Software Development Products

Product Errata

Intel® Fortran Compiler for Linux* and Windows*

10th February 2003

Number of entries − 30

DISCLAIMER

Linux*

Windows*

21888	The ifl compiler displays syntax error if a backslash is used in Format statement
23975	Problem with I0 edit descriptor and INTEGER(8) values
24663	Cannot output large real or integer values from a NAMELIST
24678	EDB does not recognize executable program path names containing blank characters
25488	Incompatibility with Compaq* Visual Fortran: () optional in function declaration
26140	Character constant with KIND not accepted in format field of WRITE
26766	Compliance with IEEE 854
28026	Unresolved external symbol _isnanf
29518	Apps with source files with names beginning with 'etrip' may fail at runtime

Linux*

Reference #	Product	Version	Operating System	Title	Last Update
22214	Intel(R) Fortran Compiler for Linux*	6.0, 7.0	Red Hat 6.2	Fortran compiler treats 'if_' as a keyword rather than a label.	27−Sep−02
Symptom	The Intel(R) Fortran Compiler for Linux* treats a label starting with if_ as a keyword rather than a label. In the following example: if_i: if(i.eq.1)then i=i+1 else if_i i=i−1 endif if_i the compiler treats the "else" as an "else if" rather than a label. It compiles if another label name is chosen.				
Current Status/Solution This is a known issue that may be resolved in a future product release.					

Reference #	Product	Version	Operating System	Title	Last Update
21708	Intel(R) Fortran Compiler for Linux*	6.0, 7.0	Red Hat* 7.1	Documentation files installed with execute permissions enabled	25−Nov−02
Symptom	Most of the compiler documentation is installed with execute permissions for user, group, and world. For example:				

```
$ ls −l /opt/intel/compiler60/docs
total 12344
−rwxr−xr−x 1 root root 282345 Jan 4 2002 asm_lan.pdf
−rwxr−xr−x 1 root root 274778 Feb 27 2002 asm_ug.pdf
...
```

Current Status/Solution

This is a known issue that may be resolved in a future product release.

Reference #	Product	Version	Operating System	Title	Last Update
24590	Intel(R) Fortran Compiler for Linux*	6.0	Red Hat* 7.1	Use of dummy argument in subscript inhibits auto−parallelization	3−Oct−02

Symptom

Use of dummy argument in subscript inhibits auto−parallelization. Use of the dummy argument ic, as shown below, inhibits loop auto−parallelization.

```
subroutine sub(b,n,ic)
dimension b(n)
do i=1,n
b(i+ic)=0
enddo
end
```

Current Status/Solution

This problem has been resolved in the Intel(R) Fortran Compiler 7.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.

As a workaround the subscript can be passed in a common block.

Reference #	Product	Version	Operating System	Title	Last Update
24814	Intel(R) Fortran Compiler for Linux*	6.0, 7.0	Red Hat* 7.1	LDB cannot distinguish source files with the same name	25−Nov−02

Symptom

If an application contains two files of the same name that are in different subdirectories and built into different libraries, LDB cannot distinguish between the two files and warns of this condition as follows::

```
Can not distinguish between the following files:
/home/test/libdir/foof.f
```

<table>
<tr><td></td><td>/home/test/otherdir/foof.f</td></tr>
<tr><td colspan="2"><b>Current Status/Solution</b>

This is a known issue that may be resolved in a future product release.</td></tr>
</table>

<table>
<tr><th>Reference #</th><th>Product</th><th>Version</th><th>Operating System</th><th>Title</th><th>Last Update</th></tr>
<tr><td>26129</td><td>Intel(R) Fortran Compiler for Linux*</td><td>6.0</td><td>Red Hat* 7.2</td><td>Documentation for lnblnk() function is missing</td><td>27−Sep−02</td></tr>
<tr><td><b>Symptom</b></td><td colspan="5">The lnblnk() function is missing from the Intel Fortran Libraries Reference document. It is included in the compiler portability library.</td></tr>
<tr><td colspan="6"><b>Current Status/Solution</b>

This problem has been resolved in the Intel(R) Fortran Compiler 7.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.</td></tr>
</table>

<table>
<tr><th>Reference #</th><th>Product</th><th>Version</th><th>Operating System</th><th>Title</th><th>Last Update</th></tr>
<tr><td>26080</td><td>Intel(R) Fortran Compiler for Linux*</td><td>6.0,7.0</td><td>Red Hat* 7.1</td><td>Erroneous re−type warning for Cray−style pointer dummy argument</td><td>26−Nov−02</td></tr>
<tr><td><b>Symptom</b></td><td colspan="5">Compilation of the following subroutine causes issuance of an erroneous warning level message regarding Cray−style pointer variable re−typing as shown below:

subroutine foo (ib)
implicit none
real b
pointer (ib, b)
end

Warning at compilation:
pointer (ib, b)

Warning 114: Pointer variable has already been declared − retyped as INTEGER</td></tr>
<tr><td colspan="6"><b>Current Status/Solution</b>

This is a known issue and may resolved in a future release. Additional information on Cray−style pointer support including type handling is available in the Intel ® Fortran Programmer's Reference manual.</td></tr>
</table>

Reference #	Product	Version	Operating System	Title	Last Update

26550	Intel(R) Fortran Compiler for Linux*	6.0	Red Hat* 7.2	Flush of unwritten file causes 'FLUSH FAILED'	26−Nov−02

Symptom

Use of FLUSH on an un−written file generates an incorrect failure. Below is a simple test and output.

```
program flushtest
open(10,FILE="output1")
open(11,FILE="output2")
write(10,*) "Hello!"
print *, "Flushing file written to"
call flush(10)
print *, "Flushing file not written to"
call flush(11)
stop
end
```

Program output:

```
Flushing file written to
Flushing file not written to
FLUSH−−FAILED:: Invalid argument
```

Current Status/Solution

This problem has been resolved in a product update with package ID l_fc_p_6.0.1.304 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Reference #	Product	Version	Operating System	Title	Last Update
26358	Intel(R) Fortran Compiler for Linux*	6.0	Red Hat* 7.1	USE IFLPORT results in Error FCE22 : Module IFLPORT USEd by program in work.pc not found	19−Dec−02

Symptom

USE IFLPORT results in Error FCE22 : Module IFLPORT USEd by program in work.pc not found
Compilation of the sample code below fails this error.

```
USE IFLPORT
CHARACTER(LEN=30) DIRECTORY
INTEGER(4) ISTAT
ISTAT = GETCWD (DIRECTORY)
IF (ISTAT == 0) PRINT *, 'Current directory is', DIRECTORY
END
```

Current Status/Solution

This problem has been resolved in the Intel(R) Fortran Compiler 7.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.

As workaround create your own work.pcl file in your local directory, containing the lines
work.pc
/opt/intel/compiler60/ia32/include/work.pc

Reference #	Product	Version	Operating System	Title	Last Update
26609	Intel(R) Fortran Compiler for Linux*	7.0	Red Hat 7.2	The −openmp switch may sometimes inhibit some −O3 optimizations resulting in lower performance	27−Jan−03
Symptom	The −openmp compiler switch may inhibit certain high level optimizations such as "Loop interchange" that would otherwise be performed if −O3 were specified in the absence of −openmp. You can get a report of high level optimizations performed using the following command line examples: efc −O3 −opt_report test.f90 // without the −openmp switch efc −O3 −openmp −opt_report test.f90 // with the −openmp switch				
Current Status/Solution This is a known issue that may be resolved in a future product release.					

Reference #	Product	Version	Operating System	Title	Last Update
30073	Intel(R) Fortran Compiler for Linux*	6.0	SuSe*	ldb cannot resolve local variable	26−Nov−02
Symptom	ldb may be unable to disambiguate between local variables in different contexts that share the same name. If the sample program below is compiled with ifc using the −g switch (produce symbolic debug info), ldb can resolve local variable 'x' from within the context of TEST, however, it cannot resolve the name 'x' from within the context of Subroutine f1. For example, the ldb print command returns the following error from within the context of f1: (ldb) print x ldb error: Name "x" is ambiguous (possibly bad debug information) PROGRAM TEST IMPLICIT NONE INTEGER :: x x=1				

```
CALL f1(x)
STOP

CONTAINS
SUBROUTINE f1(x)
IMPLICIT NONE
INTEGER :: x
PRINT *,x
END SUBROUTINE f1

END PROGRAM TEST
```

Current Status/Solution

This is a known issue that may be resolved in a future product release.

Reference #	Product	Version	Operating System	Title	Last Update
26758	Intel(R) Fortran Compiler for Linux*	6.0,7.0	Red Hat* 7.1	minval/maxval may produce incorrect results when a pointer is used as an argument	26−Nov−02
Symptom	Compiler may produce an incorrect result of the minval (or maxval) function when a pointer is used as an argument.				

Current Status/Solution

This problem has been resolved in the Intel(R) Fortran Compiler 7.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.

Reference #	Product	Version	Operating System	Title	Last Update
26970	Intel(R) Fortran Compiler for Linux*	7.0	Red Hat* 7.1	Compiling a module with −parallel and −openmp may cause an internal compiler error	29−Oct−02
Symptom	When compling a module with the −parallel and −openmp options using the Itanium(R) compiler, the compiler may report the following internal error: Internal Error: ECG_MODULE_ID_TRANS_5769				

Current Status/Solution

This problem has been resolved in Intel(R) Fortran Compiler 7.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Reference #	Product	Version	Operating System	Title	Last Update
27134	Intel(R) Fortran Compiler for Linux*	6.0	Red Hat* 7.1	MOD operation produces wrong result	5−Dec−02
Symptom	The statement "print*,mod(1.0,0.2)" produces a value of −1.490116E−08 which is incorrect: Since 0.2 in machine form is not exactly 0.2 (it is 3E4CCCCD, and in digital form it is 0.200000002980), we cannot get exact 0.0 as the result. The sign of result should be the same as the first argument. Therefore the correct result should be 0.2000000 assuming the default format for real*4. If the format of output is changed to put more digits (e.g. f20.12), the result should be 0.199999988079.				
Current Status/Solution This problem has been resolved in the Intel(R) Fortran Compiler 7.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.					

Reference #	Product	Version	Operating System	Title	Last Update
26420	Intel(R) Fortran Compiler for Linux*		Red Hat* 7.1	gprof: gmon.out file is missing call−graph data	30−Oct−02
Symptom	The gmon.out produced by the Itanium(R) compiler with −p option crashes gprof when trying to evaluate the profiling data. The following message is displayed when running gprof: gprof: gmon.out file is missing call−graph data				
Current Status/Solution This problem is currently under investigation and may be resolved in a future product release. As a workaround use −g option along with the −p option.					

Reference #	Product	Version	Operating System	Title	Last Update
28138	Intel(R) Fortran Compiler for Linux*	6.0,7.0	Debian	ifc: selected_real_kind returns different value when used in parameter declaration.	30−Oct−02
Symptom	selected_real_kind(33, 4932) returns 16 when called from a program and −2 when called on the following line: integer, parameter :: k2 = selected_real_kind(33, 4932) for a range value of 4931 both return 16 while for a range of 4933, the "normal" call returns −3 while the parameter call return −2.				

Current Status/Solution

This problem has been resolved in a product update with package ID l_fc_pu_7.0.076 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Reference #	Product	Version	Operating System	Title	Last Update
28685	Intel(R) Fortran Compiler for Linux*	6.0	Red Hat* 7.2	Files opened with REWIND and F_UFMTENDIAN are handled in Little endian format	28−Oct−02
Symptom	If a file is opened using REWIND, then the F_UFMTENDIAN environment variable is not checked. In this case the file is treated as having little endian format.				

Current Status/Solution

This problem has been resolved in a product update with package ID l_fc_p_7.0.064 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Reference #	Product	Version	Operating System	Title	Last Update
28719	Intel(R) Fortran Compiler for Linux*	6.0	Debian	Problem with line length > 132	27−Sep−02
Symptom	The Fortran compiler truncates all lines longer than 132 characters for free format source (using the −FR switch) without any warning or error. This may lead to incorrect code.				

Current Status/Solution

This problem has been resolved in the Intel(R) Fortran Compiler 7.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Reference #	Product	Version	Operating System	Title	Last Update
24114	Intel(R) Fortran Compiler for Linux*	6.0,7.0	Red Hat* 7.2	The maximum array size is limited to (2**31−1) bytes on a IA32 machine	27−Nov−02
Symptom	The maximum array size seems to be limited to (2**31−1) bytes on IA32. Is there any way to exceed this limitation? I presently get this error message: "In program unit MAIN the size of				

	array A1 exceeds the implementation limit (2**31−1)"				
Current Status/Solution This is a known issue that may be resolved in a future product release. As a workaround, if it is possible to divide the work of your application into processes, MPI can be used as each process will have its own 2 gigabyte address space.					

Reference #	Product	Version	Operating System	Title	Last Update
29087	Intel(R) Fortran Compiler for Linux*	6.0	Caldera*	The compiler does not compile a file with 250 or more equivalence statements	29−Oct−02
Symptom	Compiling a file with 250 or more equivalence statements results in *Compiler Internal Error*				
Current Status/Solution This problem has been resolved in a product update with package ID l_fc_pu_6.0.1.311 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support					

Reference #	Product	Version	Operating System	Title	Last Update
29676	Intel(R) Fortran Compiler for Linux*	7.0	Red Hat* 7.0	Compilation hangs at all optimization levels when building the POP software package	23−Nov−02
Symptom	Compilation hangs at all optimization levels when compiling the vertical_mix.f module which is part of the Parallel Ocean Program (POP) software package.				
Current Status/Solution This is a known issue that may be resolved in a future product release.					

Reference #	Product	Version	Operating System	Title	Last Update
30098	Intel(R) Fortran Compiler for Linux*	6.0,7.0	Other (specify below)	ieee_flags returns blanks in "out" parameter	24−Jan−03
Symptom	The ieee_flags portability intrinsic function may return a blank value in the "out" parameter string rather than the expected string. For example, status=ieee_flags('get', 'exception', in, out) write(*,*) out should print out 'division' to stdout when a floating point divide by zero exception is caught.				

<table>
<tr><td></td><td>Instead, you will only see blanks.</td></tr>
<tr><td colspan="2"><b>Current Status/Solution</b>

This is a known issue that may be resolved in a future product release.</td></tr>
</table>

Windows*

<table>
<tr><th>Reference #</th><th>Product</th><th>Version</th><th>Operating System</th><th>Title</th><th>Last Update</th></tr>
<tr><td>21888</td><td>Intel(R) Fortran Compiler for Windows*</td><td>6.0, 7.0</td><td>Windows* 2000 Server</td><td>The ifl compiler displays syntax error if a backslash is used in Format statement</td><td>30−Oct−02</td></tr>
<tr><td><b>Symptom</b></td><td colspan="5">If the following program is compiled with
ifl −c test.f
the first print statement gives a warning that / as an escape character is an extension to Fortran95, but the Format statement gives an error. If it is compiled with
ifl −c −nbs test.f
the first print statement compiles correctly, but format statement still gives an error.

Program test
PRINT '(" Hello ",\, " World!")'
PRINT 10
10 FORMAT(' Hello',\,' World!')
END</td></tr>
<tr><td colspan="6"><b>Current Status/Solution</b>

This is a known issue that may be resolved in a future product release.</td></tr>
</table>

<table>
<tr><th>Reference #</th><th>Product</th><th>Version</th><th>Operating System</th><th>Title</th><th>Last Update</th></tr>
<tr><td>23975</td><td>Intel(R) Fortran Compiler for Windows*</td><td>6.0,7.0</td><td>Windows* Me</td><td>Problem with I0 edit descriptor and INTEGER(8) values</td><td>28−Oct−02</td></tr>
<tr><td><b>Symptom</b></td><td colspan="5">The Fortran Compiler does not print INTEGER(8) values when using the I0 descriptor. The program below will not print any values (compile with −i8 −w95).

integer(8) k
k = 123456789123456789_8
write (6,100) k
100 format (1x,i0)
end</td></tr>
<tr><td colspan="6"><b>Current Status/Solution</b>

This problem has been resolved in the Intel(R) Fortran Compiler 7.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered</td></tr>
</table>

user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support .

Reference #	Product	Version	Operating System	Title	Last Update
24663	Intel(R) Fortran Compiler for Windows*	5.0.1, 6.0, 7.0	Windows* Me	Cannot output large real or integer values from a NAMELIST	25−Nov−02
Symptom	NAMELIST output can fail with an address error with large REAL(16) or INTEGER(8) values. ! Example to show the problem with NAMELIST real(16) x integer(8) q namelist/nam1/x,q x=1234567891234e1000_16 q=12345678901234_8 write(*,nml=nam1) end When the above test case is executed, an address error is reported.				
Current Status/Solution This is a known issue that may be resolved in a future product release.					

Reference #	Product	Version	Operating System	Title	Last Update
24678	Intel(R) Fortran Compiler for Windows*	5.0, 6.0, 7.0	Windows* 2000 Professional	EDB does not recognize executable program path names containing blank characters	27−Jan−03
Symptom	When invoked via the EDB button on the Microsoft* Visual* C++ 6.0 IDE toolbar, EDB cannot find the executable file to be loaded when any part of the path name contains one or more blank characters. For example the executable file /mydirectory/test cases/test.exe cannot be loaded because in the above path, the directory "test cases" contains a blank character.				
Current Status/Solution This is a known issue that may be resolved in a future product release. As a workaround you can load the executable file by using the "Load" option from the "File" menu in EDB.					

<table>
<tr><th>Reference #</th><th>Product</th><th>Version</th><th>Operating System</th><th>Title</th><th>Last Update</th></tr>
<tr><td>25488</td><td>Intel(R) Fortran Compiler for Windows*</td><td>5.0, 6.0</td><td>Windows* 2000 Professional</td><td>Incompatibility with Compaq* Visual Fortran: () optional in function declaration</td><td>6−Dec−02</td></tr>
<tr><td><b>Symptom</b></td><td colspan="5">The following simple function compiles with the Compaq* Visual Fortran compiler but suffers a compilation error (as shown) when compiled with the Intel v6.0 Fortran compiler:

FUNCTION TESTSH
TESTSH=0
END

ifl −c testsh.f
external function TESTSH

FUNCTION TESTSH
^
q.f(1): Error 7 : incomplete statement

1 Error
compilation aborted for q.f (code 1)</td></tr>
<tr><td colspan="6"><b>Current Status/Solution</b>

This problem has been resolved in a product update with package ID W_FC_P_7.0.076 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

As a workaround, you can add the empty parenthesis () to the function declaration as follows:

FUNCTION TESTSH()
TESTSH=0
END</td></tr>
</table>

<table>
<tr><th>Reference #</th><th>Product</th><th>Version</th><th>Operating System</th><th>Title</th><th>Last Update</th></tr>
<tr><td>26140</td><td>Intel(R) Fortran Compiler for Windows*</td><td>6.0,7.0</td><td>Windows* Me</td><td>Character constant with KIND not accepted in format field of WRITE</td><td>28−Oct−02</td></tr>
<tr><td><b>Symptom</b></td><td colspan="5">The compiler doesn't accept a character constant with a kind in the format field of a WRITE statement. A sample test case is shown below:

program testcase
implicit none
write(*,1_'(i2)')3
end</td></tr>
</table>

Current Status/Solution

This is a known issue that may be resolved in a future product release.

Reference #	Product	Version	Operating System	Title	Last Update
26766	Intel(R) Fortran Compiler for Windows*	6.0	Windows* 2000 Professional	Compliance with IEEE 854	29−Oct−02

Symptom

Some code produced by the Intel Fortran Compiler 6.0 using the default optimization (/O2) does not comply with IEEE standard 854 (Floating Point Arithmetic). Section 5.7 and Table 3 of the standard specify the results of comparisons involving unordered quantities such as NaN. Of the 12 possible comparisons, code generated by this compiler gets the wrong answer in 7 cases.

Current Status/Solution

The IEEE 854 conforming behavior is produced if you compile without optimization (/Od). If you compile with optimization at any level, then you must supply the /Qprec switch to get the conforming behavior. /Op will also give the conforming behavior, but the performance impact is much greater than /Qprec, which has a negligible performance impact.

This problem has been resolved in a product update with package ID W_FC_P_6.0.1.306 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Reference #	Product	Version	Operating System	Title	Last Update
28026	Intel(R) Fortran Compiler for Windows*	6.0	Windows* 2000 Professional	Unresolved external symbol _isnanf	30−Oct−02

Symptom

When isnan function is used, the linker reports that it cannot find the _isnanf symbol.

```
program developTest
write(*,*) isNAN(3.4)
end program
```

```
error LNK2001: unresolved external symbol _isnanf
```

Current Status/Solution

This problem has been resolved in the Intel(R) Fortran Compiler 7.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.

Reference #	Product	Version	Operating System	Title	Last Update
29518	Intel(R) Fortran Compiler for Windows*	6.0	Windows* 2000 Professional	Apps with source files with names beginning with 'etrip' may fail at runtime	21−Nov−02
Symptom	Source file names beginning with 'etrip' (for example, etripab.f) can be mis−interpreted by the compiler and lead to unintended uses of the EBP register which may cause unexpected application runtime failures when compiling at optimization level /O1 or above.				
Current Status/Solution This problem has been resolved. The solution will be available in a future product release. As a workaround, use the /Oy− option to prevent the EBP register from being used in optimizations.					

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL’S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.
Intel may make changes to specifications and product descriptions at any time, without notice.Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.
Intel, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.