

**Intel Corporation**  
5200 N.E. Elam Young Parkway  
Hillsboro, OR 97124-6497

(503) 696-8080



March 1994

**Dear Paragon™ Customer:**

This package contains Release 4.5 of the Paragon™ Fortran compiler for your Paragon™ system.

**Before using your system:**

- **Read this letter completely.**
- **Verify the contents of this package.**
- **Read the *Paragon™ Fortran Compiler Software Product Release Notes*.**

**Package Contents**

Your Paragon™ Fortran compiler software package should include the items listed in Table 1 (Installation Media) and Table 2 (Documentation). If any items are missing, or if you have any questions, please contact Intel Supercomputer Systems Division as described in the "Comments and Assistance" section.

Your package should contain one (and only one) of the cartridge tapes listed in Table 1 (depending on your host).

**Table 1. Installation Media**

<b>Description</b>	<b>Order Number</b>
Paragon™ Fortran Compiler Release 4.5 Native, Sun4, and Solaris-Hosted	313013-001
Paragon™ Fortran Compiler Release 4.5 Silicon Graphics-Hosted	313017-001



**Table 2. Documentation**

Description	Order Number
<i>Paragon™ Fortran Compiler Software Product Release Notes</i>	313012-001
<i>Paragon™ Fortran Compiler User's Guide</i>	312491-002

### **Restrictions and Limitations of Compiler Release 4.5**

Every effort has been taken to ensure the quality of this release, but at shipping time we are aware of a few problems. Please refer to the *Paragon™ Fortran Compiler Software Product Release Notes* for known limitations and workarounds.

### **Installation**

For directions on how to install your Paragon™ Fortran compiler, refer to Chapter 5 of the *Paragon™ Fortran Compiler Software Product Release Notes*.

### **NOTE**

You must have System Software Release 1.1.4 or greater installed on your system in order to install Compiler Release 4.5.



## Comments and Assistance

Intel Supercomputer Systems Division is eager to hear of your experiences with the Paragon™ Fortran compiler. Please call us if you need assistance, have questions, or otherwise want to comment on your Paragon™ system.

**U.S.A./Canada Intel Corporation**  
**Phone: 800-421-2823**  
**Internet: [support@ssd.intel.com](mailto:support@ssd.intel.com)**

---

**Intel Corporation Italia s.p.a.**  
Milanofiori Palazzo  
20090 Assago  
Milano  
Italy  
1678 77203 (toll free)

**France Intel Corporation**  
1 Rue Edison-BP303  
78054 St. Quentin-en-Yvelines Cedex  
France  
0590 8602 (toll free)

**Intel Japan K.K.**  
**Supercomputer Systems Division**  
5-6 Tokodai, Tsukuba City  
Ibaraki-Ken 300-26  
Japan  
0298-47-8904

**United Kingdom Intel Corporation (UK) Ltd.**  
**Supercomputer System Division**  
Pipers Way  
Swindon SN3 IRJ  
England  
0800 212665 (toll free)  
(44) 793 491056 (*answered in French*)  
(44) 793 431062 (*answered in Italian*)  
(44) 793 480874 (*answered in German*)  
(44) 793 495108 (*answered in English*)

**Germany Intel Semiconductor GmbH**  
Dornacher Strasse 1  
85622 Feldkirchen bei Muenchen  
Germany  
0130 813741 (toll free)

---

**World Headquarters**  
**Intel Corporation**  
**Supercomputer Systems Division**  
15201 N.W. Greenbrier Parkway  
Beaverton, Oregon 97006  
U.S.A.  
(503) 629-7600 (Monday through Friday, 8 AM to 5 PM Pacific Time)  
Fax: (503) 629-9147

If you have comments about our manuals, please fill out and mail the enclosed Comment Card. You can also send your comments electronically to the following address:

**[techpubs@ssd.intel.com](mailto:techpubs@ssd.intel.com)** (Internet)



## Users' Group

The Intel Supercomputer Users Group promotes the exchange of information among users. Intel strongly supports the Users Group and encourages participation in its activities, which include: Special Interest Groups (SIGs), an annual international users conference, an electronic mail task force, and a "freeware" library of user-contributed software, available electronically to all members of the Intel Supercomputer Users' Group. For membership information contact:

**JoAnne Wold** (503-629-5322)  
**joanne@ssd.intel.com** (Internet)

Sincerely,



Steve Cannon

Product Marketing Manager  
Intel Supercomputer Systems Division

---

Paragon is a registered trademark of Intel Corporation  
Silicon Graphics is a registered trademark of Silicon Graphics, Inc.

Copyright © 1994 Intel Corporation





March 1994

Order Number: 313012-001

---

**Paragon™ Fortran Compiler**  
**Release 4.5**  
**Software Product Release Notes**

---

**Intel® Corporation**

Copyright ©1994 by Intel Supercomputer Systems Division, Beaverton, Oregon. All rights reserved. No part of this work may be reproduced or copied in any form or by any means...graphic, electronic, or mechanical including photocopying, taping, or information storage and retrieval systems...without the express written consent of Intel Corporation. The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update or to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication, or disclosure is subject to restrictions stated in Intel's software license agreement. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraphs (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013. Intel Corporation, 2200 Mission College Boulevard, Santa Clara, CA 95052-8119. For all Federal use or contracts other than DoD, Restricted Rights under FAR 52.227-14, ALT. III shall apply.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

286	i386	Intel	iPSC
287	i387	Intel386	Paragon
Concurrent File System	i486	Intel387	ProSolver
Direct-Connect Module	i487	Intel486	
i	i860	Intel487	

APSO is a service mark of Verdex Corporation

DGL is a trademark of Silicon Graphics, Inc.

Ethernet is a registered trademark of XEROX Corporation

EXABYTE is a registered trademark of EXABYTE Corporation

Excelan is a trademark of Excelan Corporation

EXOS is a trademark or equipment designator of Excelan Corporation

FORGE is a trademark of Applied Parallel Research, Inc.

Green Hills Software, C-386, and FORTRAN-386 are trademarks of Green Hills Software, Inc.

GVAS is a trademark of Verdex Corporation

IBM and IBM/VS are registered trademarks of International Business Machines

Lucid and Lucid Common Lisp are trademarks of Lucid, Inc.

NFS is a trademark of Sun Microsystems

OpenGL is a trademark of Silicon Graphics, Inc.

OSF, OSF/1, OSF/Motif, and Motif are trademarks of Open Software Foundation, Inc.

PGI and PGF77 are trademarks of The Portland Group, Inc.

PostScript is a trademark of Adobe Systems Incorporated

ParaGraph was developed at Oak Ridge National Laboratory by M. Heath and J. Finger under a research grant from D.O.E.

ParaSoft is a trademark of ParaSoft Corporation

SCO and OPEN DESKTOP are registered trademarks of The Santa Cruz Operation, Inc.

Seagate, Seagate Technology, and the Seagate logo are registered trademarks of Seagate Technology, Inc.

SGI and SiliconGraphics are registered trademarks of Silicon Graphics, Inc.

Sun Microsystems, Solaris, and the combination of Sun and a numeric suffix are trademarks of Sun Microsystems

The X Window System is a trademark of Massachusetts Institute of Technology

UNIX is a trademark of UNIX System Laboratories

VADS and Verdex are registered trademarks of Verdex Corporation

VAST2 is a registered trademark of Pacific-Sierra Research Corporation

VMS and VAX are trademarks of Digital Equipment Corporation

VP/ix is a trademark of INTERACTIVE Systems Corporation and Phoenix Technologies, Ltd.

XENIX is a trademark of Microsoft Corporation

## **WARNING**

Some of the circuitry inside this system operates at hazardous energy and electric shock voltage levels. To avoid the risk of personal injury due to contact with an energy hazard, or risk of electric shock, do not enter any portion of this system unless it is intended to be accessible without the use of a tool. The areas that are considered accessible are the outer enclosure and the area just inside the front door when all of the front panels are installed, and the front of the diagnostic station. There are no user serviceable areas inside the system. Refer any need for such access only to technical personnel that have been qualified by Intel Corporation.

## **CAUTION**

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

## **LIMITED RIGHTS**

The information contained in this document is copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure by the U.S. Government is subject to Limited Rights as set forth in subparagraphs (a)(15) of the Rights in Technical Data and Computer Software clause at 252.227-7013. Intel Corporation, 2200 Mission College Boulevard, Santa Clara, CA 95052. For all Federal use or contracts other than DoD Limited Rights under FAR 52.2272-14, ALT. III shall apply. Unpublished—rights reserved under the copyright laws of the United States.



# Preface

---

These release notes provide the latest information on Release 4.5 of the Paragon Fortran compiler.

These release notes assume that you are an application programmer proficient in the Fortran language and the UNIX operating system.

## Organization

- |           |  |
|-----------|--|
| Chapter 1 | Introduces the features of the Release 4.5 Fortran compiler. |
| Chapter 2 | Describes resolved limitations for this release.             |
| Chapter 3 | Describes known limitations and workarounds of this release. |
| Chapter 4 | Provides some hints and suggestions for using the compiler.  |
| Chapter 5 | Provides installation instructions for this release.         |

## Notational Conventions

This manual uses the following notational conventions:

- |               |   |
|---------------|---|
| <b>Bold</b>   | Identifies command names and switches, system call names, reserved words, and other items that must be used exactly as shown.   |
| <i>Italic</i> | Identifies variables, filenames, directories, processes, user names, and writer annotations in examples. Italic type style is also occasionally used to emphasize a word or phrase. |
-

**Plain-Monospace**

Identifies computer output (prompts and messages), examples, and values of variables. Some examples contain annotations that describe specific parts of the example. These annotations (which are not part of the example code or session) appear in *italic* type style and flush with the right margin.

**Bold-Italic-Monospace**

Identifies user input (what you enter in response to some prompt).

**Bold-Monospace**

Identifies the names of keyboard keys (which are also enclosed in angle brackets). A dash indicates that the key preceding the dash is to be held down *while* the key following the dash is pressed. For example:

**<Break>**      **<s>**      **<Ctrl-Alt-Del>**

- [ ] (Brackets) Surround optional items.
- ... (Ellipsis dots) Indicate that the preceding item may be repeated.
- | (Bar) Separates two or more items of which you may select only one.
- { } (Braces) Surround two or more items of which you must select one.

## Applicable Documents

For more information, refer to the *Paragon™ System Technical Documentation Guide*.

## Comments and Assistance

Intel Supercomputer Systems Division is eager to hear of your experiences with our products. Please call us if you need assistance, have questions, or otherwise want to comment on your Paragon system.

**U.S.A./Canada Intel Corporation**  
**Phone: 800-421-2823**  
**Internet: [support@ssd.intel.com](mailto:support@ssd.intel.com)**

---

**Intel Corporation Italia s.p.a.**  
Milanofiori Palazzo  
20090 Assago  
Milano  
Italy  
1678 77203 (toll free)

**France Intel Corporation**  
1 Rue Edison-BP303  
78054 St. Quentin-en-Yvelines Cedex  
France  
0590 8602 (toll free)

**Intel Japan K.K.**  
**Supercomputer Systems Division**  
5-6 Tokodai, Tsukuba City  
Ibaraki-Ken 300-26  
Japan  
0298-47-8904

**United Kingdom Intel Corporation (UK) Ltd.**  
**Supercomputer System Division**  
Pipers Way  
Swindon SN3 IRJ  
England  
0800 212665 (toll free)  
(44) 793 491056 (*answered in French*)  
(44) 793 431062 (*answered in Italian*)  
(44) 793 480874 (*answered in German*)  
(44) 793 495108 (*answered in English*)

**Germany Intel Semiconductor GmbH**  
Dornacher Strasse 1  
85622 Feldkirchen bei Muenchen  
Germany  
0130 813741 (toll free)

---

**World Headquarters**  
**Intel Corporation**  
**Supercomputer Systems Division**  
15201 N.W. Greenbrier Parkway  
Beaverton, Oregon 97006  
U.S.A.

(503) 629-7600 (Monday through Friday, 8 AM to 5 PM Pacific Time)  
Fax: (503) 629-9147

If you have comments about our manuals, please fill out and mail the enclosed Comment Card. You can also send your comments electronically to the following address:

**[techpubs@ssd.intel.com](mailto:techpubs@ssd.intel.com)**





# Table of Contents

---

## Chapter 1 Product Features

Release 4.5 Features .....	1-1
Compiler Switch Changes .....	1-3
Default Compiler Switch Settings .....	1-4
Scope of Directives .....	1-5
Beta Vector Library .....	1-6

## Chapter 2 Resolved Limitations

## Chapter 3 Limitations and Workarounds

PostScript Copies of the Manuals .....	3-1
The Buglist File .....	3-1

## **Chapter 4 Hints and Suggestions**

## **Chapter 5 Installation**

<b>Installing the Native Compilers .....</b>	<b>5-1</b>
<b>Installing the Cross-Development Compilers .....</b>	<b>5-5</b>

# Product Features

1

## NOTE

Report any problems you encounter while using the Release 4.5 Fortran Compiler software to SSD Technical Support at:

**1-800-421-2823** (Customer Support Response Center)  
**Your Local Intel Sales Office** (in Europe)  
**support@ssd.intel.com** (Internet address)

## Release 4.5 Features

- There are several new compiler switches and some changes to the existing compiler switches. For a description of the new switches and the changes to existing switches, refer to the section *Compiler Switch Changes*.
- A new vectorization switch has been added, **-Mvect=streamlim:n**, to allow you to control the minimum size of vectors for which data streaming optimizations are performed. For a complete description of this switch, refer to the section *Compiler Switch Changes*.
- A new switch, **-Msafealloc**, is available to allow better optimization of programs that use allocatable commands that are allocated only once. For a complete description of this switch, refer to the section *Compiler Switch Changes*.
- New loop splitting switches are available. **-M[no]split\_loop\_ops** and **-M[no]split\_loop\_refs** allow you to set thresholds for loop splitting. For a complete description of these switches, refer to the section *Compiler Switch Changes*.
- Some changes have been made to the scope rules for directives. Refer to the section *Scope of Directives* for a complete description of the scope changes.
- There is a new directive, **c\$pragma c** (*<id>*, [, *<id>* ] ... ). This directive marks external functions written in C. The compiler does not append an underscore to the specified functions.

- The environment variable *MAKECPP* is supported. *MAKECPP* is a colon-separated list of directories that is added to the compiler's search path for include files.
- The path names for tools and libraries are no longer partially hardcoded. The Fortran Compiler driver now executes the first *iftn*, *ld860*, *as860*, and so on that it finds in your path. To use the R4.5 compilers and tools you must add the directory where they reside to your *PATH* environment variable.
- *libnx.a* is no longer automatically linked in if *-Mperfmon* is in effect. For Unix applications that use any of the nx functions such as *dclock*, *-lnx* should be added to the driver or linker command line when these applications are built.
- The search path for libraries has changed. If *PARAGON\_LPATH* or *LPATH* are defined, these directories are searched before any directories added with the *-L* switch.
- This release has been validated by NIST.
- Improvements have been made to the *-Minfo* option to accurately detail the optimizations being performed.
- A global default configuration file is now supported. The compiler searches the following directories in the order listed for the *.icfrc* file.
  1. your current working directory
  2. your home directory
  3. the directory where the compiler driver resides
- The installation instructions for the cross-development environment now include instruction to allow a site to create a new cross-dev directory without having to make copies of system libraries and include files.
- A beta library containing some vector functions that have been optimized for the Paragon X/P has been added. Refer to the section *Beta Vector Library* for complete information.
- The set of intrinsics affected by *-Mr8intrinsic*s has been expanded to include the following intrinsics:

<b>alog</b>	<b>alog10</b>	<b>amax1</b>	<b>amin1</b>	<b>amod</b>
<b>csqrt</b>	<b>clog</b>	<b>cexp</b>	<b>csin</b>	<b>ccos</b>

## Compiler Switch Changes

This section outlines the new compiler switches that have been added in this release and changes to the existing switches. The new switches are:

**-Msplit\_loop\_ops=*n***

Set a threshold of *n* floating-point operations within a loop. Innermost loops whose number of floating-point operations exceeds *n* are split. Each floating-point operation counts as two. The default for *n* is 40 when **-Mvect** is used.

**-Mnosplit\_loop\_ops**

Do not split loops when the floating-point operation threshold is exceeded. When **-Mvect** is specified, innermost loops whose number of floating point operations exceed 40 are split by default. This switch turns the default off.

**-Msplit\_loop\_refs=*n***

Set a threshold of *n* array element loads and stores within a loop. Innermost loops whose number of loads and stores exceeds *n* are split. The default for *n* is 20 when **-Mvect** is used.

**-Mnosplit\_loop\_refs**

Do not split loops when the array element loads and stores threshold is exceeded. When **-Mvect** is specified, innermost loops whose number of array element loads and stores exceeds 20 are split by default. This switch turns the default off.

**-Mvect=streamlim:*n***

This sets a limit for application of the vectorizer data streaming optimization. If data streaming requires cache vectors of length less than *n*, the optimization is not performed. Other vectorizer optimizations are still performed. The data streaming optimization has a high overhead compared to other loop optimizations, and can be counter-productive when used for short vectors. The *n* specifier is not optional. The default limit is 32 elements if **streamlim** is not used.

**-nostdinc**

Equivalent to **-Mnostdinc**.

There have also been changes to some of the existing switches.

- **-I-** is accepted but has no effect.
- **-Mnostride0** is now the default.

## Default Compiler Switch Settings

The default compiler switch settings are set for ease of porting, safe optimization, and high-speed compilation. Some of the defaults are:

- O1** Optimization level one
- Mnostride0** Do not check for zero stride induction variables.
- Mnodebug** Debugging disabled
- Mperfmon** Performance monitoring enabled
- Mnoframe** Don't include stack frame pointers on stack
- Kieee** Math conforms to IEEE 754 standard
- Mdepchk** Assume that potential data dependencies exist
- Msplit\_loop\_ops=40**  
Split innermost loops whose number of floating-point operations exceeds 40 if **-Mvect** is specified.
- Msplit\_loop\_refs=20**  
Split innermost loops whose number of array element loads and stores exceeds 20 if **-Mvect** is specified.

For better performance, you may use values other than the defaults, or change your defaults with a configuration file. For example, some appropriate user-defined defaults might be:

- O2** Optimization level two
- Mnoperfmon** No performance monitoring
- Knoieee** Non-IEEE math, if floating point accuracy is not critical
- Mnodepch** Assume that no potential data dependencies exist

If you use these suggested values as user-defined defaults, then in order to debug the program you have to override several of them. For example, to debug, you would want to use the **-g** command line switch. The **-g** switch is equivalent to the following:

- O0 -Mframe -Mdebug**

For best performance you may need to override the suggested defaults with command line switches such as the following:

**-O3** or **-O4**

**-Mvect**

For more information on **-Mnostride0**, **-Knoieee**, **-Mvect**, and other switches, see the *Paragon OSF/1 Fortran Compiler User's Guide*

## NOTE

If your application contains a loop with an induction variable whose increment (stride) is zero, you should add the **-Mstride0** switch to the compiler command line. **-Mstride0** is no longer the default.

## Scope of Directives

The scope of Fortran directives has changed in this release. For directives that allow **loop**, **routine**, and **global** scope, the following rules apply:

<b>loop</b>	Indicates the directive applies to the next lexical loop. The directive does not apply to any loops that are enclosed by the next loop. Loop-scoped directives are only applied to <b>DO</b> loops.
<b>routine</b>	Indicates the directive applies to the code that follows the directive until the end of the routine.
<b>global</b>	Indicates the directive applies to the code that follows the directive until the end of the file.

For directives where **loop** scope is not allowed, the scope rules fall into two groups.

The following rules apply to directives **func32**, **frame**, and **opt**:

<b>routine</b>	Indicates the directive applies to the current routine, if it is in a routine. If it is not in a routine, it applies to the next routine.
<b>global</b>	Indicates the directive applies to all routines that follow it.

The following rules apply to the directive **bounds**:

- |                |  |
|----------------|--|
| <b>routine</b> | Indicates the directive applies to the code that follows the directive until the end of the routine. |
| <b>global</b>  | Indicates the directive applies to the code that follows the directive until the end of the file.    |

## Beta Vector Library

When you use the **-Mvect** switch, the compiler usually generates calls to hand-coded library routines to perform certain vector operations. Some of these functions have been optimized for the Paragon system and have been placed in *libvbeta.a*. The amount of improvement you see by using functions from this library depends on the number of times the functions are called in your application. In particular, single-precision applications that use stride one array references should show improvement when you use this library. Some applications of this type had better performance with only software pipelining of loops than with both pipelining and vectorization.

Since these new versions have not undergone rigorous testing, they are provided as a beta library. To use this library, add **-lvbeta** to the driver command line. If you invoke **ld860** directly, insert it before **-lic**.



# Resolved Limitations

2

The following problem reports are fixed in this release. The number in brackets following each description is the problem report number.

- Fortran **namelist** no longer fails with three-dimensional arrays. [6865]
- The **secnds** function no longer returns a whole number. [3500]
- Opening an existing file with **status="new"** no longer fails when running on multiple nodes. [4046]
- Compile aborts with "if77: child XXXX caught signal" has been fixed. [7011]
- The precision problem causing wrong answers has been fixed. [5440]
- Line numbers for namelist read/write are now generated. [5491]
- An internal compiler error no longer occurs when **-Minline=50** is specified. [6287]
- A comma in column 25 no longer produces a data conversion error on the next **READ**. [6512]
- Compilation errors with **-Mvect** no longer occur. [6806]
- The driver no longer incorrectly passes the **-Msafeptr** switch to the compiler. [6940]
- **cosh(x)** and **sinh(x)** now return the correct value when  $709.783 < x < 710.471$ . [6365]
- Bessel functions no longer core dump when *arg* is between 0 and 298156826. [5794]
- **exp()** and **sin()** no longer return invalid results for an infinite input operand. [7010]
- The NAS parallel benchmark **appbt** running on 64 nodes no longer catches a SIGBUS signal and core dumps. [7477]

- The value of a constant passed to a subroutine can no longer be changed. [7463]
- The *libkmath.a* function **dtran** now works correctly. [7696]
- The function **dcos** no longer gives incorrect answers when a large negative input value is specified while pipelining is turned on. [7275]
- The compiler no longer introduces undefined symbols. [7120]
- Programs using **atan2** with vector lengths greater than 256 no longer fail when vectorized. [7456]

# Limitations and Workarounds

3

This chapter contains a list of known limitations and workarounds. This list is available in the file */usr/share/release\_notes/lf77\_buglist.ps* on the Paragon XP/S system.

The *buglist* file is updated just before shipment. The file included with your software may not be the latest version. Please contact the SSD Customer Support Response Center for information about how to get the latest version.

## PostScript Copies of the Manuals

PostScript copies of the Paragon compiler manuals are available in the directory */usr/share/ps.docs* on the Paragon system. Postscript copies of the release notes are available in the directory */usr/share/release\_notes*. The hardcopy version may be more up-to-date than the online version included on the release tapes, but the very latest online version can be obtained by contacting SSD Customer Support.

## The Buglist File

The rest of this chapter lists the outstanding bugs. The number on the left side of the first line for each bug is the bug number. Use this number when communicating with SSD Customer Support about the bug. An ASCII version of the *buglist* file is available in the directory */usr/share/release\_notes*. The hardcopy version may be more up-to-date than the online version included on the release tapes, but the very latest online version can be obtained by contacting SSD Customer Support.

- 7086            Compiling with **-Mvect -O4** results in the error "f\_dest must not be same as f\_src1."
- 7123            The Fortran compiler has some problems compiling Fortran code with structures.
- 7162            The compiler hangs when compiling *kernel.f* with **-Mvect** and **-Mnodepchk**. The workaround is to compile without using **-Mnodepchk**.
- 7751            The compiler produces a severe error on **t(mapt(*type*)) = 1.0** where *type* is **character\*(\*)**.
- 7885            Implicit definitions must precede all declarations or a program will not compile.

# Hints and Suggestions

## 4

This chapter provides some hints and suggestions for making the best use of the compiler.

1. The compiler may occasionally generate internal compiler messages. If they are of severity **W** (Warning) or **I** (Informational), the generated code is correct. However, please report all internal messages to SSD.
2. When using pipelining (**-O4**), the **-Mnodepchk** switch generally increases pipelining opportunities. If the program does not produce correct results with this switch, then it must be omitted. Use the switch only if you are sure no data dependencies that inhibit vectorization exist.
3. To ensure that the values of local variables are retained for the next invocation of a subprogram, use the **-Msave** switch. This allows older Fortran programs to run, but can greatly reduce performance.
4. The internal value of `.TRUE.` is `-1` unless the **-Munixlogical** switch is used. In that case, it has a value of `1`.
5. The **-Mvect** switch or the **-O4** switch may generate incorrect results if elements within a vector are accessed with stride zero. If this happens, add the **-Mstride0** switch.
6. The **-Mvect=unroll** switch is no longer supported or documented. The compiler prints the following warning message if you use it.

```
if77 - warning -- Mvect=unroll not implemented
```

7. No features are currently enabled by the **-Mbeta** switch.

8. The **-Mstride0** compiler switch should be used if a loop may contain an induction variable whose increment (stride) is zero. For example:

```
IS = 0
J = 0
DO I = 1,N
  A(J) = B(I) + 1.0
  J = J + IS
ENDDO
```

This switch may degrade performance so should only be used if zero-stride induction variables are possible.

9. The **-Knoieee** switch can give a substantial performance improvement. Division that does not conform to IEEE is several times faster than IEEE division, and some benchmarks run about twice as fast overall with the **-Knoieee** switch set. The penalty you pay for this performance is up to three low order bits of accuracy on certain division operations, and denormals are flushed to zero. The majority of division operations give identical results, whether or not IEEE math is used.
10. If your application runs slower when you use **-Mvect -O4**, try **-Mvect=streamlim:999 -O4**. The additional overhead of streaming in and streaming out data to and from cache could result in decreased performance if the vectors are short.
11. If your application uses only stride 1 array references, you may see increased performance if you use **-Mvect -Mstreamall** and link in *libvbeta.a*.
12. For applications with array references that are not stride 1, you may see increased performance if you use **-Mvect=streamlim:999**.

## NOTE

If you must re-install your system software, you must install the compilers at the end of the system software installation. If you installed compilers earlier in the system software installation, you must re-install them after all other system software is installed.

## Installing the Native Compilers

---

**Installation Time:**

Approximately 45 minutes.

**Installation Media:**

One 0.25-inch QIC 150 cartridge tape labelled "Paragon<sup>TM</sup> Fortran Compiler Release 4.5 Native, Sun4, and Solaris-Hosted (313013-001)."

The tape contains an installation **tar** file for the native compiler and another for the cross-compiler for Sun-4 workstations. The files also contain the examples and PostScript copies of the manuals.

---

These instructions assume you are reading the tape on the Paragon diagnostic station, which is running SCO Unix. The tapes were written on a diagnostic station, therefore the *tar* format and the physical characteristics of the tape are compatible with that system. You can, if you prefer, try reading the tape on some other networked system, such as a workstation. If that system does not successfully read the files from the tape, you are advised to read it on the diagnostic station.

---

1. Log in to the diagnostic station as *root*.
2. Copy the installation **tar** files from the release tape into */u/tmp* on the diagnostic station. Each compressed **tar** file is about 4M bytes. After installation each compiler requires about 8M bytes.

A. First make */u/tmp* your working directory. Then perform the following steps.

```
DS# cd /u/tmp
```

B. Insert the release tape into the cartridge tape drive on the diagnostic station.

C. Issue the command,

```
DS# tar xvf /dev/rStp0 native_install nat_ftn.tar.Z
      if77.doc.tar.Z
```

D. After the file has been copied, remove the tape from the cartridge tape drive.

3. Log in to the Paragon system as *root*.
4. If you have already installed the native C compilers and */tmp/native\_install* still exists on your system, you can proceed to step 5.

Establish an **ftp** connection with the diagnostic station and transfer the following file:

*native\_install* This file copies the compiler and documentation files from the diagnostic station and installs them in / or an alternate directory.

On the Paragon system, issue the following commands:

```
# cd /tmp ✓
# ftp diagnostic station IP address
ftp> cd /u/tmp
ftp> get native_install
ftp> bye
# chmod 544 native_install
```

5. Execute the installation script. ✓

```
# cd /
# /tmp/native_install f77
```



The following is displayed. The distribution information is read from */etc/defaults/install* if the file exists:

```

=====
Native Compiler Installation
=====

Install Fortran? [y/n]:          y
Root directory for compiler installation [path]: /
Distribution Node:                myhost
Distribution IP Addr:             myhost_ip_addr
Distribution Path:                /my_default_path
Is this correct? [y/n]:

```

To change any of these values, answer "n" to the "Is this correct?" prompt and enter the desired value when prompted to do so. If you enter <CR> at a prompt, the value is not changed. If you change the value of the root installation directory, and the directory does not exist, you are asked if you want to create it. When you are satisfied with all the values displayed, enter "y" in response to the "Is this correct" prompt.

The files are copied to the Paragon system and installed. The following is an example of the output seen when installing in the directory */*.

```
Username for FTP'ing files from roadkill: [anonymous] myname
```

*FTP output from file transfers*

```

221 Goodbye.
Uncompressing nat_ftn.tar.Z...
Uncompressing if77.doc.tar.Z...
Installing Native Fortran compiler...
Native Fortran compiler has been installed
Installing Fortran manual pages...

Installation complete ✓

```

6. Verify that your path is set correctly.

If the root directory for the install was not */*, set *PARAGON\_XDEV* to be the root directory you entered, and add *\$PARAGON\_XDEV/usr/bin* to the beginning of your execution path. You must also add *\$PARAGON\_XDEV/usr/man* to the beginning of your *MANPATH* environment variable to access the R4.5 manual pages.

The following should display when you use the compiler **-VV** switch. If it does not, examine your *PATH* environment variable and make any needed corrections.

```
#if77 -VV ✓
```

```
if77/Paragon Paragon Version 4.5  
Copyright 1994, Intel Corporation and The Portland Group Inc.  
All Rights Reserved
```

```
View $PARAGON_XDEV/usr/share/IFC/release_notes/if77_4.5_release_notes.ps  
for a list of new features for Release 4.5
```

7. Execute the installation verification test.

```
# cd root_installation_directory/usr/testinstall ✓  
# ./testinstall_f  
Installation successful
```

8. Remove the *testinstall* directory.

```
# cd ..  
# rm -rf testinstall
```

## Installing the Cross-Development Compilers

---

<b>Installation Time:</b>	Approximately 45 minutes.
<b>Installation Media:</b>	One 0.25-inch QIC 150 cartridge tape labelled "Paragon™ Fortran Compiler Release 4.5 Native, Sun4, and Solaris-Hosted (313013-001)."  The tape contains an installation <b>tar</b> file for the native compiler and another for the cross-compiler. The files also contain the online manual pages, examples, and PostScript copies of the manuals.
<b>Installation Media (SGI):</b>	One 0.25-inch QIC 150 cartridge tape labelled "Paragon™ Fortran Compiler Release 4.5 Silicon Graphics Hosted (313017-001)."

---

The cross-development tools and compilers are installed by reading in a set of **tar** files from the installation tape onto the diagnostic station. You then **ftp** one or more compressed **tar** files to your workstation or workstation server, where you untar them. Do not install the cross-development compilers on the Paragon.

1. Log in to the diagnostic station as *root*.
2. Copy the installation **tar** files from the release tape into */u/tmp* on the diagnostic station. Each compressed **tar** file is about 4M bytes. After installation each compiler requires about 8M bytes. The total space for all compiler-related files can be as much as 51M bytes.
  - A. First, make */u/tmp* your working directory, Then perform the following steps.

```
DS# cd /u/tmp
```
  - B. Insert the release tape into the cartridge tape drive on the diagnostic station.
  - C. Extract the compressed tar files and installation script. The installation script and documentation files are the same for each host and can be extracted only once.

If you are copying the cross-compiler for a Sun4 workstation, issue the command:

```
DS# tar xvf /dev/rStp0 cross_install sun_ftn.tar.Z
    if77.doc.tar.Z
```

If you are copying the cross-compiler for an SGI workstation, issue the command:

```
DS# tar xvf /dev/rStp0 cross_install sgi_ftn.tar.Z
    if77.doc.tar.Z
```

If you are copying the cross-compiler for a Sun4/Solaris workstation, issue the command:

```
DS# tar xvf /dev/rStp0 cross_install sol_ftn.tar.Z
    if77.doc.tar.Z
```

D. After the files have been copied, remove the tape from the cartridge tape drive.

3. If you have installed new system software, you must copy the system libraries and include files to the cross-development environment. If you have not installed new system software, proceed to step 4.

Log in to the Paragon system as *root*. Then:

```
# cd /tmp
# /usr/bin/mksysfiles
# exit
```

This step may take up to 30 minutes to complete. You will also see the following message. This message should be ignored.

```
rm: sysfiles.tar No such file or directory
```

4. If you have already installed the C cross compiler(s) and */tmp/cross\_install* still exists on your system, you can proceed to step 5.

Establish an **ftp** connection with the diagnostic station and transfer the following file:

*cross\_install* This file copies the compiler(s) and documentation files from the diagnostic station and installs them.

On your workstation, issue the following commands:

```
CROSS# cd /tmp
CROSS# ftp diagnostic station IP address
ftp> cd /u/tmp
ftp> get cross_install
ftp> bye
CROSS# chmod 544 cross_install
```

5. If you do not need to create a new directory for the R4.5 compilers, you can proceed to step 6.

```
CROSS# mkdir directory
```

6. Make the directory in which the cross compiler(s) will be installed your current working directory and execute the installation script.

```
CROSS# cd directory
CROSS# /tmp/cross_install f77
```

The following is displayed.

```
=====
                          Cross Compiler Installation
=====

Install Fortran? [y/n]:      Y
Install Sun4? [y/n]:        Y
Install Solaris? [y/n]:     Y
Install SGI? [y/n]:         Y
Root directory for compiler installation [path]: $PARAGON_XDEV
Location for compressed tar files [path]:      /tmp
Distribution host name:      unknown
Distribution host user name for ftp:          anonymous
Distribution Path:          /u/tmp
Install system files? [y/n]: n
Create links for system files? [y/n]:       n
Is this correct? [y/n]:
```

If you respond with “y” to the “Install system files” prompt, you are asked for the following information:

```
Enter name of paragon system where sysfiles.tar.Z was created:
Enter user name for ftp from your_system: [anonymous]
Enter path of sysfiles.tar on your_system: [/tmp]
```

If you respond with "y" to the "Create links for system files" prompt, you are asked for the following information:

Enter root directory of actual files:

To change any of these values, answer "n" to the "Is this correct?" prompt and enter the desired value when prompted to do so. If you enter <CR> at a prompt, the value is not changed. When you are satisfied with all the values displayed, enter "y" in response to the "Is this correct" prompt.

All of the compressed tar files needed for your installation selections are copied to the location you specified. Each file is uncompressed and installed, and the uncompressed tar files are deleted. If any of the uncompress or tar commands fail, the installation is aborted. The most likely cause for a failure is lack of disk space. If this occurs, you may need to install one compiler at a time.

In the following example, the user installs the Sun4 compiler, the Sun4/Solaris compiler, the SGI compiler, and creates links for the system libraries and header files.

CROSS# /tmp/cross\_install f77 ✓

```
=====
                        Cross Compiler Installation
=====
```

```
Install Fortran? [y/n]:          Y
Install Sun4? [y/n]:            Y
Install Solaris? [y/n]:         Y
Install SGI? [y/n]:             Y
Root directory for compiler installation [path]: /vol/scratch/install
Location for compressed tar files [path]:       /vol/scratch/tmp
Distribution host name:          fred
Distribution host user name for ftp:            root
Distribution Path:               /u/tmp
Install system files? [y/n]:     n
Create links for system files? [y/n]:         Y
  Create links to:                /vol/crossdev/R1.2
Is this correct? [y/n]:         Y
```

```
Connected to fred.
220 roadkill FTP server (SunOS 4.1) ready.
331 Password required for root.
Password:
230 User root logged in.
200 Type set to I.
Local directory now /vol/scratch/temp
200 PORT command successful.
```

...

*ftp files from distribution system*

...

221 Goodbye.  
Uncompressing if77.doc.tar.Z...  
Installing Fortran manual pages...  
Uncompressing sun\_ftn.tar.Z...  
Installing Sun Fortran compiler...  
Uncompressing sol\_ftn.tar.Z...  
Installing Sun4/Solaris Fortran compiler...  
Uncompressing sgi\_ftn.tar.Z...  
Installing SGI Fortran compiler...  
Creating links for header files and libraries...

Installation complete ✓  
CROSS#

7. Verify that your path is set correctly.

Set *PARAGON\_XDEV* to be the root directory you entered, and add *\$PARAGON\_XDEV/paragon/bin."arch"* to the beginning of your execution path. For example, on Sun4/Solaris systems you would add *\$PARAGON\_XDEV/paragon/bin.solaris*. You must also add *\$PARAGON\_XDEV/paragon/man* to the beginning of your *MANPATH* environment variable to access the R4.5 manual pages.

The following should display when you use the compiler -VV switch. If it does not, examine your *PATH* environment variable and make any needed corrections.

#if77 -VV ✓

if77/Paragon "host" Version 4.5  
Copyright 1994, Intel Corporation and The Portland Group Inc.  
All Rights Reserved

View *\$PARAGON\_XDEV/paragon/release\_notes/if77\_4.5\_release\_notes.ps*  
for a list of new features for Release 4.5

- ✓ 8. Execute the installation verification test. The *testinstall\_c* script requires the name of the paragon system where the test will be executed as an argument. You must be able to execute **rcp** and **rsh** commands on the Paragon system you specify.

```
# cd $PARAGON_XDEV/paragon/testinstall ✓  
# ./testinstall_f paragon_system  
Installation successful
```

9. Remove the *testinstall* directory.

```
# cd ..  
# rm -rf testinstall
```