public-domain and open-source software ports, and specifically ports of packages for use on OpenVMS, at:
  - http://nchrem.tnw.tudelft.nl/openvms/
- The Kermit file transfer package is available at:
  - http://www.columbia.edu/kermit/
  - ftp://kermit.columbia.edu/

- ZMODEM is available at:
  - ftp://ftp.cs.pdx.edu/pub/zmodem

  See the FILES file in that directory for further details. Note that this freeware version of ZMODEM will interoperate only with ZMODEM software that is licensed from Omen Technology. (This package is also available on various of the Freeware distributions.)
- A good source of software for OpenVMS systems and for various other platforms is the Encompass (formerly known as DECUS) library. Online catalogs are available as well as some software via
  - ftp://ftp.decus.org
  - gopher://gopher.decus.org/
  - http://www.decus.org/
- DECUS SIG Tape collections are available on Mark Berryman's system,
  - ftp://mvb.saic.com/
- David Jones's DECthreads-based HTTP_SERVER World-Wide Web server for OpenVMS, often known as the OSU webserver:
  - http://kcgl1.eng.ohio-state.edu/www/doc/serverinfo.html
- Secure Shell (SSH) support for OpenVMS is available. Use of SSH V2 with current updates is recommended, as the security of older SSH releases have been compromised. Server for OpenVMS:
  - Secure Shell (SSH) Server for OpenVMS:
    - http://kcgl1.eng.ohio-state.edu/~JONESD/ssh/DOC/

    Secure Shell (SSH) FISH Client for OpenVMS:
    - http://www.free.lp.se/fish/

    SSH V2 support is included with TCP/IP Services V5.4 and later, and support is likely also available within available third-party IP stacks.

    For information on the SSH key file configuration, see topic 151.22 and following in the VMS notes conference on the Deathrow Cluster (deathrow.vistech.net and openvms-rocks.com). You can telnet into the host, and an announced anonymous login is/was available; username DEMO, password USER.

  Information on OpenSSL (SSLeay) for OpenVMS:
  - http://www.free.lp.se/openssl/

  - http://www.free.lp.se/fish/

  Information on OpenSSL (SSLeay) and OSU Web server interoperation:

  - http://www.ourservers.net/openvms_ports/

  OpenSSL is included with OpenVMS V7.3-1 and later.

- DECwindows Motif V1.2-3 includes NCSA Mosaic 2.4 built for TCP/IP Services (UCX). V1.2-4 includes Spyglass Enhanced Mosaic, which supports many of the Netscape enhancements. Versions of the Netscape Navigator and particularly the Mozilla Web Browser are also available for OpenVMS. The Compaq Secure Web Browser (CSWB) kit is a packaged version of the Mozilla.org Mozilla Web Browser.
- A port of the NCSA Mosaic web browser that supports TCP/IP Services, Multinet and SOCKETSHR/NETLIB is available from:
  - ftp://wvnvms.wvnet.edu/mosaic/

  Versions of the Mosaic web browser are also available on the Freeware.
- Lynx (a character-cell World-Wide-Web web browser) is available from
  - ftp://ftp2.cc.ukans.edu/pub/lynx

  Versions of Lynx, a character-cell web browser, are also available on the OpenVMS Freeware.
- The Netscape Navigator and Mozilla web browsers are available at:
  - http://www.openvms.compaq.com/ebusiness/Technology.html
  - http://www.mozilla.org/
- PGP (Phil Zimmerman's ''Pretty Good Privacy'') is available from various distribution sites, including those listed in the PGP FAQ. Information on PGP and on OpenVMS downloads of PGP is available. Relevant URLs include:
  - http://www.ipgpp.org/
  - http://web.mit.edu/network/pgp.html
  - http://www.pgpi.com/
  - http://www.yrl.co.uk/~phil/pds/pds.html
  - http://www.nai.com/
- GNU Privacy Guard (GPG, GnuPG) is available.

Search the comp.os.vms newsgroup archives for information regarding GnuPG; the source code, binaries for various platforms, and documentation are all available at:

- http://www.gnupg.org/

The OpenVMS source code and OpenVMS Alpha images can be found at:

- http://saf.bio.caltech.edu/pub/SOFTWARE/openvms/
- ftp://ftp.process.com/vms-freeware/mathog/

As of this writing, the port only runs on OpenVMS Alpha, but an investigation of an OpenVMS VAX port is reportedly under consideration.

- An archive of the CENA DECwindows, X Windows, and VMS software packages can be found at the following sites:
  - http://decwarch.free.fr/
  - ftp://ftp2.cnam.fr/decwindows/
  - ftp://ftp.ctrl-c.liu.se/decwindows/
  - ftp://ftp.vms.stacken.kth.se/mirrors/decwindows/
  - http://www.multimania.com/pmoreau/decw/

  Other FTP mirror sites for the DECwindows archive include:

  - axp.psl.ku.dk (Multinet)
  - ftp2.cnam.fr (MadGoat)

  ftp.x.org (in /contrib/vms) is effectively not a mirror, but it does have various OpenVMS packages from the DECwindows archive.

  A list of the http mirror sites for the DECwindows archive is available at:

  - http://axp616.gsi.de:8080/wwwar/cena/decwindows/cena.html

  Various of these packages are also available on the Freeware.

- ImageMagick is an X11 package for display and interactive manipulation of images. The package includes tools for image conversion, annotation, compositing, animation, and creating montages. ImageMagick can read and write many of the more popular image formats (e.g. JPEG, TIFF, PNM, XPM, Photo CD, etc.).
  - http://www.imagemagick.org/

  Versions of ImageMagick are also included on the Freeware.

- XV is a image viewing and editing tool and is available from:
    - ftp://ftp.cis.upenn.edu/pub/xv
    - http://www.sanface.com/
    - ftp://www-pi.physics.uiowa.edu/~dyson/xv/
- Many software packages are available for displaying various bitmap files (.JPG, .GIF, .BMP, etc) on OpenVMS. Xloadimage, Xli, Xv, ImageMagick are the most common tools used under OpenVMS. Various web browsers such as Mozilla (qv) can also display various file formats on OpenVMS. You can find some of these tools at the DECwindows Archive:
    - http://www.multimania.com/pmoreau/decw/index.html
    - http://www.multimania.com/pmoreau/decw/images.html
- GHOSTSCRIPT (gs) and GHOSTVIEW (gv) are available from the Freeware V5.0 and Freeware V6.0 distributions:
    - http://www.hp.com/go/openvms/freeware/

  Versions are also available on the Freeware.

  Also see:
    - http://www.cs.wisc.edu/~ghost/gnu/
- XPDF, a DECwindows viewer for PDF (Adobe Acrobat) files, is available from:
    - http://www.foolabs.com/xpdf/

  Software and installation kits for XPDF are also available on the Freeware.

  Beware: the XPDF tool included on OpenVMS Freeware V4.0 is dated, and has been found to have various bugs. Use the Freeware V5.0 version of the XPDF kit, or (better) later.

  A Java-based PDF viewer is available from Adobe, and is known to operate on recent OpenVMS Alpha releases:
    - http://www.adobe.com/
- Various OpenVMS-related tools—both freeware and shareware—such as txt2pdf—are available from at:
    - http://www.sanface.com/

- The MPEG library version 1.1 is available for OpenVMS VAX and Alpha at
  - ftp://ftp.x.org/contrib/vms/mpeglib-11-vms.readme
  - ftp://ftp.x.org/contrib/vms/mpeglib-11-vms.zip

  Various MPEG players and related tools are available on the Freeware.
- An Archie clone is available at INFO.CS.PUB.RO. Telnet to that machine, and login as VMSARCI. It contains now listings for over 30 ftp servers with over 14 gigabytes of OpenVMS software.

  The most useful commands are LIST, which generates a list of scanned ftp servers, and FIND string, whichs looks for a file containing "string" in the name; the search modes are only "substring" [default] and "exact", and regex search is not supported (so FIND EMACS will work, but FIND *EMACS* or FIND *EMACS*.* will not). The search is case-insensitive.

  The maintainer of the site (stfp@roipb.cs.ipb.ro) requests that anyone submit notifications of FTP servers with OpenvMS software that are not listed on the site.
- The Levitte (extended :-) Family (and OpenVMS) website:
  - http://www.levitte.org/
  - http://www.levitte.org/~ava/
- Robert Alan Byer's OpenVMS Hobbyist Systems, including OpenVMS public domain software and various ports:
  - http://www.ourservers.net/
- CalTech Software Archives:
  - http://saf.bio.caltech.edu/pub/software/openvms/aaa_contents.txt
- DJE Systems Website (David J. Dachtera)
  - http://www.djesys.com/freeware/vms/
- Web Servers:

  Apache Web server (HP Secure Web Server (SWS, formerly CSWS)):
  - http://www.openvms.compaq.com/ebusiness/Technology.html
  - http://www.er6.eng.ohio-state.edu/~jonesd/apache/1_3_9/

  The SOAP Toolkit is available at the OpenVMS website.

  OSU Web server

  - http://www.er6.eng.ohio-state.edu/www/doc/serverinfo.html
  - http://www.kjsl.com/archives/
  - email list: VMS-WEB-daemon-Request@KJSL.COM

  WASD Web server

  - http://wasd.vsm.com.au/wasd/

  Purveyor Web server:

  - email list: listserv@cjis.ci.lincoln.ne.us, no subject, one line message: SUBSCRIBE PURVEYOR

  FastCGI software is available from:

  - http://www.DigitalSynergyInc.com/

- CD-R (CD-Recordable) media tools are available, please see Section 9.7.
- Grace (WYSIWYG 2D plotting tool)
  - http://plasma-gate.weizmann.ac.il/Grace/
- The POV-Ray ("Persistance of Vision" Raytracer) ray-tracing graphics package is available on the OpenVMS Freeware.
- Majordomo mailing list handler:
  - http://www.openvms.compaq.com/ebusiness/Technology.html
- PINE (OpenVMS tools for sending and receiving MIME mail):
  - ftp://ftp2.kcl.ac.uk/pub/vms/pine-vms/
  - http://www.agh.cc.kcl.ac.uk/files/vms/pine-vms/

  A MIME tool is available in OpenVMS V7.2 and later.

  Also see the mmencode base64 encode and decode available at:

  - http://nucwww.chem.sunysb.edu/htbin/software_list.cgi
- Menufinder (menu-driven system management environment):
  - http://www.itre.com/mf/download.html
- SYSLOGCLIENT (a client for processing SYSLOG requests) has been provided for download by Mark Hemker at:
  - http://home.insightbb.com/~hemker/vms.html
- tcgmsg, pvm, mpi, linda:
  - ftp://v36.chemie.uni-konstanz.de/tcgmsg_vms/tcgmsg_vms.zip

- OpenVMS software that can control a Tripp-Lite Uninterruptable Power Supply (UPS) is available from:
  - http://saf.bio.caltech.edu/pub/software/openvms/tcontrol.zip

  UPShot web-based software for controlling a UPS is available from:
  - http://www.tmesis.com/apc/beta.htmlx

  OpenVMS software for controlling Liebert UPS devices are available from the Liebert website:
  - http://www.liebert.com/apc/beta.htmlx
- Examples of using the OpenVMS Foreign MAIL interface are available at:
  - http://www.hhs.dk/anonymous/pub/vms/collection/foreignmail.zip
  - http://www.hhs.dk/anonymous/pub/vms/nbl/nbl.zip
- For tools to manage or to search your OpenVMS MAIL file, see:
  - http://vms.process.com/scripts/fileserv/fileserv.com?MLSEARCH
- AscToHTM attempts to convert any plain text file to HTML, while AscToTab restricts itself to files that are plain text tables. (Versions are also availabe on the OpenVMS Freeware).
  - http://www.jafsoft.com/asctohtm/index.html
  - http://www.jafsoft.com/asctotab/index.html
- Information on the SAMBA package, a package that provides SMB-based Microsoft Windows PC disk and print services for OpenVMS, is available at:
  - http://ifn03.ifn.ing.tu-bs.de/ifn/sonst/samba-vms.html
  - http://www.samba.org/
  - http://www.hp.com/go/openvms/freeware/

  To subscribe to the SAMBA-VMS mailing list e-mail listproc@samba.org with no subject line and the following single line of text:

  ```
  subscribe samba-vms Your Full Name
  ```

  Also see:
  - http://lists.samba.org/
- The Perl language is available for OpenVMS, see Section 13.9 for details.

- XML is available for OpenVMS.

  Source code of an XML Parser is available from Oracle.

  Also see:

  - http://www.python.org/sigs/xml-sig/

  An XML parser is available as part of OpenVMS V7.3 and later.
- Python kit, and resource, and documentation sites include:
  - http://www.pi-net.dyndns.org/anonymous/kits/
  - http://vmspython.dyndns.org/
  - http://www.python.org/
- GTK+ (The GIMP GUI Tookit) for OpenVMS:
  - http://www.openvms.compaq.com/ebusiness/Technology.html
- The OpenVMS Porting Library now available, and is intended to permit easier porting of C and C++ applications from UNIX systems to OpenVMS:
  - http://www.openvms.compaq.com/ebusiness/Technology.html

  GTK is also available.
- Mlucas (specialized FFT):
  - ftp://hogranch.com/pub/mayer/README.html
- Tools to monitor the terminals and the activity of other OpenVMS users (in addition to existing auditing capabilities in OpenVMS) are available. Peek and Spy (Networking Dynamics) and Contrl (Raxco) are two of the commercial packages, while the freeware Supervisor package is available on OpenVMS VAX.
  - http://www.networkingdynamics.com/
  - http://www.raxco.com/
- Python for OpenVMS:
  - http://www.python.org/

  Also see the OpenVMS Freeware.
- Various packages for OpenVMS:
  - http://richj.home.mindspring.com/richware/index.html
  - http://www3.sympatico.ca/n.rieck/links/cool_vax_vms.html

- TSM (Terminal Server Manager) is available via:
  - Look at Freeware V5.0 at http://www.hp.com/go/openvms/freeware/
  - Beware: The TSM saveset shipped on the Freeware V5.0 disk media is known to be corrupted. Download a new copy of the saveset from the Freeware V5.0 FTP server or from the Freeware V5.0 website.
- TCL for OpenVMS:
  - ftp://sapodilla.rsmas.miami.edu/pub/VMS-tcl/
- make, gmake, mmk and other build tools are available on the Freeware.
- An OpenVMS port of the ht://Dig web search engine is available at:
  - ftp://ftp.pdv-systeme.de/vms/
- A mySQL database client is available at:
  - http://mysql.holywar.net/Downloads/MySQL-3.22/mysql-3.22.25-clients-vms.zip
  - http://mysql.holywar.net/Downloads/MySQL-3.22/mysql-3.22.25-clients-vms.readme

  For additional information related to the mySQL port, please search the comp.os.vms newsgroup archives. A MySQL port is also available on the OpenVMS Freeware.
  - http://www.hp.com/go/openvms/freeware/
- If you need to change the file modification date and are looking for a utility such as the UNIX touch tool, look at DFU on the OpenVMS Freeware (DFU SET or simular), or use an existing DCL commands such as:

  ```
  $ SET FILE/PROTECT=(current_protection_mask) [...]*.*
  ```
- A table listing translations between UNIX shell and OpenVMS DCL commands was posted to comp.os.vms by Christopher Smith some time ago. This page should be available from the google newsgroup archives.

  For information on and the status of the OpenOffice port for OpenVMS, please see:
  - http://www.oooovms.dyndns.org

- The UNIX touch tool is available via various means:

  ```
  $ RENAME filename.ext;version *
  ```

  - http://nucwww.chem.sunysb.edu/helplib/@hvmsapps/TOUCH

  MadGoat FILE tool (see the MadGoat archives)
- use /REVISION_DATE or /CREATION_DATE
- The DFU tool (see the OpenVMS Freeware)
- The pair:

```
$ set file 'p1' /acl=(ident=[system],access=none)
$ set file 'p1' /acl=(ident=[system],access=none) /delete
```

```
$ SET FILE/VERSION=0
```

- The following touch hack:

```
$!   Command procedure SETDATE.COM
$!
$!   Changes the DATES for an input file to a
$!   file named OUTFILE.
$!
$assign/nolog 'p1' outfile
$convert/fdl=sys$input 'p1' outfile:
date
creation   01-apr-2010
expiration 01-Apr-2012
revision   01-Apr-2011
backup
...
```

- The following RMS system service sequence:
  - sys$open( ), with the XABRDT XAB structure chained.
  - set the desired values within the XABRDT XAB.
  - sys$close( )

Various OpenVMS tools and utilities are available at:

- http://wwwvms.mppmu.mpg.de/vmssig/src/

OpenVMS ports of the xmcd and MPlayer tools have also been reported as available.

13.2 Where can I find UNIX tools for OpenVMS?

There are OpenVMS DCL and UNIX shell command comparison tables posted at:

- http://wwwvms.mppmu.mpg.de/vmsdoc/UNIX_VMS_CMD_XREF.HTML
- http://www.mcsr.olemiss.edu/unixhelp/VMStoUNIX.html

The GNV package (Section 13.2.6) and the various C library calls available in current/recent libraries are the replacement for the POSIX package.

13.2.1 C system and library routines

Common C system and library routines are present in the DEC C run-time library, which is available for V5.5 and later, and is shipped in V6.1 and later. DEC C is the upgrade for VAX C, DEC C and VAX C can coexist on the same system OpenVMS VAX system, and both compilers can be enabled via the "C" license PAK.

Also see SYS$EXAMPLES:, and (if either is installed) the DECW$EXAMPLES: and TCPIP$EXAMPLES (or prior to V5.0, UCX$EXAMPLES:) areas.

The HP C Run-Time Library documentation is now part of the OpenVMS Operating System Documentation, and separate from the HP C compiler documentation.

- http://www.hp.com/go/openvms/doc/

Additionally, there have been large-scale increases to the capabilities and features available within recent HP C compilers. If you have not taken the opportunity to skim the current C compiler documentation, you may well be surprised with the sheer volume of new C capabilities discussed there.

13.2.2 X Windows utilities and routines

Various X Windows utilities are available for DECwindows. DECwindows is an implementation of the X Windows environment and libraries, and provides various libraries, and provides various desktop interfaces, including COE, Motif, and XUI.

xwd, xev, mosaic web browser, xrdb, bmtoa and atobm, xpr, ico, etc. are available. Look in DECW$UTILS: in DECwindows Motif V1.2-3 and later. Also see DECW$EXAMPLES: for example X and C programs.

Miscellaneous tools and examples are also available. Examples include the older DWAUTH (X Windows SYSUAF authorize-like tool) tool, various versions of grep, fgrep, yacc, vmstar, uuencode, gawk, etc. html tools, the mx SMTP mail exchange package, X windows flight simulator, the mxrn X windows news reader, the OSU HTTPD WWW server, a WWW gopher browser, Castle Wolfenstein (Wolf3D), etc. are all on the various OpenVMS Freeware distributions. (Also see the GNV package (Section 13.2.6) for related materials, APIs, and tools.)

13.2.3 TCP/IP Tools and Utilities for OpenVMS?

TCP/IP Services (formerly known as UCX) contains tools such as ping, uuencode, smtp, snmp, rcp, nfs, tnfs, etc.

OpenVMS V6.2 and later includes DCL-integrated support for various IP tools, with commands such as SET HOST/TELNET, and COPY/FTP. This interface requires the installation of an IP stack, and TCP/IP Services (UCX) V3.3 and later as well as any then-current or now-current third-party IP stack can be used. Once the IP stack is installed and configured, the DCL command qualifiers such as /FTP, /RCP, /RLOGIN, /TELNET, and /TN3270 are available on various DCL commands including DIRECTORY.

Various C programming examples in TCPIP$EXAMPLES and (on releases prior to V5.0) in UCX$EXAMPLES:.

13.2.4 The vi text editor

vile, vim and elvis are all clones of the vi text editor, and all operate on OpenVMS.

Versions of vile are available on the Freeware and at:

- http://www.clark.net/pub/dickey/vile/vile.html

vim: vi improved

- http://www.polarfox.com/vim/

13.2.5 The Emacs Text Editor

OpenVMS ports of versions of the Emacs text editor can be found on various OpenVMS Freeware distributions (eg: OpenVMS Freeware V7.0 has a port of Emacs 21.2), and at various web sites including:

- http://www.glug.org/people/ttn/software/emacs-for-vms/

''Emacs isn't a text editor, it's a way of life.''

For comparisions of OpenVMS text editors and UNIX, see the following:

- http://www.unh.edu/cis/docs/vms-to-unix/Emacs/cheat-sheet.html

13.2.6 GNV: Various GNU tools

Information on the GNU on VMS (GNV; ''GNU's Not VMS'') Project, which aims to port and provide GNU software (bash, flex, bison, tar, grep, gcc emulation, etc) to OpenVMS, is available at:

- http://gnv.sourceforge.net/

Software info:

- http://vms.gnu.ai.mit.edu/software/

Software archive:

- ftp://vms.gnu.ai.mit.edu/gnu-vms/software/

Various GNU tools are also available on the Freeware.

13.2.6.1 GCC compiler

Yes, gcc is available for OpenVMS Alpha and OpenVMS I64.

- gcc compilation compatibility for OpenVMS Alpha and I64 is available as a component of the GNV package. This causes the HP C compiler to emulate the interface and compilation behaviours of gcc.
  http://h71000.www7.hp.com/opensource/opensource.html#gnv
- GNV Home Page
  http://gnv.sourceforge.net/
- GCC Home Page - GNU Project - Free Software Foundation (FSF)
  http://gcc.gnu.org/

- GNU/FSF
  ftp://ftp.tmk.com/vms-freeware/gcc-for-alpha/
- Help GCC - SUNY NCSB
  http://nucwww.chem.sunysb.edu/helplib/@hvmsapps/GCC
- Redhat’s gcc Archive
  ftp://ftp.mirror.ac.uk/sites/sources.redhat.com/ftp/gcc/releases
- Malmberg’s Ftp Service - gcc281_u
  ftp://ftp.qsl.net/pub/wb8tyw/gcc281_u/
- updated header-files for GNU C 2.8
  ftp://ftp.qsl.net/pub/wb8tyw/gcc281_u/

In addition to gcc, the HP C compiler and other development tools are part of the OpenVMS Hobbyist licensing program for non-commercial users, and these and other tools are available to commercial developers via the HP DSPP partner program. (See Section 2.15 for information on DSPP.)

13.3 What is the status of PL/I on OpenVMS?

Kednos now owns and supports the former DIGITAL PL/I compiler and run-time support on OpenVMS, and is the contact for product status, support and associated plans.

As of this writing, (older) versions of the PL/I run-time library and associated supporting images remain available on OpenVMS VAX and on OpenVMS Alpha, including the DECmigrate VEST translated images run-time support on OpenVMS Alpha, though neither the PL/I shareable image nor the DECmigrate (AEST) translated image support for the PL/I run-time are available (from HP) on nor present on OpenVMS I64 systems. Newer versions of the PL/I run-time library may be (are?) available from Kednos for various OpenVMS platforms; please contact Kednos for product details and availability.

- http://www.kednos.com

13.4 Where can I get the Mozilla Web Browser?

Mozilla.org is an open source organization providing HTML-related tools; software that is the basis for various utilities including the Mozilla web browser and the Secure Web Browser (SWB) package.

OpenVMS Engineering is continuously porting Mozilla.org's web browser to OpenVMS, and OpenVMS ports of the current Mozilla baselevels and releases are available. The OpenVMS Mozilla port includes the web browser, the mail client, the Composer HTML editor, an IRC chat client, a netnews (NNTP) reader, and various other tools.

The Mozilla web browser download and the development and release schedules for this and for other Mozilla-related tools are available at:

- http://www.mozilla.org/

The available Secure Web Browser (SWB) kit is a packaged version of the Mozilla Web Browser, and typically the SWB version number matches the underlying Mozilla version.

A hardware configuration appropriate for Mozilla generally involves an OpenVMS Alpha system with an EV56 Alpha microprocessor, or an EV6 or more recent processor, and with 256 megabytes of system memory. The performance of Mozilla on EV5-based and earlier Alpha microprocessor systems is generally viewed as inadequate, this due to the extensive use of an Alpha instruction subset that is first available with the EV56 microprocessor generation.

Mozilla is not available for OpenVMS VAX.

Various versions of the Netscape Navigator web browser are based on the Mozilla code-base.

13.5 Where can I get Java for OpenVMS?

Java is available on and is included with OpenVMS Alpha, starting with the OpenVMS Alpha V7.2 and later releases. Java download kits are available for OpenVMS Alpha V7.1 and later releases.

Java is not available on OpenVMS VAX. As for why: the Java language definition requires a floating point format (IEEE) that is not native to VAX, and this would require the emulation of all floating point operations within Java applications. Further, the C source code used to implement for Java itself is heavily dependent on passing IEEE floating point values around among the many internal subroutines, and adding support for

VAX would entail changes to the HP C compiler for OpenVMS VAX—and specifically to the VAX VCG code generator that is used by HP C on OpenVMS VAX systems—in order to add support for passing IEEE-format floating point doubles around. Alternatively, extensive changes to the Java source code to remove the assumption that the double is an IEEE floating point value.

There are currently no plans to make a version of Java available for OpenVMS VAX. (A prototype version of Java was created for OpenVMS VAX, and performance was found to be inadequate. At best.)

If Java2 or other environment lifts the requirements for IEEE floating point as part of the language definition, this decision may be revisited.

If you are having problems with Display Postscript, you need to upgrade your Java kit—1.2.2-3 and later remove the requirement for Display Postcript extensions, and Java 1.2.2-3 is required with DECwindows V1.2-6 and later.

For additional information on Java for Alpha systems, please see the OpenVMS documentation (V7.2 and later), and the following site:

- http://www.compaq.com/java/alpha/index.html

HP Secure Web Server (SWS, formerly CSWS) includes CSWS_JAVA, which provides the following Apache Tomcat technologies: JavaServer Pages 1.1, Java Servlet 2.2, and MOD_JK. Also available is CSWS_PHP, a PHP implementation. (SWS is based on the Apache web server. See SOFT1.)

13.6 Obtaining user input in DCL CGI script?

If you choose to use the GET method, then the form data is available in the DCL symbol QUERY_STRING, in URL-encoded format.

If you use the POST method, then you need to read the form data from stdin. For a DCL CGI script running under the Netscape FastTrack web server, you can read the data using the following READ command:

```
$ READ SYS$COMMAND postdata
```

to read the information in.

The following describes the use of DCL command procedures as CGI scripts with the OSU web server:

- http://www.levitte.org/~ava/cgiscripts_other.htmlx

DCL CGI is also discussed in the Writing Real Programs in DCL book, and in the Ask The Wizard website.

13.7 How can a batch job get its own batch entry number?

To have a batch procedure retrieve its own batch entry number, use the following:

```
$ Entry = F$GETQUI("DISPLAY_ENTRY", -
    "entry_number","display_entry","this_job")
```

Remember that the entry numbers issued by the OpenVMS Job Controller are always opaque longword values. Do *not* assume you know the format of the entry number, nor the range of entry numbers you might see, nor the algorithm that is used to assign enty numbers. You should simply assume *opaque longword*.

13.8 How do I convert to new CMS or DTM libraries?

A change was made to the format of the CMS database for CMS libraries starting with V3.5-03—to ensure that earlier versions of CMS are unable to access the database once the ‘‘conversion’’ to V3.5-05 and later is made, you must issue the following two commands when upgrading from V3.5-03 and prior. (The only differences between CMS version V3.5-03 and CMS version V3.5-05 involve changes to ensure that no earlier version of CMS can access the ‘‘converted’’ database, and corrupt it.)

To perform the ‘‘conversion’’, issue the following commands for each CMS library present:

```
$ RENAME disk:[directory]00CMS.* 01CMS.*
$ COPY NLA0: disk:[directory]00CMS.CMS
```

The new file 00CMS.CMS must have the same security settings as the 01CMS.CMS file, and is created solely to ensure continued compatibility with tools that expect to find a 00CMS.CMS file (eg: various versions of the Language-Sensitive text editor LSEDIT).

If you choose to install and use the longer variant names support that is available with CMS V4.1 or later, you cannot mix earlier CMS versions within a cluster. If you attempt to mix older and newer versions, you will typically see the following BADLIB and BADTYPSTR error sequence when accessing the CMS library from the older CMS versions:

```
%CMS-F-BADLIB, there is something wrong with your library
-CMS-F-BADTYPSTR, header block type is 145; it should be 17
```

Please see the CMS V4.1 release notes for additional details on this.

To perform the equivalent ‘‘conversion’’ for DEC Test Manager (DTM) V3.5 and prior versions to V3.6 and later versions, issue the following DCL commands for each DTM library present:

```
$ RENAME disk:[directory]00DTM.* 01DTM.*
$ COPY NLA0: disk:[directory]00DTM.DTM
```

Like CMS, this change is intended to prevent older versions of DTM from accessing newer libraries, and corrupting the contents. Like CMS, once the libraries are renamed, they cannot and should not be renamed back to the older names; like CMS, the changes are not downward-compatible.

To convert version 1 (ancient) DTM and CMS libraries forward, please see the DTM CONVERT and the CMS CONVERT commands.

13.9 Where can I get Perl for OpenVMS?

OpenVMS support is included in the standard distribution of Perl, the popular scripting language created by Larry Wall. In addition to nearly all of the functionality available under Unix, OpenVMS-specific Perl modules provide interfaces to many native features, as well as access to Oracle, Ingres, and Sybase databases via the Perl DBI available on OpenVMS.

A website useful for getting started with Perl on OpenVMS—where you will find such things as download links, instructions, auxiliary tools, and sample scripts—is available at:

- http://www.sidhe.org/vmsperl

If you have a C compiler, the best way to obtain Perl is to download and build it yourself. The latest production quality source kit is available from:

- http://www.perl.com/CPAN/src/stable.tar.gz

You will need GUNZIP and VMSTAR (both available from the OpenVMS Freeware CD, or from other sites) to unpack the archive; once you’ve done that, read the instructions in the README.vms file.

Binary distributions for most Alpha and VAX environments are available on the OpenVMS Freeware CD-ROM and from various websites, including the following:

- http://www.sidhe.org/vmsperl/prebuilt.html
- http://www.hp.com/go/openvms/freeware/

During active Perl development cycles, test kits are sometimes found at: from:

- ftp://ftp.sidhe.org/

Watch the mailing list (see below) for details on experimental releases.

Charles Lane maintains pages on how to write CGI scripts in Perl for the OSU HTTP server, as well as more general tips, tricks, and patches for building and running Perl on OpenVMS:

- http://www.crinoid.com/crinoid.htmlx

There are OpenVMS-specific Perl modules that implement interfaces to a subset of the VMS System Services. With these modules, you can get (and often set) device, job, queue, user, system, and performance information. The lock manager, RMS indexed files, screen management utilities, and Intracluster Communication Services are also accessible via Perl. The relevant modules are all available from:

- http://www.perl.com/CPAN/modules/by-module/VMS

To subscribe to the OpenVMS Perl mailing list (a discussion forum for both user support and new development), send an email message to vmsperl-subscribe@perl.org

The mailing list archives may be searched at:

- http://www.xray.mpe.mpg.de/mailing-lists/vmsperl

13.10 Obtaining the DECmigrate (AEST or VEST, and TIE) translator?

The DECmigrate image translation family provides tools that translate OpenVMS VAX images for use on OpenVMS Alpha, and OpenVMS Alpha images for use on OpenVMS I64, Details are available at:

- http://h71000.www7.hp.com/openvms/products/omsva/omsva.html

VEST is the name sometimes given to the DECmigrate translation tool for VAX images, AEST is the name given to the Alpha translation tools, and TIE names the DECmigrate run-time environment within OpenVMS. (If you’ve ever noticed images with filenames ending with _TV and wondered what this meant, these images are part of TIE.) And yes, you can use AEST to re-translate images that were translated using VEST; you can perform a second translation of a VAX image.

Please see Section 13.12 for related information. Please see the website for the most current details on availability and plans and status of translations for OpenVMS I64 platforms.

13.11 Where can I get Zip, Unzip, self-extracting zip, etc?

Many packages are provided in ZIP, GZIP, or BZIP2 format, which requires you to acquire the associated unzip tool to unpack it. You can get ZIP and UNZIP and related and similar tools from the following areas:

- http://www.hp.com/go/openvms/freeware/ . Look in the [000TOOLS...] and [*ZIP*...] directories.
- ftp://ftp.process.com/vms-freeware/unzip.alpha_exe
- ftp://ftp.process.com/vms-freeware/unzip.vax_exe
- http://zinser.no-ip.info/www/vms/sw/zip.htmlx
- http://www.djesys.com/zip.html
- http://www.djesys.com/unzip.html
- The master Info-Zip web site is at http://www.info-zip.org/ . OpenVMS ports of current versions of zip and unzip are typically available at this web site, as is a mailing list.

Freeware V4.0 [000TOOLS...]*ZIP*.EXE

The Freeware V4.0 [000TOOLS...] pre-built versions of ZIP will erroneously return BILF errors on OpenVMS V7.2 and later. This is not the only error lurking within these pre-built versions, just the most obvious. Accordingly, please use one of the far more current versions that are now readily available, whether on the most recent Freeware distribution, or from one of the sites listed above.

Do not use the Freeware V4.0 [000TOOLS...]*ZIP*.EXE images.

Directions for creating and using the sfx self-extracting zip file compression mechanism are available in the unzip kit that is available at:

- Look in a recent unzip* directory at http://www.hp.com/go/openvms/freeware/
- With the UNZIP542 directory from Freeware V5.0, look for the file UNZIPSFX.TXT.

If you want to build the zip images for yourself (eg: for an older OpenVMS version), pull over the entire contents of a recent unzip and unzip directory, or Info-Zip directory, or visit one of the web sites. With most OpenVMS ports of the tools, find and invoke LINK.COM. No compilers are needed, as objects are provided with most distributions.

HP OpenVMS Engineering uses a tool known as FTSV for creating self-extracting compressed files using the OpenVMS DCX compression tools, as seen with various OpenVMS ECO (patch) kits. (sfx typically provides better compression than does DCX.) FTSV and FTSO are available on Freeware V7.0, for OpenVMS VAX and OpenVMS Alpha. Due to changes in the image headers, no version of FTSV is presently available for OpenVMS I64.

13.12 Are VAX Hardware Emulators Available?

Software-based emulators of the VAX architecture and for specific VAX hardware platforms are available from various sources:

- Software Resources International (SRI) CHARON-VAX
  http://www.softresint.com/
- Tim Stark’s TS10
  http://sourceforge.net/projects/ts10/
- Bob Supnik’s Trailing Edge
  http://simh.trailing-edge.com/

VAX emulators that operate on PC systems and/or on OpenVMS Alpha systems are available. For information on an alternative to using a VAX emulator— on the available DECmigrate VAX executable image translator—please see Section 13.10.

14 Hardware Information

If you are searching for something here,
please consider using the text-format FAQ.

14.1 What are the OpenVMS differences among VAX, Alpha, and IA-64?

In terms of software, very few. As of OpenVMS V6.1, the OpenVMS VAX and OpenVMS Alpha platforms achieved ‘‘feature parity’’. Subsequent work has seen significant enhancements and new features added on OpenVMS Alpha. OpenVMS I64 started with ‘‘feature parity’’ with OpenVMS Alpha at the V8.2 release, and OpenVMS Alpha and OpenVMS I64 are based on and built from the same source pool. (There do exist low-level platform-specific differences, and platform-specific code is present within the shared source code pool, obviously.) Most applications can just be recompiled and run.

Some differences to be aware of:

- The default double-precision floating type on OpenVMS Alpha is VAX G_float, whereas on VAX it is usually D_float. D_float is available on Alpha, but D_float values are converted to G_float for computations and then converted back to D_float when stored. Because the G_float type has three fewer fraction bits than D_float, some applications may get different results. IEEE float types are also available on OpenVMS Alpha.
- The preferred floating point format on the Alpha and on the IA-64 architectures is IEEE.
- Data alignment is extremely important for best performance on OpenVMS Alpha and on OpenVMS I64. This means that data items should be allocated at addresses which are exact multiples of their sizes. Quadword alignment will offer the best performance, especially for character values and those smaller than 32 bits. Compilers will naturally align variables where they can and will issue warnings if they detect unaligned data items.
- HP C is the only C compiler HP offers on OpenVMS Alpha and on OpenVMS I64, and is a direct descendant of Compaq C and DEC C on OpenVMS Alpha. HP C is highly compatible with DEC C on OpenVMS VAX, but does differ somewhat in its syntax and support

when compared with the older VAX C compiler most OpenVMS VAX programmers are traditionally familiar with. Read up on the /EXTERN_MODEL and /STANDARD qualifiers to avoid the most common problems, and see the documentation in the DEC C for OpenVMS VAX manuals around migrating from VAX C to DEC C. (In addition to HP C, there have been open-source ports such as Gnu C available for OpenVMS.)

- The page size on Alpha and IA-64 systems is variable, but is at least 8 kilobytes. This can have some effect on applications which use the $CRMPSC system service as well as on the display of available memory pages. The page size is available from $GETSYI using the SYI$_PAGE_SIZE itemcode.

There are also a number of manuals which discuss migration to OpenVMS Alpha and to OpenVMS I64 available in the OpenVMS documentation, both in the main documentation and (depending on the age of the manuals involved) in the archived documentation section.

As mentioned earlier, more recent OpenVMS Alpha and OpenVMS I64 releases have added features and support that are not available on OpenVMS VAX. Salient additions include the following:

- 64-bit addressing in OpenVMS Alpha V7.0 and later, and on OpenVMS I64.
- Multi-host SCSI support (SCSI TCQ) in V6.2 and later
- PCI support (platform-dependent)
- OpenVMS Galaxy (vPars) support in OpenVMS Alpha V7.2 and later

Please see Section 14.4.5 for Intel Itanium terminology.

14.2 Seeking performance information for Alpha (and VAX) systems?

HP makes a wide range of performance documents available through its FTP and WWW Internet servers (see Section 3.2).

The following contain information on Integrity, Alpha and VAX products, with the VAX information largely accessable via archive-related links at the Alpha-related product web pages:

- http://www.hp.com/go/server/
- http://www.compaq.com/alphaserver/vax/index.html

The following sites are reachable via the AlphaServer information pages, and contain information on various retired VAX and Alpha products:

- http://www.compaq.com/alphaserver/archive/index.html
- http://www.compaq.com/alphaserver/performance/perf_tps.html

Also see CPU2000:

- http://www.spec.org/osg/cpu2000/
- http://www.spec.org/osg/cpu2000/results/cpu2000.html

14.3 Console Commands, Serial Lines, and Controls?

This section contains information on VAX and Alpha consoles, and details related to console commands, serial lines, and configuration settings.

14.3.1 What commands are available in the Alpha SRM console?

In addition to the normal BOOT commands and such (see Section 14.3.5.2 for some details) and the normal contents of the console HELP text, operations such as I/O redirection and floppy disk access are possible at the SRM console prompt:

1. Format a FAT floppy, and insert it into the AlphaStation floppy drive.
2. Perform the following at AlphaStation SRM Console :

```
>>> show * > env.dat
>>> show conf > conf.dat
>>> cat env.dat > fat:env.dat/dva0
>>> cat conf.dat > fat:conf.dat/dva0
```

3. You may use the SRM ‘‘ls’’ command to display the contents of the floppy.

```
>>> ls fat:env.dat/dva0
 >>> ls fat:conf.dat/dva0
```

4. You can now transfer the FAT-format floppy to another system.

14.3.2 What does SRM mean? What is PALcode?

The abbreviation SRM is derived from the Alpha System Reference Manual, the specification of the Alpha architecture and the associated firmware.

PALcode is a name assigned to a particular set of functions provided by the SRM firmware. PALcode is used to provide low-level functions required by higher-level operating system or application software, functions which may not be directly available in Alpha hardware. PALcode is implemented using available Alpha instructions and using the Alpha processor, though PALcode operates in a mode which simplifies programming. PALcode is also permitted access to processor-specific and otherwise internal features of a particular Alpha microprocessor implementation; microprocessor-specific features which are not easily accessable to operating system or application code.

14.3.3 Alpha COM ports and VAX console serial line information?

This section contains information on the Alpha COM communication ports, and related settings, as well as on the VAX console bulkhead and VAX console serial line connection.

14.3.3.1 Which terminal device name is assigned to the COM ports?

COM2 is normally TTA0:. COM1 is normally TTB0: if the Alpha workstation is booted with the SRM console environment variable set to graphics, and is OPA0: if the console is set to serial.

On the DEC 2000 series (sometimes incorrectly known by the name of the system as sold for Microsoft Windows NT Alpha; as the DECpc 150 AXP series) on older OpenVMS Alpha releases, COM1 through COM4 are known as OPA0: through OPA3:. On all current OpenVMS releases, these ports are serviced by the terminal driver and not by the console OPDRIVER driver.

Often the easiest way to determine the OpenVMS terminal name assigned to the port is to connect a terminal, log in interactively, and look at the output of SHOW TERMINAL. (Device names can vary by OpenVMS version, as well as by the SRM console environment variable selection.)

For serial console hardware and related information, and for pin-outs and related information, please see Section 14.3 and Section 14.26.

14.3.3.2 Which serial port is the console on the MicroVAX 3100?

Just to keep life interesting, the MicroVAX 3100 has some ‘‘interesting’’ console ports behaviours based on the setting of the BREAK enable switch. When the console is not enabled to respond to BREAK, MMJ-1 is the console port. MMJ-3 will (confusingly) output the results of the selftest in parallel with MMJ-1. When the console is enabled to respond to BREAK, MMJ-3 becomes the console port, and MMJ-1 will (confusingly) output the results of selftest in parallel with MMJ-3.

14.3.3.3 How can I set up an alternate console on a VAXstation?

Most VAXstation series systems and a few Alpha series systems have a switch – most often labeled S3, largely for historical reasons—that enables one of the serial lines as the system console device; as OPA0:. This disables console output to the graphics display. (For a related behaviour, please see Section 11.10.)

All VAXstation 3100 series systems provide a S3 slide switch, though the oldest may be missing the cut-out through the enclosure that provides access to the switch. The slide switch is located near the diagnostic LED display. (The slide switch is accessable with the cover removed.)

Various members of the DEC 3000 series Alpha systems also have a similarly-labled S3 switch for selection of the alternate console.

The particular port that becomes the console can vary. The printer MMJ connection is used on all VAXstation 3100 series. On VAXstation II, the console DB9 is used, rather than the graphics display. On most (all?) AlphaStation series systems, typically the COM1 serial port becomes the console.

Also see Section 14.3.6, Section 11.10, and Section 14.17. Beware the two different DB9 pin-outs; see Section 14.27 for related details.

For information on registering software license product authorization keys (PAKs), please see Section 5.6.2.

14.3.3.4 Please explain the back panel of the MicroVAX II

The MicroVAX-series console bulkhead interface was used with the KA630, as well as with the KA650 and KA655 processors.

There are three controls on the console bulkhead of these systems:

```
Triangle-in-circle-paddle: halt enable.
  dot-in-circle: halt (break) is enabled,
                 and auto-boot is disabled.
  dot-not-in-circle: halt (break) is disabled,
                 and auto-boot is enabled.

Three-position-rotary: power-up bootstrap behaviour
  arrow: normal operation.
  face: language inquiry mode.
  t-in-circle: infinite self-test loop.

Eight-position-rotary: console baud rate selection
  select the required baud rate; read at power-up.
```

There are several different bulkheads involved, including one for the BA23 and BA123 enclosures, and one for the S-box (BA2xx) series enclosure. The console bulkheads typically used either the MMJ serial line connection, or the MicroVAX DB9 (not the PC DB9 pin-out), please see the descriptions of these in section Section 14.26. For available adapters, see Section 14.27.

Also present on the console bulkhead is a self-test indicator: a single-digit LED display. This matches the final part of the countdown displayed on the console or workstation, and can be used by a service organization to determine the nature of a processor problem. The particular countdown sequence varies by processor type, consult the hardware or owner's manual for the processor, or contact the local hardware service organization for information the self-test sequence for a particular processor module. Note that self-tests 2, 1 and 0 are associated with the transfer of control from the console program to the (booting) operating system.

14.3.4 What are Alpha console environment variables?

Alpha systems have a variety of variables with values set up within the SRM system console. These environment variables control the particular behaviour of the console program and the system hardware, the particular console interface presented to the operating system, various default values for the operating system bootstrap, and related control mechanisms—in other words, ‘‘the environment variables provide an easily extensible mechanism for managing complex console state.’’

The specific environment variables differ by platform and by firmware version—the baseline set is established by the Alpha Architecture:

```
AUTO_ACTION ("BOOT", "HALT", "RESTART", any other value
assumed to be HALT),  BOOT_DEV, BOOTDEF_DEV, BOOTED_DEV,
BOOT_FILE, BOOTED_FILE, BOOT_OSFLAGS, BOOTED_OSFLAGS,
BOOT_RESET ("ON", "OFF"), DUMP_DEV, ENABLE_AUDIT ("ON",
"OFF"), LICENSE, CHAR_SET, LANGUAGE, TTY_DEV.
```

OpenVMS Galaxy (vPars) firmware can add console environment variables beginning with such strings as LP_* and HP_*, and each particular console implementation can (and often does) have various sorts of platform-specific extensions beyond these variables. These variables allow both vPars (virtual partitions) and lPars and lPars (logical partition) support; vPars is a generic name for soft partitioning constructs such as OpenVMS Galaxy, while lPars is a generic name applied to hard partitioning constructs.

The contents of a core set of SRM console environment variables are accessible from OpenVMS Alpha using the f$getenv lexical and the sys$getenv system service. (These calls are first documented in V7.2, but have been present in OpenVMS Alpha for many releases.) Access to arbitary SRM console environment variables is rather more involved, and not directly available to application software operating outside of kernel-mode.

14.3.5 What are the boot control flag values?

Integrity, VAX and Alpha primary bootstraps support flag values; a mechanism which permits the system manager to perform specific customizations or site-specific debugging of the OpenVMS system bootstrap. While very similar, there are differences among the boot flag implementations for the various architectures.

14.3.5.1 What are the I64 IPB boot flag values?

The OpenVMS I64 primary bootstrap flags are processed within the IA-64 primary bootstrap image IPB.EXE; within the SYS$EFI.SYS structures. The primary bootstrap boot flags are largely parallel to those of OpenVMS Alpha (see Section 14.3.5.2, though the console and the console mechanisms used to specify the boot command, the boot flags, and boot command options do differ markedly.

You can specify the boot flags via an EFI environment variable VMS_FLAGS , or via the boot alias boot options mechanism, or by appending the requested boot flags onto the specification of VMS_LOADER.EFI.

To set the bootstrap flags environment variable at the EFI shell prompt, use:

```
Shell> SET VMS_FLAGS "0,1"
```

When you register an EFI boot alias (please see Section 14.4.5 for Intel Itanium terminology), you will be asked if you want to enter boot options, and what type. To add boot flags to a boot alias, select Unicode as the boot options type, and enter an SRM-like options string, such as the conversational bootstrap selected by the following example:

```
-flages 0,1
```

For related information on managing EFI boot aliases from OpenVMS I64, please see Section 14.3.10.

When using VMS_LOADER.EFI to request boot flags, you will want to specify the invocation as follows:

```
fsn:\efi\vms\vms_loader -flags 0,1
```

The above shows a conversational bootstrap request.

Typical boot flags are listed in Table 14–1.

Table 14–1 I64 Conversational Bootstrap Flags

Bit	Example	Mnemonic	Description
0	0,1	CONV	Conversational bootstrap
1	0,2	DEBUG	Load SYSTEM_DEBUG.EXE (XDELTA)
2	0,4	INIBPT	Stop at initial system breakpoints
16	0,10000	DBG_INIT	Enable verbose bootstrap messages
17	0,20000	USER_MSGS	Enable additional bootstrap messages
17	0,200000	?	Request for a bootstrap from USB keydisk

For a conversational bootstrap of the OpenVMS I64 root SYS4 associated with the fs2: EFI file system device with full bootstrap messaging enabled, specify:

```
fs2:\efi\vms\vms_loader -flags 4,30001
```

14.3.5.2 What are the Alpha APB boot flag values?

The flags listed in Table 14–2 are passed (via register R5) to the OpenVMS Alpha primary bootstrap image APB.EXE. These flags control the particular behaviour of the bootstrap.

```
BOOT -FL root,flags
```

Table 14–2 Alpha Conversational Bootstrap Flags

Bit	Mnemonic	Description
0	CONV	Conversational bootstrap
1	DEBUG	Load SYSTEM_DEBUG.EXE (XDELTA)
2	INIBPT	Stop at initial system breakpoints if bit 1 set (EXEC_INIT)
3	DIAG	Diagnostic bootstrap (loads diagboot.exe)
4	BOOBPT	Stop at bootstrap breakpoints (APB and Sysboot)
5	NOHEADER	Secondary bootstrap does not have an image header
6	NOTEST	Inhibit memory test
7	SOLICIT	Prompt for secondary bootstrap file
8	HALT	Halt before transfer to secondary bootstrap
9	SHADOW	Boot from shadow set
10	ISL	LAD/LAST bootstrap
11	PALCHECK	Disable PAL rev check halt
12	DEBUG_BOOT	Transfer to intermediate primary bootstrap
13	CRDFAIL	Mark CRD pages bad
14	ALIGN_FAULTS	Report unaligned data traps in bootstrap

Table 14–2 (Cont.) Alpha Conversational Bootstrap Flags

Bit	Mnemonic	Description
15	REM_DEBUG	Allow remote high-level language debugger
16	DBG_INIT	Enable verbose boot messages in EXEC_INIT
17	USER_MSGS	Enable subset of verbose boot messages (user messages)
18	RSM	Boot is controlled by RSM
19	FOREIGN	Boot involves a foreign disk

If you want to set the boot flags ‘‘permanently’’, use the SET BOOT_FLAGS command, e.g.:

```
>>> SET BOOT_OSFLAGS 0,1
```

14.3.5.3 What are the VAX VMB boot flag values?

The flags described in Table 14–3 are passed (via register R5) to the OpenVMS VAX primary bootstrap image VMB.EXE. These flags control the particular behaviour of the bootstrap.

Table 14–3 VAX Conversational Bootstrap Flags

Bit	Mnemonic	Description
0	CONV	Conversational boot. At various points in the system boot procedure, the bootstrap code solicits parameter and other input from the console terminal. If DIAG is set, then the diagnostic supervisor should enter its menu mode and prompt user for the devices to test.
1	DEBUG	Debug. If this flag is set, OpenVMS VAX maps the code for the XDELTA debugger into the system page tables of the running system.

Table 14–3 (Cont.) VAX Conversational Bootstrap Flags

Bit	Mnemonic	Description
2	INIBPT	Initial breakpoint. If RPB$V_DEBUG is set, OpenVMS VAX executes a BPT instruction immediately after enabling mapping.
3	BBLOCK	Secondary boot from the boot block. Secondary bootstrap is a single 512-byte block, whose LBN is specified in R4.
4	DIAG	Diagnostic boot. Secondary bootstrap is the Diagnostic Supervisor image; the image [SYSMAINT]DIAGBOOT.EXE.
5	BOOBPT	Bootstrap breakpoint. Stops the primary and secondary bootstraps with an XDELTA breakpoint instruction prior to the memory test.
6	HEADER	Image header. Takes the transfer address of the secondary bootstrap image from that file's image header. If the RPB$V_HEADER bit is not set, the image is assumed to have no image header, and control is transfered to the first byte of the secondary boot file.
7	NOTEST	Memory test inhibit. Sets a bit in the PFN bit map for each page of memory present. Does not test the memory.
8	SOLICT	File name. VMB prompts for the name of a secondary bootstrap file.
9	HALT	Halt before transfer. Executes a HALT instruction before transferring control to the secondary bootstrap.
10	NOPFND	No PFN deletion (not implemented; intended to tell VMB not to read a file from the boot device that identifies bad or reserved memory pages, so that VMB does not mark these pages as valid in the PFN bitmap).

Table 14–3 (Cont.) VAX Conversational Bootstrap Flags

Bit	Mnemonic	Description
11	MPM	Specifies that multi-port memory is to be used for the total EXEC memory requirement. No local memory is to be used. This is for tightly-coupled multi-processing. If the RPB$V_DIAG bit is also enabled, then the Diagnostic Supervisor enters its AUTOTEST mode.
12	USEMPM	Specifies that multi-port memory should be used in addition to local memory, as though both were one single pool of pages.
13	MEMTEST	Specifies that a more extensive algorithm be used when testing main memory for hardware uncorrectable (RDS) errors.
14	FINDMEM	Requests the use of MA780 multiport memory if the main MS780 memory is insufficient for booting. This is a remnant of the support for the VAX-11/782 series system and its Asymmetric Multiprocessing (ASMP) environment. Support for the VAX-11/782 and for ASMP was withdrawn with the V5.0 release; with the advent of Symmetric Multiprocessing (SMP) support.

The exact syntax is console-specific, recent VAX consoles tend to use the following:

```
>>> BOOT/R5:flags
```

14.3.6 How do I boot an AlphaStation without monitor or keyboard?

The AlphaStation series will boot without a keyboard attached. To use a serial terminal as the console, issue the SRM console command SET CONSOLE SERIAL followed by the console INIT command. Once this SRM command sequence has been invoked and the CONSOLE environment variable is set to SERIAL, the Alpha system will use the serial terminal. (Set the environment variable to GRAPHICS to select the console display output via the graphics display.)

The DEC 3000 series has a jumper on the motherboard for this purpose. Various older Alpha workstations generally will not (automatically) bootstrap without a keyboard connected, due to the self-test failure that arises when the (missing) keyboard test fails.

The usual settings for the console serial terminal (or PC terminal emulator acting as a serial console are:

```
9600 baud, 8 bits, no parity, one stop bit (9600 baud, 8N1).
```

AlphaServer 4100 and derivative series platforms, and AlphaServer GS80, GS160, and GS320 series system consoles are capable of 57600 baud. See the COM2_BAUD console environment variable, and ensure that you have current SRM firmware version loaded.

The AlphaStation and AlphaServer series use a PC-compatible DB9 serial connector for the COM1 and COM2 serial lines (and for the OPA0: console line, if that was configured via SRM), please see Section 14.26 for details and pin-out.

For information on registering software license product authorization keys (PAKs), please see Section 5.6.2.

For a related behaviour of DECwindows, please see Section 11.10. For information on the VAXstation alternate console mechanisms, please see Section 14.3.3.3.

14.3.7 Downloading and using SRM console Firmware?

This section discusses downloading and using Alpha console firmware, sometimes called PALcode.

14.3.7.1 Where can I get updated console firmware for Alpha systems?

Firmware updates for HP Alpha systems are available from:

- ftp://ftp.digital.com/pub/Digital/Alpha/firmware/index.html
- ftp://ftp.digital.com/pub/Digital/Alpha/firmware/
- ftp://ftp.digital.com/pub/Digital/Alpha/firmware/readme.html

The latest and greatest firmware—if updated firmware has been released after the most recent firmware CD was distributed—is located at:

wideftp://ftp.digital.com/pub/Digital/Alpha/firmware/interim/

For information on creating Alpha bootable floppies containing the firmware, and for related tools, please see the following areas:

wideftp://ftp.digital.com/pub/DEC/Alpha/firmware/utilities/mkboot.txt

wideftp://ftp.digital.com/pub/DEC/Alpha/firmware/utilities/mkbootarc.txt

wideftp://ftp.digital.com/pub/DEC/Alpha/firmware/utilities/mkntboot.txt

The SROM firmware loader expects an ODS-2 formatted floppy, see mkboot. As for which image to use, the ROM image uses a header and the file extension .ROM, and the SROM bootable floppy cannot use the .ROM file.

To check the firmware loaded on recent OpenVMS Alpha systems, use the command:

```
$ write sys$output f$getsyi("console_version")
$ write sys$output f$getsyi("palcode_version")
SDA> CLUE CONFIG
```

Also see Section 14.3.7.2. For information on HP Integrity EFI firmware upgrades and for a sequence useful in generating CD-R or CD-RW media containing a firmware disk image, please see Section 14.3.11.

14.3.7.2 How do I reload SRM firmware on a half-flash Alpha system?

Some of the AlphaStation series systems are ‘‘half-flash’’ boxes, meaning only one set of firmware (SRM or AlphaBIOS) can be loaded in flash at a time. Getting back to the SRM firmware when AlphaBIOS (or ARC) is loaded can be a little interesting...

That said, this usually involves shuffling some files, and then getting into the AlphaBIOS firmware update sequence, and then entering ‘‘update srm’’ at the apu-> prompt.

To shuffle the files, copy the target SRM firmware file (as200_v7_0.exe is current) to a blank, initialized, FAT-format floppy under the filename A:\FWUPDATE.EXE

From the AlphaBIOS Setup screen, select the Upgrade AlphaBIOS option. Once the firmware update utility gets going, enter:

```
     Apu-> update srm

           Answer "y" to the "Are you ready...?"

     Apu-> quit
```

You’ve reloaded the flash. Now power-cycle the box to finish the process.

Also see Section 14.3.7.1.

14.3.7.3 How do I switch between AlphaBIOS/ARC and SRM consoles?

The specific steps required vary by system. You must first ensure that the particular Alpha system is supported by OpenVMS (see the SPD), that all core I/O components (graphics, disk controllers, etc) in the system are supported by OpenVMS (see the SPD), and that you have an OpenVMS distribution, that you have the necessary license keys (PAKs), and that you have the necessary SRM firmware loaded.

A typical sequence used for switching over from the AlphaBIOS graphics console to the SRM console follows:

1 Press F2 to get to the AlphaBIOS setup menu.

2 Pick the ‘‘CMOS Setup...’’ item.

3 Press F6 to get to the ‘‘Advanced CMOS Setup’’ menu.

4 Change the ‘‘Console Selection’’ to ‘‘OpenVMS Console (SRM)’’.

5 Press F10, F10, then Enter to save your changes.

6 Power-cycle the system.

Most Alpha systems support loading both the AlphaBIOS/ARC console and the SRM console at the same time, but systems such as the AlphaStation 255 are ‘‘half-flash’’ systems and do not support the presence of both the AlphaBIOS/ARC and SRM console firmware at the same time. If you have a ‘‘half-flash’’ system, you must load the SRM firmware from floppy, from a network download, or from a firmware CD-ROM. Following the normal AlphaBIOS or ARC firmware update sequence to the APU prompt, and then explictly select the target console. In other words, power up the system to the AlphaBIOS or ARC console, use the supplementary options to select the installation of new firmware (typically from CD-ROM), and then rather than using a sequence which updates the current firmware:

```
Apu-> update
  -or-
Apu-> update ARC
Apu-> verify
Apu-> quit
Power-cycle the system
```

Use the following sequence to specifically update (and load) SRM from AlphaBIOS/ARC on a ‘‘half-flash’’ system:

```
Apu-> update SRM
Apu-> verify
Apu-> quit
Power-cycle the system
```

Use the following sequence to specifically update (and load) the AlphaBIOS/ARC console from SRM on a ‘‘half-flash’’ system:

```
>>> b -fl 0,A0 ddcu
BOOTFILE: firmware_boot_file.exe

Apu-> update ARC
Apu-> verify
Apu-> quit
Power-cycle the system
```

Once you have the SRM loaded, you can directly install OpenVMS or Tru64 UNIX on the system. Do not allow Microsoft Windows NT or other operating system(s) to write a ‘‘harmless’’ signature to any disk used by OpenVMS Alpha or OpenVMS VAX, as this will clobber a key part of the disk; this will overwrite the OpenVMS bootblock. (On OpenVMS Alpha and OpenVMS VAX, you can generally recover from this so-called ‘‘harmless’’ action by using the WRITEBOOT.EXE tool.

Using OpenVMS I64 and the EFI console, the bootblock structures are expected to be compatible with those of Microsoft Windows and other Integrity operating systems; please see the discussion of the SET BOOTBLOCK command and the SYS$SETBOOT.EXE image in Section 9.7.3, in Section 14.3.9, and in the OpenVMS documentation for related details.)

If you have a ‘‘full-flash’’ system and want to select the SRM console from the AlphaBIOS or ARC console environment, select the ‘‘Switch to OpenVMS or Tru64 UNIX console’’ item from the ‘‘set up the system’’ submenu. Then power-cycle the system. If you have a ‘‘full-flash’’ system with the SRM console and want to select AlphaBIOS/ARC, use the command:

```
>>> set os_type NT
```

and power-cycle the system.

For information on acquiring firmware, see Section 14.3.7.1. For information on OpenVMS license PAKs (for hobbyist use) see Section 2.8.1. For information on the Multia, see Section 14.4.1.

Information on enabling and using the failsafe firmware loader for various systems—this tool is available only on some of the various Alpha platforms—is available in the hardware documentation for the system. This tool is used/needed when the firmware has been corrupted, and cannot load new firmware.

The full list of AlphaBIOS key sequences—these sequences are needed when using an LK-series keyboard with AlphaBIOS, as AlphaBIOS expects a PC-style keyboard:

F1	Ctrl/A
F2	Ctrl/B
F3	Ctrl/C
F4	Ctrl/D
F5	Ctrl/E
F6	Ctrl/F
F7	Ctrl/P
F8	Ctrl/R
F9	Ctrl/T
F10	Ctrl/U
Insert	Ctrl/V
Delete	Ctrl/W
Backspace	Ctrl/H
Escape	Ctrl/[
Return	Ctrl/M
LineFeed	Ctrl/J
(Plus) +	upselect (some systems)
(Minus) -	downselect (some systems)
TAB	down arrow
SHIFT+TAB	up arrow

14.3.8 Console Management Options

Options to collect multiple consoles into a single server are available, with both hardware options and software packages that can provide advanced features and capabilities.

Some of the available console management options for OpenVMS:

- Heroix: http://www.robomon.com/
- KI Products: http://www.ki.com/products/clim/
- Global Maintech: http://www.globalmt.com/
- TECsys: http://www.tditx.com/
- CA: http://www.cai.com/products/commandit.htm

Computer Associates is the owner of what was once known as the VAXcluster Console System (VCS) console management package, and has integrated this capability into the CA management product suite.

14.3.9 Why do my EFI Boot Aliases Fail?

OpenVMS I64 boot aliases contain signature information referencing the specific volume, meaning that the entries are specific to the disk volume and not the disk device. This also means that certain operations, such as the SET BOOTBLOCK command or the RUN SYS$SETBOOT.EXE operation that can rewrite these volume signatures (signature or GUID values) can render existing boot aliases unusable.

If your boot aliases do not function as expected, first try removing and re-adding them; this will resynchronize the boot aliases with the volume contents. If you are using the SET BOOTBLOCK command or the RUN SYS$SETBOOT.EXE operation to rewrite the disk bootblock, you can request that the current signatures (if any) be preserved, and this will typically maintain the validity of your EFI console boot aliases.

14.3.10 Can OpenVMS access the EFI console Boot Aliases?

For access to the EFI console environment from OpenVMS I64, see the BOOT_OPTIONS.COM command procedure, and the EFI SET, SHOW and BCFG mechanisms. Details on these are in the System Manager’s and particularly in the System Manager’s Utilities manual.

For related information on boot aliases, please see Section 14.3.5.1.

14.3.11 Downloading and using EFI Console Firmware?

HP Integrity EFI system firmware can be downloaded in the form of a bootable image master, unzipped and then burned onto CD or DVD media (please see Section 9.7 for details of recording optical media directly on OpenVMS), and the system can then generally be booted off the created media to perform the EFI firmware upgrade.

The HP Integrity Server website is accesssable via the following URL, and the available services and support information there has links to the available platform-specific firmware images and upgrade-related materials:

- http://www.hp.com/go/integrity/

To use the following sequence, you will need a writable or rewritable CD drive and software, and a blank CD-R or CD-RW disk. If you use CD writer software for another platform, you will want to use the block, binary, ISO or raw mode operations appropriate for the particular chosen recording package. The following directions assume use of OpenVMS and native CD-R capabilities, please see Section 9.7 for associated details.

1. First, you must acquire the Integrity server firmware from the above URL. Select the platform, and navigate to the supporting information and specifically to the Download Drivers and Software link
2. Select Cross operating system (BIOS, Firmware, etc.)
3. Locate the appropriate ISO-format firmware file. There are several firmware file formats available and there are also various off-line diagnostic images, choose the ISO-format firmware file.
4. Read the directions for the firmware file, then download the ISO-format firmware (zip-compressed) file. A binary-mode FTP transfer should be used.
5. Unzip the file into the corresponding .ISO data file. Somewhat confusingly, the .ISO extension can indicate either a block-oriented raw image of a disk, or a disk with the ISO-9660 volume structure. In this case, the former is intended and this file contains a a block copy or disk image of the firmware disk for the platform, and may or may not be an ISO-9660 volume structure. The unzip tool is available on the OpenVMS Freeware and elsewhere; please see Section 13.11 for details and locations.
6. Use CDRECORD or other available recording tool (please see Section 9.7 for related details) to burn a CD-R or CD-RW disk, specifying the ISO file as the source for the burn operation.
7. Shut the Integrity Server system down to the EFI console level.
8. Unload the recorded CD media from the CD-R drive, label it, and load it into the Integrity console drive. This assuming the disk was not generated directly on an Integrity CD-R/RW-capable drive, of course.
9. Using the EFI shell, display the current firmware version using the command

   ```
   info fw
   ```

10. Exit the EFI shell and select the boot options maintenance menu; create a boot alias for the removable media drive for the CD; for the newly-created firmware disk.

11 Boot it. Follow the directions displayed by the firmware loader and related documentation, heeding the release notes that were reviewed earlier.

12 Perform a cold restart of the Integrity server.

For information on Alpha SRM console firmware upgrades, please see Section 14.3.7.

14.4 What platforms will OpenVMS operate on?

For the list of boxes that are officially and formally supported by OpenVMS Engineering, please see the OpenVMS Software Product Description (SPD).

- http://h18000.www1.hp.com/info/spd/

  OpenVMS typically uses SPD 25.01.xx, SPD 41.87.xx, and SPD 82.35.xx.

Sometimes a particular and officially unsupported Alpha box or Alpha motherboard will sufficiently resemble a supported box that the platform can effectively mimic and can bootstrap OpenVMS. Alternatively, somebody (usually one or more engineers within the OpenVMS Engineering group) will have put together a bootstrap kit – such as the kit for the Alpha Multia—which permits OpenVMS to bootstrap on the platform.

Contrary to the assumptions of some folks, there are platform-level differences even within the VAX and within the Alpha platforms—hardware-level differences that can require moderate to extensive new coding within OpenVMS. Within a platform series, and particularly within Alpha platforms (and those few VAX systems) that support Dynamic System Recognition (DSR), OpenVMS can usually bootstrap.

DSR is a mechanism by which OpenVMS can gather platform-specific information, and DSR is the reason why newer Alpha systems can be more easily and more commonly supported on older OpenVMS Alpha releases. DSR is implemented with OpenVMS Alpha code, with SRM console code, and with platform non-volatile memory.

OpenVMS users with experience on older OpenVMS VAX releases and VAX hardware will recall that then-new VAX systems either required an OpenVMS VAX upgrade, or that earlier releases would mis-identified then-newer VAX systems—such as the case of the VAX 7000 model 800 being (mis)identified as a VAX 7000 model 600 when bootstrapped on OpenVMS VAX V5.5-2. (This (mis)identification was the outcome of a deliberate engineering effort to permit the VAX 7000 model 800 to bootstrap on V5.5-2; the system manager could configure the VAX 7000 model 800 to (mis)identify itself as a model 600, to permit the system to bootstrap on V5.5-2.) OpenVMS VAX and VAX platforms lack DSR support.

OpenVMS I64 (please see Section 14.4.5 for Intel Itanium terminology) supports a platform-level feature similar to the OpenVMS Alpha DSR mechanism, based on the ACPI interface and the byte-code interpreter implemented within OpenVMS, within the EFI console, and particularly within non-volatile memory located on (byte-code interpreter compliant) PCI I/O hardware. ACPI tables provide the information that was formerly retrieved from DSR and from the SRM, and the byte-code interpreter can (theoretically) permit at least limited operations with (compliant) PCI hardware, whether or not OpenVMS has a driver for the particular hardware.

The byte code interpreter may or may not permit operations with any particular PCI hardware, and may or may not have sufficient performance for local requirements, and PCI hardware may or may not include the necessary ROM-based drivers in the PCI hardware non-volatile storage. (The intent of this Intel platform-level effort is to move the host software drivers out onto the specific PCI hardware, and to permit the same byte code to operate regardless of the particular host platform.) At least the initial releases of OpenVMS I64 will *not* have support for the byte code interpreter nor for arbitrary PCI or system hardware, but will have support for ACPI-based system identification and system configuration.

14.4.1 on the Alpha Multia?

Yes, there are a set of unsupported images that permit specific OpenVMS Alpha versions to bootstrap on the Multia UDB system. These images and the associated instructions are available at the OpenVMS Freeware website:

- http://www.hp.com/go/openvms/freeware/

Look in the Freeware V5.0 /multia/ directory.

Instructions are included IN the kits. READ THE INSTRUCTIONS. PLEASE!

Some of the restrictions involved when running OpenVMS on the Multia system include (but may well not be limited to) the following:

- The PCMCIA support was completely removed, because the Intel chip on the Multia was not compatable with the Cirrus chip on the Alphabook.

  This means, of course, that you will not see and cannot use any PCMCIA cards on a Multia.

  The Multia uses shared interrupts, and as a result, a special ZLXp-E series graphics device driver—one that does not use interrupts—is needed. This driver is provided in the kit.
- The serial lines don't work.
- If you have a Multia with a PCI slot, you can't use any PCI card that requires interrupts.
- The SRM console on this system is very old and very fragile. (This SRM console was designed only and strictly for diagnostic use, and was not particularly tested or used with OpenVMS.)
- If things don't work for you, don't expect to see any OpenVMS updates, nor SRM console updates, nor any support.
- Do not expect to see any new versions of OpenVMS on the Multia nor on any other unsupported systems. If such new versions do appear and do work, please consider it as a pleasant surprise.

The Multia images are not included on the OpenVMS Freeware V4.0 CD-ROM kit, the kit that was distributed with OpenVMS V7.2. (These images became available after Freeware V4.0 shipped.)

Other sources of information for OpenVMS on Multia include:

- http://www.djesys.com/vms/hobbyist/multia.html
- http://www.djesys.com/vms/hobbyist/mltianot.html
- http://www.djesys.com/vms/hobbyist/support.html
- http://www.netbsd.org/Ports/alpha/multiafaq.html
- http://www.brouhaha.com/~eric/computers/udb.html

14.4.2 on AlphaPC 164LX? AlphaPC 164SX?

OpenVMS Alpha is not supported on the AlphaPC 164LX and 164SX series, though there are folks that have gotten certain of the LX series to load SRM and bootstrap OpenVMS. (The Aspen Durango II variant, specifically.)

One problem has been generally reported: ATA (IDE) bootstraps will fail; SCSI storage and a SCSI CD-ROM device is required.

Also see Section 14.4.2.1.

14.4.2.1 on the NoName AXPpci33 system?

Information on bootstrapping OpenVMS (using the Multia files described in Section 14.4.1) on the (unsupported) NoName AXPpci33 module is available at:

- http://www.jyu.fi/~kujala/vms-in-axppci33.txt

Tips for using the Multia files with the AXPpci33:

- You have to use the Multia kit and follow the directions in ALPHA8, but do *not* load the Multia SRM firmware into the AXPpci33. Rather, download and use the latest firmware for the AXPpci33 from the HP Alpha firmware website instead.
- 64 MB memory is generally necessary.
- you cannot use any PCI cards, and if you plan on networking, you need to find an ISA Ethernet card supported by OpenVMS.
- When the AXPpci33 board bootstraps, it will dump some stuff like a crash dump, but it will continue and—so far—this hasn't caused any particular hassles.
- The system shutdown and reboot procedures do not work properly.
- The serial console is reported to not work, though the serial ports apparently do work. The status of the parallel port is unknown.
- Rumour has it that you have one of the AXPpci33 motherboards with the PS/2 mouse and keyboard connectors and a VGA card (one that will work under DECwindows) and you can run DECwindows on the system.

14.4.3 on the Alpha XL series?

No.

OpenVMS Engineering does not formally support the Alpha XL series, nor will OpenVMS (informally) bootstrap on the Alpha XL series.

OpenVMS can not, will not, and does not bootstrap on the Alpha XL series. The Alpha XL series was targeted for use (only) with the Microsoft Windows NT operating system.

The Alpha XL platform does not resemble other supported platforms.

14.4.4 OpenVMS on the Personal Workstation -a and -au series?

Though OpenVMS is not supported on the Personal Workstation -a series platforms, OpenVMS might or might not bootstrap on the platform.

If you wish to attempt this, you must ensure that all graphics and all I/O controllers in the system are supported by OpenVMS. You must also ensure that you have the most current firmware loaded.

Here are some salient differences within the various Personal Workstation series:

- The -a series was designed and was tested for Windows NT use. Only. It is not supported for use with OpenVMS.
- The -au series was designed and tested for Windows, OpenVMS, and Tru64 UNIX compatibility, and is considered a supported system.
- There are at two different and distinct variants of the family, and usually refered to by their internal hardware project names.
    - The Miata MX5. The Miata MX5 variant has no USB ports and no on-board SCSI. The on-board Intel SIO chipset is not supported by OpenVMS, and thus OpenVMS cannot bootstrap ATAPI CD-ROM devices.

      That said, the Miata MX5 -a series typically came with DEC branded Adaptec 2940UW SCSI controllers, Matrox Millennium graphics cards, no L3 cache module, and an Toshiba IDE CD-Rom. Some came with very high end Powerstorm graphics card if the system was destined to do CAD or movie rendering.

      Graphics and other I/O can and does vary by package.

      The Miata MX5 is not supported by OpenVMS.

- The Miata GL. The Miata GL variant has USB ports and on-board SCSI and bootstraps using the on-board Cypress IDE chipset and an ATAPI CD-ROM are supported by OpenVMS. The Miata GL -a variant is need not be configured with an add-on SCSI controller, given both the ability to bootstrap from ATAPI CD-ROM and the on-board SCSI.

  Graphics and other I/O can and does vary by package.

  Various of the Miata GL systems are supported by OpenVMS.

For obvious reasons, most folks will prefer and will select a Miata GL system, given the choice between the Miata MX5 and the Miata GL series. And as for your next question, you cannot necessarily nor easily distinguish the Miata MX5 from the Miata GL based solely on the model number.

See Section 14.4.4.2 for related details.

14.4.4.1 OpenVMS on the Whitebox Windows-Only series Alpha?

Though OpenVMS is not supported on the ‘‘Whitebox’’ series of Alpha platforms, OpenVMS might or might not bootstrap on the platform. These systems were specifically configured, targeted and supported only for use with the Microsoft Windows NT operating system.

On some of the ‘‘Whitebox’’ systems, the following sequence of console commands can potentially be used to convert the system over to unsupported use by and for OpenVMS Hobbyist users. (But please note that if you wish to attempt this, you must ensure that all graphics and all I/O controllers in the system are supported by OpenVMS, and you must ensure that you have the most current SRM firmware loaded. (For information on locating and downloading the most current Alpha SRM firmware, please see Section 14.3.7.1.) And you must realize that the resulting Whitebox configuration will be entirely unsupported and may or may not be stable and useful.)

```
set os_type vms
cat nvram  ! too see what is in this, if anything
edit nvram
10 set srm_boot on
20 e
init
```

If your nvram has other contents, you will need to change the line numbers (10 and 20) to reflect the contents of your configuration. To obtain documentation on the commands of the console editor, enter the ? command within the editor.

The above sequence was reportedly tested on the DIGITAL Server 3300 series, a relative of the AlphaServer 800 series. The DIGITAL Server 3300 is not supported by OpenVMS, though the AlphaServer 800 series is a supported platform. The sequence may or may not work on other platforms, and may or may not work on the DIGITAL Server 3300 platform.

Also see Section 5.33.

14.4.4.2 OpenVMS and Personal Workstation ATA (IDE) bootstrap?

OpenVMS will boot and is supported on specific Personal Workstation -au series platforms, though OpenVMS will require a SCSI CD-ROM if the Intel Saturn I/O (SIO) IDE chip is present in the configuration— only the Cypress IDE controller chip is supported by OpenVMS for IDE bootstraps. (Configurations with the Intel SIO are not generally considered to be supported systems.)

If you have an -au series system, you can determine which IDE chip you have using the SRM console command:

```
SHOW CONFIGURATION
```

If you see ‘‘Cypress PCI Peripheral Controller’’, you can bootstrap OpenVMS from IDE storage. If you see ‘‘Intel SIO 82378’’, you will need to use and bootstrap from SCSI. (A procedure to load DQDRIVER on the Intel SIO—once the system has bootstrapped from a SCSI device—is expected to be included as part of the contents of the DQDRIVER directory on Freeware V5.0 and later.)

Many of the -a series systems will include the Intel SIO, and thus cannot bootstrap from IDE.

See Section 14.4.4 for related details.

14.4.5 On the Intel Itanium IA-64 platform?

OpenVMS has been ported to the Intel IA-64 architecture; to HP Integrity systems based on the Intel Itanium Processor Family.

The first release of OpenVMS I64 was V8.0, with the first general or first production release of OpenVMS I64 known as V8.2. Yes, there was a V8.1 release, too.

Some Intel and HP terminology: Itanium Processor Family is the name of the current implementation; of the current Intel microprocessor family implementing the IA-64 architecture. IA-64 is the name of the Intel architecture implementing the VLIW (Very Long Instruction Word) design known as EPIC (Explicitly Parallel Instruction Computing).

I64 is the name of a family of HP computer systems that use Intel Itanium processors and that are supported by ‘‘HP OpenVMS for Integrity Servers’’ (and itself more commonly known as ‘‘OpenVMS I64’’); by one of the HP operating systems that runs on HP Integrity hardware.

The Extensible Firmware Interface (EFI) is the name of the console environment for Itanium systems, and the Baseboard Management Console (BMC) and the optional Management Processor (MP) are the most typical hardware interfaces into the system console.

14.4.5.1 Where can I get Intel Itanium information?

Intel Itanium Processor Family and IA-64 Architecture, Hardware, Software, and related docoumentation materials are available at:

- ftp://download.intel.com/design/IA-64/manuals/
- ftp://download.intel.com/design/IA-64/Downloads/
- ftp://download.intel.com/design/IA-64/Downloads/archSysSoftware.pdf
- ftp://download.intel.com/design/IA-64/Downloads/24870101.pdf

Information on the classic Intel Extensible Firmware Interface (EFI) (for IA-64) and of the multi-platform Unified EFI (UEFI) console project documentation are available at the following URLs:

- Intel
  http://developer.intel.com
- UEFI
  http://www.uefi.org

Please see Section 14.4.5 for Intel Itanium terminology.

14.5 What is the least expensive system that will run OpenVMS?

The cheapest systems that are or have been recently offered by HP that will run OpenVMS Alpha are the AlphaServer DS10 server, the AlphaStation XP900 workstation, the AlphaStation VS10 workstation, and the AlphaStation XP1000 workstation. Other companies sell Alpha-powered systems and Alpha motherboards, some of which will run (and can be purchased with) OpenVMS—see the OpenVMS Software Product Description (SPD) for details on the supported systems and configurations. There are also many used AlphaStation, AlphaServer, and DEC 3000 series models available which are quite suitable. For more experienced OpenVMS system managers, the (unsupported) Multia can bootstrap OpenVMS—see Section 14.4.1 for details.

Used Itanium-based systems that a hobbyist could likely use to bootstrap OpenVMS I64 have been seen selling on auction websites for under US$1000. New Integrity rx2620 series systems (officially supported by OpenVMS I64) have been offered as part of a week-long DSPP porting and training package for US$2000. See Section 2.8.3 for details on the DSPP program. Also see the HP Renew used- and/or refurbished-equipment program for any hardware that might be available.

Free and commercial VAX software-based hardware emulators are available for various platforms. See Section 13.12 for details on those.

Hobbyist-related hardware platform selection information—various options and considerations around VAX, Alpha and Integrity systems, and an introduction to hardware emulation—is available at: http://www.hoffmanlabs.org/)

Depending on the OpenVMS version and configuration, the OpenVMS Software Product Description (SPD) is available at:

- http://www.hp.com/go/openvms/doc/

When purchasing a system, ensure that the system itself is supported, that the system disk drive is supported or closely compatible, that the optical (CD or DVD) drive is supported or is closely compatable and that (in the case of SCSI devices) it also specifically supports 512-byte block transfers; no equivalent requirement exists for IDE devices. Also particularly ensure that the video controller is supported. Use of supported HP hardware will generally reduce the level of integration effort involved.

A CD-ROM, CD-R or DVD drive is required for OpenVMS Alpha installations, and a DVD-ROM or recordable or rewritable DVD DVD drive is required for OpenVMS I64 installations.

CD-ROM drive compatibility information is available at:

- http://sites.inka.de/pcde/dec-cdrom-list.txt

14.6 Where can I get more information on Alpha systems?

HP operates an AlphaServer information center at:

- http://www.hp.com/go/server

Alpha Technical information and documentation is available at:

- ftp://ftp.compaq.com/pub/products/alphaCPUdocs/
- http://h18000.www1.hp.com/products/software/alpha-tools/
- ftp://ftp.digital.com/pub/DEC/Alpha/systems/
- http://ftp.digital.com/pub/ Digital/info/semiconductor/literature/dsc-library.html
- Alpha Systems Update: http://www.compaq.com/alphaserver/fb_acu.html

Software Product Description (SPD) information, including platform support documentation:

- http://h18000.www1.hp.com/info/spd/

  OpenVMS typically uses SPD 25.01.xx, SPD 41.87.xx, and SPD 82.35.xx.

Information on Multia hardware is available at:

- http://www.netbsd.org/Ports/alpha/multiafaq.html

Information on DEC 3000 series hardware is available at:

- http://www.phys.ufl.edu/~prescott/linux/alpha/dec3000-sysinfo.html
- http://www.phys.ufl.edu/~prescott/linux/alpha/dec3000-docs.html
- http://ftp.netbsd.org/pub/NetBSD/misc/dec-docs/index.html

The NetBSD folks maintain useful Alpha hardware information at:

- http://www.netbsd.org/Ports/alpha/models.html

14.7 Describe Alpha instruction emulation and instruction subsets?

The Alpha architecture is upward- and downward-compatible, and newer instructions are emulated on older platforms, for those cases where the compiler is explicitly requested to generate the newer Alpha instructions.

In particular, OpenVMS Alpha V7.1 and later include the instruction emulation capabilities necessary for the execution of newer Alpha instructions on older Alpha microprocessors. (Instruction emulation capabilities are available for user-mode application code, and are not available to device drivers or other similar kernel-mode code.)

Alpha instructions are available in groups (or subsets). Obviously, there is the base instruction set that is available on all Alpha microprocessors. Then, the following are the current instruction extension groups (or subsets) that are available on some of various recent Alpha microprocessors:

- byte/word extension (BWX):
  LDBU, LDWU, SEXTB, SEXTW, STB, and STW.
- floating-point and square root extension (FIX):
  FTOIS, FTOIT, ITOFF, ITOFS, ITOFT, SQRTF, SQRTG, SQRTS, and SQRTT.
- count extension (CIX):
  CTLZ, CTPOP, and CTTZ.
- multi-media extension (MVI):
  MAXSB8, MAXSW4, MAXUB8, MAXUW4, MINSB8, MINSW4, MINUB8, MINUW4, PERR, PKLB, PKWB, UNPKBL, and UNPKBW.

The typical instruction subset that provides the biggest win—and of course, your mileage may vary—is typically the instruction set that is provided by the EV56 and later; specifically, the byte-word instruction subset. To select this subset, use the following:

```
/ARCHITECTURE=EV56/OPTIMIZE=TUNE=GENERIC
```

The /ARCHITECTURE controls the maximum instruction subset that the compiler will generally use, while the /OPTIMIZE=TUNE controls both the instruction-level scheduling and also the instructions generated inside loops—any code resulting from /OPTIMIZE=TUNE that is specific to an instruction subset will be generated only inside loops and will also be ‘‘protected’’ by an AMASK-based test that permits the execution of the proper code for the particular current Alpha microprocessor.

Typically /OPTIMIZE=TUNE=GENERIC is the appropriate choice for tuning, and the /ARCHITECTURE selects the minimum target architecture for general use throughout the generated code.

generated for later architectures and instruction subsets will run on older Alpha systems due to the emulation, but if /ARCHITECTURE is a significant benefit, then the emulation might be a performance penalty.

Please see the OpenVMS Ask The Wizard area for the source code of a (non-privileged) tool that looks at the instruction subsets available on the particular Alpha microprocessor that the tool is run on. This tool demonstrates the use of the Alpha AMASK and IMPLVER instructions.

Please see Section 10.22 and Section 14.9 for additional details and related considerations.

14.8 So how do I open up the DEC 3000 chassis?

After removing those two little screws, tilt the back end of the top shell upwards—then you can remove the lid.

14.9 What is byte swizzling?

‘‘Swizzling’’ is the term used to describe the operation needed to do partial longword (i.e. byte or word) accesses to I/O space on those systems that don’t support it directly. It involved shifting the offset into an address space by 5 (or 7 for one older system), and ORing this into the base address. It then required the size of the operation to be ORed into the low order bits.

That is, because the EV4 and EV5 CPUs did not bring bits 0 and 1 off the chip, to do programmed I/O for bytes/words, the information on the size/offset of the transfer was encoded into the address data. The data itself then had to be shifted into the correct ‘‘byte lane’’ ; into the required offset position within a longword transfer;

The EV56 microprocessor supports byte/word instruction references in memory space, however only specific EV56 systems can support byte/word accesses into I/O space; device drivers may or may not be able to utilize to byte/word instructions to access device registers. Further, even on an EV56 system with hardware support for byte/word accesses into I/O space, the relevant OpenVMS routines typically do not support byte/word access into I/O space.

Systems based on the EV6 microprocessor (with the salient exception of the AlphaServer GS60 and AlphaServer GS140 series, for reasons of platform compatability) support a flat, byte addressable I/O space.

If a device driver uses CRAM or IOC$WRITE_IO/IOC$READ_IO, then OpenVMS will correctly process the swizzling requirements without requiring changes the driver; OpenVMS will transparently swizzle and unswizzle the I/O space references, if needed for the particular target platform. (Access and use of these routines may or may not be feasible within the requirements for a particular device driver, with the decision typically based on the target performance requirements and the expected frequency of device references and thus the expected frequency of calls to these or other similar routines.)

To use byte/word operations on MEMORY, you need to tell the compiler to use the EV56 or EV6 architecture (/ARCHITECTURE=EV56). Memory operations did not swizzle, but the compiler would do long/quad access, and extract/insert bytes as needed. Using /ARCHITECTURE=EV56 allows smaller, more efficient byte/word access logic to memory.

If the application is directly referencing I/O space access across a range of Alpha systems such as is done with the X Windows device drivers, then the driver will need to know how to do swizzling for old platforms, and byte access for new platforms. Device drivers for new graphics controllers can specifically target and specifically require platforms based on EV6 and later Alpha microprocessors because of this requirement, for instance.

Please see Section 10.22 and Section 14.7 for additional details and related considerations.

14.10 What is the layout of the VAX floating point format?

The VAX floating point format is derived from one of the PDP-11 FP formats, which helps explain its strange layout. There are four formats defined: F 32-bit single-precision, D and G 64-bit double-precision and H 128-bit quadruple precision. For all formats, the lowest addressed 16-bit ‘‘word’’ contains the sign and exponent (and for other than H, some of the most significant fraction bits). Each successive higher-addressed word contains the next 16 lesser-significant fraction bits. Bit 15 of the first word is the sign, 1 for negative, 0 for positive. Zero is represented by a biased exponent value of zero and a sign of zero; the fraction bits are ignored (but on Alpha, non-zero fraction bits in a zero value cause an error.) A value with biased exponent zero and sign bit 1 is a ‘‘reserved operand’’ - touching it causes an error - fraction bits are ignored. There are no minus zero, infinity, denormalized or NaN values.

For all formats, the fraction is normalized and the radix point assumed to be to the left of the MSB, hence the following range: 0.5 less than or equal to f and less than 1.0. The MSB, always being 1, is not stored. The binary exponent is stored with a bias varying with type in bits 14:n of the lowest-addressed word.

FP Type	Exponent Bits	Exponent Bias	Mantissa (Fraction) bits, including hidden bit
F	8	128	24
D	8	128	56
G	11	1024	53
H	15	16384	113

The layout for D is identical to that for F except for 32 additional fraction bits.

Example: +1.5 in F float is hex 000040C0 (fraction of .11[base 2], biased exponent of 129)

14.11 Where can I find more info about VAX systems?

- HP provides limited VAX platform information via links at the AlphaServer website, itself available via:
  http://www.hp.com/go/server/
- Jim Agnew maintains a MicroVAX/VAXstation FAQ at:
  http://www.people.vcu.edu/~agnew/MVAX/MVAX_FAQ.HTML

- The VAXstation 3100 Owner’s Guide:
  http://www.whiteice.com/~williamwebb/intro/DOC-i.html
- VAXstation 3520 and VAXstation 3540 (Firefox) Manual:
  http://starfish.osfn.org/rcs/VAX/vs3540sg.pdf
- VAX Console information:
  http://www.mcmanis.com/chuck/computers/vaxen/panels.htm
- A field guide to PDP-11 (and VAX) Q-bus and UNIBUS modules can be found at:
  http://metalab.unc.edu//pub/academic/computer-science/history/pdp-11/hardware/field-guide.txt
- Various VAX historical information (also see Section 2.1) can be found at:
  http://telnet.hu/hamster/vax/e_index.html

14.12 Where can I find information on NetBSD for VAX systems?

Gunnar Helliesen maintains a NetBSD VAX FAQ at

- http://vaxine.bitcon.no/

14.13 What system disk size limit on the MicroVAX and VAXstation 3100?

System disks larger than 1.073 gigabytes (GB)—1fffff hexidecimal blocks – are not supported on any member of the VAXstation 3100 series and on certain older members of the MicroVAX 3100 series, and are not reliable on these affected systems. (See below to identify the affected systems—the more recent members of the MicroVAX 3100 series systems are NOT affected.)

Various of the SCSI commands used by the boot drivers imbedded in the console PROM on all members of the VAXstation 3100 series use "Group 0" commands, which allow a 21 bit block number field, which allows access to the first 1fffff hexidecimal blocks of a disk. Any disk references past 1fffff will wrap—this wrapping behaviour can be of particular interest when writing a system crashdump file, as this can potentially lead to system disk corruptions should any part of the crashdump file be located beyond 1.073 GB.

More recent systems and console PROMs use "Group 1" SCSI commands, which allow a 32 bit block number field.

There was a similar limitation among the oldest of the MicroVAX 3100 series, but a console boot PROM was phased into production and was made available for field retrofits—this PROM upgrade allows the use of the "Group 1" SCSI commands, and thus larger system disks. There was no similar PROM upgrade for the VAXstation 3100 series.

Systems that are affected by this limit:

- VAXstation 3100 series, all members. No PROM upgrade is available.
- MicroVAX 3100 models 10 and 20. No PROM upgrade is available.
- MicroVAX 3100 models 10e and 20e. Only systems with console VMB versions prior to V6.4 are affected. A PROM upgrade for these specific systems is (or was once) available.

Also see

- http://www.whiteice.com/~williamwebb/intro/DOC-i.html

Also see Section 9.5.

14.14 What are the VAX processor (CPU) codes?

```
CPU:    Platform:
-----   ---------
KA41-A : MicroVAX 3100 Model 10 and 20
KA41-B : VAXserver 3100 Model 10 and 20
KA41-C : InfoServer
KA41-D : MicroVAX 3100 Model 10e and 20e
KA41-E : VAXserver 3100 Model 10e and 20e
KA42-A : VAXstation 3100 Model 30 and 40
KA42-B : VAXstation 3100 Model 38 and 48
KA43-A : VAXstation 3100 Model 76
KA45   : MicroVAX 3100 Model 30 and 40
KA46   : VAXstation 4000 Model 60
KA47   : MicroVAX 3100 Model 80
KA48   : VAXstation 4000 VLC
KA49-A : VAXstation 4000 Model 90/90A
KA49-B : VAXstation 4000 Model 95
KA49-C : VAXstation 4000 Model 96
KA50   : MicroVAX 3100 Model 90
KA51   : MicroVAX 3100 Model 95
KA52   : VAX 4000 Model 100
KA53   : VAX 4000 Model 105
KA54   : VAX 4000 Model 106
KA55   : MicroVAX 3100 Model 85
KA56   : MicroVAX 3100 Model 96
KA57   : VAX 4000 Model 108
KA58   : MicroVAX 3100 Model 88
```

```
KA59   : MicroVAX 3100 Model 98
KA85   : VAX 8500
KA86   : VAX 8600
KA88   : VAX 8800
KA600  : VAX 4000-50 (aka VAXbrick)
KA610  : MicroVAX I, VAXstation I (aka KD32)
KA620  : rtVAX (VAXeln)
KA62A  : VAX 6000-200
KA62B  : VAX 6000-300
KA630  : MicroVAX II, VAXstation II
KA640  : MicroVAX 3300, MicroVAX 3400
KA650  : VAXstation 3200, MicroVAX 3500, MicroVAX 3600, MicroVAX III
KA64A  : VAX 6000-400
KA655  : MicroVAX 3800, MicroVAX 3900, MicroVAX III+
KA65A  : VAX 6000-500
KA660  : VAX 4000-200, VAX 4 upgrade
KA66A  : VAX 6000-600
KA670  : VAX 4000-300
KA675  : VAX 4000-400
KA680  : VAX 4000-500
KA681  : VAX 4000-500A
KA690  : VAX 4000-600
KA691  : VAX 4000-605A
KA692  : VAX 4000-700A
KA693  : VAX 4000-605A
KA694  : VAX 4000-705A
KA730  : VAX-11/730
KA750  : VAX-11/750
KA780  : VAX-11/780, VAX-11/782
KA785  : VAX-11/785
KA7AA  : VAX 7000-600
KA7AB  : VAX 7000-700
KA7AC  : VAX 7000-800
KA800  : VAXrta
KA820  : VAX 8200, VAX 8300
KA825  : VAX 8250, VAX 8350
KA865  : VAX 8650
```

14.15 Where can I get software and hardware support information?

Please contact the HP Customer Support Center. Services and information, manuals, guides, downloads, and various other information is available via the support link at:

- http://www.hp.com/products/openvms/

Various hardware and system documentation is available at:

- http://www.hp.com/go/services/

TSM (Terminal Server Manager), DEChub, DECserver, etc. information:

- http://www.compaq.com/support/digital_networks_archive/

The owner and maintainer of current DECserver and related hardware is DIGITAL Network Products Group (DNPG):

- http://www.dnpg.com/

14.16 Where can I get hardware self-maintenance support assistance?

The HP Parts Directory and the HP Parts Reference Guide (arguably the most direct descendents of the HP Assisted Services program, of the Compaq Assisted Services program, and of the now-ancient DECmailer program) are available to customers that wish to maintain their own system(s) (self-maintenance), but that wish some level of assistance in acquiring specific parts, hardware diagnostics and hardware manuals for the system(s), and that wish to have access to spares and module-level repairs for customer-performed hardware module swaps:

- http://www.hp.com/go/parts/
- http://www.hp.com/buy/parts/

The HP Parts Reference Guide replaces the CAS-Catalog and DAS-Catalog parts catalogs and related resources.

Details of the available self-maintenance programs and services can vary by geography and by the particular services channel(s), and current program specifics are available via the above URLs.

14.17 Why does my system halt when I power-cycle the console terminal?

Various VAX and Alpha consoles are designed to process the BREAK signal, treating it as a HALT request.

A BREAK is a deliberately-generated serial line framing error.

When a serial line device such as a terminal powers up (or sometimes when powering down) it can generate framing errors. These framing errors are indistingushable from a BREAK signal.

When a BREAK is received on a serial line console for various VAX systems—including most VAXstation, MicroVAX, and VAX 4000 series—it is typically interpreted as a HALT. Alpha systems will also often process a BREAK in a similar fashion, halting the system.

There is no uniform or generally-available way to disable this behaviour on every VAX or Alpha system. On some systems, BREAK processing can be disabled in favor of [CTRL/P], or [CTRL/P] is the only way to halt the processor.

The most common way to avoid these halts is to disable the serial line console or to simply not power-cycle the console terminal. There is certain important system state information that is displayed only on the console, OpenVMS expects to always have access to the system console.

Also see Section 5.6.

14.18 Can I reuse old keyboards, mice and monitors with a PC?

Older HP keyboards (those with the DIGITAL logo and the RJ modular jacks), older HP mice (those with the DIGITAL logo and with the RJ modular jacks, or with a DIN connector with pins in a configuration other than the PC-standard DIN connector pin orientation), and older video monitors (with RGB synch-on-green video signaling) all use signaling formats and/or communications protocols that differ from the PC standards, and are not (easily) interchangable nor (easily) compatible with typical PC peripheral device controllers. The LK201 and LK401 keyboards, the VSXXX series mice, the VR260 and VR290 monitors, etc., are incompatible with most PC systems and with most KVM switches.

Newer HP (and Compaq) keyboards (those with with PC-style DIN plugs, and the HP, Compaq or DIGITAL logo), newer HP mice (with PC-pin DIN plugs, and the HP, Compaq or DIGITAL logo), and newer video monitors (multi-synch, usually with a VGA or SVGA connection, or later) are often interchangeable with "industry standard" PC systems, and can often be used with most PC peripheral device controllers. LK461, LK463, LK46W, LK471, PC7XS-CA, VRC16, VRC21, TFT-series LCD flat-panel displays, etc., are typically reasonably compatible with most PC systems, and will usually perform as expected within the limits of the hardware. (For details of CRT and LCD display compatibility, please see Section 14.19.)

Rule of thumb: if the peripheral device component was sold for use with the DEC 2000 (DECpc 150 AXP), an AlphaServer series, an AlphaStation series, or a more recent Alpha system, it will probably work with a PC peripheral controller or with a PC-compatible KVM switch. If the peripheral device component was sold for use with an VT420 or older terminal, most VAX, most VAXstation, and most Alpha systems with names in the format DEC [four-digit-number], it probably won’t work on a PC system or with a PC-compatible KVM.

Note that the above is a general guideline, and should not be read to indicate that any particular peripheral device will or will not work in any particular configuration, save for those specific configurations the device is explicitly supported in.

Software Integrators sells a video adapter card called Gemini P1 which will drive many of the older HP (DIGITAL-logo) fixed-frequency monitors on a PC system:

- http://www.si87.com/

The DIGITAL (classic 2-5-2-style) part number 29-32549-01 converts the output from the RGB cable (3 BNC, synch-on-green) that comes with the VAXstation 3100 and VAXstation 4000 series to a female SVGA D connector. You may be able to find third-party converters or adapters (3 BNCs with synch-on-green signaling to 5 BNCs with VGA/SVGA, or to 15-pin VGA/SVGA.

This adapter will allow PC multisync monitors with the needed frequency specifications to be used with the VAXstation series synch-on-green video connection. It may well also work with a VAXstation 2000 series systems, but specifics and performance of that combination are not immediately known at this writing.

The protocol definition for the old DIGITAL keyboard and mouse interfaces is buried at the back of the QDSS section in the old VAXstation II manual, specifically, in the back of the VCB02 Video Subsystem Technical Manual (EK-104AA-TM). The keyboard wiring and protocol is in appendix B, and occupies circa 44 pages. The mouse is in appendix C, circa 12 pages.

Also see Section 14.19.

14.19 Which video monitor works with which graphics controller?

To determine the answer to the ‘‘will this video monitor or this LCD panel work with this graphics controller?’’ question, please first locate the resolution(s) and the frequencies that are possible/supported at both ends of the video cable (on the display and on the graphics controller, in other words), and then determine if there are any matching settings available. If there are multiple matches, you will need to determine which one is most appropriate for your needs.

You will also need to determine if the video monitor or graphics controller requires the 3 BNC signaling with the synchronization signals on the green wire, or the 5 BNC signaling common on many PCs, or other connections such as the DB15 video connector or USB connector used on various systems. (BNC signaling is comparatively old, but prevalent with many older hobbyist AlphaStation or VAXstation configurations.)

If there are no matches, you will likely need to change the hardware at one or both ends of the video cable.

The refresh frequencies for many devices have been posted to comp.os.vms and/or other newsgroups. Search the archives for details. Also see:

- http://www.repairfaq.org/
- http://www.mirage-mmc.com/faq/
- http://www.geocities.com/SiliconValley/Foothills/4467/fixedsync.html
- http://saturn.tlug.org/sunstuff/ffmonitor.html
- http://hawks.ha.md.us/hardware/monitor.html

LCD-based and plasma-based flat-panel displays are generally compatible with all recent OpenVMS Alpha systems and supported graphics controllers. For best results, you should generally set the graphics controller to match the native LCD or plasma display resolution and (for LCD displays) also set the controller refresh rate to 60Hz. Check your graphics controller and your display documentation for any device-specific requirements and/or configuration recommendations.

Some of the older graphics controllers around do not necessarily generate stable signals at 60 Hz, if the controller can even generate that refresh rate; you may end up upgrading to a less-old controller. (At least some of the PowerStorm 3D30 and PowerStorm 4D20 series controllers, for instance, are not necessarily the best choice for 60 Hz operations with an LCD, based on empirical testing with an AlphaStation XP1000, PowerStorm 3D30, and a TFT2025 series LCD. Degraded or mismatched signals produce degraded displays, obviously. The newest graphics controllers compatible with your particular system are generally better choices here for use with LCD; the Radeon 7500 series is a good choice for most EV6-class AlphaStation systems, for instance.

Also see Section 14.18.

14.20 Where can I get information on storage hardware?

Information on various HP (Compaq, DIGITAL) OpenVMS and other disk storage hardware and controllers, and related technical information on SCSI, device jumpers, etc., is available at:

- http://theref.aquascape.com/

Note

the aquascape website appears to have become unavailable, and the FAQ maintainer is unaware of a new or replacement server. You may or may not have some success looking for this or of any other now-unavailable sites using the world-wide web archives at:

- **http://www.archive.org/**

14.21 Why does my LK401 keyboard unexpectedly autorepeat?

There are several modes of failure:

- Pressing 2 and 3 keys at the same time causes one key to autorepeat when released. Check the hardware revision level printed on the bottom of the keyboard. If the revision level is C01, the keyboard firmware is broken. Call field service to replace the keyboard with any revision level other than C01.
- Pressing certain keys is always broken. Typical symptoms are: delete always causes a autorepeat, return needs to be pressed twice, etc. This is frequently caused by having keys depressed while the keyboard is being initialized. Pressing ^F2 several times or unplugging and replugging the keyboard frequently fix this problem. (Ensure you have current ECO kits applied; there is a patch available to fix this problem.)
- A key that was working spontaneously stops working correctly. This may be either of the two previous cases, or it may be bad console firmware. Ensure that you have the most recent firmware installed on your Alpha system. In particular, an old version of the DEC 3000 SRM firmware is known to have a bug that can cause this keyboard misbehaviour.

14.22 Problem - My LK411 sends the wrong keycodes or some keys are dead

Check the firmware revision on the keyboard. Hardware revision B01 introduced an incompatability with the device driver which causes the keyboard to not be recognized correctly. There is a patch available to fix this problem: [AXPDRIV06_061] - the fix is also included in OpenVMS V6.2. The rev A01 keyboard, and the LK450 should work without problems.

If you are working from another operating system platform, please see the DECxterm tool and related information on OpenVMS Freeware V5.0.

14.23 Which DE500 variant works with which OpenVMS version?

Ensure you have a version of the Alpha SRM console with support for the DE500 series device. Apply ALL mandatory ECO kits for the OpenVMS version in use, and also apply the CLUSIO, ALPBOOT, and ALPLAN kits, and apply any available ALPCPU ECO kit for the platform.

- DE500-XA
  auto-detection, no auto-negotiation,
  OpenVMS V6.2-1H1 and ALPBOOT ECO, also V7.0 and later and ECO.
  Device hardware id 02000011 and 02000012.
  Component part number 54-24187-01
- DE500-AA
  auto-detection, auto-negotiation,
  OpenVMS V6.2 and ALPBOOT and ALPLAN ECOs, or V7.1 and later and ECO.
  Device hardware id 02000020 and 20000022.
  Component part number 54-24502-01
- DE500-BA
  auto-detection, auto-negotiation,
  OpenVMS V6.2-1H3 and CLUSIO, ALPBOOT, ALPLAN and ALPCPU ECOs, or V7.1-1H1 or later and ECO.
  Device hardware id 02000030 (check connector, vs DE500-FA) (other values on old Alpha SRM firmware)
  Component part number 54-24602-01
- DE500-FA (100 megabit fibre optic Ethernet)
  OpenVMS V7.1-1H1 and later
  Device hardware id 02000030 (check connector, vs DE500-BA) (other values possible on old Alpha SRM firmware)

Component part number 54-24899-01

To check the DE500 device hardware id from OpenVMS, use the following command:

```
$ ANALYZE/SYSTEM
SDA> SHOW LAN/DEVICE=EWc:
```

The ‘‘hardware version’’ will be displayed.

To set the DE500 speed and duplex settings via the associated Alpha SRM console environment variable, see Table 14–4.

Table 14–4 DE500 Speed and Duplex Settings

EWx0_MODE setting	Meaning
Twisted-Pair	10 Mbit/sec, nofull_duplex
Full Duplex, Twisted-Pair	10 Mbit/sec, full_duplex
AUI	10 Mbit/sec, nofull_duplex
BNC	10 Mbit/sec, nofull_duplex
Fast	100 Mbit/sec, nofull_duplex
FastFD (Full Duplex)	100 Mbit/sec, full_duplex
Auto-Negotiate	Negotiation with remote device

To override the console setting and use LANCP:

```
$ RUN SYS$SYSTEM:LANCP
LANCP> SET DEVICE EWA0/SPEED=10
LANCP> DEFINE DEVICE EWA0/SPEED=10
LANCP> SET DEVICE EWA0/SPEED=100/full_duplex
LANCP> DEFINE DEVICE EWA0/SPEED=100/full_duplex
```

Fast Ethernet (100Base, 100 megabit) controllers such as the DE500 series have a pair of connections available—while traditional Ethernet (10Base, 10 megabit) is inherently a half-duplex protocol, Fast Ethernet can be configured to use one or both of the available connections, depending on the controller. Fast Ethernet can thus be half- or full-duplex depending on the configuration and the capabilities of the network controller and the Ethernet network plant. Some Fast Ethernet controllers can also operate at traditional Ethernet speeds, these controllers are thus often refered to as 10/100 Ethernet controllers.

14.24 How do I set the speed and duplex on OpenVMS I64?

OpenVMS I64 on Integrity servers does not provide a console-level environment variable akin to the SRM console variables used to manage the network speed and duplex settings on OpenVMS Alpha and Alpha systems. On OpenVMS I64 on Integrity servers, LANCP is used to manage the speed and the duplex setting of the network controllers.

```
$ RUN SYS$SYSTEM:LANCP
LANCP> SET DEVICE EWA0/SPEED=10
LANCP> DEFINE DEVICE EWA0/SPEED=10
LANCP> SET DEVICE EWA0/SPEED=100/full_duplex
LANCP> DEFINE DEVICE EWA0/SPEED=100/full_duplex
```

The EFI-level network bootstrap operations for a network-based upgrade or a network-based installation of OpenVMS I64 require the use of autonegotiation and a switch capable of supporting it.

See Section 14.23 for a related discussion.

14.25 Third-party or Unsupported disk/tape/controllers/SCSI/widgets?

A wide variety of third-party and formally-unsupported widgets—SCSI and ATA/ATAPI (IDE) disks and tapes, graphics controllers, etc—are obviously widely available, and are used on various platforms.

If you purchase third-party or unsupported or generic SCSI, ATA/ATAPI (IDE) storage devices, you and your device vendor will be responsible for the testing and the support of the devices. In general, you can expect that HP will address non-standards-compliance problems within OpenVMS (changes that will also not prevent operations with other supported devices, of course), but you and/or the device vendor and/or the device manufacturer are responsible for finding and fixing problems in the particular third-party device and or controller involved.

In particular, realize that neither SCSI nor ATA/ATAPI (IDE) is a particularly standard interface, these interfaces tend to be a collection of optionally-implemented and standardized interface features. You should not and can not simply assume that all SCSI nor ATA/ATAPI (IDE) storage devices are interchangeable. If you want to try to use a generic SCSI device, use V6.2 or later, or (better) V7.1-2 or later. If you wish to try to use ATA/ATAPI (IDE), use OpenVMS V7.1-2 or later.

On older OpenVMS releases, see the disk capacity limits (Section 9.5).

With SCSI disks on releases prior to V6.2, ensure that you have the ARRE and ARWE settings configured correctly (disabled). (If not, you will see DRVERR fatal drive errors and error log entries.)

Some SCSI disks set the medium type byte as part of the SCSI size field—this is a SET CAPACITY extension to SCSI specs. This problem also applies to VAX V7.1 and later.

Disks with SCSI disk sizes past 8.58 GB and/or with the SET CAPACITY extension require ALPSCSI07 ECO or the OpenVMS Alpha V7.1-2 or later release. (See Section 9.5 for further details.)

Based on the displays of the (undocumented) SYS$ETC:SCSI_INFO tool; this tool is present in OpenVMS V6.2 and later:

```
Issuing 6-byte MODE SENSE QIOW to get current values for page 01h
       Page Code ................. 01h
       Page Name ................. Read-Write Error Recovery
       Saveable .................. Yes
       Size ...................... 10
       Hex Data .................. E6 08 50 00 00 00 08 00
                                   00 00
```

The E6 shown indicates that the AWRE and ARRE bits are set, and this is incompatible with OpenVMS versions prior to V6.2. Further along in the same SCSI_INFO display, if you also see:

```
Issuing 6-byte MODE SENSE QIOW to get changeable values for page 81h
       Page Code ................. 01h
       Page Name ................. Read-Write Error Recovery
       Saveable .................. Yes
       Size ...................... 10
       Hex Data .................. C0 08 50 00 00 00 08 00
                                   00 00
```

The C0 value means that the AWRE and ARRE values can be changed on this particular SCSI device. (This is not always the case.) If the bits are set, you can use RZDISK from the OpenVMS Freeware, and can reset the E6 flag byte to hexadecimal 26 (or whatever the remaining mask when you remove bits C0) on page one.

Each SCSI and ATA/ATAPI (IDE) host contains non-trivial SCSI and IDE driver software, and each device contains equally non-trivial firmware—taken together with the mechanical and electronic components, this software and firmware will determine whether or not a particular device will function as expected.

Also note that various devices—such as various SCSI CD-R devices —can implement and can require vendor-specific protocol extensions, and these extensions can require modifications to OpenVMS or the addition of various utilities. In various of these cases, these devices perform functions that will require them to use SCSI or ATA/ATAPI (IDE) commands that are (hopefully) architecturally-compatible SCSI or ATA/ATAPI (IDE) command extensions. (Also see Section 7.1 and Section 9.7.)

Some SCSI tapes lack odd-byte transfer support, making operations with OpenVMS problematic at best, as OpenVMS expects odd-byte support. Examples of such include LTO-1 devices such as the HP Ultrium 230 series tape, and the DLT VS80 series tapes. Due to the lack of odd-byte transfer support, LTO-1 devices are not supported by OpenVMS. LTO devices in the LTO-2 and later series do reportedly presently all have odd-byte transfer support, and operations are reportedly rather easier. Do check for formal support, of course.

In order for OpenVMS to officially support a particular device, integration and testing work is mandated. There can be no certainty that any particular device will operate as expected in any particular configuration without first performing this (non-trivial) work.

It is quite possible to find two devices—both entirely compliant with applicable standards or interface documents—that will not interoperate.

The same general statement holds for OpenVMS bootstrapping on an unsupported VAX or Alpha platform. It might or might not work.
In particular, please see the OpenVMS Software Product Description (SPD) for the list of platforms supported by OpenVMS. OpenVMS is not supported on the Personal Workstation -a series, on the Digital Server series platforms, on the AlphaServer 2100 series 5/375 CPU, on the Multia, on the AlphaServer DS20L, and on a variety of other platforms. (You might or might not see success booting OpenVMS on any of these platforms.)

14.25.1 Lists of third-party widgets on OpenVMS?

Various folks have successfully used common third-party disk disk devices with OpenVMS, such as the ATA (IDE) and SCSI variants of the Iomega Zip250 removable disk device.

Common SCSI CD-R/CD-RW devices such as the Plextor PlexWriter 12/10/32S SCSI series and the HP DVD200i series (recording CD-R) have also been successfully utilized with various AlphaStation and VAXstation systems, and with tools such as CDRECORD. (A Plextor PlexWriter burn of 614400000 bytes (300000 sectors) requires just over six minutes at 12x, using an AlphaStation XP1000 666 MHz EV67 system UltraSCSI host.) (See Section 9.7 for detailed discussions of recording optical media on OpenVMS, and the available tools.)

If you choose to attempt to use third-party devices, ensure that you have the most current OpenVMS version and the most current ECO kit(s) applied. In the specific case of the ATA (IDE) Iomega Zip250 drive, ensure that you have the most current revision of SYS$DQDRIVER installed.

14.25.2 Are the 2X-KZPCA-AA and SN-KZPCA-AA LVD Ultra2 SCSI?

Yes. Both of these controllers are Ultra2 low-voltage differential (LVD) SCSI controllers.

14.25.3 Resolving DRVERR fatal device error?

If this is on an OpenVMS version prior to V6.2, please see the AWRE and ARRE information included in section Section 14.25.

14.26 Looking for connector wiring pin-outs?

The DECconnect DEC-423 Modified Modular Jack (MMJ) appears similar to a telphone or network modular jac, though with the key offset to one side. The DECconnect MMJ connector pin-out is listed in Table 14–5, with an end-on view of the connector pins and the connector key shown below.

Table 14–5 DEC MMJ Pin-out

Pin	Description
1	Data Terminal Ready (DTR)
2	Transmit (TXD)
3	Transmit Ground (TXD-)
4	Receive Ground (RXD-)
5	Receive (RXD)
6	Data Set Ready (DSR)

```
+------------------+
| 1  2  3  4  5  6 |
+------------+    ++
             +____+
```

The BC16E-nn (where the ‘‘-nn’’ indicates the cable length) cabling and keying ‘‘flips over’’ or ‘‘crosses-over’’ the signal wires, and this allows all DECconnect MMJ connections to be wired identically; the ends of the BC16E are symmetrical and fully interchangeable, and allows either end of the cable to be connected either to the terminal or to the host. Specifically, the BC16E-nn cross-over wiring looks like this:

```
    Terminal                         Host
    MMJ                              MMJ

 DTR 1 --->---------->----------->--- 6 DSR
 TXD 2 --->---------->----------->--- 5 RXD
     3 ------------------------------ 4
     4 ------------------------------ 3
 RXD 5 ---<----------<-----------<--- 2 TXD
 DSR 6 ---<----------<-----------<--- 1 DTR
```

DECconnect parts and connections are available from HP, and MMJ crimping dies for use in typical telco-style crimping tools, and MMJ connectors, are available from Blackbox and from other communications equipment vendors.

The PC-compatible DB9 connector pin-out found on Alpha and Integrity COM serial ports—and on most PC systems is listed in Table 14–6.

Table 14–6 PC DB9 Pin-out

Pin	Description
1	Data Carrier Detect (DCD)
2	Received Data
3	Transmit Data
4	Data Terminal Ready (DTR)
5	Ground
6	Data Set Ready (DSR)
7	Request To Send (RTS)
8	Clear To Send
9	floating

The MicroVAX DB9 console connector pin-out predates the PC-style DB9 pin-out (adapters discussed in Section 14.27), and uses a then-common (and older) standard pin-out, and uses the EIA-232 series standard signals shown in Table 14–7.

Table 14–7 MicroVAX DB9 Pin-out

Pin	Description
1	Protective Ground
2	Transmited Data
3	Received Data
4	Request To Send (RTS)
5	Data Terminal Ready (DTR)
6	Data Set Ready (DSR)
7	Signal Ground
8	Shorted to pin 9 on MicroVAX and VAXstation 2000...
9	...series systems, otherwise left floating.

When pin 8 is shorted to pin 9, this is a BCC08 (or variant) cable, most commonly used as a console cable on the MicroVAX 2000 and VAXstation 2000 series. (Other systems may or may not tolerate connecting pin 8 to pin 9.)

The BN24H looks like this:

```
    MMJ       RJ45

     1---------8
     2---------2
     3---------1
     4---------3
     5---------6
     6---------7
```

The BN24J looks like this:

```
    MMJ       RJ45

     1---------7
     2---------6
     3---------3
     4---------1
     5---------2
     6---------8
```

Also see:

- http://www.hp.com/go/openvms/wizard/
- http://www.airborn.com.au/rs232.html
- http://www.stanq.com/cable.html

- For adapters and connectors, see Section 14.27.

14.27 What connectors and wiring adapters are available?

The H8571-B and H8575-B convert the (non-2000-series) MicroVAX DB9 to the DECconnect DEC-423 Modified Modular Jack (MMJ) pin-out; to the MMJ DECconnect wiring system. The MicroVAX 2000 and VAXstation 2000 requires a BCC08 cable (which has the 8-9 short, see Section 14.26) and the H8571-C or the H8571-D DB25-to-MMJ adapter for use with DECconnect. (For a discussion of the console bulkhead on the MicroVAX II series and on other closely-related series systems, please see Section 14.3.3.4.)

Somewhat less ancient HP (HP, Compaq or DIGITAL logo) systems will use either the DECconnect MMJ wiring directly or—on most (all?) recent system designs—the PC-compatible DB9 9-pin pin-out; the PC-style COM serial port interface and connection.

There are two DB9 9-pin pin-outs, that of the H8571-B and similar for the MicroVAX and other and older systems, and that of the H8571-J for the PC-style COM port, AlphaStation, Integrity, and other newer systems. The older MicroVAX DB9 and the PC-style DB9 pin-outs are not compatible.

Table 14–8 DECconnect MMJ Connectors and Adapters

Part	Converts BC16E MMJ male to fit into
H8571-A	EIA232 DB25 25-pin female (common). Functionally similar to the H8575-A, though the H8575-A has better ESD shielding.
H8571-B	Older MicroVAX (other than the MicroVAX 2000) DB9 EIA232 serial port. Functionally similar to the H8575-B, though the H8575-B has better ESD shielding. *Note: Cannot be used on a PC, Alpha nor Integrity DB9 9-pin connector.*
H8571-C	25 pin DSUB Female to MMJ, Unfiltered
H8571-D	EIA232 25 pin male (modem-wired)
H8571-E	25 pin DSUB Female to MMJ, Filtered
H8571-J	PC, Alpha, Integrity 9 pin (DB9) male (PC-style COM serial port) *Note: Cannot be used on the older MicroVAX DB9 9-pin connector*
H8572-0	BC16E MMJ double-female (MMJ extender)

Table 14–8 (Cont.) DECconnect MMJ Connectors and Adapters

Part	Converts BC16E MMJ male to fit into
H8575-A	EIA232 DB25 25-pin female (common). Functionally similar to the H8571-A, though the H8575-A has better ESD shielding.
H8575-B	Older MicroVAX (other than the MicroVAX 2000) DB9 EIA232 serial port. Functionally similar to the H8571-B, though the H8575-B has better ESD shielding. *Note: Cannot be used on a PC, Alpha nor Integrity DB9 9-pin connector*
H8575-D	25 Pin to MMJ with better ESD Protection
H8575-D	25 Pin to MMJ with better and ESD Protection
H8575-E	25 Pin Integrity rx2600 Management Processor (MP) port to MMJ, with ESD Protection
H8577-AA	6 pin Female MMJ to 8 pin MJ
BC16E-**	MMJ cable with connectors, available in various lengths

Numerous additional adapters and cables are available from the (now out of print) *OPEN DECconnect Building Wiring Components and Applications Catalog*, as well as descriptions of the above-listed parts.

The DECconnect wiring system has insufficient signaling for modems, and particularly lacks support for modem control signals.

The H8571-A and H8575-A are MMJ to DB25 (female) and other connector wiring diagrams and adapter-, cable- and pin-out-related discussions are available at:

- http://www.hp.com/go/openvms/wizard/

Jameco has offered a USB-A to PS/2 Mini DIN 6 Adapter (as part 168751), for those folks wishing to (try to) use PS/2 Keyboards via USB-A connections.

The LK463 USB keyboard is also a potential option, for those wishing to connect an OpenVMS keyboard to USB systems or (via the provided adapter) to PS/2 systems. The LK463 provides the classic OpenVMS keyboard and keyboard layout on USB-based system configurations, including operations with the USB connection on specific Alpha systems (and specifically on those with supported USB connections) and on Integrity servers.

For information on the Alpha console COM port(s) or on the VAX console port, please see Section 14.3.

14.28 What is flow control and how does it work?

XON/XOFF is one kind of flow control.

In ASCII, XON is the CTRL/Q character, and XOFF is the CTRL/S.

XON/XOFF flow control is typically associated with asynchronous serial line communications. XON/XOFF is an in-band flow control, meaning that the flow control is mixed in with the data.

CTS/RTS is another type of flow control, and is sometimes called hardware flow control. Out-of-band means that seperate lines/pins from the data lines (pins) are used to carry the CTS/RTS signals.

Both kinds of flow control are triggered when a threshold is reached in the incoming buffer. The flow control is suppose to reach the transmitter in time to have it stop transmitting before the receiver buffer is full and data is lost. Later, after a sufficient amount of the receiver's buffer is freed up, the resume flow control signal is sent to get the transmitter going again.

DECnet Phase IV on OpenVMS VAX supports the use of asynchronous serial communications as a network line; of asynch DECnet. The communication devices (eg. modems, and drivers) *must not* be configured for XON/XOFF flow control. The incidence of these (unexpected) in-band characters will corrupt data packets. Further, the serial line device drivers might normally remove the XON and XOFF characters from the stream for terminal applications, but DECnet configures the driver to pass *all* characters through and requires that all characters be permitted. (The communication devices must pass through not only the XON and XOFF characters, they must pass *all* characters including the 8-bit characters. If data compression is happening, it must reproduce the source stream exactly. No addition or elimination of null characters, and full data transparency.

An Ethernet network is rather different than an asynchronous serial line. Ethernet specifies the control of data flow on a shared segment using CSMA/CD (Carrier Sense Multiple Access, with Collision Detect) An Ethernet station that is ready to transmit listens for a clear channel (Carrier Sense). When the channel is clear, the station begins to transmit by asserting a carrier and encoding the packet appropriately. The station concurrently listens to its own signal, to permit the station to detect if another station began to transmit at the same time—this is called collision detection. (The collision corrupts the signal in a way that can reliably be detected.) Upon detecting the collision, both stations will stop transmitting, and will back off and try again a little later. (You can see a log of this activity in the DECnet NCP network counters.)

DECnet provides its own flow control, above and beyond the flow control of the physical layer (if any). The end nodes handshake at the beginning to establish a transmit window size—and a transmitter will only send that much data before stopping and waiting for an acknowledgement. The acknowledgement is only sent when the receiver has confirmed the packet is valid. (A well-configured DECnet generally avoids triggering any underlying (out-of-band) flow control mechanism.)

14.29 CD and DVD device requirements?

Read access to DVD-ROM, DVD+R/RW, DVD-R/RW, CD-ROM, and CD-R/RW devices on ATAPI (IDE) connections is generally handled transparently by SYS$DQDRIVER, and SYS$DQDRIVER will transparently de-block the media-native 2048 byte disk blocks with the 512-byte blocks expected by OpenVMS and by native OpenVMS software.

Read access to DVD-ROM, DVD+R/RW, DVD-R/RW, CD-ROM, and CD-R/RW devices on SCSI is handled by DKDRIVER, though SYS$DKDRIVER will not transparently de-block the native 2048-byte disk blocks into the 512-byte blocks expected by OpenVMS. The drive or external software is expected to provide this de-blocking, thus either a 512-byte block capable drive (such as all RRD-series SCSI CD-ROM drives) is required, or host software is required for a 2048-byte block drive. Third-party SCSI drives with UNIX references in their support documentation or with explicit 512-byte selectors or swiches will generally (but not always, of course) operate with OpenVMS.

At least some of the Plextor PlexWriter SCSI drives can be successfully accessed (for read and write) from OpenVMS, as can at least one Pioneer SCSI DVD drive (for CD media). The Pioneer SCSI DVD drive switches to 2048 byte blocks for DVD media, and a block-size conversion tool (written by Glenn Everhart) or other similar tool can be applied.

OpenVMS also has supported HP DVD drives for the ATAPI (IDE) bus.

For some related information (and details on a commercial DVDwrite package), please see:

- http://home.tiscali.de/dvd4openvms/supported_hardware.html

No device driver currently presently permits direct block-oriented recording on DVD-RAM nor DVD+RW media, nor other recordable or rewritable media.

Recording (writing) of CD and DVD optical media requires a recording or media mastering application or tool, and both commercial and non-commercial options are available. See Section 9.7 for related details on CDRECORD (both non-DVD and DVD versions are available, and at least one commercial version is available), and also see DVDwrite (commercial) or DVDRECORD (open source).

For information on the GKDRIVER (SYS$GKDRIVER) generic SCSI device driver and of the the IO$_DIAGNOSE $qio[w] interfaces (of SYS$DKDRIVER, SYS$DNDRIVER and SYS$DQDRIVER) that are utilized by most CD and DVD recording tools to send commands to SCSI, USB or ATAPI devices (most USB and ATA devices—or more correctly, most ATAPI devices—can use SCSI-like command packets), please see the SYS$EXAMPLES:GKTEST.C example, and see DECW$EXAMPLES:DECW$CDPLAYER.C example and please see the various associated sections of the OpenVMS I/O User’s Reference Manual.

For information on creating bootable optical media on OpenVMS, please see Section 9.7.3.

14.30 Office-Friendly (Quieter) Integrity rx2620 series?

The part number for the so-called Office-Friendly or Office Environment Integrity rx2620 series is AD244A.

The AD244A option is incompatible with and eliminates the redundant cooling and power supply capabilities of, but greatly reduces the sound output from, the Integrity rx2620 rack-mount computer-room variant. (Server rack-mount configurations are rated for sound output in the range of 70dB, per the HP QuickSpecs. Suitable for the intended installation into a computer room of course, but likely rather loud for an office environment installation.)

If you wish to retrofit this option into an existing Integrity rx2620 series, you must also order the HA113A1 (#53E) installation service; the option reportedly can not be installed by an end-user. Also reportedly, the option can be factory-ordered on custom configurations. If configuring for or converting an existing rack-mount configuration for the office environment, also consider acquiring the available pedestal-mounting option for the series.

14.31 KVM switches for AlphaServer and Integrity Servers?

Various folks have reported success with the Raritan series KVM (and specifically APKME the), and with the HP-branded KVM switches.

If you choose a KVM for use with Alpha or Integrity servers, do ensure it provides either Unix compatibility, X Windows compatibility, or specifies the Alpha or Integrity system. (This all usually means that the switch supports keyboards that use the so-called scanset three key-down and key-up sequences—key-down and key-up are key to the way X Windows and thus DECwindows works.)

Also see the discussion in Section 14.18 for related information, and particularly for older pinouts—for those keyboard and mouse devices that pre-date the PS/2 DIN connections, the PC VGA/SVGA video connection, and such. Almost all VAX systems pre-date these pinouts, and peripherals intended for the older Alpha systems also pre-date these pinouts.

15 Information on Networks and Clusters

If you are searching for something here, please consider using the text-format FAQ.

The following sections contain information on OpenVMS Networking with IP and DECnet, and on clustering and volume shadowing, on Fibre Channel, and on related products and configurations.

15.1 How to connect OpenVMS to a Modem?

Please see the Ask The Wizard area topics starting with (81), (1839), (2177), (3605), etc.

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

15.2 OpenVMS and IP Networking?

The following sections contain information on OpenVMS and IP networking, as well as IP printing topics.

15.2.1 How to connect OpenVMS to the Internet?

Some tutorial information and tips for connecting OpenVMS systems to the Internet are available at:

- http://www.tmesis.com/internet/

15.2.2 Connecting to an IP Printer?

To connect a printer via the IP telnet or lpr/lpd protocols, you will need to install and configure an IP stack on OpenVMS, and configure the appropriate print queue.

With current OpenVMS IP implementations, the choice of telnet or lpr/lpd really amounts to determining which of these works better with the particular printer involved.

To support network printing, the printer must include an internal or external NIC or JetDirect; an adapter connecting the network and the printer.

While it is normally possible to use a host-connected printer—when the host supports an LPD or telnet daemon, and OpenVMS and most other operating systems have the ability to serve locally-attached printers to other hosts on the network—it is generally far easier and far more effective to use a printer that is directly attached to the network. If your present printer does not have a NIC or a JetDirect, acquire an internal (if available) or external NIC or JetDirect. Or replace the printer. And obviously, most any operating system that can serve its local printers usually also provides a client that can access remote network-connected printers.

Please see the Ask The Wizard (ATW) area topics—starting with topic (1020)—for additional information on IP-based network printing.

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

Please see Section 15.2.3 for information on Postscript printing.

15.2.3 How do I connect a PostScript printer via TCP/IP?

Using TCP/IP Services (UCX) as the TCP/IP stack, it is possible to configure queues using the UCX$TELNETSYM (TCP/IP Services prior to V5.0) or TCPIP$TELNETSYM (with V5.0 and later) in order to print to Postscript printers. This assumes however that the printer itself can convert whatever is passed to it into something intelligible. As an example, if the printer has an IP address of 123.456.789.101 and jobs should be passed to port 9100 then :

```
$ INITIALIZE/QUEUE/ON="123.456.789.101:9100" -
    /PROCESSOR=UCX$TELNETSYM  -
    my_ip_queue
```

```
$ INITIALIZE/QUEUE/ON="123.456.789.101:9100" -
    /PROCESSOR=TCPIP$TELNETSYM  -
    my_ip_queue
```

The port number of 9100 is typical of HP JetDirect cards but may be different for other manufacturers cards.

As a better alternative, DCPS Version 1.4 and later support IP queues using either HP TCP/IP Services for OpenVMS software or Process Software Multinet for OpenVMS. The usage of this type of interface is documented in the DCPS documentation or release notes, and the DCPS$STARTUP.TEMPLATE startup template file.

For general and additional (non-Postscript) IP printing information, please see topic (1020) and other topics referenced in that topic elsewhere within the Ask The Wizard area.

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference. Also see:

- http://www.wotsit.org/

Please see Section 15.2.2 for pointers to an introduction to IP printing.

15.2.4 How do I set a default IP route or gateway on OpenVMS?

If you have TCP/IP Services, then use the command for TCP/IP Services V5.0 and later:

```
$ TCPIP
SET ROUTE/GATE=x.x.x.x/DEFAULT/PERMANENT
```

And for earlier TCP/IP Services versions, use the command:

```
$ UCX
SET ROUTE/GATE=x.x.x.x/DEFAULT/PERMANENT
```

15.2.5 How can I set up reverse telnet (like reverse LAT)?

Though it may seem obvious, Telnet and LAT are quite different—with differing capabilities and design goals.

Please see the documentation around the TCP/IP Services for OpenVMS TELNET command CREATE_SESSION. This command is the equivilent of the operations performed in LTLOAD.COM or LAT$SYSTARTUP.COM. There is no TELNET equivilent to the sys$qio[w] control interface for LTDRIVER (as documented in the I/O User's Reference Manual) available, though standard sys$qio[w] calls referencing the created TN device would likely operate as expected.

15.2.6 Why can't I use PPP and RAS to connect to OpenVMS Alpha?

OpenVMS Alpha IP PPP does not presently support authentication, and the Microsoft Windows NT option to disable authentication during a RAS connection apparently doesn't currently work—RAS connections will require authentication—and this will thus prevent RAS connections.

Future versions of OpenVMS and TCP/IP Services may add this, and future versions of Microsoft Windows may permit operations with authentication disabled.

15.3 OpenVMS and DECnet Networking?

The following sections contain information on OpenVMS and DECnet networking.

15.3.1 Can DECnet-Plus operate over IP?

Yes. To configure DECnet-Plus to operate over IP transport and over IP backbone networks, install and configure DECnet-Plus, and install and configure the PWIP mechanism available within the currently-installed IP stack. Within TCP/IP Services, this is a PWIPDRIVER configuration option within the UCX$CONFIG (versions prior to V5.0) or TCPIP$CONFIG (with V5.0 and later) configuration tool.

15.3.2 What does "failure on back translate address request" mean?

The error message:

```
BCKTRNSFAIL, failure on the back translate address request
```

indicates that the destination node is running DECnet-Plus, and that its naming service (DECnet-Plus DECdns, LOCAL node database, etc) cannot locate a name to associate with the source node's address. In other words, the destination node cannot determine the node name for the node that is the source of the incoming connection.

Use the DECNET_REGISTER mechanism (on the destination node) to register or modify the name(s) and the address(es) of the source node. Check the namespace on the source node, as well.

Typically, the nodes involved are using a LOCAL namespace, and the node name and address settings are not coherent across all nodes. Also check to make sure that the node is entered into its own LOCAL namespace. This can be a problem elsewhere, however. Very rarely, a cache corruption has been known to cause this error. To flush the cache, use the command:

```
$ RUN SYS$SYSTEM:NCL
flush session control naming cache entry "*"
```

Also check to see that you are using the latest ECO for DECnet-Plus for the version you are running. DECnet-Plus can use the following namespaces:

- DECdns: DECnet-Plus distributed name services.
- LocalFile: a local file containing names and addresses.
- DNS/BIND: the TCP/IP distributed name services mechanism.
- The TCP/IP Services (UCX) local host file.

Of these, searching DNS/BIND and LocalFile, respectively, is often the most appropriate configuration.

15.3.3 Performing SET HOST/MOP in DECnet-Plus?

First, issue the NCL command SHOW MOP CIRCUIT *

```
$ RUN SYS$SYSTEM:NCL
SHOW MOP CIRCUIT *
```

Assume that you have a circuit known as FDDI-0 displayed. Here is an example of the SET HOST/MOP command syntax utilized for this circuit:

```
$ SET HOST/MOP/ADDRESS=08-00-2B-2C-5A-23/CIRCUIT=FDDI-0
```

Also see Section 15.6.3.

15.3.4 How to flush the DECnet-Plus session cache?

```
$ RUN SYS$SYSTEM:NCL
FLUSH SESSION CONTROL NAMING CACHE ENTRY "*"
```

15.4 How to determine the network hardware address?

Most Alpha and most VAX systems have a console command that displays the network hardware address. Many systems will also have a sticker identifying the address, either on the enclosure or on the network controller itself.

The system console power-up messages on a number of VAX and Alpha systems will display the hardware address, particularly on those systems with an integrated Ethernet network adapter present.

If you cannot locate a sticker on the system, if the system powerup message is unavailable or does not display the address, and if the system is at the console prompt, start with the console command:

```
HELP
```

A console command similar to one of the following is typically used to display the hardware address:

```
SHOW DEVICE
SHOW ETHERNET
SHOW CONFIG
```

On the oldest VAX Q-bus systems, the following console command can be used to read the address directly off the (DELQA, DESQA, or the not-supported-in-V5.5-and-later DEQNA) Ethernet controller:

```
E/P/W/N:5 20001920
```

Look at the low byte of the six words displayed by the above command. (The oldest VAX Q-bus systems—such as the KA630 processor module used on the MicroVAX II and VAXstation II series—lack a console HELP command, and these systems typically have the primary network controller installed such that the hardware address value is located at the system physical address 20001920.)

If the system is a VAX system, and another VAX system on the network is configured to answer Maintenance and Operations Protocol (MOP) bootstrap requests (via DECnet Phase IV, DECnet-Plus, or LANCP), the MOM$SYSTEM:READ_ADDR.EXE tool can be requested:

```
B/R5:100 ddcu
Bootfile: READ_ADDR
```

Where ddcu is the name of the Ethernet controller in the above command. The primarly local DELQA, DESQA, and DEQNA Q-bus controllers are usually named XQA0. An attempt to MOP download the READ_ADDR program will ensue, and (if the download is successful) READ_ADDR will display the hardware address.

If the system is running, you can use DECnet or TCP/IP to display the hardware address with one of the following commands.

```
$! DECnet Phase IV
$ RUN SYS$SYSTEM:NCP
SHOW KNOWN LINE CHARACTERISTICS

$! DECnet-Plus
$ RUN SYS$SYSTEM:NCL
SHOW CSMA-CD STATION * ALL STATUS

$! TCP/IP versions prior to V5.0
$ UCX
SHOW INTERFACE/FULL

$! TCP/IP versions V5.0 and later
$ TCPIP
SHOW INTERFACE/FULL
```

A program can be created to display the hardware address, reading the necessary information from the network device drivers. A complete example C program for reading the Ethernet or IEEE 802.3 network controller hardware address (via sys$qio calls to the OpenVMS network device driver(s)) is available at the following URL:

- http://www.hp.com/go/openvms/wizard/

To use the DECnet Phase IV configurator tool to watch for MOP SYSID activity on the local area network:

```
$ RUN SYS$SYSTEM:NCP
SET MODULE CONFIGURATOR KNOWN CIRCUIT SURVEILLANCE ENABLED
```

Let the DECnet Phase IV configurator run for at least 20 minutes, and preferably longer. Then issue the following commands:

```
$ RUN SYS$SYSTEM:NCP
SHOW MODULE CONFIGURATOR KNOWN CIRCUIT STATUS TO filename.txt
SET MODULE CONFIGURATOR KNOWN CIRCUIT SURVEILLANCE DISABLED
```

The resulting file (named filename.txt) can now be searched for the information of interest. Most DECnet systems will generate MOP SYSID messages identifying items such as the controller hardware address and the controller type, and these messages are generated and multicast roughly every ten minutes.

Information on the DECnet MOP SYSID messages and other parts of the maintenance protocols is included in the DECnet network architecture specifications referenced in section DOC9.

15.4.1 How do I reset the LAN (DECnet-Plus NCL) error counters?

On recent OpenVMS releases:

```
$ RUN SYS$SYSTEM:LANCP
SET DEVICE/DEVICE_SPECIFIC=FUNCTION="CCOU" devname
```

15.4.2 How do I install DECnet Phase IV on VMS 7.1?

On OpenVMS V7.1, all DECnet binaries were relocated into separate installation kits—you can selectively install the appropriate network: DECnet-Plus (formerly known as DECnet OSI), DECnet Phase IV, and HP TCP/IP Services (often known as UCX).

On OpenVMS versions prior to V7.1, DECnet Phase IV was integrated, and there was no installation question. You had to install the DECnet-Plus (DECnet/OSI) package on the system, after the OpenVMS upgrade or installation completed.

During an OpenVMS V7.1 installation or upgrade, the installation procedure will query you to learn if DECnet-Plus should be installed. If you are upgrading to V7.1 from an earlier release or are installing V7.1 from a distribution kit, simply answer "NO" to the question asking you if you want DECnet-Plus. Then—after the OpenVMS upgrade or installation completes – use the PCSI PRODUCT INSTALL command to install the DECnet Phase IV binaries from the kit provided on the OpenVMS software distribution kit.

If you already have DECnet-Plus installed and wish to revert, you must reconfigure OpenVMS. You cannot reconfigure the "live" system, hence you must reboot the system using the V7.1 distribution CD-ROM. Then select the DCL ($$$ prompt) option. Then issue the commands:

```
$$$ DEFINE/SYSTEM PCSI$SYSDEVICE DKA0:
$$$ DEFINE/SYSTEM PCSI$SPECIFIC DKA0:[SYS0.]
$$$ PRODUCT RECONFIGURE VMS /REMOTE/SOURCE=DKA0:[VMS$COMMON]
```

The above commands assume that the target system device and system root are ‘‘DKA0:[SYS0.]’’. Replace this with the actual target device and root, as appropriate. The RECONFIGURE command will then issue a series of prompts. You will want to reconfigure DECnet-Plus off the system, obviously. You will then want to use the PCSI command PRODUCT INSTALL to install the DECnet Phase IV kit from the OpenVMS distribution media.

Information on DECnet support, and on the kit names, is included in the OpenVMS V7.1 installation and upgrade documentation.

Subsequent OpenVMS upgrade and installation procedures can and do offer both DECnet Phase IV and DECnet-Plus installations.

15.5 How can I send (radio) pages from my OpenVMS system?

There are third-party products available to send messages to radio paging devices (pagers), communicating via various protocols such as TAP (Telocator Alphanumeric Protocol); paging packages.

RamPage (Ergonomic Solutions) is one of the available packages that can generate and transmit messages to radio pagers. Target Alert (Target Systems; formerly the DECalert product) is another. Networking Dynamics Corp has a product called Pager Plus. The System Watchdog package can also send pages. The Process Software package PMDF can route specific email addresses to a paging service, as well.

Many commercial paging services provide email contact addresses for their paging customers—you can simply send or forward email directly to the email address assigned to the pager.

Some people implement the sending of pages to radio pagers by sending commands to a modem to take the "phone" off the "hook", and then the paging sequence, followed by a delay, and then the same number that a human would dial to send a numeric page. (This is not entirely reliable, as the modem lacks "call progress detection", and the program could simply send the dial sequence when not really connected to the paging company’s telephone-based dial-up receiver.)

See Section 13.1 for information on the available catalog of products.

15.6 OpenVMS, Clusters, Volume Shadowing?

The following sections contain information on OpenVMS and Clusters, Volume Shadowing, and Cluster-related system parameters.

15.6.1 OpenVMS Cluster Communications Protocol Details?

The following sections contain information on the OpenVMS System Communications Services (SCS) Protocol. Cluster terminology is available in Section 15.6.1.2.1.

15.6.1.1 OpenVMS Cluster (SCS) over DECnet? Over IP?

The OpenVMS Cluster environment operates over various network protocols, but the core of clustering uses the System Communications Services (SCS) protocols, and SCS-specific network datagrams. Direct (full) connectivity is assumed.

An OpenVMS Cluster *does not* operate over DECnet, nor over IP.

No SCS protocol routers are available.

Many folks have suggested operating SCS over DECnet or IP over the years, but SCS is too far down in the layers, and any such project would entail a major or complete rewrite of SCS and of the DECnet or IP drivers. Further, the current DECnet and IP implementations have large tracts of code that operate at the application level, while SCS must operate in the rather more primitive contexts of the system and particularly the bootstrap—to get SCS to operate over a DECnet or IP connection would require relocating major portions of the DECnet or IP stack into the kernel. (And it is not clear that the result would even meet the bandwidth and latency expectations.)

The usual approach for multi-site OpenVMS Cluster configurations involves FDDI, Memory Channel (MC2), or a point-to-point remote bridge, brouter, or switch. The connection must be transparent, and it must operate at 10 megabits per second or better (Ethernet speed), with latency characteristics similar to that of Ethernet or better. Various sites use FDDI, MC2, ATM, or point-to-point T3 link.

15.6.1.2 Configuring Cluster SCS for path load balancing?

This section discusses OpenVMS Cluster communications, cluster terminology, related utilities, and command and control interfaces.

15.6.1.2.1 Cluster Terminology?

SCS: Systems Communication Services. The protocol used to communicate between VMSCluster systems and between OpenVMS systems and SCS-based storage controllers. (SCSI-based storage controllers do not use SCS.)

PORT: A communications device, such as DSSI, CI, Ethernet or FDDI. Each CI or DSSI bus is a different local port, named PAA0, PAB0, PAC0 etc. All Ethernet and FDDI busses make up a single PEA0 port.

VIRTUAL CIRCUIT: A reliable communications path established between a pair of ports. Each port in a VMScluster establishes a virtual circuit with every other port in that cluster.

All systems and storage controllers establish "Virtual Circuits" to enable communications between all available pairs of ports.

SYSAP: A "system application" that communicates using SCS. Each SYSAP communicates with a particular remote SYSAP. Example SYSAPs include:

VMS$DISK_CL_DRIVER connects to MSCP$DISK
The disk class driver is on every VMSCluster system. MSCP$DISK is on all disk controllers and all VMSCluster systems that have SYSGEN parameter MSCP_LOAD set to 1

VMS$TAPE_CL_DRIVER connects to MSCP$TAPE
The tape class driver is on every VMSCluster system. MSCP$TAPE is on all tape controllers and all VMSCluster systems that have SYSGEN parameter TMSCP_LOAD set to 1

VMS$VAXCLUSTER connects to VMS$VAXCLUSTER
This SYSAP contains the connection manager, which manages cluster connectivity, runs the cluster state transition algorithm, and implements the cluster quorum algorithm. This SYSAP also handles lock traffic, and various other cluster communications functions.

SCS$DIR_LOOKUP connects to SCS$DIRECTORY
This SYSAP is used to find SYSAPs on remote systems

MSCP and TMSCP
The Mass Storage Control Protocol and the Tape MSCP servers are SYSAPs that provide access to disk and tape storage, typically operating over SCS protocols. MSCP and TMSCP SYSAPs exist within OpenVMS (for OpenVMS hosts serving disks and tapes), within CI- and DSSI-based storage controllers, and within host-based MSCP- or TMSCP storage controllers. MSCP and TMSCP can be used to serve MSCP and TMSCP storage devices, and can also be used to serve SCSI and other non-MSCP/non-TMSCP storage devices.

SCS CONNECTION: A SYSAP on one node establishes an SCS connection to its counterpart on another node. This connection will be on ONE AND ONLY ONE of the available virtual circuits.

15.6.1.2.2 Cluster Communications Control?

When there are multiple virtual circuits between two OpenVMS systems it is possible for the VMS$VAXCLUSTER to VMS$VAXCLUSTER connection to use any one of these circuits. All lock traffic between the two systems will then travel on the selected virtual circuit.

Each port has a ‘‘LOAD CLASS’’ associated with it. This load class helps to determine which virtual circuit a connection will use. If one port has a higher load class than all others then this port will be used. If two or more ports have equally high load classes then the connection will use the first of these that it finds. Prior to enhancements found in V7.3-1 and later, the load class is static and normally all CI and DSSI ports have a load class of 14(hex), while the Ethernet and FDDI ports will have a load class of A(hex). With V7.3-1 and later, the load class values are dynamic.

For instance, if you have multiple DSSI busses and an FDDI, the VMS$VAXCLUSTER connection will chose the DSSI bus as this path has the system disk, and thus will always be the first DSSI bus discovered when the OpenVMS system boots.

To force all lock traffic off the DSSI and on to the FDDI, for instance, an adjustment to the load class value is required, or the DSSI SCS port must be disabled.

In addition to the load class mechanisms, you can also use the ‘‘preferred path’’ mechanisms of MSCP and TMSCP services. This allows you to control the SCS connections used for serving remote disk and tape storage. The preferred path mechanism is most commonly used to explicitly spread cluster I/O activity over hosts and/or storage controllers serving disk or tape storage in parallel. This can be particularly useful if your hosts or storage controllers individually lack the necessary I/O bandwidth for the current I/O load, and must thus aggregate bandwidth to serve the cluster I/O load.

For related tools, see various utilities including LAVC$STOP_BUS and LAVC$START_BUS, and see DCL commands including SET PREFERRED_PATH.

15.6.1.2.3 Cluster Communications Control Tools and Utilities?

In most OpenVMS versions, you can use the tools:

- SYS$EXAMPLES:LAVC$STOP_BUS
- SYS$EXAMPLES:LAVC$START_BUS

These tools permit you to disable or enable all SCS traffic on the on the specified paths.

You can also use a preferred path mechanism that tells the local MSCP disk class driver (DUDRIVER) which path to a disk should be used. Generally, this is used with dual-pathed disks, forcing I/O traffic through one of the controllers instead of the other. This can be used to implement a crude form of I/O load balancing at the disk I/O level.

Prior to V7.2, the preferred path feature uses the tool:

- SYS$EXAMPLES:PREFER.MAR

In OpenVMS V7.2 and later, you can use the following DCL command:

```
$ SET PREFERRED_PATH
```

The preferred path mechanism does not disable nor affect SCS operations on the non-preferred path.

With OpenVMS V7.3 and later, please see the SCACP utility for control over cluster communications, SCS virtual circuit control, port selection, and related.

15.6.2 Cluster System Parameter Settings?

The following sections contain details of configuring cluster-related system parameters.

15.6.2.1 What is the correct value for EXPECTED_VOTES in a VMScluster?

The VMScluster connection manager uses the concept of votes and quorum to prevent disk and memory data corruptions—when sufficient votes are present for quorum, then access to resources is permitted. When sufficient votes are not present, user activity will be blocked. The act of blocking user activity is called a "quorum hang", and is better thought of as a "user data integrity interlock". This mechanism is designed to prevent a partitioned VMScluster, and the resultant massive disk data corruptions. *The quorum mechanism is expressly intended to prevent your data from becoming severely corrupted.*

On each OpenVMS node in a VMScluster, one sets two values in SYSGEN: VOTES, and EXPECTED_VOTES. The former is how many votes the node contributes to the VMScluster. The latter is the total number of votes expected when the full VMScluster is bootstrapped.

Some sites erroneously attempt to set EXPECTED_VOTES too low, believing that this will allow when only a subset of voting nodes are present in a VMScluster. It does not. Further, an erroneous setting in EXPECTED_VOTES is automatically corrected once VMScluster connections to other nodes are established; user data is at risk of severe corruptions during the earliest and most vulnerable portion of the system bootstrap, before the connections have been established.

One can operate a VMScluster with one, two, or many voting nodes. With any but the two-node configuration, keeping a subset of the nodes active when some nodes fail can be easily configured. With the two-node configuration, one must use a primary-secondary configuration (where the primary has all the votes), a peer configuration (where when either node is down, the other hangs), or (preferable) a shared quorum disk.

Use of a quorum disk does slow down VMScluster transitions somewhat – the addition of a third voting node that contributes the vote(s) that would be assigned to the quorum disk makes for faster transitions—but the use of a quorum disk does mean that either node in a two-node VMScluster configuration can operate when the other node is down.

Note

The quorum disk must be on a non-host-based shadowed disk, though it can be protected with controller-based RAID. Because host-based volume shadowing depends on the lock manager and the lock manager depends on the connection manager and the connection manager depends on quorum, it is not technically feasible (nor

even particularly reliable) to permit host-based volume shadowing to protect the quorum disk.

If you choose to use a quoum disk, a QUORUM.DAT file will be automatically created when OpenVMS first boots and when a quorum disk is specified – well, the QUORUM.DAT file will be created when OpenVMS is booted without also needing the votes from the quorum disk.

In a two-node VMScluster with a shared storage interconnect, typically each node has one vote, and the quorum disk also has one vote. EXPECTED_VOTES is set to three.

Using a quorum disk on a non-shared interconnect is unnecessary—the use of a quorum disk does not provide any value, and the votes assigned to the quorum disk should be assigned to the OpenVMS host serving access to the disk.

For information on quorum hangs, see the OpenVMS documentation. For information on changing the EXPECTED_VOTES value on a running system, see the SET CLUSTER/EXPECTED_VOTES command, and see the documentation for the AMDS and Availability Manager tools. Also of potential interest is the OpenVMS system console documentation for the processor-specific console commands used to trigger the IPC (Interrrupt Priority Level %x0C; IPL C) handler. (IPC is not available on OpenVMS I64 V8.2.) AMDS, Availability Manager, and the IPC handler can each be used to clear a quorum hang. Use of AMDS and Availability Manager is generally recommended over IPC, particularly because IPC can cause CLUEXIT bugchecks if the system should remain halted beyond the cluster sanity timer limits, and because some Alpha consoles and most (all?) Integrity consoles do not permit a restart after a halt.

The quorum scheme is a set of ‘‘blade guards’’ deliberately implemented by OpenVMS Engineering to provide data integrity—remove these blade guards at your peril. OpenVMS Engineering did *not* implement the quorum mechanism to make a system manager’s life more difficult— *the quorum mechanism was specifically implemented to keep your data from getting scrambled.*

15.6.2.1.1 Why no shadowing for a Quorum Disk?

Stated simply, Host-Based Volume Shadowing uses the Distributed Lock Manager (DLM) to coordinate changes to membership of a shadowset (e.g. removing a member). The DLM depends in turn on the Connection Manager enforcing the Quorum Scheme and deciding which node(s) (and quorum disk) are participating in the cluster, and telling the DLM when it needs to do things like a lock database rebuild operation. So you can't introduce a dependency of the Connection Manager on Shadowing to try to pick proper shadowset member(s) to use as the Quorum Disk when Shadowing itself is using the DLM and thus indirectly depending on the Connection Manager to keep the cluster membership straight—it's a circular dependency.

So in practice, folks simply depend on controller-based mirroring (or controller-based RAID) to protect the Quorum Disk against disk failures (and dual-redundant controllers to protect against most cases of controller and interconnect failures). Since this disk unit appears to be a single disk up at the VMS level, there's no chance of ambiguity.

15.6.2.2 Explain disk (or tape) allocation class settings?

The allocation class mechanism provides the system manager with a way to configure and resolve served and direct paths to storage devices within a cluster. Any served device that provides multiple paths should be configured using a non-zero allocation class, either at the MSCP (or TMSCP) storage controllers, at the port (for port allocation classes), or at the OpenVMS MSCP (or TMSCP) server. All controllers or servers providing a path to the same device should have the same allocation class (at the port, controller, or server level).

Each disk (or tape) unit number used within a non-zero disk (or tape) allocation class must be unique, regardless of the particular device prefix. For the purposes of multi-path device path determination, any disk (or tape) device with the same unit number and the same disk (or tape) allocation class configuration is assumed to be the same device.

If you are reconfiguring disk device allocation classes, you will want to avoid the use of allocation class one (1) until/unless you have Fibre Channel storage configured. (Fibre Channel storage specifically requires the use of allocation class 1. eg: 1DGA0:.)

15.6.2.2.1 How to configure allocation classes and Multi-Path SCSI?

The HSZ allocation class is applied to devices, starting with OpenVMS V7.2. It is considered a port allocation class (PAC), and all device names with a PAC have their controller letter forced to "A". (You might infer from the the text in the "Guidelines for OpenVMS Cluster Configurations" that this is something you have to do, though OpenVMS will thoughtfully handle this renaming for you.)

You can force the device names back to DKB by setting the HSZ allocation class to zero, and setting the PKB PAC to -1. This will use the host allocation class, and will leave the controller letter alone (that is, the DK controller letter will be the same as the SCSI port (PK) controller). Note that this won't work if the HSZ is configured in multibus failover mode. In this case, OpenVMS requires that you use an allocation class for the HSZ.

When your configuration gets even moderately complex, you must pay careful attention to how you assign the three kinds of allocation class: node, port and HSZ/HSJ, as otherwise you could wind up with device naming conflicts that can be painful to resolve.

The display-able path information is for SCSI multi-path, and permits the multi-path software to distinguish between different paths to the same device. If you have two paths to 1DKA100, for example by having two KZPBA controllers and two SCSI buses to the HSZ, you would have two UCBs in a multi-path set. The path information is used by the multi-path software to distinguish between these two UCBs.

The displayable path information describes the path; in this case, the SCSI port. If port is PKB, that's the path name you get. The device name is no longer completely tied to the port name; the device name now depends on the various allocation class settings of the controller, SCSI port or node.

The reason the device name's controller letter is forced to "A" when you use PACs is because a shared SCSI bus may be configured via different ports on the various nodes connected to the bus. The port may be PKB on one node, and PKC on the other. Rather obviously, you will want to have the shared devices use the same device names on all nodes. To establish this, you will assign the same PAC on each node, and OpenVMS will force the controller letter to be the same on each node. Simply choosing "A" was easier and more deterministic than negotiating the controller letter between the nodes, and also parallels the solution used for this situation when DSSI or SDI/STI storage was used.

To enable port allocation classes, see the SYSBOOT command SET/BOOT, and see the DEVICE_NAMING system parameter.

This information is also described in the Cluster Systems and Guidelines for OpenVMS Cluster Configurations manuals.

15.6.3 Tell me about SET HOST/DUP and SET HOST/HSC

The OpenVMS DCL commands SET HOST/DUP and SET HOST/HSC are used to connect to storage controllers via the Diagnostics and Utility Protocol (DUP). These commands require that the FYDRIVER device driver be connected. This device driver connection is typically performed by adding the following command(s) into the system startup command procedure:

On OpenVMS Alpha:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> IO CONNECT FYA0/NOADAPTER/DRIVER=SYS$FYDRIVER
```

On OpenVMS VAX:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT FYA0/NOADAPTER
```

Alternatives to the DCL SET HOST/DUP command include the console SET HOST command available on various mid- to recent-vintage VAX consoles:

Access to Parameters on an Embedded DSSI controller:

```
SET HOST/DUP/DSSI[/BUS:{0:1}] dssi_node_number PARAMS
```

Access to Directory of tools on an Embedded DSSI controller:

```
SET HOST/DUP/DSSI[/BUS:{0:1}] dssi_node_number DIRECT
```

Access to Parameters on a KFQSA DSSI controller:

```
SHOW UQSSP ! to get port_controller_number PARAMS
SET HOST/DUP/UQSSP port_controller_number PARAMS
```

These console commands are available on most MicroVAX and VAXstation 3xxx series systems, and most (all?) VAX 4xxx series systems. For further information, see the system documentation and—on most VAX systems—see the console HELP text.

EK-410AB-MG, _DSSI VAXcluster Installation and Troubleshooting_, is a good resource for setting up a DSSI VMScluster on OpenVMS VAX nodes. (This manual predates coverage of OpenVMS Alpha systems, but gives good coverage to all hardware and software aspects of setting up a DSSI-based VMScluster—and most of the concepts covered are directly applicable to OpenVMS Alpha systems. This manual specifically covers the hardware, which is something not covered by the standard OpenVMS VMScluster documentation.)

Also see Section 15.3.3, and for the SCS name of the OpenVMS host see Section 5.7.

15.6.4 How do I rename a DSSI disk (or tape?)

If you want to renumber or rename DSSI disks or DSSI tapes, it's easy—if you know the secret incantation...

From OpenVMS:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT FYA0/NOADAPTER
SYSGEN> ^Z
$ SET HOST/DUP/SERV=MSCP$DUP/TASK=PARAMS <DSSI-NODE-NAME>
...
PARAMS> STAT CONF
<The software version is normally near the top of the display.>
PARAMS> EXIT
...
```

From the console on most 3000- and 4000-class VAX system consoles... (Obviously, the system must be halted for these commands...)

Integrated DSSI:

```
SET HOST/DUP/DSSI[/BUS:[0:1]] dssi_node_number PARAMS
```

KFQSA:

```
SET HOST/DUP/UQSSP port_controller_number PARAMS
```

For information on how to get out into the PARAMS subsystem, also see the HELP at the console prompt for the SET HOST syntax, or see the HELP on SET HOST /DUP (once you've connected FYDRIVER under OpenVMS).

Once you are out into the PARAMS subsystem, you can use the FORCEUNI option to force the use of the UNITNUM value and then set a unique UNITNUM inside each DSSI ISE—this causes each DSSI ISE to use the specfied unit number and not use the DSSI node as the unit number. Other parameters of interest are NODENAME and ALLCLASS, the node name and the (disk or tape) cluster allocation class.

Ensure that all disk unit numbers used within an OpenVMS Cluster disk allocation class are unique, and all tape unit numbers used within an OpenVMS Cluster tape allocation class are also unique. For details on the SCS name of the OpenVMS host, see Section 5.7. For details of SET HOST/DUP, see Section 15.6.3.

15.6.5 Where can I get Fibre Channel Storage (SAN) information?

- http://www.openvms.compaq.com/openvms/fibre/index.html

15.6.6 Which files must be shared in an OpenVMS Cluster?

The following files are expected to be common across all nodes in a cluster environment, and though SYSUAF is very often common, it can also be carefully coordinated—with matching UIC values and matching binary identifier values across all copies. (The most common use of multiple SYSUAF files is to allow different quotas on different nodes. In any event, the binary UIC values and the binary identifier values must be coordinated across all SYSUAF files, and must match the RIGHTSLIST file.) In addition to the list of files (and directories, in some cases) shown in Table 15–1, please review the VMScluster documentation, and the System Management documentation.

Table 15–1 Cluster Common Shared Files

Filename	Default Specification
SYSUAF	SYS$SYSTEM:.DAT
SYSUAFALT	SYS$SYSTEM:.DAT
SYSALF	SYS$SYSTEM:.DAT
RIGHTSLIST	SYS$SYSTEM:.DAT
NETPROXY	SYS$SYSTEM:.DAT

Table 15–1 (Cont.) Cluster Common Shared Files

Filename	Default Specification
NET$PROXY	SYS$SYSTEM:.DAT
NETOBJECT	SYS$SYSTEM:.DAT
NETNODE_REMOTE	SYS$SYSTEM:.DAT
QMAN$MASTER	SYS$SYSTEM:; this is a set of related files
LMF$LICENSE	SYS$SYSTEM:.LDB
VMSMAIL_PROFILE	SYS$SYSTEM:.DATA
VMS$OBJECTS	SYS$SYSTEM:.DAT
VMS$AUDIT_SERVER	SYS$MANAGER:.DAT
VMS$PASSWORD_HISTORY	SYS$SYSTEM:.DATA
NETNODE_UPDATE	SYS$MANAGER:.COM
VMS$PASSWORD_POLICY	SYS$LIBRARY:.EXE
LAN$NODE_DATABASE	SYS$SYSTEM:.DAT
VMS$CLASS_SCHEDULE	SYS$SYSTEM:.DATA
SYS$REGISTRY	SYS$SYSTEM:; this is a set of related files

In addition to the documentation, also see the current version of the file SYS$STARTUP:SYLOGICALS.TEMPLATE. Specifically, please see the most recent version of this file available, starting on or after OpenVMS V7.2.

A failure to have common or (in the case of multiple SYSUAF files) synchronized files can cause problems with batch operations, with the SUBMIT/USER command, with the general operations with the cluster alias, and with various SYSMAN and related operations. Object protections and defaults will not necessarily be consistent, as well. This can also lead to system security problems, including unintended access denials and unintended object accesses, should the files and particularly should the binary identifier values become skewed.

15.6.7 How can I split up an OpenVMS Cluster?

Review the VMScluster documentation, and the System Management documentation. The following are the key points, but are likely not the only things you will need to change.

OpenVMS Cluster support is directly integrated into the operating system, and there is no way to remove it. You can, however, remote site-specific tailoring that was added for a particular cluster configuration.

First: Create restorable image BACKUPs of each of the current system disks. If something gets messed up, you want a way to recover, right?

Create standalone BACKUP kits for the OpenVMS VAX systems, and create or acquire bootable BACKUP kits for the OpenVMS Alpha systems.

Use CLUSTER_CONFIG or CLUSTER_CONFIG_LAN to remove the various system roots and to shut off boot services and VMScluster settings.

Create as many architecture-specific copies of the system disks as required. Realize that the new systems will all likely be booting through root SYS0—if you have any system-specific files in any other roots, save them.

Relocate the copies of the VMScluster common files onto each of the new system disks.

Reset the console parameters and boot flags on each system for use on a standalone node.

Reset the VAXCLUSTER and NISCS_LOAD_PEA0 parameters to 0 in SYSGEN and in MODPARAMS.DAT.

Clobber the VMScluster group ID and password using SYSMAN.

Reboot the systems seperately, and run AUTOGEN on each.

Shut off MOP services via NCP or LANCP on the boot server nodes.

Permanent seperation also requires the duplication of shared files. For a list of the files commonly shared, please see Section 15.6.6.

Also see the topics on ‘‘cluster divorce’’ in the Ask The Wizard area.

- http://www.hp.com/go/openvms/wizard/ (retired; use ITRC forums)

For additional information on the OpenVMS Ask The Wizard (ATW) area and for a pointer to the available ATW Wizard.zip archive, please see Section 3.8. ATW has been superceded (for new questions) by the ITRC discussion forums; the area remains available for reference.

Information on changing node names is included in Section 5.7.

15.6.8 Details on Volume Shadowing?

This section contains information on host-based volume shadowing; on the disk mirroring capabilities available within OpenVMS.

15.6.8.1 Does volume shadowing require a non-zero allocation classes?

Yes, use of host-based Volume Shadowing requires that the disk(s) involved be configured in a non-zero allocation class.

Edit SYS$SYSTEM:MODPARAMS.DAT to include a declaration of an non-zero allocation class, such as setting the host allocation class to the value 7:

```
ALLOCLASS = 7
```

Then AUTOGEN the system, and reboot.

You should now be able to form the shadow set via a command such as the following:

```
$ MOUNT dsa1007: /SHADOW=($7$dkb300:,$7$dkb500:) volumelabel
```

When operating in an OpenVMS Cluster, this sequence will typically change the disk names from the SCSNODE prefix (scsnode$dkann) to the allocation-class prefix (7dkannn). This may provide you with the opportunity to move to a device-independent scheme using logical name constructs such as the DISK$volumelabel logical names in your startup and application environments; an opportunity to weed out physical device references.

Allocation class one is used by Fibre Channel devices; it can be best to use another non-zero allocation class even if Fibre Channel is not currently configured and not currently planned.

Index