

Using OpenVMS Migration Software for VAX to Alpha Systems (OMSVA)

Identifying Restrictions for Translating VAX Images

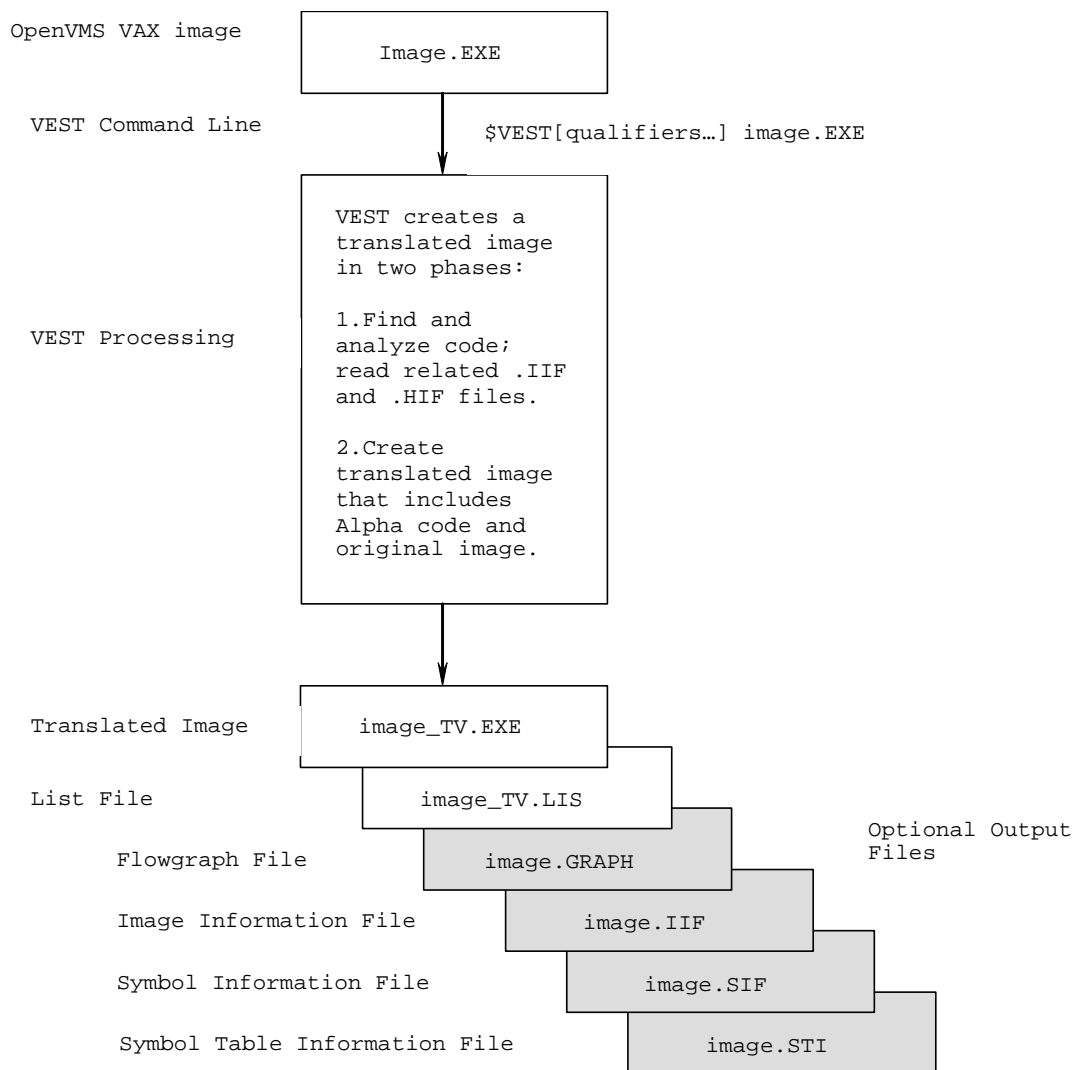
© 2003 Software Resources International

Using OpenVMS MSVA in the binary translation feasibility analysis is a major step in the migration strategy definition.

In order to understand if binary translation approach is applicable, OpenVMS MSVA package provides a wide range of analysis features and possibilities to fix translation problems that are of course limited if no application source code is available.

You can use the VAX Environment Software Translator (VEST) utility itself to identify most translation and performance restrictions in OpenVMS VAX images.

By using the OpenVMS MSVA translation diagnostic messages it is possible to identify source code problems that can affect how you rebuild an application for the OpenVMS AXP operating system, and what could be done for improving the translated image performance.



During the VEST code analysis phase, messages that flag and describe problematic code encountered are issued. If any message severity equals to either FATAL or ERROR, VEST does not create a translated image. If the most severe message level is WARNING, VEST

creates a translated image that can run properly on an OpenVMS AXP system. However, Compaq recommends that you examine each WARNING message carefully.

For the cases where source code is not available or its modification is problematic it is possible to fix binary translation by using image information files.

Four categories are identified according to the severity of the Translation problems. The tables below summarize the specific VEST messages for each category.

Untranslatable Images

Images described in the table incur FATAL or ERROR messages.

Description	Message
Images linked prior to VAX VMS Version 4.0	BADEXE
Images that have user-written system services or other nonuser-mode code	ISDPROTECT ISDVECTOR
Images linked against a specific version of VMS and that reference system space	LNKSYS
Images that make direct reference to system space addresses	UNSUPABSREF
Images that make absolute reference to P1 space outside the supported range of OpenVMS VAX system service vector entries, which includes all system services prefixed by SYS\$ (SYS\$QIOW through SYS\$EVDPOSTEVENT)	UNSUPABSREF
Images that make absolute reference to P0 space addresses not mapped by any image	UNSUPABSREF
Images that are based, that is, they have been linked such that code and data are tied to fixed addresses	ISDBASED
Images that cannot be translated because VEST cannot include both shared and unshared writeable image sections in a single Alpha AXP page	ISDCONFLICT

Images Translatable with Warnings

Images described in the table incur WARNING messages.

Description	Message
Images that contain vector instructions	VECTOROPC
Images that contain privileged instructions	PRIVOPC
Images that modify return addresses	WRITECF4 WRITEJSBRET
Images that modify the call frame	WRITECF0 WRITECF1 WRITECF2 WRITECF3

It is recommended that you examine each WARNING message carefully. You may determine that the code flagged is not likely to interfere with the image running translated. In other cases, that may not be true and you may need to take steps to ensure that the translated version of the image executes properly on an OpenVMS AXP system. You may need to make changes to the source code, if available, and then recompile and re-link the image before translating. Or you may need to re-link the image with different options. Or you may be able to eliminate the warning by using specific VEST switches when you translate the image.

Note that VEST can also issue WARNING messages for other reasons than those identified in the table above.

Images with Undetectable Translation Problems

The image features described in the table will not incur error messages but represent compatibility problems not detectable during translation.

Description	What to do
Images that depend on 512-byte page protection granularity	Recompiling and Relinking Applications may be required
Images that depend on global sections being aligned on 512-byte boundaries	Recompiling and Relinking Applications may be required

Description	What to do
Images that use most of the VAXP0 or P1 space or are otherwise sensitive to the space taken up by the Translated Image Environment (TIE) and translated code, for example, an image that allocates a lot of dynamic memory	Undetectable

Two of the three features relate to the increase in page size from OpenVMS VAX to OpenVMS AXP systems. Page size can affect the amount of virtual memory allocated by memory management routines and system services. It is also the basis on which protection is assigned to code and data in memory. The OpenVMS VAX operating system allocates memory in multiples of 512 bytes, whereas OpenVMS AXP pages range in size from 8KB to 64KB, depending on the specific hardware platform.

When VEST translates an image, it integrates multiple small VAX pages into a single 64KB Alpha AXP page. The 64KB page assumes the most permissive protection of all the VAX pages combined within it. In most cases, combining multiple VAX pages within a single Alpha AXP page does not affect the translated image's function. However, if your code has built-in dependencies on 512-byte granularity, the translated image may not run properly. For example, suppose your image depends on trapping attempts to write to a read-only section that has been combined into a read-write page in the translated image. When the translated image runs on an OpenVMS AXP system, the trap will not occur.

Translatable Images with Performance Issues

Images described in the table are translatable but include code that affects translated image performance (for example, requires interpretation at run time).

Description	Message
Images that include self-modifying VAX code, which must be interpreted	RWINTERP
Images that generate VAX code at run time, which must be interpreted	Undetectable
Images with code that inspects the instruction stream (I-stream); VEST generates slower RET and RSB code in this case	READCF4 READJSBRET
Images that depend on D_floating, H_floating, or packed-decimal instructions; these instructions require software emulation	VAXDFLOAT VAXHFLOAT VAXPACKED

In addition to the pointers to performance problems that VEST offers, you can use TIE run-time statistics to discover exactly how the image uses resources that slow performance.

Important Note:

This document is provided for information purposes only. The information contained in this document does not represent any commitment or obligation by Software Resources International or by Compaq/HP. Software Resources International may decide any time, without prior notice, to no longer provide this information, or change its contents, at its sole discretion.