VSI Fortran X8.5-0004 for OpenVMS x86 Systems
Release Notes

15 June 2023

This document contains information about VSI Fortran X8.5-0004 including new and changed features, differences between X8.5-0004 and previous versions, corrections, and other topics. This file is of interest to both system managers and application programmers.

CONTENTS

CHAPTER 1 VSI FORTRAN X8.5-0004 FOR OPENVMS X86 SYSTEMS RELEASE NOTES

1.1 Overview 1-1
1.2 Getting Help And Reporting Problems 1-1

CHAPTER 2 NEW AND CHANGED FEATURES IN PREVIOUS RELEASES

2.1 New And Changed Features 2-1
2.2 Known Issues 2-1
2.3 Features Missing From Documentation 2-2
2.3.1 New /ALIGN=COMMONS=PAD_ALIGN_SIZE Qualifier . . 2-2
2.3.2 New /ASSUME=64BIT_STRING_PARAMS Qualifier . . . 2-2
2.3.3 IA64,_IA64_ , X86 And _X86_64_ Predefines Are Implemented. 2-3
2.4 Floating-Point Arithmetic 2-3

CHAPTER 3 VSI FORTRAN DOCUMENTATION AND ONLINE INFORMATION

CHAPTER 1

VSI FORTRAN X8.5-0004 FOR OPENVMS X86 SYSTEMS RELEASE NOTES

1.1 Overview

VSI Fortran X8.5-0004 for OpenVMS x86 Systems provides a compiler which runs on the OpenVMS x86 version 9.2 or later.

This chapter provides general information about the VSI Fortran product. The remaining chapters provide the following information:

- Highlights new and changed features in this version of VSI Fortran.
- Highlights new and changed features from previous versions of VSI Fortran.
- Details some known problems and incomplete functionality in this version of the compiler.
- Lists the organization of the documentation set.

The image identification for the Fortran compiler is F90 X8.5-0004

The FORTRAN/VERSION string is "VSI Fortran X8.5-0002 (GEM 50xxx) for X86 systems".

1.2 Getting Help And Reporting Problems

Please report problems or offer feedback using the VSI Support Portal.

You can also send comments, questions and suggestions about the VSI Fortran product to info@vmssoftware.com. Note that these addresses are for informational inquiries and is not a formal support channel.

CHAPTER 2

NEW AND CHANGED FEATURES IN PREVIOUS RELEASES

2.1 New And Changed Features

VSI Fortran X8.5-0004 is the initial release of Fortran on OpenVMS x86. It is based on VSI Fortran on OpenVMS I64 for source compatibility. There may be platform-specific features from OpenVMS Alpha and OpenVMS I64 that may not be supported.

2.2 Known Issues

- Some of the run-time overflow checking has not been implemented yet.
- Debug support is not fully implemented and the compiler may generate an assertion when using /DEBUG.
- VAX floating numbers will not print correctly on V9.2 systems. This problem will be corrected in a future release of OpenVMS.
- Quadruple precision floating point (REAL*16, REAL(KIND=16)) is currently not supported. This will be addressed in a future release of the compiler along with changes to various system libraries.
- The /SEPARATE_COMPILATION qualifier is currently ignored. A single Fortran source file is create a single object module.
- The /OPTIMIZE qualifier does not fully work. The compiler does contain some optimization, but still lacks additional support for pointer-alias analysis as well as support for the /OPT=TUNE keywords.

- The /MACHINE_CODE qualifier is ignored. As a workaround, you can use the ANALYZE/OBJECT/DISASSEMBLE to show the generated code and line numbers. For V9.2 systems, the output from ANALYZE/OBJECT/DISASSEMBLE shows source line numbers are are not very useful when looking at the compiler-generated listing file. The ANALYZE problem is corrected in the upcoming E9.2-1 field test kit. It will show listing line numbers to match the listing file output. In addition, The output from ANALYZE has line numbers but does not static data declarations.

2.3 Features Missing From Documentation

2.3.1 New /ALIGN=COMMONS=PAD_ALIGN_SIZE Qualifier

When the qualifier /ALIGN=COMMONS=MULTILANGUAGE is in effect, COMMON blocks within the compiled Fortran source are padded at the end of each such COMMON block in the same way that the C compiler would pad it, i.e. by a pad size based on the data types in the block.

HP Fortran 95 previously contained an error (fixed in V7.6) which caused the padding to be based on the alignment specified in a directive. This error would cause a COMMON block that was aligned on an 8K boundary to also be padded out to an 8K size, for example.

The new /ALIGN=COMMON=PAD_ALIGN_SIZE qualifier provides the previous padding behavior.

2.3.2 New /ASSUME=64BIT_STRING_PARAMS Qualifier

By default, when a string variable is passed as an argument, it is passed with a 32-bit descriptor. When a 64-bit descriptor is desired (for example, when calling a routine written in another language using 64-bit descriptors), you need to individually designate each string variable for which you would like the 64-bit descriptor.

Some customers have requested an option to allow all such string variables used within a source file to be passed or received with 64-bit descriptors without needing to so specify for each individual string variable.

The new [NO]64BIT_STRING_PARAMS keyword for the /ASSUME qualifier can be used to specify that requested behavior. When /ASSUME=64BIT_STRING_PARAMS is in effect, then all string variables used as parameters will be so used with 64-bit descriptors.

NOTE

This feature must be used with caution, as unexpected side effects may cause you problems. For example, because the Fortran Runtime Library expects to receive parameters with 32-bit addresses, statements that access the RTL either implicitly (e.g. print statements) or explicitly may fail when passed string parameters with 64-bit addresses.

The default is /ASSUME=NO64BIT_STRING_PARAMS.

2.3.3 IA64,_IA64_ , X86 And _X86_64_ Predefines Are Implemented.

2.4 Floating-Point Arithmetic

- IEEE is the default floating-point datatype (that is, the default is /FLOAT=IEEE_FLOAT)

  HP Fortran for OpenVMS Alpha Systems defaults to the VAX G_float floating-point format (/FLOAT=G_FLOAT). On OpenVMS I64 or OpenVMS x86-64 systems, however, there is no hardware support for floating-point representations other than IEEE. Instead, the compiler supports VAX floating-point formats by generating run-time code which converts operands to IEEE format, performs the needed arithmetic operations, and then converts the IEEE result back to the appropriate VAX format. Depending on the application, this may impose significant additional run-time overhead and some loss of accuracy compared to performing the operations in hardware on an Alpha.

  This software support for the VAX formats is an important functional compatibility requirement for certain applications that need to deal with on- disk binary floating-point data, but its use should be strongly discouraged.

  If at all possible, users should use /FLOAT=IEEE_FLOAT (the default) for the highest performance and accuracy.

  Note that the changed /FLOAT default will have implications for the use of /CONVERT=NATIVE (the default). This switch causes unformatted data to remain unconverted on input, on the assumption that it matches the prevailing floating-point datatype.

Files written from Fortran applications built with /FLOAT=G_FLOAT/CONVERT=NATIVE (the default) on Alpha can be read by Integrity server applications built with /FLOAT=G_FLOAT/CONVERT=NATIVE or /FLOAT=IEEE/CONVERT=VAXG.

- The /IEEE_MODE qualifier defaults to /IEEE_MODE=DENORM_RESULTS This differs from Alpha, where the default is /IEEE_MODE=FAST. Despite the name, /IEEE_MODE=FAST does not have a significant effect on run-time performance on Integrity or x86-64 servers (or on Alpha processors from EV6 onward).

- On Integrity or x86-64 servers, users will have to pick one compile-time /FLOAT value and one compile-time /IEEE_MODE value and stick with it for the whole of their application. This is because mixed-mode applications will not (in general) work on OpenVMS I64 or OpenVMS x86-64 systems as a result of architectural differences in the hardware. This is a change from OpenVMS Alpha systems, where mixed-mode applications work. In particular, per- routine/per-file/per-library settings of a mode will not work.

- Exception handlers will be entered with the floating- point mode in effect at the time the exception was raised, not the mode with which the handler was compiled.

CHAPTER 3

VSI FORTRAN DOCUMENTATION AND ONLINE INFORMATION

The VSI Fortran documentation set can be found online at https://docs.vmssoftware.com/