Software Development Products

Product Errata

Intel® Fortran Compiler for Linux* and Windows*

17th March 2003

Number of entries − 51

DISCLAIMER

Linux*

22214	Fortran compiler treats 'if_' as a keyword rather than a label.
21708	Documentation files installed with execute permissions enabled
14358	Compiler does not support vectorization of complex data types
24833	Nested modules can cause slow compiles
24201	Sum reduction is not parallelized
24425	Initializing OpenMP code within pthreads causes a segmentation fault
24590	Use of dummy argument in subscript inhibits auto−parallelization
24814	LDB cannot distinguish source files with the same name
24797	OpenMP program aborts when dummy array argument is in private list.
26129	Documentation for lnblnk() function is missing
26080	Erroneous re−type warning for Cray−style pointer dummy argument
26550	Flush of unwritten file causes 'FLUSH FAILED'
26358	USE IFLPORT results in Error FCE22 : Module IFLPORT USEd by program in work.pc not found
26609	The −openmp switch may inhibit some −O3 optimizations resulting in lower performance
30073	ldb cannot resolve local variable
26758	minval/maxval may produce incorrect results when a pointer is used as an argument
26970	Compiling a module with −parallel and −openmp may cause an internal compiler error
27134	MOD operation produces wrong result
27459	Representation of Infinity and NaN
26420	gprof: gmon.out file is missing call−graph data

28138	ifc: selected_real_kind returns different value when used in parameter declaration.
28685	Files opened with REWIND and F_UFMTENDIAN are handled in Little endian format
28719	Problem with line lengths that exceed 132 characters
24114	The maximum array size is limited to (2**31−1) bytes on a IA32 machine
29087	The compiler does not compile a file with 250 or more equivalence statements
29260	Compilation hangs at all optimization levels when building the POP software package
31312	rewind doesn't force flushing of a buffer
29654	Function erf() not defined in Fortran library for Itanium(R) compiler
30098	ieee_flags returns blanks in "out" parameter
29665	Module 328.fma3d in SpecOMP2001fp test suite may get into an infinite loop with −O3 −openmp
31310	Using idb under GNU Emacs v21 fails with error
30702	Use of OPENMP ALLOCATE statement fails in a PRIVATE clause
31200	OpenMP code using many OMP SECTIONS causes long compile times
31677	Using −prof_use −O3 switches may produce incorrect program output

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
26417	prof_gen/prof_use may not always generate the most efficient code
26766	Compliance with IEEE 854
25191	Call to function that outputs to internal file from within formatted write statement causes crash
28026	Unresolved external symbol _isnanf
28447	Support for /Qdoubletemps
29518	Apps with source files with names beginning with 'etrip' may fail at runtime
31303	Installer changes paths in Microsoft* Visual C++* 6.0 IDE even when build tools not installed
31043	Fortran project files rebuild even if target is up to date
30924	Unable to compile individual Fortran file in Microsoft* Visual Studio .NET* environment
31492	C preprocessor "stringization" and "merging" operators not recognized by Fortran preprocessor

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−2002
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−2002
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

14358	Intel(R) Fortran Compiler for Linux*	6.0, 7.0	Red Hat* 7.1	Compiler does not support vectorization of complex data types	28−Oct−2002

Symptom	The Intel Fortran Compiler will not vectorize complex data types. The following test routine fails to vectorize the complex data structure, but does vectorize if the data types are real.
	
	$ cat p.f90
	subroutine test(x,y,z)
	complex, dimension(1:100)::x,y,z
	
	x = y*z
	
	return
	end
	
	$ ifc −c −xW p.f90 −vec_report3
	external subroutine PARVEC
	p.f90(4) : (col. 0) remark: loop was not vectorized: data type unsupported on given target architecture.
	
	7 Lines Compiled				

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**
24833	Intel(R) Fortran Compiler for Linux*	6.0, 7.0	Red Hat* 7.1	Nested modules can cause slow compiles	20−Dec−2002
Symptom	Nested modules can degrade compile−time performance. The complexity of the nested modules can determine when the compile−time performance degradation is observed. For one application, at a depth greater than 15 modules compile−time performance degraded. A module that should take 15−20 minutes to compile took 12 hours to compile.				
Current Status/Solution This is a known issue that may be resolved in a future product release. As a workaround reduce the depth of nested modules. Here are some suggestions to accomplish this task: Don't put USE statements for more modules than are needed; don't USE modules via multiple paths; combine smaller "header file" modules into a single, larger module.					

Reference #	**Product**	**Version**	**Operating System**	**Title**	**Last Update**
24201	Intel(R) Fortran Compiler for Linux*	6.0	Red Hat* 7.1	Sum reduction is not parallelized	27−Sep−2002

Symptom	The compiler does not auto−parallelize summation reduction. The sample code shown below demonstrates this issue.
	
	subroutine sum(a,n,s)
	dimension a(n)
	do i=1,n
	s =s +a(i)
	enddo
	end
	
	$ efc −parallel −par_threshold0 −par_report3 −c test.f
	external subroutine SUM
	procedure: sum
	serial loop: line 3
	output data dependence assumed from line 4 to line 4, due to "s"
	flow data dependence assumed from line 4 to line 4, due to "s"
	anti data dependence assumed from line 4 to line 4, due to "s"
	
	6 Lines Compiled

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
24425	Intel(R) Fortran Compiler for Linux*	6.0	SuSE Linux 7.3 (i386) − Kernel	Initializing OpenMP code within pthreads causes a segmentation fault	1−Nov−2002
Symptom	If code to initialize OpenMP declarations is called from a thread created by pthreads a segmentation fault will result. If the OpenMP initialization is done before the threads are spawned, the threads can call routines containing OpenMP declarations.				

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
24590	Intel(R) Fortran Compiler for Linux*	6.0	Red Hat* 7.1	Use of dummy argument in subscript inhibits auto−parallelization	3−Oct−2002

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−2002
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 /home/test/otherdir/foof.f				

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
24797	Intel(R) Fortran Compiler for Linux*	6.0	Red Hat* 7.1	OpenMP program aborts when dummy array argument is in private list.	19−Dec−2002
Symptom	Use of a subroutine dummy array argument in the private list of a OpenMP parallel directive may cause run−time Address error fault.				

	program TEST real testarray(50) call testsub(testarray) end subroutine testsub(testarray) integer i real testarray(50) !$omp parallel do private(testarray) do i=1,100 testarray(i)=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.

Reference #	Product	Version	Operating System	Title	Last Update
26129	Intel(R) Fortran Compiler for Linux*	6.0	Red Hat* 7.2	Documentation for lnblnk() function is missing	27−Sep−2002
Symptom	The lnblnk() function is missing from the Intel Fortran Libraries Reference document. It is included in the compiler portability library.				

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
26080	Intel(R) Fortran Compiler for Linux*	6.0,7.0	Red Hat* 7.1	Erroneous re−type warning for Cray−style pointer dummy argument	26−Nov−2002
Symptom	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
Current Status/Solution 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.	

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−2002
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−2002

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 inhibit some −O3 optimizations resulting in lower performance	27−Jan−2003
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 efc −O3 −openmp −opt_report test.f90				
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*	Idb cannot resolve local variable	26−Nov−2002
Symptom	Idb 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−2002
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−2002
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−2002
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
27459	Intel(R) Fortran Compiler for Linux*	6.0	Red Hat* 7.2	Representation of Infinity and NaN	21−Nov−2002
Symptom	The Intel Fortran compiler currently generates '++++++' and '??????' output−style representations for the values Infinity and NaN, respectively. These values are further not accepted as valid program input and attempts to use these representations as input values will lead to an error condition that may be indicated by the error "Input/Output Error 141: Invalid real".				

Current Status/Solution

Future Run−time support will generate [+|−] Inf and NaN for Infinity and NaN, respectively. These will however still not be accepted as valid input values. A valid floating point value will continue to be required for input.

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−2002
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−2002
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 Linux 7.3	Files opened with REWIND and F_UFMTENDIAN are handled in Little endian format	28−Oct−2002
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 lengths that exceed 132 characters	27−Sep−2002
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 A source program with free−form lines longer than 132 columns is non−standard and non−portable. The latest Intel(R) Fortran Compiler 7.0. has implemented extensions to support line lengths up to 255 characters. 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−2002
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	SuSe 8.0	The compiler does not compile a file with 250 or more equivalence statements	29−Oct−2002
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
29260	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−2002
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
31312	Intel(R) Fortran Compiler for Linux*	7.0	Red Hat* 7.2	rewind doesn't force flushing of a buffer	6−Mar−2003
Symptom	A rewind call doesn't force the flushing of a buffer. A sample test case that exhibits this problem is shown below. do i = 1, 10 print *, i write(10,'(i10)') i rewind 10 call sleep(2) end do end				
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
29654	Intel(R) Fortran Compiler for Linux*	7.0	Red Hat* 7.2	Function erf() not defined in Fortran library for Itanium(R) compiler	5−Mar−2003
Symptom	When a test case containing a call to erf() is compiled using the Itanium(R) compiler, the following error message is reported: "undefined reference to 'erf_' "				

Current Status/Solution

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

The erf() function is not part of the Fortran standard, although it is widely used. Intel has an implementation in C, with Fortran wrappers for erf() and derf() on IA−32, but no Fortran wrappers for Itanium(R) architecture. The implementation of erf() as a generic Fortran intrinsic function on both IA−32 and Itanium architectures is being considered for a future compiler.

However, as a workaround, you can call the C version directly, e.g.,

```
program test_erf
implicit none
real, external :: erf

!DEC$ATTRIBUTES C, ALIAS:'erf' :: erf
real*8 x, y
integer i

do i = 0, 10
x = i
y = erf( x )
print*, "x = ", x , "y = ", y
enddo

end
```

This both defines the symbol erf without the trailing underscore, and gives it C linkage.

Only !DEC$ or !MS$ prefixes are accepted, not !DIR$.

Reference #	Product	Version	Operating System	Title	Last Update
30098	Intel(R) Fortran Compiler for Linux*	6.0,7.0	Red Hat* 7.3	ieee_flags returns blanks in "out" parameter	24−Jan−2003
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. Instead, you will only see blanks.				

Current Status/Solution

This problem has been resolved in a product update with package ID l_fc_pu_7.0.083 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

Reference #	Product	Version	Operating System	Title	Last Update
29665	Intel(R) Fortran Compiler for Linux*	7.0	Red Hat* 7.2	Module 328.fma3d in SpecOMP2001fp test suite may get into an infinite loop with −O3 −openmp	6−Mar−2003
Symptom	When run on multiple processor systems, the training data set for the module 328.fma3d_m in the SpecOMP2001fp test suite can get stuck into an infinite loop. This has been observed when the module was built with the −O3 and −openmp compiler switches.				
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
31310	Intel(R) Fortran Compiler for Linux*	7.0	Red Hat 7.3	Using idb under GNU Emacs v21 fails with error	5−Mar−2003
Symptom	When using the Linux Application Debugger, idb, under Emacs v21, after loading idb.el, the command M−x idb fails with the error "Can't figure out which 'gud' is being used." This is due to a change in the syntax of the function `gud−common−init' between GNU Emacs 20.7 and 21. In 20.7, gud−common−init takes 4 arguments: (gud−common−init COMMAND−LINE MASSAGE−ARGS MARKER−FILTER FIND−FILE) In Emacs 21, the last argument has been made optional: (gud−common−init COMMAND−LINE MASSAGE−ARGS MARKER−FILTER &optional FIND−FILE)				
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
30702	Intel(R) Fortran Compiler for Linux*	7.0	Red Hat* 7.2	Use of OPENMP ALLOCATE statement fails in a PRIVATE clause	6−Mar−2003
Symptom	When compiling a program that uses the OPENMP allocate statement inside a PRIVATE clause, the compiler generates a Compiler Internal Error. Please see the program below, followed by the build results. subroutine test() integer i,n				

	real (kind=8), allocatable::ar(:)
	
	do i=1,n
	
	!$OMP PARALLEL PRIVATE(ar)
	allocate(ar(20))
	!$OMP END PARALLEL
	
	end do
	return
	end subroutine test
	$ efc −c −O0 −w95 p.f90 −openmp
	Intel(R) Fortran Itanium(R) Compiler for Itanium(R)−based applications
	Version 7.0, Build 20021113
	Copyright (C) 1985−2002 Intel Corporation. All rights reserved.
	
	efc: Command line warning: openmp requires C style preprocessing; fpp level is reset to 2
	EPC Fortran−95 Version F95 Intel:200200:131124
	Copyright (c) 1993−2000 EPCL. All Rights Reserved.
	/tmp/efcNIyFQg.f90
	external subroutine TEST
	p.f90(7) : (col. 0) remark: OpenMP DEFINED REGION WAS PARALLELIZED.
	Compiler Internal Error : Please report to Intel(R) Corporation
	compilation aborted for p.f90 (code 1)

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
31200	Intel(R) Fortran Compiler for Linux*	7.0	Other (specify below)	OpenMP code using many OMP SECTIONS causes long compile times	5−Mar−2003
Symptom	The compiler isn't checking for OMP SECTIONS directives efficiently, so if you have many of them in your code with many sub OMP SECTION directives, the compiler can take a very long time to compile.				
Current Status/Solution This problem has been resolved in a product update with package ID l_fc_pu_7.0.087 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
31677	Intel(R) Fortran Compiler for Linux*	7.0	Red Hat* 7.2	Using −prof_use −O3 switches may produce incorrect program output	5−Mar−2003

Symptom	The compiler may produce incorrect code for some programs when the code is compiled with the −O3 −prof_use compiler switches. This is not a general issue and does not apply to all programs. Correct code is produced using "−O1 −prof_use", or "−O2 −prof_use", or simply the "−O3" switch without "−prof_use".
Current Status/Solution This is a known issue that may be resolved in a future product release.	

Windows*

Reference #	Product	Version	Operating System	Title	Last Update
21888	Intel(R) Fortran Compiler for Windows*	6.0, 7.0	Windows* 2000 Server	The ifl compiler displays syntax error if a backslash is used in Format statement	30−Oct−2002
Symptom	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				
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
23975	Intel(R) Fortran Compiler for Windows*	6.0,7.0	Windows* Me	Problem with I0 edit descriptor and INTEGER(8) values	28−Oct−2002
Symptom	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				

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
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−2002

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	10−Feb−2003

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.

Reference #	Product	Version	Operating System	Title	Last Update
25488	Intel(R) Fortran Compiler for Windows*	5.0, 6.0	Windows* 2000 Professional	Incompatibility with Compaq* Visual Fortran: () optional in function declaration	6−Dec−2002
Symptom	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)				
Current Status/Solution 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					

Reference #	Product	Version	Operating System	Title	Last Update
26140	Intel(R) Fortran Compiler for Windows*	6.0,7.0	Windows* Me	Character constant with KIND not accepted in format field of WRITE	28−Oct−2002
Symptom	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

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
26417	Intel(R) Fortran Compiler for Windows*	6.0, 7.0	Windows* 2000 Professional	prof_gen/prof_use may not always generate the most efficient code	22−Nov−2002
Symptom	In some circumstances, profile guided optimization may unnecessarily inhibit software pipelining of a loop.				

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−2002
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

25191	Intel(R) Fortran Compiler for Windows*	5.0.1, 6.0	Windows* 2000 Professional	Call to function that outputs to internal file from within formatted write statement causes crash	23−Dec−2002
Symptom	Calling a function from within a write statement, which itself does I/O, may cause the executable to crash with an error message that begins: ***Fatal: bad switch value... This error would be exhibited by the following code fragment. If your code contained the following write statement: WRITE (*, *) FOO(int_to_char (12345)) where FOO also contains a write statement: WRITE (string, FMT='(I5)') val				
Current Status/Solution 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					

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−2002
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
28447	Intel(R) Fortran	6.0	Windows*	Support for /Qdoubletemps	5−Mar−2003

	Compiler for Windows*		2000 Professional		
Symptom	The documentation and command line help indicates that there is a /Qdoubletemps option. We do not support this option.				
Current Status/Solution The documentation has been corrected in the Intel(R) C++ 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 .					

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−2002
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.					

Reference #	**Product**	**Version**	**Operating System**	**Title**	**Last Update**
31303	Intel(R) Fortran Compiler for Windows*	7.0	Windows* XP Professional	Installer changes paths in Microsoft* Visual C++* 6.0 IDE even when build tools not installed	5−Mar−2003
Symptom	During the installation of the Intel® Fortran Compiler, if you choose not to integrate with the Microsoft* Visual C++* 6.0 IDE, the installation will still add Intel compiler directories in the IDE bin, include, and lib paths. This will interfere with Compaq* Visual Fortran compilations done in this IDE.				
Current Status/Solution This is a known issue that may be resolved in a future product release. As a workaround you can manually remove these directories by going to Tools−>Options−>Directories within the Microsoft Visual C++ 6.0 IDE.					

Reference #	**Product**	**Version**	**Operating System**	**Title**	**Last Update**
31043	Intel(R) Fortran Compiler for Windows*	7.0	Windows* XP Professional	Fortran project files rebuild even if target is up to date	5−Mar−2003

Symptom	While using version 7.0 compiler from within Visual Studio .NET Fortran projects rebuild even if the target is up to date. If a file has an include statement in it, then it will always be rebuilt even if it is up to date.

Current Status/Solution

This problem has been resolved in a product update with package ID w_fc_pu_7.0.089_pg090.1 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
30924	Intel(R) Fortran Compiler for Windows*	7.0	Windows* XP Professional	Unable to compile individual Fortran file in Microsoft* Visual Studio .NET* environment	5−Mar−2003
Symptom	In Microsoft* Visual Studio .NET* it is possible to right click on a C/C++ file and select "compile" to build just that file. There is also a function key for this as well. It is not possible to compile an individual Fortran file, however, with the Intel(R) Fortran Compiler for Windows*.				

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
31492	Intel(R) Fortran Compiler for Windows*	7.0	Windows* 2000 Professional	C preprocessor "stringization" and "merging" operators not recognized by Fortran preprocessor	5−Mar−2003

Symptom

The C preprocessor "stringization" operator, "#", and "merging" operator, "##" are not recognized by the Fortran preprocessor. In addition, the comment character, "!", is interpreted as a comment in the preprocessor directive.

For example, given the following test program, test.fpp:

```
program test
#define quote(str) #str
quote(test1)
#define concat(a,b) a ## b
concat (1,2)
#define comment(string) ! string
comment(test2)
end
```

When preprocessing with ifl /E test.fpp the output is:

```
# 1 "test.fpp"
program test
```

	#test1 1 ## 2 end When preprocessing with icl /E test.fpp the (correct) output is: #line 1 "test.fpp" program test "test1" 12 ! test2 end
Current Status/Solution This is a known issue that may be resolved in a future product release. As a workaround in fixed format source for the comment character issue, you can use the "*" or "C" comment characters.	

Reference #	**Product**	**Version**	**Operating System**	**Title**	**Last Update**
31556	Intel(R) Fortran Compiler for Windows*	7.0	Windows* XP Professional	The "build analysis" feature is only available after the initial build of an open solution	5−Mar−2003
Symptom	Open a solution in Visual Studio .NET*. On the first build, all Fortran files are rebuilt even if it is not necessary. Subsequent builds do not recompile the Fortran files, but if the solution is closed and reopened all of the Fortran files are rebuilt again.				
Current Status/Solution This is a known issue that may be resolved in a future product release.					

to them.
Intel, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.