



Intel® Fortran Compiler for Linux* Systems Options Quick Reference Guide

[Legal Information](#)

Copyright © 2003 Intel Corporation

Document Number: 253258-001

Disclaimer and Legal Information

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. 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, or life sustaining applications.

This Quick Reference Guide as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

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.

The software described in this Quick Reference Guide may contain software defects which may cause the product to deviate from published specifications. Current characterized software defects are available on request.

Intel SpeedStep, Intel Thread Checker, Celeron, Dialogic, i386, i486, iCOMP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetStructure, Intel Xeon, Intel Centrino, Intel XScale, Itanium, MMX, MMX logo, Pentium, Pentium II Xeon, Pentium III Xeon, Intel Pentium M processor, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

* Other names and brands may be claimed as the property of others.

Copyright © Intel Corporation 2003.

Table Of Contents

Disclaimer and Legal Information	ii
How to Use This Guide	1
Alphabetical Listing	1
Windows* and Linux* Cross-reference	1
Conventions used in the Options Quick Guide Tables.....	1
New Compiler Options	3
Compiler Options Quick Reference Alphabetical	13
Windows* to Linux* Options Cross-reference	44
Index.....	89

How to Use This Guide

This document provides three sets of tables comprising Intel® Fortran Compiler Options Quick Reference Guide:

- New compiler options for the current release
- Alphabetical listing of all options
- Windows* and Linux* Cross-reference

For complete information on each option, refer to the *Intel® Fortran Compiler User's Guide, Volumes I and II*, and the *Intel® Fortran Compiler Manpages*.

Alphabetical Listing

Alphabetical listing is alphabetic tabular reference of all compiler and compilation as well as linker and linking control, and all other options implemented by the Intel Fortran Compiler available for both IA-32 and Itanium® architectures as well as those available exclusively for each architecture.

Each entry in the table has a link to a section in the Intel® Fortran Compiler User's Guide. Within that section, you will find the following:


- the option's synonyms (if applicable)
- detailed description of option's functionality.

Windows* and Linux* Cross-reference

A table containing the Intel Fortran Compiler Options for Windows* and Linux* cross-reference. The table is based on alphabetical order of the Intel Fortran Compiler Options for Linux.

Conventions used in the Options Quick Guide Tables

ON in "Default" column	indicates that the option is in effect by default when compiler is invoked; if an option has a value for the ON state, it is indicated in parenthesis.
OFF in "Default" column	indicates that by default, the option is not used when compiler is invoked; if an option has a value for the OFF state, it is indicated in parenthesis.
[-]	indicates that if option includes a trailing "-", the option is disabled; for example, <code>-ansi_alias-</code> disables <code>-ansi_alias</code> option.

[n]	indicates that the value in [] can be omitted or have various values; for example, in <code>-unroll[n]</code> option, n can be omitted or have different values starting from 0.
Values in { } with vertical bars	are used for option's version; for example, option <code>-i{2 4 8}</code> has these versions: <code>-i2</code> , <code>-i4</code> , <code>-i8</code> .
{n}	indicates that option must include one of the fixed values for n; for example, in option <code>-zp{n}</code> , n can be equal to 1, 2, 4, 8, 16.
Words in <i>this style</i> following an option	indicate option's required argument(s). Arguments are separated by comma if more than one are required. For example, the option <code>-Qoption,tool,opts</code> looks in the command line like this: <code>ifort -Qoption,link,-w myprog.f</code>
<code>-option parameter</code>	indicates that an option requires a parameter; for example, <code>-Ldir</code> : the option <code>-L</code> instructs linker to search directory <code>dir</code> for libraries.
<code>-option keyword</code>	indicates that an option requires one of the values of the <code>keyword</code> .
<code>-option [keyword]</code>	indicates that an option can be used with an optional keyword.
<code>-[no]option</code>	indicates that an option can be used as an <code>option</code> or <code>nooption</code> in which case it instructs the compiler not perform something; for example, <code>-altparam</code> specifies that alternate form of parameter constant declarations is recognized, <code>-noaltparam</code> specifies that alternate form of parameter constant declarations is not recognized,.  Note The <code>[no]options</code> are listed in the alphabetical order of an <code>option</code> .

New Compiler Options

The following table lists new options in this release. See [Conventions Used in the Options Quick Guide Tables](#).

- Options specific to IA-32 architecture
- Options specific to the Itanium® architecture
- All other options are available for both IA-32 and Itanium architectures.

For more details on each of these options refer to the *Intel® Fortran Compiler User's Guide, Volumes I and II*, and the *Intel® Fortran Compiler Manpages*.

Option	Description	Default
<code>-align keyword</code>	Specifies how data items are aligned by adding padding bytes as indicated by a keyword: [no]commons, dcommons, [no]records, [no]recnbyte, [no]sequence.	<code>-align</code> <code>nocommons</code> <code>-align records</code> <code>-align</code> <code>nosequence</code> <code>-align</code> <code>rec8bytes</code>
<code>-assume keyword</code>	Specifies that assumptions are made by the optimizer and code generator as indicated by a keyword: none, [no]accuracy_sensitive, [no]byterecl, [no]buffered_io, [no]dummy_aliases, [no]protect_constants, [no]source_include, [no]underscore See <i>Intel® Fortran Compiler Manpages</i> for more details.	OFF (<code>-assume none</code>)
<code>-assume bsccl</code>	Tells the compiler to treat the backslash character (\) as a C-style control (escape) character syntax in character literals. The default is <code>-assume nobsccl</code> , which tells the compiler to treat the backslash character as a normal character instead of a control character in character literals.	<code>-assume nobsccl</code>
<code>-assume cc_omp</code>	Enables conditional compilation as defined by the OpenMP Fortran API.	<code>-assume cc_omp</code> with <code>-openmp</code> specified

<p><code>-assume minus0</code></p>	<p>Tells the compiler to use Fortran 95 standard semantics for the treatment of IEEE* floating value -0.0 in the <code>SIGN</code> intrinsic, if the processor is capable of distinguishing the difference between -0.0 and +0.0, and to write a value of -0.0 with a negative sign on formatted output.</p> <p>The default is <code>-assume nominus0</code>, which tells the compiler to use Fortran 90/77 standard semantics in the <code>SIGN</code> intrinsic to treat -0.0 and +0.0 as 0.0, and to write a value of -0.0 with no sign on formatted output.</p>	<p>OFF (<code>-assume nominus0</code>)</p>
<p><code>-automatic</code></p>	<p>Puts local variables, except those declared as <code>SAVE</code>, on the runtime stack. Same as <code>-auto</code> or <code>-nosave</code>.</p> <p>This option is one of the three possible states: <code>-auto-scalar</code> (the default state), <code>-automatic</code>, or <code>-static</code>.</p>	<p><code>-auto-scalar</code>; with <code>-recursive</code> or <code>-openmp</code>, the default is <code>-auto</code>.</p>
<p><code>-auto_ilp32</code> Itanium-based systems</p>	<p>Specifies that the application cannot exceed a 32-bit address space, which allows the compiler to use 32-bit pointers whenever possible. To use this option, you must also specify <code>-ipo</code>. Using the <code>-auto_ilp32</code> option on programs that can exceed 32-bit address space (2^{32}) may cause unpredictable results during program execution.</p>	<p>OFF</p>
<p><code>-axN</code> IA-32 only</p>	<p>Automatically optimizes for Intel® Pentium® 4 processors with additional optimizations to Intel processor-specific optimizations.</p>	<p>OFF</p>

-axB IA-32 only	Automatically optimizes for Intel® Pentium® M and compatible Intel processors with additional optimizations to Intel processor-specific optimizations.	OFF
-axP IA-32 only	Automatically optimizes for Intel processors code-named "Prescott" with additional optimizations to Intel processor-specific optimizations.	OFF
-ccdefault <i>keyword</i>	Specifies the type of carriage control used for units 6 and *; <i>keyword</i> = default, fortran, list, or none.	-ccdefault default can be affected by the -vms option, see <i>manpages</i>
-check <i>keyword</i>	Checks runtime conditions, according to the <i>keyword</i> : all, none, [no]arg_temp_created, [no]bounds, [no]format, [no]output_conversion See <i>Intel® Fortran Compiler Manpages</i> for more details.	OFF (-nocheck or (-check none)
-complex_limited_range[-]	Enables the use of basic algebraic expansions of some arithmetic operations involving data of type COMPLEX. This can cause some performance improvements in programs that use a lot of COMPLEX arithmetic, but values at the extremes of the exponent range may not compute correctly.	OFF (-complex_limited_range-, option disabled)
-convert <i>keyword</i>	Specifies the format of unformatted files containing numeric data indicated in a <i>keyword</i> : big_endian, cray, fdx, fgx, ibm, little_endian, native, vaxd, vaxg. See <i>Intel® Fortran Compiler Manpages</i> for more details.	OFF
-d_lines	Compiles debug statements (indicated by D in column 1). Same as -DD.	-nod_lines

<code>-double_size size</code>	Defines the size of DOUBLE PRECISION and DOUBLE COMPLEX declarations, constants, functions, and intrinsics. The size can be 64 (default) or 128.	<code>-double_size 64</code>
<code>-error_limit n</code>	Specifies the maximum number of error-level or fatal-level compiler errors allowed for a file specified on the command line. A maximum of 30 error-level and fatal-level messages are allowed before the compiler stops the compilation.	<code>-error_limit 30</code>
<code>-f66</code> or <code>-66</code>	Enforces FORTRAN-66 semantics.	OFF (<code>-nof66</code>)
<code>-f77rtl</code>	Specifies that the FORTRAN-77-specific run-time support should (or not) be used instead of Intel(R) Fortran.	OFF (<code>-nof77rtl</code>)
<code>-fast</code>	Enhances speed across the entire program. Sets the following command options that can improve runtime performance: <code>-O3</code> , <code>-ipo</code> , and <code>-static</code> .	OFF (<code>-nofast</code>)
<code>-fcommon</code>	Tells the compiler to treat common symbols as global definitions and to allocate memory for each symbol at compile time. This may permit the compiler to use the more efficient GP-relative addressing mode when accessing the symbol.	OFF (<code>-fno-common</code>)
<code>-fixed</code>	Specifies source files are in fixed format. By default, source file format is determined by the file suffix.	OFF (<code>-nofixed</code>)
<code>-fminshared</code>	Tells the compiler to treat a compilation unit as a component of a main program and not to link it as a shareable object. Implies <code>-fvisibility=protected</code> .	OFF

<code>-fpconstant</code>	Tells the compiler to extend the precision to double precision for single-precision constants assigned to double-precision variables.	OFF (<code>-nofpconstant</code>)
<code>-fpe{n}</code>	<p>Specifies floating-point exception handling at run time for the main program, $n=0, 1, 3$.</p> <p><code>-fpe0</code> - floating underflow results in zero; all other floating-point exceptions abort execution;</p> <p><code>-fpe1</code> - floating underflow results in zero; all other floating-point exceptions produce exceptional values (signed Infinities or NaNs) and execution continues;</p> <p><code>-fpe3</code> - all floating-point exceptions produce exceptional values (signed infinities, denormals, or NaNs) and execution continues; this is the default. Also see <code>-ftz</code>.</p>	<code>-fpe3</code>
<code>-fpic</code> or <code>-fPIC</code>	Generates position-independent code. Can also be specified as <code>-fPIC</code> .	OFF
<code>-free</code>	Specifies source files are in free format. By default, source file format is determined by the file suffix.	OFF (<code>-nofree</code>)
<code>-fpscomp [keyword]</code>	Specifies a level of compatibility with Microsoft* Fortran PowerStation as indicated by a keyword: all, none, [no]filesfromcmd, [no]general, [no]ioformat, [no]libs, [no]logicals. See <i>Intel® Fortran Compiler Manpages</i> for more details.	<p>For all and nolibs: <code>-fpscomp libs</code></p> <p>For the rest: <code>-fpscomp none</code></p>

<p>-fpstkchk</p> <p>IA-32 only</p>	<p>Generates extra code after every function call to ensure that the FP (floating-point) stack is in the expected state. By default, there is no checking. So when the FP stack overflows, NaN value is put into FP calculations, and the program's results differ. Unfortunately, the overflow point can be far away from the point of the actual bug. The <code>-fpstkchk</code> option places code that would access-violate immediately after an incorrect call occurred, thus making it easier to locate these issues.</p>	<p>OFF</p>
<p>-fvisibility=keyword</p> <p>-fvisibility=keyword=filename</p>	<p>The first form specifies the default visibility for global symbols using one of the five command line options corresponding to the keyword: external, default, protected, hidden, and internal.</p> <p>The second form specifies the visibility for symbols that are in a file (this form overrides the first form). The <i>filename</i> is the pathname of a file containing the list of symbols whose visibility you want to set; the symbols are separated by whitespace (spaces, tabs, or newlines).</p>	<p>OFF</p>
<p>-[no]intconstant</p>	<p>Tells the compiler to use FORTRAN 77 semantics, rather than Fortran 95/90 semantics, to determine the <code>KIND</code> for integer constants.</p>	<p>OFF (-nointconstant)</p>
<p>-integer_size size</p>	<p>Defines the size of <code>INTEGER</code> and <code>LOGICAL</code> variables. The <i>size</i> can be 16, 32, or 64.</p>	<p>-integer_size 32</p>
<p>-names keyword</p>	<p>Specifies how source code identifiers and external names are interpreted as indicated by a</p>	<p>OFF</p>

	keyword: <code>as_is</code> , <code>lowercase</code> , <code>uppercase</code>	
<code>-no_cpprt</code>	Prevents linking of the C++ runtime libraries.	OFF
<code>-noaltparam</code>	Specifies if alternate form of parameter constant declarations is recognized or not.	<code>-altparam</code>
<code>-nodefaultlibs</code>	Prevents the compiler from using standard libraries when linking.	OFF
<code>-nodefine</code>	Specifies that all preprocessor definitions apply only to <code>-fpp</code> and not to Intel Fortran conditional compilation directives.	OFF
<code>-nofor_main</code>	Specifies the main program is not written in Fortran, and prevents the compiler from linking <code>for_main.o</code> into applications.	<code>-for_main</code>
<code>-nolib_inline</code>	Disable inline expansion of intrinsic functions.	OFF
<code>-nostartfiles</code>	Prevents the compiler from using standard startup files when linking.	OFF
<code>-nostdinc</code>	Removes standard directories from include file search path (same as the <code>-x</code> option.)	OFF
<code>-nostdlib</code>	Prevents the compiler from using standard libraries and startup files when linking.	OFF
<code>-prof_format_32</code>	Produces profile data with 32-bit counters; allows compatibility with earlier compilers. The default is to produce profile data with 64-bit counters to handle large numbers of events.	OFF
<code>-real_size size</code>	Defines the size of REAL and COMPLEX declarations, constants, functions, and intrinsics. The <code>size</code> can be 32, 64, or 128.	<code>-real_size 32</code>

-recursive	Specifies that all routines should be compiled for possible recursive execution. This option sets the <code>-auto</code> option.	OFF (-norecursive)
-reentrancy <i>keyword</i>	Specifies that the compiler should generate reentrant code that supports a multithreaded application. <i>keyword</i> : none, threaded, async.	-reentrancy none
-shared-libcxa	Links the Intel libcxa C++ library dynamically, overriding the default behavior when <code>-static</code> is used. This option is the opposite of <code>-static-libcxa</code> .	ON
-stand <i>keyword</i>	Causes the compiler to issue compile-time messages for nonstandard language elements. <i>keyword</i> : f90, f95, none.	OFF (-nostand)
-static-libcxa	Links the Intel libcxa C++ library statically. This option is the opposite of <code>-shared-libcxa</code> .	OFF
-T <i>file</i>	Tells the linker to read link commands from the specified <i>file</i> .	OFF
-threads	Specifies that multithreaded libraries should be linked. This option sets the <code>-reentrancy threaded</code> option.	OFF (-nothreads)
-tpp7 IA-32 only	Optimizes for the Intel® Pentium® 4 processors, Intel® Xeon(TM) processors, Intel® Pentium® M processors, and Intel processors code-named "Prescott." This is the default on IA-32 systems.	ON
-traceback	Tells the compiler to generate extra information in the object file to allow the display of source file traceback information at runtime when a severe error occurs.	OFF (-notraceback)

<code>-warn keyword</code>	Specifies the level of warning messages issued by the compiler as indicated by a keyword: all, none, [no]alignments, [no]declarations, [no]errors, [no]general, [no]ignore_loc, [no]stderrs, [no]truncated_source, [no]uncalled, [no]unused, [no]usage. See <i>Intel® Fortran Compiler Manpages</i> for details.	OFF (-warn none or -warn nokeyword)
<code>-what</code>	Prints the version strings of the Fortran command and the compiler.	OFF
<code>-Wl o1 [, o2, ...]</code>	Passes options -o1, -o2, etc. to the linker for processing.	OFF
<code>-Wp o1 [, o2, ...]</code>	Passes options -o1, -o2, etc. to the preprocessor.	OFF
<code>-Xlinker val</code>	Pass <i>val</i> directly to the linker for processing.	OFF
<code>-xN</code> IA-32 only	Optimizes for Intel® Pentium® 4 processor. When the main program is compiled with this option, it will detect non-compatible processors and generate an error message during execution. This option also enables new optimizations in addition to Intel processor specific-optimizations.	OFF
<code>-xB</code> IA-32 only	Optimizes for Intel® Pentium® M and compatible Intel processors. When the main program is compiled with this option, it will detect non-compatible processors and generate an error message during execution. This option also enables new optimizations in addition to Intel processor specific-optimizations.	OFF
<code>-xP</code> IA-32 only	Optimizes for Intel processors code-named "Prescott." When	OFF

	<p>the main program is compiled with this option, it will detect non-compatible processors and generate an error message during execution. This option also enables new optimizations in addition to Intel processor specific-optimizations.</p>	
--	--	--

Compiler Options Quick Reference Alphabetical

The following table describes options that you can use for compilations you target to either IA-32- or Itanium®-based applications or both.

- Options specific to IA-32 architecture (IA-32 only)
- Options specific to the Itanium® architecture (Itanium-based systems only)
- All other options are available for both IA-32 and Itanium architectures.
- The options that are new for this release are marked with (***new**). If a new keyword that does not change the functionality is added, this mark appears near that keyword value. If a new keyword adds a new functionality, the option with the new keyword takes a separate entry and is marked as (***new**).

For more details on each of these options refer to the *Intel® Fortran Compiler User's Guide, Volumes I and II*, and the *Intel® Fortran Compiler Manpages*.

See [Conventions Used in the Options Quick Guide Tables](#) for more details on conventions used.

Option	Description	Default
-1	Executes at least one iteration of DO loops (same as the <code>-onetrip</code> option). This option has the same effect as <code>-f66</code> or <code>-66</code> .	OFF
-72, -80, -132	Treats the statement field of each fixed-form source line as ending in column 72 (the default), 80, or 132 (same as the <code>-extend_source</code> option).	-72
<code>-align keyword</code> (*new)	Specifies how data items are aligned by adding padding bytes as indicated by a keyword: <code>all</code> , <code>none</code> , <code>[no]commons</code> , <code>dcommons</code> , <code>[no]records</code> , <code>recnbyte</code> ($n=1, 2, 4, 8, 16$), <code>[no]sequence</code> . See <i>Intel® Fortran Compiler Manpages</i> for more details.	<code>-align</code> <code>nocommons</code> <code>-align records</code> <code>-align</code> <code>nosequence</code> <code>-align</code> <code>rec8bytes</code>
<code>-ansi_alias</code>	Tells the compiler to assume that the program adheres to the Fortran 95 Standard type aliasability rules.	<code>-ansi_alias-</code>

<p><code>-arch keyword</code> (*new) IA-32 only</p>	<p>Determines the version of the architecture for which the compiler generates instructions.</p> <p>keyword:</p> <p>pn1 Optimizes for the Intel® Pentium® processor.</p> <p>pn2 Optimizes for the Intel® Pentium® Pro, Intel® Pentium® II, and Intel® Pentium® III processors.</p> <p>pn3 This is the same as specifying the <code>-arch pn2</code> option.</p> <p>pn4 Optimizes for the Intel® Pentium® 4 processor.</p>	<p><code>-arch pn4</code></p>
<p><code>-assume bsccl</code> (*new)</p>	<p>Tells the compiler to treat the backslash character (\) as a C-style control (escape) character syntax in character literals. The default is <code>-assume nobsccl</code>, which tells the compiler to treat the backslash character as a normal character instead of a control character in character literals.</p>	<p><code>-assume nobsccl</code></p>
<p><code>-assume keyword</code> (*new)</p>	<p>Specifies assumptions made by the optimizer and code generator according to the indicated keyword:</p> <p>none, [no]bsccl, [no]byterecl, [no]buffered_io, [no]dummy_aliases, [no]minimums0, [no]protect_constants, [no]source_include, [no]underscore</p> <p>See <i>Intel® Fortran Compiler Manpages</i> for more details.</p>	<p>OFF (-assume none)</p>
<p><code>-assume cc_omp</code> (*new)</p>	<p>Enables conditional compilation as defined by the OpenMP Fortran API.</p> <p><code>-openmp</code> enables this option: sets <code>-assume cc_omp</code></p>	<p><code>-assume cc_omp</code> with <code>-openmp</code> specified</p>

<p><code>-assume minus0</code> (*new)</p>	<p>Tells the compiler to use Fortran 95 standard semantics for the treatment of IEEE* floating value -0.0 in the SIGN intrinsic, if the processor is capable of distinguishing the difference between -0.0 and +0.0, and to write a value of -0.0 with a negative sign on formatted output.</p> <p>The default is <code>-assume nominus0</code>, which tells the compiler to use Fortran 90/77 standard semantics in the SIGN intrinsic to treat -0.0 and +0.0 as 0.0, and to write a value of -0.0 with no sign on formatted output.</p>	<p><code>-assume nominus0</code></p>
<p><code>-auto</code></p>	<p>Places variables, except those declared as SAVE, on the runtime stack (same as <code>-automatic</code> or <code>-nosave</code>).</p>	<p><code>-auto_scalar</code> or: if you specify <code>-recursive</code> or <code>-openmp</code>, the default is <code>-auto</code></p>
<p><code>-auto_ilp32</code> Itanium-based systems</p>	<p>Specifies that the application cannot exceed a 32-bit address space, which allows the compiler to use 32-bit pointers whenever possible. To use this option, you must also specify <code>-ipo</code>.</p> <p>Using the <code>-auto_ilp32</code> option on programs that can exceed 32-bit address space (2^{32}) may cause unpredictable results during program execution.</p>	<p>OFF</p>
<p><code>-auto_scalar</code></p>	<p>Makes AUTOMATIC all scalar local variables of intrinsic type INTEGER, REAL, COMPLEX, or LOGICAL. You cannot specify <code>-save</code>, <code>-auto</code>, or <code>-automatic</code> with this option.</p>	<p>OFF</p>
<p><code>-autodouble</code></p>	<p>Defines real variables to be REAL (KIND=8). This option is the same as specifying <code>-r8</code>.</p>	<p>OFF</p>

<code>-automatic</code> (*new)	Places variables, except those declared as <code>SAVE</code> , on the runtime stack (same as <code>-auto</code> or <code>-nosave</code>). This option is one of the three possible states: <code>-auto-scalar</code> (the default state), <code>-automatic</code> , or <code>-static</code> . If you specify <code>-recursive</code> or <code>-openmp</code> , the default is <code>-auto</code> .	<code>-auto_scalar</code> or: if you specify <code>-recursive</code> or <code>-openmp</code> , the default is <code>-auto</code> .
<code>-ax{K W N B P}</code> IA-32 compiler	Generates processor-specific code corresponding to one of <i>codes</i> : K – Intel® Pentium® III processors and compatible Intel processors. W – Intel Pentium 4 processors and compatible Intel processors. The *new codes: these options enable new optimizations in addition to Intel processor-specific optimizations. (*new) N – Intel Pentium 4 processors and compatible Intel processors. (*new) B – Intel® Pentium® M and compatible Intel processors. (*new) P – Intel processors code-named "Prescott" and compatible Intel processors. You can use more than one of the <code>-ax</code> options by combining the characters that denote the processor type. For example, you can specify <code>-axKW</code> to generate code for Intel® Pentium® III and Intel Pentium 4 processors.	OFF
<code>-Bdynamic</code>	Enables dynamic linking of libraries at runtime. Smaller executables are created than with static linking.	OFF
<code>-Bstatic</code>	Enables static linking of a user's library.	OFF

-c	Causes the compiler to compile to an object (.o) file only and not link.	OFF
-CB	Performs runtime checks on whether array subscript and substring references are within declared bounds. Same as <code>-check bounds</code> .	OFF
<code>-ccdefault keyword</code> (*new)	Specifies the type of carriage control used for units 6 and *; keyword = default, fortran, list, or none.	<code>-ccdefault</code> default can be affected by the <code>-vms</code> option, see <i>manpages</i>
<code>-check keyword</code> (*new)	Checks runtime conditions, according to the keyword: all, none, [no]arg_temp_created, [no]bounds, [no]format, [no]output_conversion See <i>Intel® Fortran Compiler Manpages</i> for more details.	OFF (<code>-nocheck</code> or <code>-check none</code>)
-cm	Suppresses all messages about questionable programming practices (same as the <code>-warn nousage</code> option).	OFF
-common_args	Tells the compiler that dummy (formal) arguments to procedures share memory locations with other dummy arguments or with COMMON variables that are assigned. This is the same as specifying <code>-assume dummy_aliases</code> .	OFF
<code>-complex_limited_range[-]</code> (*new)	Enables the use of basic algebraic expansions of some arithmetic operations involving data of type COMPLEX. This can cause some performance improvements in programs that use a lot of COMPLEX arithmetic, but values at the extremes of the exponent range may not compute correctly.	OFF (<code>-complex_limited_range-</code> , option disabled)
<code>-convert keyword</code> (*new)	Specifies the format of unformatted files containing numeric data indicated in a keyword: big_endian, cray, fdx, fgx, ibm, little_endian, native, vaxd, vaxg.	OFF

-cpp	Runs the Fortran preprocessor on source files prior to compilation (same as the -fpp option).	OFF
-Dname -Dname[=value]	Defines the <i>name</i> as a definition to use with conditional compilation directives or the Fortran preprocessor (-fpp). The <i>value</i> can be an integer or it can be a character string delimited by double quotes; for example, -Dname="string". If no definition is given, the <name> is defined as "1".	OFF
-d_lines (*new)	Compiles debug statements (indicated by D in column 1); this is the same as specifying -DD.	-nod_lines
-DD	Compiles debugging statements indicated by the letter D in column 1 of the source code; this is the same as specifying -d_lines.	OFF
-double_size <i>size</i> (*new)	Defines the size of DOUBLE PRECISION and DOUBLE COMPLEX declarations, constants, functions, and intrinsics. The size can be 64 (default) or 128.	-double_size 64
-dryrun	Specifies that driver tool commands should be shown but not executed.	OFF
-dynamic-linker <i>file</i>	Specifies a dynamic linker in <i>file</i> other than the default.	OFF
-dyncom "a,b,c"	Enables dynamic allocation of the specified COMMON blocks at run time. The quotes are required.	OFF
-E	Causes the Fortran preprocessor to send output to stdout.	OFF
-e90, -e95	Causes the compiler to issue errors instead of warnings for nonstandard Fortran 90 (-e90) or Fortran 95 (-e95). No such errors or warnings are issued by default.	OFF

<code>-[no]error_limit n</code> (*new)	Specifies the maximum number of error-level or fatal-level compiler errors allowed for a file specified on the command line. A maximum of 30 error-level and fatal-level messages are allowed before the compiler stops the compilation.	<code>-error_limit 30</code>
<code>-EP</code>	Causes the Fortran preprocessor to send output to stdout, omitting <code>#line</code> directives.	OFF
<code>-error_limit num</code>	Specifies the maximum number of error-level or fatal-level compiler errors allowed for a file specified on the command line. If you specify <code>-noerror_limit</code> , there is no limit to the number of errors that are allowed. The default indicates a maximum of 30 error-level and fatal-level messages before the compiler stops the compilation.	<code>-error_limit 30</code>
<code>-extend_source [size]</code>	Specifies the column number to use to end the statement field in fixed-form source files. <i>size</i> can be 72, 80, or 132. The default behavior is <code>-noextend_source</code> , which implies column 72. If you specify <code>-extend_source</code> with no size, the default is <code>-extend_source 132</code> . Specifying <code>-extend_source</code> sets the <code>-fixed</code> option.	OFF
<code>-F</code>	Causes the Fortran preprocessor to send output to a file (same as the <code>-preprocess_only</code> and <code>-P</code> options). To use this option, you must also specify <code>-fpp</code> .	OFF
<code>-f66</code> or <code>-66</code> (*new)	Tells the compiler to apply FORTRAN-66 semantics: the execution of at least one iteration of DO loops, different EXTERNAL statement syntax and semantics, and different behavior of the BLANK= and STATUS= specifiers on the OPEN statement. The default is <code>-nof66</code> , which applies Fortran 95 semantics.	OFF

<code>-f77rtl</code> (*new)	Tells the compiler to use the runtime behavior of FORTRAN 77 instead of Intel® Fortran. This affects some INQUIRE specifiers when the unit is not connected to a file, PAD= defaults to 'NO' for formatted input, NAMELIST input format is different, and NAMELIST and list-directed input of character strings must be delimited by apostrophes or quotes.	OFF (<code>-nof77rtl</code>)
<code>-fast</code>	Enhances speed across the entire program. Sets the following command options that can improve runtime performance: <code>-O3</code> , <code>-ipo</code> , and <code>-static</code> .	OFF
<code>-fcode_asm</code>	Produces an assembly file with optional code annotations. To use this option, you must also specify <code>-S</code> .	OFF
<code>-fcommon</code> (*new)	Tells the compiler to treat common symbols as global definitions and to allocate memory for each symbol at compile time. This may permit the compiler to use the more efficient GP-relative addressing mode when accessing the symbol.	OFF (<code>-fno-common</code>)
<code>-FI</code>	Specifies source files are in fixed format (same as the <code>-fixed</code> option).	Based on file extension
<code>-fixed</code> (*new)	Specifies source files are in fixed format. By default, source file format is determined by the file extension.	Based on file extension
<code>-fminshared</code> (*new)	Tells the compiler to treat a compilation unit as a component of a main program and not to link it as a shareable object. Implies <code>-fvisibility=protected</code> .	OFF
<code>-fno-alias</code>	Specifies that aliasing should not be assumed in the program.	<code>-falias</code>
<code>-fno-fnalias</code>	Specifies that aliasing should not be assumed within functions, but should be assumed across calls.	<code>-ffnalias</code>
<code>-fnsplit</code> Itanium-based systems	Enables function splitting (enabled with <code>-prof_use</code>). The default is <code>-fnsplit-</code> , which disables the splitting within a routine but leaves function grouping enabled.	<code>-fnsplit-</code>

-fp IA-32 only	Disables using EBP as a general purpose register so it can be used as a stack frame pointer.	OFF
-fp_port IA-32 only	Rounds floating-point results after floating-point operations, so rounding to user-declared precision happens at assignments and type conversions; this has some impact on speed. The default is to keep results of floating-point operations in higher precision; this provides better performance but less consistent floating-point results.	OFF
-fpconstant (*new)	Tells the compiler to extend the precision to double precision for single-precision constants assigned to double-precision variables.	OFF (-nofpconstant)
-fpen (*new)	Specifies floating-point exception handling at run time for the main program, $n=0, 1, 3$. -fpe0 - floating underflow results in zero; all other floating-point exceptions abort execution; -fpe1 - floating underflow results in zero; all other floating-point exceptions produce exceptional values (signed Infinities or NaNs) and execution continues; -fpe3 - all floating-point exceptions produce exceptional values (signed infinities, denormals, or NaNs) and execution continues; this is the default. Also see -ftz.	-fpe3
-fpic, -fPIC (*new)	Generates position-independent code. Can also be specified as -fPIC.	OFF

<code>-fpp[n]</code>	Runs the Fortran preprocessor on source files prior to compilation. n=0: disables # directives (equivalent to <code>-nofpp</code>). n=1: enables # directives. This is equivalent to <code>-fpp</code> and is the default if the Fortran preprocessor is invoked. 2 - Same as 1. 3 - Same as 1.	OFF (if not invoked) <code>-fpp1</code> if the preprocessor is invoked
<code>-fpscomp keyword</code> (*new)	Specifies a level of compatibility with Microsoft* Fortran PowerStation as indicated by a keyword: all, none, [no]filesfromcmd, [no]general, [no]ioformat, [no]libs, [no]logicals.	For all and nolibs: <code>-fpscomp libs</code> For the rest: <code>-fpscomp none</code>
<code>-fpstkchk</code> (*new) IA-32 compiler	Generates extra code after every function call to ensure that the FP (floating-point) stack is in the expected state. By default, there is no checking. So when the FP stack overflows, NaN value is put into FP calculations, and the program's results differ. Unfortunately, the overflow point can be far away from the point of the actual bug. The <code>-fpstkchk</code> option places code that would access-violate immediately after an incorrect call occurred, thus making it easier to locate these issues.	OFF
<code>-FR</code>	Specifies source files are in free format (same as the <code>-free</code> option).	Based on source file extension
<code>-fr32</code> Itanium compiler	Disables use of high floating-point registers. Uses only the lower 32 floating-point registers.	OFF
<code>-free</code> (*new)	Specifies source files are in free format. By default, source file format is determined by the file suffix.	Based on source file extension
<code>-fsource_asm</code>	Produces an assembly file with optional code annotations. To use this option, you must also specify <code>-S</code> .	OFF

-ftz[-]	<p>Enables (or disables: -ftz-) floating underflow results set to zero.</p> <p>For Itanium-based systems only: option -O3 sets -ftz on.</p>	-ftz-
-fverbose-asm	<p>Produces an assembly file with compiler comments, including options and version information. To use this option, you must also specify -S, which sets -fverbose-asm. If you do not want this default when you specify -S, specify -fnoverbose-asm.</p>	-fnoverbose-asm
<p>-fvisibility=keyword d (*new)</p> <p>-fvisibility-keyword=file (*new)</p>	<p>The first form specifies the default visibility for global symbols using one of the five command line options corresponding to the keyword: external, default, protected, hidden, and internal.</p> <p>The second form specifies the visibility for symbols that are in a file (this form overrides the first form). The <i>file</i> is the pathname of a file containing the list of symbols whose visibility you want to set; the symbols are separated by whitespace (spaces, tabs, or newlines) See <i>Intel® Fortran Compiler Manpages</i> for more details.</p>	OFF
-g	<p>Produces symbolic debug information in the object file. The compiler does not support the generation of debugging information in assemblable files. If you specify the -g option, the resulting object file will contain debugging information, but the assemblable file will not.</p> <p>On IA-32 systems, specifying the -g or -O0 option automatically enables the -fp option. See "Optimizations and Debugging" in the Intel® Fortran Compiler User's Guide, Volume II.</p>	OFF

<code>-help</code>	Prints the list of compiler options.	OFF
<code>-I<i>dir</i></code>	Specifies a directory to add to the include path, which is used to search for module files (<code>USE</code> statement) and include files (<code>INCLUDE</code> statement).	OFF
<code>-i_dynamic</code>	Links Intel-provided libraries dynamically.	OFF
<code>-i{2 4 8}</code>	Defines the default <code>KIND</code> for integer variables and constants to be 2, 4, and 8 bytes (same as <code>-integer_size {16 32 64}</code>)	<code>-i4</code> same as <code>-integer_size 32</code>
<code>-implicitnone</code>	Sets the default type of a variable to undefined (<code>IMPLICIT NONE</code>). Same as the <code>-u</code> option.	OFF
<code>-inline_debug_info</code>	Preserves the source position of inlined code instead of assigning the call-site source position to inlined code.	OFF
<code>-intconstant (*new)</code>	Tells the compiler to use FORTRAN 77 semantics, rather than Fortran 95/90 semantics, to determine the <code>KIND</code> for integer constants.	OFF (<code>-nointconstant</code>)
<code>-integer_size <i>size</i> (*new)</code>	Defines the size of <code>INTEGER</code> and <code>LOGICAL</code> variables. The <i>size</i> can be 16, 32, or 64.	<code>-integer_size 32</code>
<code>-ip</code>	Enables single-file interprocedural optimizations. If you specify this option, the compiler performs inline function expansion for calls to functions defined within the current source file.	OFF
<code>-ip_no_inlining</code>	Disables full and partial inlining enabled by <code>-i8</code> . To use this option, you must specify <code>-ip</code> or <code>-ipo</code> .	ON
<code>-ip_no_pinlining</code> IA-32 compiler	Disables partial inlining. To use this option, you must specify <code>-ip</code> or <code>-ipo</code> .	OFF

<p><code>-IPF_fltacc</code> Itanium compiler</p>	<p>Disables optimizations that affect floating-point accuracy. If the default setting is used (<code>-IPF_fltacc-</code>), the compiler may apply optimizations that reduce floating-point accuracy. You can use <code>-IPF_fltacc</code> or <code>-mp</code> to improve floating-point accuracy, but at the cost of disabling some optimizations.</p>	<p><code>-IPF_fltacc-</code></p>
<p><code>-IPF_flt_eval_method0</code> Itanium compiler</p>	<p>Directs the compiler to evaluate the expressions involving floating-point operands in the precision indicated by the variable types declared in the program. By default, intermediate floating-point expressions are maintained in higher precision.</p>	<p>OFF</p>
<p><code>-IPF_fma[-]</code> Itanium compiler</p>	<p>Enables the combining of floating-point multiplies and add/subtract operations. Also enables the contraction of floating-point multiply and add/subtract operations into a single operation. The compiler contracts these operations whenever possible. However, if <code>-mp</code> is specified, these contractions are disabled.</p>	<p>OFF (<code>-IPF_fma-</code>)</p>
<p><code>-IPF_fp_speculation mode</code> Itanium compiler</p>	<p>Enables floating-point speculations with one of the following <code>mode</code> conditions:</p> <p><i>fast</i> -Speculate floating-point operations. <i>off</i> -Disables speculation of floating-point operations. <i>safe</i> -Speculate only when safe. <i>strict</i> -This is the same as specifying <i>off</i>.</p>	<p><code>-IPF_fp_speculation fast</code></p>
<p><code>-ipo</code></p>	<p>Enables mult file IP optimizations (between files). When you specify this option, the compiler performs inline function expansion for calls to functions defined in separate files. For this reason, it is important to compile the entire application or multiple, related source files together when you specify <code>-ipo</code>.</p>	<p>OFF</p>

-ipo_c	Generates a multifile object file (ipo_out.o) that can be used in further link steps.	OFF
-ipo_obj	Forces the generation of real object files. Requires -ipo.	IA-32: OFF Itanium Compiler: ON
-ipo_S	Generates a multi file assembly file (ipo_out.s) that can be used in further link steps.	OFF
-ivdep_parallel Itanium compiler	Tells the compiler that there is no loop-carried memory dependency in any loop following an IVDEP directive.	OFF
-Kpic	This is a deprecated option ; it can also be specified as -KPIC. Use -fpic instead.	OFF
-Ldir	Tells the linker to search for libraries in dir before searching the standard directories.	OFF
-lowercase	Causes the compiler to ignore case differences in identifiers and to convert external names to lowercase (same as the -names lowercase option). This is the default.	ON
-mixed_str_len_arg (*new)	Tells the compiler that the hidden length passed for a character argument is to be placed immediately after its corresponding character argument in the argument list. The default (-nomixed_str_len_arg) places the hidden lengths in sequential order at the end of the argument list.	OFF
-module dir	Specifies the directory dir where module (.mod) files should be placed when created and where they should be searched for (USE statement).	OFF

<code>-mp</code>	<p>Maintains floating-point precision (while disabling some optimizations). Restricts optimization to maintain declared precision and to ensure that floating-point arithmetic conforms more closely to the ANSI* and IEEE standards.</p> <p>For most programs, specifying this option adversely affects performance. If you are not sure whether your application needs this option, try compiling and running your program both with and without it to evaluate the effects on both performance and precision.</p>	OFF
<code>-mp1</code>	<p>Improves floating-point precision. This option disables fewer optimizations and has less impact on performance than <code>-mp</code>.</p>	OFF
<code>-names keyword</code> (*new)	<p>Specifies how source code identifiers and external names are interpreted as indicated by a keyword: <code>as_is</code>, <code>lowercase</code>, <code>uppercase</code>.</p>	OFF
<code>-nbs</code>	<p>Tells the compiler to treat a backslash as a normal character, not an escape character (same as the <code>-assume nobsc</code> option).</p>	OFF
<code>-no_cpprt</code> (*new)	<p>Prevents linking of the C++ runtime libraries.</p>	OFF
<code>-noalign</code>	<p>Prevents the alignment of data items. This is the same as specifying <code>-align none</code>.</p>	<code>-align</code>
<code>-noaltparam</code> (*new)	<p>Specifies if alternate form of parameter constant declarations is recognized or not.</p>	<code>-altparam</code>
<code>-nobss_init</code>	<p>Places any variables that are explicitly initialized with zeros in the DATA section. By default, variables explicitly initialized with zeros are placed in the BSS section.</p>	OFF
<code>-nodefaultlibs</code> (*new)	<p>Prevents the compiler from using standard libraries when linking.</p>	OFF

<code>-nodefine</code> (*new)	Specifies that all preprocessor definitions apply only to <code>-fpp</code> and not to Intel Fortran conditional compilation directives.	OFF
<code>-nodps</code>	Specifies that the alternate form of parameter constant declarations (without parenthesis) should not be recognized (same as the <code>-noaltparam</code> option).	<code>-dps</code>
<code>-nofor_main</code> (*new)	Specifies the main program is not written in Fortran, and prevents the compiler from linking <code>for_main.o</code> into applications.	OFF
<code>-noinclude</code>	Prevents the compiler from searching in <code>/usr/include</code> for files specified in an <code>INCLUDE</code> statement. You can specify the <code>-I<i>dir</i></code> option along with this option. This option does not affect <code>cpp(1)</code> behavior, and is not related to the Fortran 95 and 90 <code>USE</code> statement.	OFF
<code>-nolib_inline</code> (*new)	Disables inline expansion of intrinsic functions.	OFF
<code>-nologo</code>	Suppresses compiler version information.	OFF
<code>-nostartfiles</code> (*new)	Prevents the compiler from using standard startup files when linking.	OFF
<code>-nostdinc</code> (*new)	Removes standard directories from include file search path (same as the <code>-x</code> option.)	OFF
<code>-nostdlib</code> (*new)	Prevents the compiler from using standard libraries and startup files when linking.	OFF
<code>-nus</code>	Disables appending an underscore to external subroutine names.	OFF
<code>-ofile</code>	Specifies the name for an output <i>file</i> .	OFF
<code>-O0</code>	Disables <code>-On</code> optimizations. On IA-32 systems, this option sets the <code>-fpp</code> option.	OFF

-01	<p>On IA-32 systems, enables optimizations for speed. Also disables intrinsic recognition and the <code>-fp</code> option. This option is the same as the <code>-O2</code> option.</p> <p>On Itanium(R)-based systems, enables optimizations for server applications (straight-line and branch-like code with flat profile). Enables optimizations for speed, while being aware of code size. For example, this option disables software pipelining and loop unrolling.</p>	OFF
-02, -0	<p>This option is the default for optimizations. However, if <code>-g</code> is specified, the default is <code>-O0</code>.</p> <p>On IA-32 systems, this option is the same as the <code>-O1</code> option.</p> <p>On Itanium-based systems, enables optimizations for speed, including global code scheduling, software pipelining, predication, and speculation.</p> <p>On these systems, the <code>-O2</code> option enables inlining of intrinsics. It also enables the following capabilities for performance gain: constant propagation, copy propagation, dead-code elimination, global register allocation, global instruction scheduling and control speculation, loop unrolling, optimized code selection, partial redundancy elimination, strength reduction/induction variable simplification, variable renaming, exception handling optimizations, tail recursions, peephole optimizations, structure assignment lowering and optimizations, and dead store elimination.</p>	ON

-O3	<p>Enables -O2 optimizations plus more aggressive optimizations, such as prefetching, scalar replacement, and loop transformations. Enables optimizations for maximum speed, but does not guarantee higher performance unless loop and memory access transformation take place.</p> <p>On IA-32 systems, when the -O3 option is used with the -ax and -x options, it causes the compiler to perform more aggressive data dependency analysis than for -O2, which may result in longer compilation times.</p> <p>On Itanium-based systems, enables optimizations for technical computing applications (loop-intensive code): loop optimizations and data prefetch.</p>	OFF
-onetrip	Executes at least one iteration of DO loops (same as the -1 option). This option has the same effect as -f66 or -66.	OFF
-openmp	<p>Enables the parallelizer to generate multithreaded code based on OpenMP* directives. The code can be executed in parallel on both uniprocessor and multiprocessor systems. The -openmp option works with both -O0 (no optimization) and any optimization level of -On.</p> <p>Specifying -O0 with -openmp helps to debug OpenMP applications.</p>	OFF

<code>-openmp_report{0 1 2}</code>	<p>Controls the OpenMP parallelizer 's level of diagnostic messages.</p> <p>0 – Displays no diagnostic information.</p> <p>1 – Displays diagnostics indicating loops, regions, and sections successfully parallelized.</p> <p>2 – Displays the diagnostics specified by <code>-openmp_report1</code> plus diagnostics indicating MASTER constructs, SINGLE constructs, CRITICAL constructs, ORDERED constructs, ATOMIC directives, etc., successfully handled.</p>	<code>-openmp_report1</code>
<code>-openmp_stubs</code>	<p>Enables the compiler to generate sequential code. The OpenMP directives are ignored and a stub OpenMP library is linked.</p>	OFF
<code>-opt_report</code>	<p>Generates optimizations report and directs to <code>stderr</code> unless <code>-opt_report_file</code> is specified.</p>	OFF
<code>-opt_report_file filename</code>	<p>Specifies the <i>filename</i> to hold the optimizations report.</p>	OFF
<code>-opt_report_help</code>	<p>Lists the logical names of optimizers available for report generation (for <code>-opt_report_phase</code>).</p>	OFF
<code>-opt_report_level {min/med/max}</code>	<p>Specifies the detail level of the optimizations report.</p>	<code>-opt_report_levelmin</code>

<p><code>-opt_report _phasephase</code></p>	<p>Specifies the phase against which reports are generated. The compiler generates reports for the optimizer you specify in <i>phase</i>. This option can be used multiple times on the same command line to generate reports for multiple optimizers. Currently, the following optimizer reports are supported:</p> <p><code>ipo</code> - Interprocedural Optimizer <code>hlo</code> - High Level Optimizer <code>ilo</code> - Intermediate Language Scalar Optimizer <code>ecg</code> - Code Generator <code>omp</code> - Open MP <code>all</code> - All phases</p> <p>When one of the above logical names for optimizers is specified for <i>phase</i>, all reports from that optimizer are generated.</p>	<p>OFF</p>
<p><code>-opt_report_ routine [substring]</code></p>	<p>Generates a report on the routines containing the specified <i>substring</i>. If <i>substring</i> is not specified, reports from all routines are generated.</p>	<p>OFF</p>
<p><code>-p</code></p>	<p>Compiles and links for function profiling with <code>gprof</code> (1). This is the same as specifying <code>-pg</code> or <code>-qp</code>.</p>	<p>OFF</p>
<p><code>-P</code></p>	<p>Causes the Fortran preprocessor to send output to a file (same as the <code>-preprocess_only</code> and <code>-F</code> options). To use this option, you must also specify <code>-fpp</code>.</p>	<p>OFF</p>
<p><code>-pad, -nopad</code></p>	<p>Enables the changing of the variable and array memory layout.</p>	<p><code>-nopad</code></p>
<p><code>-pad_source</code></p>	<p>Specifies that fixed-form source records shorter than the statement field width are to be padded with spaces (on the right) to the end of the statement field. This affects the interpretation of character and Hollerith literals that are continued across source records.</p>	<p>OFF</p>

<pre>-par_report {0 1 2 3}</pre>	<p>Controls the auto-parallelizer diagnostic messages.</p> <p>0 – Displays no diagnostic information.</p> <p>1 – Displays diagnostics indicating loops successfully auto-parallelized. This is the default. Issues a "LOOP AUTO-PARALLELIZED" message for parallel loops.</p> <p>2 – Displays diagnostics indicating loops successfully auto-parallelized, as well as unsuccessful loops.</p> <p>3 – Displays the diagnostics specified by <code>-par_report2</code> plus additional information about any proven or assumed dependencies inhibiting auto-parallelization (reasons for not parallelizing).</p>	<pre>-par_report1</pre>
<pre>-par_thresholdn</pre>	<p>Sets a threshold for the auto-parallelization of loops based on the probability of profitable execution of the loop in parallel. This option is used for loops whose computation work volume cannot be determined at compile-time. The threshold is usually relevant when the loop trip count is unknown at compile-time.</p> <p>n=0 to 100. The compiler applies a heuristic that tries to balance the overhead of creating multiple threads versus the amount of work available to be shared amongst the threads.</p>	<pre>n=100</pre>
<pre>-parallel</pre>	<p>Enables the auto-parallelizer to generate multithreaded code for loops that can be safely executed in parallel. To use this option, you must also specify <code>-O2</code> or <code>-O3</code>.</p>	<pre>OFF</pre>

<p>-pc32 -pc64 -pc80 IA-32 compiler</p>	<p>Enables control of floating-point significand precision. Some floating-point algorithms are sensitive to the accuracy of the significand, or fractional part of the floating-point value. For example, iterative operations like division and finding the square root can run faster if you lower the precision with the <code>-pcn</code> option.</p> <p>-pc32 Rounds the significand to 24 bits -pc64 Rounds the significand to 53 sbit -pc80 Rounds the significand to 64 bits</p>	<p>-pc64</p>
<p>-pg</p>	<p>Compile and link for function profiling with <code>gprof(1)</code>. This is the same as specifying <code>-p</code> or <code>-qp</code>.</p>	<p>OFF</p>
<p>-prec_div IA-32 compiler</p>	<p>Improves precision of floating-point divides; it has some speed impact. With some optimizations, such as <code>-xK</code> and <code>-xW</code>, the compiler changes floating-point division computations into multiplication by the reciprocal of the denominator. For example, A/B is computed as $A \times (1/B)$ to improve the speed of the computation. However, for values of B greater than 2126, the value of $1/B$ is "flushed" (changed) to 0.</p> <p>When it is important to maintain the value of $1/B$, use <code>-prec_div</code> to disable the floating-point division-to-multiplication optimization. The result of <code>-prec_div</code> is more accurate, with some loss of performance.</p>	<p>OFF</p>
<p>-prefetch[-] IA-32 compiler</p>	<p>Enables or disables prefetch insertion (requires <code>-O3</code>).</p>	<p>ON</p>
<p>-preprocess_only</p>	<p>Causes the Fortran preprocessor to send output to a file (same as the <code>-F</code> and <code>-P</code> options). To use this option, you must also specify <code>-fpp</code>.</p>	<p>OFF</p>

<code>-prof_dir</code> <i>dir</i>	Specifies a directory <i>dir</i> for the profiling output files, *.dyn and *dpi.	OFF
<code>-prof_file</code> <i>file</i>	Specifies a file name <i>file</i> for the profiling summary file.	OFF
<code>-prof_format_32</code>	Produces profile data with 32-bit counters; allows compatibility with earlier compilers. The default is to produce profile data with 64-bit counters to handle large numbers of events.	OFF
<code>-prof_gen</code>	Instruments a program for profiling.	OFF
<code>-prof_use</code>	Enables use of profiling information during optimization.	OFF
<code>-Qinstall</code> <i>dir</i>	Sets <i>dir</i> as a root directory for compiler installation.	OFF
<code>-Qlocation,</code> <i>tool,path</i>	Sets <i>path</i> as the location of the tool specified by <i>tool</i> .	OFF
<code>-Qoption,</code> <i>tool,opts</i>	Passes options, <i>opts</i> , to the tool specified by <i>tool</i> , which can be <code>fpp</code> , <code>f</code> , <code>c</code> , <code>asm</code> (on IA-32 systems), <code>ias</code> (on Itanium-based systems), or <code>link</code> .	OFF
<code>-qp</code>	Compile and link for function profiling with <code>prof(1)</code> tool. This is the same as specifying <code>-p</code> or <code>-pg</code> .	OFF

-r{8 16}	<p>Defines the KIND for real variables in 8 and 16 bytes.</p> <p>-r8: Defines REAL declarations, constants, functions, and intrinsics as DOUBLE PRECISION REAL*8, and defines COMPLEX declarations, constants, functions, and intrinsics as DOUBLE COMPLEX (COMPLEX*16). This option is the same as specifying -real_size 64 or -autodouble.</p> <p>-r16: Defines REAL and DOUBLE PRECISION declarations, constants, functions, and intrinsics as REAL*16, and defines COMPLEX and DOUBLE COMPLEX declarations, constants, functions, and intrinsics as COMPLEX*32. This option is the same as specifying -real_size 128.</p>	-r8
-rcd IA-32 compiler	Disables the change to truncation of the rounding mode for all floating-point calculations, including floating point-to-integer conversions. This option can improve performance, but floating-point conversions to integer will not conform to Fortran semantics.	OFF
-real_size <i>size</i> (*new)	Defines the size of REAL and COMPLEX declarations, constants, functions, and intrinsics. The <i>size</i> can be 32, 64, or 128.	-real_size 32
-recursive (*new)	Specifies that all routines should be compiled for possible recursive execution. This option sets the -auto option.	OFF (-norecursive)
-reentrancy <i>keyword</i> (*new)	Specifies that the compiler should generate reentrant code that supports a multithreaded application. <i>keyword</i> : none, threaded, async.	OFF (-noreentrancy)
-S	Causes the compiler to compile to an assembly file (.s) only and not link.	OFF
-safe_cray_ptr	Specifies that CRAY* pointers do not alias with other variables.	OFF

-save	Places variables, except those declared as <code>AUTOMATIC</code> , in static memory (same as <code>-noauto</code> or <code>-noautomatic</code>). The default is <code>-auto_scalar</code> . However, if you specify <code>-recursive</code> or <code>-openmp</code> , the default is <code>-auto</code> .	OFF (<code>-auto_scalar</code>)
-scalar_rep[-] IA-32 compiler	Enables scalar replacement performed during loop transformation. To use this option, you must also specify <code>-O3</code> .	OFF (<code>-scalar_rep-</code>)
-shared	Instructs the compiler to build a Dynamic Shared Object (DSO) instead of an executable. On Itanium-based systems , you must specify <code>-fpic</code> for the compilation of each object file you want to include in the shared library.	OFF
-shared-libcxa (*new)	Links the Intel <code>libcxa</code> C++ library dynamically, overriding the default behavior when <code>-static</code> is used. This option is the opposite of <code>-static-libcxa</code> .	ON
-sox[-] IA-32 compiler	Enables saving of the compiler options and version in the executable.	<code>-sox-</code>
-stand <i>keyword</i> (*new)	Causes the compiler to issue compile-time messages for nonstandard language elements. keyword: <code>f90</code> , <code>f95</code> , <code>none</code> .	OFF (<code>-nostand</code> or <code>-stand none</code>)
-static	Prevents linking with shared libraries. Causes the executable to link all libraries statically.	OFF
-static-libcxa (*new)	Links the Intel <code>libcxa</code> C++ library statically. This option is the opposite of <code>-shared-libcxa</code> .	OFF
-std90	Causes the compiler to issue messages for language elements that are not standard in Fortran 90 (same as the <code>-stand f90</code> option).	OFF

<code>-std95</code> or <code>-std</code>	Causes the compiler to issue messages for language elements that are not standard in Fortran 95 (same as the <code>-stand f95</code> option). This option is set if you specify <code>-warn stderrors</code> .	OFF
<code>-syntax_only</code>	Specifies that the source file should be checked only for correct syntax (same as the <code>-syntax</code> and <code>-y</code> options). No code is generated, no object file is produced, and some error checking done by the optimizer is bypassed. This option lets you do a quick syntax check of your source file.	OFF
<code>-T file</code> (*new)	Tells the linker to read link commands from the specified <i>file</i> .	OFF
<code>-Tffile</code>	Specifies that <i>file</i> should be compiled as a Fortran source file. This option is useful when you have a file with a nonstandard filename suffix.	OFF
<code>-threads</code> (*new)	Specifies that multithreaded libraries should be linked. This option sets the <code>-reentrancy threaded</code> option.	OFF (<code>-nothreads</code>)
<code>-tpp1</code> Itanium compiler	Optimizes for the Intel® Itanium® processor.	OFF
<code>-tpp2</code> Itanium compiler	Optimizes for the Intel® Itanium® 2 processor. This is the default on Itanium-based systems.	ON
<code>-tpp{5 6 7}</code> IA-32 compiler	<p><code>-tpp5</code> optimizes for the Intel Pentium® processor.</p> <p><code>-tpp6</code> optimizes for the Intel Pentium Pro, Pentium II, and Pentium III processors.</p> <p><code>-tpp7</code> optimizes for the Intel® Pentium® 4 processors, Intel® Xeon(TM) processors, Intel® Pentium® M processors, and (*new) Intel processors code-named "Prescott." This is the default on IA-32 systems.</p>	<code>-tpp7</code>

<code>-traceback</code> (*new)	Tells the compiler to generate extra information in the object file to allow the display of source file traceback information at runtime when a severe error occurs.	OFF (<code>-notraceback</code>)
<code>-tune keyword</code> IA-32 compiler	Determines the version of the architecture for which the compiler generates instructions. keyword: <code>-tune pn1</code> - optimizes for the Intel® Pentium® processor. <code>-tune pn2</code> - optimizes for the Intel® Pentium® Pro, Intel® Pentium® II, and Intel® Pentium® III processors. <code>-tune pn3</code> - optimizes for the Intel® Pentium® Pro, Intel® Pentium® II, and Intel® Pentium® III processors. This is the same as specifying the <code>-tune pn2</code> option. <code>-tune pn4</code> - optimizes for the Intel® Pentium® 4 processor. This is the default.	<code>-tune pn4</code>
<code>-u</code>	Sets the default type of a variable to undefined (<code>IMPLICIT NONE</code>). This is the same as specifying the <code>-implicitnone</code> option.	ON
<code>-Uname</code>	Removes the predefined macro <i>name</i> .	OFF
<code>-unroll[n]</code>	<ul style="list-style-type: none"> - Use <i>n</i> to set maximum number of times to unroll a loop. - Omit <i>n</i> to let the compiler decide whether to perform unrolling or not. This is the default; the compiler uses default heuristics or defines <i>n</i>. - Use <i>n</i> = 0 to disable unroller. Itanium® compiler currently recognizes only <i>n</i> = 0; any other value is ignored. 	<code>-unroll</code>
<code>-uppercase</code>	Causes the compiler to ignore case differences in identifiers and to convert external names to uppercase (same as the <code>-names uppercase</code> option). The default is <code>-lowercase</code> (or <code>-names lowercase</code>).	OFF

-us	Tells the compiler to append an underscore character to external user-defined names (opposite of -nus). Specifying -us is the same as specifying the -assume underscore option.	ON
-use_asm	Tells the compiler to produce objects through the assembler.	OFF
-V	Displays compiler version information.	OFF
-v	Tells the driver that tool commands should be shown and executed.	OFF
-vec_report {0 1 2 3 4 5} IA-32 compiler	Controls amount of vectorizer diagnostic information as follows: n = 0: no information n = 1: indicates vectorized loops (default) n = 2: indicates vectorized and non-vectorized loops n = 3: indicates vectorized and non-vectorized loops and prohibiting data dependence information n = 4: indicates non-vectorized loops n = 5: indicates non-vectorized loops and prohibiting data dependence information.	-vec_report1
-vms	Causes the runtime system to behave like HP* Fortran for OpenVMS Alpha systems and VAX systems (VAX FORTRAN*) in various ways. See <i>Intel® Fortran Compiler Manpages</i> for details.	OFF
-w	Disables all warning messages (same as the -nowarn and -warn nogeneral options).	OFF
-W{n}	Disables warnings (n =0) or enables warnings (n =1). The default is -W1 (same as the -warn general option). -W0 is the same as specifying -warn nogeneral, -nowarn, or -w.	-W1
-w90	Suppresses warning messages about Fortran features that are deprecated or obsolescent in Fortran 95 (same as the -w95 option).	

-w95	Suppresses warning messages about Fortran features that are deprecated or obsolescent in Fortran 95 (same as the -w95 option).	OFF
-warn <i>keyword</i> (*new)	Specifies the level of warning messages issued by the compiler as indicated by a keyword: all, none, [no]alignments, [no]declarations, [no]errors, [no]general, [no]ignore_loc, [no]stderrs, [no]truncated_source, [no]uncalled, [no]unused, [no]usage. See <i>Intel® Fortran Compiler Manpages</i> for details.	OFF (-warn none or -warn nokeyword)
-what (*new)	Prints the version strings of the Fortran command and the compiler.	OFF
-Wl o1 [, o2, ...] (*new)	Passes options -o1, -o2, etc. to the linker for processing.	OFF
-Wp o1 [, o2, ...] (*new)	Passes options -o1, -o2, etc. to the preprocessor.	OFF
-X	Removes standard directories from the include file search path (same as the -nostdinc option). You can use the -X option with the -I option to prevent the compiler from searching the default path for include files and direct it to use an alternate path.	OFF

<p><code>-x{K W N B P}</code></p> <p>IA-32 compiler</p>	<p>Generates specialized code to run on processors supporting the extensions indicated by processor-specific <i>codes</i>:</p> <p>K – Intel® Pentium® III processors and compatible Intel processors.</p> <p>w – Intel Pentium 4 processors and compatible Intel processors.</p> <p>The *new codes: when the main program is compiled with one of these options, it will detect non-compatible processors and generate a fatal error message during execution. These options also enable new optimizations in addition to Intel processor specific-optimizations.</p> <p>(*new) N – Intel Pentium 4 processors and compatible Intel processors.</p> <p>(*new) B – Intel® Pentium® M and compatible Intel processors.</p> <p>(*new) P – Intel processors code-named "Prescott" and compatible Intel processors.</p> <p>See <i>Intel® Fortran Compiler Manpages</i> for more details.</p>	OFF
<p><code>-Xlinker val</code> (*new)</p>	<p>Pass <i>val</i> directly to the linker for processing.</p>	OFF
<p><code>-y</code></p>	<p>Specifies that the source file should be checked only for correct syntax (same as the <code>-syntax_only</code> and <code>-syntax</code> options).</p>	OFF
<p><code>-zero</code></p>	<p>Initializes to zero all local scalar variables of intrinsic type <code>INTEGER</code>, <code>REAL</code>, <code>COMPLEX</code>, or <code>LOGICAL</code>, which are saved and not already initialized.</p>	OFF (<code>-zero-</code>)

<code>-Zp{1 2 4 8 16}</code>	Aligns fields of records and components of derived types on the smaller of the size boundary specified or the boundary that will naturally align them (same as the <code>-align recnbyte</code> option). The <i>n</i> can be: 1, 2, 4, 8, or 16. If you do not specify <i>n</i> , you get <code>-Zp8</code> , which is the default.	<code>-Zp8</code>
------------------------------	---	-------------------

Windows* to Linux* Options Cross-reference

This section provides cross-reference table of the Intel® Fortran Compiler options used on the Windows* and Linux* operating systems. The options described can be used for compilations targeted to either IA-32 or Itanium®-based applications or both. See [Conventions Used in the Options Quick Guide Tables](#).

- Options specific to IA-32 architecture
- Options specific to the Itanium® architecture
- All other options are available for both IA-32 and Itanium architectures.

Note

The table is based on the alphabetical order of compiler options for Linux, second column.

Note

The value in the Default column is used for both Windows and Linux operating systems unless indicated otherwise.

For more details on each of these options refer to the *Intel® Fortran Compiler User's Guide, Volumes I and II*, and the *Intel® Fortran Compiler Manpages*.

Windows Option	Linux Option	Description	Default
/1	-1	Executes at least one iteration of DO loops (same as the <code>-onetrip</code> option). This option has the same effect as <code>-f66</code> or <code>-66</code> .	OFF
/4L {72 80 132}	-72, -80, -132	Treats the statement field of each fixed-form source line as ending in column 72 (the default), 80, or 132 (same as the <code>-extend_source</code> option).	/4L72 -72

<p>/align: keyword</p>	<p>-align keyword</p>	<p>Tells the compiler to align data items by adding padding bytes (same as the -align all option) as indicated by a keyword: all, none, [no]commons, dcommons, [no]recnbytes (n=1, 2, 4, 8, 16), [no]records, [no]sequence See <i>Intel® Fortran Compiler Manpages</i> for more details.</p>	<p>-align nocommons -align records -align nosequence -align rec8bytes</p>
<p>/Qansi_alias [-]</p>	<p>-ansi_alias</p>	<p>Tells the compiler to assume that the program adheres to the Fortran 95 Standard type aliasability rules.</p>	<p>-ansi_ alias-</p>
<p>None</p>	<p>-arch keyword IA-32 only</p>	<p>Determines the version of the architecture for which the compiler generates instructions. keyword: pn1 Optimizes for the Intel® Pentium® processor. pn2 Optimizes for the Intel® Pentium® Pro, Intel® Pentium® II, and Intel® Pentium® III processors. pn3 This is the same as specifying the -arch pn2 option. pn4 optimizes for the Intel® Pentium® 4 processor.</p>	<p>-arch pn4</p>

<code>/assume:bscc</code>	<code>-assume bscc</code>	Tells the compiler to treat the backslash character (\) as a C-style control (escape) character syntax in character literals. The default is <code>-assume nobbcc</code> , which tells the compiler to treat the backslash character as a normal character instead of a control character in character literals.	<code>-assume nobbcc</code>
<code>/assume: keyword</code>	<code>-assume keyword</code>	Specifies that assumptions are made by the optimizer and code generator according to the indicated keyword: none, [no]accuracy_sensitive, [no]bscc, [no]byterecl, [no]buffered_io, [no]dummy_aliases, [no]protect_constants, [no]source_include, [no]underscore. See <i>Intel® Fortran Compiler Manpages</i> for details.	<code>-assume none</code>
<code>/assume: cc_omp</code>	<code>-assume cc_omp</code>	Enables conditional compilation as defined by the OpenMP Fortran API. <code>-openmp</code> enables this option: sets <code>-assume cc_omp</code> .	<code>-assume cc_omp with -openmp specified</code>

<pre>/assume: minus0</pre>	<pre>-assume minus0</pre>	<p>Tells the compiler to use Fortran 95 standard semantics for the treatment of IEEE* floating value -0.0 in the <code>SIGN</code> intrinsic, if the processor is capable of distinguishing the difference between -0.0 and +0.0, and to write a value of -0.0 with a negative sign on formatted output.</p> <p>The default is <code>-assume nominus0</code>, which tells the compiler to use Fortran 90/77 standard semantics in the <code>SIGN</code> intrinsic to treat -0.0 and +0.0 as 0.0, and to write a value of -0.0 with no sign on formatted output.</p>	<p>OFF (<code>-assume nominus0</code>)</p>
<pre>/Qauto</pre>	<pre>-auto</pre>	<p>Places variables, except those declared as <code>SAVE</code>, on the runtime stack (same as <code>-automatic</code> or <code>-nosave</code>).</p>	<p><code>-auto_scalar</code> or: if you specify <code>-recursive</code> or <code>-openmp</code>, the default is <code>-auto</code></p>
<pre>/Qauto_ilp32 Itanium-based systems</pre>	<pre>-auto_ilp32 Itanium-based systems</pre>	<p>Specifies that the application cannot exceed a 32-bit address space, which allows the compiler to use 32-bit pointers whenever possible. To use this option, you must also specify <code>-ipo</code>.</p> <p>Using the <code>-auto_ilp32</code> option on programs that can exceed 32-bit address space (2^{**32}) may</p>	<p>OFF</p>

		cause unpredictable results during program execution.	
/Qauto_scalar	-auto_scalar	Makes AUTOMATIC all scalar local variables of intrinsic type INTEGER , REAL , COMPLEX , or LOGICAL . You cannot specify -save , -auto or -automatic with this option.	OFF
/Qautodouble	-autodouble	Defines real variables to be REAL(KIND=8) . This option is the same as specifying -r8 .	OFF
/automatic	-automatic	Places variables, except those declared as SAVE , on the runtime stack (same as -auto or -nosave). The default is -auto_scalar . However, if you specify recursive or -openmp , the default is -auto .	-auto_scalar or: if you specify -recursive or -openmp , the default is -auto .

<p>/Qax {K W N B P} IA-32 only</p>	<p>-ax {K W N B P} IA-32 only</p>	<p>Generates processor-specific code corresponding to one of <i>codes</i>:</p> <p>K - Intel® Pentium® III processors and compatible Intel processors.</p> <p>W - Intel Pentium 4 processors and compatible Intel processors.</p> <p>The *new codes: these options enable new optimizations in addition to Intel processor-specific optimizations.</p> <p>(*new) N - Intel Pentium 4 processors and compatible Intel processors.</p> <p>(*new) B - Intel® Pentium® M and compatible Intel processors.</p> <p>(*new) P - Intel processors code-named "Prescott" and compatible Intel processors.</p>	<p>OFF</p>
<p>None</p>	<p>-Bdynamic</p>	<p>Enables dynamic linking of libraries at runtime. Smaller executables are created than with static linking.</p>	<p>OFF</p>
<p>/[no]browser</p>	<p>None</p>	<p>Specifies that source browser information should be (or not) generated in the indicated file if present.</p>	<p>OFF (-no browser)</p>
<p>None</p>	<p>-Bstatic</p>	<p>Enables static linking of a user's library.</p>	<p>OFF</p>

/c	-c	Causes the compiler to compile to an object (.o) file only and not link.	OFF
/CB	-CB	Performs runtime checks on whether array subscript and substring references are within declared bounds. Same as -check bounds.	OFF
/ccdefault: keyword	-ccdefault keyword	Specifies the type of carriage control used for units 6 and *; keyword = default, fortran, list, or none.	-ccdefault default can be affected by the -vms option, see manpages
/check: keyword	-check keyword	Checks runtime conditions, according to the keyword: all, none, [no]arg_temp_created, [no]bounds, [no]format, [no]output_conversion	OFF (-nocheck or (-check none)
/cm	-cm	Suppresses all messages about questionable programming practices (same as the -warn nousage option).	OFF
/Qcommon_args	-common_args	Tells the compiler that dummy (formal) arguments to procedures share memory locations with other dummy arguments or with COMMON variables that are assigned. This is the same as specifying -assume dummy_aliases.	OFF

/Qcomplex_limited_range [-]	-complex_limited_range [-]	Enables the use of basic algebraic expansions of some arithmetic operations involving data of type COMPLEX. This can cause some performance improvements in programs that use a lot of COMPLEX arithmetic, but values at the extremes of the exponent range may not compute correctly.	OFF (-complex_limited_range-, option disabled)
/convert: keyword	-convert keyword	Specifies the format of unformatted files containing numeric data indicated in a keyword: big_endian, cray, fdx, fgx, ibm, little_endian, native, vaxd, vaxg. See <i>Intel® Fortran Compiler Manpages</i> for more details.	OFF
/Qcpp	-cpp	Runs the Fortran preprocessor on source files prior to compilation (same as the -fpp option).	OFF
/Dname /Dname[=value]	-Dname -Dname [value]	Defines the <i>name</i> as a definition to use with conditional compilation directives or the Fortran preprocessor (-fpp). The <i>value</i> can be an integer or it can be a character string delimited by double quotes; for example, -Dname="string". If no definition is given, the <name> is defined as "1".	OFF

/d_lines	-d_lines	Compiles debug statements (indicated by D in column 1); this is the same as specifying -DD.	-nod_lines
/Qd_lines	-DD	Compiles debugging statements indicated by the letter D in column 1 of the source code; this is the same as specifying -d_lines.	OFF
/double_size: size	-double_size size	Defines the size of DOUBLE PRECISION and DOUBLE COMPLEX declarations, constants, functions, and intrinsics. The size can be 64 (default) or 128.	-double_size 64
/Qdps[-]	-dps	Enable (default) or disable DEC* parameter statement recognition.	ON
/Bz	-dryrun	Specifies that driver tool commands should be shown but not executed.	OFF
None	-dynamic-linker(file)	Specifies a dynamic linker in file other than the default.	OFF
/Qdyncom:A,B,C	-dyncom "a,b,c"	Enables dynamic allocation of the specified COMMON blocks at run time. The quotes are required.	OFF
/E	-E	Causes the Fortran preprocessor to send output to stdout.	OFF
/4{Y N}s	-e90, -e95	Causes the compiler to issue errors instead of warnings for nonstandard Fortran 90 (-e90) or Fortran 95 (-e95). No such errors or warnings are issued by default.	OFF (/4Ns)

<code>/EP</code>	<code>-EP</code>	Causes the Fortran preprocessor to send output to stdout, omitting <code>#line</code> directives.	OFF
<code>/[no]error_limit:num</code>	<code>-[no]error_limit num</code>	Specifies the maximum number of error-level or fatal-level compiler errors allowed for a file specified on the command line. A maximum of 30 error-level and fatal-level messages are allowed before the compiler stops the compilation.	<code>-error_limit 30</code>
<code>/Qextend_source:[size]</code>	<code>-extend_source [size]</code>	Specifies the column number to use to end the statement field in fixed-form source files. <i>size</i> can be 72, 80, or 132. The default behavior is <code>-noextend_source</code> , which implies column 72. If you specify <code>-extend_source</code> with no size, the default is <code>-extend_source 132</code> . Specifying <code>-extend_source</code> sets the <code>-fixed</code> option.	<code>-noextend_source</code>
<code>/F</code>	<code>-F</code>	Causes the Fortran preprocessor to send output to a file (same as the <code>-preprocess_only</code> and <code>-P</code> options). To use this option, you must also specify <code>-fpp</code> .	OFF
<code>/f66</code> or <code>-66</code>	<code>-f66</code> or <code>-66</code>	Enforces FORTRAN-66 semantics.	OFF
<code>/f77rtl</code>	<code>-f77rtl</code>	Tells the compiler to use the runtime behavior of FORTRAN 77 instead of Intel® Fortran. This affects	OFF (<code>-no f77rtl</code>)

		some INQUIRE specifiers when the unit is not connected to a file, PAD= defaults to 'NO' for formatted input, NAMELIST input format is different, and NAMELIST and list-directed input of character strings must be delimited by apostrophes or quotes.	
/fast	-fast	Enhances speed across the entire program. Sets the following command options that can improve runtime performance: -O3, -ipo, and -static.	OFF
/FAc	-fcode-asm	Produces an assembly file with optional code annotations. To use this option, you must also specify -S.	OFF
None	-fcommon	Tells the compiler to treat common symbols as global definitions and to allocate memory for each symbol at compile time. This may permit the compiler to use the more efficient GP-relative addressing mode when accessing the symbol.	OFF (-fno-common)
/FI	-FI	Specifies source files are in fixed format (same as the -fixed option).	Based on file extension
/fixed	-fixed	Specifies source files are in fixed format. By default, source file format is determined by the file suffix.	Based on file extension

None	-fminshared	Tells the compiler to treat a compilation unit as a component of a main program and not to link it as a shareable object. Implies -fvisibility=protected	OFF
/Oa[-]	-fno-alias	Specifies that aliasing should not be assumed in the program.	-falias /Oa
/Ow[-]	-fno-fnalias	Specifies that aliasing should not be assumed within functions, but should be assumed across calls.	-ffnalias /Ow
/Qfnsplit	-fnsplit- Itanium-based systems	Enables function splitting (enabled with -prof_use). The default is -fnsplit-, which disables the splitting within a routine but leaves function grouping enabled.	/Qfnsplit- -fnsplit-
/Oy- IA-32 only	-fp IA-32 only	Disables using EBP as a general purpose register so it can be used as a stack frame pointer.	OFF
/Qfp_port	-fp_port IA-32 only	Rounds floating-point results after floating-point operations, so rounding to user-declared precision happens at assignments and type conversions; this has some impact on speed. The default is to keep results of floating-point operations in higher precision; this provides better performance but less consistent floating-point results.	OFF

/fpconstant	-fpconstant	Tells the compiler to extend the precision to double precision for single-precision constants assigned to double-precision variables.	-no fpconstant
/fpe:n	-fpen	<p>Specifies floating-point exception handling at run time for the main program, $n=0, 1, 3$.</p> <p>-fpe0 - floating underflow results in zero; all other floating-point exceptions abort execution;</p> <p>-fpe1 - floating underflow results in zero; all other floating-point exceptions produce exceptional values (signed Infinities or NaNs) and execution continues;</p> <p>-fpe3 - all floating-point exceptions produce exceptional values (signed infinities, denormals, or NaNs) and execution continues; this is the default. Also see -ftz.</p>	/fpe:3 -fpe3
None	-fpic, -fPIC	Generates position-independent code. Can also be specified as -fPIC.	OFF

/Qfpp:n	-fpp n	<p>Runs the Fortran preprocessor on source files prior to compilation.</p> <p>n=0: disables CVF and # directives (equivalent to -nofpp).</p> <p>n=1: enables CVF conditional compilation and # directives; when Fortran preprocessor is invoked, -fpp1 is the default</p> <p>n=2: enables only # directives;</p> <p>n=3: enables only CVF conditional compilation directives.</p>	OFF
/fpscomp [:keyword]	-fpscomp [keyword]	<p>Specifies a level of compatibility with Microsoft* Fortran PowerStation as indicated by a keyword: all, none, [no]filesfromcmd, [no]general, [no]ioformat, [no]libs, [no]logicals.</p>	<p>For all and nolibs: -fpscomp libs</p> <p>For the rest: -fpscomp none</p>
/Qfpstkchk IA-32 only	-fpstkchk IA-32 only	<p>Generates extra code after every function call to ensure that the FP (floating-point) stack is in the expected state. By default, there is no checking. So when the FP stack overflows, NaN value is put into FP calculations, and the program's results differ. Unfortunately, the overflow point can be far away from the point of the actual bug. The -fpstkchk option</p>	OFF

		places code that would access-violate immediately after an incorrect call occurred, thus making it easier to locate these issues.	
/FR	-FR	Specifies source files are in free format (same as the <code>-free</code> option).	Based on source file extension
None	-fr32 Itanium compiler	Disables use of high floating-point registers. Uses only the lower 32 floating-point registers.	OFF
/free	-free	Specifies source files are in free format. By default, source file format is determined by the file suffix.	Based on source file extension
/FAs	-fsource-asm	Produces an assembly file with optional code annotations. To use this option, you must also specify <code>-S</code> .	OFF
/Qftz[-]	-ftz[-]	Enables (or disables: <code>-ftz-</code>) floating underflow results set to zero. For Itanium-based systems only: option <code>-O3</code> sets <code>-ftz</code> on.	/Qftz- -ftz-
None	-fverbose-asm	Produces an assembly file with compiler comments, including options and version information. To use this option, you must also specify <code>-S</code> , which sets <code>-fverbose-asm</code> . If you do not want this default when you specify <code>-S</code> , specify <code>-fnoverbose-asm</code> .	-fnoverbose-asm
None	-fvisibility= <i>keyword</i>	The first form specifies the default visibility for global symbols using	OFF

	<code>-fvisibility-keyword=</code> <i>file</i>	<p>one of the five command line options corresponding to the keyword: <code>external</code>, <code>default</code>, <code>protected</code>, <code>hidden</code>, and <code>internal</code>.</p> <p>The second form specifies the visibility for symbols that are in a file (this form overrides the first form). The <i>file</i> is the pathname of a file containing the list of symbols whose visibility you want to set; the symbols are separated by whitespace (spaces, tabs, or newlines). See <i>Intel® Fortran Compiler Manpages</i>.</p>	
<code>/Zi, /Z7</code>	<code>-g</code>	<p>Produces symbolic debug information in the object file. The compiler does not support the generation of debugging information in assemblable files. If you specify the <code>-g</code> option, the resulting object file will contain debugging information, but the assemblable file will not.</p> <p>On IA-32 systems, specifying the <code>-g</code> or <code>-O0</code> option automatically enables the <code>-fp</code> option.</p>	OFF
<code>/help</code>	<code>-help</code>	Prints the list of compiler options.	OFF

<code>/I<i>dir</i></code>	<code>-I<i>dir</i></code>	Specifies a directory to add to the include path, which is used to search for module files (<code>USE</code> statement) and include files (<code>INCLUDE</code> statement).	OFF
None	<code>-i_dynamic</code>	Links Intel-provided libraries dynamically.	OFF
<code>/4I{2 4 8}</code>	<code>-i{2 4 8}</code>	Defines the default <code>KIND</code> for integer variables and constants to be 2, 4, and 8 bytes (same as <code>-integer_size {16 32 64})</code>	<code>/4I4</code> <code>-i4</code>
<code>/4{Y N}d</code>	<code>-implicit none</code>	Sets the default type of a variable to undefined (<code>IMPLICIT NONE</code>). Same as the <code>-u</code> option.	OFF
<code>/Qinline_debug_info</code>	<code>-inline_debug_info</code>	Preserves the source position of inlined code instead of assigning the call-site source position to inlined code.	OFF
<code>/intconstant</code>	<code>-int constant</code>	Tells the compiler to use FORTRAN 77 semantics, rather than Fortran 95/90 semantics, to determine the <code>KIND</code> for integer constants.	OFF
<code>/integer_size : size</code>	<code>-integer_size size</code>	Defines the size of <code>INTEGER</code> and <code>LOGICAL</code> variables. The <code>size</code> can be 16, 32, or 64.	<code>-integer_size 32</code>
<code>/Qip</code>	<code>-ip</code>	Enables single-file interprocedural optimizations.	OFF
<code>/Qip_no_inlining</code>	<code>-ip_no_inlining</code>	Disables full and partial inlining enabled by <code>-i8</code> . To use this option, you must specify <code>-ip</code> or <code>-ipo</code> .	OFF

/Qip_no_pinning IA-32 only	-ip_no_pinning IA-32 only	Disables partial inlining. To use this option, you must specify <code>-ip</code> or <code>-ipo</code> .	OFF
/QIPFfltacc[-] Itanium-based systems	-IPFfltacc[-] Itanium-based systems	Disables optimizations that affect floating-point accuracy. If the default setting is used (<code>-IPFfltacc-</code>), the compiler may apply optimizations that reduce floating-point accuracy. You can use <code>-IPFfltacc</code> or <code>-mp</code> to improve floating-point accuracy, but at the cost of disabling some optimizations.	-IPFfltacc-
/QIPFflt_eval_method0 Itanium-based systems	-IPFflt_eval_method0 Itanium-based systems	Directs the compiler to evaluate the expressions involving floating-point operands in the precision indicated by the variable types declared in the program. By default, intermediate floating-point expressions are maintained in higher precision.	OFF
/QIPFfma[-] Itanium-based systems	-IPFfma Itanium-based systems	Enables the combining of floating-point multiplies and add/subtract operations. Also enables the contraction of floating-point multiply and add/subtract operations into a single operation. The compiler contracts these operations whenever possible. However, if <code>-mp</code> is specified, these contractions are	-IPFfma-

		disabled.	
/QIPF_fp_speculation_mode Itanium-based systems	-IPF_fp_speculation_mode Itanium-based systems	Enables floating-point speculations with one of the following mode conditions: <i>fast</i> : Speculate floating-point operations. <i>off</i> : Disables speculation of floating-point operations. <i>safe</i> : Speculate only when safe. <i>strict</i> : This is the same as specifying off.	-IPF_fp_speculation_fast
/Qipo	-ipo	Enables mult file IP optimizations (between files). When you specify this option, the compiler performs inline function expansion for calls to functions defined in separate files. For this reason, it is important to compile the entire application or multiple, related source files together when you specify <code>-ipo</code> .	OFF
/Qipo_c	-ipo_c	Generates a multifile object file (<code>ipo_out.o</code>) that can be used in further link steps.	OFF
/Qipo_obj	-ipo_obj	Forces the generation of real object files. Requires <code>-ipo</code> .	IA-32: OFF Itanium Compiler: ON
/Qipo_S	-ipo_S	Generates a multi file assembly file (<code>ipo_out.s</code>) that can be used in further link steps.	OFF

<code>/Qivdep_parallel</code> Itanium-based systems	<code>-ivdep_parallel</code> Itanium-based systems	Tells the compiler that there is no loop-carried memory dependency in any loop following an IVDEP directive.	OFF
None	<code>-Kpic</code>	This is a deprecated option; it can also be specified as <code>-KPIC</code> . Use <code>-fpic</code> instead.	OFF
None	<code>-Ldir</code>	Tells the linker to search for libraries in <i>dir</i> before searching the standard directories.	OFF
None	<code>-lname</code>	Links with the library indicated in <i>name</i> .	OFF
<code>/Qlowercase</code>	<code>-lowercase</code>	Causes the compiler to ignore case differences in identifiers and to convert external names to lowercase (same as the <code>-names lowercase</code> option). This is the default.	Windows: OFF Linux: ON
<code>/iface:mixed_str_len_arg</code>	None	Tells the compiler that the hidden length passed for a character argument is to be placed immediately after its corresponding character argument in the argument list. The default (<code>-nomixed_str_len_arg</code>) places the hidden lengths in sequential order at the end of the argument list.	OFF
<code>/Fmfilename</code>	None	Instructs the linker to produce a map file.	OFF

<code>/module:dir</code>	<code>-module dir</code>	Specifies the directory <i>dir</i> where module (.mod) files should be placed when created and where they should be searched for (USE statement).	OFF
<code>/Op[-]</code>	<code>-mp</code>	Maintains floating-point precision (while disabling some optimizations). Restricts optimization to maintain declared precision and to ensure that floating-point arithmetic conforms more closely to the ANSI* and IEEE standards. For most programs, specifying this option adversely affects performance. If you are not sure whether your application needs this option, try compiling and running your program both with and without it to evaluate the effects on both performance and precision.	OFF
<code>/Qprec</code>	<code>-mp1</code>	Improves floating-point precision. This option disables fewer optimizations and has less impact on performance than <code>-mp</code> .	OFF
<code>/names:keyword</code> <i>d</i>	<code>-names keyword</code>	Specifies how source code identifiers and external names are interpreted as indicated by a keyword: <code>as_is</code> , <code>lowercase</code> , <code>uppercase</code>	OFF

/nbs	-nbs	Tells the compiler to treat a backslash as a normal character, not an escape character (same as the <code>-assume nobsc</code> option).	OFF
None	-no_cpprt	Prevents linking of the C++ runtime libraries.	OFF
/align:none	-noalign	Prevents the alignment of data items. This is the same as specifying <code>-align none</code> .	-align
/noaltparam	-noaltparam	Specifies that the alternate form of parameter constant declarations (without parenthesis) should not be recognized (same as the <code>-nodps</code> option). This form has no parentheses surrounding the list, and the form of the constant, rather than implicit or explicit typing, determines the variable data type.	-altparam
/Qnobss_init	-nobss_init	Places any variables that are explicitly initialized with zeros in the DATA section. By default, variables explicitly initialized with zeros are placed in the BSS section.	OFF
None	-no defaultlibs	Prevents the compiler from using standard libraries when linking.	OFF
/nodefine	-nodefine	Specifies that all preprocessor definitions apply only to <code>-fpp</code> and not to Intel Fortran conditional compilation directives.	OFF

/Qdps-	-nodps	Specifies that the alternate form of parameter constant declarations (without parenthesis) should not be recognized (same as -noaltparam).	-dps
None	-nofor_main	Specifies the main program is not written in Fortran, and prevents the compiler from linking <code>for_main.o</code> into applications.	OFF
/noinclude	-noinclude	Prevents the compiler from searching in <code>/usr/include</code> for files specified in an <code>INCLUDE</code> statement. You can specify the <code>-Idir</code> option along with this option. This option does not affect <code>cpp(1)</code> behavior, and is not related to the Fortran 95 and 90 <code>USE</code> statement.	ON
/Oi-	-nolib_inline	Disables inline expansion of intrinsic functions.	ON
/nologo	-nologo	Suppresses compiler version information.	OFF
None	-no_startfiles	Prevents the compiler from using standard startup files when linking.	OFF
None	-nostdinc	Remove standard directories from include file search path (same as the <code>-x</code> option.)	OFF
None	-nostdlib	Prevents the compiler from using standard libraries and startup files when linking.	OFF

None	-nus	Disables appending an underscore to external subroutine names.	OFF
/F <i>file</i> /Fo <i>file</i> /Fe <i>file</i>	-o <i>file</i>	Specifies the name for an output <i>file</i> .	OFF
/Od	-O0	Disables -O <i>n</i> optimizations. On IA-32 systems, this option sets the -fp option.	OFF
/O1	-O1	On IA-32 systems, enables optimizations for speed. Also disables intrinsic recognition and the -fp option. This option is the same as the -O2 option. On Itanium(R)-based systems, enables optimizations for server applications (straight-line and branch-like code with flat profile). Enables optimizations for speed, while being aware of code size. For example, this option disables software pipelining and loop unrolling.	OFF
/O2	-O2, -O	This option is the default for optimizations. However, if -g is specified, the default is -O0. On IA-32 systems, this option is the same as the -O1 option. On Itanium-based systems, enables optimizations for speed,	ON

		<p>including global code scheduling, software pipelining, predication, and speculation.</p> <p>On these systems, the <code>-O2</code> option enables inlining of intrinsics. It also enables the following capabilities for performance gain: constant propagation, copy propagation, dead-code elimination, global register allocation, global instruction scheduling and control speculation, loop unrolling, optimized code selection, partial redundancy elimination, strength reduction/induction variable simplification, variable renaming, exception handling optimizations, tail recursions, peephole optimizations, structure assignment lowering and optimizations, and dead store elimination.</p>	
--	--	--	--

/O3	-O3	<p>Enables -O2 optimizations plus more aggressive optimizations, such as prefetching, scalar replacement, and loop transformations. Enables optimizations for maximum speed, but does not guarantee higher performance unless loop and memory access transformation take place.</p> <p>On IA-32 systems, when the -O3 option is used with the -ax and -x options, it causes the compiler to perform more aggressive data dependency analysis than for -O2, which may result in longer compilation times.</p> <p>On Itanium-based systems, enables optimizations for technical computing applications (loop-intensive code): loop optimizations and data prefetch.</p>	OFF
/Qonetrip	-onetrip	Executes at least one iteration of DO loops (same as the -1 option). This option has the same effect as -f66 or -66.	OFF

/Qopenmp	-openmp	Enables the parallelizer to generate multithreaded code based on OpenMP* directives. The code can be executed in parallel on both uniprocessor and multiprocessor systems. The <code>-openmp</code> option works with both <code>-O0</code> (no optimization) and any optimization level of <code>-On</code> . Specifying <code>-O0</code> with <code>-openmp</code> helps to debug OpenMP applications.	OFF
/Qopenmp_report _report {0 1 2}	-openmp_report _report {0 1 2}	Controls the OpenMP parallelizer's level of diagnostic messages. 0 – Displays no diagnostic information. 1 – Displays diagnostics indicating loops, regions, and sections successfully parallelized. 2 – Displays the diagnostics specified by <code>-openmp_report1</code> plus diagnostics indicating MASTER constructs, SINGLE constructs, CRITICAL constructs, ORDERED constructs, ATOMIC directives, etc., successfully handled.	-openmp_report1
/Qopenmp_stubs	-openmp_stubs	Enables the compiler to generate sequential code. The OpenMP directives are ignored and a stub OpenMP	OFF

		library is linked.	
/Qopt_report	-opt_report	Generates an optimization report and directs to <code>stderr</code> unless <code>-opt_report_file</code> is specified.	OFF
/Qopt_report_filefilename	-opt_report_file filename	Specifies the <i>filename</i> to hold the optimizations report.	OFF
/Qopt_report_help	-opt_report_help	Lists the logical names of optimizers available for report generation (for <code>-opt_report_phase</code>).	OFF
/Qopt_report_level {min med max}	-opt_report_level {min med max}	Specifies the detail level of the optimizations report.	-opt_report_levelmin
/Qopt_report_phasephase	-opt_report_phasephase	Specifies the phase against which reports are generated. The compiler generates reports for the optimizer you specify in <i>phase</i> . This option can be used multiple times on the same command line to generate reports for multiple optimizers. Currently, the following optimizer reports are supported: ipo: Interprocedural Optimizer hlo: High Level Optimizer ilo: Intermediate Language Scalar Optimizer ecg: Code Generator omp: Open MP all: All phases When one of the above logical names for	OFF

		optimizers is specified for <i>phase</i> , all reports from that optimizer are generated.	
/Qopt_report_routine [<i>substring</i>]	-opt_report_routine [<i>substring</i>]	Generates a report on the routines containing the specified <i>substring</i> . If <i>substring</i> is not specified, reports from all routines are generated.	OFF
None	-P	Compiles and links for function profiling with gprof (1). This is the same as specifying -pg or -qp.	OFF
/P	-P	Causes the Fortran preprocessor to send output to a file (same as the -preprocess_only and -F options). To use this option, you must also specify -fpp.	OFF
/Qpad[-]	-pad	Enables the changing of the variable and array memory layout.	OFF (-nopad)
/Qpad_source	-pad_source	Specifies that fixed-form source records shorter than the statement field width are to be padded with spaces (on the right) to the end of the statement field. This affects the interpretation of character and Hollerith literals that are continued across source records.	OFF

<pre>/Qpar_ report{0 1 2 3}</pre>	<pre>-par_ report{0 1 2 3}</pre>	<p>Controls the auto-parallelizer diagnostic messages.</p> <p>0 – Displays no diagnostic information. 1 – Displays diagnostics indicating loops successfully auto-parallelized. This is the default. Issues a "LOOP AUTO-PARALLELIZED" message for parallel loops. 2 – Displays diagnostics indicating loops successfully auto-parallelized, as well as unsuccessful loops. 3 – Displays the diagnostics specified by <code>-par_report2</code> plus additional information about any proven or assumed dependencies inhibiting auto-parallelization (reasons for not parallelizing).</p>	<pre>-par _report1</pre>
------------------------------------	-----------------------------------	--	--------------------------

/Qpar _threshold{n}	-par _threshold {n}	Sets a threshold for the auto-parallelization of loops based on the probability of profitable execution of the loop in parallel. This option is used for loops whose computation work volume cannot be determined at compile-time. The threshold is usually relevant when the loop trip count is unknown at compile-time. n=0 to 100. The compiler applies a heuristic that tries to balance the overhead of creating multiple threads versus the amount of work available to be shared amongst the threads.	n=100
/Qparallel	-parallel	Enables the auto-parallelizer to generate multithreaded code for loops that can be safely executed in parallel. To use this option, you must also specify -O2 or -O3.	OFF
/Qpc{32 64 80} } IA-32 only	-pc32 -pc64 -pc80 IA-32 only	Enables floating-point significand precision control as follows: -pc32 to 24-bit significand -pc64 to 53-bit significand -pc80 to 64-bit significand See <i>Intel® Fortran Compiler Manpages</i> for more details.	/Qpc64 -pc64

/[no]pdbfile: [file]	None	Specifies that any debug-related information should (or should not) be generated to a program database file.	OFF (-nopdbfile)
None	-pg	Compile and link for function profiling with <code>gprof(1)</code> . This is the same as specifying <code>-p</code> or <code>-qp</code> .	OFF
/Qprec_div IA-32 only	-prec_div IA-32 only	Disables floating point division-to-multiplication optimization resulting in more accurate division results. Slight speed impact.	OFF
/Qprefetch[-] IA-32 only	-prefetch[-] IA-32 only	Enables or disables prefetch insertion (requires <code>-O3</code>).	OFF
None	-preprocess _only	Causes the Fortran preprocessor to send output to a file (same as the <code>-F</code> and <code>-P</code> options). To use this option, you must also specify <code>-fpp</code> .	OFF
/Qprof_dirdir	-prof_dirdir	Specifies a directory for profiling output files, <code>*.dyn</code> and <code>*dpi</code> .	OFF
/Qprof_file file	-prof_ filefile	Specifies a file name <i>file</i> for the profiling summary file.	OFF
/Qprof_format _32	-prof_format _32	Produces profile data with 32-bit counters; allows compatibility with earlier compilers. The default is to produce profile data with 64-bit counters to handle large numbers of events.	OFF
/Qprof_gen	-prof_gen	Instruments a program for profiling.	OFF

/Qprof_use	-prof_use	Enables use of profiling information during optimization.	OFF
None	-Qinstall, <i>dir</i>	Sets <i>dir</i> as a root directory for compiler installation.	OFF
/Qlocation, <i>tool,path</i>	-Qlocation, <i>tool,path</i>	Specifies an alternate version of a tool located at path.	OFF
/Qoption, <i>tool,opts</i>	-Qoption, <i>tool,opts</i>	Passes options, <i>opts</i> , to the tool specified by <i>tool</i> , which can be <i>fpp</i> , <i>f</i> , <i>c</i> , <i>asm</i> (on IA-32 systems), <i>ias</i> (on Itanium-based systems), or <i>link</i> .	OFF
None.	-qp	Compile and link for function profiling with <i>prof(1)</i> tool. This is the same as specifying <i>-p</i> or <i>-pg</i> .	OFF

/4R{8 16}	-r{8 16}	<p>Defines the KIND for real variables in 8 and 16 bytes.</p> <p>-r8: Defines REAL declarations, constants, functions, and intrinsics as DOUBLE PRECISION REAL*8, and defines COMPLEX declarations, constants, functions, and intrinsics as DOUBLE COMPLEX (COMPLEX*16). This option is the same as specifying -real_size 64 or -autodouble.</p> <p>-r16: Defines REAL and DOUBLE PRECISION declarations, constants, functions, and intrinsics as REAL*16, and defines COMPLEX and DOUBLE COMPLEX declarations, constants, functions, and intrinsics as COMPLEX*32. This option is the same as specifying -real_size 128.</p>	-r8
/Qrcd IA-32 only	-rcd IA-32 only	<p>Disables the change to truncation of the rounding mode for all floating-point calculations, including floating point-to-integer conversions. This option can improve performance, but floating-point conversions to integer will not conform to Fortran semantics.</p>	OFF

<code>/real_size: size</code>	<code>-real_size size</code>	Defines the size of REAL and COMPLEX declarations, constants, functions, and intrinsics. The <i>size</i> can be 32, 64, or 128.	<code>-real_size 32</code>
<code>/recursive</code>	<code>-recursive</code>	Specifies that all routines should be compiled for possible recursive execution. This option sets the <code>-auto</code> option.	OFF
<code>/reentrancy: keyword</code>	<code>-reentrancy keyword</code>	Specifies that the compiler should generate reentrant code that supports a multithreaded application. keyword: none, threaded, async.	OFF
<code>/S</code>	<code>-S</code>	Causes the compiler to compile to an assembly file (.s) only and not link.	OFF
<code>/Qsafe_cray _ptr</code>	<code>-safe_cray _ptr</code>	Specifies that CRAY* pointers do not alias with other variables.	OFF
<code>/Qsave</code>	<code>-save</code>	Places variables, except those declared as AUTOMATIC, in static memory (same as <code>-noauto</code> or <code>-noautomatic</code>). The default is <code>-auto_scalar</code> . However, if you specify <code>-recursive</code> or <code>-openmp</code> , the default is <code>-auto</code> .	OFF
<code>/Qscalar_rep [-] IA-32 only</code>	<code>-scalar_rep [-] IA-32 only</code>	Enables scalar replacement performed during loop transformation. To use this option, you must also specify <code>-O3</code> .	OFF (<code>-scalar_rep-</code>)

None	-shared	Tells the compiler to produce a dynamic shared object instead of an executable. On Itanium-based systems , you must specify -fpic for the compilation of each object file you want to include in the shared library.	OFF
None	-shared-libcxa	Links the Intel libcxa C++ library dynamically, overriding the default behavior when -static is used. This option is the opposite of -static-libcxa.	ON
/Qsox[-]	-sox[-] IA-32 only	Enables saving of the compiler options and version in the executable.	OFF (-sox-)
/stand:keyword	-stand keyword	Causes the compiler to issue compile-time messages for nonstandard language elements. keyword: f90, f95, none.	OFF (-nostand or -stand none)
None	-static	Prevents linking with shared libraries. Causes the executable to link all libraries statically.	OFF
None	-static-libcxa	Links the Intel libcxa C++ library statically. This option is the opposite of -shared-libcxa.	OFF
/stand:f90	-std90	Causes the compiler to issue messages for language elements that are not standard in Fortran 90 (same as the -stand f90 option).	OFF

/stand:f95	-std95 or -std	Causes the compiler to issue messages for language elements that are not standard in Fortran 95 (same as the -stand f95 option). This option is set if you specify -warn stderrs.	OFF
/Zs	-syntax_only	Specifies that the source file should be checked only for correct syntax (same as the -syntax and -y options). No code is generated, no object file is produced, and some error checking done by the optimizer is bypassed. This option lets you do a quick syntax check of your source file.	OFF
None	-T <i>file</i>	Tells the linker to read link commands from the specified <i>file</i> .	OFF
/Tffile	-Tffile	Specifies that <i>file</i> should be compiled as a Fortran source file. This option is useful when you have a file with a nonstandard filename suffix.	OFF
/threads	-threads	Specifies that multithreaded libraries should be linked. This option sets the -reentrancy threaded option.	OFF (-nothreads)
/G1 Itanium-based systems	-tppl Itanium-based systems	Optimizes for the Intel® Itanium® processor.	OFF

/G2 Itanium-based systems	-tpp2 Itanium-based systems	Optimizes for the Intel® Itanium® 2 processor. This is the default on Itanium-based systems.	/G2 -tpp2
/G{5 6 7} IA-32 only	-tpp{5 6 7} IA-32 only	-tpp5 optimizes for the Intel® Pentium® processor. -tpp6 optimizes for the Intel Pentium Pro, Pentium II, and Pentium III processors. -tpp7 optimizes for the Intel Pentium 4 processors, Intel® Xeon(TM) processors, Intel® Pentium® M processors, and Intel processors code-named "Prescott." This is the default on IA-32 systems.	/G7 -tpp7
/traceback	-traceback	Tells the compiler to generate extra information in the object file to allow the display of source file traceback information at runtime when a severe error occurs.	OFF

None	-tune <i>keyword</i> IA-32 only	Determines the version of the architecture for which the compiler generates instructions. keyword: -tune pn1: optimizes for the Intel® Pentium® processor. -tune pn2: optimizes for the Intel® Pentium® Pro, Intel® Pentium® II, and Intel® Pentium® III processors. -tune pn3: optimizes for the Intel® Pentium® Pro, Intel® Pentium® II, and Intel® Pentium® III processors. This is the same as specifying the -tune pn2 option. -tune pn4: optimizes for the Intel® Pentium® 4 processor. This is the default.	-tune pn4
/4{Y N}d	-u	Sets the default type of a variable to undefined (IMPLICIT NONE). This is the same as specifying the -implicitnone option.	ON
/Uname	-Uname	Removes the predefined macro <i>name</i> .	OFF
/Qunroll[n], /unroll[:n]	-unroll[n]	- Use <i>n</i> to set maximum number of times to unroll a loop. - Omit <i>n</i> to let the compiler decide whether to perform unrolling or not. This is the default; the compiler uses default heuristics or defines <i>n</i> . - Use <i>n</i> = 0 to disable	/Qunroll -unroll

		unroller. Itanium® compiler currently recognizes only <code>n = 0</code> ; any other value is ignored.	
<code>/Quppercase</code>	<code>-uppercase</code>	Causes the compiler to ignore case differences in identifiers and to convert external names to uppercase (same as the <code>-names uppercase</code> option). The default is <code>-lowercase</code> (or <code>-names lowercase</code>).	Windows*: ON Linux*: OFF
None	<code>-us</code>	Tells the compiler to append an underscore character to external user-defined names (opposite of <code>-nus</code>). Specifying <code>-us</code> is the same as specifying the <code>-assume underscore</code> option.	
None	<code>-use_asm</code>	Tells the compiler to produce objects through the assembler.	OFF
<code>/QV</code>	<code>-V</code>	Displays compiler version information.	OFF
None	<code>-v</code>	Tells the driver that tool commands should be shown and executed.	OFF

<pre>/Qvec_report {n} IA-32 only</pre>	<pre>-vec_report {n} IA-32 only</pre>	<p>Controls amount of vectorizer diagnostic information as follows:</p> <p>n = 0: no information</p> <p>n = 1: indicates vectorized loops (default)</p> <p>n = 2: indicates vectorized and non-vectorized loops</p> <p>n = 3: indicates vectorized and non-vectorized loops and prohibiting data dependence information</p> <p>n = 4: indicates non-vectorized loops</p> <p>n = 5: indicates non-vectorized loops and prohibiting data dependence information.</p>	<pre>-vec_report1</pre>
<pre>/Qvms</pre>	<pre>-vms</pre>	<p>Causes the runtime system to behave like HP* Fortran for OpenVMS Alpha systems and VAX systems (VAX FORTRAN*) in various ways. See <i>Intel® Fortran Compiler Manpages</i> for details.</p>	<pre>OFF</pre>
<pre>/w</pre>	<pre>-w</pre>	<p>Disables all warning messages (same as the <code>-nowarn</code> and <code>-warn nogeneral</code> options).</p>	<pre>OFF</pre>
<pre>/W{0 1}</pre>	<pre>-W{0 1}</pre>	<p>Disables warnings (n =0) or enables warnings (n =1). The default is <code>-w1</code> (same as the <code>-warn general</code> option). <code>-w0</code> is the same as specifying <code>-warn</code></p>	<pre>-w1</pre>

		nogeneral, -nowarn, or -w.	
/w90, /w95	-w90, -w95	Suppresses warning messages about Fortran features which are deprecated or obsolescent in either Fortran 90 or Fortran 95.	OFF
/warn:keyword	-warn keyword	Specifies the level of warning messages issued by the compiler as indicated by a keyword: all, none, [no]alignments, [no]declarations, [no]errors, [no]general, [no]ignore_loc, [no]stderrs, [no]truncated_source, [no]uncalled, [no]unused, [no]usage. See <i>Intel® Fortran Compiler Manpages</i> for details.	OFF (-warn none or -warn nokeyword)
/what	-what	Prints the version strings of the Fortran command and the compiler.	OFF
/link o1[,o2]	-Wl o1 [, o2...]	Passes options -o1, -o2, etc. to the linker for processing.	OFF
/fpp:"o1[,o2]"	-Wp o1 [, o2...]	Passes options -o1, -o2, etc. to the preprocessor.	OFF

/X	-X	Removes standard directories from the include file search path (same as the <code>-nostdinc</code> option). You can use the <code>-X</code> option with the <code>-I</code> option to prevent the compiler from searching the default path for include files and direct it to use an alternate path.	OFF
----	----	--	-----

<p>/Qx {K W N B P} IA-32 only</p>	<p>-x {K W N B P} IA-32 only</p>	<p>Generates specialized code to run on processors supporting the extensions indicated by processor-specific <i>codes</i>:</p> <p>K - Intel® Pentium® III processors and compatible Intel processors. W - Intel Pentium 4 processors and compatible Intel processors.</p> <p>The *new codes: when the main program is compiled with one of these options, it will detect non-compatible processors and generate a fatal error message during execution. These options also enable new optimizations in addition to Intel processor specific-optimizations.</p> <p>(*new) N - Intel Pentium 4 processors and compatible Intel processors.</p> <p>(*new) B - Intel® Pentium® M and compatible Intel processors.</p> <p>(*new) P - Intel processors code-named "Prescott" and compatible Intel processors.</p>	<p>OFF</p>
--	---	--	------------

<code>/link val</code>	<code>-Xlinker val</code>	Pass <i>val</i> directly to the linker for processing.	OFF
<code>/Zs</code>	<code>-y</code>	Specifies that the source file should be checked only for correct syntax (same as the <code>-syntax_only</code> and <code>-syntax</code> options).	OFF
<code>/Qzero[-]</code>	<code>-zero</code>	Initializes to zero all local scalar variables of intrinsic type INTEGER, REAL, COMPLEX, or LOGICAL, which are saved and not already initialized.	OFF (/Qzero-) (-zero-)
<code>/Zp {1 2 4 8 16}</code>	<code>-Zp {1 2 4 8 16}</code>	Aligns fields of records and components of derived types on the smaller of the size boundary specified or the boundary that will naturally align them (same as the <code>-align recnbyte</code> option). The <i>n</i> can be: 1, 2, 4, 8, or 16. If you do not specify <i>n</i> , you get <code>-Zp8</code> , which is the default.	<code>-Zp8</code>

Index

C

Compiler

Options Quick Reference
Alphabetical 13

Compiler..... 4

Compiler..... 13

D

Disclaimer 1

L

Linux..... 47

N

New compiler options 4

O

Options

Overviews 2

Quick reference alphabetical 13

Options 2

Options 4

Options 13

Options 47

Options Cross-reference..... 47

W

Windows* to Linux* options
crossreference..... 47