HP COBOL V3.0 - Release Notes

# June 25, 2010

File location:
OpenVMS: SYS$HELP:COBOL030_RELEASE_NOTES.PS
OpenVMS: SYS$HELP:COBOL030.RELEASE_NOTES

# Contents

# 1 Release Summary

These release notes contain important information about HP COBOL V3.0 including a short descriptive summary of features and sections on problems corrected, restrictions and known problems, recent enhancements, debugger notes, documentation notes, troubleshooting tips, and an HP COBOL compatibility matrix.

This release includes updated COBOL native compilers for OpenVMS I64.

A few of the problems reported against the V2.9-1453 were in fact bugs in the RTL. Those reported problems have been fixed. Please see Section 1.3 for details on the RTL problems resolved, and see Section 3 for more information about how to obtain the updated RTL, should you require the fixes included in it, as it is no longer shipped as part of the compiler kit.

The COBOL documentation is available on the web at:

http://h71000.www7.hp.com/commercial/cobol/

This documentation has been updated to cover V2.9 on OpenVMS I64. These Release Notes serve as the documentation update for V3.0 for OpenVMS I64.

These release notes should be made available to all users of HP COBOL.

The release notes include many details to help facilitate migrations from and coexistence between HP COBOL on the three supported platforms on OpenVMS. Additional details concerning differences between HP COBOL on the supported platforms appear in the documentation.

When a release note is specific to a particular platform, a platform tag is used.

V3.0 is currently shipping on OpenVMS I64 and supports the following OpenVMS versions at the time of the writing of these release notes:

- OpenVMS I64 V8.2-1 or higher

V3.0 is not supported for Tru64 Unix.

Refer to the latest operating system support policies for the currently supported versions of both COBOL and the operating systems.

HP COBOL V3.0 includes the following beyond the V2.9 releases on OpenVMS Alpha and OpenVMS I64:

- Performance enhancements in the Decimal support library, LIBOTS2.
- Bug fixes as detailed in Section 1.3.

Restrictions and problems specific to the OpenVMS platform are detailed in Section 1.4. This is the same set that can be found in the V2.8 release notes.

See the compatibility matrix at the end of the release notes for a summary of the functionality available across the platforms supported by HP COBOL currently and in previous versions.

## 1.1 Version Numbers

The edit level number, which appears at the end of the version number, reflects the number of revisions made to the compiler or RTL.

The version number of the compiler is printed at the top of each page of the source listing generated by the compiler. For example, if the following string appears in the top line:

```
HP COBOL V3.0-0001
```

then the compiler is Version V3.0, edit number 0001.

The compiler version number is also displayed as follows:

```
[OpenVMS] ANAL/IMAGE SYS$SYSTEM:COBOL
```

The version number of run-time library is V2.9-785 for OpenVMS I64.

To verify the version of the run-time library:

```
ANAL/IMAGE SYS$LIBRARY:DEC$COBRTL
```

The version number of decimal support run-time library is V2.8-71 for OpenVMS I64.

To verify the version of the decimal support run-time library:

```
ANAL/IMAGE SYS$LIBRARY:LIBOTS2
```

## 1.2 Installation information

HP COBOL on OpenVMS I64 and OpenVMS Alpha are each installed via a single PCSI kit which installs the COBOL compiler, DCL command, DCL help, and Release Notes.

The single PCSI kit for OpenVMS Alpha is now the standard. A separate PSCI kit for the RTL is no longer necessary because this version of the compiler is only supported on OpenVMS Alpha V7.3-2 or later and the run-time library is now installed during the OpenVMS installation.

To install a PCSI kit use the PRODUCT INSTALL command. You may type HELP PRODUCT to see a full range of options available for this command.

For more detailed installation instructions please see the *HP COBOL* Installation Guide which can be found in the same directory on the CDrom as the .PCSI kit.

For information on obtaining and installing DEC$COBRTL V2.9-782 independent of the OpenVMs installation please see Section 3: Run-Time Library Notes.

## 1.3 Problems Corrected Since V2.9-1453

For problems corrected through V2.9-1453 please refer the Release Notes for that compiler.

Problems corrected that do not apply to both Alpha and I64 platforms are tagged with

- [OpenVMS I64]
- [OpenVMS Alpha]

Problems corrected that apply only to the RTL (DEC$COBRTL.EXE) are tagged with

- [RTL]
- [RTL,OpenVMS I64]
- [RTL,OpenVMS Alpha]

Problems corrected that apply only to the Decimal support library, LIBOTS2 are tagged with

- [LIBOTS2]
- [LIBOTS2,OpenVMS Alpha]
- [LIBOTS2,OpenVMS I64]

| Version | Description |
|---|---|
| V3.0-0001 | Under certain circumstances, the compiler would emit error messages for COBOL programs with currency statement. This has been corrected. |
| V3.0-0001 | A performance issue related to compound ADD statement has been fixed.<br>Adding a single value to multiple variables by having multiple destination fields within the same ADD statement like for example, ADD 1 to CTRL1,CTRL2,CTRL3 slowed down the application considerably compared to using 3 individual ADD statements. |
| V3.0-0001 | A large COBOL program with lots of variables and MOV statements resulting in incorrect value at optimisation level 2 or higher has been fixed. |
| V3.0-0001 | A performance problem with the compiler generating alignment faults during compilation has been fixed. |

| Version | Description |
|---|---|
| V3.0-0001 | At times MOVE to adjacent fields in a record got optimised away incorrectly at higher optimisation levels. This has been fixed.<br>A very specific set of circumstances in the program caused a problem in the optimizer. |
| V2.9-785 | [RTL] A problem with the function ACCEPT FROM DAY-OF-WEEK returning two bytes instead of one byte has been fixed. |
| V2.9-785 | [RTL I64] A performance problem with the routine DCOB$CALLED has been fixed.<br>There was performance degradation with the COBOL CALL statement on IA64 and this has been fixed. |
| V2.9-785 | [RTL] When LNM$FILE_DEV was modified to point first to a table belonging to LNM$SYSTEM_DIRECTORY, COB$SWITCHES remained defined even after image exit. This issue has been fixed. |
| V2.8-71 | [LIBOTS2,OpenVMS I64] The COBOL Packed Decimal support routines have been recoded for performance benefits.<br>The COBOL Packed Decimal support routines located in LIBOTS2.EXE have very poor performance on OpenVMS I64 as compared to their OpenVMS Alpha couterparts. A subset of the routines have been recoded using the high performance algorithms from the Alpha versions. Some individual routines saw performance increases of over 75%.<br>COBOL applications do not need to be recompiled or relinked to see this improvement. |

## 1.4 Restrictions and Known Problems

Restrictions or known problems in the compiler that do not apply to both Alpha and I64 platforms are tagged with

- [OpenVMS I64]
- [OpenVMS Alpha]

Restrictions or known problems that apply only to the RTL (DEC$COBRTL.EXE) are tagged with

- [RTL]
- [RTL,OpenVMS I64]
- [RTL,OpenVMS Alpha]

| Version | Description |
|---|---|
| V3.0-0001 | [OpenVMS I64] The COBOL compiler requires OpenVMS I64 V8.2-1 or higher. |
| V3.0-0001 | [OpenVMS Alpha] The COBOL compiler requires OpenVMS Alpha V7.3-2 or higher. |
| V3.0-0001 | [OpenVMS I64] Under certain circumstances, the COBOL compiler generates code that produces incorrect results when adding 1 to a decimal number. |
| V3.0-0001 | [OpenVMS I64] Writing object files on Integrity servers can be very slow when compared to Alpha if the program has large static data that has lots of zeros but some initialization. On Alpha, the object file format has the ability to describe those large sections of zeros in a compressed form. However, ELF does not have that ability and hence all the leading zeros upto the non-zero initialization have to be written. ELF does allow omitting the trailing zeros but that is all. |
| V3.0-0001 | [OpenVMS I64] Under certain circumstances, the cobol compiler generates instuctions of the form<br>ld8 r36 = [r36], #8<br>when compiled with /optimized qualifier. Such instructions will generate the fault SYSTEM-F-OPCDEC at runtime. |

| Version | Description |
|---|---|
| V3.0-0001 | [OpenVMS I64] OpenVMS I64 V8.2-1 supports up to 5 entry points per routine. COBOL currently uses an entry point for each<br>• PROGRAM-ID<br>• USE procedure<br>• REPORT WRITER procedure<br>This OpenVMS I64 restriction impacts all versions of COBOL.<br>Routines with 6 or more entry points will show<br>• Incorrect exception handling behavior (for example, INTDIV instead of DIVBY_ZER when the program does a divide by 0)<br>• %DEBUG-E-LASTCHANCE, stack exception handlers lost, re-initializing stack - on debug startup<br>A new set of [SYS$LDR]EXCEPTION*.* files is required to fix this problem. This OpenVMS I64 fix is available with a remedial kit for OpenVMS I64 V8.2-1. |
| V3.0-0001 | [OpenVMS I64] Alpha values for the /ARCHITECTURE and /OPTIMIZE=TUNE qualifiers are accepted on the compiler invocation command. An informational is printed saying they are ignored. I64 values for /ARCHITECTURE and /OPTIMIZE=TUNE are defined in the OpenVMS I64 compiler for development purposes only. Their behavior is unpredictable and they should not be used. |
| V3.0-0001 | [OpenVMS I64] Arithmetic exception handling is not 100% compatible between I64 and Alpha in undefined situations such as when ON SIZE ERROR is not used. In particular, if the size error condition is raised, the undefined result value on I64 may differ from the undefined results value on Alpha or VAX. Also, the size error condition is not raised in all cases of 0/0. To accomplish COBOL semantics in the areas of exception handling and rounding, the COBOL RTL makes calls to two SYS$IEEE routines. COBOL does not support /IEEE_MODE and /CHECK=UNDERFLOW. |
| V3.0-0001 | [OpenVMS I64] Floating point defaults to /FLOAT=IEEE_FLOAT. Since all floating point operations are done using IEEE floating, results with non-IEEE floating point datatypes are subject to these conversions and precision of IEEE operations and thus are not 100% compatible between OpenVMS I64 and OpenVMS Alpha. |
| V3.0-0001 | [OpenVMS I64] Tape support with OPEN EXTEND is not fully working. |
| V3.0-0001 | [OpenVMS I64] COBOL images translated from OpenVMS Alpha which use Oracle DBMS require that STOP RUN (not EXIT PROGRAM) be used in the main program of a run unit if you want to stop program execution. |

| Version | Description |
|---|---|
| V3.0-0001 | [OpenVMS I64] COBOL images translated from OpenVMS VAX and OpenVMS Alpha have other restrictions as detailed in the binary translator release notes. |
| V3.0-0001 | To obtain all relevant V3.0 bug fixes and to use all V3.0 functionality, you must use the latest HP COBOL RTL which is V2.9-782. This RTL will be available on a release of OpenVMS after V8.3 for both Alpha and I64.<br>To obtain the RTL containing the fixes described in section 1.3 please contact your regular HP support channel to obtain an ECO kit with these fixes in them for any OpenVMS Alpha V7.3-2 or higher or OpenVMS I64 V8.2-1 or higher. |
| V3.0-0001 | [OpenVMS Alpha] The RTL can acquire multiple record locks with automatic record locking in certain cases with START and WRITE. You can see this situation if you get an unexpected<br>`%COB-F-LOCKED_FAILED`<br>diagnostic. If you encounter this situation, you can clear the record locks using either UNLOCK ALL RECORDS or CLOSE followed by OPEN. |
| V3.0-0001 | If a file is remote, results on Alpha/I64 match results on VAX with manual record locking only with OPEN ALLOWING NO OTHERS. Other OPEN ALLOWING options on Alpha/I64 result in the diagnostic at run-time with REWRITE. |
| V3.0-0001 | The compiler with /ANALYSIS_DATA writes a PROGRAM-ID to the .ANA file with hyphen (-) if the PROGRAM-ID contains underscore (_). |
| V2.8-1286 | For certain floating-point operations that raise floating exceptions (traps, faults, or signals) not all processors leave the same result in output registers. |
| V3.0-0001 | When the starting position or length of reference modification is specified by an expression which is out of range, HP COBOL on Alpha and I64 detects the out of range at run-time, if /CHECK=BOUNDS (-check bounds) is used. HP COBOL on VAX in some cases detects the out of range at compile-time. HP COBOL on Alpha and I64 issues compile-time diagnostics for constant expressions in many cases. |
| V3.0-0001 | External names with the following prefixes are reserved for use by HP COBOL: COB$, DCOB$, cob_. |

| Version | Description |
|---|---|
| V3.0-0001 | Because of the differences in the way expressions are evaluated in a COMPUTE between HP COBOL for OpenVMS VAX and HP COBOL on Alpha and HP COBOL on I64, the number of divide by 0 and undefined exponentiation diagnostics produced may vary. |
| V3.0-0001 | The scope of EXTERNAL and DISPLAY UPON environment variables or OpenVMS logicals is the current run unit. These language features do not preserve data between run units. |
| V3.0-0001 | If a nonnumeric literal is continued across a line break, converting the file from terminal format to ANSI format using the reformat utility causes spaces to be inserted in the literal at the point of the line break. This can result in a literal that is longer than the PIC clause, causing a syntax error. |
| V3.0-0001 | See the section "Debugger Notes" for problems and restrictions specific to debugging with HP COBOL. |
| V3.0-0001 | SORT/MERGE does not support octaword binary keys on any platform except in the case on OpenVMS Alpha and I64 where SYS$LIBRARY:SORTSHR.EXE is used. SORTSHR.EXE is the default on OpenVMS Alpha and OpenVMS I64. |
| V3.0-0001 | The order of evaluation of subscripts for the implied statements created for corresponding operands for ADD, MOVE, and SUBTRACT with the CORRESPONDING phrase is not defined under the ANSI-85 standard. Some implementations such as HP COBOL-85 for NSK (Tandem COBOL) differ from HP COBOL on I64, Alpha, and VAX and instead evaluate all subscripts only once, at the start of the execution of the actual ADD, MOVE, or SUBTRACT statement. See the Reference Manual for the details on how HP COBOL on I64, Alpha, and VAX handles this situation. |
| V3.0-0001 | There are several known restrictions with SORT32 and Hypersort on OpenVMS Alpha/I64. The restrictions are documented in the OpenVMS V7.3, V7.3-1, V7.3-2, and V8.2-1 Release Notes. Some of the restrictions have been removed with the latest versions of SORT32 and Hypersort which are now shipping with<br>• OpenVMS Alpha V7.3-2<br>• OpenVMS I64 V8.2-1<br>If you use SORT or MERGE with COBOL, these restrictions may impact your programs. |

| Version | Description |
|---|---|
| V3.0-0001 | [OpenVMS Alpha] Because of the addition of Extended File Support in OpenVMS Alpha V7.2, you may notice changes in the handling of I/O run-time diagnostics and RMS special registers on OpenVMS Alpha V7.2 and higher. In particular, a long filename which produced RMS$_FNM under versions of OpenVMS Alpha prior to V7.2 now produces RMS$_CRE on OpenVMS Alpha V7.2 and higher, and this difference is reflected in I/O run-time diagnostics and RMS special registers. You do not need to be using the new ODS-5 support to see these RMS differences. |
| V3.0-0001 | In the Debugger, use SET BREAK %LABEL with numeric paragraph labels. On VAX, you instead use SET BREAK %NAME. |
| V3.0-0001 | [OpenVMS Alpha] The minimum Oracle CDD/Repository version required for HP COBOL for OpenVMS Alpha is CDD/Repository V5.3. Contact Oracle for the details on the current supported version of Oracle CDD/Repository. |
| V3.0-0001 | [OpenVMS Alpha] The minimum Oracle DBMS version required for HP COBOL is Oracle DBMS V6.1-2. Use of the HP COBOL DBMS support requires the separate COBOL-DBMS license/PAK; see the HP COBOL SPD for more information. Contact Oracle for the details on the current supported version of Oracle DBMS. |
| V3.0-0001 | APPLY LOCK-HOLDING (manual record locking) is not supported for REWRITE on a primary key WITH DUPLICATES if the file is on a remote system and accessed via DECnet. Use automatic record locking instead in this case. |
| V3.0-0001 | Stream_LF files (ORGANIZATION LINE SEQ or sequential VFC files with /NOVFC) are not supported for file-sharing (ALLOWING ALL or READERS) when opened for writing. |
| V3.0-0001 | Stream_LF files (ORGANIZATION LINE SEQUENTIAL or ORGANIZATION SEQUENTIAL with /NOVFC) are not supported if the file is assigned to a network device. |
| V3.0-0001 | If you mix ANSI ACCEPT statements and extended ACCEPT statements in one program, the RECALL (UP-arrow or DOWN-arrow) will not be supported for the ANSI ACCEPT statements. |

| Version | Description |
|---|---|
| V3.0-0001 | [OpenVMS Alpha] Attempts to write before the end of data (OPEN EXTEND) when using a TZK10 drive (a quarter-inch cartridge (QIC) standard drive) result in a write append error. This is a restriction in the QIC standard. The QIC standard allows drives to write at the end of data (append) or write the entire tape from beginning of tape (BOT), but does not allow writing before the end of data.<br><br>This restriction applies to all OpenVMS Alpha hardware platforms that support the TZK10 drive. |
| V3.0-0001 | There are known problems with CDD/Repository regarding translation of quadword INITIAL_VALUEs. Some values are not converted correctly and will not produce a COBOL VALUE clause or a diagnostic message.<br><br>Specifically, quadword fields with initial values that cannot fit into a quadword are silently ignored. This occurs in one of the translation layers of CDD and is not reported to the compiler.<br><br>Initial values that are 19 digits in length and that will fit into a quadword, but exceed the length of a COBOL literal are detected and flagged by the compiler with a diagnostic.<br><br>Initial values that are 18 digits or less work correctly.<br><br>The initial value clauses for all unsigned quadwords are ignored by the compiler. Again, this occurs in a CDD translation layer and cannot be detected by the compiler. This occurs with the HP COBOL for OpenVMS VAX compiler as well as the HP COBOL compiler. |

The table above is a summary of the restrictions and known problems with using HP COBOL on OpenVMS Alpha and OpenVMS I64. For other restrictions in using the product, see

- *HP COBOL Reference Manual*
- *HP COBOL User Manual*

# 2 Run-Time Library Notes

DEC$COBRTL V2.9-782 will automatically ship with a version of OpenVMS Alpha and OpenVMS I64 after V8.3.

The bug fixes associated with the HP COBOL V2.9-782 Run-Time Library (DEC$COBRTL.EXE), are not required by this release of the compiler. If you should require any of the fixes described in section 1.3 prior to the release of DEC$COBRTL V2.9-782, please contact your regular HP support channel to obtain a DEC$COBRTL V2.9-782 ECO kit for your current version of the operating system.

Please note a DEC$COBRTL ECO kit cannot be built for any version of OpenVMS Alpha prior to V7.3-2 or OpenVMS I64 V8.2-1.

# 3 Debugger Notes

## 3.1 Debugger Notes for OpenVMS Alpha and OpenVMS I64

Please read the following release notes for information about COBOL language support in OpenVMS Alpha/I64 DEBUG. Also see the debugger release notes included with the OpenVMS Alpha/I64 documentation for information on corrections for some of the problems listed below. Also see the appendix on tools in the *HP COBOL User Manual*.

Following are some restrictions with this debugger:

- COBOL variables with edit strings are not fully supported in DEBUG.
- When debugging, ambiguous paragraph names cannot be disambiguated through qualification by the section name. To set breakpoints on paragraph names, the paragraph names need to be unique. Alternatively, breakpoints can be set using line number notation.
- [OpenVMS I64] DEBUG support on OpenVMS I64 for OCCURS DEPENDING ON still uses the maximum upper array bound in some cases.
- Setting a watch point on a variable in a cobol program, sometimes causes the following information to be generated:

  %DEBUG-I-ASYNCSSWAT, possible asynchronous system service and static watchpoint collision This happens even though the program itself does not call any asynchronous system service, but calls a COBOL RTL call, which in turn could be calling an asynchronous system service. The above is only an informational message and the user can continue. The workaround to not seeing this message is to set a non-static watch point on the variable; however this can slowdown the program as it traces every instruction.
- If you use the debugger to deposit data into Edited Picture variables, an error message will be displayed and the deposit operation will not be completed. The debugger session will not be compromised, however, and may be continued after this error. The error message is

  %DEBUG-F-BUG_CHECK, internal consistency check failure
- The debugger divide operators for display numeric and packed decimal data items are not implemented.
- The debugger generally does not support numerics with more than 18 digits.
- In previous versions of the debugger, lookups of certain COBOL record components resulted in nonunique symbol errors. This has been corrected starting with the debugger which shipped with OpenVMS Alpha V7.3.
- In previous versions of the debugger, DEBUG STEP would take extra steps when INITIALIZE was used. This has been corrected starting with the debugger which shipped with OpenVMS Alpha V7.2.

- Starting with OpenVMS Alpha V7.3-1, attempts to STEP after the program has reached completion can result in the DEBUG diagnostic DYNLNGSET AMACRO. You can then use SET LANGUAGE COBOL if needed.
- When using the debugger, you may notice that variable names that contain underscores and hyphens may have been changed, under certain circumstances. These problems have been corrected in the OpenVMS Alpha V7.1 debugger.

  Specifically, variable names that are "local" to your COBOL program (that is, names that are not visible to other separately compiled program modules) have any underscores changed to hyphens when reported to the debugger. Thus, the data item "A_B" can only be referenced as "A-B" in a debugging session.

  Names that are visible to other separately compiled program modules (for example, EXTERNAL items, PROGRAM-ID names, etc.), have any hyphens in their names changed to underscores when reported to the debugger. Thus, the PROGRAM-ID name "SUBR-CALL" can only be referenced as "SUBR_CALL" in a debugging session.

## 4 Documentation Notes

### 4.1 Documentation Changes and Updates

The *HP COBOL Installation Guide* has been updated for this release. There have been no changes or updates to the *HP COBOL Reference Manual*, *HP COBOL User Manual*, or *HP COBOL DBMS Database Programming* manual. Information contained in these Release Notes is to be considered supplemental to the currently published documentation.

### 4.2 Summary of the HP COBOL Documentation Set

- *HP COBOL Reference Manual*

  Describes the source language environment for the HP COBOL programmer. Defines the format and use of statements in the HP COBOL language. This document highlights extensions to the 1985 ANSI COBOL Standard by color.

- *HP COBOL User Manual*

  Describes the development and run-time environment for the HP COBOL programmer working on OpenVMS or Tru64 UNIX.

  Describes the commands to compile, link, run, and debug COBOL programs.

  It also explains HP COBOL I/O, the language interface with the calling standard, error handling and run-time messages, using structures and records, writing a condition handler, native data characteristics, using nonnative numeric formats, compatibility issues with HP COBOL for OpenVMS VAX, and using CDD/Repository, LSE, SCA, FUSE, and the debuggers.

- HP COBOL Online Help file

  The online Help file presents information on the HP COBOL command qualifiers and options.

  [OpenVMS] Presents the COBOL general formats and describes the data items. Describes a subset of the compiler messages and the complete set of run-time messages.

- *HP COBOL Installation Guide*

  Explains how to install HP COBOL V2.9 and subsequent point releases on OpenVMS Alpha and OpenVMS I64, including registering a license PAK (product authorization key), disk space, and other prerequisites.

  This installation guide is provided in PostScript form (.PS file type) and in plain ASCII form (.TXT file type) on the Layered Products Media CD.

- *HP COBOL DBMS Database Programming*

  Describes how to use language elements specific to Oracle DBMS database programming on OpenVMS.

This book is an adjunct to the HP COBOL basic doc set. We have collected the elements specific to Oracle DBMS database programming, and concentrated them in this optional book for the convenience of programmers who write COBOL programs that access Oracle DBMS databases.

Documentation for HP COBOL is on the Online Documentation Library (ODL) in .pdf format.

## 4.3 Feedback on Our Documentation

HP welcomes your comments on our documentation. Please send comments to either of the following addresses:

Internet: *openvmsdoc@hp.com*

## 5 Troubleshooting Tips

If your program is not working as expected, here are some things to do:

- [OpenVMS Alpha] If you get

  > %LINK-I-UDFSYM, DCOB$IMAGE_INIT this may mean that you have incorrectly attempted to install a COBOL RTL kit on OpenVMS Alpha V7.3-2 or higher. COBOL RTL kits are needed for versions of OpenVMS Alpha prior to V7.3-2. If you attempt to install a COBOL RTL kit on OpenVMS Alpha V7.3-2, you may corrupt SYS$LIBRARY:STARLET.OLB. To correct this corruption, you must get the module COB_IMAGE_INIT from the SYS$LIBRARY:STARLET.OLB for you OpenVMS distribution, and reinsert that module in your current SYS$LIBRARY:STARLET.OLB.

- If you have HP COBOL for OpenVMS VAX, for the purpose of comparison try compiling with /STANDARD=OPENVMS_AXP to determine if any diagnostics are issued indicating you are using features on VAX that are not supported in HP COBOL for OpenVMS Alpha and OpenVMS I64.
- Try /NOOPT (-O0). /OPT (-O) and /NOOPT (-O0) are intended to produce the same results (probably with different compile-time and run-time performance), but you may be able to bypass a problem by compiling /NOOPT (-O0). Please report any differences you find with any level of optimization.
- Try /CHECK (-check). The various /CHECK (-check) options can help you find some coding errors at run-time.
- Try /WARNINGS=ALL (-warn all). The various warnings might help identify possible areas of the program to be investigated as potential sources of a problem.

  HP COBOL includes an informational diagnostic to help identify arithmetic statements that may get different results from other COBOL compilers. The INTERMED diagnostic has the text:

  > Possible truncation due to use of intermediate data item This indicates an expression for which HP COBOL introduces a floating point or COBOL Intermediate Temp intermediate data type. This diagnostic identifies places where binary or decimal floating-point arithmetic is being used instead of fixed-point arithmetic for intermediate computations. This can lead to unexpected final results and possibly results different from HP COBOL for OpenVMS VAX. A statement that triggers this diagnostic should be examined. One option is to compile with

  - COBOL /MATH_INTERMEDIATE=CIT4 <

  Another option is to rewrite the expression using two or more simpler statements. If you know the intended ranges of values for the input operands, you can specify the precision of intermediates, resulting in expression evaluation that can be accomplished using fixed-point arithmetic.

- HP COBOL includes an informational diagnostic when it detects situations when high order numeric truncation is possible. See details on /TRUNCATE (-truncate) for one way to control some of the situations when this diagnostic is issued.
- If the program is using a file created on a VAX and you compiled the program on Alpha or I64 using /ALIGNMENT or /ALIGNMENT=PADDING, try /NOALIGNMENT (the default for HP COBOL).
- There are two diagnostic tools to help acquire information on how to proceed with correcting alignment mismatches between an HP COBOL program and another program when they are sharing a record structure such as a file record structure:

  — /NOALIGNMENT/WARNINGS=ALL

  — /MAP=DECLARED/LIST

  With /NOALIGNMENT/WARNINGS=ALL, you will receive warnings for all data whose allocation will change if /ALIGNMENT is used. This information can be used with the listing produced with /MAP=DECLARED/LIST to determine which records might need to be differently aligned to be compatible with other programs that will be sharing files based on these records.

  There are two options for utilizing Alpha Natural Alignment and Padding for record formats:

  — /ALIGNMENT[=PADDING]

  — Alignment directives

  The alignment directives allow you to set different alignment options for different records within the program's source file.
- If you are having trouble reading a SEQUENTIAL or RELATIVE file created on OpenVMS, try moving the file to Tru64 UNIX using FTP binary mode or ZIP. In addition, for RELATIVE files, you must specify no compression using the -0 option.

  If the file on OpenVMS is in VFC format (for example, a file created with EXTERNAL or Report Writer or LINAGE on OpenVMS), you will need to skip over the VFC bytes when you read the file on Tru64 UNIX. Another option with HP COBOL on OpenVMS Alpha and OpenVMS I64 is to suppress the use of VFC with /NOVFC.
- For Indexed files on OpenVMS, a key defined in the file but not the program is not considered a mismatch, and /CHECK=DUPLICATE_KEYS only considers this a mismatch if the key is declared both in the file and the program, and differs with respect to whether duplicates are allowed. For HP COBOL on Tru64 UNIX without -relax_key_checking, a missing key is always considered a mismatch, and with -relax_key_checking, it is not considered a mismatch for the keys to differ with respect to duplicates. In other words, to get consistent behavior with respect to duplicate key checking for all four supported platforms, specify all the keys in both the file and the program.

- If the program is attempting to read multiple files from a tape, use OPEN WITH NOREWIND to get consistent results between HP COBOL for OpenVMS VAX and HP COBOL for Alpha and I64.
- If the program is compiled /STANDARD=V3 (-std v3), make sure that the program does not depend on the HP COBOL for OpenVMS VAX /STANDARD=V3 features that are not supported by HP COBOL on Alpha and I64. These features can result in different behavior in the following situations:
  - When subscripts are evaluated in STRING, UNSTRING, and INSPECT (Format 3) statements and the REMAINDER phrase of the DIVIDE statement
  - When reference modification is evaluated in STRING, UNSTRING, and INSPECT (Format 3) statements
  - When the variable associated with the VARYING phrase is augmented in PERFORM... VARYING... AFTER statements (Format 4)
  - How PIC P digits are interpreted in some moves
  - When the size of variable-length tables is determined

  /WARNINGS=ALL can help you determine the effects of /STANDARD=V3. For full information on the HP COBOL for OpenVMS VAX implementation of /STANDARD=V3, see the appendix on qualifiers in the *VAX COBOL User Manual*.

  In addition, FILE STATUS 02 with /STANDARD=V3 is handled differently between HP COBOL for OpenVMS VAX and HP COBOL. With HP COBOL for OpenVMS VAX, if a record with a duplicate alternate key is written, FILE STATUS 02 is set and the DECLARATIVES procedure executes. With HP COBOL, FILE STATUS 02 is set, but the DECLARATIVES procedure does not execute.
- If the program is calling or called by another module (whether the other module is written in COBOL or another language), make sure the other module is compiled using compatible alignment and floating point types for any arguments passed in the calls.

  For example, if a C module is compiled with the C equivalent of /NOALIGNMENT (the HP COBOL default) that is, "#pragma nomember_alignment", then an HP COBOL module that calls or is called by that C module should be compiled /NOALIGNMENT. The HP C default is MEMBER_ALIGNMENT. For compatibility with HP C's "#pragma member_alignment", use /ALIGNMENT=PADDING or *DC SET PADALIGN.
- COBOL, like the other GEM based languages, sets the granularity option as specified on the command line and passes that to GEM. GEM generates the code and issues the diagnostics.

  Byte and word granularity are not supported with DISPLAY or PACKED-DECIMAL numerics. For other data types, you must have a version of LIBOTS on Alpha or I64 which supports byte and word granularity.

- If you encounter the NAMCLASS diagnostic on compile on Alpha or I64, check uses of COPY. You can often avoid NAMCLASS diagnostics on Alpha and I64 if you enclose COPY names in quotes. In other cases, you will need to rename to avoid the NAMCLASS diagnostic.
- If you are using READ NEXT and READ PREVIOUS / READ PRIOR, make sure you use START when you shift between READ NEXT and READ PREVIOUS / READ PRIOR. The COBOL program results are undefined if you immediately follow a READ NEXT by a READ PREVIOUS / READ PRIOR or vice versa.
- If you are using CALL dataname, you must explicitly include in the link of the run unit all modules which might be referenced by CALL dataname. CALL dataname is resolved at run-time, so there is no link-time name resolution for potential values for the user item used in CALL dataname.
- [OpenVMS Alpha/I64] If you encounter the UNALIGNFIX diagnostic on LINK on OpenVMS Alpha or I64, you can usually avoid this LINK diagnostic by compiling /ALIGN or otherwise ensuring that references from the COBOL program to entry points in shareable images are properly aligned.
- [OpenVMS Alpha/I64] If you encounter problems compiling

  /ANALYSIS_DATA/SEPARATE_COMPILATION instead compile

  /ANALYSIS_DATA/NOSEPARATE_COMPILATION
- If you encounter a problem possibly related to reserved words, use

  /RESERVED_WORDS=NOXOPEN This allows use of the X/Open reserved words as though they were not reserved words.
- When you compile a program with debugging also specify /NOOPTIMIZE to expedite your debugging session. Optimization often changes the order of execution of statements in a program, and it may keep values in registers and deallocate user variables. These effects can be confusing when you use the debugger. A diagnostic message warns you if you compile an HP COBOL program /DEBUG without specifying anything about optimization on the command line.
- If you are running out of virtual memory during compilation, try compiling /NOOPT and increasing virtual memory using the guidelines in the User Manual. Not all programs can be compiled with all optimization levels for a given virtual memory configuration. Some programs may not even compile /NOOPT. In those cases, you will need to reduce the program size per module by reducing the number of executable statements in each module and/or by reducing the size of data structures in each module.
- With extended ACCEPT/DISPLAY, HP COBOL for OpenVMS VAX and HP COBOL for Alpha and I64 may use different escape sequences to update the screen. Also, the screen update behavior is not identical between the products.

- If you attempt to use extended ACCEPT/DISPLAY with input redirected from a file and/or output redirected to a file, the operation between HP COBOL for OpenVMS VAX and HP COBOL on Alpha and I64 is not identical. In general, the HP COBOL RTL attempts to use ANSI ACCEPT/DISPLAY to handle all ACCEPT/DISPLAYs in this situation. For example, if you use DISPLAY AT LINE or ACCEPT DEFAULT, the HP COBOL RTL will ignore the extensions (that is, LINE or DEFAULT) if you redirect output to a file and/or input from a file.
- There are various things to consider related to floating point (COMP-1, COMP-2) support. HP COBOL supports IEEE floating point on all Alpha and I64 platforms. HP COBOL for OpenVMS Alpha supports F, D, and G floating. HP COBOL for OpenVMS I64 supports F, D, and G floating via conversion to and from IEEE floating, so results with F, D, and G floating on I64 are subject to the limitations and precision of these conversions to and from IEEE floating. HP COBOL for OpenVMS VAX supports VAX F and VAX D floating. VAX D is not equivalent to the D float available on Alpha.

  Floating point results should not be tested for exact equality or inequality; many decimal values containing fractional digits cannot be represented exactly in binary floating point. Instead of testing that the value exactly equals (or does not equal) the expected result, test that the value is within some small range of the expected result. COBOL's algorithm for converting a floating value to another datatype is not 100% compatible between VAX and Alpha/I64.
- Invalid decimal data checking with /CHECK=DECIMAL in HP COBOL is not 100% compatible between Alpha and I64 and is not 100% compatible with HP COBOL for OpenVMS VAX's use of the VAX instructions to do invalid decimal data checking. In particular, no platform currently detects all cases of invalid decimal data. Results with invalid decimal data are undefined on all supported HP COBOL platforms independent of whether or not the debugging aids /CHECK=DECIMAL or the VAX hardware actually are able to detect the invalid decimal data.
- [OpenVMS Alpha/I64] If you get the following diagnostic at run-time

  > %LIB-F-INVCLADTY, invalid class data type combination in descriptor  it may indicate you are using a LIB$ routine such as LIB$CVT_DX_DX which does not fully support unsigned COMP data types. With CALL USING BY DESCRIPTOR, HP COBOL for OpenVMS VAX uses a signed data type in the descriptor for unsigned COMP data types. HP COBOL for OpenVMS Alpha and I64 uses an unsigned data type in the descriptor for unsigned COMP data types. With routines such a LIB$CVT_DX_DX which do not fully support unsigned COMP data types, you should use signed COMP data types for any calls to these routines.
- [OpenVMS Alpha/I64] If you compile /NOVFC, make sure to take into account the initial blank lines in the output files for any post-processing of these output files.

- [OpenVMS Alpha] If the LINK of a COBOL program fails with any DCOB$ or OTS$ undefined symbols, it is possible that the RTL has not been properly installed.
- [OpenVMS Alpha/I64] If you encounter problems using a comma list during compile, compile each module separately to reduce compilation resources and the chance for a compiler failure.
- [OpenVMS Alpha/I64] If the program uses floating point data from a file created on an OpenVMS VAX system with HP COBOL for OpenVMS VAX, try /FLOAT=D_FLOAT.
- [OpenVMS Alpha/I64] If a file is accessed that contains G-float data, use /FLOAT=G_FLOAT; if the file contains IEEE-float data, use /FLOAT=IEEE_FLOAT.
- [OpenVMS Alpha/I64] If you are using C or C++ on OpenVMS Alpha or I64, make sure to review the defaults for /FLOAT and choose a /FLOAT option to match how you compile COBOL modules which interact with the floating data in the C/C++ modules.

  The default for C on OpenVMS Alpha is /FLOAT=G_FLOAT, while the default for HP COBOL on OpenVMS Alpha is /FLOAT=D_FLOAT. Other languages such as HP Pascal also have /FLOAT=G_FLOAT as the default on OpenVMS Alpha. On OpenVMS I64, the default is /FLOAT=IEEE_FLOAT.
- [OpenVMS Alpha/I64] If you are using /NATIONALITY=JAPAN, and you get the following at run-time

  %COB-F-BUG_CHECK, internal consistency check failure make sure you have the RTL with a version number no less than V2.8.
- [OpenVMS Alpha/I64] If your application uses any of the following character attribute clauses with ACCEPT or DISPLAY statements:

  BOLD or HIGHLIGHT
  BLINKING or BLINK
  UNDERLINED or UNDERLINE
  REVERSED or REVERSE-VIDEO you may find that the screen scrolls unexpectedly when the terminal is in wrap mode and HP COBOL displays data in the lower right corner of the screen.

  This behavior occurs when the data being displayed fits on the bottom line, but the control sequences which turn off the character attributes (and which HP COBOL appends to the data), extend past the end of the line. The unexpected scrolling is caused by the OpenVMS Alpha/I64 terminal driver because it interprets the control sequences as normal data and performs wrapping when it reaches the end of the line.

  Screen scrolling may also occur if your application displays escape sequences when the cursor is in the lower right corner of the screen. Setting the terminal to nowrap mode will solve this problem, or you can adjust the cursor's position prior to displaying the escape sequence.

  One solution is to set the terminal to nowrap mode using

  ```
  $ SET TERMINAL/NOWRAP
  ```

If this is not an acceptable solution, adjust the placement and/or size of the item being accepted or displayed.

# 6 HP COBOL Compatibility Matrix

Legend x = Available today N = Not supported P = Partial support

| | OpenVMS VAX | OpenVMS Alpha | OpenVMS I64 | Tru64 UNIX |
|---|---|---|---|---|
| | V5.7 | V2.9 | V3.0 | V2.8 |
| ANSI-85/-89 HIGH | x | x | x | x |
| ANSI-85 REPORT WRITER | x | x | x | x |
| Standard arithmetic | N | P | P | P |
| Table sort | N | x | x | x |
| Tape handling | x | x | P | x |
| Segmented keys | x | x | x | x |
| /CHECK=DUPLICATE_KEYS | x | x | x | N |
| RMS special registers | x | x | x | N |
| RMS APPLY extensions | x | x | x | N |
| /STANDARD=V3 | x | P | P | P |
| ANSI-74 FILE STATUS support | x | x | x | x |
| Print control files with VFC | x | x | x | N |
| Print control files without VFC | N | x | x | x |
| ISAM READ PRIOR/START LESS | N | x | x | x |
| Extended ACCEPT/DISPLAY | x | x | x | x |
| ACCEPT/DISPLAY WITH CONVERSION | x | x | x | x |
| ACCEPT support for 4-digit years | x | x | x | x |
| Y2K intrinsic functions | x | x | x | x |
| File sharing / record locking | x | x | x | x |
| UCX/NFS support (nolocking) | N | N | N | x |
| FUNCTION ARGCOUNT | x | x | x | N |
| Little-endian COMP data | x | x | x | x |
| 31-digit user items | N | x | x | x |
| 18-digit intermediates | x | x | x | x |
| 32-digit intermediates | N | x | x | x |
| F,D floating | x | x | x | N |
| G floating | N | x | x | N |
| IEEE S,T floating | N | x | x | x |
| Floating exception handling | x | x | P | x |
| Floating point "E" literal | x | x | x | x |
| Extended alphanumerics | N | x | x | x |
| 64-bit pointers | N | P | P | x |
| SYS$CURRENCY | x | x | x | N |

| | OpenVMS VAX | OpenVMS Alpha | OpenVMS I64 | Tru64 UNIX |
|---|---|---|---|---|
| | V5.7 | V2.9 | V3.0 | V2.8 |
| Invalid decimal data checking | P | P | P | P |
| Exception handling | x | x | P | x |
| ON SIZE ERROR | x | x | P | x |
| X/Open RETURN-CODE | N | x | x | x |
| X/Open COMP-5/COMP-X | N | x | x | x |
| X/Open LINE SEQUENTIAL | N | x | x | x |
| X/Open ASSIGN TO | N | x | x | x |
| X/Open SCREEN SECTION | N | x | x | x |
| X/Open DISPLAY ON EXCEPTION | N | x | x | x |
| X/Open file sharing / locking | N | x | x | x |
| X/Open environment variables | N | x | x | x |
| X/Open command line | N | x | x | x |
| /CHECK=(PERFORM,BOUNDS) | x | x | x | x |
| VAX-compatible alignment | x | x | x | x |
| Alpha natural alignment / padding | N | x | x | x |
| Targetted code generation / optimization | x | x | x | x |
| /[NO]SEPARATE_COMPILATION | P | x | x | N |
| PIC N for /NATIONALITY=JAPAN | x | x | x | x |
| CALL USING BY DESCRIPTOR | x | x | x | N |
| cobfunc, cobcall, and cobcancel | N | x | x | x |
| Reformat | x | x | x | x |
| Terminal source format | x | x | x | x |
| Lowercase, -/_ in source | x | x | x | x |
| Oracle CDD/DML support | x | x | x | N |
| Transarc Encina (-tps) support | N | N | N | x |
| DECset PCA/LSE/SCA support | x | x | x | N |
| DECset PDF support | x | N | N | N |
| FUSE support | N | N | N | x |
| Symbolic debugger support | x | x | x | x |
| Common command line for OpenVMS | x | x | x | N |
| Docs - UM/RM HTML and PDF | x | x | x | x |

# 7 Providing Product Feedback

If you have comments, suggestions, or questions about HP COBOL on I64, Alpha, or VAX, or if you are interested in signing up to field test a future version of HP COBOL, use the following URL for general information about HP COBOL:

http://h71000.www7.hp.com/commercial/cobol/

We are interested in your feedback both on the capabilities of HP COBOL V3.0 and on future capabilities you would like to see in HP COBOL.

When reporting a problem to Hewlett-Packard Company, please supply as much of the following information as you can:

- Description of the problem
- Version of the product
- Version of the operating system and other related software
- Sample source file which can be used to duplicate the problem
- Run-time call stack from the debugger if you are reporting an abnormal run-time termination
- Command line used to produce the problem

You can report problems via Internet mail, but use your regular HP support channel if you are reporting a problem and you need a response back when the problem is resolved.