VMS Software

# Application Evaluation Checklist

Use this checklist to guide you through the planning and porting process.

## General Information

1. What does your application do?

_____

_____

_____

_____

## Development History and Plans

2. a. What operating systems and hardware architectures does the application currently run on?

_____

_____

   b. Does the application currently run on the latest version for this architecture? □ YES □ NO

*If your application already runs on multiple platforms, it is likely to be platform independent, and therefore, easier to port. However, if the application code is conditionalized per platform, then minor modifications might be required. For more information, see section "Conditionalized Code" of the Porting Applications from VSI OpenVMS Alpha to VSI OpenVMS Industry Standard 64 for Integrity Servers Manual.*

*If you answer YES to b, your application will be easier to port.*

3. When was the last time the application environment was completely recompiled and rebuilt from source?
   a. Is the application rebuilt regularly? □ YES □ NO
   b. How frequently? _____

*Applications that are not built frequently might require additional work before being ported. For example, changes in the development environment might cause incompatibilities. Prior to porting your applications, confirm that they can be built on the latest version of OpenVMS for this architecture. For latest version information, refer to the VMS Software official website (https://vmssoftware.com/products/versions/).*

4. Is the application actively maintained by developers who know it well? □ YES □ NO

   List developer names and contact information:

| Developer | Contact |
|---|---|
| | |
| | |
| | |

5. a. How is a new build of the application tested or verified for proper operation?

   b. Do you have performance characterization tools to assist with optimization? □ YES □ NO

      If YES, list the tools and their version numbers:

| Tool | Version |
|---|---|
| | |
| | |
| | |

   c. Which source-code configuration management tools (e.g. CMS) are used? List the tools and their version numbers:

| Tool | Version |
|---|---|
| | |
| | |
| | |

6. Do you have a development and test environment separate from your production systems? □ YES □ NO

7. What procedures are in place for rolling out a new version of the application into production?

# Composition of the Application

Consider the size and components of your application that need to be ported. This will help you "size" the effort and the resources required for porting. Most helpful is knowing the number of lines of code.

8. a. How large is your application? _____
   b. How many modules does it have? _____
   c. How many lines of code? _____
   d. How much disk space is required? _____

9. a. Do you have access to all source files that make up your application? □ YES □ NO
   b. If you are considering using VSI Services, will it be possible to give VSI access to these source files and build procedures? □ YES □ NO

10. a. List the languages used to write the application. If multiple languages are used, give the percentages of each.

| Language | Percentage |
|---|---|
| | |
| | |
| | |

   b. Do you use PL/I or Ada? □ YES □ NO

   *If you use Ada, you must rewrite the code because no Ada compiler exists for x86-64. If possible, VSI recommends rewriting the code into a high-level language using documented system interfaces.*

   *PL/I is not yet supported on x86-64. If your application has code written in PL/I, please contact VSI to discuss your options.*

   *In general, if the compilers are not available on x86-64, you must translate or rewrite code in a different language. For information on availability of compilers and translators, see the VMS Software website.*

11. Is there application documentation? □ YES □ NO

    *If you answer YES, note that if any changes were required to the application, the documentation might have to be updated accordingly.*

# External Dependencies

Consider the system configuration, environment, and software required for developing, testing, and running your application.

12. What is the system configuration (CPUs, memory, disks) required to set up a development environment for the application? (This will help you plan for the resources needed for porting.)

    _____

    _____

    _____

13. What is the system configuration (CPUs, memory, disks) required to set up a typical user environment for the application, including installation verification procedures, regression tests, benchmarks, or work loads? (This will help you determine whether your entire environment is available on x86-64.)

    _____

    _____

    _____

14. Does the application rely on any special hardware? (This will help you determine whether the hardware is available on x86-64, and whether the application includes hardware-specific code.) □ YES □ NO

15. What version of OpenVMS does your application currently run on? _____

    *VSI recommends having your application running on the latest version for the architecture before porting it to x86-64.*

16. Does the application require layered and third-party products to run?

    a. From VSI, other than compiler RTLs: □ YES □ NO

    List the VSI layered products and versions:

| VSI Product | Version |
|---|---|
| | |
| | |
| | |

    b. From third parties: □ YES □ NO

List the third-party products and versions:

| Third-Party Product | Version |
|---|---|
| | |
| | |
| | |

*If you answer YES to a and are uncertain whether the VSI layered products are yet available for x86-64, check with an VSI sales representative. If you answer YES to b, check with your third-party product supplier.*

17. a. Do you have regression tests for the application? □ YES □ NO
    b. If YES, do they require any particular software product such as VSI Digital Test Manager? □ YES □ NO

*If you answer YES to a, you should consider porting those regression tests. If you answer YES to b, VSI Digital Test Manager (part of the DECset product set) is available with this release of OpenVMS. If you require other tools for testing purposes, contact your VSI support representative. VSI recommends a disciplined testing process to check regressions in product development.*

## Architectural Dependencies

Consider differences between processor architectures and versions of OpenVMS that your application runs on. The following questions will help you account for the most significant differences. Consider the changes that might be necessary to application components because of these differences. Any user-written code or assembly code that depends specifically on the old architecture must be rewritten.

18. Does the application use OpenVMS VAX floating-point data types? □ YES □ NO

*If you answer YES, note that the default for x86-64 compilers is IEEE floating data types. x86-64 compilers provide for VAX floating support by automatically converting from OpenVMS VAX to IEEE floating data types. Slight precision differences might result. In addition, run-time performance will be somewhat slower (compared to the direct use of IEEE floating data types without conversion), depending on the extent to which the application is dominated by floating point calculations.*

19. a. Does the application use multiple cooperating processes? □ YES □ NO
    b. If you answer YES, how many processes? _____
    c. What interprocess communication method is used?

□ Shared memory □ ICC □ Mailboxes
□ DLM □ PPL routines □ Other

d. If you use global sections ($CRMPSC) to share data with other processes, how is data access synchronized? (This will help you determine whether you will need to use explicit synchronization, and the level of effort required to guarantee synchronization among the parts of your application. Use of a high-level synchronization method generally allows you to port an application most easily.)

_____

_____

20. Does the application currently run in a multiprocessor (SMP) environment? □ YES □ NO

*If you answer YES, it is likely that your application already uses adequate interprocess synchronization methods.*

21. Does the application use AST (asynchronous system trap) mechanisms? □ YES □ NO

*If you answer YES, you should determine whether the AST and main process share access to data in process space. If so, you may need to explicitly synchronize such accesses.*

22. a. Does the application run in privileged mode or link against SYS$BASE_IMAGE? If YES, why? □ YES □ NO

b. Does the application depend on OpenVMS internal data structures or interfaces? □ YES □ NO

*Applications that link against the OpenVMS executive or run in privileged mode might require additional porting work. Undocumented interfaces in SYS$BASE_IMAGE might have changed on x86-64.*

*Applications that depend on OpenVMS internal data structure definitions (as defined in SYS$LIBRARY:LIB.INCLUDE, SYS$LIBRARY:LIB.L32, SYS$LIBRARY:LIB.MLB, SYS$LIBRARY:LIB.R64, SYS$LIBRARY:LIB.REQ, and SYS$LIBRARY:SYS$LIB_C.TLB) might also require additional porting work, as some internal data structures might have changed on x86-64.*

23. Does the application use connect-to-interrupt mechanisms? □ YES □ NO

If YES, with what functionality?

_____

_____

_____

*Connect-to-interrupt is not supported on x86-64 systems. Contact an VSI representative if you need this feature.*

24. Does the application create or modify machine instructions? □ YES □ NO

*Guaranteeing correct execution of instructions written to the instruction stream requires great care on x86-64. Any code dealing with specific machine instructions must be rewritten.*

25. Does the application include any other user-written code or assembler code that depends specifically on the architecture? □ YES □ NO

    *If you answer YES, rewrite such code. Optimally, rewrite it so that it is independent of architecture.*

26. Does the application include or depend on any conditionalized statements in source code or build files, or any code that includes logic that assumes it is running on VAX, OpenVMS Alpha, or I64 system? □ YES □ NO

    *If you answer YES, you might have to modify the conditional directives slightly. For more information, see section "Conditionalized Code" of the Porting Applications from VSI OpenVMS Alpha to VSI OpenVMS Industry Standard 64 for Integrity Servers Manual.*

27. What parts of the application are most sensitive to performance? I/O, floating point, memory, realtime (that is, interrupt latency, and so on).

    _____

    _____

    _____

    *This will help you determine how to prioritize work on the various parts of your application and allow VSI to plan performance enhancements that are most meaningful to customers.*

28. Does the application use any OpenVMS Alpha or I64 Calling Standard routines? □ YES □ NO

    *If you answer YES, note that applications that use some calling standard data structures and calling standard routines may have to be modified. For more information about OpenVMS calling standards, see the VSI OpenVMS Calling Standard.*

29. Does the application use non-standard routine linkage declarations? □ YES □ NO

    *If you answer YES, note that by definition applications with non-standard linkages know the OpenVMS Alpha or I64 calling standard and might need modification to conform to the x86-64 calling standard.*

30. a. Does the application depend on the format or content of an object file? □ YES □ NO
    b. Does the application depend on the format or content of an executable image? □ YES □ NO
    c. Does the application depend on the debug symbol table content of an image? □ YES □ NO

    *If you answer YES to a, b, or c, note that applications that depend on any of these image and object file data structures must be modified. The industry-standard object and image file layout called ELF has been adopted by x86-64. The industry-standard DWARF debug symbol table format has also been adopted. Applications with dependencies on OpenVMS Alpha's object, image or debug symbol table formats must be modified. For more information, see section "Reliance on Alpha Object File Format" of the Porting Applications from VSI OpenVMS Alpha to VSI OpenVMS Industry Standard 64 for Integrity Servers Manual.*

31. Does the application use the C *asm* statement? □ YES □ NO

32. Does the application use BLISS register Built-ins? □ YES □ NO

*If you answer YES, note that applications that use the BLISS BUILTIN statement to declare register built-ins must be modified.*

## Porting Your Application

Once the preliminary evaluation and planning stages are completed, these are the main porting tasks.

33. Have you upgraded your system to the latest version available? □ YES □ NO

*If you answer NO, first upgrade your system before beginning to port the application.*

34. a. Have you test compiled your application on the latest version for your current architecture, using the most recent version of the compiler? □ YES □ NO

b. Have you corrected any problems detected during the compile? □ YES □ NO

*If you answer NO to a, compile your application on the latest version of OpenVMS for your architecture, using the latest compiler available. If you answer YES to a but have rewritten some code since, compile again. After compiling, correct any problems detected. When you have finished this phase, and when you have responded to all other considerations detailed thus far in this checklist and in chapters "Migrating Source Modules" through "Other Considerations" of the Porting Applications from VSI OpenVMS Alpha to VSI OpenVMS Industry Standard 64 for Integrity Servers Manual, your application is ready for porting to x86-64.*

35. Have you copied the modules and related code to x86-64? □ YES □ NO

36. Compile, link, and run the application on the x86-64 system. Is your application compiled, linked, and running? □ YES □ NO

37. Perform unit testing on the x86-64 system. Have you completed the unit testing? □ YES □ NO

38. Perform system testing on the x86-64 system. Have you completed the system testing? □ YES □ NO

39. Perform regression testing on the x86-64 system. Have you completed the regression testing? □ YES □ NO

40. Did testing succeed? □ YES □ NO

*If you answer NO, resolve the problems, then recompile, relink, and retest the application.*