

# **HP FORTRAN 77 for HP-UX 10.20 Release Note**

**Version 1.2 for HP-UX 10.20**

**HP 9000 Systems**



**Manufacturing Part Number : 5969-0901  
090199**

U.S.A.

© Copyright 1999 © Hewlett-Packard Company. All rights reserved..

---

## Legal Notices

The information in this document is subject to change without notice.

*Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.* Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

**Warranty.** A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

**Restricted Rights Legend.** Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY

3000 Hanover Street

Palo Alto, California 94304

U.S.A.

Use of this release note is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

**Copyright Notices.** ©copyright 1983-99 Hewlett-Packard Company, all rights reserved.

Reproduction, adaptation, or translation of this document without prior written permission is prohibited, except as allowed under the copyright laws.

©copyright 1979, 1980, 1983, 1985-93 Regents of the University of California

**Trademark Notices.** UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X Window System is a trademark of the Massachusetts Institute of Technology.

OSF/Motif is a trademark of the Open Software Foundation, Inc. in the U.S. and other countries.

# 1 HP FORTRAN 77 v1.2 for HP-UX 10.20 Release Note

---

## Announcement

HP FORTRAN 77 v1.2 for HP-UX 10.20 is the first release of the HP FORTRAN 77 compiler since November 1996. This release of the compiler corrects runtime defects as well as improves buildtime performance. New features have been added to enhance the performance of the compiler on both PA-RISC 1.1 and 2.0 architectures.

## What's in This Version

This version of the HP FORTRAN 77 distribution package consists of the following components:

- The FORTRAN 77 compiler based on the International ANSI/ISO Standard
- Standard HP FORTRAN 77 library
- Compiler utilities (lintfor, ratfor, fsplit)
- HP Distributed Debugging Environment (DDE)
- HP PAK Performance Analyzer
- xdb debugger

### New in FORTRAN 77 v1.2

HP FORTRAN 77 v1.2 for HP-UX 10.20 contains one new feature: DA1.1 data prefetching. The `+DA1.1 +Odataprefetch` flags allow programs to be built to run on both the PA-RISC 1.1 and 2.0 architectures with the same benefits as 2.0 prefetching on 2.0 architectures. These flags cause DA2.0 prefetches to be inserted into the code in the same place they would be for `+DA2.0 +Odataprefetch`. See “Benefits” of this release note for more information on DA1.1 data prefetching.

---

**NOTE** HP FORTRAN 77 for HP-UX 10.20 does *not* support parallelism (including parallel directives) or 64-bit addressing ("wide mode"). If you want to take advantage of these features, you must upgrade your operating system to HP-UX 11.x and run the HP Fortran 90 for 11.x compiler.

---

---

## Benefits

HP FORTRAN 77 v1.2 for HP-UX 10.20 includes DA1.1 data prefetching. This feature is beneficial if you want to run a single `+Odataprefetch` executable on both PA-RISC 1.1 and 2.0 architectures. Previously, if you built a `+Odataprefetch` executable on a 2.0 machine, the 1.1 architecture would not run the prefetch instructions.

## Updates to user documentation

The following are optimization options and features included with HP FORTRAN 77 HP-UX 10.20 but are not documented in the *HP FORTRAN/9000 Programmer's Guide* or *HP FORTRAN/9000 Programmer's Reference*.

- `+Odataprefetch` improves cache performance by inserting data-prefetch instructions in program loops.
- `+Ofail_safe` compiles at a lower level of optimization when internal optimization errors occur.
- `+Oinline_budget` allows more aggressive inlining at the expense of increased code size and compilation time.
- `+Oloop_unroll` enables loop unrolling.
- `+Ostatic_prediction` performs static branch prediction when used with the `+P` option to perform Profile-Based Optimization (PBO).
- `+Owhole_program_mode` asserts that only files compiled with this option reference globals; allows advanced optimizations.
- The 10.20 release of the HP FORTRAN 77 compiler for HP-UX 10.20 supports basic-style debugging of optimized code, as described in the online help for the HP/DDE debugger.
- The `+DA` and `+DS` compile-line options have two additional arguments:
  - 2.0: `+DA2.0` and `+DS2.0` support the new PA-RISC 2.0 architecture.
  - `portable` (available only with `+DA`): `- +DAportable` generates code that is compatible with both PA-RISC 1.1 and 2.0 architectures.
- The semantics of the `+T` compile-line option have changed. If you are using the `+T` option on a program that is being optimized at level 2 or higher, you must specify the `+T` option for each optimized file in the program.

## Benefits

- If you specify the `-O` option with the `fort77` command when compiling your programs, you must include an optimization level number with the option. The level number must be an integer in the range of 0 - 4, where 0 specifies no optimization and 4 specifies full optimization. If the option does not include a level number, the compiler will issue a warning message and ignore the option. The following command-line shows how to use the `-O` option with the `fort77` command to specify optimization `+O2`: `fort77 -O2 prog.f`  
See the `f77` manpage for more information.

## Known Problems and Workarounds

This section details known problems and workarounds for HP FORTRAN 77 v1.2 that are not currently documented in the *HP FORTRAN/9000 Programmer's Guide* or *HP FORTRAN/9000 Programmer's Reference*.

- Programs with loops containing integer multiplies may fail at +O2 with the error "Utils: Sanity Check Inconsistent internal data structures (6933)". Use +Onoloopunroll to prevent this failure.

### Memory Consumption When Compiling at Optimization Level 4

When you link a program, the compiler brings all modules that were compiled at optimization level 4 into virtual memory at the same time. Depending on the size and number of the modules, compiling at +O4 can consume a large amount of virtual memory. Compiling with +O4 will consume about .5 megabytes per 1000 lines of noncommented source. If you are linking a very large program that was compiled with the +O4 option, you may notice a system slow-down. In the worst case, you can get an error indicating that you have run out of memory. There are several things you can do.

The simplest and best solution is to compile at +O4 only those modules that need to be compiled at optimization level 4 and to compile the remaining modules at a lower level.

If you still find that you are running out of memory, the first thing you should do is to increase the per-process data size limit. Run the System Administrator Manager (SAM) to increase the MAXDSIZ process parameter from 64 MB to 128 MB (this can be increased up to 960 MB). You can use SAM to set the different process parameters, including MAXDSIZ.

If increasing the per-process data size limit does not solve the problem, you should increase the system swap space. Refer to the *HP-UX System Administrative Tasks* (B2355-90051), Chapter 6, to learn how to increase the amount of swap space. Pay particular attention to the section "Adding File System Swap", since adding file system swap is easier than increasing the amount of device swap, which requires re-configuring your disk. However, if you find that you are consistently compiling beyond the available amount of device swap, you may not have a choice.

### Optimization and the ON Statement

When compiling at optimization level 2, 3, or 4, the user should be aware that the optimizer makes assumptions about the program that do not take into account the behavior of procedures called by the ON ... CALL statement. Such procedures must therefore be well-behaved in optimized programs. In particular:

## Known Problems and Workarounds

- The ON procedure must not assume that any variable in the interrupted procedure or in its caller has its current value. For example, the optimizer may have placed the variable in a register until after the call to the interrupted procedure is complete.
- The ON procedure must not change the value of any variable in the interrupted procedure or in its caller if the effect of the ON procedure is to return program control to the point of interrupt.

---

**NOTE** Note that if you compile at optimization levels 0 (+O0) and 1 (+O1), the ON procedure does not have to be restricted by these requirements.

---

## Restrictions in HP FORTRAN 77

This section lists restrictions you should observe when using the HP FORTRAN 77 compiler and when using HP FORTRAN 77 language features in your programs.

- If you do not use the +DA option to specify an architecture, the default architecture object code generation is that of the machine on which you compile. Likewise, if you do not use the +DS option, the default instruction scheduling is that of the machine on which you compile.

---

**NOTE** The portable argument to the +DA and +DS options creates object code compatible across all PA-RISC 1.1 and 2.0 workstations and servers. It however does not support the PA-RISC 1.0 architecture.

---



## Compatibility Information and Installation Requirements

This section describes system requirements for the installation and running of HP FORTRAN 77 v1.2 for HP-UX 10.20.

### Software requirements

- You can install HP FORTRAN 77 v1.2 after loading the HP-UX 10.20 operating system. To install your software, run the SD-UX swinstall command. It invokes a user interface that leads you through the installation process and give you information about product size, version numbers, and dependencies.
- The HP FORTRAN 77 compiler requires approximately 46 megabytes (MB) of disk space. This includes approximately 16 MB for the compiler, with remaining space for other components such as HP/PAK.
- The following patches are required for Fortran 77 v1.2 to correctly run on HP-UX 10.20:
  - PHSS\_14507 (HP-UX 10.01 or 10.10)
  - PHSS\_17689 (HP-UX 10.20)
  - PHSS\_15389 (HP-UX 10.01 or 10.10)
  - PHSS\_17903 (HP-UX 10.01, 10.10, or 10.20)
  - PHSS\_17225 (HP-UX 10.01, 10.10 or 10.20)

### Hardware requirements

HP FORTRAN 77 v1.2 is supported on HP9000 Series 700/800 hardware running the HP-UX 10.20 operating system.

### Operating system requirements

HP FORTRAN 77 v1.2 recommends that HP-UX version 10.20 be installed.

For more information about installation procedures and related issues, refer to *Managing HP-UX Software with SD-UX* and other README, installation, and upgrade documentation included or described in your HP-UX 10.20 operating system package.

## Support information on HP FORTRAN 77

HP customers who have purchased support contracts can find a list of HP FORTRAN 77 language problems and their fixes in the current Software Status Bulletin (SSB). This information can be found by referencing the following product numbers:

- B3906AA -- HP FORTRAN/9000 Series 700
- B3908AA -- HP FORTRAN/9000 Series 800

The *Software Status Bulletin* (SSB) is available to customers on support. To display the product number and the release version of your HP FORTRAN 77 compiler, execute this HP-UX command:

```
what /opt/fortran/bin/f77
```

Customers can access HP's Electronic Support Center on the World Wide Web, which permits searching for bug descriptions and available patches. This is available at the following web addresses:

- <http://us-support.external.hp.com>  
for customers in the US, Canada, Asia-Pacific, and Latin America.
- <http://europe-support.external.hp.com>  
for European customers

## Related Documentation

The following documents are available for your use in understanding the HP FORTRAN 77 compiler. Please note that this release note supersedes information in these manuals and white papers.

- *f77(1)* man page, which provides a summary reference to the compile-line options
- HP FORTRAN/9000 Online Reference (part of the Common Desktop Environment help system and the VUE help system)
- *HP PA-RISC Compiler Optimization Technology White Paper* (5964-9846E). For a PostScript version of this document, see `/opt/langtools/newconfig/white_papers/optimize.ps`.
- Information in these documents supplements the *HP FORTRAN/9000 Programmer's Reference* (B2408-90002) and the *HP FORTRAN/9000 Programmer's Guide* (B2408-90001).

---

**NOTE** This release note is also available for viewing and printing in the following location and file formats:  
</opt/fortran/newconfig/RelNotes/Fortran.10.20.16.html> | pdf | ps | txt.

---

## Patches and Fixes in This Version

This section describes patches and fixes for HP FORTRAN 77 that are included with this release.

### Patches

The following patch and associated fixes are included with this version of HP FORTRAN 77 for HP-UX 10.20:

#### Patch Number Problem/Fix

**PHSS\_12183** Cumulative patch which corrects loop transformation errors, incorrect optimizations, memory problems, and miscellaneous compiler errors such as wrong answers. These fixes are described in “Fixes” of this release note.

### Fixes

The following list describes problems that have been fixed in HP FORTRAN 77 v1.2 for HP-UX.

- Loop transformation errors at level +O3 with +DA1.1 and/or +DS1.1
- Internal compiler errors caused by complicated declarations with function parameters which contain large parameters.
- Nonstandard-conforming multiple unnamed common block subprogram execution problems when +I (Profile-based Optimization or PBO) was used.
- IEEE invalid operation errors resulting from compiling with +O3.
- Procedures with a large number of ASSIGNs and ASSIGNED GOTOs took a long time to compile when used with +I.
- Routines with greater than a fixed number of arguments were incorrectly optimized.
- Incorrect optimization of loops containing array references and conditional updates of a variable used as an array index occurred at optimization level +O2 and higher.
- Some loop constructs caused incorrect answers when used with +DA2.0 and +O2 command line flags.
- At optimization level +O2 the MAX and MIN intrinsic functions produced wrong answers.
- Functions with more than 256 parameters were incorrectly optimized at optimization levels +O3 or higher.

- Some loop constructs caused fatal internal errors when compiled at optimization level +O2 or higher.
- Compiling programs with +DA2.0 +Odataprefetch at +O2 or higher could cause runtime memory faults. The failure was eliminated when +Odataprefetch was not used.
- When some complex loops were compiled at optimization level +O2, FORTRAN 77 would die with a segmentation violation.
- When two different-sized arrays were equivalenced together at optimization level +O3 using the -K flag, and only the smaller array was referenced, there was not always enough memory allocated to hold the larger array. This resulted in incorrect results or runtime failures.
- Compiling with +O4 +Oall +Owhile\_program\_mode failed with fatal errors in cases where the addresses of user routines were passed as arguments and the routines were completely cloned or inlined.
- Subprograms compiled at optimization level +O2 with the -K flag failed with a segmentation violation if the subprogram contained an expression where a parameter variable was negated.
- Compiling some programs with +O2 +DA2.0 +DS2.0 caused fatal errors.
- When programs were compiled with +O3 +Onolimit, the system displayed a "Memory fault (coredump)" message. Using +Onoloop\_transform corrected this.
- The compiler aborted at +O3 +Oreg\_reassoc when used with programs with nested loops.
- Some statement functions which contained an intrinsic function caused the compiler to issue an error if the statement function was never used and IMPLICIT NONE was in effect.
- Labels on a source line number greater than 65535 were not handled correctly.
- Using xdb, you could not display a dynamically allocated array variable in a program compiled with the -K option (or +e).
- When an element of a dynamically allocated array of structures was passed as an argument, the passed pointer always pointed to the first element of the array instead of to the correct element.
- Incorrect data was displayed by the xdb or dde debuggers for parameters declared as CHARACTER\*(\*).
- All symbols in the main program occasionally became invisible to SoftStatic when the +ppu option was used.
- Incorrect results occurred when compiling with +O2 or +Opipeline.

**Patches and Fixes in This Version**

- Some programs ran out of stack space when certain compiler options, such as +DA or +DS were used, and sometimes resulted in incorrect behavior.
- FORTRAN 77 sometimes failed while optimizing multiple nested loops.
- FORTRAN 77 ran out of memory when compiling with +O4 or PBO and some of the modules used +Onolimit.
- Incorrect results occurred when a Cray pointer was passed to a subprogram compiled with optimization level +O2 or higher.

## **Software Availability in Native Languages**

There are no non-English translations for HP FORTRAN 77 v1.2 for HP-UX 10.20.