

# HP-UX

---

## Porting Compaq Fortran Applications on Tru64 UNIX to HP-UX

**May 2005**

This white paper contains information about porting Fortran applications from Tru64 UNIX Version 5.1 and higher to HP-UX 11i v1.6 and higher. It is intended for software architects and developers who want to assess or begin porting Fortran applications.



---

# Porting Compaq Fortran applications on Tru64 UNIX to HP-UX

## Executive summary

This white paper contains information about porting Fortran applications from Tru64 UNIX® Version 5.1 and higher to HP-UX 11i v1.6 and higher. The intended audience for this document is the application architect or developer who wants to assess or begin porting applications.

The HP Fortran Compiler is being enhanced to assist you with the porting of your Fortran applications to HP-UX on Itanium®-based systems. Also, the HP Software Transition Kit for Tru64 UNIX (STK-T) contains tools and documentation to help you transition your source code from Tru64 UNIX to HP-UX, and is freely available on the Internet at the following location:

<http://www.hp.com/go/STK/>

For more information on the tools that are currently available, see the following Web site:

<http://www.hp.com/go/tru64appmigration>

This white paper is intended as a planning aid for application architects and developers who are interested in porting their Tru64 UNIX Alpha based Fortran applications to HP-UX 11i systems. It is assumed that you are familiar with the following:

- The effort required to port applications
- The technical aspects of the application to be ported

The goals of this white paper include:

- The identification of known, potential porting issues in application code.
- The identification of references and resources available to further investigate these potential issues.

This paper is not intended to be a complete resource for migrating Fortran applications from Tru64 UNIX to HP-UX. It is a sampling of issues that have been identified as differences on both platforms.

This paper is one of a series of papers in the HP-UX application migration series.

## Using this white paper

You can read all the topics in this paper sequentially, or you can go to a specific topic. This paper covers the following topics:

- *Features planned for a future release of HP Fortran*
- *Features not planned for a future release of HP Fortran*

## Features planned for a future release of HP Fortran

This section identifies those Compaq Fortran extensions and features that are not currently supported in HP Fortran, but are under consideration for a future release of the HP Fortran Compiler.

### Compiler issues

- Inline assembly for Itanium<sup>®</sup>-based systems.
- Fortran90 defines the percent sign (%) as the derived type field selector. Compaq Fortran also recognizes, as an extension to f90, the period (.) as an equivalent operator.
- Relaxation of restrictions on extensions. Specifically, allowing Cray pointers in derived types, f90 pointers in VMS records, and Cray pointers in module declarations.
- Inline directives for f90, possibly a command-line option for explicitly inlining.

### Input/output

- Support for binary data conversion on input of data from non-HP systems.
- The ability to write binary data to nonformatted direct files in non-HP format (for example, VAXD, VAXG, CRAY, IEEE, and little endian).
- Output format compatibility, including the ability to set the default field width for integers based on integer type.
- Support for the `UNIT' RECNUM` syntax in I/O statements.

### Intrinsic functions and subroutines

- EOF function, which inquires whether a file is at or beyond the end-of-file record.
- ERRSNS function, which returns the latest I/O subsystem error condition.
- IARGPTR intrinsic, which returns a pointer to the arguments passed to the program.

## Statement compatibility

- Support for the `VOLATILE` attribute.
- Even/Odd for true/false under flag control. This includes a compiler option to allow Boolean expressions to operate on a single bit instead of a whole word. For example, Compaq F90 evaluates the expression `(B'10' .AND. B'11')` to false while the current HP F90 evaluates it to true. An option will be added so that the HP compiler will evaluate Boolean expressions in the same way as the Compaq Fortran compiler.
- Improved documentation of the `FORALL` statement.
- Enhanced support for binary, octal, hexadecimal, and Hollerith constants so that they are legal wherever numeric constants are allowed.

## Compiler directive compatibility

- `ALIAS`
- `ATTRIBUTES`
- `FREEFORM` and `NOFREEFORM`
- `REFERENCE`
- `C`
- `VALUE`

## Compile-time option compatibility

The following directives are expected to be in the next release of HP-UX to enable the customer to easily port Tru64 UNIX makefiles (Alpha versions) to HP-UX (Itanium® versions):

- `/CHECK =`
  - `{[NO]BOUNDS }`
  - `{BIG_ENDIAN }`
  - `{CRAY }`
- `/CONVERT =`
  - `{IBM }`
  - `{LITTLE_ENDIAN }`
  - `{NATIVE }`
- `/[NO]EXTEND_SOURCE`
- `/ASSUME = [NO]UNDERSCORE`

- /CHECK =
  - {ALL }
  - {[NO]OVERFLOW }
  - {[NO]UNDERFLOW }
  - {NONE }
- /NOCHECK
- /[NO]F77
- /FLOAT = {IEEE\_FLOAT }
- /[NO]I4 (and /real\_size == -r8
- /[NO]RECURSIVE

## Non-standard numerical expressions

Compaq and HP Fortran compilers both implement extensions to standard Fortran expression syntax, allowing unary operators (+ and -) to appear after other operators. Both compilers assign the same interpretation to standard-conforming expressions. However, they do not use identical interpretations of the nonstandard expressions.

Compaq Fortran implements the following operator precedence:

```
A**-B*C -> A**(-(B*C))
X/-Y*Z -> X/(-(Y*Z))
```

See Chapter 4 of the *Compaq Fortran Language Reference Manual* at the following location for more information:

<http://h21007.www2.hp.com/dspp/files/unprotected/Fortran/docs/lrm/dflrm.htm>

HP implements the preceding expressions as follows:

```
A**-B*C -> (A**(-B))*C
X/-Y*Z -> (X/(-(Y)))*Z
```

See Chapter 5 of the *HP Fortran Programmer's Reference* at the following location for more information:

<http://docs.hp.com/hpux/pdf/B3908-90003.pdf>

The following code is accepted by the Compaq Fortran compiler, but currently not accepted by the HP Fortran compiler:

```
integer a
a = 12+-3
print *,a
end
```

## Others

The ellipsis (. . .) C notation for escaped C-style strings.

## Features not planned for a future release of HP Fortran

The following Compaq Fortran features are not currently planned to be made available in HP Fortran. Wherever applicable a workaround has been suggested.

### Intrinsic functions and subroutines

- **COTAN intrinsic** (COTAN, DCOTAN, QCOTAN, COTAND, and DCOTAND).
- **DEFINE FILE I/O statement**, which allows a direct file to be treated as a sequential file.
- **FIND I/O statement**, which finds a record in a direct file.
- **H edit descriptor** on input.
- **REWRITE statement**, which writes to the current record of a direct access file. The current record is defined to be the last record accessed by a direct access READ statement. The BACKSPACE statement provides the same functionality for direct access files.
- **NUMBER\_OF\_PROCESSORS([DIM]) intrinsic**, which returns the total number of processors (peers) available to the program. (HPF only)
- **Fortran-66 version of the EXTERNAL statement**.
- **FP\_CLASS(X) intrinsic**, which returns the class of an IEEE real (S\_floating, T\_floating, or X\_floating) argument. On OpenVMS systems, the compiler option specifying IEEE floating format must be set.
- **NWORKERS() intrinsic**, which returns the number of processes executing a routine. NWORKERS is a specific function with no generic name. It is provided for compatibility with Compaq Fortran 77 for OpenVMS VAX systems.
- **PROCESSORS\_SHAPE() intrinsic**, which returns the shape of an implementation-dependent hardware processor array. (HPF only)
- **VAX and Win32 API support**. There are many Compaq Fortran INTRINSIC arguments that are documented to work only on VAX and Win32 systems. They are accepted by the Compaq FE, and in many cases do not even give a Runtime Error, but simply silently fail. The HP Fortran FE will give Compile Time errors for all such nonsupported arguments.

- **Windows Systems Character Sets** (the extended character sets for line drawing, such as ANSI and IBM ASCII extended line drawing character sets).
- **Key Codes.** Visual Fortran includes a number of run-time routines to query and set the NLS system, to change code pages, and set and retrieve current language settings. No routines were included in the source.
- **Variable seed size** used by the `PUT` parameter on the `RANDOM_SEED( )` intrinsic. The Compaq Fortran compiler requires only a 2-element array; the HP Fortran compiler requires at least a 17-element array. Also, the random number algorithms differ. Programs should not rely on the seed size value. Instead, programs should query for the seed size.

## Compiler directive compatibility

- `ALLOW_NULL`
- `DECLARE` **and** `NODECLARE`
- `DECORATE`
- `DEFAULT`
- `DEFINE`
- `EXTERN`
- `FIXEDFORMLINESIZE`
- `IDENT`
- `IF`
- `IF DEFINED`
- `IGNORE_LOC`
- `INTEGER`
- `MESSAGE`
- `NO_ARG_CHECK`
- `NOMIXED_STR_LEN_ARG`
- `OBJCOMMENT`
- `OPTIONS`
- `PACK`
- `PSECT`
- `REAL`
- `STDCALL`
- `STRICT` **and** `NOSTRICT`

- SUBTITLE
- TITLE
- UNDEFINE
- UNROLL
- VARYING

## Compiler options compatibility

- /CONVERT =
  - {FDX }
  - {FGX }
  - {VAXD }
  - {VAXG }
- /FLOAT =
  - {D\_FLOAT } (VMS only)
  - {G\_FLOAT } (VMS only)
- /[NO]G\_FLOATING (VMS only)

## Mathematical exactness

While all of the mathematical functions are supported for both the HP and Tru64 UNIX systems, their implementation and the hardware on which they run are different. This may cause inequalities in precise mathematical computations.

## The Hollerith issue

Formats in Fortran typically require a comma (,) to separate format edit items. For example:

```
10 FORMAT(3x,5hhello)
```

As an extension to how formats are defined in Fortran, the Compaq Fortran compiler allows the comma to be omitted. For example:

```
10 FORMAT(3x5hhello)
```

This functionality would not be available for Holleriths in HP Fortran.

Also, a Hollerith split across line boundaries is not supported by the HP Fortran compiler. For example, the following format is accepted by the Compaq Fortran compiler, but generates an error with the HP Fortran compiler:

```
FORMAT(////34H TRIPLE EXPONENTIAL SMOOTHING.....,A4,A2//22H NUMBER  
1 OF DATA POINTS,I6/19H SMOOTHING CONSTANT,F9.3/)
```

## Compaq zero-fills remainder of TRANSFER result

The Fortran specification states the following:

“If the physical representation of the result is longer than that of SOURCE, the physical representation of the leading part is that of SOURCE and the remainder is processor dependent.”

The Compaq Fortran compiler zero-fills the remainder of the result; the HP Fortran compiler leaves the remainder undefined. Source code changes might be required if the behavior depends on the undefined portions of the transfer result to be zero.

## Unresolved symbols from OpenMP

OpenMP is the only Parallel Programming Model supported by the HP Fortran compiler. Any symbol that is not part of OpenMP (for example, `_OtsGetThreadNum`) is not supported.

## Integer\*2 not allowed for labels

The HP Fortran compiler does not support integers less than four bytes in size for labels. For example, the following code fragment generates a syntax error:

```
integer*2 :: i  
assign 10 to i
```

or:

```
integer*2 :: i  
goto i
```

In these cases, change `integer*2` to `integer*4`.

## Problems recognizing some implied DO loop

The HP Fortran grammar for I/O arguments expects that in an I/O list an open parenthesis ( "(" ) denotes an implied DO and that two consecutive opening parentheses denotes a nested implied DO.

The following loop is recognized by the Compaq Fortran compiler, but not the HP Fortran compiler:

```
PRINT 105, ((ARE(I), AIM(I)), I=1, 5)
```

Rewrite the code as follows:

```
PRINT 105, (ARE(I), AIM(I), I=1, 5)
```

For a two-dimensional array, use a format similar to the following:

```
PRINT 106, ((ARE(I, J), J=1, 5), I=1, 5)
```

## Labeled FORMAT statements cannot be modified

The HP Fortran compiler does not allow labeled FORMAT statements to be modified. Consider the following example:

```
IRVI=5
NUVI=6
READ(IRVI, 0070)
0070 FORMAT(40H BASED ON ASA FORTRAN X3.9-1966  /)
WRITE(NUVI, 0070)
END
```

To obtain the desired result, use a character variable in place of the format, as follows:

```
CHARACTER*40 FORMAT0070
DATA FORMAT0070/" BASED ON ASA FORTRAN X3.9-1966  "/
IRVI=5
NUVI=6
READ(IRVI, '(A40)')FORMAT0070
WRITE(NUVI, '(A40)')FORMAT0070
END
```

## For more information

For more information on porting from Tru64 UNIX to HP-UX, see the following documentation and Web sites:

- The Software Transition Kit (STK), which includes file scanning tools and documentation on application migration at the following location:  
<http://www.hp.com/go/STK/>
- The chapters on migrating to HP Fortran and porting to HP Fortran in the *HP Fortran Programmer's Guide*, 3rd Edition:  
<http://docs.hp.com/hpux/dev/index.html#Fortran>
- *Tru64 UNIX to HP-UX 11i Porting Guide* at the following location:  
[http://h30097.www3.hp.com/transition/apps/porting\\_guide.html](http://h30097.www3.hp.com/transition/apps/porting_guide.html)
- Tru64 UNIX to HP-UX application transition Web site:  
<http://www.hp.com/go/tru64appmigration>

For more information about HP-UX development tools, especially the C compiler and linker, go to the following location:

<http://www.docs.hp.com/hpux/dev>

---

**Note**

---

Be sure to check the operating system versions to which the manuals apply; otherwise, you might try to use an Itanium<sup>®</sup> architecture feature on PA-RISC, or vice versa.

---

For all HP-UX documentation, go to the following location:

<http://www.docs.hp.com>

The following document might be useful:

For Tru64 UNIX Alpha Fortran documentation, go to the following location:

[http://h21007.www2.hp.com/dspp/tech/tech\\_TechDocumentDetailPage\\_IDX/1,1701,7155,00.html](http://h21007.www2.hp.com/dspp/tech/tech_TechDocumentDetailPage_IDX/1,1701,7155,00.html)

The HP IT Resource Center at <http://www.itrc.hp.com> is useful for searches. From the main page, click on search, enter the search text and check the ITRC forums option.

## Legal notices

© Copyright 2004, 2005 Hewlett-Packard Development Company, L.P.

Intel<sup>®</sup>, Itanium<sup>®</sup>, Itanium<sup>®</sup> 2, and Itanium<sup>®</sup>-compatible are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

POSIX<sup>®</sup> is a registered trademark of the Institute of Electrical and Electronic Engineers, Inc.

UNIX<sup>®</sup> is a trademark of The Open Group in the U.S. and/or other countries.

All other product names mentioned herein may be trademarks of their respective owners.

Confidential computer software. Valid license from HP required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.