

Wind River Diab Compiler Options Reference, 5.9.7

1 August 2019

Copyright Notice

Copyright © 2019 Wind River Systems, Inc.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the prior written permission of Wind River Systems, Inc.

Wind River, Simics, Tornado, and VxWorks are registered trademarks of Wind River Systems, Inc. Helix, Pulsar, Rocket, Titanium Cloud, Titanium Control, Titanium Core, Titanium Edge, Titanium Edge SX, Titanium Server, and the Wind River logo are trademarks of Wind River Systems, Inc. Any third-party trademarks referenced are the property of their respective owners. For further information regarding Wind River trademarks, please see:

www.windriver.com/company/terms/trademark.html

This product may include software licensed to Wind River by third parties. Relevant notices (if any) are provided for your product on the Wind River download and installation portal, Wind Share:

<http://windshare.windriver.com>

Wind River may refer to third-party documentation by listing publications or providing links to third-party websites for informational purposes. Wind River accepts no responsibility for the information provided in such third-party documentation.

Corporate Headquarters

Wind River
500 Wind River Way
Alameda, CA 94501-1153
U.S.A.
Toll free (U.S.A.): +1-800-545-WIND
Telephone: +1-510-748-4100
Facsimile: +1-510-749-2010

For additional contact information, see the Wind River website:

www.windriver.com

For information on how to contact Customer Support, see:

www.windriver.com/support

Wind River Diab Compiler Options Reference, 5.9.7

1 August 2019

TABLE OF CONTENTS

1. Compiler Options.	1
1.1. Compiler Driver Options.	1
1.2. Compiler W Options.	14
1.3. Compiler X Options.	20
1.4. Compiler Driver X Options.	136
1.5. Compiler C-Plus-Plus X Options.	140
1.6. Compiler Option Macros.	146
1.7. Compiler Architecture-Specific Options.	148
1.7.1. ARM-Specific Compiler Options.	148
1.7.2. ARM-T-Specific Compiler Options.	151
1.7.3. ColdFire-Specific Compiler Options.	153
1.7.4. MIPS-Specific Compiler Options.	157
1.7.5. MIPS16-Specific Compiler Options.	160
1.7.6. RH850-Specific Compiler Options.	162
1.7.7. PowerPC-Specific Compiler Options.	166
1.7.8. PowerPCVLE-Specific Compiler Options.	177
1.7.9. TriCore-Specific Compiler Options.	187
1.7.10. Intelx86_Pentium-Specific Compiler Options.	191
2. Assembler Options.	193
2.1. Assembler Basic Options.	193
2.2. Assembler X Options.	198
2.3. Assembler Architecture-Specific Options.	207
2.3.1. ARM-Specific Assembler Options.	208
2.3.2. ColdFire-Specific Assembler Options.	210
2.3.3. MIPS-Specific Assembler Options.	216
2.3.4. RH850-Specific Assembler Options.	219
2.3.5. PowerPC-Specific Assembler Options.	221
2.3.6. TriCore-PCP-Specific Assembler Options.	225
2.3.7. TriCore-Specific Assembler Options.	226
2.3.8. Intelx86_Pentium-Specific Assembler Options.	227
3. Category and Numeric Code Indices.	231

3.1. Alphabetical Assembler Option Index.	231
3.2. Alphabetical Compiler Option Index.	235
3.3. C Plus Plus Index.	248
3.4. Checking and Profiling Index.	249
3.5. Compiler Command-Line Options Index.	250
3.6. Debugging Index.	250
3.7. Diagnostic and Lint Index.	251
3.8. Driver Index.	252
3.9. Instruction Index.	252
3.10. Memory Index.	255
3.11. Numerical Compiler Option Index.	256
3.12. Optimization Index.	264
3.13. Output Index.	266
3.14. Sections Index.	267
3.15. Syntax Index.	268
3.16. Type Index.	269

1. COMPILER OPTIONS

- [Compiler Driver Options on page 1](#)
- [Compiler W Options on page 14](#)
- [Compiler X Options on page 20](#)
- [Compiler Driver X Options on page 136](#)
- [Compiler C-Plus-Plus X Options on page 140](#)
- [Compiler Option Macros on page 146](#)
- [Compiler Architecture-Specific Options on page 148](#)

1.1. Compiler Driver Options

Table of Contents

- [-#, -##, -### on page](#)
- [-A- on page](#)
- [-A on page](#)
- [-b on page](#)
- [-c on page](#)
- [-C on page](#)
- [-D on page](#)
- [-e on page](#)
- [-E on page](#)
- [-@E on page](#)
- [-g on page](#)
- [-?, -??, -h, --help, -?X, -?W on page](#)
- [-l@ on page](#)
- [-l on page](#)
- [-i on page](#)
- [-M on page](#)
- [-@, -@@ on page](#)
- [-O on page](#)
- [-o on page](#)
- [-@O on page](#)
- [-P on page](#)
- [-S on page](#)
- [-t on page](#)
- [-U on page](#)
- [-V on page](#)
- [-v on page](#)
- [-VV on page](#)
- [-XO on page](#)
- [-Xsize-opt on page](#)
- [-Y l,dir on page](#)

-#, -##, -###

Print subprograms with arguments.

Synopsis

```
-#
```

Description

Print subprogram command lines with arguments as executed.

```
-##
```

Print subprogram command line with arguments without actually executing them.

```
-###
```

Print subprogram command lines with arguments inside quotes without executing them.

```
-A-
```

Ignore predefined macros and assertions.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-A-
```

Description

Cause the preprocessor to ignore all predefined macros and assertions.

```
-A
```

Define assertion.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-A pred (ident1) (ident2)
```

Description

Cause the assertion **pred(ident)** to be defined. For information about the **#assert** and **#unassert** preprocessor directives, see the *Additions to ANSI C and C++* chapter in the *Wind River Diab Compiler User's Guide* for your architecture.

```
-b
```

Inform the c++ compiler about the compilation of C++ runtime libraries.

Synopsis

```
-b
```

Description

This option is exclusively used to inform the c++ compiler that C++ runtime libraries will be compiled. This is an internal option only, not intended to be used by the customer.

-C

Stop after assembly, produce object file.

Synopsis

```
-C
```

Description

Stop after the assembly step and produce an object file. The default file extension is `.o`. See **-o** to use a different extension.

See Also

[-o on page](#)

-C

Pass along comments.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-C
```

Description

Cause the C processor to pass along all comments. Useful only in conjunction with **-E** or **-P**.

Note

The preprocessor may be used with any language supported by Wind River.

-C is not necessary when **-Xpass-source** is used to output source as comments when generating assembly output because in that case the source code is taken before preprocessing.

-D

Define preprocessor macro name.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-D name [ =definition]
```

Description

Define the preprocessor macro **name** as if by the **#define** directive. If no **-definition** is given, the value 1 is used.

Macros may be either function-like macros or object-like macros. Function-like macros take arguments; this sample macro converts inches to centimeters:

```
dcc -DIN_TO_CM(x)=((x)*2.54) foo.c
```

Note that, to prevent unexpected results, both the argument and the entire macro expression should be enclosed in parentheses.

Object macros do not take arguments:

`dcc -DYEAR_LENGTH=366 bar.c`

For rules about using spaces, quotations, and the like on the command line, please see the *Wind River Diab Compiler User's Guide* for your specific architecture.

-e

Change diagnostic severity level.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-esn[,n...]
```

Description

For each of one or more diagnostic message numbers **n** in the comma-separated list, change the severity level of the message to **s** where **s** is one of:

i

Information, equivalent to ignore.

w

Warning.

e

Error (continue compilation).

f

Fatal error (terminate immediately).

Each diagnostic message has the form:

"file", line #: severity-level (compiler: error #): message

Example:

"err1.c", line 2: warning (dcc:1025): division by zero

To raise the severity level of this message from "warning" to "error", invoke the compiler with the option **-ee1025**. To reduce the level to "ignore", use **-ei1025**.

Note

Some messages have a minimum severity level. The severity level of a message may be raised above its minimum but not lowered below it. Attempting to do so will generate warning 1641.

-Xmismatch-warning and **-Xmismatch-warning=2** override the **-e** option. If either form of **-Xmismatch-warning** is used, mismatched types will only produce a warning, even if **-e** is used to increase the severity level of the diagnostic.

See Also

[-Xmismatch-warning on page](#)

-E

Stop after preprocessor, write source to standard output.

Synopsis

```
-E
```

Description

Run only the preprocessor on the named files and send the output to the standard output. All preprocessor directives are removed except for line-number directives used by the compiler to generate line-number information. (To suppress line-number information, use **-Xpreprocessor-lineno-off**.) The source files do not require any particular suffix.

When **-E** is invoked, the preprocessor implicitly includes the `lpragma.h` file. To suppress inclusion of `lpragma.h`, use **-Xclib-optim-off**. For more on `lpragma.h`, see **-Xclib-optim-off**.

Note

Source file and line number information are displayed the following way

```
#<lineNumber> <fileName> <inclusionLevel> <flags>
```

Where inclusion level starts by a value of "0". Files which will be included from the source file are marked by a value of "1".

It's value will be incremented by nested include files and represent the corresponding inclusion level.

In case of <flags> we exclusively generate the value of "3" to denote system header files (`#include <file.h>`).

See Also

[-Xclib-optim-off on page](#)

[-P on page](#)

[-Xpreprocessor-lineno-off on page](#)

-@E

Redirect standard error to file.

Synopsis

```
-@E=file
-@E+file
```

Description

Redirect any error output to the given file. **-@E=** creates a new file, deleting an existing file if necessary. **-@E+** appends to an existing file.

-g

Generate symbolic debugger information.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-gn
```

Description

The several **-gn** options enable generation of varying levels of debugging information. If optimization options are also present (**-O** or **-XO**), optimization will be affected as described.

The **-gn** options may be specified at the beginning of a source file using a pragma statement. For example:
`#pragma option -gn`

-g

Same as **-g2**.

-g0

Do not generate symbolic debugger information. This is the default. No effect on optimization.

-g1

Generate symbolic debugger information, but leave out line number information.

-g2

Generate symbolic debugger information.

Do most target-independent optimizations, but disable optimizations that cannot be properly described using the debugging data format.

See also the *Optimization* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

See **-Xoptimized-debug-off** for how to disable optimizations that interfere with debugging.

-g3

Generate symbolic debugger information and do most optimizations. Highly optimized code can be difficult to debug. To debug inlined functions you must also add the option **-Xdebug-inline-on**.

This option is intended to serve as a lightweight debug option, adding debug information with minimal effect on the generated code. It is useful for situations in which you do not want to maintain separate "debug" and "production" builds.

Note that using debug and optimization options together will generate a binary file that differs from one generated with only optimization options.

See Also

[-Xoptimized-debug-off on page](#)

[-Xdebug-inline-on on page](#)

-?, -??, -h, --help, -?X, -?W

Show information about compiler options.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-?[option]
-??
-h
--help
```

Description

-?

-h

--help

Show synopsis of commonly used compiler options.

-??

-h?

Show synopsis of less frequently used options.

-?W

-hW

Show synopsis of **-W** options.

-?X

-hX

Show synopsis of **-X** options.

-?Xstring

Show synopsis of **-X** options whose names contain the specified string. For example, entering **dcc -?Xbss** returns information about **-Xbss-off** and **-Xbss-common-off**.

-I@

Control search for user-defined header files.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-I@
```

Description

Search for user-defined header files (those enclosed in double quotes (")) in the order specified only by **-I** options (modified by **-Y** **I** options if any). That is, do not search the current directory by default; search the current directory only when an **-I@** option is encountered. Example:

```
dcc -Iabc -I@ -Idef file.c
```

will result in a search order of:

- the directory abc
- the current directory
- the directory def

-I

Specify directory for header files.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-I dir
```

Description

Add **dir** to the list of directories to be searched for header files. A full pathname is allowed. More than one **-I** option can be given.

For an **#include** "file" directive, search for the file in the following locations:

1. The directory of the file containing the include directive. Thus, if an **#include** directive includes a path, that path defines the current directory for **#include** directives in the included file. Example (using UNIX notation):

Assume file `f1.c` contains:

```
#include "p1/h1.h"
#include "h3.h"
```

and file `h1.h` contains:

```
#include "h2.h"
```

The search for `h2.h` will begin in directory **p1**; the search for `h3.h` will begin in the directory containing `f1.c`.

2. Directories given by the **-ldir** option, in the order encountered.
3. The directory or directories given by either:
 - any **-Y I** option appearing prior to the **-I** option
 - or –
 - versionDir/include (UNIX)
 - versionDir\include (Windows)

(The **-Y I** option effectively replaces the **versionDir** directory. See **-Y I,dir**.)
 For an **#include<file>** directive, search only the second and third locations.

See Also

[-Y I,dir on page](#)

-i

Modify header file processing.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-i file1=file2
-i file1=
-i =file2
```

Description

Substitute **file2** for **file1** in an **#include** directive.

If **file2** is empty, ignore any **#include** directive for **file1**.

If **file1** is empty, include **file2** before processing any other source file.

The **-i** option is disabled by **-P**.

See Also

[-P on page](#)

-M

Specify pathname of target-spec file.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-M target-spec
```

Description

This option is primarily for use by Wind River. Specify the pathname of the **target-spec** file to the compiler. This file contains the target description and is read by the compiler at startup. If the **-M** option is set more than once, the final setting is used.

-@, -@@

Read command-line options from file or variable.

Synopsis

```
-@name
```

Description

Read command-line options from either a file or an environment variable. When **-@name** is encountered on the command line, the driver first looks for an environment variable with the given name and substitutes its value. If an environment variable is not found then the driver tries to open a file with given **name** and substitutes the contents of the file. If neither an environment variable or a file can be found, an error message is issued and the driver terminates.

-@@name

Same as **-@name**; also prints all command-line options on standard output.

-O

Optimize code.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-O, -O0, -O1, -O2, -O3, -O4, -Os
```

Description

Optimize code. Either this or **-XO** must be present to enable optimization and to invoke the reorder program. See **-XO** for the difference between these options.

Set the level of effort to optimize for speed.

Increased levels apply greater compilation effort. Unless restricted by the **-Os** (or **-Xsize-opt**) flags, higher levels may result in larger code size.

-O0: Disable optimization

-O1: Enable basic optimization (equivalent to **-XO**)

-O2: Enable moderate optimization

-O3: Enable advanced optimization

-O4: Enable extensive optimization

-Os - Identical to **-Xsize-opt**. Optimize for size rather than speed when there is a choice

Optimization switches **-XO** versus **-O** with or without **-Xsize-opt**

Numerical Value	Name	-XO	-XO -Xsize-opt	-O	-O -Xsize-opt
X6	-Xtest-at-	both	both	bottom	bottom
X15	-Xunroll	2	1	2	1

X16	-Xunroll-size	20	1	20	1
X18	-Xstring-align	4	1	4	1
X19	-Xinline	40	10	10	10
X25	-Xopt-count	2	2	1	1
X29	-Xrestart	Used	Used	Not used	Not used
X424	-Xparse-count	600000	600000	300000	300000

See Also

[-XO on page](#)
[-Xsize-opt on page](#)
[-Xtest-at-both on page](#)
[-Xunroll on page](#)
[-Xstring-align on page](#)
[-Xinline on page](#)
[-Xopt-count on page](#)
[-Xparse-count on page](#)
[-Xrestart on page](#)

-O

Specify output file.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-o file
```

Description

Output to the given file instead of the default. This option works with the **-P**, **-S** and **-c** options as well as when none of these are specified. When compiling **file.ext** the following filenames are used by default if the **-o** option is not given:

-P	file.i
-S	file.s
-c	file.o
not -P , -S , or -c	a.out

-@O

Redirect standard output to file.

Synopsis

```
-@O=file  
-@O+file
```

Description

Redirect standard output to the given file. **-@O=** creates a new file, deleting an existing file if necessary. **-@O+** appends to an existing file.

-P

Stop after preprocessor, produce source file.

Synopsis

```
-P
```

Description

Stop after the preprocessor step and produce a source file. The default file extension is **.i**. See **-o** to use a different extension.

Unlike with the option **-E**, the output will not contain any preprocessing directives, and the output is written to a file instead of standard out. See **-o** to use a different extension.

When this option is used, the compiler driver does not invoke the assembler or linker. Thus, any switches intended for the assembler or linker must be given separately on command lines which invoke them. The **-P** option also disables **-i**.

When **-P** is invoked, the preprocessor implicitly includes the `lpragma.h` file. To suppress inclusion of `lpragma.h`, use **-Xclib-optim-off**. For more on `lpragma.h`, see **-Xclib-optim-off**.

See Also

[-o on page](#)
[-E on page](#)
[-Xclib-optim-off on page](#)

-S

Stop after compilation, produce assembly file.

Synopsis

```
-S
```

Description

Stop after the compilation step and produce an assembly source code file with the default file extension **.s** (unless modified by **-o**). If **-Xshow-configuration=1** is enabled, the assembly file contains a list of options in effect during compilation.

See Also

[-Xshow-configuration on page](#)

-t

Select the target processor.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-t tof:environ
```


Description

Select the target processor with **t** (a several character code), the object format with **o** (a one letter code), the floating point support with **f** (**H** for hardware, **S** for software, and **N** for none), and libraries suitable for the target environment with **environ**.

To determine the proper **tof**, execute **dctrl -t** to interactively display all valid combinations.

See also the *Selecting a Target and Its Components* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-U

Undefine preprocessor macro name.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-U name
```

Description

Undefine the preprocessor macro **name** as if by the **#undef** directive.

-V

Display current driver version number.

Synopsis

```
-V
```

Description

Display the current version number of the driver.

See Also

[-VV on page](#)

-V

Run driver in verbose mode.

Synopsis

```
-V
```

Description

Run the main driver program in verbose mode, printing a message as each subprogram is started.

-VV

Display current program version numbers.

Synopsis

```
-VV
```

Description

Display the current version number of the driver and the version number of all subprograms. Do not complete the compilation.

See Also

[-V on page](#)

-XO

Enable extra optimizations.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-XO
-X26
```

Description

Enable an extended set of optimizations. See **-O** for details.

See Also

[-O on page](#)

-Xsize-opt

Optimize for size rather than speed.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xsize-opt
-X73
```

Description

Optimize for size rather than speed when there is a choice. Optimizations affected include inlining, loop unrolling, and branch to small code. For details about the used options, see **-O**. For character arrays, **-Xstring-align=value** will override **-Xsize-opt**. See the description of array alignment in the *Internal Data Representation* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

The optimization of performing integer division by a reciprocal-multiply generates more instructions than using a divide instruction, and so is disabled when **-Xsize-opt** is specified. Use **-Xint-reciprocal** to allow divide-by-reciprocal-multiply even when optimizing for size (i.e., when **-Xsize-opt** is specified).

Note

-Xsize-opt without activating optimizations by either option -O or -XO doesn't have any effect. However you may turn off optimizing for size rather than speed by setting #pragma option -Xsize-opt=0 as the very first line on top of a specific module.

See Also

[-O on page](#)
[-Xint-reciprocal on page](#)
[-Xstring-align on page](#)

-Y I,dir

Specify default header file search path.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-Y I, [+]dir
```

Description

Use **dir** as the default directory or directories to search for header files specified with the **-I** option. A full pathname is allowed; also, a colon-separated path list may be supplied for multiple directories. Must occur prior to a **-I** option to be effective for that option.

Use the optional **+** character as the first character of the path to append the path to the current (possibly empty) default path. Subsequent use of **-Y I** without the **+** character overwrites the system header path.

See Also

[-I on page](#)

1.2. Compiler W Options

Table of Contents

- [-W a,-W :as: on page](#)
- [-W D on page](#)
- [-Wl, -W:ld: on page](#)
- [-W m on page](#)
- [-W s on page](#)
- [-W x,arguments on page](#)
- [-W x.ext on page](#)
- [-W x on page](#)

-W a,-W :as:

Pass arguments to the assembler.

Synopsis

```
-W a, arg1[, arg2...]
-W :as:, arg1[, arg2...]
```

Description

Pass the arguments to the assembler. Example:

-Wa,-l

-W:as:,-l

Pass the option “-l” (lower case letter L) to the assembler to get an assembler listing file.

-W D

Define configuration variable.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-W Dname= value
```

Description

Set a configuration variable equal to a value for use during configuration file processing.

More than one **-WD** option can be used to set several variables. The effect is as if an assignment statement for each such **-WD** variable had been added to the beginning of the main configuration file.

-Wl, -W:ld:

Pass arguments to linker.

Synopsis

```
-Wl, arg1[, arg2...]
-W:ld:, arg1[, arg2...]
```

Description

Pass the arguments to the linker.

Any option which is not recognized by the driver or compiler is automatically passed to the linker. **-Wl** may be used to pass options to third-party linkers in cases where such an option resembles a driver or compiler option. See **-Xforeign-as-ld**.

Example:

-Wl,-m or -W:ld:,-m

Pass the option **-m** to the linker to get a link map.

Please see the *Linker User's Guide* for information on passing compiler options with arguments to the linker.

See also option **-m** in the *Linker User's Guide*.

See Also

[-Xforeign-as-ld on page](#)

-W m

Specify linker command file.

Synopsis

```
-W mfile
```

Description

Use the given linker command file instead of the default **versionDir/conf/default.dld**.

Note

To suppress use of the `default.lnk` file, specify just **-Wm** with no **file** on the command line.

-W s

Specify startup (`crt0.o`) module.

Synopsis

```
-W sfile
```

Description

Use the given object file instead of the default startup file (`crt0.o`). Additional object files to be loaded along with the startup file and before any other files can be given separated by commas.

Note

To provide a `crt0.s` file or substitute to be assembled on the command line, or to use an existing non-default `crt0.o` file or substitute, specify just **-Ws** with no name to suppress use of the default.

-W x,arguments

Pass arguments to subprogram.

Synopsis

```
-W x,arg1[, arg2...]
```

Description

Pass the arguments to the subprogram designated by **x**. **x** is one of the following:

:cpp, **p**

The preprocessor. The preprocessor is incorporated in the compiler, so this becomes a synonym for **0**.

0

The compiler implied by the extension of the source file.

:c:

The C compiler.

:c++:

The C++ compiler.

a, **:as**:

The assembler. See **-W a,args**.

l, :ld:

The linker. See **-W l,args**.

L

The object converter. Usually not implemented. If given, it will execute after the linker.

1

For all architectures except ARM, MIPS, and PPC: The reorder program. Specifying **-W1** with no substitute program name will disable the reorder program.

For ARM, MIPS, and PPC: The llopt assembly-level optimizer. Specifying **-W1** with no substitute program will disable llopt.

2 – 6

Other filter programs; usually not implemented. For ARM, MIPS, and PPC: **-W2** invokes the reorder program. **-W1** and **-W2** are only executed if **-O** or **-XO** is given. They process the output from the compiler. **-W3** and **-W4** are always executed if given and process the output from the compiler. **-W5** and **-W6** process the input to the assembler.

Example:

-W:as:-l or **-Wa,-l**

Pass the option **"-l"** (lower case letter L) to the assembler to get an assembler listing file.

See Also

[-W a,args on page](#)

-W l,args

-W x.ext

Associate source file extension.

Synopsis

```
-W x.ext
```

Description

Associate a source file extension with a tool; that is, indicate to the main driver program **dcc** or **dplus** which tool should be invoked for an input file with a particular extension. **ext** specifies the extension and **x** specifies a tool, as follows:

0

The compiler implied by the extension of the source file.

:c:

The C compiler.

:c++:

The C++ compiler.

:as: a

The assembler.

:pas: , A

Preprocessor and assembler: both the preprocessor and assembler will be applied to the source. Allows use of preprocessor directives with assembly language. Note: when hash '#' characters will become used as assembly comments the preprocessor will generate warning (dcc:1750): # directive not understood You may however use C++ style backslashes '/' as comments with your assembly files in this case.

Example:

-W:as:.asm

Specify that `file.asm` is an assembly source file.

-W x

Substitute program or file for default.

Synopsis

```
-W xname[ =value]
```

Description

Except for the common cases **-W m** and **-W s** documented above, this option is primarily for use by Wind River.

Use the given program or file instead of the default program or file for the case indicated by **x**. Some cases take the form **-Wxname=value**. **x** is one of the following:

:as:, **a**

The assembler.

C

The configuration file to be used. The default is `dtools.conf` (`DTOOLS.CON` for Windows) in the **versionDir**/`conf` subdirectory.

:cpp:, **p**

The C preprocessor. The preprocessor is incorporated in the compiler, so this becomes a synonym for 0.

:c:

The C compiler.

:c++:

The C++ compiler.

c

Pass the string following the **-Wc** exactly as if it is as an option to the linker. More than one option can be given following **-Wc**, separated by commas.

Examples:

If you might compile and link a simple hello world example by `dcc` or `dplus` for instance by
`dcc -# -tPPCVLEEN:windiss hello.c`

library **libc.a** will become linked in by option **-lc** as you can see below:

```
dld -Y P,version_dir\PPCVLEEN\windiss:version_path\PPCVLEEN:version_path\PPCVLEE\windiss:version_path\PPCVLEE -
l:windiss/crt0.o -tPPCVLEEN:windiss hello.o -Xlibc-old
-lc version_path\conf\windissvle.dld
```

If you'll modify this call to

```
dcc -# -tPPCVLEEN:windiss -Wc-lproj hello.c
```

one can see that library **libc.a** will be replaced by library **libproj.a**

```
dld -Y P,version_dir\PPCVLEEN\windiss:version_path\PPCVLEEN:version_path\PPCVLEE\windiss:version_path\PPCVLEE -
l:windiss/crt0.o -tPPCVLEEN:windiss hello.o -Xlibc-old
-lproj version_path\conf\windissvle.dld
```

If you might modify this call to

```
dcc -# -tPPCVLEEN:windiss -Wc-lc,-lproj hello.c
```

library **libroj.a** will become linked in in addition to **libc.a** :

```
dld -Y P;version_dir\PPCVLEEN\windiss:version_path\PPCVLEEN:version_path\PPCVLEE\windiss:version_path\PPCVLEE -
l:windiss/crt0.o -tPPCVLEEN:windiss hello.o -Xlibc-old
-lc -lproj version_path\conf\windissvle.dld
```

However no library will become linked in at all if you don't specify any argument to option **-Wc** by
 dcc -# -tPPCVLEEN:windiss -Wc hello.c

results in
 dld -Y P;version_dir\PPCVLEEN\windiss:version_path\PPCVLEEN:version_path\PPCVLEE\windiss:version_path\PPCVLEE -
 l:windiss/crt0.o -tPPCVLEEN:windiss hello.o -Xlibc-old
 version_path\conf\windissvle.dld

Note: The examples were written for ppc architecture. For different targets you have to specify your specific target. The linker **-l** option is the more usual way to specify libraries.

D

See **-WD**.

d

C++library. The default is **-ld**. See "**c**" for meaning of **-ld** and additional rules.

:ld; 1

The linker.

L

The object converter; will execute after the linker.

m

See **-W m**.

s

See **-W s**.
 The compiler implied by the extension of the source file.

1

ColdFire, TriCore, x86:

The **reorder** program. Specifying **-W1** with no substitute program name will disable the **reorder** program.

ARM, MIPS, PowerPC, RH850:

The **llopt** assembly-level optimizer.

2 - 6

Other filter programs. **-W1** and **-W2** execute if **-O** or **-XO** is given and process the output from the compiler. **-W3** and **-W4** also process the output from the compiler. **-W5** and **-W6** process the input to the assembler. For ARM, MIPS, and PowerPC, **-W2** invokes the **reorder** program.

Example:

```
-W:ld:/usr/lib/dcc/3.6e/bin/dld
```

Use an old version of the linker.

See Also

[-W D on page](#)

[-W m on page](#)

[-W s on page](#)

1.3. Compiler X Options

Table of Contents

- [-Xabsolute18-const, -Xabsolute18-data on page](#)
- [-Xaddr-code on page](#)
- [-Xaddr-const on page](#)
- [-Xaddr-data on page](#)
- [-Xaddr-sconst on page](#)
- [-Xaddr-sdata on page](#)
- [-Xaddr-string on page](#)
- [-Xaddr-user on page](#)
- [-Xaggressive-static-live on page](#)
- [-Xalign-functions on page](#)
- [-Xalign-loop-to-cache on page](#)
- [-Xalign-min on page](#)
- [-Xalign-min-hardware-float on page](#)
- [-Xalternate-coloring on page](#)
- [-Xargs-in-regs on page](#)
- [-Xargs-not-aliased on page](#)
- [-Xarray-align-min on page](#)
- [-Xbit-fields-access-as-byte, -Xbit-fields-access-as-type on page](#)
- [-Xbit-fields-compress on page](#)
- [-Xbit-fields-signed, -Xbit-fields-unsigned on page](#)
- [-Xblock-count on page](#)
- [-Xbool-is-... on page](#)
- [-Xbottom-up-init on page](#)
- [-Xbss-off, -Xbss-common-off on page](#)
- [-Xbss-zero-init-const-on, -Xbss-zero-init-data-off on page](#)
- [-Xcga-min-use on page](#)
- [-Xcharset-ascii on page](#)
- [-Xchar-signed, -Xchar-unsigned on page](#)
- [-Xclib-optim-off on page](#)
- [-Xcode-absolute... on page](#)
- [-Xcode-factor on page](#)
- [-Xcode-relative... on page](#)
- [-Xcomdat on page](#)
- [-Xcomdat-info-file on page](#)
- [-Xcompress-prolog-epilog on page](#)
- [-Xconst-in-... on page](#)
- [-Xcpp-cr-line-terminator on page](#)
- [-Xcpp-dump-symbols on page](#)
- [-Xcpp-no-space on page](#)
- [-Xc++-template-name-lookup-old on page](#)
- [-Xdata-absolute... on page](#)
- [-Xdata-relative... on page](#)
- [-Xdebug-align on page](#)
- [-Xdebug-dwarf1 on page](#)
- [-Xdebug-dwarf2 on page](#)
- [-Xdebug-dwarf2-extensions-off on page](#)
- [-Xdebug-dwarf3 on page](#)
- [-Xdebug-inline-on on page](#)
- [-Xdebug-local-all on page](#)
- [-Xdebug-local-cie on page](#)
- [-Xdebug-mode on page](#)
- [-Xdebug-source-line-barriers-off on page](#)
- [-Xdebug-source-line-barriers-on on page](#)

- [-Xdebug-struct-all](#) on page
- [-Xdebug-struct-compact](#) on page
- [-Xdiagnose-inline...](#) on page
- [-Xdialect...](#) on page
- [-Xdialect-c++...](#) on page
- [-Xdisable-keywords](#) on page
- [-Xdollar-in-indent](#) on page
- [-Xdouble-avoid](#) on page
- [-Xdouble-error](#) on page
- [-Xdouble-warning](#) on page
- [-Xdynamic-init](#) on page
- [-Xendian-little](#) on page
- [-Xenum-bitfields](#) on page
- [-Xenum-is-...](#) on page
- [-Xexceptions-...](#) on page
- [-Xexplicit-inline-factor](#) on page
- [-Xextend-args](#) on page
- [-Xfeedback](#) on page
- [-Xfeedback-frequent](#) on page
- [-Xfeedback-seldom](#) on page
- [-Xforce-declarations](#) on page
- [-Xforce-prototypes](#) on page
- [-Xfp-fast, -Xfp-normal, -Xfp-pedantic](#) on page
- [-Xfp-long-double-off, -Xfp-float-only](#) on page
- [-Xfp-min-prec...](#) on page
- [-Xframe-info](#) on page
- [-Xfull-pathname](#) on page
- [-Xgcc-options-off](#) on page
- [-Xgcc-options-on](#) on page
- [-Xgcc-options-verbose](#) on page
- [-Xgnu-asm-usage](#) on page
- [-Xhll-symbol](#) on page
- [-Xident-off](#) on page
- [-Xieee754-pedantic](#) on page
- [-Xif-conversion](#) on page
- [-Ximport](#) on page
- [-Xincfile-missing-ignore](#) on page
- [-Xinit-locals](#) on page
- [-Xinit-section](#) on page
- [-Xinit-section-default-pri](#) on page
- [-Xinit-value](#) on page
- [-Xinline](#) on page
- [-Xinline-asm-off](#) on page
- [-Xinline-explicit-force](#) on page
- [-Xinline-single-use-statics, -Xinline-single-use-statics-off](#) on page
- [-Xinline-volatile-off](#) on page
- [-Xinline-without-body-warn](#) on page
- [-Xinsert-nop-end](#) on page
- [-Xinsert-nop-init](#) on page
- [-Xint-reciprocal](#) on page
- [-Xintrinsic-mask, -Xdisable-intrinsic-mask](#) on page
- [-Xkeep-function-asm](#) on page
- [-Xkeywords](#) on page
- [-Xkill-opt](#) on page
- [-Xkill-reorder](#) on page
- [-Xlicense-proxy-path](#) on page
- [-Xlicense-proxy-use](#) on page
- [-Xlicense-wait](#) on page

- [-Xlimit-reaching](#) on page
- [-Xlink-time-lint](#) on page
- [-Xlint](#) on page
- [-Xlno](#) on page
- [-Xlocal-data-area](#) on page
- [-Xlocal-data-area-static-only](#) on page
- [-Xlocals-on-stack](#) on page
- [-Xlto-group](#) on page
- [-Xmac-conventions](#) on page
- [-Xmacro-in-pragma](#) on page
- [-Xmacro-patterns](#) on page
- [-Xmacro-undefined-warn](#) on page
- [-Xmake-dependency](#) on page
- [-Xmake-dependency-canonicalize-path-off](#) on page
- [-Xmake-dependency-savefile](#) on page
- [-Xmake-dependency-target](#) on page
- [-Xmember-max-align](#) on page
- [-Xmemory-is-volatile, -Xglobals-volatile, -Xstatics-volatile, -Xpointers-volatile](#) on page
- [-Xmetadata](#) on page
- [-Xmismatch-warning](#) on page
- [-Xname-...](#) on page
- [-Xno-builtin](#) on page
- [-Xno-gnu-inline-asm](#) on page
- [-Xno-if-conversion](#) on page
- [-Xno-wpo-dead](#) on page
- [-Xold-inline-asm-casting](#) on page
- [-Xopt-count](#) on page
- [-Xoptimistic-aliasing](#) on page
- [-Xoptimized-debug-off](#) on page
- [-Xoptimized-debug-on](#) on page
- [-Xparse-count](#) on page
- [-Xpass-source](#) on page
- [-Xpic](#) on page
- [-Xpragma-section-first, -Xpragma-section-last](#) on page
- [-Xpreprocessor-lineno-off](#) on page
- [-Xpreprocessor-old](#) on page
- [-Xreduce-reaching](#) on page
- [-Xrelax-alias](#) on page
- [-Xrestart](#) on page
- [-Xreverse-inline](#) on page
- [-Xrtc](#) on page
- [-Xrtti, -Xrtti-off](#) on page
- [-Xsection-pad](#) on page
- [-Xsection-split](#) on page
- [-Xsection-split-name](#) on page
- [-Xsect-pri-off](#) on page
- [-Xsect-pri-on](#) on page
- [-Xsemi-is-comment](#) on page
- [-Xsemi-is-newline](#) on page
- [-Xshow-configuration](#) on page
- [-Xsmall-const](#) on page
- [-Xsmall-const-registers](#) on page
- [-Xsmall-data](#) on page
- [-Xsmall-data-registers](#) on page
- [-Xstack-delay, -Xstack-delay-off](#) on page
- [-Xstack-probe](#) on page
- [-Xstack-protection](#) on page
- [-Xstack-protection-all](#) on page

- [-Xstack-protection-off](#) on page
- [-Xstack-protection-strong](#) on page
- [-Xstack-protection-verbose](#) on page
- [-Xstatic-addr-error](#) on page
- [-Xstatic-addr-warning](#) on page
- [-Xstderr-fully-buffered](#) on page
- [-Xstop-on-warning](#) on page
- [-Xstrict-ansi](#) on page
- [-Xstrict-bitfield-promotions](#) on page
- [-Xstrict-eabi](#) on page
- [-Xstring-align](#) on page
- [-Xstruct-arg-warning](#) on page
- [-Xstruct-as-arg](#) on page
- [-Xstruct-assign-split-...](#) on page
- [-Xstruct-as-static](#) on page
- [-Xstruct-best-align](#) on page
- [-Xstruct-min-align](#) on page
- [-Xsuppress-warnings](#) on page
- [-Xswap-cr-nl](#) on page
- [-Xswitch-array-in-function-section](#) on page
- [-Xswitch-array-off](#) on page
- [-Xswitch-table](#) on page
- [-Xswitch-table-in-function-section](#) on page
- [-Xswitch-table-off](#) on page
- [-Xsystem-headers-warn](#) on page
- [-Xtest-at-both](#) on page
- [-Xtest-at-bottom](#) on page
- [-Xtest-at-top](#) on page
- [-Xtruncate](#) on page
- [-Xunderscore-leading](#) on page
- [-Xunderscore-surround](#) on page
- [-Xunderscore-trailing](#) on page
- [-Xunroll](#) on page
- [-Xunroll-register-rename](#) on page
- [-Xunroll-size](#) on page
- [-Xuser-metadata](#) on page
- [-Xvec](#) on page
- [-Xvec-aligned](#) on page
- [-Xvec-args-aligned](#) on page
- [-Xvec-verbose](#) on page
- [-Xvoid-ptr-arith-ok](#) on page
- [-Xwchar](#) on page
- [-Xwhole-program-diagnostics](#) on page
- [-Xwhole-program-optim](#) on page
- [-Xwpo-const](#) on page
- [-Xwpo-dead](#) on page
- [-Xwpo-delete-return-value](#) on page
- [-Xwpo-inline](#) on page
- [-Xwpo-inline-single-calls](#) on page
- [-Xwpo-no-alias](#) on page

-Xabsolute18-const, -Xabsolute18-data

Take advantage of the 18-bit absolute addressing mode supported on TriCore hardware.

Belongs to:

[Sections Index](#) on page 267

Synopsis

```
-Xabsolute18-const=n
-X448=n
-Xabsolute18-data=n
-X196=n
```

Description

Place small **const** static and global variables with a size in bytes less than or equal to **n** in **ZCONST** or **ZDATA** sections.

For more information, see the discussion of **section** and **use_section** pragmas in the *Locating Code and Data, Addressing, Access* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

Absolute addressing is suitable for addressing peripheral registers or global data. It uses an 18-bit constant for the memory address, this being specified in the instruction itself. The complete 32-bit address is formed by using the top 4 bits of the 18-bit constant as the top 4 bits of the 32-bit address, the lower 14 bits being directly copied into the address, and inserting 14 zero-bits between them.

-Xaddr-code

Set addressing mode for code sections.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xaddr-code=n
-X105=n
```

Description

Specify addressing for code.

For more information, see the discussion of **addr-mode** in the *Locating Code and Data, Addressing, Access* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

For x86, these options are maintained for compatibility and currently have no effect on generated code.

-Xaddr-const

Set addressing mode for const sections.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xaddr-const=n
-X102=n
```

Description

Specify addressing for constant static and global variables.

For x86 these options are maintained for compatibility and currently have no effect on generated code.

-Xaddr-data

Set addressing mode for data sections.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xaddr-data=n
-X100=n
```

Description

Specify addressing for non-constant static and global variables.

For more information, see the discussion of variable addressing in the *Wind River Diab Compiler User's Guide* for your architecture.

For x86, options are maintained for compatibility and currently have no effect on generated code.

-Xaddr-sconst

Set addressing mode for sconst sections.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xaddr-sconst=n
-X103=n
```

Description

Specify addressing for constant static and global variables with size less than or equal to **-Xsmall-const**.

For more information, see the discussion of **addr-mode** in the *Locating Code and Data, Addressing, Access* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xaddr-sdata

Set addressing mode for sdata sections.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xaddr-sdata=n
-X101=n
```

Description

Specify addressing for non-constant static and global variables with size less than or equal to that set with **-Xsmall-data**.

For more information, see the discussion of **addr-mode** in the *Locating Code and Data, Addressing, Access* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

See Also

[-Xsmall-data on page](#)

-Xaddr-string

Set addressing mode for string sections.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xaddr-string=n
-X104=n
```

Description

Specify addressing for strings.

For more information, see the discussion of **addr-mode** in the *Locating Code and Data, Addressing, Access* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

For x86 these options are maintained for compatibility and currently have no effect on generated code.

-Xaddr-user

Set addressing mode for user-defined sections.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xaddr-user=n
-X106=n
```

Description

Specify addressing for user-defined sections.

For more information, see the discussion of **addr-mode** in the *Locating Code and Data, Addressing, Access* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

For x86 these options are maintained for compatibility and currently have no effect on generated code.

-Xaggressive-static-live

Optimize static and global variable access aggressively.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xaggressive-static-live
-X139
```

Description

Make optimizations of static and global variable accessing more aggressive; for example, delete assignments to such variables in infinite loops from which there is no apparent return.

-Xalign-functions

Align functions on *n*-byte boundaries.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xalign-functions=n
-X54=n
```

Description

Set the minimum alignment for each function to *n*. This option is designed for targets having some type of burst-mode memory access, for example a target that can fetch multiple instructions if aligned on a 32-byte boundary.

If *n* is absent or *n* ≤ 0, *n* is reset to 1.

If *n* > 0, *n* is rounded down to a power of 2 (bytes).

Note

This option doesn't necessarily mean that the alignment of each function is set to exactly *n*. There may be other more stringent alignment requirements that need to be satisfied.

-Xalign-loop-to-cache

Optimize, using instruction-cache alignment.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xalign-loop-to-cache[=n]
-X451[=n]
```

Description

Depending on the software and the architecture used, a loop construct aligned to the start of the instruction cache will lead to a better performance than one not thus aligned. This option aligns the loop to the (assumed) start of the instruction cache line.

This optimization attempts to fit the loop in as small of a cache footprint as possible. This trades off code size for performance, as padding may be introduced to position the top of the loop precisely with the start of the instruction-cache line.

If *n* is not specified or *n* is set to 1 (the default), the compiler assumes a cache-line size of 32 bytes for PowerPC, and 16 bytes for TriCore.

If **n** > 1, the compiler will overwrite the default size of the instruction-cache line.

n is rounded down to a power of 2 (bytes).

-Xalign-min

Specify minimum alignment for single memory access to multi-byte values.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xalign-min=n
-X93=n
```

Description

Set the minimum alignment required by the target processor to access a multi-byte value (e.g., **short**, **long**) in memory as an atomic unit, that is, in a single memory access. This option is set automatically by the compiler based on the target processor and should seldom be set by the user.

Note

This option does not change how data is aligned; it changes the instructions which the compiler generates to access multi-byte unaligned objects.

Technical details

If the target processor can access objects at any alignment with a single instruction, **n** is set to 1. For a processor which requires that multi-byte objects be aligned on even-byte boundaries for direct access, **n** is set to 2. Unaligned objects on such a processor must be accessed byte-by-byte. For a processor that requires 4-byte objects be on a 4-byte boundary, **n** is set to 4 (2-byte objects aligned on 2-byte boundaries can still be accessed with a single instruction).

The default value of **n** equals the maximum alignment restriction as given in the manufacturer's documentation for the processor. Note that it may differ among processors in a family.

The defaults are as follows:

ARM

4.

ColdFire

1 for the 51xx, 5206E and the 5300 and 5400 series, and 4 for all other ColdFire family members.

MIPS

8 for MIPS and 4 for MIPS16 because MIPS16 does not support hardware floating point.

PPC

1 for all PowerPC family members, including the default "generic" processor (invoked when **-tPPC....** is used without a specific number), except the following: 4 for the 505, 509, 553, and 555, and 8 for the 403 and 405. See also **-Xalign-min-hardware-float**.

RH850

4.

TriCore

4.

x86

4.

Note

If **-Xalign-min** is > 1 (which is always the case for ARM, MIPS, TriCore, and x86), in a packed structure: (a) bit-fields members are not allowed, (b) volatile members will not be accessed atomically, and (c) compound operators (for example, "+=") cannot be used with volatile members.

Synonyms: **-Xmin-align=n**. In addition for PowerPC, **-Xunaligned-slow** is a synonym for **-Xalign-min=8**; and for RH850 it is a synonym for **-Xalign-min=4**.

See Also

[-Xalign-min-hardware-float on page](#)

-Xalign-min-hardware-float

Specify minimum alignment for memory access to float registers.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xalign-min-hardware-float=n
-X149=n
```

Description

Set the minimum alignment required by the target processor to access a multi-byte value in memory as an atomic unit going to or from a float register. This option is similar to **-Xalign-min**, but affects only the float registers. It supports the PowerPC 500 family, which has different restrictions for float registers and general registers.

This option is set automatically by the compiler and should seldom be set by the user. The default value is processor-dependent.

See Also

[-Xalign-min on page](#)

-Xalternate-coloring

(PPC only) Use alternate frame-packing algorithm for register allocation

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xalternate-coloring
```

Description

Use an alternate frame-packing algorithm for register allocation, which may reduce the overall size of the stack by reducing the stack frame footprint of functions. This option is not enabled by default and is only valid for PowerPC targets.

Since

5.9.3.1

-Xargs-in-regs

Pass argument in register.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xargs-in-regs
-X61
```

Description

For a function with a prototype declaring an argument with the **register** keyword, try to pass the variable in a register. **-Xargs-in-regs** is not supported with PIC (position-independent code).

-Xargs-not-aliased

Assume no aliasing of pointer arguments.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xargs-not-aliased
-X65
```

Description

This declares that no two argument pointers passed to a function can reference the same memory location, even with array subscripting offsets. So for example this allows the optimizer to assume that there is no overlap in the pointer or array dereferences of "a" and "b" in the function: `foo(int *a, int *b) { ... }`

Assume that pointer arguments to a function are not aliased with each other, nor with any global data. This enables greater optimization. Example:

```
int g;
foo(int* a, int* b);
void main () {
  int i = 1;
  int j = 2;
  foo(&i,&j); /* OK */
  foo(&i,&i); /* not OK */
  foo(&i,&g); /* not OK */
}
```

This option is also checked by the auto-vectorization optimizer when vectorizing operations on pointer type function arguments.

See also the discussion of the **no_alias** pragma in the *Additions to C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

See Also

[-Xvec on page](#)

-Xarray-align-min

Specify minimum array alignment.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xarray-align-min=n
-Xl61=n
```

Description

Align arrays on the larger of **n** or the default alignment for the type of the array elements. **n** should be a power of 2. When this option is used, values given for **-Xstring-align** are ignored.

-Xbit-fields-access-as-byte, -Xbit-fields-access-as-type

Disable bit-field access optimization.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xbit-fields-access-as-byte
-Xl18=0
```

Description

Enable use of faster byte instructions to access a small non-volatile bit-field even though that bit-field is declared with an underlying type other than **char**. This is the default.

-Xbit-fields-access-as-type

-Xl18=1

Force the compiler to always access bit-fields with load and store instructions of the same size as the bit-field declaration.

Synonym: **-Xbitfield-no-optim.**

-Xbit-fields-compress

Control bit-field storage.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xbit-fields-compress=n
-Xl35=n
```

Description

Adjust bit-field offsets in a structure.

By default, bit-fields are allocated sequentially, without additional padding. With **-Xbit-fields-compress** the bit-fields are allocated in units, where each unit contains a series of consecutive bit-fields. The size of a unit cannot exceed **n** bytes, and each unit starts on an **n**-byte boundary.

For example, without **-Xbit-fields-compress**, the following will require two bytes:

```
unsigned int f1:1;
unsigned int f2:4;
unsigned int f3:4;
unsigned int f4:7;
```

With **-Xbit-fields-compress=1**, it will require three bytes. The first unit contains **f1** and **f2**. The second unit contains **f3** and the third unit contains **f4**. Using a higher value of **n**, it would only require two bytes because all the bit-fields could fit in a single unit.

Synonym: **-Xbitfield-compress**.

-Xbit-fields-signed, -Xbit-fields-unsigned

Specify sign of plain bit-field.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xbit-fields-signed
-X12=0
```

Description

C only. Handle bit-fields without the **signed** or **unsigned** keyword as signed integers.

Synonym: **-X signed-bitfields**.

-Xbit-fields-unsigned

-X12

C only. Treat bit-fields without the **signed** or **unsigned** keyword as unsigned integers. For all but x86 this is the default setting.

Synonym: **-Xunsigned-bitfields**.

See Also

[-Xstrict-bitfield-promotions on page](#)

-Xblock-count

Insert profiling code.

Belongs to:

[Checking and Profiling Index on page 249](#)

Synopsis

```
-Xblock-count
-X24
```

Description

Insert code in the compiled program to keep track of the number of times each basic block (the code between labels and branches) is executed.

See also the *Wind River Diab Compiler Utilities Reference: D-BCNT Profiling Basic Block Counter*, and **-Xfeedback**.

Note

-Xblock-count and **-Xfeedback** are not supported for VxWorks applications.

See Also

[-Xfeedback on page](#)

-Xbool-is-...

Set type for bool.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xbool-is-char
```

Description

Implement type **bool** as a plain **char**. This is the default.

-Xbool-is-int

-X119=4

C++ only. Implement type **bool** as a **signed int**. This may produce less code on some architectures but will require more data space.

-Xbottom-up-init

Parse initial values bottom-up.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xbottom-up-init
-X21
```

Description

C only. Both K&R and ANSI C specify that structure and array initializations with missing braces should be parsed top-down, however some C compilers parse these bottom-up instead. Example:

```
struct z { int a, b; };
```

```
struct x {
```

```
  struct z z1[2];
```

```
  struct z z2[2];
```

```
} x = { {1,2},{3,4} };
```

Should be parsed according to ANSI & K&R as:

```
{ { {1,2},{0,0} } , { {3,4},{0,0} } };
```

-Xbottom-up-init causes bottom-up parsing:

```
{ {1,2},{3,4} } , { {0,0},{0,0} };
```

This option is set when **-Xdialect-pcc** is set.

-Xbss-off, -Xbss-common-off

Control allocation of uninitialized variables in "COMMON" and bss sections.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xbss-common-off
-X83=3
```

Description

Disable use of the "**COMMON**" feature so that the compiler or assembler will allocate each uninitialized public variable in the **.bss** section for the module defining it, and the linker will require exactly one definition of each public variable.

For more information, see the discussion of **COMMON** sections in the *Wind River Linker* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

Synonym: **-Xno-common**.

-Xbss-off

-X83=1

Put all variables in the **.data** section instead of allocating uninitialized variables in the **.bss** section.

Synonym: **-Xno-bss**.

-Xbss-zero-init-const-on, -Xbss-zero-init-data-off

Control movement of variables initialized to zero to BSS.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xbss-zero-init-const-on
-Xbss-zero-init-data-off
```

Description

A **BSS** section contains uninitialized variables which must be initialized to zero by the startup code. Declaration of an initialized global variable, such as in the following examples, means that these variables can be put into a **BSS** section:

```
int x = 0;
```

```
int array[] = {0,0};
```

Note that if zero-initialized variables are moved to **BSS**, the startup code must zero-out **BSS** before these variables are referenced.

By default, the compiler moves zero initialized **vars** to **BSS**, but does not move zero initialized **consts** to **BSS**.

These options provide the opposite of the default behavior.

-Xbss-zero-init-const-on

Move constants to **BSS** sections.

-Xbss-zero-init-data-off

Do not move variables to **BSS** sections.

-Xcga-min-use

Optimize global assignments in conditionals.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xcga-min-use=n
-Xl80=n
```

Description

When a global variable is accessed repeatedly within a conditional statement, the compiler can replace the global variable with a temporary local copy (which can be stored in a register), then reassign the local variable to the global variable when the conditional finishes execution.

If conditional global assignment is enabled, the compiler determines whether to copy a global variable by estimating the number of times the global variable is accessed within the conditional block at runtime. (The exact number of accesses may depend on factors, such as the value of a loop counter, that cannot be known at compile time.) If the global variable is accessed **n** or more times, the compiler performs the optimization. The default value of **n** is 20.

Conditional global assignment is enabled by default (**-Xcga-min-use=20**) whenever optimizations are enabled (**-O** or **-XO**). To disable conditional global assignment, set **n** to 0 (**-Xcga-min-use=0**). Conditional global assignment is never performed on variables declared or treated as volatile (see **-Xmemory-is-volatile**) and should be used with caution in multi-threaded environments.

See Also

[-Xmemory-is-volatile on page](#)

-Xcharset-ascii

Generate code using ASCII character set.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xcharset-ascii
-X60=1
```

Description

Generate code using the ASCII character set. All strings and character constants are converted to ASCII. The default is to use the same character system as the host machine.

Synonym: **-Xascii-charset**.

-Xchar-signed, -Xchar-unsigned

Specify sign of plain char.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xchar-signed
-X23=0
```

Description

Treat variables declared **char** without either of the keywords **signed** or **unsigned** as signed characters.

Synonym: **-Xsigned-char**.

```
-Xchar-unsigned
-X23
```

Treat variables declared **char** without either of the keywords **signed** or **unsigned** as unsigned characters.

Synonym: **-Xunsigned-char**.

The default setting is **signed**, except for ARM and PowerPC, in which cases it is **unsigned**.

See also the *Wind River Diab Compiler User's Guide* (for your architecture) for a table describing C/C++ data Types, sizes, and alignments; and for the discussion of **__SIGNED_CHARS__**.

In C++, plain **char**, **signedchar** and **unsignedchar** are always treated as different types, but this option defines how arithmetic with plain **char** is done.

-Xclib-optim-off

Disregard ANSI C library functions.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xclib-optim-off
-X66
```

Description

Direct the compiler to disregard all knowledge of ANSI C library functions.

By default, the compiler automatically includes, before all other header files, the file `lpragma.h`, which contains **pure_function**, **no_return**, and **no_side_effects** pragmas and other statements that allow optimization of calls to C library functions. (If the default include directory **versionDir**/`include` exists, the compiler looks for `lpragma.h` only in this directory. If **versionDir**/`include` does not exist, the compiler searches for `lpragma.h` in other user-specified directories.)

This option disables use of `lpragma.h`.

In case of auto-vectorization (**-Xvec**) the vectorizer can turn some math functions into vector math operations, for example **sqrt** to **VSQRT**, but it won't do this if the **-Xclib-optim-off** option is enabled.

Synonym: **-Xno-recognize-lib**.

See Also

[-Xvec on page](#)

-Xcode-absolute...

Use absolute addressing for code.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xcode-absolute-[far|near]
```

Description

-Xcode-absolute-far

\-X58=6

Use 32-bit absolute addressing for code.

For TriCore, function calls are made by loading a scratch register with the full 32-bit function address and calling indirectly through the registers. This overcomes the 24-bit limit of PC-relative addressing.

For more information, see the discussion of addressing modes in the *Wind River Diab Compiler User's Guide* for your architecture.

-Xcode-absolute-near

-X58=5

For ColdFire, MIPS, PowerPC, and RH850, use 16-bit absolute addressing for code.

For ARM and x86, use 32-bit absolute addressing for code. Note that ARM is the same as **-Xcode-absolute-far**. (maintained for compatibility.)

For TriCore, use PC-relative, 24-bit absolute addressing for code.

For more information, see the discussion of addressing modes in the *Wind River Diab Compiler User's Guide* for your architecture.

-Xcode-factor

Share common code sequences.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xcode-factor
```

Description

Find common code sequences at link time and share them, reducing code size at the cost of inserting some additional branches.

-Xcode-factor is a compiler option, not a linker option. It should be applied to those modules you wish to participate in link time code factoring. The option only has an effect in conjunction with an optimization option such as **-O** or **-XO**. Note that **-Xcode-factor** implicitly enables **-Xsection-split** for code (**-Xsection-split=1**).

Code sharing will only be considered between functions meeting the following criteria:

- Both functions must have been compiled for the same target (so VLE functions will not share code with E500 functions or vice-versa)
- Both input sections must map to the same output memory. So all functions that end up in the **.text** section will be candidates for factoring, but code in **.text** will not be shared with code in **.boot**.

To disable **-Xcode-factor** on a particular module, do one of the following:

- Set **-Xcode-factor=0**
- Put the following line at the top of the source file:
#pragma option -Xcode-factor=0

Note that no special linker option is needed to enable code factoring. To enable diagnostic output, use the following linker option:

-Xcode-factor-diagnostics

Note

This optimization might make it hard to debug the tail ends of affected functions.

-Xcode-relative...

Generate position-independent code (PIC).

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xcode-relative-far
-X58=2
```

Description

Notes for TriCore and RH850

-Xcode-relative options are not available for TriCore and RH850 targets.

Notes for ColdFire

Generate position-independent code (PIC).

Branches and function calls use 32 bit offsets from the PC, relative to **PC**.

By default, global **const** or **static const** variables and string constants are included in the code section and are referenced using 32-bit offsets relative to **PC**. The default may be changed using option **-Xconst-in-text** which controls whether **const** variables and string constants are in "text" (code) or "data" sections.

For details, see **-Xconst-in-...** and the discussion of moving data from text to data in the *Wind River Diab Compiler User's Guide* for your architecture.

Option **-Xconst-in-text=0** should usually be used with **-Xcode-relative-far** because it will usually be faster to access **const** variables and string constants through register **a5** than via the PC. References to the small **const area**, if any, still use a 16-bit offset (see the discussion of moving data from text to data in the *Wind River Diab Compiler User's Guide* for your architecture.)

For global or **static** pointers to be position-independent, they must be initialized dynamically and are therefore always stored in a "data" section even if declared **const**. See option **-Xdynamic-init**).

This option produces larger and slower code.

Synonyms: **-Xfar-code-relative**, **-Xlong-pc-relative**

-Xcode-relative-far-all

-X58=4

Equivalent to **-Xcode-relative-far** except that all global and **static** variables are by default placed in the code section, not just those which are **const**.

Note that references to the small **const area**, if any, use a 32-bit offset rather than the more efficient 16-bit offset, and for this reason this option is deprecated.

Synonyms: **-Xall-far-code-relative**, **-Xall-long-pc-relative**

-Xcode-relative-near

-x58=1

Like **-Xcode-relative-far** except that offsets are 16-bit rather than 32-bit. See it for details.

Synonyms: **-Xnear-code-relative**, **-Xshort-pc-relative**

Note: Because offsets are 16-bit PC-relative, a string or **const** variable must be within 32KB of the instruction referencing it.

-Xcode-relative-near-all

-x58=3

Extends **-Xcode-relative-near** to all global and static variables, not just those declared **const**. See it for details.

Synonyms: **-Xall-near-code-relative**, **-Xall-short-pc-relative**

Notes for MIPS

Generate position-independent code (PIC).

Branches and function calls use 32 bit offsets from the PC, relative to **\$23**.

By default, global **const** or **static const** variables and string constants are included in the code section and are referenced using 32-bit offsets relative to **\$23**.

The default may be changed using option **-Xconst-in-text** which controls whether **const** variables and string constants are in "text" (code) or "data" sections. For details, see **-Xconst-in-...** and the the discussion of moving initialized data from text to data in the *Wind River Diab Compiler User's Guide* for your architecture.

References to the small **const area**, if any, still use a 16-bit offset (see the discussion of moving data from text to data in the *Wind River Diab Compiler User's Guide* for your architecture.)

For global or **static** pointers to be position-independent, they must be initialized dynamically and are therefore always stored in a "data" section even if declared **const**. See option **-Xdynamic-init**).

Synonym: **-Xfar-code-relative**

-Xcode-relative-far-all

-x58=4

Equivalent to **-Xcode-relative-far** except that all global and **static** variables are by default placed in the code section, not just those which are **const**.

Note that references to the small **const area**, if any, use a 32-bit offset rather than the more efficient 16-bit offset, and for this reason this option is deprecated.

Synonyms: **-Xall-far-code-relative**

-Xcode-relative-near

-x58=1

Like **-Xcode-relative-far** except that offsets are 16-bit rather than 32-bit. See it for details.

Synonym: **-Xnear-code-relative**

Note: Because offsets are 16-bit, the combined size of all sections to which this option applies is limited to 64KB (for better optimization, actually 64KB - 0x10. For more information, see the discussion of register use in the *Wind River Diab Compiler User's Guide* for your architecture

-Xcode-relative-near-all

-x58=3

Extends **-Xcode-relative-near** to all global and static variables, not just those declared **const**. See it for details.

Synonym: **-Xall-near-code-relative**

Notes for PowerPC

Generate position-independent code (PIC).

Branches and function calls use 26 bit offsets from the PC, relative to **r2**.

By default, global **const** or **static const** variables and string constants are included in the code section and are referenced using 32-bit offsets relative to **r2**.

The default may be changed using option **-Xconst-in-text** which controls whether **const** variables and string constants are in "text" (code) or "data" sections. For details, see **-Xconst-in-...** and the discussion of moving initialized data from text to data in the *Wind River Diab Compiler User's Guide* for your architecture.

References to the small **const area**, if any, still use a 16-bit offset (see the discussion of moving data from text to data in the *Wind River Diab Compiler User's Guide* for your architecture.)

For global or **static** pointers to be position-independent, they must be initialized dynamically and are therefore always stored in a "data" section even if declared **const**. See option **-Xdynamic-init**).

Synonym: **-Xfar-code-relative**

-Xcode-relative-far-all

-X58=4

Equivalent to **-Xcode-relative-far** except that all global and **static** variables are by default placed in the code section, not just those which are **const**.

Note that references to the small **const area**, if any, use a 32-bit offset rather than the more efficient 16-bit offset, and for this reason this option is deprecated.

Synonym: **-Xall-far-code-relative**.

-Xcode-relative-near

-X58=1

Like **-Xcode-relative-far** except that offsets are 16-bit rather than 32-bit. See it for details.

Synonym: **-Xnear-code-relative**

Note: Because offsets are 16-bit, the combined size of all sections to which this option applies is limited to 64KB (for better optimization, actually 64KB - 0x10. For more information, see the discussion of register use in the *Wind River Diab Compiler User's Guide* for your architecture

-Xcode-relative-near-all

-X58=3

Extends **-Xcode-relative-near** to all global and static variables, not just those declared **const**. See it for details.

Synonym: **-Xall-near-code-relative**.

See Also

[-Xconst-in-... on page](#)

[-Xdynamic-init on page](#)

-Xcomdat

Mark Sections as COMDAT for Linker Collapse

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xcomdat
-Xl20
```

Description

C++ only. Mark all generated sections as COMDAT. The linker automatically collapses identical COMDAT sections to a single section in memory. This is the default. By default, the compiler automatically generates a section for each instantiation of each member function or static class variable in a template in each module where the member function or variable is used. Given **-Xcomdat**, the compiler marks all implicit template instantiations as COMDAT and the linker collapses identical instances.

See Also

[-Xcomdat-info-file on page](#)

-Xcomdat-info-file

Maintain project-wide COMDAT list.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xcomdat-info-file=filename
```

Description

C++ only. When **-Xcomdat** is enabled, generate and maintain (in **filename**) a list of COMDAT entries across modules. The list is automatically updated and checked for consistency with each build. This option speeds up builds and reduces object-file size in projects that make extensive use of templates. Since COMDAT sections are ultimately collapsed by the linker, this option has no effect on the final executable file.

-Xcompress-prolog-epilog

Reduce size of function prologue and epilogue.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xcompress-prolog-epilog
```

Description

This switch reduces the size of functions by compressing their prologues and epilogues, but at the expense of runtime efficiency. The compression is achieved by calling non-ABI compliant functions (that do not have stack frames), which create or remove stack frames in the usual way. The function call is achieved with fewer instructions than stack frame creation or removal,

-Xconst-in-...

Locate constants with "text" or "data".

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xconst-in-text=mask
-X74=mask
-Xconst-in-data
-X74=0
```

Description

Locate **CONST**, **SCONST**, and **STRING** section class data in text (typically `.text`) or data (`.data` or `.sdata`) section, based on the mask setting.

0x1

CONST mask bit.

0x2

SCONST mask bit.

0x4

STRING mask bit.

Set the mask bit to 1 to locate the data in a text section; clear the mask bit to locate the data in a data section. The mask may be given in hex, and mask bits may be OR-ed to select more than one. For example, **-Xconst-in-text=0x5**. Undefined mask bits are ignored.

This option is used by default and mask depends on the architecture:

- For ColdFire and MIPS the default is 0x5 (small constants in `.sdata`, larger constants and strings in `.text`). For MIPS16 the default is 0x0.
- For ARM, PowerPC, RH850, TriCore, and x86 the default is 0x7 (all constants in `.text` section).

-Xconst-in-data and **-Xstrings-in-text** are shortcuts for locating all "constants" (**CONST**, **SCONST**, and **STRING** classes, not just "const" or string data) in "data" sections (**mask=0**) or "text" sections (**mask=0xff**) respectively.

The exact name of the "text" and "data" sections depends on the target architecture. For the names (and for examples), see the *Wind River Diab Compiler User's Guide* for your architecture.

When **STRING** is in a text section, identical string constants will be stored only once. This is the default in version 3.6 and later.

For Tricore, see also **-Xtricore-no-relax-ldata** in the *Linker User's Guide*.

-Xcpp-cr-line-terminator

Enable line terminator behavior of carriage return (CR) characters.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xcpp-cr-line-terminator
-X462
```

Description

Accept carriage return (CR) characters as line terminators without being followed by line feed (LF) characters. This allows the compiler to do a correct line count when the source code uses CR characters as line terminators. It is disabled by default and can be enabled by adding it to the command line.

-Xcpp-dump-symbols

Dump symbol information for macros or assertions.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xcpp-dump-symbols=mask
-X158=mask
```

Description

Dump symbol information for macros, assertions, or both. To show macros, set bit 0 (the LSB) of **mask** to 1. To show assertions, set bit 1 to 1. To show line numbers, set bit 2 to 0. The default **mask** is 7 (show macros and assertions, no line numbers).

-Xcpp-no-space

Suppress preprocessor spacing.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xcpp-no-space
-X117
```

Description

C only. Do not insert spaces around macro names and arguments during preprocessing.

-Xc++-template-name-lookup-old

Disable standards-conformant dependent name processing.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xc++-template-name-lookup-old
```

Description

By default, the compiler performs standards-conformant two-phase lookup for names used in templates. **-Xc++-template-name-lookup-old** disables this, which may be useful for some older C++ source.

-Xdata-absolute...

Use absolute addressing for data.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xdata-absolute-far
-X59=6
```

Description

Use 32-bit absolute addressing for data.

-Xdata-absolute-near

-X59=5

For TriCore, use the 18-bit absolute addressing mode for data.

For all others, use the 16-bit absolute addressing mode for data.

See also the discussion of addressing mode in the *Locating Code and Data, Addressing, Access* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xdata-relative...

Generate position-independent data (PID).

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xdata-relative-[far|near]
-X59=1
```

Description

Notes for ColdFire

\-Xdata-relative-far

\X59=1

Generate position-independent data (PID) references to all global or static variables (except strings and **const** variables if the **-Xconst-in-text=0xf** option is used).

Use 32-bit offsets from register **a5** except for those global or static variables in the Small Data Area (SDA), which will be accessed through fast 16-bit offsets from **a5**, which means the SDA is limited to 64KB (to facilitate certain optimizations, actually 64KB - 0x10. For more information, see the discussion of register use in the *Wind River Diab Compiler User's Guide* for your architecture.

Because **a5** is used as a base pointer, all files must be compiled with either this or the companion option **-Xdata-relative-near**.

Synonyms: **-Xfar-data-relative**, **-Xlong-a5-relative**.

-Xdata-relative-near

-X59=1

Generate position-independent data (PID) references to all global or static variables (except strings and **const** variables if the **-Xconst-in-text=0** option is used).

All references use a 16-bit offset from register **a5**, which means the combined size of all sections to which this attribute applies is limited to 64KB (to facilitate certain optimizations, actually 64KB - 0x10. For more information, see the discussion of register use in *Wind River Diab Compiler User's Guide* for your architecture.

Because **a5** is used as a base pointer, all files must be compiled with either this or the companion option **-Xdata-relative-far**.

Synonyms: **-Xnear-data-relative**, **-Xshort-a5-relative**.

Note

If the option **-Xconst-in-text = 0xf** (equivalent to the older option **-Xstrings-in-text**), strings and **const** variables will be placed in “text” sections and addressed as code rather than as position-independent data. For more information, see the discussion of moving initialized data from text to data in the *Wind River Diab Compiler User's Guide* for your architecture.

Notes for MIPS

\-Xdata-relative-far

\X59=1

Generate position-independent data (PID) references to all global or static variables (except strings and **const** variables if the **-Xconst-in-text=0xf** option is used).

Use 32-bit offsets from register **\$28** except for those global or static variables in the Small Data Area (SDA), which will be accessed through fast 16-bit offsets from **\$28**, which means the SDA is limited to 64KB (to facilitate certain optimizations, actually 64KB - 0x10. For more information, see the discussion of register use in the *Wind River Diab Compiler User's Guide* for your architecture.

Synonym: **-Xfar-data-relative**.

-Xdata-relative-near

-X59=1

Generate position-independent data (PID) references to all global or static variables (except strings and **const** variables if the **-Xconst-in-text=0** option is used).

All references use a 16-bit offset from register **\$28**, which means the combined size of all sections to which this attribute applies is limited to 64KB (to facilitate certain optimizations, actually 64KB - 0x10. For more information, see the discussion of register use in the *Wind River Diab Compiler User's Guide* for your architecture.

Synonym: **-Xnear-data-relative**.

Note

If the option **-Xconst-in-text = 0xf** (equivalent to the older option **-Xstrings-in-text**), strings and **const** variables will be placed in “text” sections and addressed as code rather than as position-independent data. For more information, see the discussion of moving initialized data from text to data in the *Wind River Diab Compiler User's Guide* for your architecture.

Notes for PowerPC

\-Xdata-relative-far

\X59=1

Generate position-independent data (PID) references to all global or static variables (except strings and **const** variables if the **-Xconst-in-text=0xf** option is used).

Use 32-bit offsets from register **r13** except for those global or static variables in the Small Data Area (SDA), which will be accessed through fast 16-bit offsets from **r13**, which means the SDA is limited to 64KB (to facilitate certain optimizations, actually 64KB - 0x10. For more information, see the discussion of register use in the *Wind River Diab Compiler User's Guide* for your architecture.

Synonym: **-Xfar-data-relative**.

-Xdata-relative-near

-X59=1

Generate position-independent data (PID) references to all global or static variables (except strings and **const** variables if the **-Xconst-in-text=0** option is used).

All references use a 16-bit offset from register **r13**, which means the combined size of all sections to which this attribute applies is limited to 64KB (to facilitate certain optimizations, actually 64KB - 0x10. For more information, see the discussion of register use in the *Wind River Diab Compiler User's Guide* for your architecture.

Synonym: **-Xnear-data-relative**.

Note

If the option **-Xconst-in-text = 0xf** (equivalent to the older option **-Xstrings-in-text**), strings and **const** variables will be placed in “text” sections and addressed as code rather than as position-independent data. For more information, see the discussion of moving initialized data from text to data in the *Wind River Diab Compiler User's Guide* for your architecture.

-Xdebug-align

Align .debug sections

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xdebug-align [=n]
-X170=n
```

Description

Align **.debug** sections on specified boundaries. **n** is a power of 2; e.g., **-Xdebug-align=3** aligns **.debug** sections on 8-byte boundaries. If **n** is omitted, alignment defaults to 4-byte boundaries.

Without this option, **.debug** sections are aligned on byte boundaries.

-Xdebug-dwarf1

Select DWARF 1 format.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-dwarf1
-X153=1
```

Description

Generate DWARF 1.1 debug information.

See Also

[-Xdebug-dwarf2 on page](#)
[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf2

Select DWARF 2 format.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-dwarf2
-X153=2
```

Description

Generate DWARF 2 debug information. This is the default.

See Also

[-Xdebug-dwarf1 on page](#)
[-Xdebug-dwarf3 on page](#)
[-Xdebug-dwarf2-extensions-off on page](#)

-Xdebug-dwarf2-extensions-off

Suppress vendor-specific extensions.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-dwarf2-extensions-off
```

Description

Suppress vendor-specific extensions in DWARF 2 and DWARF 3 debug information.

See Also

[-Xdebug-dwarf2 on page](#)
[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf3

Select DWARF 3 format.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-dwarf3  
-X153=3
```

Description

Generate DWARF 3 debug information.

See Also

[-Xdebug-dwarf1 on page](#)
[-Xdebug-dwarf2 on page](#)
[-Xdebug-dwarf2-extensions-off on page](#)

-Xdebug-inline-on

Generate debug information for inlined functions.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-inline-on
```

Description

Generate debugging information for all inlined functions. Works with DWARF 2 and DWARF 3 only. Can result in very large executables. This option is disabled by default.

-Xdebug-local-all

Emit debug information for unused local variables.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-local-all
-X181
```

Description

Emit debugging information for all local variables, even variables that are never used. This option is disabled by default.

-Xdebug-local-cie

Generate local CIE for each unit.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-local-cie
-X176
```

Description

Generate a local Common Information Entry (CIE) for each unit. This option, which eliminates the dependency on the debug library `libg.a`, is applicable only with DWARF 2 or DWARF 3 debug information.

-Xdebug-mode

Disable debugging information extensions.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-mode=mask
-X99=mask
```

Description

Disable extensions to debugging information per bits in **mask**. May be necessary for other vendors' assemblers or for debuggers which cannot process the extensions.

mask may be given in hex, and mask bits may be OR-ed to select more than one, for example, **-Xdebug-mode=0x6**. Undefined mask bits are ignored.

0x2

Information regarding executable code in a header file (DWARF1, ELF).

0x4

Use of **.d1line** assembler directive (DWARF1, ELF).

0x8

Forward references to structures in generated assembler code (COFF only).

0x10

Line number information for **asm** statements (DWARF1, DWARF2, DWARF3).

0x40

Use of **.d1_line_start** and **.d1_line_end** assembler directives (DWARF1).

0x100

Column information (DWARF 2 and DWARF 3, C++).

0x200

Do not generate debug information to describe source code labels.

0x400

Do not emit line number information.

0x800

Do not emit any producer information.

0x1000

Do not emit coordinate information for variables, types, and so on.

0x2000

Do not emit entries for source code labels.

-Xdebug-source-line-barriers-off

Do not restrict scheduling optimization.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-source-line-barriers-off
```

Description

This is the default. The schedulers in **reorder** (and in llopt for ARM, MIPS, and PowerPC) are not restricted to scheduling only within source code statements.

See Also

[-Xdebug-source-line-barriers-on](#) on page

-Xdebug-source-line-barriers-on

Restrict scheduling optimization by source line.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-source-line-barriers-on
```

Description

With **-Xdebug-source-line-barriers-on**, the compiler marks a boundary between lines of source code. The schedulers in **llopt** (ARM, MIPS, and PPC) and **reorder** do not schedule instructions across the boundary, so scheduling can only change the order of instructions that belong to the same source line, and do not interleave instructions that belong to different source lines. The resulting generated code may be easier to understand and debug. It may also be easier to certify.

This reduces the benefit of pipelining, so the program may stall more and run slower.

Note

This does not affect other optimizations that may move code beyond source code boundaries. This option affects scheduling only.

See Also

[-Xdebug-source-line-barriers-off](#) on page

-Xdebug-struct-all

Disable debug optimization of type information.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-struct-all
-X116=1
```

Description

Force generation of type information for **typedef**, **struct**, and **union**, and **class** types, even when such types are not referenced in a file.

See Also

[-Xdebug-struct-compact](#) on page

-Xdebug-struct-compact

Enable debug optimization of type information.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xdebug-struct-compact
-X116=0
```

Description

Do not output types which are not used in debug information. This is the default, and it generates more compact but still complete version of debug information.

See Also

[-Xdebug-struct-all on page](#)

-Xdiagnose-inline...

List all functions that have not been inlined.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xdiagnose-inline
```

Description

This switch lists all functions either explicitly specified to be inlined or those which should become inlined automatically, but which the compiler has chosen not to inline.

-Xdiagnose-inline-verbose

List all functions that have not been inlined and explain why they are not inlined.

Examples of output include the following:

... callee too large: callee size (100) > CALLEE_MAX_SIZE (40): try increasing -Xinline

In this case the size of the function being called (100) exceeds the threshold specified by -Xinline (40).

... caller has grown too large: caller size (4010) > CALLER_MAX_SIZE (4000)

In this case the size of the calling function (the recipient of the inlining) has grown too large (which is usually the result of previous inlining). No more inlining will be performed into this function.

-Xdialect...

Specify C dialect.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xdialect-xxxx
```

Description

-Xdialect-ansi

Follow the ANSI C standard with some additions.

Synonyms:

-Xansi, -Xa.

-Xdialect-strict-ansi

Strictly follow the ANSI C and C++ standards. For C++, see **-Xstrict-ansi**.

Synonym:

-Xstrict-ansi.

-Xdialect-c11

Follow the C 2011 standard (language only, experimental)

-Xdialect-c89

Follow the C89 standard

-Xdialect-c99

Follow the C99 standard.

See Also

[-Xstrict-ansi on page](#)

-Xdialect-c++...

Specify C++ dialect.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xdialect-std
```

Description

Enable support for the latest C++ standards.

By default, the Diab C++ compiler supports the ANSI C++ standard (ISO/IEC FDIS 14882:2003), aka the "C++03" standard. Beginning with release 5.9.6.3, the Diab C++ compiler also supports the C++14 language standard (ISO/IEC 14882:2011 and ISO/IEC 14882:2014) and a "bare metal" subset of the C++14 library.

Use the **-Xdialect-std** option to enable the latest C++ standards and libraries supported by Diab. Note that this option must be specified to both the compiler and linker in order to select both the correct language dialect and the correct libraries.

To link against the C++ libraries, use the linker option **-lstl**. This automatically selects the correct implementation depending on whether or not **-Xdialect-std** is specified.

Note

We recommend that customers who want to use the latest standards use **-Xdialect-std**, which always selects the latest C and C++ standards (and corresponding libraries) supported by Diab. Only very rarely should customers need to explicitly pick a specific modern standard such as C++11. Code written for older standards will usually work fine with **-Xdialect-std**.

To avoid breaking legacy applications, the default standard for C is C89 with legacy Diab C libraries, and for C++ the default is the C++03 standard.

Alternative options:

-Xdialect-c++03

Support the C++ 2003 standard.

-Xdialect-c++11

Support the C++ 2011 standard.

-Xdialect-c++14

Support the C++ 2014 standard.

Note

All these options should be specified to both the compiler and the linker because they also select the library implementation.

C++11/14

The Diab C++11/C++14 language implementation has the following limitations:

- The `thread_local` keyword is not supported yet
- Unicode character and string types (`char16_t`, `char32_t`) are not yet supported.

The Diab C++11/C++14 library implementation includes all functionality that does not require extended operating system support. This includes all library functionality except for the following:

Atomic Operations Library

<atomic>

Chron and Signal Utilities

<chrono>

<csignal>

Thread Support

<condition_variable>

<future>

<mutex>

<shared_mutex>

<thread>

Filesystem Library

<filesystem>

Localization Library

<locale>

<codecvt>

<locale>

Note

Using any of the unsupported language or library features listed above may lead to compilation and/or execution errors.

-Xdisable-keywords

Disable keywords selectively.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xdisable-keywords=mask
-X418
```

Description

Selectively disable keywords enabled by **-Xkeywords**. **-Xdisable-keywords** overrides **-Xkeywords** regardless of their relative order on the command line. However, as with other **-X** flags, the last occurrence of **-Xdisable-keywords** “wins”; thus the default keyword state can be restored by appending **-Xdisable-keywords=0** to the command line.

See **-Xkeywords**, for information on **-Xkeywords**.

For example, you can enable the **packed** (0x8) and **interrupt** (0x10) keywords like so:

```
dcc -Xkeywords=0x18 file.c
```

Adding **-Xdisable-keywords=0x8** disables **packed** and leaves **interrupt** enabled:

```
dcc -Xdisable-keywords=0x8 -Xkeywords=0x18 file.c
```

For PowerPC, on AltiVec-based chips, **-Xdisable-keywords=0x60** disables the non-standard “vector” and “pixel” keywords without affecting any other keywords.

See Also

[-Xkeywords on page](#)

-Xdollar-in-ident

Allow dollar signs in identifiers.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xdollar-in-ident
-X67
```

Description

Allow dollar sign characters, “\$”, in identifiers. This option is off by default.

-Xdouble-avoid

Force single precision.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xdouble-avoid
-X96=3
```

Description

C only. Force all **float** literals (that have not been explicitly typecast) to single-precision. This option applies to constants such as `#define VALUE (6.606)` where `6.606` get implicitly treated as a **double**.

See Also

[-Xdouble-error on page](#)
[-Xdouble-warning on page](#)

-Xdouble-error

Generate error for double precision operations.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xdouble-error
-X96=1
```

Description

Generate an error if any double precision operation is used. It will also force all double constants to single precision and generate only single-precision instructions.

For PowerPC and TriCore, note that **-Xdouble-error** will not generate an error if the compiler is called with target-configuration option **G** (All Single Hardware Floating Point). For example:

```
% dcc -tPPC5554EG:cross example.c -Xdouble-error -c
```

```
% dcc -tTC1910LG:cross example.c -Xdouble-error -c
```

This is because when you use the **G** option, the compiler uses **-Xno-double** by default, which forces **double** to be the same as **float** on machines where they differ. (The **-Xno-double** option is a synonym for **-Xfp-float-only**.)

For more on selecting floating-point options with **-t**, see the *Wind River Diab Compiler User's Guide* for your architecture.

See Also

[-Xfp-long-double-off on page](#)
[-Xfp-float-only on page](#)
[-Xdouble-avoid on page](#)
[-Xdouble-warning on page](#)

-Xdouble-warning

Warn about double precision operations.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xdouble-warning
-X96=2
```

Description

Generate a warning if any double precision operation is used. It will also force all double constants to single precision and generate only single precision instructions.

For PowerPC and TriCore: Note that **-Xdouble-warning** will not generate an error if the compiler is called with target-configuration option **G** (All Single Hardware Floating Point). For example:

```
% dcc -tPPC5554EG:cross example.c -Xdouble-warning -c
```

```
% dcc -tTC1910LG:cross example.c -Xdouble-warning -c
```

This is because when you use the **G** option, the compiler uses **-Xno-double** by default, which forces **double** to be the same as **float** on machines where they differ. (The **-Xno-double** option is a synonym for **-Xfp-float-only**.)

For more on selecting floating-point options with **-t**, see the *Wind River Diab Compiler User's Guide* for your architecture.

See Also

[-Xfp-long-double-off on page](#)

[-Xfp-float-only on page](#)

[-Xdouble-avoid on page](#)

[-Xdouble-error on page](#)

-Xdynamic-init

Generate initializers for static variables.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xdynamic-init=1
-X121=1
```

Description

Cause the compiler to generate code in the initialization section to initialize addresses in static initializers. This option can be applied to any code, but is required for position-independent code and for C++ virtual tables. Example:

```
static int * address_p = & static_var;
```

Without this option, the above initializer would generate an error message if the code is compiled to be position-independent.

```
-Xdynamic-init=2
```

```
-X121=2
```

Extends the **-Xdynamic-init=1** option to generate code in the initialization section for all initializers, not just addresses.

-Xendian-little

Compile in little-endian mode.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xendian-little
-X94
```

Description

Compile in little-endian mode. This option is generated automatically by the driver when little-endian mode is specified as part of the **-t** option. This option should not be given by the user; doing so may lead to undefined behavior. To specify little-endian mode, use the appropriate argument to the **-t** option (see chapter 1: Target Configuration). This option is documented here for informational purposes only.

-Xenum-bitfields

Do not treat bitfields of type enum the same as bitfields of type int.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xenum-bitfields
```

Description

By default, the **enum** type with a bit field specified is converted to the **unsigned** type. With this option, the type would be kept as **enum**, which may improve debug capability.

-Xenum-is-...

Specify enum type.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xenum-is-best
-X8=2
```

Description

Use the smallest signed or unsigned integer type permitted by the range of values for an enumeration that is sufficient to represent the values of the enumeration constants. That is, the first of the following:

- the default **char** type for the architecture (**signed char** or **unsigned char**)
- the non-default **char** type for the architecture (**signed char** or **unsigned char**)
- **short**
- **unsigned short**
- **int**
- **unsigned int**
- **long**
- **unsigned long**

Note that **long** is not available for enumerated types.

For example, an enumeration with values from 1 through 128 will have base type **unsigned char** and require one byte. (Using the packed keyword on an enumerated type yields the same result as **-Xenum-is-best**.)

-Xenum-is-int**-X8**

This is the default. For C modules, the **enum** type is always equivalent to **int**. For C++, each **enum** type is equivalent to **int** if the range will fit, or **unsigned int** if it will not; if the range will not fit into either, a warning is issued and **unsigned int** is used.

-Xenum-is-short**-X8=3**

Each **enum** type is always equivalent to **signed short** if the range will fit, or **unsigned short** if it will not. If the range will not fit into either, a warning is issued and **unsigned short** is used.

-Xenum-is-small**-X8=0**

Use the smallest signed integer type permitted by the range of values for an enumeration, that is, the first of **signed char**, **short**, **int**, or **long** sufficient to represent the values of the enumeration constants. Thus, an enumeration with values from 1 through 128 will have base type **short** and require two bytes.

-Xenum-is-unsigned**-X8=4**

Use the smallest unsigned integer type permitted by the range of values for an enumeration, that is, the first of **unsigned char**, **unsigned short**, **unsigned int**, or **unsigned long** sufficient to represent the values of the enumeration constants. Thus, an enumeration with values from 1 through 128 will have base type **unsigned char** and require one byte.

-Xenum-is-best-old**-X8=5**

Provides the legacy (pre 5.9.3) operation of **-Xenum-is-best**. This means that for architectures for which the default type for **char** is unsigned (ARM, PPC), the **-Xenum-is-best-old** option chooses **signed char** when the type fits into a byte. This may result in extra sign extension instructions when the **enum** type is accessed. This option is provided for backward-compatibility only for cases in which the legacy behavior must be replicated.

Note

If modules compiled with different **-Xenum-is-...** options are mixed in a program, compatibility problems may result. When an enumerated type occurs within a packed structure, the default behavior is to use the smallest possible integer type for the enumeration constants (**-Xenum-is-best**). To override this behavior, specify **-Xenum-is-short** or **-Xenum-is-unsigned**.

-Xexceptions-...

Enable exceptions.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xexceptions-off
-X200=0
```

Description

C++ only. Disable exceptions. Compiling a program with any of the keywords **try**, **catch**, or **throw** will cause a compilation error. (But **throw()** is still allowed in function declarations to indicate that **new** or **delete** will not throw exceptions.) Compiling with this option will reduce stack space and increase execution speed when classes with destructors are used.

Synonym: **-Xno-exception**.

-Xexceptions

-X200

C++ only. Enable exceptions. This is the default.

For mixed C/C++ programs, see also **-Xframe-info**.

Synonym: **--Xexception**.

See Also

[-Xframe-info on page](#)

-Xexplicit-inline-factor

Control inlining expansion.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xexplicit-inline-factor
-Xexplicit-inline-factor=n
-X136=n
```

Description

Limits the inlining in a function (explicit and implicit) to an expansion of **n** times (measured in nodes where, roughly, each operator or operand counts as one node).

Given a function **f**, the compiler first inlines all functions explicitly declared inline which **f** calls, as well as any other small functions which can be inlined based on the other inlining optimization controls. It then divides the new size of the function (number of nodes) by the size with no inlining. If the result is **<=n**, it looks for new inlining opportunities in the resulting code and repeats the cycle. Once an expansion of **n** times is exceeded, inlining stops.

If **-Xexplicit-inline-factor** is specified with no value, **n** defaults to 3. If **-Xexplicit-inline-factor** is not specified, the default value is 0 (which means no limit) for C and 3 for C++.

See Also

[-Xinline-explicit-force on page](#)

-Xextend-args

Force precision of real arguments.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xextend-args
-X77
```

Description

Make all floating point arguments use the precision given by whichever of **-Xfp-min-prec-double**, **-Xfp-min-prec-long-double**, or **-Xfp-min-prec-float** is in force (all are settings of **-X3**), even if prototypes are used. (If none of the **-X3** options are also given, the default is **-Xfp-min-prec-double** as that is equivalent to **-X3=0**).

Note

If this option is used, libraries containing functions with floating point parameters must be recompiled. For safety, recompile all libraries to avoid missing any such functions.

-Xfeedback

Optimize using profile data.

Belongs to:

[Checking and Profiling Index on page 249](#)

Synopsis

```
-Xfeedback
-Xfeedback=file
(no numeric equivalent)
```

Description

Use profiling information generated by **-Xblock-count** to optimize for faster code. **file** is the name of the profiling file. The default is `dbcnt.out`.

To use this option:

- Compile a program with **-Xblock-count**.
- Run the program, which now creates `dbcnt.out` with profiling information. (For information about file I/O in an embedded environment, see the *Use in an Embedded Environment* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.)
- Recompile, now with the **-XO** and **-Xfeedback** options to produce high-level speed optimized code. Use **-Xfeedback-frequent** and **-Xfeedback-seldom** described below to control how the feedback data affects optimization.
Note: **-Xblock-count** and **-Xfeedback** are not supported for VxWorks applications.

See Also

[-Xblock-count on page](#)

-Xfeedback-frequent

Set upper feedback limit for optimization parameters used with profile data.

Belongs to:

[Checking and Profiling Index on page 249](#)

Synopsis

```
-Xfeedback-frequent=n
-X68=n
```

Description

Change the parameters used to control optimization of basic blocks when using profile data, for example, the amount of inlining, loop unrolling, and reorganization to reduce branches actually taken, all to increase speed (sometimes at the expense of space).

When using **-Xfeedback**, the compiler divides the basic blocks into three categories: code executed "frequently", "sometimes", and "seldom". More of the above optimizations are done for "frequent" code, while less or none is done for code executed "seldom".

The higher the thresholds, the more often code must be executed to get into the "frequent" category.

The defaults are **-Xfeedback-seldom=10** and **-Xfeedback-frequent=50** and are used as follows: each execution of a basic block recorded in the profile counts as one tick. The low-mark and high-mark values are normalized on a basis of 1,000 ticks, which means that the options have units of a tenth of a percent. That is, the default values mean that, if exactly 1,000 ticks are recorded, blocks executed fewer than 10 times (up to 1%) are marked "seldom", those executed from 10 to 50 times (1% to 5%) are marked "sometimes", and those executed 50 or more times (5% of more) are marked "frequent". Example:

-Xfeedback-frequent=30

means that blocks accounting for 3% or more of all ticks will go into the "frequent" category, and the compiler will do more inlining of functions called within these blocks, more loop unrolling, etc., to decrease their execution time.

Synonym: **-Xhi-mark**

See Also

[-Xfeedback on page](#)

[-Xfeedback-seldom on page](#)

-Xfeedback-seldom

Set lower feedback limit for optimization parameters used with profile data.

Belongs to:

[Checking and Profiling Index on page 249](#)

Synopsis

```
-Xfeedback-seldom=n
-X69=n
```

Description

Change the parameters used to control optimization of basic blocks when using profile data, for example, the amount of inlining, loop unrolling, and reorganization to reduce branches actually taken, all to increase speed (sometimes at the expense of space).

When using **-Xfeedback**, the compiler divides the basic blocks into three categories: code executed "frequently", "sometimes", and "seldom". More of the above optimizations are done for "frequent" code, while less or none is done for code executed "seldom".

The higher the thresholds, the more often code must be executed to get into the "frequent" category.

The defaults are **-Xfeedback-seldom=10** and **-Xfeedback-frequent=50** and are used as follows: each execution of a basic block recorded in the profile counts as one tick. The low-mark and high-mark values are normalized on a basis of 1,000 ticks, which means that the options have units of a tenth of a percent. That is, the default values mean that, if exactly 1,000 ticks are recorded, blocks executed fewer than 10 times (up to 1%) are marked "seldom", those executed from 10 to 50 times (1% to 5%) are marked "sometimes", and those executed 50 or more times (5% of more) are marked "frequent". Example:

-Xfeedback-frequent=30

means that blocks accounting for 3% or more of all ticks will go into the "frequent" category, and the compiler will do more inlining of functions called within these blocks, more loop unrolling, etc., to decrease their execution time.

Synonym: **-Xlo-mark**

See Also

[-Xfeedback on page](#)

[-Xfeedback-frequent on page](#)

-Xforce-declarations

Generate warnings on undeclared functions.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xforce-declarations
-X9
```

Description

Generate warnings if a function is used without a previous declaration.

This is useful to make C a more strongly typed language. This option is ignored when compiling C++ modules.

Note that **-Xforce-prototypes** and **-Xforce-declarations** provide different values for the same underlying option (**-X9**). If both options are used, then the last one specified takes precedence.

See Also

[-Xforce-prototypes on page](#)

-Xforce-prototypes

Generate warnings on functions without previous prototype.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xforce-prototypes
-X9=3
```

Description

Generate warnings if a function is used without a previous prototype declaration.

This is useful to make C a more strongly typed language. This option is ignored when compiling C++ modules.

Note that **-Xforce-prototypes** and **-Xforce-declarations** provide different values for the same underlying option (**-X9**). If both options are used, then the last one specified takes precedence.

See Also

[-Xforce-declarations on page](#)

-Xfp-fast, -Xfp-normal, -Xfp-pedantic

Specify degree of conformance to the IEEE754 standard.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xfp-fast
-X82=2
-Xfp-normal
-X82=0
-Xfp-pedantic
-X82=1
```

Description

Some compiler optimizations do not conform strictly to the IEEE754 floating point standard. These options control which optimizations the compiler can perform with floating point operations, and how closely it must conform to the standard.

-Xfp-fast**-X82=2**

Favor floating-point performance over conformance to the IEEE754 floating-point standard.

The **-Xfp-fast** option allows floating-point division by a constant to be optimized into a multiply by the reciprocal of the constant. This optimization is inhibited for **-Xfp-normal** and **-Xfp-pedantic** unless the constant is a power of two.

If auto-vectorization (**-Xvec**) is enabled together with **-Xfp-fast**, the compiler is allowed to vectorize some operations (for example, reductions), which can alter the order of expression evaluation.

-Xfp-normal**-X82=0**

Use normal (relaxed) conformance to the IEEE754 floating-point standard. This is the default. Optimizations disabled by **-Xfp-pedantic** are enabled by **-Xfp-normal**.

-Xfp-pedantic**-X82=1**

Enable strict implementation of the IEEE754 floating point standard at some cost in performance. This ensures that all calculations allow for **NaN**. Specifically,

Do not optimize a divide by a constant to a multiply of its reciprocal. That is, do not change
(fp / C)

into
(fp * (1/C))

Do not optimize calculations of the type **(fp - fp)** to zero. Instead, calculate at run time so that possible **NaN** values are preserved. Similarly, always calculate the following at run time: **(fp == fp)**.

When performing comparisons of the type **fp1 <= fp2** and **fp1 >= fp2**, account for **NaN** values as defined by the standard.

On processors where hardware floating point operations are always calculated in extended precision, truncate the intermediate results to take rounding issues into account.

This option is equivalent to using **-Xieee754-pedantic**.

See Also

[-Xieee754-pedantic on page](#)

[-Xvec on page](#)

-Xfp-long-double-off, -Xfp-float-only

Convert double and long double.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xfp-float-only
-X70=2
```

Description

Force **double** and **long double** to be the same as **float**.

Synonym: **-Xno-double**.

-Xfp-long-double-off

-X70

Force **long double** to be the same as **double** on machines where they differ.

Synonym: **-Xno-long-double**.

Note

If this option is used, libraries containing functions with floating point parameters must be recompiled. For safety, recompile all libraries to avoid missing any such functions. Also, operation of library routines designed to process a suppressed type is undefined.

-Xfp-min-prec...

Specify minimum floating point precision.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xfp-min-prec-double
-X3=0
```

Description

Use **double** as the minimum precision in expressions and for floating point arguments. Lesser precisions are used in expressions if the **-Xdialect-ansi** option is used. If prototypes are used, use the declared precision for arguments, unless the **-Xextend-args** option is used.

Synonym: **-Xuse-double**.

-Xfp-min-prec-float

-X3=1

Use **float** as the minimum precision in expressions and for floating point arguments.

Synonym: **-Xuse-float**.

-Xfp-min-prec-long-double

-X3=2

Use **long double** as the minimum precision in expressions and for floating point arguments. Lesser precisions are used in expressions if the **-Xdialect-ansi** option is used.

If prototypes are used, use the declared precision for arguments, unless the **-Xextend-args** option is also given.

Synonym: **-Xuse-long-double**.

Note

If this option is used, libraries containing functions with floating point parameters must be recompiled. For safety, recompile all libraries to avoid missing any such functions. Also, operation of library routines designed to process a suppressed type is undefined.

-Xframe-info

Generate `.frame_info` for C functions.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xframe-info
```

Description

Force the compiler to generate **.frame_info** sections for C functions. Use this option when compiling mixed C/C++ programs in which C++ exceptions may propagate back through C functions.

For more information, see the discussion of **.frame_info** sections in the *Wind River Diab Compiler Utilities Reference: D-DUMP File Dumper*.

-Xfull-pathname

Include filename path in debug information.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xfull-pathname
-X125
```

Description

Include the path prefix in filenames in debug information (specifically, in the **.file** assembler directive). Without this option, only the filename is included.

-Xgcc-options-off

Disable GNU option translator.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xgcc-options-off
-X194=1
```

Description

Disable automatic translation of GNU compiler (GCC) options.

See Also

[-Xgcc-options-on on page](#)

-Xgcc-options-on

Enable GNU option translator.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xgcc-options-on
-X194=0
```

Description

Enable automatic translation of GNU compiler (GCC) options. This is the default.

When **-Xgcc-options-on** is enabled, GCC option flags from the command line or makefile are parsed and, if possible, translated to equivalent Wind River Diab Compiler options. Translations are determined by the tables in the file **versionDir/conf/gcc_parser.conf**.

See Also

[-Xgcc-options-off on page](#)

-Xgcc-options-verbose

Display GNU option translations.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xgcc-options-verbose
-X195
```

Description

Display all translations. Valid only if translation is enabled (**-Xgcc-options-on**).

See Also

[-Xgcc-options-on on page](#)

-Xgnu-asm-usage

generate warning or error for GNU style inline asm.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-Xgnu-asm-usage
```

Description

option to generate a warning or error for GNU style inline asm

The option `-Xgnu-asm-usage` is supported with EDG based front-end and this is enabled by using `-Xc-new` option. Options like `-Xdialect-c99` or `-Xlibc-new` implicitly enables `-Xc-new`.

`-Xgnu-asm-usage=0`: pass

`-Xgnu-asm-usage=1`: warning (default)

`-Xgnu-asm-usage=2`: error

Note

If `-Xgnu-asm-usage` is specified but not set to any value it is treated as `-Xgnu-asm-usage=0`. If `-Xgnu-asm-usage` is not specified at all it is treated as `-Xgnu-asm-usage=1` (default)

Since

5.9.4.6

See Also

[-Xc-new on page](#)
[-Xdialect... on page](#)
[-Xlibc... on page](#)

-Xhll-symbol

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xhll-symbol
```

Description

Generate high level language information into object files.

See Also

[-Xsymbol-debug on page](#)

-Xident-off

Do not pass `#ident` strings.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xident-on  
-X63=0
```

Description

Pass `#ident` strings to the assembler. This is the default.

-Xident-off

-X63

Do not pass **#ident** strings to the assembler.

Synonym: **-Xno-ident**.

-Xieee754-pedantic

Enable strict implementation of IEEE754 floating point standard.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xieee754-pedantic
-X82=1
```

Description

This option is equivalent to **-Xfp-pedantic**.

See Also

[-Xfp-pedantic on page](#)

-Xif-conversion

Enable optimization of conditional jumps.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xif-conversion
```

Description

Enable the transformation of conditional jumps into branch-less or simpler control path code sequences. This option requires that either one of the **-O** options or **-XO** switch is also present to be effective.

This is automatically enabled at -O3 and -O4.

Since

5.9.4.0

See Also

[-O on page](#)

[-XO on page](#)

[-Xno-if-conversion on page](#)

-Ximport

Treat #include as #import.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Ximport
-X75
```

Description

Treat all **#include** directives as if they are **#import** directives. This means that any include file is included only once.

-Xincfile-missing-ignore

Ignore missing include files.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xincfile-missing-ignore
-X172
```

Description

This option, which suppresses error reporting, is effective only when used with **-Xmake-dependency**. It causes preprocessing to continue even when a required header is not found. If **-Xincfile-missing-ignore** is used with **-Xmake-dependency=2** or **-Xmake-dependency=6**, the preprocessor issues a warning (but not an error) when a required system file (**#include <filename>**) is not found.

See Also

[-Xmake-dependency on page](#)

-Xinit-locals

Initialize local variables.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xinit-locals=mask
-X87=mask
```

Description

Initialize all local variables to zero or the value specified with **-Xinit-value** at every function entry. **mask** is a bit mask specifying the kind of variables to be initialized.

mask may be given in hex, e.g., **-Xinit-locals=0x9**. Mask bits may be OR-ed to select more than one. Undefined mask bits are ignored.

0x1 integers

0x2 pointers

0x4 floats

0x8 aggregates

If **n** is not given, all local variables will be initialized.

This option is useful in finding “memory dependent” bugs.

-Xinit-section

Control generation of initialization and finalization sections.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xinit-section=n
-X91=n
```

Description

This option controls generation of sections for run-time initialization and finalization invocation, including constructor and destructor functions and global class objects in C++. For more information, see the discussion of run-time initialization and termination in the *Use in an Embedded Environment* chapter of the *Wind River Diab Compiler User's Guide* for your architecture .

-Xinit-section=0

Suppress generation of initialization and finalization sections. This option is not recommended and may result in incorrect run-time behavior.

-Xinit-section

-Xinit-section=1

Create **.ctors** and **.dctors** sections containing pointers to initialization and finalization functions, sorted by priority. This is the default.

Initialization and finalization functions are designated with attribute specifiers. See the discussion of constructor and destructor attributes in the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xinit-section=2

Create **.init\$nn** and **.fini\$nn** code sections containing calls to initialization and finalization functions, sorted by priority. Provides compatibility with previous versions of the compiler, including recognition of old-style function prefix designations for initialization and finalization functions.

Synonym: **-Xuse-init**.

-Xinit-section-default-pri

Control default priority for initialization and finalization sections.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xinit-section-default-pri=n
-X175=n
```

Description

Assign the default priority for constructor and destructor functions and for C++ global class objects. The specified priority **n** applies to functions referenced in **.ctors**, **.dtors**, **.init**, and **.fini** sections. Functions with lower priority numbers execute first.

-Xinit-value

Define initial value for -Xinit-locals.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xinit-value=n
-X90=n
```

Description

Define the initial value used by **-Xinit-locals**. This option can be useful to identify uninitialized variables, since it can be used to initialize variables to some invalid or recognizable value that might produce a memory access error.

The value **n** is 32-bits, right-justified, zero-filled and may be specified as a decimal or hexadecimal number (**0x...**).

See Also

[-Xinit-locals on page](#)

-Xinline

Inline functions with fewer than n nodes.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xinline=n
-X19=n
```

Description

Set the limit on the number of nodes for automatic inlining. (Roughly, there is one node for each operator or operand.) Because the compiler collects functions until the number of nodes specified by **-Xparse-count** are reached, **the inlined function does not need to be defined before the function using it**. For a information about inlining, see **-Xparse-count**. See also `always_inline` Attribute, in the *Additions to C and C++* and *Optimization* chapters of the *Wind River Diab Compiler User's Guide* for your architecture.

For purposes of automatic inlining, nodes that do not correspond to an operator or operand are not counted. Hence setting **-Xinline** to 0 inlines no functions automatically, and setting **-Xinline** to 1 inlines only "dummy" functions containing no code.

Defaults: **-Xinline** is 10 by default. **-XO** sets **-Xinline** to 40 by default, except if it is used in conjunction with **-Xsize-opt**, in which case it is 10.

Note

Inlining occurs only if optimization is selected by using the **-XO** or **-O** option.

See Also

[-Xparse-count on page](#)
[-Xunroll on page](#)

-Xinline-asm-off

Disable inlining for functions with ASMs.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xinline-asm-off=mask
-X443
```

Description

Disable inlining for functions that contain inline assembly (**asm** strings), assembly macros, or intrinsic functions.

mask is an optional value that controls which types of function are affected. Set the following mask bits to selectively disable inlining for specific function types:

- 0x1, functions containing **asm** strings
- 0x2, functions containing **asm** macros
- 0x4, functions containing intrinsic assembly

The default value for **mask** is 0x3, disable inlining for functions containing **asm** strings or **asm** macros.

This affects optimization. In many cases it may be better to disable inlining for individual functions instead. See the discussion of the **noinline** attribute in the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

Note

Some processor-specific extensions are implemented using intrinsic assembly. This includes the Neon extensions on ARM processors; and AltiVec SPE , and LSP extensions on PowerPC processors. Setting a mask bit of 0x4 disables inlining of functions using these extensions.

When this option is specified, affected functions will not be inlined even if they are marked with **__attribute__((always_inline))**, and no warning will be given for this case.

See also the *Embedding Assembly Code* chapter, and the discussion on inlining in the *Optimization* chapter, in the *Wind River Diab Compiler User's Guide* for your architecture.

-Xinline-explicit-force

Allow inlining of recursive function calls.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xinline-explicit-force
-Xinline-explicit-force=n
```

```
-X163
-X163=n
```

Description

Inline recursive function calls up to **n** times. The default is 50. If this option is not used, the compiler inlines a function at most once.

If this option is combined with **-Xinline=0**, the compiler inlines only functions declared within a C++ class or with **inline**, **__inline__**, or **#pragma inline**.

This option is overridden by **-Xexplicit-inline-factor**. (See **-Xdefer-pop**.) By default, **-Xexplicit-inline-factor=3** is in effect for C++ programs; C++ programmers who want to use **-Xinline-explicit-force** should therefore specify **-Xexplicit-inline-factor=0**.

See Also

[-Xdefer-pop \(x86\) on page](#)

-Xinline-single-use-statics, -Xinline-single-use-statics-off

Enable or disable inlining of single-use static functions.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xinline-single-use-statics
-Xinline-single-use-statics-off
```

Description

When inlining is enabled, always inline calls to static functions that are only called once. Note that "called once" is measured in terms of calls appearing in the original source code. It is possible after inlining that a function that was originally only called once might ultimately be called several times. As a result in some cases this option may lead to increased code size. Use **-Xinline-single-use-statics** to enable this feature, and **-Xinline-single-use-statics-off** to disable it. By default, the feature is disabled.

Since

5.9.4.0

-Xinline-volatile-off

Inhibit inlining of functions that access volatiles.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xinline-volatile-off
```

Description

Disable inlining for all functions that contain accesses to volatile variables. This option is disabled by default.

This affects optimization. In many cases, it may be preferable to disable inlining for individual functions instead; see the discussion of the **noinline** attribute in the *Wind River Diab Compiler User's Guide* for your architecture.

Note

This option was introduced in version 5.9.1 for backward compatibility. Functions containing volatile variables were not previously inlined. We do not recommend setting this option for new projects.

When this option is specified, affected functions will not be inlined even if they are marked with `__attribute__((always_inline))`, and no warning will be given for this case.

-Xinline-without-body-warn

Warn about inlined functions with no definition.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xinline-without-body-warn
```

Description

Issue a warning if the compiler perceives an inlined function to have no content. This typically happens when an inlined function is declared in one file and defined in another.

-Xinsert-nop-end

Insert nops into end of functions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xinsert-nop-end=n  
-X52=n
```

Description

Insert **n** nops after the epilog of each function. The purpose is to allow emulators to insert instrumentation code.

-Xinsert-nop-init

Insert nops into beginning of functions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xinsert-nop-init=n  
-X51=n
```

Description

Insert **n** nops before the prolog of each function. The purpose is to allow emulators to insert instrumentation code.

-Xint-reciprocal

Allow division by reciprocal-multiply when optimizing.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xint-reciprocal
-X407
```

Description

The optimization of performing integer division by a reciprocal-multiply generates more instructions than using a divide instruction, and so is disabled when **-Xsize-opt** is specified. Use **-Xint-reciprocal** to allow divide-by-reciprocal-multiply even when optimizing for size (i.e., when **-Xsize-opt** is specified).

See Also

[-Xsize-opt on page](#)

-Xintrinsic-mask, -Xdisable-intrinsic-mask

Enable and disable specific intrinsic functions.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xintrinsic-mask=n
-X154=n
```

Description

Enable specified intrinsic functions.

For information about intrinsic functions (and which of them are enabled by default), see the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xdisable-intrinsic-mask

-X456

Disable specified intrinsic functions.

-Xkeep-function-asm

Generate function body if it contains an assembly string or calls an assembly macro.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xkeep-function-asm
```


Description

Normally, static functions that are not called and do not have their address taken are deleted by the optimizer. This option prohibits that action for any function that has an assembly string or that “calls” an assembly macro in its body.

-Xkeywords

Enable extended keywords.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xkeywords=mask
-X78=mask
```

Description

Recognize new keywords according to **mask**, a bit mask specifying which keywords to add.

mask may be given in hex, e.g., **-Xkeywords=0x9**. Mask bits may be OR-ed to select more than one. Undefined mask bits are ignored.

0x01 **extended** (C only)

0x02 **pascal** (C only)

0x04 **inline** (this keyword always available in C++)

0x08 **packed** 0x10 **interrupt** (C only)

PowerPC:

For AltiVec on the PPC7400 or PPC970 processor, the following keywords are enabled by the vector floating point selection.

0x20 **vector** 0x40 **pixel** 0x80 **bool** 0x100 **vec_step**

For e500 processors, the following keywords are available.

0x0010000 **__ev64_u16__**

0x0020000 **__ev64_s16__**

0x0040000 **__ev64_u32__**

0x0080000 **__ev64_s32__**

0x0100000 **__ev64_u64__ S**

0x0200000 **__ev64_s64__**

0x0400000 **__ev64_fs__** 0x0800000 **__ev64_opaque__**

For more information about these key words, see the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

PowerPC: For a detailed description of the AltiVec-related keywords, see the AltiVec Technology Programming Environments Manual; and for information about e500 keywords, see the SPE Programming Interface Manual.

For example, enable the **packed** (0x8) and **interrupt** (0x10) keywords using the following:

```
dcc -Xkeywords=0x18 file.c
```

Keywords can also be selectively disabled with **-Xdisable-keywords**.

See Also

[-Xdisable-keywords on page](#)

-Xkill-opt

Disable target-dependent optimizations.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xkill-opt=mask
-X27=mask
```

Description

Note

This option is reserved for internal Wind River use. It should be used only on the advice of Wind River Customer Support. **mask** is a bit mask with one bit for each optimization type. **mask** may be given in hex, e.g., **-Xkill-opt=0x12**. Multiple optimizations can be disabled by OR-ing their mask bits. Undefined mask bits are ignored. **-Xkill-opt=0xffffffff** has a similar (but not exactly the same) effect as not using the **-O** option at all.

For more details about the optimizations, see the discussion of target-independent optimizations in the *Wind River Diab Compiler User's Guide* for your architecture.

The mask bits are as follows:

- Various Optimizations (0x1)
- Tail Recursion (0x2)
- Inlining (0x4)
- Argument Address Optimization (0x8)
- Structure Members to Registers (0x10)
- Disable generating arrays as jump tables for switch blocks—equivalent to **-Xswitch-array-off** (0x20)
- Assignment Optimization (0x80)
- Tail Call Optimization (0x100)
- Common Tail Optimization (0x200)
- Variable Live Range Optimization (0x400)
- Constant and Variable Propagation (0x800)
- Complex Branch Optimization (0x1000)
- Loop strength reduction (0x2000)
- Loop Count-Down Optimization (0x4000)
- Loop Unrolling (0x8000)
- Global Common Subexpression Elimination (0x10000)
- Undefined variable propagation (0x20000)
- Unused assignment deletion (0x40000)
- Minor Transformations to Simplify Code Generation (0x80000)
- Delayed register saving (0x100000)—for Coldfire.
- Register Coloring (0x200000)
- Interprocedural Optimizations (0x400000)
- Remove Entry and Exit Code (0x800000)
- Use Scratch Registers for Variables (0x1000000)

- Extend Optimization (0x2000000)
- Loop Statics Optimization (0x4000000)
- Loop Invariant Code Motion (0x8000000)
- Static Function Optimization (0x20000000)—for ColdFire, MIPS, TriCore
- Live-Variable Analysis (0x40000000)
- Local Data Area Optimization (0x80000000)

Either the **-O** or **-XO** option must be given to enable optimization before **-Xkill-opt** can be used. To compile with almost no optimization, do not specify **-O** or **-XO**.

Two minor optimizations required by the code generation algorithms cannot be disabled: local strength reduction (e.g., multiply by power of 2 becomes shift or add) and simple branch optimization (e.g., branches to branches).

See Also

[-Xkill-reorder on page](#)

-Xkill-reorder

Disable target-independent optimizations.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xkill-reorder=mask
-X28=mask
```

Description

Note

This option is reserved for internal Wind River use. It should be used only on the advice of Wind River Customer Support. Disable individual target-dependent optimizations in the **reorder** program.

Either the **-O** or **-XO** option must be given to enable optimization before **-Xkill-reorder** can be used. To compile with almost no optimization, do not specify **-O** or **-XO**. See also **-Xkill-opt**.

mask is a bit mask with one bit for each optimization type. **mask** may be given in hex, e.g., **-Xkill-reorder=0x12**. Multiple optimizations can be disabled by OR-ing their mask bits. Undefined mask bits are ignored.

For more details about the optimizations, see the discussion of target-independent optimizations in the *Optimization* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

The mask bits are as follows:

ARM

- General Peephole Optimization (0x8)
- Make Conditional (0x9)
- Simple Scheduling Optimization (0x1000)

ColdFire

- Basic Reordering (0x1)
- Delete TST (0x2)
- General Peephole Optimization (0x8)
- Find Auto-Increment / Decrement (0x10)

- Merge Moves (0x40)
- Simple Scheduling Optimization (0x1000)

MIPS

- Basic Reordering (0x1)
- General Peephole Optimization (0x8)
- Peephole Reaching Analysis (0x20)
- Additional Loop Optimizations (0x400)
- Simple Scheduling Optimization (0x1000)

PowerPC

- Basic Reordering (0x1)
- General Peephole Optimization (0x8)
- Peephole Reaching Analysis (0x20)
- Merge Common Block Entry or Exit Code (0x200)
- Additional Loop Optimizations (0x400)
- rlwinm Peephole Optimization (0x800)
- Simple Scheduling Optimization (0x1000)
- Float-Move Peephole (0x4000)
- Vector-Move Peephole (0x8000)

RH850

- Basic Reordering (0x1)
- General Peephole Optimization (0x8)
- Additional Loop Optimizations (0x400)

TriCore

- General Peephole Optimization (0x8)
- Instruction Scheduling (0x01)
- Post-increment Addressing Optimizations (0x10)
- Space Optimizations (0x200)
- Local Value Numbering (0x20000)
- Extended Value Numbering (0x40000)
- Tail Space Optimization (0x80000)
- Constant Following/folding (0x1000000)

x86

- Basic Reordering (0x1)
- Delete TEST (0x2)
- General Peephole Optimization (0x8)
- Find Auto-Increment / Decrement (0x10)
- Peephole Reaching Analysis (0x20)
- Merge Moves (0x40)
- Simple Scheduling Optimization (0x1000)

See Also

[-Xkill-opt on page](#)

-Xlicense-proxy-path

Change license proxy path.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xlicense-proxy-path=path
-X49
```

Description

Use **-Xlicense-proxy-path** to specify the location of the license proxy process (**licproxy**). Users will seldom need to change the value of this path. Compilation failure will result if proxy use is enabled and **-Xlicense-proxy-path** is set to an invalid location.

See Also

[-Xlicense-proxy-use on page](#)
[-Xlicense-wait on page](#)

-Xlicense-proxy-use

Turn license proxy off.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xlicense-proxy-use=n
-X191
```

Description

By default, the compiler uses a proxy process (**licproxy**) to obtain a license, saving time that might otherwise be spent in contacting a license server for multiple compilations. During an initial compilation, the proxy starts up and obtains a license; subsequent licenses need only contact the proxy, not the server. The proxy stays alive for a few minutes after each compilation.

If **-Xlicense-proxy-use** is set to 0, then the license proxy is disabled.

See Also

[-Xlicense-proxy-path on page](#)
[-Xlicense-wait on page](#)

-Xlicense-wait

Wait for license.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xlicense-wait
-X138
```

Description

If a license is not available, request that the compiler wait and retry once a minute, rather than returning with an error.

See Also

[-Xlicense-proxy-use on page](#)
[-Xlicense-proxy-path on page](#)

-Xlimit-reaching

Disable reaching analysis optimizations.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xlimit-reaching=size
-X=422
```

Description

Disable reaching optimizations when the size limit **size** is exceeded. **size** is defined as the number of nodes times the number of basic blocks. The default value is zero, meaning that reaching is not limited (i.e., that reaching optimizations will run no matter how big the function is).

If you use the following option **-W0,-v** (enables "verbose" mode) and use **-O** or **-XO**, the compiler will print something like:

```
main: Parsing (58 nodes)
```

```
reaching-size=2
```

The "reaching-size" value is the value compared against the size limit for **-Xlimit-reaching**.

See Also

[-Xreduce-reaching on page](#)

-Xlink-time-lint

Perform link-time lint.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xlink-time-lint
-X405
```

Description

Enable the checking of object and function declarations across compilation units, as well as the consistency of compiler options used to compile source files. **-Xlink-time-lint** may be called as an option to the compiler, the linker, or the driver (**dcc**).

Information used by link-time lint is preserved during linking and is passed to the output file, so if a program is linked incrementally, **-Xlink-time-lint** can be used at any stage of a build.

Note

This option may increase code and data size, and is therefore not recommended for final production builds.

Since **-Xlink-time-lint** causes extra debug information to be generated, it can create references to symbols that would otherwise be unreferenced, meaning that dead code elimination (**-Xremove-unused-sections**) may not be as effective with the option as it would be without it. In some cases this can lead to larger code or data sizes with the option than without.

-Xlint

Generate warnings on suspicious/non-portable code.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xlint[=mask]
-X84[=mask]
```

Description

Generate warnings when suspicious and non-portable C code is encountered. For C++ modules, see note below. The two usual cases are:

-Xlint enables all warnings (equivalent to **-Xlint=1**).

-Xlint=0xffffffff disables all present and future warnings (equivalent to **-Xlint=0** or the default of not using the option at all).

Individual warnings can be disabled by OR-ing the following values. In effect, **-Xlint=1** is assumed, enabling all warnings, and then individual warnings are disabled. **mask** may be given in hex, e.g., **-Xlint=0x1a**. Undefined bits are ignored.

0x02

Variable used before being set.

0x04

Label not used.

0x08

Condition always true/false, for example, **i==i**.

To suppress warnings for conditional constructs for which you don't want a warning, add an extra set of parentheses, e.g.:
do {} while ((0));

0x10

Variable/function not used

0x20

Missing return expression.

0x40

Variable set but not used.

0x80

Statement not reached.

0x100

Conversion problems.

0x200

In non-ANSI mode, warn when the compiler selects an unsigned integral type for an expression which would be signed under ANSI mode. For example:

"a.c", line 3: warning (1671):

non-portable behavior: type of

'>'

operator is unsigned only

in non-ANSI mode

0x400

Possibly assignment (=) should be comparison (==).

0x1000

Missing function declaration (equivalent to **-Xforce-declarations**).

0x2000

Possible redundant expression. (Examples: **x=x**, **x&x**, **x|x**, **x/x**.)

For an example of a program that generates most of the **-Xlint** warnings, see the *Wind River Diab Compiler Utilities Reference: Lint Facility*.

For information about using the **_lint** macro to avoid use of non-ANSI extensions in header files, see the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

Note

For C++, **-Xlint** is equivalent to **-Xsyntax-warning-on**.

See Also

[-Xsyntax-warning-... on page](#)

-Xlno

Enable loop nest optimizations.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xlno
```

Description

This switch enables the compiler to attempt to reorder and aggressively unroll loops, using heuristics designed to improve the memory locality and pipeline performance of code nested within loops. This option requires that the **-O** or **-XO** switch is also present to be effective.

This is automatically enabled at -O3 and -O4.

Since

5.9.4.0

See Also

[-O on page](#)

[-XO on page](#)

-Xlocal-data-area

Allocate static and global variables to local data area.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xlocal-data-area=n
-Xl15=n
```

Description

Allocate the static and global variables which are defined in a module and referenced at least once into a contiguous block of memory, called the local data area (LDA), and make fast, efficient references to those variables via a temporary base register selected by the compiler.

n specifies the maximum of the LDA, and defaults to:

- 64 bytes for ARM, RH850, TriCore, and x86.
- 32,767 bytes for ColdFire, MIPS, PowerPC.

(If **n** is greater than the default, references to variables in the LDA will be less efficient.)

The optimization does not apply to unreferenced variables. Also, for ColdFire, MIPS, PowerPC and TriCore it does not apply to variables smaller than the size set with **-Xsmall-data**, which go in the **.sdata** or **.sbss** sections.

-Xlocal-data-area should be used with caution in multithreaded environments. To restrict the optimization to static variables, use **-Xlocal-data-area-static-only**; VxWorks developers are strongly advised to use this option.

For additional information, see discussion of local data areas in the *Wind River Diab Compiler User's Guide* for your architecture.

Synonym: **-Xlocal-struct**.

Note

If at least one variable in the LDA has an initial value, the LDA is in the **.data** section; otherwise it is in the **.bss** section. Because **-Xlocal-data-area** is nonzero by default, uninitialized static and global variables—or for ColdFire, MIPS, PowerPC, and TriCore, global variables larger than the size set with **-Xsmall-data**—that are referenced at least once are not stored in a **.bss** section. To store such variables in **.bss**, use **--Xlocal-data-area=0**.

See Also

[-Xsmall-data on page](#)

-Xlocal-data-area-static-only

Restrict local data area optimization to static variables.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xlocal-data-area-static-only
-Xl66
```

Description

Apply the local data area optimization only to static variables; do not optimize global variables.

For information about this optimization, see discussion of local data areas in the *Locating Code and Data, Addressing, Access* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xlocals-on-stack

Do not assign locals to registers.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xlocals-on-stack
-X5
```

Description

By default, the compiler attempts to assign all local variables to registers. If **-Xlocals-on-stack** is given, only variables declared with the **register** keyword are assigned to registers.

-Xlto-group

Synopsis

```
-Xlto-group = "name of link time optimization group"
```

Description

Name of Link time Optimization (LTO) group to which this file belongs to.

Group names are case-insensitive and can only contain alphanumeric characters and underscores.

Link time optimizations such as `-Xwhole-program-optim` and `-Xcode-factor` will only perform inter-module optimizations for modules that lie in the same LTO group. This option can be used to partition your application into isolated components each of which will be separately optimized at link time.

Files that are not compiled with an explicit `-Xlto-group` option go into a default group named "DefaultLTOGroup".

Note

Specifying an LTO group does not automatically enable any link time optimizations. It simply specifies what group this file should be part of if it does participate in any link time optimizations.
The linker option `-Xremove-unused-sections` always operates across the entire program, regardless of LTO groups.

-Xmac-conventions

Use Macintosh calling conventions for pascal functions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xmac-conventions
-X79
```

Description

Use Macintosh calling conventions for pascal functions instead of standard 68K calling conventions. For more information, see the discussion of argument passing in the *Calling Convention* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xmacro-in-pragma

Expand macros in pragmas.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xmacro-in-pragma
-X157
```

Description

Expand preprocessor macros in **#pragma** directives.

-Xmacro-patterns

Use extended preprocessor syntax.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xmacro-patterns
-X457
```

Description

If **-Xmacro-patterns** is defined the compiler (both ctoa and etoa) supports the following extended preprocessor syntax:

```
/* true if any macro whose name starts with ABC_1 is defined */
#ifdef "ABC_1*"
/* true if any macro whose name ends with BC_1 followed by any single character is defined */
#if defined("*BC_1?")
/* true if no macro whose name starts with ABC_ and ends with 25 is defined */
#ifndef "ABC_*25"
/* undefs all macros whose names start with ABC_2 */
#undef "ABC_2*"
Note that #define is missing from this list as #define with a wildcard does not make any sense.
```

The wildcard syntax follows the standard UNIX glob notation and supports two wildcard characters:

Any sequence of characters (including the empty string).

?

Any single character.

-Xmacro-undefined-warn

Warn on undefined macro in #if statement.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xmacro-undefined-warn
-X171
```

Description

Generate a warning when an undefined macro name occurs in a **#if** preprocessor directive.

-Xmake-dependency

Show make rules.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xmake-dependency
-Xmake-dependency=mask
-X156
-X156=mask
```

Description

Generate a list of include files required to build each object file. Example:

```
main.o: main.c stdio.h
```

```
command list
```

This output means that `main.c` and `stdio.h` are required to build the target `main.o`. A list of make commands follows the dependency.

mask, which defaults to 1, is a bit mask—always interpreted as hexadecimal—of which the four least significant bits are meaningful: the fourth (least significant) bit, if set to 1, means that all required files are shown; this is the default. The third bit means that only files enclosed in double quotation marks (**#include"filename"**) are shown. (If both the third and the fourth bits are set, the fourth overrides the third.) The second bit means that compilation continues after the dependency list is generated (if this bit is 0, no output is emitted other than the list of dependencies) and that the dependency list is sent to a file (instead of the standard output). The first bit creates a “phony target” for each dependency other than the main file; this is a work around for errors caused by missing header files and is provided for GNU compatibility. The **-o** option can be used to specify the output file, the target name, or both. Hence:

-Xmake-dependency=1

Same as **-Xmake-dependency**. Show all required include files. If **-o** is used, the target is the name specified with **-o**. Results go to the standard output unless **-Xmake-dependency-savefile=filename** is specified. No further output is emitted.

-Xmake-dependency=2

Same as **-Xmake-dependency=1**, but show only files enclosed in double quotation marks (**#include"filename"**).

-Xmake-dependency=4

Same as **-Xmake-dependency=1**, but write the dependency list to a file and then continue with normal compilation. The output file can be specified with either **-o** or **-Xmake-dependency-savefile=filename** (which overrides **-o**); otherwise it is called **filename.d**, where **filename** is the name of the main source file, and is created in the directory where the compiler was invoked. If **-o** is used without **-Xmake-dependency-savefile**, the output file is the basename specified by **-o** with **.d** appended.

-Xmake-dependency=8

Same as **-Xmake-dependency=1**, but output a phony target for each dependency other than the main file. The bits can be OR-ed to combine options. Example:

-Xmake-dependency=6

Show only files enclosed in double quotation marks (**-Xmake-dependency=2**); write output to a file, then continue with normal compilation (**-Xmake-dependency=4**).

-Xmake-dependency=a

Show only files in double quotation marks (**-Xmake-dependency=2**) and output phony targets (**-Xmake-dependency=8**).

-Xmake-dependency=c

Output phony targets (**-Xmake-dependency=8**); write output to a file, then continue with normal compilation (**-Xmake-dependency=4**).

-Xmake-dependency=e

Show only files enclosed in double quotation marks (**-Xmake-dependency=2**); output phony targets (**-Xmake-dependency=8**); write output to a file, then continue with normal compilation (**-Xmake-dependency=4**).

Ordinarily, the preprocessor returns an error and stops when a required file is not found. To continue preprocessing when files are missing, use **-Xmake-dependency** with **-Xincfile-missing-ignore**.

On a Windows host, **-Xmake-dependency** translates path separator characters from back slashes to forward slashes. This may be required if the output is processed by utilities such as **gmake**. To disable this translation, use the option **-Xmake-dependency-canonicalize-path-off** in conjunction with **-Xmake-dependency**.

See Also

[-Xincfile-missing-ignore on page](#)
[-Xmake-dependency-canonicalize-path-off on page](#)
[-Xmake-dependency-savefile on page](#)
[-Xmake-dependency-target on page](#)

-Xmake-dependency-canonicalize-path-off

Disable translation of path separators.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xmake-dependency-canonicalize-path-off
```

Description

Disable translation of path separators in the output from **-Xmake-dependency**. This only applies to a Windows host.

See Also

[-Xmake-dependency on page](#)

-Xmake-dependency-savefile

Specify dependency output file.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xmake-dependency-savefile=filename
```

Description

This option is valid only when used with **-Xmake-dependency**.

Specify the output file for **-Xmake-dependency**.

See Also

[-Xmake-dependency on page](#)

[-Xmake-dependency-target on page](#)

-Xmake-dependency-target

Specify dependency name.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xmake-dependency-target=string
```

Description

This option is valid only when used with **-Xmake-dependency**.

Change the target name in the rule emitted by **-Xmake-dependency** to **string** (instead of using the name of the object file). To specify multiple target names, repeat the **-Xmake-dependency-target** option on the command line.

See Also

[-Xmake-dependency on page](#)

[-Xmake-dependency-savefile on page](#)

-Xmember-max-align

Set maximum structure member alignment.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xmember-max-align=n
-X88=n
```

Description

Set the maximum byte boundary to which structure members will be aligned. If the natural alignment of a member is less than **n**, the natural alignment is used for it. See the discussion of the **pack** pragma and the **__packed__** and **packed** keywords in the *Additions to C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture for details. See also - **Xstruct-min-align**.

The default value of **n** is dependent on the processor as described in the *Internal Data Representation* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

Synonym: **-Xstruct-max-align**.

See Also

[-Xstruct-min-align on page](#)

-Xmemory-is-volatile, -Xglobals-volatile, -Xstatics-volatile, -Xpointers-volatile

Treat all variables as volatile.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xmemory-is-volatile
-X4
-X4=7
```

Description

Treat all variables as volatile. The following are related:

-Xglobals-volatile

-X4=1

Treat all global variables as volatile.

-Xstatics-volatile

-X4=2

Treat all static variables as volatile.

-Xpointers-volatile

-X4=4

Treat all pointer accesses as volatile

These options tell the compiler not to perform optimizations that can cause device drivers or other systems to fail. By default, the compiler keeps data in registers as long as possible whenever it is safe. Difficulties can arise if a memory location changes because it is mapped to an external hardware device and the compiler, unaware of the change, continues to use the old value stored in a register. While these situations can now be handled with the **-volatile** keyword, the **-X4** options allow compilation of older programs.

To combine these options, use the sum of their values with a single occurrence of the option flag. For example, use **-X4=3** to treat all global and static variables as volatile.

Note that **-X4=7**, equivalent to **-X4 (-Xmemory-is-volatile)**, which combines all of the options.

-Xmetadata

Activate or deactivate metadata generation.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xmetadata=n
```

Description

The compiler can add metadata to output object files (for detailed information, see **-Xdump-metadata** in the *Linker User's Guide*). The metadata contains information about the version of the compiler, the switches that are used, and so on. The **-Xmetadata** switch can be used to enable or disable metadata generation. Set **n** to **1** to enable metadata generation, and set it to **0** to disable metadata generation. Metadata generation is enabled by default. Diab provides the following equivalents to **-Xmetadata=0**:

- Xmetadata-off
- Xno-metadata

-Xmismatch-warning

Warn on type and argument mismatch.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xmismatch-warning
-X2
-Xmismatch-warning=2
-X2=2
```

Description

Generate a warning only (instead of a fatal error) when either pointers of different types, or pointers and integers, are mixed in expressions. -Example:

```
long i1, i2 = &i1;
```

is invalid in ANSI C but is allowed in some non-ANSI dialects. This option is set implicitly by **-Xdialect-pcc(-X7=3)**.

If the option **-Xmismatch-warning=2** is given, the compiler also generates a warning instead of an error when identifiers are redeclared and when a function call has the wrong number of arguments.

This option is ignored when compiling C++ modules.

Note

-Xmismatch-warning and **-Xmismatch-warning=2** override the **-e** option. If either form of **-Xmismatch-warning** is used, mismatched types will only produce a warning, even if **-e** is used to increase the severity level of the diagnostic.

See Also

[-e on page](#)

-Xname-...

Specify section name.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xname-type=name
```

Description

Use the following options to specify the name of a default section.

-Xname-code=name

Set the section name for code.

-Xname-const=name

Set the section name for initialized constants.

-Xname-data=name

Set the section name for initialized **data**.

-Xname-eh=name

C++ only.

Set the section name for all exception-handling tables.

-Xname-internal-const=name

Set the section name for internal constants generated by the compiler; for example, tables used to efficiently implement **switch** statements. If the section name is not set, the internal constants will be put in a default section for constants (depending on the target architecture and the size of the constants).

-Xname-rtti=name

C++ only.

Set the section name for all RTTI tables.

-Xname-sconst=name

Set the section name for initialized small **const**.

-Xname-sdata=name

Set the section name for initialized small **data**.

-Xname-string=name

Set the section name for strings.

-Xname-uconst=name

Set the section name for uninitialized constants.

-Xname-udata=name

Set the section name for uninitialized **data**.

-Xname-usconst=name

Set the section name for uninitialized small **const**.

-Xname-usdata=name

Set the section name for uninitialized small **data**.

-Xname-vtbl=name

C++ only.

Set the section name for all virtual-function tables.

Section names can also be specified using the **section** pragma. For example, setting **-Xname-code=.code** has the same effect as:

```
#pragma section CODE ".code"
```

For additional information, see the discussion of the **section** pragma the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xno-builtin

Disable Diab built-in functions.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xno-builtin
```

Description

Disable Diab built-in functions (like `__diab_memset` and `__diab_memcpy`).

If the optimizer encounters a loop like this:

```
for (int j = 0; j < 128; j++) {
  for (int i = 0; i < 128; i++) {
    a[j][i] = 0;
  }
}
```

It might transform it to the following:

```
_diab_memset(a, 0, 128*128*sizeof(a[0][0])).
```

Use **-Xno-builtin** to disable this kind of transformation.

-Xno-gnu-inline-asm

(ARM, PPC, TC, RH850 only) Turn off extended GNU inline assembly extension use

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xno-gnu-inline-asm
```

Description

This option turns off any use of GNU inline assembly extended syntax. It is available only with the etoa front-end, which must be explicitly invoked for targets for which etoa is not the default. To explicitly invoke etoa, use the option **-Xdialect-c99** or **-Xc-new**. For information about the GNU inline assembly extended syntax, see the Wind River Diab Compiler User's Guide.

Since

5.9.4.0

See Also

[-Xdialect-... on page](#)

[-Xc-new on page](#)

-Xno-if-conversion

Disable optimization of conditional jumps.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xno-if-conversion
```

Description

Disable the transformation of conditional jumps into branch-less or simpler control path code sequences.

This is the default.

Since

5.9.4.0

See Also

[-O on page](#)
[-XO on page](#)
[-Xif-conversion on page](#)

-Xno-wpo-dead

Disable whole program dead code and data elimination.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xno-wpo-dead
```

Description

Disable -Xwpo-dead which is enabled by default when **-Xwhole-program-optim** is set.

See Also

[-Xwpo-dead on page](#)
[-Xwhole-program-optim on page](#)
[-Xwpo-inline on page](#)
[-Xwpo-inline-single-calls on page](#)
[-Xwpo-const on page](#)
[-Xwpo-no-alias on page](#)
[-Xwpo-delete-return-value on page](#)

-Xold-inline-asm-casting

Use old inline assembly casting.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xold-inline-asm-casting  
-X137
```

Description

This option affects small arguments to **asm** macros (arguments with size less than **int**).

By default, the compiler does not extend such arguments to **int**. Prior to version 4.2, the compiler did extend such arguments to **int**. Use this option to force the old behavior for compatibility with existing **asm** macros which depend on it.

-Xopt-count

Execute the compiler's optimizing stage **n** times.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xopt-count=n
-X25=n
```

Description

Execute the compiler's optimizing stage **n** times. The default is once. In most cases this is enough. In rare instances, one stage of the optimizer will generate an opportunity for a previous stage. Setting **-Xopt-count=2** or more will cause a somewhat longer compilation time but may produce slightly better code. This option is set to 2 by **-XO**.

-Xoptimistic-aliasing

Permit more optimizations.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xoptimistic-aliasing
```

Description

Permit more optimizations by assuming that an expression such as ***(p + e)** or **p[e]** can never refer to part or all of an ordinary variable, unless **p** is a **char ptr**, or **e** is the constant **0**.

Do not compile code with this mask if it contains an expression **p[e]** that might alias a global variable **v**, where **p** is a pointer to the same type as **v**. If **p** is a **char***, then there is no problem. If **p** is a pointer to any other type, then the code is illegal according to the aliasing rules defined by the ANSI C Standard.

Notice that referring to a global variable by indirecting through a pointer (***p**, or **p[0]**, or ***(p + 0)**) is fine. The non-aliasing assumption is only made about expressions in which the pointer is added to a non-constant-zero offset.

This option is equivalent to **-Xrelax-alias=0x33**.

See Also

[-Xrelax-alias on page](#)

-Xoptimized-debug-off

Disable most optimizations with -g.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xoptimized-debug-off
-X89
```

Description

When using the **-g** option to generate debug information, disable most optimizations and force line numbers in debug information to be in increasing order — assists with debuggers that cannot handle optimized code.

Synonym: **-Xno-optimized-debug**.

See Also

[-g on page](#)
[-Xdebug-mode on page](#)
[-Xdebug-struct-all on page](#)
[-Xoptimized-debug-on on page](#)

-Xoptimized-debug-on

Allow optimizations with **-g**.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xoptimized-debug-on
-X89=0
```

Description

Do not disable optimizations when using **-g**. This is the default.

See Also

[-g on page](#)
[-Xoptimized-debug-off on page](#)

-Xparse-count

Specify optimization buffer size.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xparse-count=n
-X424=n
```

Description

Delay code generation of functions until **n** nodes have been used for internal tables. (A node in this case is generally equivalent to an operator or an operand.) By delaying generation, the compiler can perform interprocedural optimizations such as inlining and register tracking. The default is 300,000 nodes with **-O**, or 600,000 nodes if **-XO** is used.

The highest useful value for a module depends on many factors; it is not practical to calculate it (for some of the factors, see the *Compiler Limits* appendix of the *Wind River Diab Compiler User's Guide* for your architecture). For very large and complex modules, experiment with larger values, e.g., **-Xparse-count =800000**, to see if code size or execution time is reduced.

-Xparse-count replaces **-Xparse-size**, which has been deprecated.

Note

Using a node count that requires more physical memory than is available will cause excessive swapping and slow compilation.

-Xpass-source

Output source as comments.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xpass-source
-X11
```

Description

Output the source as comments in the generated assembly language code.

Note

Using **-Xpass-source** significantly increases the size of generated (temporary) assembly files. In the worst case, this can break the build because of file I/O problems. If a message like "name.s line number: error: read error from temp file" is seen with a large number (like a double digit million), consider removing **-Xpass-source** from the command line.

-Xpic

Generate position-independent code for shared libraries.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xpic
-X62
```

Description

For VxWorks RTP application development. Allows a single copy of a shared library, loaded in a single memory location, to be called by different programs. RTP shared-library code must be compiled with this option.

-Xpragma-section-first, -Xpragma-section-last

Control interpretation of multiple section pragmas.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xpragma-section-first
-Xpragma-section-last
```

Description

These options control the compiler's behavior when multiple **#pragma section** directives are used with different parameters for the same section class. The default is **-Xpragma-section-first**. For additional information, see the discussion of the **section** and **use_section** pragmas the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

If **-Xpragma-section-first** is in effect when a variable or function is defined, the compiler uses the earliest currently-valid **section** pragma that specifies a location for the variable or function.

If **-Xpragma-section-last** is in effect when a variable or function is defined, the compiler uses the last currently-valid **-section** pragma that specifies a location for the variable or function.

-Xpreprocessor-lineno-off

Suppress line numbers in preprocessor output.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xpreprocessor-lineno-off
-X165
```

Description

Suppress line-number information in the preprocessor output. Use this with the **-E** option (send preprocessor output to standard output) when line-number information is not needed.

-Xpreprocessor-old

Use old preprocessor.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xpreprocessor-old
-X155
```

Description

Use the preprocessor from release 4.3. When **-Xpreprocessor-old** is specified, **vararg** macros are not supported and the following options are not available: **-Xmake-dependency**, **-Xmake-dependency-...**, **-Xmacro-in-pragma**, and **-Xcpp-dump-symbols**.

This option is valid only when compiling C modules.

-Xreduce-reaching

Reduce reaching analysis optimizations.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xreduce-reaching=size
-X=423
```

Description

Reduce reaching optimizations when the size limit **size** is exceeded. **size** is defined as the number of nodes times the number of basic blocks. The default value is 6,000,000.

The compiler will perform the expensive reaching analysis fewer times when optimizing the current function than it would otherwise. It does run reaching optimizations at least once to get some optimization improvements, but it will not run them over and over again. To disable reaching analysis altogether after a certain threshold, use **-Xlimit-reaching** instead.

If you use the following option **-W0,-v** (enables "verbose" mode) and use **-O** or **-XO**, the compiler will print something like:

```
main: Parsing (58 nodes)
```

```
reaching-size=2
```

The "reaching-size" value is the value compared against the size limit for **-Xreduce-reaching**.

See Also

[-Xlimit-reaching on page](#)

-Xrelax-alias

Control aliasing rules applied by the compiler.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xrelax-alias=n
-X114=n
```

Description

When a variable can be accessed through a pointer (or a reference), it is said to be aliased. Whenever the compiler sees an access through a pointer it has to make assumptions about which variables in the current scope can be aliased by this access. Only then can it make safe optimization decisions.

The **-Xrelax-alias** option provides control over which aliasing assumptions are made by the compiler. The integer **n** that is passed to **-Xrelax-alias** is interpreted as a bit mask, and the individual bits in the mask are interpreted in the following way:

0x1

If set, types other than **char** may not alias other types.

0x2

Structure members may not alias ordinary variables.

For example if **s** is a struct pointer, **m** is a member, and **v** is a variable, then setting this flag tells the compiler that **s->m** and **v** cannot refer to the same object (or overlapping objects), even if **m** has the same type as **v**, or has **char** type.

Note

Do not use `n & 0x4`. It is no longer supported (as of 5.9.3), and using it may result in incorrect code. Instead, add `0x20` to the default option (giving `0x33`), or equivalently use **-Xoptimistic-aliasing**.

`0x8`

If set, even **chars** may not alias other types.

`0x10`

If set, **voids** may alias other types.

`0x20`

Permit more optimizations by assuming that an expression such as ***(p + e)** or **p[e]** can never refer to part or all of an ordinary variable, unless **p** is a **char ptr**, or **e** is the constant **0**.

Do not compile code with this mask if it contains an expression **p[e]** that might alias a global variable **v**, where **p** is a pointer to the same type as **v**. If **p** is a **char***, then there is no problem. If **p** is a pointer to any other type, then the code is illegal according to the aliasing rules defined by the ANSI C Standard.

Notice that referring to a global variable by indirecting through a pointer (***p**, or **p[0]**, or ***(p + 0)**) is fine. The non-aliasing assumption is only made about expressions in which the pointer is added to a non-constant-zero offset.

The convenience option **-Xoptimistic-aliasing** enables this mask added to the default mask (see **-Xoptimistic-aliasing**).

The default mask is `0x13`.

Note that the default includes that arrays may not alias each other. This is an intentional deviation from the C standard. As a consequence, **-Xstrict-ansi** changes the default mask to `0x13`.

Consider the following example. The **main()** routine allocates a message buffer and then calls the function **generateMessages()** to fill this buffer with some information. Afterwards the buffer is streamed out using **printf()**.

Note how the message buffer is filled: a container **struct** is created that holds a couple of **short** integers. The **struct** is aliased by the unsigned integer pointer **cont_converter**. Then (in the loop) the container is filled with values, and afterwards the integer pointer is used to "serialize" the **struct** into a sequence of unsigned integers.

```
#include <stdlib.h>
typedef
struct container {
    short a, b, c, d, e, f;
} container;
void generateMessages(unsigned int * messageBuff);
int main(int argc, char** argv)
{
    unsigned int * messageBuff = malloc(5*sizeof(container));
    unsigned int i=0;
    generateMessages(messageBuff);
    printf("Streaming out messages\n");
    for (i=0;i<(5*sizeof(container))/4;i++){
        printf("Stream word %d is 0x%08x:\n",
i, *(messageBuff+i));
    }
    return 0;
}
/*
DANGER: This is an example of BAD!! code
*/
void generateMessages(unsigned int * messageBuff)
{
    container cont;
    unsigned int * cont_converter=(unsigned
```

```

int*)&cont;
int i,j;
for (i=0;i<5;i++){
    cont.a=i*6;
    cont.b=i*6+1;
    cont.c=i*6+2;
    cont.d=i*6+3;
    cont.e=i*6+4;
    cont.f=i*6+5;
    for (j=0; j<sizeof(container)/4; j++)
        messageBuff[i* sizeof(container)/4+j]=
cont_converter [j];
}
}

```

If you compile this code without optimization, you will obtain the desired behavior. If, however, you compile it with **-XO**, you will see strange behavior. It seems as if the **messageBuffer** is corrupted.

The reason is that the code above violates the aliasing rules of C. The only type that is actually allowed to alias another type is **char**. But here, a **struct** is aliased using an integer pointer. So the compiler is allowed to assume that accesses to the container can under no circumstances change the content pointed to by **cont_converter**. Consequently, the compiler can aggressively optimize the writes. It may locate them behind the reading loop, or even remove them.

The issue can be solved by compiling with **-Xrelax-alias=0x12** (which is the default mask minus flag 1), thus allowing types other than **char** to alias other types. Now the compiler has to ensure that the writes complete before the reading loop, because it now assumes that **cont_converter** aliases the container structure (which it actually does).

See Also

[-Xoptimistic-aliasing on page](#)

-Xrestart

Restart optimization from scratch.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```

-Xrestart
-X29

```

Description

Restart optimization from scratch if too many optimistic predictions were made.

Compilers may have difficulty predicting the best way to perform specific optimizations when the information needed is not available until a later compiler stage. For example, better code may be produced by moving a loop invariant expression outside the loop if the result can be placed in a register. However, the compiler does not know if any register is available until after register allocation, which is performed later in the compilation.

The compiler uses an optimistic approach which generates optimal code when registers are available but not when all registers are taken. The **-Xrestart** option will restart optimization and code generation if any optimistic prediction is false. This will typically slow the compilation of large functions by a factor of almost two while generating better code. This option is turned on by **-XO**.

-Xreverse-inline

Optimize code by moving some code to new functions.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xreverse-inline
-X449
```

Description

This option reduces code size by factoring out repeated code sequences into new functions. The effectiveness of this optimization is heavily dependent on the structure of the code. For best results, the options should be used in conjunction with:

- **-Xcode-factor** (code factoring across the entire program at link time)
- **-Xsection-split** (create a section per function)
- **-Xremove-unused-sections** (remove sections with unused functions)

This optimization is not suitable for all use-cases because it results in code that may look radically different from the original un-factored object code. Therefore we do not recommend using this switch for debug builds.

See also **-Xremove-unused-sections** in the *Linker User's Guide*.

See Also

[-Xcode-factor on page](#)

[-Xsection-split on page](#)

-Xrtc

Generate code for the run-time error checker (RTEC).

Belongs to:

[Checking and Profiling Index on page 249](#)

Synopsis

```
-Xrtc=mask
-X64=mask
```

Description

With no **mask**, this option directs the compiler to insert checking code for all checks made by the Run-Time Error Checker. Specify **mask** to select specific checks rather than all. Multiple checks can be bitwise OR'd.

0x01

Register and check static (global) variables.

0x02

Register and check automatic variables.

0x08

Pointer reference checks.

0x10

Pointer arithmetic checks.

0x20

Pointer increment/decrement checks.

0x40

Standard function checks; for example **memset()** and **bcopy()**.

0x80

Report source code filename and line number in error logs.

The RTEC facility only detects errors in code that is compiled with **-Xrtc**. If calls are made to code that is not compiled with **-Xrtc**, errors may not be detected.

-Xrtti, -Xrtti-off

Enable run-time type information.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xrtti
-X205=1
```

Description

Enable run-time type information. This is the default.

There are two approaches to generating run-time type information for a class:

- Compile all modules with **-Xrtti**: the run-time type information will be emitted for every module but will be marked COMDAT and collapsed to a single instance by the linker. This is the preferred method.
- For a class declaring one or more virtual functions, compile only the module defining the key function for the class with **-Xrtti**. Key functions are described in the *Internal Data Representation* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

```
-Xrtti-off
-X205=0
```

C++ only. Disable run-time type information. Using this option will save space because the compiler does not need to create type tables.

Synonym: **-Xno-rtti**.

-Xsection-pad

Pad sections for optimized loading.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xsection-pad
-X152
```

Description

Allow the linker to pad loadable sections for optimized loading.

-Xsection-split

Generate data or functions in separate section classes.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xsection-split=split-val
-X129=split-val
```

Description

Generate a separate section class for each function or variable.

split-val can be one of four values:

off

0

Do not split off either functions or variables. A single module generates only one **CODE** section class containing the code for all functions for that module. Either off or zero may be used.

1

Place each function in its own **CODE** section class, and use naming scheme A (see *Naming Scheme A* below). This is the default.

2

Place each global or static variable in its own **DATA** section class, and use naming scheme A (see *Naming Scheme A* below). Note that this has the side effect of disabling the use of a local data area (LDA) optimization and might increase the size of the code, depending on your application.

3

Place each function and each variable in its own section class (**CODE** for functions, **DATA** for variables), and use naming scheme A (see *Naming Scheme A* below). This also disables the use of a local data area (LDA).

4

Same as 0.

5

Same as 1, but with naming scheme B (see *Naming Scheme B* below).

6

Same as 2, but with naming scheme B (see *Naming Scheme B* below).

7

Same as 3, but with naming scheme B (see *Naming Scheme B* below).

Note: This option is especially useful in combination with **-Xremove-unused-sections** to reduce code size.

Using **#pragma use_section** or **__attribute__((section))** operation to place a variable or function in a section prevents it from being split off from the section. If you want to use a user-defined section that will still be split, only use **#pragma section**.

Naming Scheme A

With naming scheme A **-Xsection-split** generates multiple instances of the split section, (that is, for example, one **.text** section for each function), and while linking, a specific section for a given symbol may be singled out using the following linker command language syntax:

```
object-filespec(input-section-name[symbol] ,...)
```

where the bracket characters ("[" and "]") are required, and in this case do not mean optional.

Example: If object file `test.o` contains functions **f1** and **f2**, then the **.text** section for **f1** may be specified as follows:

```
test.o(.text[f1])
```

Naming Scheme B

Using naming scheme B (enabled by using a **split-val** of **5**, **6**, or **7**), the section names generated by **-Xsection-split** default to the form of **base-section-name.symbol**. This can be overridden using the **-Xsection-split-name** switch. Without using **-Xsection-split-name**, the part of the content of the linker command file for the above example would be:

```
test.o(.text.f1)
```

Note: Wild cards can be used in the linker command file. For example, ***(" .text.*")** would catch all splits of the **.text** section in all object files.

See also **-Xremove-unused-sections** in the *Linker User's Guide*.

See Also

[-Xsection-split-name on page](#)

-Xsection-split-name

Specify name pattern for split sections.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xsection-split-name=name-spec
```

Description

The **name-spec** element is a string and can contain the following special placeholders:

- **s** Base section name.
- **n** Function or variable name as it appears in the object file. For C++ this is the "mangled" name.
- **N** Function or variable name as it appears in the source code. Same as **n** for C, but for C++ this is the unqualified "demangled" name.

The base section name is the name of the section where the symbol would have been placed without the split. The default value for **name-spec** is as follows: **s.n**

Example:

- If object file `test.o` contains functions **f1** and **f2**
- And **-Xsection-split=5**
- And **-Xsection-split-name="the_%s_section_of_symbol_%n"**

Then the **.text** section for **f1** is specified as follows:

```
test.o(the_.text_section_of_symbol_f1)
```

Note

Section names must not contain spaces—including section names defined by **-Xsection-split-name**.

For information on how to change the base section name, see the discussion of **#pragma** section in the *Wind River Diab Compiler User's Guide* for your architecture.

-Xsect-pri-off

Disable generation of priority section names.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xsect-pri-off
-X122
```

Description

Disable generation of section names of the form "...\$n" for use by third-party assemblers or linkers unable to process this form of name.

See Also

[-Xsect-pri-on on page](#)

-Xsect-pri-on

Enable generation of priority section names.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xsect-pri-on
-X122=0
```

Description

Enable section names of the form "...\$n". For use of this form, see the discussion of stored sections in the *Wind River Linker* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

This is the default.

See Also

[-Xsect-pri-off on page](#)

-Xsemi-is-comment

Treat semicolons as comment instead of a statement separators.

Synopsis

```
-Xsemi-is-comment
```

Description

In embedded assembly code, treat the semicolon (;) as a comment character instead of a statement separator. This is useful for GNU compatibility.

-Xsemi-is-newline

Treat semicolons as statement separators.

Synopsis

```
-Xsemi-is-newline
```

Description

In embedded assembly code, treat the semicolon (;) as a statement separator instead of a comment character. This is useful exclusively for assembly macros like for instance `asm("nop;nop");`

-Xshow-configuration

Control listing of -X options in assembly output.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xshow-configuration=n  
-X179=n
```

Description

When **n** = 0, compiler-generated assembly listings (saved with the **-S** option) do not show **-X** options. This is the default.

When **n** = 1, assembly listings contain **-X** options, but only user-configurable options are shown; internal compiler flags are suppressed.

-Xsmall-const

Set size limit for "small const" variable.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xsmall-const=n  
-X98=n
```

Description

This option applies only to chips supporting **SCONST**.

Use register relative addressing mode (r2-relative for PPC) to access all small **const** static and global variables with a size in bytes less than or equal to *n*. For small **const** variables defined in current compilation unit, place them in the **SCONST** section class.

For more information, see the discussion of the **section** pragma the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

Note

A base register (r2 for PPC) points to the Small Const Area (SDA2), and register relative addressing mode can be used to access the limited space (a maximum size of 64K for PPC). For **const** variables in SDA2, it will be always successful to use the register relative addressing mode to access them. However, for non-SDA2 **const** variables, it may fail during the linking process. It all depends on whether they are within the address space relative to the base register.

Therefore, for small **const** variables that are accessed but not defined in current compilation unit, make sure they are placed in SDA2 by checking whether `-Xsmall-const` is used to compile the files where they are defined.

For example, if you compile a source file with `-Xsmall-const=8`, which accesses a small variable defined in a library which has been compiled with `-Xsmall-const=0`, it may fail during the linking process. If such error occurs, you will have to **relinquish** `-Xsmall-const` or use `-Xsmall-const=0` explicitly for current source file, as a result far-absolute mode is used to access them.

-Xsmall-const-registers

Extend small constant areas.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xsmall-const-registers=m
-X455=m
```

Description

For information about both **-Xsmall-const-registers** and **-Xsmall-data-registers**, see **-Xsmall-data-registers**.

See Also

[-Xsmall-data-registers on page](#)

-Xsmall-data

Set size limit for "small data" variables.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xsmall-data=n
-X97=n
```

Description

Use register relative addressing mode (r13 relative for PPC) to access all small **non-constant** static and global variables with a size in bytes less than or equal to *n*. For small **non-constant** variables defined in current compilation unit, place them in the **SDATA** section class.

For more information, see the discussion of the **section** pragma in the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

For TriCore, **-Xsmall-data** works only with the TriCore 32-bit relative addressing mode (for example, **-tTC12MS**). It should not be used with the 18-bit absolute addressing mode (for example, **-tTC12LS**).

Note

A base register (r13 for PPC) points to the Small Data Area (SDA), and register relative addressing mode can be used to access the limited space (a maximum size of 64K for PPC). For **non-constant** variables in SDA, it will be always successful to use the register relative addressing mode to access them. However, for **non-constant** variables in non-SDA, it may fail during the linking process. It all depends on whether they are within the address space relative to the base register.

Therefore, for small **non-constant** variables that are accessed but not defined in current compilation unit, make sure they are placed in SDA by checking whether **-Xsmall-data** is used to compile to compile the files where they are defined.

For example, if you compile a source file with **-Xsmall-data=8**, which accesses a small variable defined in a library which has been compiled with **-Xsmall-data=0**, it may fail during the linking process. If such error occurs, you will have to relinquish **-Xsmall-data** or use **-Xsmall-data=0** explicitly for current source file, as a result far-absolute mode is used to access them.

-Xsmall-data-registers

Extend small data areas.

Belongs to:

[Sections Index on page 267](#)

Synopsis

```
-Xsmall-data-registers=n
-X454=n
```

Description

Extend small data and small constant areas to more than one register on targets that support this feature, using **-Xsmall-data-registers** and **-Xsmall-const-registers**. For information about small data and constant areas, see the *Locating Code and Data, Addressing, Access* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

Targets that support small data and small constant areas have a register that is reserved by the ABI to support referencing small variables and small constants using base-register-plus-offset addressing modes. This can provide greater efficiency and smaller code size. In some cases, these areas may not be big enough to hold all of the application's data, which requires the user to take some sort of action to resolve the problem.

One solution is to manually place some variables in different sections, and to reserve a register globally across the application, thus reducing the number of variables in the small areas to fit the size limitations. (For information about this process, see the discussion of the **SECTIONS** command in the *Linker Command Language* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.)

Another solution is to use the **-Xsmall-data-registers=n** and **-Xsmall-const-registers=m** options tell the compiler and the rest of the toolchain to automatically extend the number of registers reserved to **n** and **m**, respectively (where **n** and **m** should be greater than 1).

For example, at link time the small constant area may overflow, and the user may decide to use two registers to fix the problem by specifying **-Xsmall-const-registers=2**. Once the entire application is recompiled with this option, the linker can link the application correctly and safely. Note, however, that by reserving more registers, fewer variables are free for use and this could have a negative impact on performance and size.

The use of these options changes the effective ABI of the target. Because this happens automatically (and by intention to simplify the process for the user), the user does not control which variables are referenced by which registers. Any variable placed in the small data and small constant areas may be accessed through the original ABI register pair, or by other registers, as part of an implied special ABI extension.

To make sure that the registers that are re-purposed as extended small area registers are not arbitrarily used for multiple purposes by the compiler, you must build the entire application—including library code— with the same **-Xsmall-data-registers=n** and **-Xsmall-const-registers=m** settings. The linker generates warnings (by default) when the ABIs of the object files are inconsistent and ignoring these warnings may result in applications misbehaving during run-time. Note that the **-Xdisable-all-warnings** causes the linker to ignore warnings about this potentially serious ABI incoherence.

To understand why the same **-Xsmall-data-registers=n** and **-Xsmall-const-registers=m** settings must be used for all of the application code (including libraries), consider the case in which two small data area registers are used: the reserved register **Rs** and an extension register **Rx**, which is normally free and used for other purposes, for some object files. Also assume that other object files are built using one small area register, and that the variable **V** goes into the small data area and is used by all of the object files. If **V** is referenced through **Rs**, the application may appear to be working correctly. However, if **V** is referenced through **Rx**, the object files that reserve **Rx** are able to access **V**, but the other object files do not reserve **Rx**, and are free to overwrite it and use it for other purposes. The code in these object files can change the value of **Rx** freely. The linker also uses **Rx** freely to access **V** (and any other variables in the extended small data area).

See also **-Xignore-extensible-sda-warnings** in the *Linker User's Guide*.

-Xstack-delay, -Xstack-delay-off

Delay popping stack after function call.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xstack-delay=n
-X127=n
```

Description

-Xstack-delay-off
-X127=1

When a function calls another function, do not pop the arguments pushed on the stack for the called function after it returns until **n** bytes have been used for such arguments. The default value of **n** is 32,767.

For example, if function **f** calls only function **g** requiring that 8 argument bytes be pushed, and then function **h** requiring that 4 more argument bytes be pushed, and if **-Xstack-delay** is greater than 12, do not remove the argument bytes from the stack on return from **g** or **h**. Clean off the stack only when **f** itself returns.

This option produces more efficient code but note that because the default is 32,767, a function calling many other functions with many arguments may require a larger stack than otherwise.

Setting **-Xstack-delay-off**, or **-Xstack-delay=0**, ensures that no extra stack space is used by this optimization.

-Xstack-probe

Enable stack checking.

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xstack-probe
-X10
```

Description

Enable stack checking (probing).

Note

-Xstack-probe cannot be used with "interrupt" functions, that is, with a function named in an **interrupt** pragma or declared using the **interrupt** or **interrupt** keywords.

-Xstack-protection

Protects functions that have local arrays or that use alloca

Synopsis

```
-Xstack-protection
```

Description

Protects functions by "Stack Smashing Protection" that have local char arrays (possibly inside structs), or that use alloca

Stack smashing protection is a secure feature of the Diab compiler that, when enabled, attempts to detect stack-based buffer overflow attacks at run time. If an attack is detected, a user defined fatal error hook is called.

See Also

[-Xstack-protection-strong on page](#)
[-Xstack-protection-all on page](#)
[-Xstack-protection-off on page](#)

-Xstack-protection-all

Protects all functions

Synopsis

```
-Xstack-protection-all
```

Description

Protects all functions by "Stack Smashing Protection"

Stack smashing protection is a secure feature of the Diab compiler that, when enabled, attempts to detect stack-based buffer overflow attacks at run time. If an attack is detected, a user defined fatal error hook is called.

See Also

[-Xstack-protection-strong on page](#)
[-Xstack-protection on page](#)
[-Xstack-protection-off on page](#)

-Xstack-protection-off

Turns stack protection off

Synopsis

```
-Xstack-protection-off
```

Description

Turns stack protection off - this is the default

Stack smashing protection is a secure feature of the Diab compiler that, when enabled, attempts to detect stack-based buffer overflow attacks at run time. If an attack is detected, a user defined fatal error hook is called.

See Also

[-Xstack-protection-strong on page](#)
[-Xstack-protection-all on page](#)
[-Xstack-protection on page](#)

-Xstack-protection-strong

Protects functions that have local arrays of any type, or local variables that have their address taken, or that use alloca

Synopsis

```
-Xstack-protection-strong
```

Description

Protects functions by "Stack Smashing Protection" that have local char arrays (possibly inside structs), or local variables that have their address taken, or that use alloca

Stack smashing protection is a secure feature of the Diab compiler that, when enabled, attempts to detect stack-based buffer overflow attacks at run time. If an attack is detected, a user defined fatal error hook is called.

See Also

[-Xstack-protection on page](#)
[-Xstack-protection-all on page](#)
[-Xstack-protection-off on page](#)

-Xstack-protection-verbose

see which functions are protected by SSP

Synopsis

```
-Xstack-protection-verbose
```

Description

If you want to see which functions are protected by "Stack Smashing Protection" set this option

This command will provide output like the following:

info: SSP enabled on '<func>' because <reason>

For example:

info: SSP enabled on 'foo' because has char array

Those options are currently available exclusively for ARM, PPC, RH850 and for TriCore targets.

See Also

[-Xstack-protection-all on page](#)
[-Xstack-protection-strong on page](#)
[-Xstack-protection on page](#)
[-Xstack-protection-off on page](#)

-Xstatic-addr-error

Diagnose static initialization using address.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xstatic-addr-error
-X81=2
```

Description

Generate an error if the address of a variable, function, or string is used by a static initializer. This is useful when generating position-independent code (PIC).

See Also

[-Xstatic-addr-warning on page](#)

-Xstatic-addr-warning

Diagnose static initialization using address.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
---Xstatic-addr-warning
-X81=1
```

Description

Generate a warning if the address of a variable, function, or string is used by a static initializer. This is useful when generating position-independent code (PIC).

See Also

[-Xstatic-addr-error on page](#)

-Xstderr-fully-buffered

Buffer stderr.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xstderr-fully-buffered
-X173
```

Description

Buffer **stderr** using 10KB buffer. Use this option to reduce network traffic; **stderr** is unbuffered by default.

-Xstop-on-warning

Terminate compilation on warning.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xstop-on-warning
-X85
```

Description

Terminate compilation on any warning. Without this option, only errors terminate compilation. (For both errors and warnings, compilation terminates after a small number of errors are output.)

Note

This option works exclusively with legacy compiler (ctoa) but not with EDG (etoa).
See also chapter "C Dialect Default Compilation Mode" of your User's Guide for your architecture!

-Xstrict-ansi

Compile C/C++ in pedantic mode.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xstrict-ansi
-X7=2
```

Description

Compile in "pedantic" mode. This option is equivalent to **-Xdialect-strict-ansi**. For C, see **-Xdialect-....**. For C++, **-Xstrict-ansi** generates diagnostic messages when nonstandard features are used and disables features that conflict with ANSI/ISO C++, including **-Xusing-std-on** and **-Xdollar-in-ident**.

Disabled by default.

See Also

[-Xdialect-... on page](#)

-Xstrict-bitfield-promotions

Ignore sign when promoting bit-fields.

Belongs to:

[Type Index on page 269](#)

Synopsis

```
-Xstrict-bitfield-promotions
-x199
```

Description

Conform to the ANSI standard when promoting bit-fields. When a bit-field occurs in an expression where an **int** is expected, the compiler promotes the bit-field to a larger integral type. Unless this option is enabled, such promotions preserve sign as well as value. If **-Xstrict-bitfield-promotions** is specified, however, an object of an integral type all of whose values are representable by an **int** (that is, an object smaller than 4 bytes) is promoted to an **int**, even if the original type is unsigned.

-Xstrict-ansi or **-Xdialect-strict-ansi** implicitly enables **-Xstrict-bitfield-promotions** by default, but can be overridden with **-Xstrict-bitfield-promotions=0**.

See Also

[-Xbit-fields-signed](#) on page

-Xstrict-eabi

Disable non-EABI conversion functions.

Belongs to:

[Instruction Index](#) on page 252

Synopsis

```
-Xstrict-eabi
-X53
```

Description

With this option enabled, the compiler will generate only those function calls for conversion between floating point and integer as are specified in the PowerPC EABI.

With this option disabled (**absent**, or **-X53=0**), the compiler will generate calls to additional conversion functions not in the EABI whenever doing so will improve code size and speed. This is the default.

-Xstring-align

Align strings on n-byte boundaries.

Belongs to:

[Memory Index](#) on page 255

Synopsis

```
-Xstring-align=n
-X18=n
```

Description

Align each string on an address boundary divisible by **n**. The default value is 4.

See Also

[-Xarray-align-min](#) on page

-Xstruct-arg-warning

Warn on large structure.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xstruct-arg-warning=n
-X92=n
```

Description

C only. Emit a warning if the size of a structure argument is larger than or equal to **n** bytes.

-Xstruct-as-arg

Select convention for returning structures and unions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xstruct-as-arg
-X80
```

Description

Rewrite functions returning C structures and unions as if the first argument is a pointer to the return area. Example:

```
struct a { int i, j; }
struct a fna(int i) {
  struct a ret;
  ret.i = i;
  return ret;
}
m() {
  struct a z = fna(1);
}
is effectively rewritten as:
void fna(struct a *retval, inti) {
  struct a ret;
  ret.i = i;
  *retval = ret;
}
m() {
  struct a z;
  fna(&z, 1);
}
```

For more information, see the discussion of class, structure, and union return types in the *Wind River Diab Compiler User's Guide* for your architecture.

ColdFire also has the related **-Xstruct-as-gnu** option.

See Also

[-Xstruct-as-static on page](#)

-Xstruct-assign-split-...

Control optimization of structure member assignments.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xstruct-assign-split-diff=n
-X147=n
```

Description

```
-Xstruct-assign-split-max=n
-X146=n
```

These options control optimization of assignments of local **struct** variables. The compiler uses a number of techniques to optimize structure members (it uses registers, etc.). A structure can be assigned as one or more blocks (depending on a number of factors) or member-by-member. However, block structure assignment disables member optimization, so options are available to control the type of structures that will be assigned as a block.

By default, the assignment is member-by-member if the structure has 6 or fewer members and if the increase in assignments (over block assignments) is 3 or fewer. Otherwise, the structure is assigned as a block.

Use **-Xstruct-assign-split-max** to set the maximum number of members in a **struct** that may be assigned member-by-member.

Use **-Xstruct-assign-split-diff** to set the maximum number of additional assignments allowed. If member-to-member assignment involves a higher number of additional assignments than the number set by **-Xstruct-assign-split-diff**, a block assignment is performed.

-Xstruct-as-static

Select convention for returning structures and unions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xstruct-as-static
-X80=2
```

Description

Cause functions returning C structures and unions to allocate static memory for the structure or union to be returned, and return a pointer to this area. This method is not reentrant.

Note

C++ classes are always returned using **-Xstruct-as-arg**.

For more information, see the discussion of class, structure, and union return types in the *Wind River Diab Compiler User's Guide* for your architecture.

ColdFire also has the related **-Xstruct-as-gnu** option.

See Also

[-Xstruct-as-arg on page](#)

-Xstruct-best-align

Align data on “natural” boundaries.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xstruct-best-align
-X17
```

Description

Align data on “natural” boundaries, e.g., 4 byte boundaries for 4 byte **int** data.

Default:

- ELF objects: **-Xstruct-best-align**
- COFF objects: **-Xstruct-best-align=0** (off)

-Xstruct-min-align

Set minimum structure member alignment.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xstruct-min-align=n
-X76=n
```

Description

Force structures to begin on at least an **n** byte boundary. If any member in a structure has a greater alignment, the structure will be aligned on a boundary divisible by the size in bytes of the largest member. Defaults to 0.

See the discussion of the **pack** pragma and the **__packed__** and **packed** keywords in the *Additions to C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

Note that this option also affects the alignment of unions.

See Also

[-Xmember-max-align on page](#)

-Xsuppress-warnings

Suppress warnings.

Belongs to:

[Diagnostic and Lint Index on page 251](#)

Synopsis

```
-Xsuppress-warnings
-X14
```

Description

Suppress compiler warnings.

-Xswap-cr-nl

Swap '\n' and '\r' in constants.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xswap-cr-nl
-X13
```

Description

C only. Swap '\n' and '\r' in character and string constants. Used on systems where carriage return and line feed are reversed.

-Xswitch-array-in-function-section

Put any compiler-generated switch tables in the same section as the parent function.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xswitch-array-in-function-section
-X472
```

Description

This option puts any compiler-generated **switch** arrays in the same section as the parent function. The option is meant to follow the behaviour of specifying the section of CODE/CONST sections on a per-function basis using #pragma section; Without this option, the switch table will be generated in the default CONST section, but with this option, a switch table will be generated in the same CONST section as specified for that function. If -Xsection-split is also specified, the switch table will be put in the split section as well.

Note

This option replaces the deprecated option -Xswitch-table-in-function-section

-Xswitch-array-off

Disables conversion of switch to look-up array

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xswitch-array-off
```

Description

Disables the following optimization (enabled by default):

Switch statements with a dense set of case values, in which every case just assigns to a variable, can sometimes be converted to use a look-up array instead of a switch:

```
switch(a) {
case 1: x = 10; break;
case 2: x = 5; break;
case 3: x = 8; break;
...
}
==>
\\ (range check not shown)
x = __switch_array1(a)
```

It is equivalent to **-Xkill-opt=0x20**.

-Xswitch-table

Set threshold for a switch statement table.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xswitch-table=n
-X143=n
```

Description

Implement a **switch** statement using compares if there are fewer than **ncase** labels in the **switch**, otherwise the compiler will decide to use a jump table or not based on an internal heuristic. This option is on by default with a value of 7. This option puts any compiler-generated switch tables in the same section as the parent function. If **-Xsection-split** is also specified, the switch table will be put in the split section as well.

-Xswitch-table-in-function-section

Put any compiler-generated switch tables in the same section as the parent function.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xswitch-table-in-function-section
-X472
```

Description

This option puts any compiler-generated **switch** tables in the same section as the parent function. The option is meant to follow the behaviour of specifying the section of CODE/CONST sections on a per-function basis using **#pragma section**; Without this

option, the switch table will be generated in the default CONST section, but with this option, a switch table will be generated in the same CONST section as specified for that function. If `-Xsection-split` is also specified, the switch table will be put in the split section as well.

Note

This function is deprecated. Please use equivalent option **`-Xswitch-array-in-function-section`**.

`-Xswitch-table-off`

Disable jump tables for switch statements.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xswitch-table-off
```

Description

Do not use a jump table to implement a **`switch`** statement under any conditions.

`-Xsystem-headers-warn`

Disable certain system header warnings.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xsystem-headers-warn=n
-X410=n
```

Description

When **`n`** = 0, inhibit certain warnings when processing system headers (headers found in directories specified with **`-YI`**).

When **`n`** = 1, treat system headers just like user headers.

`-Xwarn-system-headers` is a synonym for this option.

`-Xtest-at-both`

Loop tests at top and bottom.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xtest-at-both
-X6=2
```

Description

Force the compiler to always test loops both before the loop is started and at the bottom of the loop. This option produces the fastest possible code but uses more space. Even if **-Xtest-at-both** is not set, other optimizations may cause the compiler to generate double tests. This option is turned on by **-XO**.

See Also

[-Xtest-at-bottom on page](#)
[-Xtest-at-top on page](#)

-Xtest-at-bottom

Loop test at bottom.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xtest-at-bottom
-X6=0
```

Description

Use one loop test at the bottom of a loop.

Note

Despite the names of these options, in many cases **-Xtest-at-top** and **-Xtest-at-bottom** will produce the same code. **-Xtest-at-bottom** uses internal heuristics to determine the best code.

See Also

[-Xtest-at-both on page](#)
[-Xtest-at-top on page](#)

-Xtest-at-top

Loop test at top.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xtest-at-top
-X6=1
```

Description

Use one loop test at the top of a loop.

See Also

[-Xtest-at-both on page](#)
[-Xtest-at-bottom on page](#)

-Xtruncate

Truncate all identifiers after *m* characters.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xtruncate=m
-X22=m
```

Description

Truncate all identifiers after **m** characters. If **m** is zero, no truncation is done. This is the default.

-Xunderscore-leading

Prefix identifiers with underscore.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xunderscore-leading
-X71=1
```

Description

Prefix every externally visible identifier with an underscore in the symbol table.

Synonym: **-Xleading-underscore**. For ColdFire this is the default.

Note

The **-Xunderscore...** options are provided for use in linking code generated by the compiler with third-party libraries or with other tools requiring generated underscores. The default values for this option are as follows:

- ARM, MIPS, x86: 0 (no extra underscore).
- ColdFire, RH850: **-Xunderscore-leading**.
- PowerPC, TriCore: Dictated by the Embedded Application Binary Interface (EABI) standard and is 0 (no extra underscore).

Because Wind River libraries are compiled with the default setting, setting this option to anything but the default will require recompiling every library used.

See Also

[-Xunderscore-surround on page](#)
[-Xunderscore-trailing on page](#)

-Xunderscore-surround

Add underscores to identifier prefix and suffix.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xunderscore-surround
-X71=3
```

Description

Prefix and suffix every externally visible identifier with an underscore in the symbol table.

Synonym: **-Xsurround-underscore.**

Note

The **-Xunderscore...** options are provided for use in linking code generated by the compiler with third-party libraries or with other tools requiring generated underscores. The default values for this option are as follows:

- ARM, MIPS, RH850, x86: 0 (no extra underscore).
- ColdFire: **-Xunderscore-leading.**
- PowerPC, TriCore: Dictated by the Embedded Application Binary Interface (EABI) standard and is 0 (no extra underscore).

Because Wind River libraries are compiled with the default setting, setting this option to anything but the default will require recompiling every library used.

See Also

[-Xunderscore-leading on page](#)

[-Xunderscore-trailing on page](#)

-Xunderscore-trailing

Add underscore to identifier suffix.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xunderscore-trailing
-X71=2
```

Description

Suffix every externally visible identifier with an underscore in the symbol table.

Synonym: **-Xtrailing-underscore.**

Note

The **-Xunderscore...** options are provided for use in linking code generated by the compiler with third-party libraries or with other tools requiring generated underscores. The default values for this option are as follows:

- ARM, MIPS, RH850, x86: 0 (no extra underscore).
- ColdFire: **-Xunderscore-leading**.
- PowerPC, TriCore: Dictated by the Embedded Application Binary Interface (EABI) standard and is 0 (no extra underscore).

Because Wind River libraries are compiled with the default setting, setting this option to anything but the default will require recompiling every library used.

See Also

[-Xunderscore-leading on page](#)
[-Xunderscore-surround on page](#)

-Xunroll

Set loop unrolling count.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xunroll=n
-X15=n
```

Description

Unroll small loops **n** times. Set to 2 by default. **n** must be a power of two. Some sufficiently small loops may be unrolled more than **n** times if total code size and speed is better.

See the discussion of loop unrolling in the *Optimization* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xlegacy-loop-unrolling

Disables unrolling loops with 'if' statements. (The optimizer currently does this by default.)

Note

Unrolling is done only if option **-O** or **-XO** is given to enable optimization.

-Xunroll-register-rename

Allow scalar variable to occupy multiple registers for unrolled loops.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xunroll-register-rename
```

Description

For unrolled loops, allows the same scalar variable on multiple iterations to occupy multiple registers when register pressure is not a concern. This may improve performance on architectures that do not support register renaming in hardware.

-Xunroll-size

Set loop unrolling size.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xunroll-size=n
-X16=n
```

Description

Specify the maximum number of nodes a loop can contain to be considered for loop unrolling. Each operator and each operand counts as one node, so the expression

`a = b - c;`

contains 5 nodes. (There is also a small number of additional nodes for each function.) **n** is set to 20 by default. Assembly files saved with **-S** show the number of nodes for each function.

Note

Unrolling is done only if option **--O** or **-XO** is given to enable optimization.

-Xuser-metadata

Add key-value pairs to the metadata of an object file.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xuser-metadata="key1=value1,key2=value2,..."
```

Description

This switch applies to both the compiler and assembler. It adds a comma separated list of key-value pairs to the metadata of the output object file. If the same key is added with both the compiler and the assembler, the value assigned with the assembler take precedence.

The metadata can be inspected using the linker (see **-Xdump-metadata** in the *Linker User's Guide*) or **ddump** (see the *Wind River Diab Compiler Utilities Reference: D-DUMP File Dumper*).

-Xvec

(MIPS64GLRN only) Enable auto-vectorization.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xvec
```

Description

This is the main option to enable vectorization. It is only effective if optimization is also enabled (**-XO** or **-O**).

Since

5.9.5.0

See Also

[-O on page](#)
[-XO on page](#)
[-Xvec-aligned on page](#)
[-Xvec-verbose on page](#)
[-Xfp-fast on page](#)
[-Xargs-not-aliased on page](#)
[-Xclib-optim-off on page](#)

-Xvec-aligned

(MIPS64GLRN only) Inform compiler that arrays are properly aligned for vectorization.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xvec-aligned
```

Description

This option tells the compiler to assume that all the starting array references in a loop that may be vectorized, will be aligned to the boundary required by the vector memory operations. For MIPS64GLRN, this is currently 16-byte alignment. This allows the vectorizer to go ahead and vectorize loops with references for which it cannot determine the alignment. Without this flag, the vectorizer will not vectorize loops with unknown alignment. This is usually indicated with a diagnostic message such as: "possibly unaligned data access ..."

Note

-Xvec-aligned will assume **-Xvec-args-aligned**.

Since

5.9.5.0

See Also

[-Xvec on page](#)
[-Xvec-args-aligned on page](#)

-Xvec-args-aligned

(MIPS64GLRN only) Assume that function array/pointer arguments are aligned for vector memory access.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xvec-args-aligned
```

Description

This option tells the compiler to assume that all function array or pointer arguments will be aligned to the boundary required by the vector memory operations. For MIPS64GLRN, this is currently 16-byte alignment.

Since

5.9.5.0

See Also

[-Xvec on page](#)
[-Xvec-aligned on page](#)

-Xvec-verbose

(MIPS64GLRN only) Control diagnostic output for auto-vectorization.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xvec-verbose=n
```

Description

This option controls the printing of vectorization diagnostic messages to standard output depending on the value of **n**.

- With **n** being 0, no messages are printed.
- With **n** being 1, basic information on the loops vectorized, and reasons for those not vectorized are printed. This is the default.

Since

5.9.5.0

See Also

[-Xvec on page](#)

-Xvoid-ptr-arith-ok

Void pointer arithmetic.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xvoid-ptr-arith-ok  
-Xl67
```

Description

Treat void pointers as **char *** for the purpose of arithmetic. For example:

```
some_void_ptr += 1; /* adds 1 to some_void_ptr */
```

-Xwchar

Define type for wchar.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xwchar=n
-X86=n
```

Description

Define the type to which **wchar** will correspond. The desired **type** is given by specifying a value **n** equal to a value returned by the operator **sizeof(type, 2)**.

The default type is **unsigned short** integer, that is, **-Xwchar=3**.

See the discussion of the **sizeof** operator extension in the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

See Also

[-Xwchar_t-off on page](#)

[-Xwchar_t-on on page](#)

-Xwhole-program-diagnostics

Set whole-program optimization diagnostics.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xwhole-program-diagnostics=mask
-X426
```

Description

Print out whole-program optimization diagnostics according to the value **mask**. For more on whole-program optimization (WPO), see **-Xwhole-program-optim** and the discussion of whole-program optimization in the *Optimization* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

mask may be given in hex, and mask bits may be OR-ed to select more than one, e.g., **-Xwhole-program-diagnostics=0x102**. Undefined mask bits are ignored.

Words in brackets indicate which component issues the diagnostics.

0x1

[linker] Print details of commands invoked by the linker. Very verbose.

0x2

[linker] Print brief details of commands invoked by the linker. This is the recommended output level for most users.

0x100

[compiler] Print the name of each module that is recompiled.

0x200

[compiler] Print the name of each function that is recompiled.

0x400

[compiler] Show which functions are inlined from other modules. If a function is inlined in more than one place, only the first instance is shown.

See Also

[-Xwhole-program-optim on page](#)

-Xwhole-program-optim

Use whole-program optimization.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xwhole-program-optim=mask
```

Description

Perform whole-program optimization (WPO). The following options are enabled by default when **-Xwhole-program-optim** is used:

- **-Xwpo-inline**
- **-Xwpo-const**
- **-Xwpo-no-alias**
- **-Xwpo-delete-return-value**
- **-Xwpo-dead**

The **-Xwpo-inline-single-calls** option (which is not enabled by default) is also provided.

For more information, see **-Xwpo-inline**, **-Xwpo-inline-single-calls**, **-Xwpo-const**, **-Xwpo-no-alias**, **-Xwpo-delete-return-value**, **-Xwpo-dead**, and the *Optimization* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

The **-Xwhole-program-optim** option's **mask** may have the following values and be given in hex, and mask bits may be OR-ed to select more than one, e.g., **-Xwhole-program-optim=0x9**. Undefined mask bits are ignored.

0x0

Do not perform whole-program optimization. The default if WPO is not invoked.

0x1

Save module information for use in link-time optimization. (You will always want this bit set if you are using WPO.) This is the default level for WPO. Equivalent to specifying **-Xwhole-program-optim**.

0x2

Optimize this module at compile time. (Normally, **-Xwhole-program-optim** disables compile-time optimization, since it is assumed that all the optimization will take place at link time.) Useful for building libraries.

0x4

Import-only. Inline functions from other modules but do not export functions from this module.

0x8

Enable link-time object-file cache. At link time, allow the linker to reuse object files that have been optimized in previous links.

Wind River recommends using this mask for development builds but not for production builds. It can dramatically reduce the time required to re-link after an initial linkage—which is useful for development builds. However, if you make changes to the code and recompile using this mask, the result may not be the same as if you had built the code from clean—which is not desirable for production builds.

If using this feature, be sure to specify a directory with **-Wl, -Xwhole-program-output-dir**.

0x10

Export-only. Allow other modules to inline functions from this module at link time, but do not recompile this module. Useful for building libraries.

0x80

Remove all temporary files, even if **-#** is specified. By design, the combination of the **-Xwhole-program-optim** with **-#** retains one temporary file, which allows you to rerun the printed compile line with exactly the same results. See also, **-#, -##, -###**.

0x100

Enable link time SDA optimization. For information about SDA optimization, see the Optimization chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xwhole-program-optim may be given as an option to either the compiler or the linker.

Note

The **-Xwhole-program-optim** option will increase compile-time memory usage, depending on the size and complexity of the source program.

Using -Xwhole-program-optim with -Xkeep-assembly-file

The assembly files created by **-Xkeep-assembly-file** are created in the same directory as the corresponding object file. Programs compiled with whole program optimization create two object files, an unoptimized file at compile time and an optimized file at link time. When used with **-Xwhole-program-optim**, the **-Xkeep-assembly-file** option creates two assembly files. For example, the following:

`dcc -tPPCFH:windiss hello.c -Wl,-Xwhole-program-output-dir=mydir -Xwhole-program-optim -XO -Xkeep-assembly-file`
creates the following at compile time:

```
./hello.o
```

```
./hello.s
```

and the following at link time:

```
mydir/hello.o
```

```
mydir/hello.s
```

By default **-Xwhole-program-optim** will recompile files at link time sequentially. However if **-Xwhole-program-jobs=n** is specified, the linker will recompile n files at a time. This can speed up the overall compile time on machines with multiple processors. (See **-Xwhole-program-jobs** in the *Linker User's Guide*.

See also **-Xwhole-program-output-dir** in the *Linker User's Guide*.)

See Also

- [-Xwpo-inline on page](#)
- [-Xwpo-inline-single-calls on page](#)
- [-Xwpo-const on page](#)
- [-Xwpo-no-alias on page](#)
- [-Xwpo-delete-return-value on page](#)

[-Xwpo-dead on page](#)
[-### on page](#)
[-Xkeep-assembly-file on page](#)

-Xwpo-const

Enable whole program auto-constant optimization.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xwpo-const
```

Description

This optimization detects global variables that have an initializer, but are never subsequently written. Such variables are treated as constants, and references to them are replaced by the initial value. This option is enabled by default when **-Xwhole-program-optim** is used.

For example:

```
int x = 5;
int main()
{
  return x;
}
```

If **x** is never written anywhere in the program, then the code is optimized to:

```
int main()
{
  return 5;
}
```

-Xno-wpo-const

Disable whole program auto-constant optimization.

See Also

[-Xwhole-program-optim on page](#)
[-Xwpo-inline on page](#)
[-Xwpo-inline-single-calls on page](#)
[-Xwpo-no-alias on page](#)
[-Xwpo-delete-return-value on page](#)
[-Xwpo-dead on page](#)

-Xwpo-dead

Enable whole program dead code and data elimination.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xwpo-dead
```

Description

This optimization removes (non-volatile) variables and functions that are not referenced in the program. This option is enabled by default when **-Xwhole-program-optim** is used.

Note that if you compile with **-Xsection-split** and link with **-Xremove-unused-sections**, then this optimization does not provide any additional reduction in code or data size. Also note that this optimization is not as powerful for dead code and data elimination as **-Xremove-unused-sections**, so it should not be regarded as a replacement for that option.

The primary benefit of this optimization is to reduce compile time by not (unnecessarily) optimizing and generating code for unused functions and variables.

See Also

[-Xwhole-program-optim on page](#)
[-Xwpo-inline on page](#)
[-Xwpo-inline-single-calls on page](#)
[-Xwpo-const on page](#)
[-Xwpo-no-alias on page](#)
[-Xwpo-delete-return-value on page](#)

-Xwpo-delete-return-value

Enable whole program return value value deletion optimization.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xwpo-delete-return-value
```

Description

This optimization detects global functions whose return values are never used, and optimizes the functions by removing computations leading to the return statements. This option is enabled by default when **-Xwhole-program-optim** is used.

For example:

```
int f(void)
{
  ++A;
  return B + 5;
}
```

If **f** is only called so that its return value is always discarded, like this:

```
f();
then it can be optimized to:
int f(void)
{
  ++A;
}
```

-Xno-wpo-delete-return-value

Disable whole program return value value deletion optimization.

See Also

[-Xwhole-program-optim on page](#)
[-Xwpo-inline on page](#)
[-Xwpo-inline-single-calls on page](#)

[-Xwpo-const on page](#)
[-Xwpo-no-alias on page](#)
[-Xwpo-dead on page](#)

-Xwpo-inline

Enable cross-module inlining.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xwpo-inline
```

Description

This optimization option allows calls to functions that are defined in different modules to be inlined. This option is enabled by default when **-Xwhole-program-optim** is used.

Consider, for example:

```
/* t1.c */
int f(void)
{
    return 4;
}
```

and

```
/* t2.c */
int main()
{
    return f();
}
```

Normally the compiler would not be able to inline the call to **f()** in **t2.c** because **f()** is defined in a different module (**t1.c**). However when **-Xwhole-program-optim** is enabled (which enables **-Xwpo-inline** by default), such calls are inlined. In the example above, the compiler would be able to optimize **t2.c** to:

```
/* t2.c */
int main()
{
    return 4;
}
```

Note that cross-module inlining may lead to an increase in code size. The **-Xinline** option can be used to control the level of inlining. In addition, you can prevent inlining of specific functions by declaring them with **__attribute__((noinline))**.

Disabling this optimization altogether only disables cross-module inlining. Inlining within a module is still performed.

-Xno-wpo-inline

disable cross-module inlining.

See Also

[-Xwhole-program-optim on page](#)
[-Xwpo-inline-single-calls on page](#)
[-Xwpo-const on page](#)
[-Xwpo-no-alias on page](#)
[-Xwpo-delete-return-value on page](#)
[-Xwpo-dead on page](#)

-Xwpo-inline-single-calls

Always inline functions that are only called once (regardless of size).

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xwpo-inline-single-calls
```

Description

Functions that are only called once across the entire program are treated as if they were marked with **`__attribute__((always_inline))`**, and, if possible, inlined regardless of their size, or the value of **-Xinline**.

This option is disabled by default.

See Also

[-Xwhole-program-optim on page](#)
[-Xwpo-inline on page](#)
[-Xwpo-const on page](#)
[-Xwpo-no-alias on page](#)
[-Xwpo-delete-return-value on page](#)
[-Xwpo-dead on page](#)

-Xwpo-no-alias

Enable whole program unaliased variable optimization.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xwpo-no-alias
```

Description

This option detects global variables that never have their address taken. This information can be used by the rest of the optimizer to remove some redundant loads and stores. This option is enabled by default when **-Xwhole-program-optim** is used.

For example:

```
int A;
int f(int *p)
{
  int x = A;
  *p = 2;
  x -= A;
  return x;
}
```

Normally the compiler would conservatively assume that the store ***p = 2** might actually be writing to **A** (**p** might be an alias for **&A**). However if whole program analysis detects that **A** never has its address taken, then the compiler can optimize this code to:

```
int f(int *p)
{
  *p = 2;
```

```
return 0;
}
```

-Xno-wpo-no-alias

Disable whole program unaliased variable optimization.

See Also

[-Xwhole-program-optim on page](#)
[-Xwpo-inline on page](#)
[-Xwpo-inline-single-calls on page](#)
[-Xwpo-const on page](#)
[-Xwpo-delete-return-value on page](#)
[-Xwpo-dead on page](#)

1.4. Compiler Driver X Options

Table of Contents

- [-Xc++-abr on page](#)
- [-Xc-new on page](#)
- [-Xc-old on page](#)
- [-Xforeign-as-ld on page](#)
- [-Xkeep-assembly-file on page](#)
- [-Xkeep-object-file on page](#)
- [-Xlibc-... on page](#)
- [-Xpreprocess-assembly on page](#)
- [-Xshow-target on page](#)

-Xc++-abr

Use abridged C++ libraries.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xc++-abr
```

Description

Link to the abridged C++ libraries. Automatically disables exception-handling (**-Xexceptions=off**).

For more information, see the discussion of C++ standard libraries in the *C++ Features and Compatibility* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xc-new

Use the 'new' compiler frontend.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xc-new
```

Description

Compile using a compiler frontend derived from one produced by the Edison Design Group. By default, invoking **-Xc-new** also invokes **-Xdialect-c99**.

VxWorks kernel-mode programming does not support **-Xdialect-c99**. To use **-Xc-new** for kernel mode, you must also set **-Xdialect-c89**.

See Also

[-Xlibc-... on page](#)

[-Xdialect-... on page](#)

-Xc-old

Use legacy DIAB front end (ctoa).

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xc-old
```

Description

Compile using the legacy DIAB compiler front end **ctoa**. The **ctoa** front end only supports C89.

-Xforeign-as-ld

Suppress assembler and linker parameters.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xforeign-as-ld
(no numeric equivalent)
```

Description

Cause the driver to call an assembler and linker without any implicit parameters.

This allows third-party assemblers and linkers to be used with the Wind River Diab Compiler. The **-W xfile** option may be used to specify a foreign assembler or linker, the **-W a** option to pass parameters to the assembler, and the **-W l** option to pass parameters to the linker.

See Also

[-W x on page](#)

[-W a,args on page](#)

[-W l,args](#)

-Xkeep-assembly-file

Create and keep assembly file.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xkeep-assembly-file
-Xkeep-assembly-file=2
(no numeric equivalent)
```

Description

Always create and keep a **.s** file without the need for a separate compilation with the **-S** option. This option can be used with the **-c** option to create both assembly and object files at once.

With **-Xkeep-assembly-file**, the name of the **.s** file is the same as the name of the source file. For example, the following command:

```
dcc -c -tPPCFH:windiss input.c -o output.o -Xkeep-assembly-file
```

will create an assembly file named `input.s`.

With **-Xkeep-assembly-file=2**, the name of the **.s** file is the same as the name of the object file. For example, the following command:

```
dcc -c -tPPCFH:windiss input.c -o output.o -Xkeep-assembly-file=2
```

will create an assembly file named `output.s`.

See Also

[-Xkeep-object-file on page](#)

-Xkeep-object-file

Create and keep object file.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xkeep-object-file
(no numeric equivalent)
```

Description

Always create and keep a **.o** file without the need for a separate compilation with the **-c** option. This is needed only when a single file is compiled, assembled, and linked in one step, because in this case the driver deletes intermediate assembly and object files automatically.

See Also

[-Xkeep-assembly-file on page](#)

-Xlibc-...

Use alternative C99 libraries.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xlibc-std
```

Description

Notes for All

Use alternative (Dinkumware) libraries. These libraries include C99 support and date from release 5.6 of the Wind River Diab Compiler.

-Xlib-old

C89/C++03 Selects legacy C/C++ (pre-release 5.6) libraries that are currently used by default.

-Xlib-std

Latest C/C++ standards

-Xlibc-old

Alias for -Xlib-old provided for backward compatibility.

-Xlibc-new

Alias for -Xlib-std provided for backward compatibility.

With release 5.6 of the Wind River Diab Compiler, updated versions of `libi.a`, `libcfp.a`, and `libm.a` were made available.

Older versions of these libraries are included as `libiold.a`, `libcfpold.a`, and `libmold.a`; you can utilize these libraries by specifying them individually on the command line or together by specifying **-Xlibc-old**.

-Xlibc-new implies C99 usage and is equivalent to **-Xc-new -Xdialect-c99**.

Note

We recommend that customers who want to use the latest standards use **-Xdialect-std**, which always selects the latest C and C++ standards (and corresponding libraries) supported by Diab. Only very rarely should customers need to explicitly pick an older standard. Code written for older standards will usually work fine with **-Xdialect-std**. To avoid breaking legacy applications, the default standard for C is C89 with legacy Diab C libraries, and for C++ the default is the C++03 standard. We strongly recommend not to set any of the `-Xlib...` options, the user should exclusively set one of the `-Xdialect...` options and use the libraries that come along with that.

Notes for ARM, ColdFire, MIPS, PowerPC, RH850, and x86

See Also

[-Xc-new on page](#)

[-Xdialect_... on page](#)

-Xpreprocess-assembly

Preprocess assembly files.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xpreprocess-assembly
```

Description

Invoke C preprocessor on assembly files before running the assembler.

-Xshow-target

Show target.

Belongs to:

[Driver Index on page 252](#)

Synopsis

```
-Xshow-target
```

Description

dcc C and **dplus** C++ driver option. Display the target processor “-t option” on standard output, but do not compile any file.

1.5. Compiler C-Plus-Plus X Options

Table of Contents

- [-Xbool-... on page](#)
- [-Xclass-type-name-visible on page](#)
- [-Xdigraphs-... on page](#)
- [-Xfor-init-scope-... on page](#)
- [-Ximplicit-templates-... on page](#)
- [-Xjmpbuf-size on page](#)
- [-Xmax-inst-level on page](#)
- [-Xnamespace-... on page](#)
- [-Xpch-... on page](#)
- [-Xshow-inst on page](#)
- [-Xsyntax-warning-... on page](#)
- [-Xusing-std-... on page](#)
- [-Xwchar_t-off on page](#)
- [-Xwchar_t-on on page](#)

-Xbool-...

Control use of bool, true, and false keywords.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xbool-on  
-X213=0
```

Description

Enable the **bool**, **true**, and **false** keywords. This is the default.

-Xbool-off

-X213

C++ only. Disable the **bool**, **true**, and **false** keywords.

Synonym: **-Xno-bool**.

-Xclass-type-name-visible

Use old for scope rules.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xclass-type-name-visible
```

Description

C only. Direct the compiler not to hide **struct** or **union** names when other identifiers with the same names are declared in the same scope. For example, consider the following statement:

```
struct S {...} S[10];
```

With or without this option, the form **struct S** may always be used later to declare additional variables of type **struct S**.

However, without the option, **sizeof(S)** will refer to the size of the array, while with this option, **sizeof(S)** will refer to the size of the structure.

-Xdigraphs-...

Disable digraphs.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xdigraphs-on
-X202=0
```

Description

C++ only. Enable digraphs. If digraphs are enabled, the compiler recognizes the following keywords as digraphs: **bitand**, **and**, **bitor**, **or**, **xor**, **compl**, **and_eq**, **or_eq**, **xor_eq**, **not**, and **not_eq**. This is the default.

-Xdigraphs-off

-X202

Disable digraphs.

Synonym: **-Xno-digraphs**.

-Xfor-init-scope-...

Use old for scope rules.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xfor-init-scope-for
-X217=0
```

Description

Use “new” scope rules for variables declared in the initialization part of a **for** statement. With this option, the scope of a variable declared in the initialization part extends to the end of the **for** statement.

```
-Xfor-init-scope-outer
-X217
```

C++ only. Use “old” scoping rules for variables declared in the initialization part of a **for** statement. With this option, the scope extends to the end of the scope enclosing the **for** statement.

Synonym: **-Xold-scoping**.

-Ximplicit-templates...

Control template instantiation.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Ximplicit-templates
-X207=0
```

Description

Instantiate each template in each module where it is used or referenced. This is the default.

```
-Ximplicit-templates-off
```

```
-X207=1
```

Instantiate templates only where explicit instantiation syntax is used.

Synonym: **-Xno-implicit-template**.

For more information, see the discussion of templates in the *C++ Features and Compatibility* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

C++ only.

-Xjmpbuf-size

Set longjmp buffer size.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xjmpbuf-size=n
-X201=n
```

Description

C++ only. Set the size in bytes of the buffer allocated for **setjmp** and **longjmp** when using exceptions. The default size as determined by the compiler should usually be sufficient.

-Xmax-inst-level

Set template instantiation recursion limit.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xmax-inst-level[=n]
-X216[=n]
```

Description

C++ only. Set the maximum level for recursive instantiation of templates. Without this option, an error is emitted when a default level of 50 is reached. With this option, but without a value **n**, the limit is 100.

-Xnamespace-...

Disable C++ keywords namespace and using.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xnamespace-on
-X219=0
```

Description

Recognize the **namespace** and **using** keywords or constructs.

```
-Xnamespace-off
-X219
```

C++ only. Do not recognize the **namespace** and **using** keywords or constructs.

-Xpch-...

Use precompiled headers.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xpch-automatic
```

Description

These options are disabled by default. At most one of **-Xpch-automatic**, **-Xpch-create**, and **-Xpch-use** can be enabled; if more than one is specified, all but the first are ignored.

For more information, see the discussion of precompiled headers in the *C++ Features and Compatibility* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xpch-automatic

Generate and use precompiled headers.

-Xpch-create=filename

Generate a precompiled header (PCH) file with specified name.

-Xpch-diagnostics

Generate an explanatory message for each PCH file that the compiler locates but is unable to use.

-Xpch-directory=directory

Look for PCH file in specified directory.

-Xpch-messages

Generate a message each time a PCH file is created or used.

-Xpch-use=filename

Use specified PCH file.

-Xshow-inst

Print instantiations.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xshow-inst
-X212
```

Description

C++ only. Print to **stderr** a list of all template instantiations made during compilation.

See also the discussion of templates in the *C++ Features and Compatibility* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

See Also

[-Ximplicit-templates on page](#)

-Xsyntax-warning-...

Disable certain syntax warnings.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xsyntax-warning-on
-X215=0
```

Description

Enable certain syntax warnings, for example, warning on a comma after the last enumerator. This is the default.

-Xsyntax-warning-off

-X215

C++ only. Disable these warnings.

-Xusing-std-...

Runtime declarations in standard namespace.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xusing-std-on
```

Description

C++ only. Automatically search for runtime library declarations in the **std** namespace (as if "**using namespace std;**" had been specified in the source code), not in global scope. This is the default behavior, but it is disabled by **-Xstrict-ansi**; use **-Xusing-std-on** on the command line to override **-Xstrict-ansi**.

This option allows you to use the newer C++ libraries, which are in the **std** namespace, without adding **using namespace std;** to legacy code.

-Xusing-std-off

Search for runtime library declarations in global scope unless an explicit **using namespace std;** is given.

-Xwchar_t-off

Disable use of wchar_t keyword.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xwchar_t-off
-X214
```

Description

C++ only. Disable the **wchar_t** keyword.

Synonym: **-Xno-wchar**.

See Also

[-Xwchar on page](#)

[-Xwchar_t-on on page](#)

-Xwchar_t-on

Enable use of wchar_t keyword.

Belongs to:

[C Plus Plus Index on page 248](#)

Synopsis

```
-Xwchar_t-on
-X214=0
```

Description

Enable the **wchar_t** keyword.

See Also

[-Xwchar on page](#)
[-Xwchar_t-off on page](#)

1.6. Compiler Option Macros

Table of Contents

- [-balanced on page](#)
- [-balanced-debug on page](#)
- [-size on page](#)
- [-size-debug on page](#)
- [-speed on page](#)
- [-speed-debug on page](#)

-balanced

Optimize code and balance between size and speed optimization.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-balanced
```

Description

This option macro is replaced by compiler switches that optimize the code, balanced between size and speed. The actual set of switches depends on the chosen target; they can be displayed by using the **-#** switch.

-balanced-debug

Optimize code and balance between size and speed optimization, and enable debugging.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-balanced-debug
```

Description

This option macro is replaced by compiler switches that optimize the code, balanced between size and speed, and enable debugging. The actual set of switches depends on the chosen target; they can be displayed by using the **-#** switch.

-size

Optimize code for size.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-size
```

Description

This option macro is replaced by compiler switches that optimize the code for size. The actual set of switches depends on the target selected; they can be displayed by using the **-#** switch.

-size-debug

Optimize code for size and enable debugging.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-size-debug
```

Description

This option macro is replaced by compiler switches that optimize the code for size and enable debugging. The code can be dramatically restructured when compiling for size and may be difficult to debug. Wind River does not recommend debugging with this option unless it is required to make the code fit into available memory. The actual set of switches depends on the target selected; they can be displayed by using the **-#** switch.

-speed

Optimize code for speed.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-speed
```

Description

This option macro is replaced by compiler switches that optimize the code for speed. The actual set of switches depends on the target selected; they can be displayed by using the **-#** switch.

-speed-debug

Optimize code for speed and enable debugging.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-speed-debug
```


Description

This option macro is replaced by compiler switches that optimize the code for speed and enable debugging. The actual set of switches depends on the target selected; they can be displayed by using the `-#` switch.

1.7. Compiler Architecture-Specific Options

- [ARM-Specific Compiler Options on page 148](#)
- [ARM-T-Specific Compiler Options on page 151](#)
- [ColdFire-Specific Compiler Options on page 153](#)
- [MIPS-Specific Compiler Options on page 157](#)
- [MIPS16-Specific Compiler Options on page 160](#)
- [RH850-Specific Compiler Options on page 162](#)
- [PowerPC-Specific Compiler Options on page 166](#)
- [PowerPCVLE-Specific Compiler Options on page 177](#)
- [TriCore-Specific Compiler Options on page 187](#)
- [Intelx86_Pentium-Specific Compiler Options on page 191](#)

1.7.1. ARM-Specific Compiler Options

Table of Contents

- [-Xadd-underscore on page](#)
- [-Xarm-fdiv-exc-hook on page](#)
- [-Xinterwork on page](#)
- [-Xno-movw on page](#)
- [-Xno-reloc-movw on page](#)
- [-Xsingle-fp on page](#)
- [-Xsoft-float on page](#)
- [-Xsplit-lld on page](#)

-Xadd-underscore

Prefix function identifiers with underscore.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xadd-underscore
-X34
```

Description

Prefix an underscore to function names only. Concatenation of underscore is useful when compiling libraries, to avoid using the same namespace as user programs.

-Xarm-fdiv-exc-hook

Allow users to customize the behavior of floating point division by zero.

Synopsis

```
-Xarm-fdiv-exc-hook
```

Description

When this option is specified, division by 0 (more precisely division by 0, NaN, or denormal) results in a call to the function `_f_div_exception` respectively `_d_div_exception` in case of double precision. The code generated for floating point division operations will:

1. Clear FPSCR bits DZC, IOC, IDC
2. Perform a division
3. Test FPSCR bits DZC, IOC, IDC and if set, invoke an exception handler, returning the result of that function rather than the result of the hardware division.

For convenience, the Diab libraries contain default definitions of `_f_div_exception` respectively `_d_div_exception` that simply return the result of doing a regular hardware division dividend/divisor.

To customize this behavior, applications should define one or both of the following functions:

```
#ifdef __cplusplus
extern "C"
#endif
float _f_div_exception(float dividend, float divisor);
or
#ifdef __cplusplus
extern "C"
#endif
double _d_div_exception(double dividend, double divisor);
```

Note

Division operations that can be completely evaluated at compile time (because both operands of the division are themselves compile time constants) do not generate a division instruction and will not call the user defined hook in case of division by zero. This is consistent with the behavior on other targets and the behavior of software floating point.

Generally this will not be an issue for applications, but test code that deliberately wants to test division by zero should use the following trick:

```
volatile float zero=0;
return 1/zero;
```

-Xinterwork

Enable interworking.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xinterwork
-X40
```

Description

Allow compiling mixed ARM and Thumb code with modules containing routines that can be called by routines for the other processor state.

-Xno-movw

Don't generate movw and movt instructions.

Synopsis

```
-Xno-movw
```

Description

This option has been added for ARMV7 and later devices(ARMV8),and ARM Thumb2 devices. If this option is set, movw and movt instructions will not become generated by the compiler.

-Xno-reloc-movw

Don't generate movw and movt instructions in case of source operand isn't of type constant.

Synopsis

```
-Xno-reloc-movw
```

Description

This option has been added for ARMV7 and later devices(ARMV8), and ARM Thumb2 devices. If this option is set, movw and movt instructions will not become generated by the compiler in case of the instructions source operand isn't a constant value.

-Xsingle-fp

internal option

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsingle-fp
```

Description

If **-Xsoft-float** is used in conjunction with option **-Xsingle-fp**, single precision operations will be done in hardware, and double-precision in software.

This is exclusively an internal option, not to be used by the user!

-Xsoft-float

internal option exclusively: Select software floating point emulation.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsoft-float
-X56
```

Description

This option causes double- and (unless `-Xsingle-fp` is specified) single-precision floating point operations to be done in software, rather than using hardware instructions. This implementation is a very fast, call-based method.

Note

Note that this option, and **`-Xsingle-fp`**, are controlled by DFP, which also selects which library to use. Both are internal options exclusively and should not usually be set explicitly by the user!

Note that if `-Xsingle-fp` is used in conjunction with this option, then single-precision operations will be done in hardware, and double-precision in software.

-Xsplit-llong

Split long long expressions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsplit-llong
```

Description

Split long long expressions into intermediate expressions.

The Wind River Diab Compiler for ARM splits long long expressions like the following:

```
x = a * b + c;
```

into a series of intermediate expressions:

```
temp = a * b;
```

```
x = temp + c;
```

A later compilation step propagates the temporary value, creating a single expression:

```
x = a * b + c;
```

If the compiler runs out of available registers trying to generate the code for this expression, it fails with the following error:

Expression too complex. Try to simplify.

To avoid this error, use the **`-Xsplit-llong`** option. This causes the compiler to skip the expression propagation step. Note that this may result in slower performance because the temporary values may become stack locations instead of registers, creating extra load and store instructions.

1.7.2. ARM-T-Specific Compiler Options

Table of Contents

- [-Xadd-underscore on page](#)
- [-Xinterwork on page](#)
- [-Xsoft-float on page](#)

-Xadd-underscore

Prefix function identifiers with underscore.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xadd-underscore
-X34
```

Description

Prefix an underscore to function names only. Concatenation of underscore is useful when compiling libraries, to avoid using the same namespace as user programs.

-Xinterwork

Enable interworking.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xinterwork
-X40
```

Description

Allow compiling mixed ARM and Thumb code with modules containing routines that can be called by routines for the other processor state.

-Xsoft-float

internal option exclusively: Select software floating point emulation.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsoft-float
-X56
```

Description

This option causes double- and (unless **-Xsingle-fp** is specified) single-precision floating point operations to be done in software, rather than using hardware instructions. This implementation is a very fast, call-based method.

Note

Note that this option, and **-Xsingle-fp**, are controlled by DFP, which also selects which library to use.

Both are internal options exclusively and should not usually be set explicitly by the user!

Note that if **-Xsingle-fp** is used in conjunction with this option, then single-precision operations will be done in hardware, and double-precision in software.

1.7.3. ColdFire-Specific Compiler Options

Table of Contents

- [-Xalign-off on page](#)
- [-Xasm-const-pound... on page](#)
- [-Xbig-switch-table on page](#)
- [-Xfloats-as-gnu on page](#)
- [-Xfloats-in-d0 on page](#)
- [-Xframe-ptr on page](#)
- [-Xhardware-divide... on page](#)
- [-Xinstr-00, -Xinstr-20 on page](#)
- [-Xptr-values-in-... on page](#)
- [-Xrts-nil on page](#)
- [-Xstruct-as-gnu on page](#)

-Xalign-off

Do not generate `.align` directive

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xalign-off
-X37
```

Description

Do not generate the `.align` directive (some assemblers do not support it). Use the `.even` directive instead.

The assembler `-Xdefault-align` option controls the alignment used by `.even` but not `.align`. Thus, alignment can be controlled by using `-Xalign-off` to the compiler and `-Xdefault-align` to the assembler or `-Xalign-functions=4` to the compiler (to align code).

See Also

[-Xdefault-align on page](#)

-Xasm-const-pound...

Disable `:::` prefix for assembly numeric constants.

Belongs to:

[Output Index on page 266](#)

Synopsis

```
-Xasm-const-pound
-X45=1
```

Description

Prefix each generated numeric constant in the assembly output with a `"#"` character. This is the default.

```
-Xasm-const-pound=2
-X45=2
```

Prefix each generated numeric constant in the assembly output with a “#” character, except for macro parameters passed as constants. That is, if the storage mode for the parameter is **con** then the “#” prefix is not added.

-Xasm-const-pound-off

-X45=0

Generate each numeric constant in the assembly output without a “#” prefix.

-Xbig-switch-table

Specify jump-table for switch statements.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xbig-switch-table[=n]
-X35=[=n]
```

Description

With **-Xbig-switch-table** or **-Xbig-switch-table=1**, use a 32-bit absolute jump-table. The code is larger but faster than with **-Xbig-switch-table=0**.

With **-Xbig-switch-table=0**, use a 16-bit relative jump table for **switch** statements. The code will be smaller but somewhat slower than with option **-Xbig-switch-table=1**. This may create difficulties with switch statements larger than 32KB. This is the default for ColdFire.

-Xfloats-as-gnu

Use GNU convention for returning floating point values.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xfloats-as-gnu
-X43
```

Description

Use GNU calling conventions and function names for software floating point emulation. GNU libraries are required in order to get a clean link.

See Also

[-Xfloats-in-d0 on page](#)

-Xfloats-in-d0

Use d0 for returning floating point values.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xfloats-in-d0
-X42
```

Description

Return floating point values both in register **fp0**, as well as registers **d0/d1**, when using hardware floating point.

See Also

[-Xfloats-as-gnu on page](#)

-Xframe-ptr

Generate link instruction.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xframe-ptr
-X36
```

Description

Generate a **link** instruction at the beginning of every function, and preserve saved registers with **moveml** as the second instruction if necessary. This makes it easy to backtrack to calling functions and to see what registers are saved. For more information, see the discussion of stack layout in the *Calling Conventions* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

This option is not on by default, and the **link** instruction is not generated (unless there are local variables on the stack) when the **-XO** option is used. Use **-Xframe-ptr** to force the compiler to generate the **link** instruction and preserve the registers in all cases.

-Xhardware-divide...

Control use of hardware divide instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xhardware-divide
-X40=1
```

Description

Use hardware divide instructions. This is the default for ColdFire processors with hardware divide instructions.

-Xhardware-divide-off

-X40=0

Do not use hardware divide instructions. This is the default for ColdFire processors without hardware divide instructions.

-Xinstr-00, -Xinstr-20

Permit 68020 instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xinstr-[00|20]
-X44=[0|1]
```

Description

Permit use of 68020 instructions on a ColdFire processor. This is the default.

-Xptr-values-in-...

Select convention for returning pointer values from functions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xptr-values-in-a0
-X32=1
```

Description

Return pointer values from functions in both register **d0** and register **a0**. The value in **a0** is used by the caller.

Note

This option is provided for compatibility with third-party tools and libraries. The Wind River libraries are compiled with the default, **-Xptr-values-in-d0**, and will not work with modules compiled with **-Xptr-values-in-a0** unless recompiled with that option.

-Xptr-values-in-d0

-X32=0

Return pointer values from functions in register **d0**. This is the default.

-Xrts-nil

Add null bytes after rts or rte for MCF5307 prefetch errata

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xrts-nil
-X47
```

Description

Add two null bytes after every **rte** or **rts** instruction to avoid an instruction prefetch errata on the ColdFire MCF5307. Necessary for this part only.

-Xrts-nil-off

-X47=0

Do not add the two null bytes. This is the default.

-Xstruct-as-gnu

Select convention for returning structures and unions. (ColdFire only)

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xstruct-as-gnu
-X80=10
```

Description

Rewrite functions returning C structures and unions as if the first argument is a pointer to the return area. This option is a ColdFire-specific version of the **-Xstruct-as-arg** and **-Xstruct-as-static** options.

Return structures from functions like the GNU C compiler. This option will cause small structures (up to a maximum of 8 bytes) to be returned in **d0** and **d1**, and larger structures to be returned by setting register **a1** to point to the return area.

This calling convention is required when interfacing with the GNU tool set.

For more information, see the discussion of class, structure, and union return types in the *Wind River Diab Compiler User's Guide* for your architecture.

1.7.4. MIPS-Specific Compiler Options

Table of Contents

- [-mfix-vr4133 on page](#)
- [-Xadd-underscore on page](#)
- [-Xconventions-eabi on page](#)
- [-Xconventions-n32 on page](#)
- [-Xmips16-hw-fp-call on page](#)
- [-Xmips-mad, -Xmips-movc, -Xmips-mul on page](#)
- [-Xmode-64bit on page](#)
- [-Xnested-interrupts on page](#)
- [-Xsoft-float on page](#)

-mfix-vr4133

Work Around VR4133 Bug.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-mfix-vr4133
```

Description

Use with the **-Xfix-vr4133** assembler option to implement a workaround for a problem that arises with NEC VR4133 processors when **mult/div/macc** instructions follow **mflo/mfhi** instructions.

See Also

[-Xfix-vr4133 on page](#)

-Xadd-underscore

Prefix function identifiers with underscore.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xadd-underscore
-X34
```

Description

Prefix an underscore to function names only. Concatenation of underscore is useful when compiling libraries, to avoid using the same namespace as user programs.

-Xconventions-eabi

Select EABI calling conventions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xconventions-eabi
-X31=0
```

Description

This option is controlled by **DOBJECT=E** and should not be set explicitly by the user. See the discussion of argument passing in the *Calling Conventions* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xconventions-n32

Select n32 calling conventions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xconventions-n32
-X31=2
```

Description

This option is controlled by **DOBJECT=E** and should not be set explicitly by the user.

-Xmips16-hw-fp-call

Pass floating point parameters to MIPS16 functions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xmips16-hw-fp-call
-X40
```

Description

When compiling MIPS32 code with hardware floating point which is to call MIPS16 code, cause the MIPS32 code to pass floating point parameters and accept return results in the general register set for use by the MIPS16 code in addition to the usual floating point registers (**\$f12** or **\$f14**).

-Xmips-mad, -Xmips-movc, -Xmips-mul

Enable specific MIPS instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xmips-mad
-Xmips=movc
-Xmips-mul
```

Description

These options enable code generation of these instructions for any MIPS processor:

When **-Xmips-mad** is specified, the compiler can generate the MIPS instructions **msub**, **msubu**, **mad** and **madu**.

When **-Xmips-movc** is specified, the MIPS conditional move instructions **movn** and **movz** can be generated.

When **-Xmips-mul** is specified, the compiler can generate the MIPS **mul** instruction.

-Xmode-64bit

Allow MIPS64 assembler syntax.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xmode-64bit
```

Description

Allows use of MIPS64 instructions in assembler code when compiling for older MIPS targets. It is not necessary to use this option when compiling with **-tMIPS64** or **-tMIPS-64**.

-Xnested-interrupts

Save registers for nested interrupts.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xnested-interrupts
-X41
```

Description

In order to support nested interrupts, when entering an interrupt function (a function named in a **#pragma interrupt** directive or declared with the **interrupt** or **__interrupt__** keywords), save the **C0_EPC** and **C0_SR** registers, and return from the function by restoring **C0_SR** and returning to the address saved from **C0_EPC** (which was saved only for use by the return and need not be restored).

-Xsoft-float

internal option exclusively: Select software floating point emulation.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsoft-float
-X56
```

Description

This option causes double- and (unless **-Xsingle-fp** is specified) single-precision floating point operations to be done in software, rather than using hardware instructions. This implementation is a very fast, call-based method.

Note

Note that this option, and **-Xsingle-fp**, are controlled by DFP, which also selects which library to use.

Both are internal options exclusively and should not usually be set explicitly by the user!

Note that if **-Xsingle-fp** is used in conjunction with this option, then single-precision operations will be done in hardware, and double-precision in software.

1.7.5. MIPS16-Specific Compiler Options

Table of Contents

- [-mfix-vr4133 on page](#)
- [-Xadd-underscore on page](#)
- [-Xdump-literals on page](#)
- [-Xmode-64bit on page](#)
- [-Xsoft-float on page](#)

-mfix-vr4133

Work Around VR4133 Bug.

Belongs to:

[Compiler Command-Line Options Index on page 250](#)

Synopsis

```
-mfix-vr4133
```

Description

Use with the **-Xfix-vr4133** assembler option to implement a workaround for a problem that arises with NEC VR4133 processors when **mult/div/macc** instructions follow **mflo/mfhi** instructions.

See Also

[-Xfix-vr4133 on page](#)

-Xadd-underscore

Prefix function identifiers with underscore.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xadd-underscore  
-X34
```

Description

Prefix an underscore to function names only. Concatenation of underscore is useful when compiling libraries, to avoid using the same namespace as user programs.

-Xdump-literals

Dump literals at the end of each function.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xdump-literals  
-X42=1
```

Description

Dump the literal pool at the end of each function. The literal pool holds large constants such as addresses. The default is off: the literal pool is dumped only when required by its size or addressing constraints.

-Xmode-64bit

Allow MIPS64 assembler syntax.

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xmode-64bit
```

Description

Allows use of MIPS64 instructions in assembler code when compiling for older MIPS targets. It is not necessary to use this option when compiling with **-tMIPS64** or **-tMIPS-64**.

-Xsoft-float

internal option exclusively: Select software floating point emulation.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsoft-float
-X56
```

Description

This option causes double- and (unless **-Xsingle-fp** is specified) single-precision floating point operations to be done in software, rather than using hardware instructions. This implementation is a very fast, call-based method.

Note

Note that this option, and **-Xsingle-fp**, are controlled by DFP, which also selects which library to use.

Both are internal options exclusively and should not usually be set explicitly by the user!

Note that if **-Xsingle-fp** is used in conjunction with this option, then single-precision operations will be done in hardware, and double-precision in software.

1.7.6. RH850-Specific Compiler Options**Table of Contents**

- [-Xadd-underscore on page](#)
- [-Xcritical-interrupts on page](#)
- [-Xhardware-divide... on page](#)
- [-Xnested-interrupts on page](#)
- [-Xrh850-interrupt-use-reg on page](#)
- [-Xsingle-fp on page](#)
- [-Xsoft-float on page](#)
- [-Xuse-fpsimd on page](#)

-Xadd-underscore

Prefix function identifiers with underscore.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xadd-underscore
-X34
```

Description

Prefix an underscore to function names only. Concatenation of underscore is useful when compiling libraries, to avoid using the same namespace as user programs.

-Xcritical-interrupts

Use critical interrupt return instruction if applicable.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xcritical-interrupts
-X41
```

Description

Option **-Xcritical-interrupts** changes the return instruction generated for an interrupt function if the selected target supports critical and non-critical interrupts.

For example, take below code.

```
#ifdef INTERRUPT
__interrupt__
#endif
void ff() {
  u = 1000;
}
```

The return instruction will depend on the target selected. There are currently 4 versions of the V850 architecture. For easier readability of the following table, we denote them as E0 to E3 as follows:

```
E0 = -tV850ES | -tV850FS
E1 = -tV850EES | -tV850EFS
E2 = -tV850E2*
E3 = -tV850E3* | -tRH850*
```

The following return instructions will be used depending on the selected target and whether the function is a plain function (i.e., no interrupt function at all), a non-critical, or critical interrupt.

	plain	non-critical	critical (-Xcritical-interrupts)
E0	jmp [r31]	reti	reti
E1	jmp [r31]	reti	reti
E2	jmp [r31]	eiret	feret
E3	jmp [r31]	eiret	feret

Since

5.9.4.0

-Xhardware-divide...

Control use of hardware divide instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xhardware-divide
-X35=1
```

Description

Use hardware divide instructions. This is the default for RH850.

```
-Xhardware-divide-off
-X35=0
```

-Xnested-interrupts

Save registers for nested interrupts.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xnested-interrupts
-X44
```

Description

In order to support nested interrupts, when entering an interrupt function (a function named in a **#pragma interrupt** directive or declared with the **interrupt** or **__interrupt__** keywords), save the **eipc** and **eipsw** basic context registers, and restore them on returning from the function.

-Xrh850-interrupt-use-reg

Generate save/restore code for system registers.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xrh850-interrupt-use-reg=reg_flag
-X31
```

Description

Interrupt routines compiled with **-Xrh850-interrupt-use-reg** will generate code to save/restore system registers.

The **reg_flag** argument is a bit mask, which indicates one of FPU/FXU/CALLT. Possible values of the bit mask in **reg_flag** are:

- 0x1 => FPU
- 0x2 => FXU
- 0x4 => CALLT

This flag instructs the compiler to generate save/restore instructions in the interrupt service routine for the system registers:

- If the FPU bit is SET, generate code to save/restore FPSR and FPEPC.
- If the FXU bit is SET, generate code to save/restore SESR and FXSR.

- If the CALLT bit is SET, generate code to save/restore CTPC and CTPSW.

-Xsingle-fp

internal option

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsingle-fp
```

Description

If **-Xsoft-float** is used in conjunction with option **-Xsingle-fp**, single precision operations will be done in hardware, and double-precision in software.

This is exclusively an internal option, not to be used by the user!

-Xsoft-float

internal option exclusively: Select software floating point emulation.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsoft-float
-X56
```

Description

This option causes double- and (unless **-Xsingle-fp** is specified) single-precision floating point operations to be done in software, rather than using hardware instructions. This implementation is a very fast, call-based method.

Note

Note that this option, and **-Xsingle-fp**, are controlled by DFP, which also selects which library to use.

Both are internal options exclusively and should not usually be set explicitly by the user!

Note that if **-Xsingle-fp** is used in conjunction with this option, then single-precision operations will be done in hardware, and double-precision in software.

-Xuse-fpsimd

Enable vector data types and intrinsics for RH850G4MH

Belongs to:

[Syntax Index on page 268](#)

Synopsis

```
-Xuse-fpsimd
```

Description

Enable vector data types and intrinsics for RH850G4MH

1.7.7. PowerPC-Specific Compiler Options

Table of Contents

- [-Xadd-underscore on page](#)
- [-Xapu-ibm-mac on page](#)
- [-Xapu-lsp on page](#)
- [-Xapu-sie on page](#)
- [-Xcode-model-absolute on page](#)
- [-Xcode-model-medium on page](#)
- [-Xcode-model-small on page](#)
- [-Xconventions-eabi on page](#)
- [-Xcrb6-always on page](#)
- [-Xcrb6-float on page](#)
- [-Xcrb6-never on page](#)
- [-Xdebug-interrupt on page](#)
- [-Xeieio on page](#)
- [-Xframe-align on page](#)
- [-Xintc-eoir on page](#)
- [-Xinterrupt-funcs-use-rfci on page](#)
- [-Xlsp-int on page](#)
- [-Xmachine-check-interrupt on page](#)
- [-Xnested-interrupts on page](#)
- [-Xsavefpr-avoid on page](#)
- [-Xsingle-fp on page](#)
- [-Xsoft-float on page](#)
- [-Xspe-fpmac on page](#)
- [-Xspe-int on page](#)
- [-Xstmw-fast, -Xstmw-ok, -Xstmw-slow on page](#)
- [-Xstsw-fast, -Xstsw-ok, -Xstsw-slow on page](#)
- [-Xtrace-table on page](#)
- [-Xupdate-slow on page](#)
- [-Xvector-... on page](#)
- [-Xvrsave-... on page](#)

-Xadd-underscore

Prefix function identifiers with underscore.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xadd-underscore
-X34
```

Description

Prefix an underscore to function names only. Concatenation of underscore is useful when compiling libraries, to avoid using the same namespace as user programs.

-Xapu-ibm-mac

Enable multiply and accumulate instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xapu-ibm-mac
```

Description

Enable the generation of multiply and accumulate (MAC) instructions for processors that support these instructions, such as PPC405, PPC440, PPC440GX, and PPC476. Enabled by default for these processors; to disable, set **-Xapu-ibm-mac=0**.

-Xapu-lsp

Enable LSP instructions and intrinsic functions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xapu-lsp
```

Description

This switch applies to both **dcc** and **das**. For **dcc** it enables the intrinsic functions for the lightweight signal processing (LSP) instructions, whereas for **das** it enables the recognition of and support for the LSP instructions. For information about LSP, see the *Lightweight Signal Processing APU Manual*.

-Xapu-sie

Enable saturation instruction extension (SIE) instructions and intrinsic functions.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xapu-sie
```

Description

Use some instructions belonging to the saturation instruction extension (SIE) in code generation to get better performance. Enable the intrinsic functions for SIE instructions.

-Xcode-model-absolute

select corresponding code model for PPC64 targets.

Synopsis

```
-Xcode-model-absolute
```

Description

Generate absolute addressing code for PPC64 targets.

See Also

[-Xcode-model-small on page](#)
[-Xcode-model-medium on page](#)

-Xcode-model-medium

select corresponding code model for PPC64 targets.

Synopsis

```
-Xcode-model-medium
```

Description

Generate medium TOC addressing code for PPC64 targets. Global data limited to 4G. This option is set by default for PPC64 targets.

See Also

[-Xcode-model-absolute on page](#)
[-Xcode-model-small on page](#)

-Xcode-model-small

select corresponding code model for PPC64 targets.

Synopsis

```
-Xcode-model-small
```

Description

Generate small TOC addressing code for PPC64 targets. Global data limited to 64K.

See Also

[-Xcode-model-absolute on page](#)
[-Xcode-model-medium on page](#)

-Xconventions-eabi

Select EABI calling conventions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xconventions-eabi  

-X31=0
```

Description

This option is controlled by **OBJECT=E** and should not be set explicitly by the user. See the discussion of argument passing in the *Calling Conventions* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xcrb6-always

Always set CR bit 6 for an unprototyped function call.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xcrb6-always
-X36=2
```

Description

With EABI conventions, normally **CRB6** is set when calling a prototyped function with a variable number of arguments if any argument is a floating type, and cleared if not. Since it is impossible for the compiler to determine if a function without a prototype uses variable arguments, the **-Xcrb6** options define when to set or clear **CRB6** if there is no prototype:

-Xcrb6-always will always set or clear **CRB6** for functions without prototypes.

See Also

[-Xcrb6-float on page](#)

[-Xcrb6-never on page](#)

-Xcrb6-float

Set CR bit 6 for an unprototyped function call using float.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xcrb6-float
-X36=1
```

Description

With EABI conventions, normally **CRB6** is set when calling a prototyped function with a variable number of arguments if any argument is a floating type, and cleared if not. Since it is impossible for the compiler to determine if a function without a prototype uses variable arguments, the **-Xcrb6** options define when to set or clear **CRB6** if there is no prototype:

-Xcrb6-float will set **CRB6** for functions without prototypes if any floating point argument is used. This is the default.

See Also

[-Xcrb6-always on page](#)

[-Xcrb6-never on page](#)

-Xcrb6-never

Never set CR bit 6 for an unprototyped function call.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xcrb6-never
-X36=0
```

Description

With EABI conventions, normally **CRB6** is set when calling a prototyped function with a variable number of arguments if any argument is a floating type, and cleared if not. Since it is impossible for the compiler to determine if a function without a prototype uses variable arguments, the **-Xcrb6** options define when to set or clear **CRB6** if there is no prototype:

-Xcrb6-never will never set nor clear **CRB6** for functions without prototypes.

See Also

[-Xcrb6-always on page](#)
[-Xcrb6-float on page](#)

-Xdebug-interrupt

Generate rfdi instruction for interrupt functions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xdebug-interrupt
```

Description

Generate an **rfdi** (return from machine check interrupt) instruction, instead of an **rfi** (return from interrupt) instruction, when returning from an **__interrupt__** function.

If **-Xnested-interrupts** is also used, **dsrr0** and **dsrr1** will also be saved for the **__interrupt__** function.

See Also

[-Xnested-interrupts on page](#)
[-Xinterrupt-funcs-use-rfdi on page](#)
[-Xmachine-check-interrupt on page](#)

-Xeieio

Prevent reordering of memory access.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xeieio
-X43
```

Description

Generate an **ieio** instruction after every load or store of a **volatile** variable. This prevents the hardware from changing the order of memory accesses.

-Xframe-align

Change the stack frame alignment.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xframe-align=n
-X49=n
```

Description

Align the stack frame to **n**-byte boundaries, **n** being 4, 8, or 16. Smaller stack alignments reduce the amount of memory required at runtime. However, changing the frame alignment to values smaller than the EABI-defined value will break EABI compliance.

It is strongly recommended that all objects and libraries that go into a program use the same stack alignment

-Xintc-eoir

Write end of interrupt register.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xintc-eoir=n
-X48=n
```

Description

For some PPC microcontrollers with e200 z3, z6 and z7 cores there is an "end of interrupt register" **INTC_EOIR** that must be written in the interrupt epilog. Examples are MPC55xx and MPC56xx.

With option **-Xintc-eoir=1** the compiler generates code to write the **INTC_EOIR** register in the interrupt epilog. The code looks like the following, where **0xFFFF48018** is the address of the **INTC_EOIR** register:

```
mbar    0
lis     r12, 0xFFFF48018\@ha
wrteei  0
stw     r0, 0xFFFF48018\@l (r12)
```

With option **-Xintc-eoir=2** the compiler generates code to write the **INTC_EOIR** register in the interrupt epilog, and also adds a **wrteei 1** instruction to the interrupt prolog to enable external interrupts. This is required if the interrupt request source is configured as hardware vector mode.

If the target is **-tPPC55xx** or **-tPPC56xx**, option **-Xintc-eoir=1** is set implicitly, and does not need to be added to the command options unless **-Xintc-eoir=2** is required.

If the target is a core name but not a microcontroller name, for example **-tPPCE200z3/6/7**, you must specify **-Xintc-eoir=1** or **-Xintc-eoir=2** if the target microcontroller has the **INTC_EOIR** register.

-Xinterrupt-funcs-use-rfci

Return from interrupt function using rfci.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xinterrupt-funcs-use-rfci
-X41=1
```

Description

When returning from an interrupt function (a function named in a **#pragma interrupt** directive or declared with the **interrupt** or **__interrupt__** keywords), use the **rfci** instruction instead of the general **rfi** instruction.

Synonym: **-Xcritical-interrupts**.

See Also

[-Xdebug-interrupt on page](#)

[-Xmachine-check-interrupt on page](#)

-Xlsp-int

Enable use of LSP instructions for integer data types.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xlsp-int
```

Description

Enable the use of certain lightweight signal processing (LSP) instructions for integer operations. This may improve the performance of programs. For information about LSP, see the *Lightweight Signal Processing APU Manual*.

-Xmachine-check-interrupt

Generate return for machine-check interrupt.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xmachine-check-interrupt
-X41=2
```

Description

Generate an **rfmci** (return from machine check interrupt) instruction, instead of an **rfi** (return from interrupt) instruction, when returning from an **__interrupt__** function.

If **-Xnested-interrupts** is also used, **mcsrr0** and **mcsrr1** will also be saved for the **__interrupt__** function.

See Also

[-Xnested-interrupts on page](#)

[-Xdebug-interrupt on page](#)

[-Xinterrupt-funcs-use-rfci on page](#)

-Xnested-interrupts

Save registers for nested interrupts.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xnested-interrupts
-X40
```

Description

In order to support nested interrupts, when entering an interrupt function (a function named in a **#pragma interrupt** directive or declared with the **interrupt** or **__interrupt__** keywords), save the **ssr0** and **ssr1** registers, and restore them on returning from the function.

-Xsavefpr-avoid

Save floating point registers without calling libimpl.a

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsavefpr-avoid
```

Description

By default, the compiler calls helper functions in `libimpl.a` to save and restore floating point registers. If **-Xsavefpr-avoid** is specified, the compiler instead generates inlined code to preserve these registers.

See Also

[-Xstmw-... on page](#)

-Xsingle-fp

internal option

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsingle-fp
```

Description

If **-Xsoft-float** is used in conjunction with option **-Xsingle-fp**, single precision operations will be done in hardware, and double-precision in software.

This is exclusively an internal option, not to be used by the user!

-Xsoft-float

internal option exclusively: Select software floating point emulation.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsoft-float
-X56
```

Description

This option causes double- and (unless **-Xsingle-fp** is specified) single-precision floating point operations to be done in software, rather than using hardware instructions. This implementation is a very fast, call-based method.

Note

Note that this option, and **-Xsingle-fp**, are controlled by DFP, which also selects which library to use.

Both are internal options exclusively and should not usually be set explicitly by the user!

Note that if **-Xsingle-fp** is used in conjunction with this option, then single-precision operations will be done in hardware, and double-precision in software.

-Xspe-fpmac

Accept SPE multiply-accumulate instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xspe-fpmac
```

Description

Allow Signal Processing Engine (SPE) multiply-accumulate instructions to be used with e500 processors.

-Xspe-int

Enable support for SPE integer instructions

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xspe-int
-x33
```

Description

Processors with SPE1 or SPE2 units may achieve better performance for their programs by using **-Xspe-int**.

The Wind River Diab Compiler previously only supported SPE vector instructions, but using certain SPE integer instructions, such as **evldd**, **evldw**, and **evabs**, may improve performance.

-Xstmw-fast, -Xstmw-ok, -Xstmw-slow

Select stmw and lmw instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xstmw-fast      (n=2)
-Xstmw-ok        (n=1)
-Xstmw-slow      (n=0)
-X32=n
```

Description

Specifies whether the **stmw** and the **lmw** instructions should be used to save/restore registers at function entry/exit.

-Xstmw-slow means that they should never be used. You must use the linker option **-lc** with **-Xstmw-slow**.

-Xstmw-ok means that they can be used in **leaf** functions.

-Xstmw-fast means that they should always be used instead of calling a library function to perform the operation. The **-t** option sets the optimal value for each processor.

See Also

[-Xsavefpr-avoid on page](#)

-Xstsw-fast, -Xstsw-ok, -Xstsw-slow

Select stswi and lswi instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xstsw-fast      (n=2)
-Xstsw-ok        (n=1)
-Xstsw-slow      (n=0)
-X35=n
```

Description

-Xstsw-fast means these instructions can always be used instead of using individual **lwz** and **stw** instructions. **-Xstsw-slow** means that they should never be used. **-Xstsw-ok** means they can be used for unaligned assignments.

PowerPC systems that have hardware paging enabled may need to use **-Xstsw-slow** to prevent an exception from an **stswi** instruction causing a page fault.

The **-tPPC...** option sets the optimal value for each processor.

-Xtrace-table

Generate trace table.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xtrace-table=n
-X38=n
```

Description

Generate the trace table needed to do a back-trace. **n** = 0 disables the option; **n** = 1 or absent enables the option. This option is disabled by default to save space.

-Xupdate-slow

Avoid pre-increment and pre-decrement instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xupdate-slow
```

Description

Avoid using pre-increment and pre-decrement instructions, as in some cases running such instructions can be slower than running two separate instructions.

For example, the following instruction

```
lwzu    r4,8(r9)
```

is the same as using

```
addi    r9,r9,8
```

```
lwz     r4,0(r9)
```

However, on some CPUs it is faster to perform the latter two instructions than to perform the pre-increment instruction; in such cases, using **-Xupdate-slow** may speed up program execution.

Note that the compiler may still generate pre-increment and pre-decrement instructions, but will do so less often.

-Xvector-...

Vector code generation.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xvector-on
-X44=1
```

Description

Specifies vector code generation. Applies to **AltiVec** with the PPC7400 or PPC970 processor only and is enabled by the Vector floating point option.

```
-Xvector-off
```

```
-X44=0
```

Note

Do not use this option directly. Vector processing is controlled by specifying a target processor with **Altivec** instructions, using the **-t** option.

-Xvrsave-...

VRSAGE special purpose register.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xvrsave-on
-X45=1
```

Description

Generate prolog and epilog code to save, adjust, and restore the VRSAGE special-purpose register needed to inform the OS about currently used vector registers. Applies only to the 7400-series and PPC970 processors with Altivec support.

-Xvrsave-off

-X45=0

This is the default.

1.7.8. PowerPCVLE-Specific Compiler Options**Table of Contents**

- [-Xadd-underscore on page](#)
- [-Xapu-lsp on page](#)
- [-Xapu-sie on page](#)
- [-Xconventions-eabi on page](#)
- [-Xcrb6-always on page](#)
- [-Xcrb6-float on page](#)
- [-Xcrb6-never on page](#)
- [-Xdebug-interrupt on page](#)
- [-Xeieio on page](#)
- [-Xframe-align on page](#)
- [-Xintc-eoir on page](#)
- [-Xinterrupt-funcs-use-rfci on page](#)
- [-Xlsp-int on page](#)
- [-Xmachine-check-interrupt on page](#)
- [-Xnested-interrupts on page](#)
- [-Xsavefpr-avoid on page](#)
- [-Xsingle-fp on page](#)
- [-Xsoft-float on page](#)
- [-Xspe-fpmac on page](#)
- [-Xspe-int on page](#)
- [-Xstmw-fast, -Xstmw-ok, -Xstmw-slow on page](#)
- [-Xstsw-fast, -Xstsw-ok, -Xstsw-slow on page](#)
- [-Xtrace-table on page](#)
- [-Xupdate-slow on page](#)
- [-Xvector-... on page](#)
- [-Xvrsave-... on page](#)

-Xadd-underscore

Prefix function identifiers with underscore.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xadd-underscore
-X34
```

Description

Prefix an underscore to function names only. Concatenation of underscore is useful when compiling libraries, to avoid using the same namespace as user programs.

-Xapu-lsp

Enable LSP instructions and intrinsic functions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xapu-lsp
```

Description

This switch applies to both **dcc** and **das**. For **dcc** it enables the intrinsic functions for the lightweight signal processing (LSP) instructions, whereas for **das** it enables the recognition of and support for the LSP instructions. For information about LSP, see the *Lightweight Signal Processing APU Manual*.

-Xapu-sie

Enable saturation instruction extension (SIE) instructions and intrinsic functions.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xapu-sie
```

Description

Use some instructions belonging to the saturation instruction extension (SIE) in code generation to get better performance. Enable the intrinsic functions for SIE instructions.

-Xconventions-eabi

Select EABI calling conventions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xconventions-eabi
-X31=0
```

Description

This option is controlled by **DOBJECT=E** and should not be set explicitly by the user. See the discussion of argument passing in the *Calling Conventions* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xcrb6-always

Always set CR bit 6 for an unprototyped function call.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xcrb6-always
-X36=2
```

Description

With EABI conventions, normally **CRB6** is set when calling a prototyped function with a variable number of arguments if any argument is a floating type, and cleared if not. Since it is impossible for the compiler to determine if a function without a prototype uses variable arguments, the **-Xcrb6** options define when to set or clear **CRB6** if there is no prototype:

-Xcrb6-always will always set or clear **CRB6** for functions without prototypes.

See Also

[-Xcrb6-float on page](#)
[-Xcrb6-never on page](#)

-Xcrb6-float

Set CR bit 6 for an unprototyped function call using float.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xcrb6-float
-X36=1
```

Description

With EABI conventions, normally **CRB6** is set when calling a prototyped function with a variable number of arguments if any argument is a floating type, and cleared if not. Since it is impossible for the compiler to determine if a function without a prototype uses variable arguments, the **-Xcrb6** options define when to set or clear **CRB6** if there is no prototype:

-Xcrb6-float will set **CRB6** for functions without prototypes if any floating point argument is used. This is the default.

See Also

[-Xcrb6-always on page](#)
[-Xcrb6-never on page](#)

-Xcrb6-never

Never set CR bit 6 for an unprototyped function call.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xcrb6-never
-X36=0
```

Description

With EABI conventions, normally **CRB6** is set when calling a prototyped function with a variable number of arguments if any argument is a floating type, and cleared if not. Since it is impossible for the compiler to determine if a function without a prototype uses variable arguments, the **-Xcrb6** options define when to set or clear **CRB6** if there is no prototype:

-Xcrb6-never will never set nor clear **CRB6** for functions without prototypes.

See Also

[-Xcrb6-always on page](#)

[-Xcrb6-float on page](#)

-Xdebug-interrupt

Generate rfdi instruction for interrupt functions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xdebug-interrupt
```

Description

Generate an **rfdi** (return from machine check interrupt) instruction, instead of an **rfi** (return from interrupt) instruction, when returning from an **__interrupt__** function.

If **-Xnested-interrupts** is also used, **dsrr0** and **dsrr1** will also be saved for the **__interrupt__** function.

See Also

[-Xnested-interrupts on page](#)

[-Xinterrupt-funcs-use-rfdi on page](#)

[-Xmachine-check-interrupt on page](#)

-Xeieio

Prevent reordering of memory access.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xeieio
-X43
```

Description

Generate an **eieio** instruction after every load or store of a **volatile** variable. This prevents the hardware from changing the order of memory accesses.

-Xframe-align

Change the stack frame alignment.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xframe-align=n
-X49=n
```

Description

Align the stack frame to **n**-byte boundaries, **n** being 4, 8, or 16. Smaller stack alignments reduce the amount of memory required at runtime. However, changing the frame alignment to values smaller than the EABI-defined value will break EABI compliance.

It is strongly recommended that all objects and libraries that go into a program use the same stack alignment

-Xintc-eoir

Write end of interrupt register.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xintc-eoir=n
-X48=n
```

Description

For some PPC microcontrollers with e200 z3, z6 and z7 cores there is an "end of interrupt register" **INTC_EOIR** that must be written in the interrupt epilog. Examples are MPC55xx and MPC56xx.

With option **-Xintc-eoir=1** the compiler generates code to write the **INTC_EOIR** register in the interrupt epilog. The code looks like the following, where **0xFFFF48018** is the address of the **INTC_EOIR** register:

```
mbar    0
lis     r12, 0xFFFF48018@ha
wrteei  0
stw     r0, 0xFFFF48018@l (r12)
```

With option **-Xintc-eoir=2** the compiler generates code to write the **INTC_EOIR** register in the interrupt epilog, and also adds a **wrteei 1** instruction to the interrupt prolog to enable external interrupts. This is required if the interrupt request source is configured as hardware vector mode.

If the target is **-tPPC55xx** or **-tPPC56xx**, option **-Xintc-eoir=1** is set implicitly, and does not need to be added to the command options unless **-Xintc-eoir=2** is required.

If the target is a core name but not a microcontroller name, for example **-tPPCE200z3/6/7**, you must specify **-Xintc-eoir=1** or **-Xintc-eoir=2** if the target microcontroller has the **INTC_EOIR** register.

-Xinterrupt-funcs-use-rfci

Return from interrupt function using rfci.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xinterrupt-funcs-use-rfci
-X41=1
```

Description

When returning from an interrupt function (a function named in a **#pragma interrupt** directive or declared with the **interrupt** or **__interrupt__** keywords), use the **rfci** instruction instead of the general **rfi** instruction.

Synonym: **-Xcritical-interrupts**.

See Also

[-Xdebug-interrupt on page](#)

[-Xmachine-check-interrupt on page](#)

-Xlsp-int

Enable use of LSP instructions for integer data types.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xlsp-int
```

Description

Enable the use of certain lightweight signal processing (LSP) instructions for integer operations. This may improve the performance of programs. For information about LSP, see the *Lightweight Signal Processing APU Manual*.

-Xmachine-check-interrupt

Generate return for machine-check interrupt.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xmachine-check-interrupt
-X41=2
```

Description

Generate an **rfmci** (return from machine check interrupt) instruction, instead of an **rfi** (return from interrupt) instruction, when returning from an **__interrupt__** function.

If **-Xnested-interrupts** is also used, **mcsrr0** and **mcsrr1** will also be saved for the **__interrupt__** function.

See Also

[-Xnested-interrupts on page](#)
[-Xdebug-interrupt on page](#)
[-Xinterrupt-funcs-use-rfci on page](#)

-Xnested-interrupts

Save registers for nested interrupts.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xnested-interrupts
-X40
```

Description

In order to support nested interrupts, when entering an interrupt function (a function named in a **#pragma interrupt** directive or declared with the **interrupt** or **__interrupt__** keywords), save the **ssr0** and **ssr1** registers, and restore them on returning from the function.

-Xsavefpr-avoid

Save floating point registers without calling libimpl.a

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsavefpr-avoid
```

Description

By default, the compiler calls helper functions in `libimpl.a` to save and restore floating point registers. If **-Xsavefpr-avoid** is specified, the compiler instead generates inlined code to preserve these registers.

See Also

[-Xstmw-... on page](#)

-Xsingle-fp

internal option

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsingle-fp
```

Description

If **-Xsoft-float** is used in conjunction with option **-Xsingle-fp**, single precision operations will be done in hardware, and double-precision in software.

This is exclusively an internal option, not to be used by the user!

-Xsoft-float

internal option exclusively: Select software floating point emulation.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsoft-float
-X56
```

Description

This option causes double- and (unless **-Xsingle-fp** is specified) single-precision floating point operations to be done in software, rather than using hardware instructions. This implementation is a very fast, call-based method.

Note

Note that this option, and **-Xsingle-fp**, are controlled by DFP, which also selects which library to use.

Both are internal options exclusively and should not usually be set explicitly by the user!

Note that if **-Xsingle-fp** is used in conjunction with this option, then single-precision operations will be done in hardware, and double-precision in software.

-Xspe-fpmac

Accept SPE multiply-accumulate instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xspe-fpmac
```

Description

Allow Signal Processing Engine (SPE) multiply-accumulate instructions to be used with e500 processors.

-Xspe-int

Enable support for SPE integer instructions

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xspe-int
-x33
```

Description

Processors with SPE1 or SPE2 units may achieve better performance for their programs by using **-Xspe-int**.

The Wind River Diab Compiler previously only supported SPE vector instructions, but using certain SPE integer instructions, such as **evldd**, **evldw**, and **evabs**, may improve performance.

-Xstmw-fast, -Xstmw-ok, -Xstmw-slow

Select stmw and lmw instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xstmw-fast      (n=2)
-Xstmw-ok        (n=1)
-Xstmw-slow      (n=0)
-X32=n
```

Description

Specifies whether the **stmw** and the **lmw** instructions should be used to save/restore registers at function entry/exit.

-Xstmw-slow means that they should never be used. You must use the linker option **-lc** with **-Xstmw-slow**.

-Xstmw-ok means that they can be used in **leaf** functions.

-Xstmw-fast means that they should always be used instead of calling a library function to perform the operation. The **-t** option sets the optimal value for each processor.

See Also

[-Xsavefpr-avoid on page](#)

-Xstsw-fast, -Xstsw-ok, -Xstsw-slow

Select stswi and lswi instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xstsw-fast      (n=2)
-Xstsw-ok        (n=1)
-Xstsw-slow      (n=0)
-X35=n
```

Description

-Xstsw-fast means these instructions can always be used instead of using individual **lwz** and **stw** instructions. **-Xstsw-slow** means that they should never be used. **-Xstsw-ok** means they can be used for unaligned assignments.

PowerPC systems that have hardware paging enabled may need to use **-Xstsw-slow** to prevent an exception from an **stswi** instruction causing a page fault.

The **-tPPC...** option sets the optimal value for each processor.

-Xtrace-table

Generate trace table.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xtrace-table=n
-X38=n
```

Description

Generate the trace table needed to do a back-trace. **n** = 0 disables the option; **n** = 1 or absent enables the option. This option is disabled by default to save space.

-Xupdate-slow

Avoid pre-increment and pre-decrement instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xupdate-slow
```

Description

Avoid using pre-increment and pre-decrement instructions, as in some cases running such instructions can be slower than running two separate instructions.

For example, the following instruction

```
lwzu    r4,8(r9)
```

is the same as using

```
addi    r9,r9,8
```

```
lwz     r4,0(r9)
```

However, on some CPUs it is faster to perform the latter two instructions than to perform the pre-increment instruction; in such cases, using **-Xupdate-slow** may speed up program execution.

Note that the compiler may still generate pre-increment and pre-decrement instructions, but will do so less often.

-Xvector...

Vector code generation.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xvector-on
-X44=1
```

Description

Specifies vector code generation. Applies to **Altivec** with the PPC7400 or PPC970 processor only and is enabled by the Vector floating point option.

```
-Xvector-off
-X44=0
```

Note

Do not use this option directly. Vector processing is controlled by specifying a target processor with **Altivec** instructions, using the **-t** option.

-Xvrsave-...

VRSAGE special purpose register.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xvrsave-on
-X45=1
```

Description

Generate prolog and epilog code to save, adjust, and restore the VRSAGE special-purpose register needed to inform the OS about currently used vector registers. Applies only to the 7400-series and PPC970 processors with Altivec support.

```
-Xvrsave-off
-X45=0
```

This is the default.

1.7.9. TriCore-Specific Compiler Options

Table of Contents

- [-Xadd-underscore on page](#)
- [-Xalign-int-by2 on page](#)
- [-Xdsync on page](#)
- [-Xerrata-mask on page](#)
- [-Xhardware-divide... on page](#)
- [-Xpipeline-short-circuit on page](#)
- [-Xsingle-fp on page](#)
- [-Xsoft-float on page](#)
- [-Xtc-fdiv-exc-hook on page](#)

-Xadd-underscore

Prefix function identifiers with underscore.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xadd-underscore
-X34
```

Description

Prefix an underscore to function names only. Concatenation of underscore is useful when compiling libraries, to avoid using the same namespace as user programs.

-Xalign-int-by2

Change alignment of long integer types to two bytes.

Belongs to:

[Memory Index on page 255](#)

Synopsis

```
-Xalign-int-by2
-X37
```

Description

Align integers to 2-byte memory boundaries instead of standard 4-byte boundaries. This option should be used with care, as it implies an intentional EABI violation and impacts inter module compatibility, but it allows for greater packing and consequently less memory consumption.

The option is intended to be used with ISA 1.6 targets, as there is a distinction in alignment for memory type in earlier versions of the TriCore architecture, where the TriCore ISA distinguished alignment requirements based on whether the memory was internal scratch memory or external off-chip memory.

Consequently, with ISA 1.2/1.3 targets this option should be used with caution. The Wind River Diab Compiler libraries are compiled in a way that allows them to be used both with code that has been compiled with and without this option.

-Xdsync

Synchronize volatile data.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xdsync-on
-X43=1
```

Description

Protect **volatile** variables by generating a **dsync** instruction after every load or store of a **volatile** variable.

-Xdsync-off

-X43=0

Turn **volatile** variable protection off.

-Xerrata-mask

Use workarounds for TriCore hardware errors.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xerrata-mask=mask
-X44=mask
```

Description

Use a workaround to avoid hardware-based errors. The following masks are used:

mask	TriCode erratum	error
0x100	CPU_TC.024	incorrect return address when performing nested calls
0x020	CPU_TC.050	load instructions following multi-cycle integer instruction can get lost
0x010	CPU_TC.060	ld.a/ld.da followed by a dependent ld.da/ld.d/ld.dw can produce unreliable results

For more information on hardware errata, consult the Errata Sheet documentation from Infineon.

-Xhardware-divide...

Enable hardware divide instructions.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xhardware-divide
-X35=1
```

Description

Enable generation of the hardware divide instructions, **div** and **divu**. This is the default.

-Xhardware-divide-off

-X35=0

Do not use the hardware divide instructions.

Synonym: **-Xno-hardware-divide**.

-Xpipeline-short-circuit

Optimize evaluation of logical operators && and ||.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xpipeline-short-circuit
-X49
```

Description

Per definition, evaluation of the logical operators **&&** and **||** is performed from left to right. The evaluation is guaranteed to stop as soon as it becomes impossible to change the current (intermediate) result by continued evaluation, making the current result the final one. This is also known as “short-circuit” evaluation. For **&&**, evaluation stops if the current result is zero; for **||**, it stops when the current result is nonzero.

Taking the logical **AND (&&)** as an example, this means that

if (expr1 && expr2) statement;

Is functionally equivalent to

if (expr1) if (expr2) statement;

The first **if**-clause is the short-circuit (if it fails, there is no need to check the second expression), and the second **if**-clause is the decisive one; the construct involves two conditional branches.

If **-Xpipeline-short-circuit** is enabled, and the compiler can determine that **expr2** has no side effects (like altering values or causing traps), the evaluation of **expr1** and **expr2** and their final logical **AND** will be pipelined, thereby saving one conditional branch. Provided **&expr2** fits into the pipeline stall cycles of the evaluation of **expr1**, the more complex evaluation does not cost any runtime (no matter if the short circuit is taken or not), but the evaluation will be faster if the short-circuit is not taken.

-Xpipeline-short-circuit should be used with care, as it should only be used on targets with deep pipelines.

-Xsingle-fp

internal option

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsingle-fp
```

Description

If **-Xsoft-float** is used in conjunction with option **-Xsingle-fp**, single precision operations will be done in hardware, and double-precision in software.

This is exclusively an internal option, not to be used by the user!

-Xsoft-float

internal option exclusively: Select software floating point emulation.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xsoft-float
-X56
```

Description

This option causes double- and (unless `-Xsingle-fp` is specified) single-precision floating point operations to be done in software, rather than using hardware instructions. This implementation is a very fast, call-based method.

Note

Note that this option, and **`-Xsingle-fp`**, are controlled by DFP, which also selects which library to use.

Both are internal options exclusively and should not usually be set explicitly by the user!

Note that if `-Xsingle-fp` is used in conjunction with this option, then single-precision operations will be done in hardware, and double-precision in software.

`-Xtc-fdiv-exc-hook`

Allow users to customize the behavior of single precision floating point division by zero.

Synopsis

```
-Xtc-fdiv-exc-hook
```

Description

When this option is specified, division by 0 (more precisely division by 0, NaN, or denormal) results in a call to the function `_f_div_exception`.

For convenience, the Diab libraries contain a default definition of `_f_div_exception` that simply returns the result of doing a regular hardware division (`div.f`).

To customize this behavior, applications should define a function with the following prototype:

```
#ifdef __cplusplus
extern "C"
#endif
float _f_div_exception(float dividend, float divisor);\\
or
#ifdef __cplusplus
extern "C"
#endif
double _d_div_exception(double dividend, double divisor);
```

Note

1. Using software defined traps requires the compiler to add check code to each division operation. This adds an additional 4 instructions to the execution path of each (non-exceptional) divide, and an additional code footprint of 14 - 20 bytes per division.
2. Division operations that can completely evaluated at compile time (because both operands of the division are themselves compile time constants) do not generate a division instruction and will not call the user defined hook in case of division by zero. This is consistent with the behavior on other targets and the behavior of software floating point.

Generally this will not be an issue for applications, but test code that deliberately wants to test division by zero should use the following trick:

```
volatile float zero=0;
return 1/zero;
```

1.7.10. Intelx86_Pentium-Specific Compiler Options

Table of Contents

- [-Xadd-underscore on page](#)
- [-Xdefer-pop on page](#)

-Xadd-underscore

Prefix function identifiers with underscore.

Belongs to:

[Instruction Index on page 252](#)

Synopsis

```
-Xadd-underscore  
-X34
```

Description

Prefix an underscore to function names only. Concatenation of underscore is useful when compiling libraries, to avoid using the same namespace as user programs.

-Xdefer-pop

Defer popping stack on function return.

Belongs to:

[Optimization Index on page 264](#)

Synopsis

```
-Xdefer-pop
```

Description

Turns on a code optimization that trades off stack space for code size. By default, the compiler will emit code that pops the callee's frame off the stack after it returns to the calling function. With this option, all these stack frame pops can be aggregated into a single pop and deferred until the end of the calling function. Deferred pops can be profitable in terms of code size when the C program consists of functions which in turn call a modest number of functions, and stack space is not too constrained.

2. ASSEMBLER OPTIONS

- [Assembler Basic Options on page 193](#)
- [Assembler X Options on page 198](#)
- [Assembler Architecture-Specific Options on page 207](#)

2.1. Assembler Basic Options

Table of Contents

- [-# on page](#)
- [-D on page](#)
- [-g on page](#)
- [-H on page](#)
- [-? on page](#)
- [-I path on page](#)
- [-I, -L on page](#)
- [-@, -@@ on page](#)
- [-@E, -@O on page](#)
- [-o file on page](#)
- [-R on page](#)
- [-T on page](#)
- [-V on page](#)
- [-WDDOBJECT on page](#)
- [-WDDTARGET on page](#)
- [-WD on page](#)
- [-x on page](#)
- [-X on page](#)

-#

Print command-line options on standard output.

Synopsis

```
-#
```

Description

The output of this option can be directed to a file. This can be convenient when contacting Technical Services. The **-#** should immediately follow the **das** command (after a space).

-D

Define symbol name.

Synopsis

```
-D name[=value]
```

Description

Define symbol name to have the given **value**. If value is not given, 1 is used. The **-D** option can be used to set symbols used with conditional assembly.

For more information, see the discussion of the **.if** expression in the *Assembler Directives* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

Note that assigning a string constant to a variable has no effect (see the discussion of string constants in the *Assembler Syntax Rules* chapter of the *Wind River Diab Compiler User's Guide* for your architecture).

-g

Generate debugging information.

Synopsis

```
-g
```

Description

Generate debug line and file information. (ELF/DWARF format only). Equivalent to **-Xasm-debug-on**.

-H

Include header in listing.

Synopsis

```
-H
```

Description

Print a header on the first line of each page of the assembly listing.

See Also

[-Xheader-format on page](#)
[-Xheader... on page](#)

-?

Show option summary.

Synopsis

```
-, -h,  
--help
```

Description

Show synopsis of command-line options.

-I path

Set header files directory.

Synopsis

```
-I path
```

Description

Specify a directory where the assembler will look for header files. May be given more than once.

For more information, see the discussion of the **.include** directive in the *Assembler Directives* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-l, -L

Generate listing file.

Synopsis

```
-l
-L
```

Description

A lowercase **-l** generates a listing file to **input-file.lst**. See **-Xlist-file-extension** to change the default extension.

An uppercase **-L** generates the listing file to standard output.

See the *Example Assembler Listing* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

See Also

[-Xlist-file-extension on page](#)

-@, -@@

Read command-line options from file or variable.

Synopsis

```
-@name
```

Description

Read command-line options from either a file or an environment variable. When **-@name** is encountered on the command line, the assembler first looks for an environment variable with the given name and substitutes its value. If an environment variable is not found then it tries to open a file with given name and substitutes the contents of the file. If neither an environment variable or a file can be found, an error message is issued and the assembler terminates.

```
-@@name
```

Same as **-@name**; also prints all command-line options on standard output.

-@E, -@O

Redirect output.

Synopsis

```
-@E=file
-@E+file
```


Description

Redirect any output to standard error to the given file.

`-@O=file -@O+file`

Redirect any output to standard output to the given file.

In both cases, use of **+** instead of **=** appends the output to the file.

-o file

Set output file.

Synopsis

```
-o file
```

Description

Write the object file to **file** instead of the default (**input-file.s**). Applies only to the first file if a list of files is presented; remaining files in the list use the default.

-R

Remove the input file on termination.

Synopsis

```
-R
```

Description

May be used by tools to remove temporary files.

-T

Specify assembler description (.ad) file.

Synopsis

```
-T ad-file
```

Description

Specify which assembler description (.ad) file to use. This is normally set automatically by using the **-t** option, defining the **DTARGET** and the **DOBJECT** environment variables, or using the **-WDDTARGET** and the **-WDDOBJECT** command-line options. It is primarily for internal use by Wind River.

-V

Print version number.

Synopsis

```
-V
```

Description

Display the version number of the assembler on standard output.

-WDDOBJECT

Select object format and mnemonic type.

Synopsis

```
-WDDOBJECT=object
```

Description

Specify the object format and mnemonic type. Overrides the environment variable **DOBJECT** if it is also set.

-WDDTARGET

Select target processor.

Synopsis

```
-WDDTARGET=target
```

Description

Specify the target processor. Overrides the environment variable **DTARGET** if it is also set.

-WD

Define configuration variable.

Synopsis

```
-WDname=value
```

Description

Set a configuration variable for use in the configuration files with the given **name** to the given **value**. Overrides an environment variable of the same name.

-X

Discard all local symbols.

Synopsis

```
-x
```

Description

Discard symbols not declared **.extern** or **.comm**.

-X

Discard all symbols starting with .L

Synopsis

```
-X
```

Description

Discard all symbols starting with **.L**; supports compilers using this form for automatically generated symbols, including the Wind River Diab Compiler.

2.2. Assembler X Options

Table of Contents

- [-Xalign-fill-text on page](#)
- [-Xalign-value, -Xalign-power2 on page](#)
- [-Xasm-debug... on page](#)
- [-Xauto-align on page](#)
- [-Xdefault-align on page](#)
- [-Xgen-shstrtab on page](#)
- [-Xgnu-locals-... on page](#)
- [-Xheader... on page](#)
- [-Xheader-format on page](#)
- [-Xlabel-colon... on page](#)
- [-Xline-format on page](#)
- [-Xlist-... on page](#)
- [-Xlist-file-extension on page](#)
- [-Xlit-marg... on page](#)
- [-Xllen on page](#)
- [-Xmacro-arg-space-... on page](#)
- [-Xobject-format on page](#)
- [-Xpage-skip on page](#)
- [-Xplen on page](#)
- [-Xrel-entry... on page](#)
- [-Xset-reorder, -Xset-noreorder on page](#)
- [-Xspace-... on page](#)
- [-Xstrip-locals..., -Xstrip-temps... on page](#)
- [-Xsubtitle on page](#)
- [-Xsymbol-debug on page](#)
- [-Xtab-size on page](#)
- [-Xtitle on page](#)

-Xalign-fill-text

Specify value to fill gaps left by **.align** or **.alignn** directive.

Synopsis

```
-Xalign-fill-text=n
```

Description

Fill gaps left by the **.align** or **.alignn** directive with the value **n**, overriding the processor-specific default.

-Xalign-value, -Xalign-power2

Interpret **.align** directive.

Synopsis

```
-Xalign-value
```

Description

Interpret the value in an **.align** directive as the value to which the location counter is to be aligned, which must be a power of 2.

Example: **-Xalign-value=8** means **.align** is to align on an 8-byte boundary. For ColdFire and x86 this is the default.

-Xalign-power2

Interpret the value in an **.align** directive as the power of 2 to which the location counter is to be aligned. Example: **-Xalign-power2=3** means **.align** is to align on an 8-byte boundary.

For ARM, MIPS, PowerPC, and TriCore this is the default.

-Xasm-debug...

Generate debugging information.

Synopsis

```
-Xasm-debug-off
```

Description

Do not generate debug line and file information. This is the default.

-Xasm-debug-on

Generate debug line and file information. (ELF/DWARF format only).

-Xauto-align

Align program data automatically based on size.

Synopsis

```
-Xauto-align-off
```

Description

The assembler performs no data alignment. This is the default.

-Xauto-align

Align program data automatically based on size.

-Xdefault-align

Set default value for section alignment.

Synopsis

```
-Xdefault-align=value
```

Description

Set the value used when calculating the default alignment for **.comm**, **.lcomm**, and **.sbss** directives, and the alignment used by the **.even** directive (for both ELF and COFF).

For COFF modules (which are supported for PowerPC), set the default alignment of a section not otherwise aligned. The section is padded so that the size becomes a multiple of the alignment value.

The default value of **-Xdefault-align** is 8 if no value is given.

Absent this directive, the default alignment for ELF sections is the maximum alignment of all objects in the section; the default alignment for COFF sections is 8.

Note that for ELF modules, **-Xdefault-align** does not set the alignment of sections — it sets the default for used by the **.comm**, **.lcomm**, **.sbss**, and **.even** directives. Only if one of these directives is in fact used in a section will the alignment be as set by **-Xdefault-align** rather than the maximum alignment of all objects in the section. See also **-Xcommon-align** in the *Linker User's Guide*.

-Xgen-shstrtab

Generate **.shstrtab** section as well as **.strtab** section.

Synopsis

```
-Xgen-shstrtab
```

Description

This option separates symbol names and section names, and generates both **.strtab** and **.shstrtab** sections. By default the assembler does not create **.shstrtab** sections; they are created by the linker. The **-Xgen-shstrtab** option is off by default to preserve the default toolchain behavior.

Since

5.9.4.0

-Xgnu-locals-...

Enable local GNU labels.

Synopsis

```
-Xgnu-locals-off
```

Description

Disable local GNU labels. The default setting is **-Xgnu-locals-on**.

-Xgnu-locals-on

Enable local GNU labels. This is the default.

See the discussion of GNU-style locals in the *Assembler Syntax Rules* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xheader...

Include header in listing.

Synopsis

```
-Xheader
```

Description

Include a header in the listing. See the **-I** and the **-L** options. This option is turned off as a default. This option has the same effect as the **-H** option.

-Xheader-off

Do not include a header in the listing file. This is the default.

See Also

[-Xheader-format on page](#)

[-H on page](#)

-Xheader-format

Set header format.

Synopsis

```
-Xheader-format="string"
```

Description

Define the format of the header in the assembly listing. (The header is enabled by options **-H** or **-Xheader** above). The header **string** can contain format specifications in any order introduced by a **"%"**. Characters not preceded by **"%"** are printed as is, including spaces and escapes such as **"\t"** for tab.

Valid format specifications are:

%nE

Use **n** columns to display the error count.

%nF

Use **n** columns to display the filename.

N

Start a new line.

%nP

Use **n** columns to display the page number.

%nS

Use **n** columns to display the subtitle given with the **-Xsubtitle** option.

%nT

Use **n** columns to display the title given with the **-Xtitle** option.

%nW

Use **n** columns to display the warning count.

The default header **string** is:

"%30T File: %10F Errors %4E"

See Also

[-H on page](#)

[-Xheader... on page](#)

-Xlabel-colon...

Set label definition syntax.

Synopsis

```
-Xlabel-colon
```

Description

Require that all label definitions have a colon ":" appended. For ARM, MIPS, PowerPC, RH850, and TriCore this is the default.

Note that this applies to all directives, including **.equ** and **.set**. For example, with the **-Xlabel-colon** option, the following is valid:

```
TRUE: .set 1
```

However, the following is invalid:

```
TRUE .set 1
```

With the **-Xlabel-colon** option, any directive that does not allow a label may start in column 1. These are directives that do not produce data in the current section, such as **.comm**, **.eject**, **.error**, **.exitm**, **.extern**, **.file**, and **.global**.

-Xlabel-colon-off

Do not require label definitions to end with a colon ":". For ColdFire and x86 this is the default.

With the **-Xlabel-colon-off** option, no directives may start in column 1.

-Xline-format

Set format of assembly line in listing.

Synopsis

```
-Xline-format="string"
```

Description

Define the format of each assembly line in a listing. The **string** can contain the following format specifications, in any order, starting with a "%". Characters not preceded by "%" are printed as is, including spaces and escapes such as "\t" for tab.

Valid format specifications are:

%nA

Use **n** columns to display current address.

%n.mC

Use **n** columns to display the generated code. A space is inserted at every **nth** column.

%nD

Display a maximum of **n** generated bytes for each source line. **n** may have a value from 1 through 32. More than one listing line might be used to display lines that produce many bytes.

%nL

Use **n** columns to display the current source line number.

%nP

Use **n** columns to display the current Program Location Counter (PLC) which corresponds to a section number. The assembly source statement follows the above items on the listing line. The default line format string is: "%8A %2P %32D%15.2C%5L\t"

See the *Example Assembler Listing* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xlist-...

Generate a listing file.

Synopsis

```
-Xlist-file
```

Description

Generate a listing file to file **input-file.lst**. Same as the **-l** option.

-Xlist-off

Generate no listing file. This is the default.

-Xlist-tty

Generate a listing file to standard output. Same as the **-L** option.

See the *Example Assembler Listing* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xlist-file-extension

Specify file extension for assembly listing.

Synopsis

```
-Xlist-file-extension="string"
```

Description

Use this option to override the default extension (**.lst**) of the listing file generated by **-l** or **-Xlist-file**. For example, **-Xlist-file-extension=".L"** specifies the file extension **.L**.

-Xlit-marg...

Set delay of literal generation.

Synopsis

```
-Xlit-marg-hard=value
```

Description

The assembler attempts to delay the generation of literals for as long as possible. This option, along with **-Xlit-marg-soft** and **-Xlit-marg-thresh**, controls that delay. The branch over the literals is only meant to be generated under unusual conditions. Use caution in invoking these options, as unpredictable results may result if the parameters are set to values beyond the capabilities of the target hardware.

Each literal has two addresses:

- **faddr**—the first address where the literal can be generated
- **laddr**—the last address where the literal can be generated

Forced generation of literals occurs when
 $\text{curPC} \geq \text{faddr} \ \&\& \ (\text{curPC} + \text{hard}) > \text{laddr}$

(**curPC** is the current PC.) Use **-Xlit-marg-hard** to define the forced threshold value **hard**. The default value is 16.

-Xlit-marg-soft=value

Define a soft (normal) threshold value. This threshold is used when the code contains an unconditional branch or return. The default value is 128.

Normal literal generation occurs when
 $\text{curPC} \geq \text{faddr} \ \&\& \ (\text{curPC} + \text{soft}) > \text{laddr}$

-Xlit-marg-thresh=value

Define a threshold value, that is, how long the thread should continue once literal generation has commenced. The default is 256.

Literal generation continues as long as


```
curPC >= faddr && (curPC + thresh) > laddr
```

-Xllen

Set line length of listing file.

Synopsis

```
-Xllen=n
```

Description

Define the number of printable character positions per line of the listing file. The default is 132 characters. A value of 0 means unlimited line length.

This value may also be set or changed by the **.llen** and **.psize** directives. See the discussion of these directives in the *Assembler Directives* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

See also the *Example Assembler Listing* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xmacro-arg-space-...

Enable blanks in macro arguments.

Synopsis

```
-Xmacro-arg-space-off
```

Description

Do not permit blanks in macro arguments. This is the default.

-Xmacro-arg-space-on

Permit blanks in macro arguments.

-Xobject-format

Set output object format.

Synopsis

```
-Xobject-format=form
```

Description

Set the object format the assembler should produce. **form** is one of:

coff

COFF (Common Object File Format)

elf

ELF (Executable and Linkable Format)

The object format is set automatically by the **DOBJECT** environment variable and the **-WDDOBJECT** option and should not be set explicitly.

-Xpage-skip

Set page break margin.

Synopsis

```
-Xpage-skip=n
```

Description

If **n** is zero (the default), page breaks in the listing file will be created using formfeed (ASCII 12). Otherwise each page will be padded with **n** blank lines, and these **n** blank lines included in the count set by **-Xplen** option.

See the *Example Assembler Listing* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

See Also

[-Xplen on page](#)

-Xplen

Set lines per page.

Synopsis

```
-Xplen=n
```

Description

Define the number of printable lines per page in the listing file. The default value of **n** is 60. See also **-Xpage-skip** above.

This value may also be set or changed by the **.lcnt** and **.psize** directives. See the discussion of these directives in the *Assembler Directives* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

See also the *Example Assembler Listing* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xrel-entry...

Specify type of relocation entry.

Synopsis

```
-Xrel-entry-default
```

Description

Generate relocation entries of the default type for the target architecture and ABI. (For x86, this is **REL**.)

-Xrel-entry-rela

Generate relocation entries of type RELA.

-Xrel-entry-rel

Generate relocation entries of type REL.

-Xset-reorder, -Xset-noreorder

Fill delay slots with NOPs.

Synopsis

```
-Xset-reorder
```

Description

The assembler automatically inserts a NOP instruction to fill any delay slot. This is the default.

-Xset-noreorder

NOP instructions are not inserted in delay slots. Equivalent to **.set noreorder**.

If **.set reorder** or **.set noreorder** appears in the assembly source, the **.set** directive takes precedence over the **-Xset-...** option. See the discussion of the **.set** option in the *Assembler Directives* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

These options do not affect compiler-generated code or embedded assembly code that is processed by the compiler.

-Xspace-...

Enable spaces between operands.

Synopsis

```
-Xspace-off
```

Description

Do not allow spaces between operands in an assembly instruction.

-Xspace-on

Allow spaces between operands in an assembly instruction. This is the default.

-Xstrip-locals..., -Xstrip-temps...

Delete local symbols.

Synopsis

```
-Xstrip-locals
```

Description

-Xstrip-locals-off

Include local symbols in the symbol table. This is the default.

-Xstrip-temps="string"

Do not include local labels starting with **string** in the symbol table. If no **string** is specified, **.L** will be used. This is the same as the **-X** option. This option can be used to suppress the temporary symbols generated by the compiler.

-Xstrip-temps-off

Include local symbols starting with **.L** in the symbol table. This is the default.

-Xsubtitle

Set subtitle.

Synopsis

```
-Xsubtitle="string"
```

Description

Define a subtitle that will be printed in the **S** field of the header. See **-Xheader-format**, for more information.

See Also

[-Xheader-format on page](#)

-Xsymbol-debug

Belongs to:

[Debugging Index on page 250](#)

Synopsis

```
-Xsymbol-debug
```

Description

Generate basic debug information for each symbol defined in an assembly source file.

See Also

[-Xhll-symbol on page](#)

-Xtab-size

Set tab size.

Synopsis

```
-Xtab-size=n
```

Description

Define the number of spaces between tab stops. The default is 8.

-Xtitle

Set title.

Synopsis

```
-Xtitle="string"
```

Description

Define a title that will be printed in the **T** field of the header. See **-Xheader-format**, for more information.

See Also

[-Xheader-format on page](#)

2.3. Assembler Architecture-Specific Options

- [ARM-Specific Assembler Options on page 208](#)
- [ColdFire-Specific Assembler Options on page 210](#)
- [MIPS-Specific Assembler Options on page 216](#)

- [RH850-Specific Assembler Options on page 219](#)
- [PowerPC-Specific Assembler Options on page 221](#)
- [TriCore-PCP-Specific Assembler Options on page 225](#)
- [TriCore-Specific Assembler Options on page 226](#)
- [Intelx86_Pentium-Specific Assembler Options on page 227](#)

2.3.1. ARM-Specific Assembler Options

Table of Contents

- [-Xcpu-... on page](#)
- [-Xdebug-dwarf1 on page](#)
- [-Xdebug-dwarf2 on page](#)
- [-Xdebug-dwarf3 on page](#)
- [-Xerrata-imx on page](#)
- [-Xoptim-... on page](#)
- [-Xsemi-is-newline on page](#)

-Xcpu-...

Set instruction type.

Synopsis

```
-Xcpu-target
```

Description

Accept instructions only for the target processor designated by **target**. This option is primarily for internal use and is set automatically by the driver in response to the user-level **-ttof:environ** option.

-Xcpu-TX49xx enables support for TX49 processors.

-Xdebug-dwarf1

Select DWARF 1 format.

Synopsis

```
-Xdebug-dwarf1
```

Description

Generate DWARF 1.1 debug information.

See Also

- [-Xdebug-dwarf2 on page](#)
- [-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf2

Select DWARF 2 format.

Synopsis

```
-Xdebug-dwarf2
```

Description

Generate DWARF 2 debug information. This is the default.

See Also

[-Xdebug-dwarf1 on page](#)
[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf3

Select DWARF 3 format.

Synopsis

```
-Xdebug-dwarf3
```

Description

Generate DWARF 3 debug information.

See Also

[-Xdebug-dwarf1 on page](#)
[-Xdebug-dwarf2 on page](#)

-Xerrata-imx

Use workaround for LDM instruction error.

Synopsis

```
-Xerrata-imx
```

Description

This feature applies to the ARM iMX, iMXS, and iMXL chips (from Freescale Semiconductor) with regard to the **LDM** instruction.

When an **LDM** instruction is used to load two specific registers from an un-cached region of memory, and the load instruction begins immediately after a delayed buffered write, the **LDM** will not load the second register correctly.

The **-Xerrata-imx** option to the assembler (**das**) converts **LDM** instructions, taking two registers into two **LDR** instructions. All **LDM** addressing modes are handled.

When used with **dcc** or **dplus**, this option must be specified as **-Wa,-Xerrata-imx**.

-Xoptim-...

Select branch size optimizations.

Synopsis

```
-Xoptim-all
```

Description

Enable branch size optimizations; choose the shortest branch instruction for local branches.

This is the default.

`-Xoptim-off`

`-Xno-optim`

Disable branch size optimizations.

-Xsemi-is-newline

Treat semicolons as statement separators.

Synopsis

```
-Xsemi-is-newline
```

Description

Treat the semicolon (;) as a statement separator instead of a comment character. This is useful for GNU compatibility.

2.3.2. ColdFire-Specific Assembler Options**Table of Contents**

- [-Xabs-ind-long, -Xabs-ind-word on page](#)
- [-Xauto-comment... on page](#)
- [-Xbra-is-rel, -Xbra-is-jra on page](#)
- [-Xcf4-d7-errata on page](#)
- [-Xcmp-normal, -Xcmp-swap on page](#)
- [-Xcpu... on page](#)
- [-Xdebug-dwarf1 on page](#)
- [-Xdebug-dwarf2 on page](#)
- [-Xdebug-dwarf3 on page](#)
- [-Xdisp... on page](#)
- [-Xfpu... on page](#)
- [-Xgen-coldfire-header on page](#)
- [-Xind16-range-off, -Xind16-range-on on page](#)
- [-Xindex-long, -Xindex-word on page](#)
- [-Xmmu-68xxx, -Xmmu-all, -Xmmu-none on page](#)
- [-Xmnem-all, -Xmnem-emb, -Xmnem-mit, -Xmnem-moto on page](#)
- [-Xoptim... on page](#)
- [-Xsemi-is-newline on page](#)

-Xabs-ind-long, -Xabs-ind-word

Select default absolute address mode.

Synopsis

```
-Xabs-ind-long
```

Description

Use "Absolute Long Addressing Mode" as the default when an address is not qualified with a size. This is the default.

`-Xabs-ind-word`

Use "Absolute Word Addressing Mode" as the default when an address is not qualified with a size.

Example of **-Xabs-ind-long**:

```
move.l (0x1234),d0    -> move.l (0x1234:w),d0
move.l (0x12345),d0   -> move.l (0x12345:l),d0
move.l (glob),d0      -> move.l (globa:l),d0
```

Example of **-Xabs-ind-word**:

```
move.l (0x1234),d0    -> move.l (0x1234:w),d0
move.l (0x12345),d0   -> move.l (0x12345:l),d0
move.l (glob),d0      -> move.l (globa:w),d0
```

-Xauto-comment...

Allow comments without comment character.

Synopsis

```
-Xauto-comment-on
```

Description

Allow comments ending an instruction or directive line to begin without a leading comment character. This option implicitly sets **-Xspace-off** which disallows spaces in operands. It also prohibits more than one statement per line.

-Xauto-comment-off

Require a comment ending an instruction or directive line to begin with a comment character. This is the default.

-Xbra-is-rel, -Xbra-is-jra

Select bra instruction.

Synopsis

```
-Xbra-is-rel
```

Description

The **bra** instruction without a size indicator must be PC-relative.

-Xbra-is-jra

The **bra** instruction without a size indicator can be converted to an absolute **jmp** if target is > 32K away.

-Xcf4-d7-errata

Fix ColdFire Version 4 hardware bug.

Synopsis

```
-Xcf4-d7-errata
```

Description

Activate a workaround for a ColdFire Version 4 hardware bug, replacing broken instructions that access the **d7** register.

-Xcmp-normal, -Xcmp-swap

Select 68K cmp instruction.

Synopsis

```
-Xcmp-normal
```


Description

The 68K **cmp** instruction behaves as described in the 68000 user's manual. This is the default.

-Xcmp-swap

The 68K **cmp** instruction has its operands swapped. This is the way many UNIX 68K assemblers work.

-Xcpu-...

Set instruction type.

Synopsis

```
-Xcpu-target
```

Description

Accept instructions only for the target processor designated by **target**. This option is primarily for internal use and is set automatically by the driver in response to the user-level **-ttof:environ** option.

-Xcpu-TX49xx enables support for TX49 processors.

-Xdebug-dwarf1

Select DWARF 1 format.

Synopsis

```
-Xdebug-dwarf1
```

Description

Generate DWARF 1.1 debug information.

See Also

[-Xdebug-dwarf2 on page](#)

[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf2

Select DWARF 2 format.

Synopsis

```
-Xdebug-dwarf2
```

Description

Generate DWARF 2 debug information. This is the default.

See Also

[-Xdebug-dwarf1 on page](#)

[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf3

Select DWARF 3 format.

Synopsis

```
-Xdebug-dwarf3
```

Description

Generate DWARF 3 debug information.

See Also

[-Xdebug-dwarf1](#) on page

[-Xdebug-dwarf2](#) on page

-Xdisp...

Select default displacement size.

Synopsis

```
-Xdisp-long
```

Description

Use "long" as the displacement size of displacements without an explicit size qualification. This is the default.

-Xdisp-word

Use "word" as the displacement size of displacements without an explicit size qualification.

This option interacts with options **-Xind16-range-...** as shown by the following examples.

Example of **-Xdisp-long** with **-Xind16-range-off**:

```
move.l 0x1234(a0),d0 -> move.l 0x1234:w(a0),d0
move.l 0x12345(a0),d0 -> move.l 0x12345:w(a0),d0
move.l glob(a0),d0 -> move.l glob:w(a0),d0
move.l 0x1234([a0]),d0 -> move.l 0x1234:w([a0]),d0
move.l 0x12345([a0]),d0 -> move.l 0x12345:l([a0]),d0
move.l glob([a0]),d0 -> move.l glob:l([a0]),d0
```

with **-Xind16-range-on**:

```
move.l 0x1234(a0),d0 -> move.l 0x1234:w(a0),d0
move.l 0x12345(a0),d0 -> move.l 0x12345:l(a0),d0
move.l glob(a0),d0 -> move.l glob:l(a0),d0
move.l 0x1234([a0]),d0 -> move.l 0x1234:w([a0]),d0
move.l 0x12345([a0]),d0 -> move.l 0x12345:l([a0]),d0
move.l glob([a0]),d0 -> move.l glob:l([a0]),d0
```

Example of **-Xdisp-word** with **-Xind16-range-off**:

```
move.l 0x1234(a0),d0 -> move.l 0x1234:w(a0),d0
move.l 0x12345(a0),d0 -> move.l 0x12345:w(a0),d0
move.l glob(a0),d0 -> move.l glob:w(a0),d0
move.l 0x1234([a0]),d0 -> move.l 0x1234:w([a0]),d0
move.l 0x12345([a0]),d0 -> move.l 0x12345:l([a0]),d0
move.l glob([a0]),d0 -> move.l glob:w([a0]),d0
```

with **-Xind16-range-on**:

```
move.l 0x1234(a0),d0 -> move.l 0x1234:w(a0),d0
move.l 0x12345(a0),d0 -> move.l 0x12345:l(a0),d0
move.l glob(a0),d0 -> move.l glob:w(a0),d0
move.l 0x1234([a0]),d0 -> move.l 0x1234:w([a0]),d0
move.l 0x12345([a0]),d0 -> move.l 0x12345:l([a0]),d0
move.l glob([a0]),d0 -> move.l glob:w([a0]),d0
```

-Xfpu...

Set FPU type.

Synopsis

```
-Xfpu-68881
```

Description

Only accept code for 68881 FPU.

-Xfpu-68882

Only accept code for 68882 FPU.

-Xfpu-68040

Only accept code for 68040 FPU.

-Xfpu-68060

Only accept code for 68060 FPU.

-Xfpu-all

Accept code for all 68K FPUs.

-Xfpu-none

Do not accept code for any FPU.

-Xgen-coldfire-header

Generate EM_COLDFIRE ELF header for output object file.

Synopsis

```
-Xgen-coldfire-header
```

Description

-Xind16-range-off, -Xind16-range-on

Select default address register indirect mode.

Synopsis

```
-Xind16-range-off
```

Description

Use "Address Register Indirect with Displacement" when possible and truncate the displacement to 16 bits regardless of the **-Xdisp-long** and **-Xdisp-word** options. This is the default.

-Xind16-range-on

Use "Address Register Indirect with Displacement" only when the displacement is known to fit in 16 bits or when **-Xdisp-word** is set.

These options interact with the **-Xdisp-...** options.

See Also

[-Xdisp... on page](#)

-Xindex-long, -Xindex-word

Select default index size.

Synopsis

```
-Xindex-long
```

Description

Use "long" as the index size for references to index registers without an explicit index size qualifier. This is the default.

-Xindex-word

Use "word" as the index size for references to index registers without an explicit index size qualifier.

Example of **-Xindex-long**:

```
move.l (a0,d0),d0    -> move.l (a0,d0:l),d0
```

Example of **-Xindex-word**:

```
move.l (a0,d0),d0    -> move.l (a0,d0:w),d0
```

-Xmmu-68xxx, -Xmmu-all, -Xmmu-none

Set MMU type.

Synopsis

```
-Xmmu-68851
```

Description

Only accept code for the 68851 MMU.

-Xmmu-68030

Only accept code for the 68030 MMU.

-Xmmu-68040

Only accept code for the 68040 MMU.

-Xmmu-68060

Only accept code for the 68060 MMU.

-Xmmu-all

Accept code for the all 68K MMUs.

-Xmmu-none

Do not accept code for any MMU.

-Xmnem-all, -Xmnem-emb, -Xmnem-mit, -Xmnem-moto

Set mnemonics type.

Synopsis

```
-Xmnem-all
```

Description

Accept all of the mnemonics specified by the other **-Xmnem-x** options.

-Xmnem-emb

Accept only Motorola (Freescale) Embedded mnemonics.

-Xmnem-mit

Accept only MIT mnemonics.

-Xmnem-moto

Accept only Motorola (Freescale) UNIX mnemonics.

-Xoptim-...

Select branch size optimizations.

Synopsis

```
-Xoptim-all
```

Description

Enable branch size optimizations; choose the shortest branch instruction for local branches. For ColdFire, also change **move** to **moveq**.

This is the default.

-Xoptim-off

-Xno-optim

Disable branch size optimizations.

-Xsemi-is-newline

Treat semicolons as statement separators.

Synopsis

```
-Xsemi-is-newline
```

Description

Treat the semicolon (;) as a statement separator instead of a comment character. This is useful for GNU compatibility.

2.3.3. MIPS-Specific Assembler Options

Table of Contents

- [-Xase-dsp on page](#)
- [-Xase-mt on page](#)
- [-Xcpu-... on page](#)
- [-Xcpu-mips32v2 on page](#)
- [-Xdebug-dwarf1 on page](#)
- [-Xdebug-dwarf2 on page](#)
- [-Xdebug-dwarf3 on page](#)
- [-Xfix-vr4133 on page](#)
- [-Xsemi-is-newline on page](#)
- [-Xwarn-use-greg on page](#)

-Xase-dsp

Enable digital signal processing.

Synopsis

```
-Xase-dsp
```

Description

Enable Digital Signal Processing (DSP) assembler instructions.

-Xase-mt

Enable multi-threading instruction support.

Synopsis

```
-Xase-mt
```

Description

Enable assembler multithreading instructions.

-Xcpu-...

Set instruction type.

Synopsis

```
-Xcpu-target
```

Description

Accept instructions only for the target processor designated by **target**. This option is primarily for internal use and is set automatically by the driver in response to the user-level **-ttof:environ** option.

-Xcpu-TX49xx enables support for TX49 processors.

-Xcpu-mips32v2

Enable MIPS32 Version 2 instructions.

Synopsis

```
-Xcpu-mips32v2
```

Description

Enable MIPS32 Version 2 assembler instructions.

-Xdebug-dwarf1

Select DWARF 1 format.

Synopsis

```
-Xdebug-dwarf1
```

Description

Generate DWARF 1.1 debug information.

See Also

[-Xdebug-dwarf2 on page](#)
[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf2

Select DWARF 2 format.

Synopsis

```
-Xdebug-dwarf2
```

Description

Generate DWARF 2 debug information. This is the default.

See Also

[-Xdebug-dwarf1 on page](#)
[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf3

Select DWARF 3 format.

Synopsis

```
-Xdebug-dwarf3
```

Description

Generate DWARF 3 debug information.

See Also

[-Xdebug-dwarf1 on page](#)
[-Xdebug-dwarf2 on page](#)

-Xfix-vr4133

Work around VR4133 bug.

Synopsis

```
-Xfix-vr4133
```

Description

Use with the **-mfix-vr4133** driver option (see **-Xmfix-vr4133**) to implement a workaround for a problem that arises with NEC VR4133 processors when **mult/div/macc** instructions follow **mflo/mfhi** instructions.

See Also

[-mfix-vr4133 on page](#)

-Xsemi-is-newline

Treat semicolons as statement separators.

Synopsis

```
-Xsemi-is-newline
```

Description

Treat the semicolon (;) as a statement separator instead of a comment character. This is useful for GNU compatibility.

-Xwarn-use-greg

Warn when specified registers are accessed.

Synopsis

```
-Xwarn-use-greg=mask
```

Description

Causes the assembler to issue a warning when a register indicated by a bit in **mask** is used. (Bit 0 corresponds to **r0** or **\$0**, bit 1 to **r1** or **\$1**, and so on.) This is helpful for detecting accesses to registers that must be preserved, such as **r2** and **r13**, in hand-written assembly code.

2.3.4. RH850-Specific Assembler Options

Table of Contents

- [-Xcpu-... on page](#)
- [-Xdebug-dwarf1 on page](#)
- [-Xdebug-dwarf2 on page](#)
- [-Xdebug-dwarf3 on page](#)
- [-Xexp on page](#)
- [-Xnoexp on page](#)

-Xcpu-...

Set instruction type.

Synopsis

```
-Xcpu-target
```

Description

Accept instructions only for the target processor designated by **target**. This option is primarily for internal use and is set automatically by the driver in response to the user-level **-ttof:environ** option.

-Xcpu-TX49xx enables support for TX49 processors.

-Xdebug-dwarf1

Select DWARF 1 format.

Synopsis

```
-Xdebug-dwarf1
```


Description

Generate DWARF 1.1 debug information.

See Also

[-Xdebug-dwarf2 on page](#)

[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf2

Select DWARF 2 format.

Synopsis

```
-Xdebug-dwarf2
```

Description

Generate DWARF 2 debug information. This is the default.

See Also

[-Xdebug-dwarf1 on page](#)

[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf3

Select DWARF 3 format.

Synopsis

```
-Xdebug-dwarf3
```

Description

Generate DWARF 3 debug information.

See Also

[-Xdebug-dwarf1 on page](#)

[-Xdebug-dwarf2 on page](#)

-Xexp

-Xexp option.

Synopsis

```
-Xexp
```

Description

NOTES

Enable macro expansion of load, store, branch, and jump operations so to allow 32-bit operations. Default is expansion.

See Also

[-Xnoexp on page](#)

-Xnoexp

-Xnoexp option.

Synopsis

```
-Xnoexp
```

Description

NOTES

Disable macro expansion of load, store, branch, and jump operations to allow 32-bit operations. Default is expansion.

See Also

[-Xexp on page](#)

2.3.5. PowerPC-Specific Assembler Options

Table of Contents

- [-Xapu-sie on page](#)
- [-Xapu-volatile-context on page](#)
- [-Xapu-wait on page](#)
- [-Xcpu-... on page](#)
- [-Xdebug-dwarf1 on page](#)
- [-Xdebug-dwarf2 on page](#)
- [-Xdebug-dwarf3 on page](#)
- [-Xfix-bc-fw-mask-bug on page](#)
- [-Xisa-vle on page](#)
- [-Xoptim-... on page](#)
- [-Xprepare-compress on page](#)
- [-Xsemi-is-newline on page](#)
- [-Xvle-opt-off on page](#)
- [-Xwarn-use-greg on page](#)

-Xapu-sie

Enable assembler to support saturation instruction extension (SIE).

Synopsis

```
-Xapu-sie
```

Description

Enable saturation instructions extension (SIE) support in the assembler, including scalar saturating arithmetic, add and subtract for both **signed** and **unsignedword**, **halfword**, and **byte** data types.

-Xapu-volatile-context

Save/restore volatile context.

Synopsis

```
-Xapu-volatile-context
```

Description

Implement the "Volatile Context Save/Restore APU" instructions for VLE. These instructions save code space in interrupt handlers. These instructions allow a multiple register load or store to a word aligned memory address. The compiler generates them for `__interrupt__` functions when this option is given; the assembler accepts the instructions when this option is given.

-Xapu-wait

Stop executing until interrupt.

Synopsis

```
-Xapu-wait
```

Description

Implement the **wait** (i.e., 'Wait APU') instruction for VLE- and e500-based processors. This instruction causes the CPU to stop executing instructions until an interrupt occurs.

Also implemented as the intrinsic function `__wait(void)`. See the discussion of intrinsic functions in the *Additions to ANSI C and C++* chapter of the *Wind River Diab Compiler User's Guide* for your architecture.

-Xcpu-...

Set instruction type.

Synopsis

```
-Xcpu-target
```

Description

Accept instructions only for the target processor designated by **target**. This option is primarily for internal use and is set automatically by the driver in response to the user-level **-ttof:environ** option.

-Xcpu-TX49xx enables support for TX49 processors.

-Xdebug-dwarf1

Select DWARF 1 format.

Synopsis

```
-Xdebug-dwarf1
```

Description

Generate DWARF 1.1 debug information.

See Also

[-Xdebug-dwarf2 on page](#)
[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf2

Select DWARF 2 format.

Synopsis

```
-Xdebug-dwarf2
```

Description

Generate DWARF 2 debug information. This is the default.

See Also

[-Xdebug-dwarf1 on page](#)
[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf3

Select DWARF 3 format.

Synopsis

```
-Xdebug-dwarf3
```

Description

Generate DWARF 3 debug information.

See Also

[-Xdebug-dwarf1 on page](#)
[-Xdebug-dwarf2 on page](#)

-Xfix-bc-fw-mask-bug

Generate NOP for certain branch conditional instructions.

Synopsis

```
-Xfix-bc-fw-mask-bug
```

Description

If a **BC** instruction branches to a forward label, generate a nop instruction after the label. This option is provided as a work-around for a bug present in 800-family PowerPC processors as of this writing (mid-1998).

-Xisa-vle

Assemble using VLE instructions.

Synopsis

```
-Xisa-vle
```

Description

Expect and assemble VLE (Variable Length Encoding) instructions rather than Book E instructions. The default code section is named **.text_vle** instead of **.text**, and the default code section fill "character" is set to 0x44444444 instead of 0. The **.text_vle** code section will have ELF section header flags marking it as VLE code, not Book E code.

-Xoptim-...

Select branch size optimizations.

Synopsis

```
-Xoptim-all
```

Description

Enable branch size optimizations; choose the shortest branch instruction for local branches.

This is the default.

-Xoptim-off

-Xno-optim

Disable branch size optimizations.

-Xprepare-compress

Limit length of conditional branch.

Synopsis

```
-Xprepare-compress=n
```

Description

Change the maximum length of a conditional branch from the default, which is 32,766 bytes; if **n** is not specified, the length is set to 1024. If a conditional branch exceeds this limit, the assembler inserts a reverse conditional around an unconditional branch to the label.

-Xsemi-is-newline

Treat semicolons as statement separators.

Synopsis

```
-Xsemi-is-newline
```

Description

Treat the semicolon (;) as a statement separator instead of a comment character. This is useful for GNU compatibility.

-Xvle-opt-off

Disable automatic instruction optimization from BookE to VLE and from long VLE to short VLE.

Synopsis

```
-Xvle-opt-off
```

Description

In VLE instruction mode, by default the assembler tries to convert BookE instructions to equivalent VLE ones, and to convert Long VLE instructions to the equivalent short format VLE ones. This option disables the default behavior.

-Xwarn-use-greg

Warn when specified registers are accessed.

Synopsis

```
-Xwarn-use-greg=mask
```

Description

Causes the assembler to issue a warning when a register indicated by a bit in **mask** is used. (Bit 0 corresponds to **r0** or **\$0**, bit 1 to **r1** or **\$1**, and so on.) This is helpful for detecting accesses to registers that must be preserved, such as **r2** and **r13**, in hand-written assembly code.

2.3.6. TriCore-PCP-Specific Assembler Options

Table of Contents

- [-Xdebug-dwarf1 on page](#)
- [-Xdebug-dwarf2 on page](#)
- [-Xdebug-dwarf3 on page](#)
- [-Xoptim-... on page](#)

-Xdebug-dwarf1

Select DWARF 1 format.

Synopsis

```
-Xdebug-dwarf1
```

Description

Generate DWARF 1.1 debug information.

See Also

- [-Xdebug-dwarf2 on page](#)
- [-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf2

Select DWARF 2 format.

Synopsis

```
-Xdebug-dwarf2
```

Description

Generate DWARF 2 debug information. This is the default.

See Also

- [-Xdebug-dwarf1 on page](#)
- [-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf3

Select DWARF 3 format.

Synopsis

```
-Xdebug-dwarf3
```

Description

Generate DWARF 3 debug information.

See Also

[-Xdebug-dwarf1 on page](#)
[-Xdebug-dwarf2 on page](#)

-Xoptim-...

Select branch size optimizations.

Synopsis

```
-Xoptim-all
```

Description

Enable branch size optimizations; choose the shortest branch instruction for local branches.

This is the default.

-Xoptim-off

-Xno-optim

Disable branch size optimizations.

2.3.7. TriCore-Specific Assembler Options

Table of Contents

- [-Xcpu-... on page](#)
- [-Xenable-fpu on page](#)
- [-Xenable-mmu on page](#)

-Xcpu-...

Set instruction type.

Synopsis

```
-Xcpu-target
```

Description

Accept instructions only for the target processor designated by **target**. This option is primarily for internal use and is set automatically by the driver in response to the user-level **-ttof:environ** option.

-Xcpu-TX49xx enables support for TX49 processors.

-Xenable-fpu

Enable parsing of FPU instructions.

Synopsis

```
-Xenable-fpu
```

Description

Enable parsing of the instructions for the Floating Point Unit.

-Xenable-mmu

Enable parsing of MMU instructions.

Synopsis

```
-Xenable-mmu
```

Description

Enable parsing of instructions for the Memory Management Unit.

2.3.8. Intelx86_Pentium-Specific Assembler Options

Table of Contents

- [-Xauto-comment... on page](#)
- [-Xcmp-normal, -Xcmp-swap on page](#)
- [-Xcpu-... on page](#)
- [-Xdebug-dwarf1 on page](#)
- [-Xdebug-dwarf2 on page](#)
- [-Xdebug-dwarf3 on page](#)
- [-Xemul-gnu-bug on page](#)
- [-Xmnem-mit, -Xmnem-intel on page](#)
- [-Xoptim-... on page](#)
- [-Xsemi-is-newline on page](#)

-Xauto-comment...

Allow comments without comment character.

Synopsis

```
-Xauto-comment-on
```

Description

Allow comments ending an instruction or directive line to begin without a leading comment character. This option implicitly sets **-Xspace-off** which disallows spaces in operands. It also prohibits more than one statement per line.

-Xauto-comment-off

Require a comment ending an instruction or directive line to begin with a comment character. This is the default.

-Xcmp-normal, -Xcmp-swap

Select 68K cmp instruction.

Synopsis

```
-Xcmp-normal
```

Description

The 68K **cmp** instruction behaves as described in the 68000 user's manual. This is the default.

-Xcmp-swap

The 68K **cmp** instruction has its operands swapped. This is the way many UNIX 68K assemblers work.

-Xcpu-...

Set instruction type.

Synopsis

```
-Xcpu-target
```

Description

Accept instructions only for the target processor designated by **target**. This option is primarily for internal use and is set automatically by the driver in response to the user-level **-ttof:environ** option.

-Xcpu-TX49xx enables support for TX49 processors.

-Xdebug-dwarf1

Select DWARF 1 format.

Synopsis

```
-Xdebug-dwarf1
```

Description

Generate DWARF 1.1 debug information.

See Also

[-Xdebug-dwarf2 on page](#)

[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf2

Select DWARF 2 format.

Synopsis

```
-Xdebug-dwarf2
```

Description

Generate DWARF 2 debug information. This is the default.

See Also

[-Xdebug-dwarf1 on page](#)

[-Xdebug-dwarf3 on page](#)

-Xdebug-dwarf3

Select DWARF 3 format.

Synopsis

```
-Xdebug-dwarf3
```

Description

Generate DWARF 3 debug information.

See Also

[-Xdebug-dwarf1 on page](#)

[-Xdebug-dwarf2 on page](#)

-Xemul-gnu-bug

Emulate GNU Assembler's encoding of `fdivp`, `fdivrp`, `fsubp`, and `fsubrp`.

Synopsis

```
-Xemul-gnu-bug
```

Description

Causes the Wind River Assembler to emulate a known behavior in the GNU assembler's encoding of **`fdivp`**, **`fdivrp`**, **`fsubp`**, and **`fsubrp`** instructions. This option should be used only when assembly code produced by or for the GNU toolchain is assembled with the Wind River Assembler; it is required for certain double-precision floating point routines. If the assembler is invoked using the driver program (**`dcc`** or **`dplus`**), **`-Xemul-gnu-bug`** should be preceded by **`-Wa`** so that it is passed to the assembler.

For more information, see the *VxWorks Architecture Supplement*.

-Xmnem-mit, -Xmnem-intel

Set mnemonics type.

Synopsis

```
-Xmnem-mit
```

Description

Accept only MIT mnemonics. This is the default.

`-Xmnem-intel`

Accept only Intel mnemonics.

-Xoptim-...

Select branch size optimizations.

Synopsis

```
-Xoptim-all
```

Description

Enable branch size optimizations; choose the shortest branch instruction for local branches.

This is the default.

-Xoptim-off

-Xno-optim

Disable branch size optimizations.

-Xsemi-is-newline

Treat semicolons as statement separators.

Synopsis

```
-Xsemi-is-newline
```

Description

Treat the semicolon (;) as a statement separator instead of a comment character. This is useful for GNU compatibility.

3. CATEGORY AND NUMERIC CODE INDICES

- [Alphabetical Assembler Option Index on page 231](#)
- [Alphabetical Compiler Option Index on page 235](#)
- [C Plus Plus Index on page 248](#)
- [Checking and Profiling Index on page 249](#)
- [Compiler Command-Line Options Index on page 250](#)
- [Debugging Index on page 250](#)
- [Diagnostic and Lint Index on page 251](#)
- [Driver Index on page 252](#)
- [Instruction Index on page 252](#)
- [Memory Index on page 255](#)
- [Numerical Compiler Option Index on page 256](#)
- [Optimization Index on page 264](#)
- [Output Index on page 266](#)
- [Sections Index on page 267](#)
- [Syntax Index on page 268](#)
- [Type Index on page 269](#)

3.1. Alphabetical Assembler Option Index

- ? (all) [on page](#) Show option summary.
- @, -@@ (all) [on page](#) Read command-line options from file or variable.
- # (all) [on page](#) Print command-line options on standard output.
- D (all) [on page](#) Define symbol name.
- @E, -@O (all) [on page](#) Redirect output.
- g (all) [on page](#) Generate debugging information.
- H (all) [on page](#) Include header in listing.
- I path (all) [on page](#) Set header files directory.
- l, -L (all) [on page](#) Generate listing file.
- o file (all) [on page](#) Set output file.
- R (all) [on page](#) Remove the input file on termination.

- T (all) on page Specify assembler description (.ad) file.
- V (all) on page Print version number.
- WD (all) on page Define configuration variable.
- WDDOBJECT (all) on page Select object format and mnemonic type.
- WDDTARGET (all) on page Select target processor.
- Xabs-ind-long, -Xabs-ind-word (ColdFire) on page Select default absolute address mode.
- Xalign-fill-text (all) on page Specify value to fill gaps left by .align or .alignn directive.
- Xalign-value, -Xalign-power2 (all) on page Interpret .align directive.
- x (all) on page Discard all local symbols.
- X (all) on page Discard all symbols starting with .L
- Xapu-sie (PowerPC) on page Enable assembler to support saturation instruction extension (SIE).
- Xapu-volatile-context (PowerPC) on page Save/restore volatile context.
- Xapu-wait (PowerPC) on page Stop executing until interrupt.
- Xase-dsp (MIPS) on page Enable digital signal processing.
- Xase-mt (MIPS) on page Enable multi-threading instruction support.
- Xasm-debug... (all) on page Generate debugging information.
- Xauto-align (all) on page Align program data automatically based on size.
- Xauto-comment... (ColdFire) on page Allow comments without comment character.
- Xauto-comment... (Intelx86_Pentium) on page Allow comments without comment character.
- Xbra-is-rel, -Xbra-is-jra (ColdFire) on page Select bra instruction.
- Xcf4-d7-errata (ColdFire) on page Fix ColdFire Version 4 hardware bug.
- Xcmp-normal, -Xcmp-swap (ColdFire) on page Select 68K cmp instruction.
- Xcmp-normal, -Xcmp-swap (Intelx86_Pentium) on page Select 68K cmp instruction.
- Xcpu-... (ARM) on page Set instruction type.
- Xcpu-... (ColdFire) on page Set instruction type.
- Xcpu-... (Intelx86_Pentium) on page Set instruction type.
- Xcpu-... (MIPS) on page Set instruction type.
- Xcpu-mips32v2 (MIPS) on page Enable MIPS32 Version 2 instructions.
- Xcpu-... (PowerPC) on page Set instruction type.
- Xcpu-... (RH850) on page Set instruction type.

- Xcpu-... (TriCore) [on page](#) Set instruction type.
- Xdebug-dwarf1 (ARM) [on page](#) Select DWARF 1 format.
- Xdebug-dwarf1 (ColdFire) [on page](#) Select DWARF 1 format.
- Xdebug-dwarf1 (Intelx86_Pentium) [on page](#) Select DWARF 1 format.
- Xdebug-dwarf1 (MIPS) [on page](#) Select DWARF 1 format.
- Xdebug-dwarf1 (PowerPC) [on page](#) Select DWARF 1 format.
- Xdebug-dwarf1 (RH850) [on page](#) Select DWARF 1 format.
- Xdebug-dwarf1 (TriCore-PCP) [on page](#) Select DWARF 1 format.
- Xdebug-dwarf2 (ARM) [on page](#) Select DWARF 2 format.
- Xdebug-dwarf2 (ColdFire) [on page](#) Select DWARF 2 format.
- Xdebug-dwarf2 (Intelx86_Pentium) [on page](#) Select DWARF 2 format.
- Xdebug-dwarf2 (MIPS) [on page](#) Select DWARF 2 format.
- Xdebug-dwarf2 (PowerPC) [on page](#) Select DWARF 2 format.
- Xdebug-dwarf2 (RH850) [on page](#) Select DWARF 2 format.
- Xdebug-dwarf2 (TriCore-PCP) [on page](#) Select DWARF 2 format.
- Xdebug-dwarf3 (ARM) [on page](#) Select DWARF 3 format.
- Xdebug-dwarf3 (ColdFire) [on page](#) Select DWARF 3 format.
- Xdebug-dwarf3 (Intelx86_Pentium) [on page](#) Select DWARF 3 format.
- Xdebug-dwarf3 (MIPS) [on page](#) Select DWARF 3 format.
- Xdebug-dwarf3 (PowerPC) [on page](#) Select DWARF 3 format.
- Xdebug-dwarf3 (RH850) [on page](#) Select DWARF 3 format.
- Xdebug-dwarf3 (TriCore-PCP) [on page](#) Select DWARF 3 format.
- Xdefault-align (all) [on page](#) Set default value for section alignment.
- Xdisp... (ColdFire) [on page](#) Select default displacement size.
- Xemul-gnu-bug (Intelx86_Pentium) [on page](#) Emulate GNU Assembler's encoding of fdivp, fdivrp, fsubp, and fsubrp.
- Xenable-fpu (TriCore) [on page](#) Enable parsing of FPU instructions.
- Xenable-mmio (TriCore) [on page](#) Enable parsing of MMU instructions.
- Xerrata-imx (ARM) [on page](#) Use workaround for LDM instruction error.
- Xexp (RH850) [on page](#) -Xexp option.
- Xfix-bc-fw-mask-bug (PowerPC) [on page](#) Generate NOP for certain branch conditional instructions.

- Xfix-vr4133 (MIPS) on page Work around VR4133 bug.
- Xfpu... (ColdFire) on page Set FPU type.
- Xgen-coldfire-header (ColdFire) on page Generate EM_COLDFIRE ELF header for output object file.
- Xgen-shstrtab (all) on page Generate .shstrtab section as well as .strtab section.
- Xgnu-locals-... (all) on page Enable local GNU labels.
- Xheader... (all) on page Include header in listing.
- Xheader-format (all) on page Set header format.
- Xind16-range-off, -Xind16-range-on (ColdFire) on page Select default address register indirect mode.
- Xindex-long, -Xindex-word (ColdFire) on page Select default index size.
- Xisa-vle (PowerPC) on page Assemble using VLE instructions.
- Xlabel-colon... (all) on page Set label definition syntax.
- Xline-format (all) on page Set format of assembly line in listing.
- Xlist-... (all) on page Generate a listing file.
- Xlist-file-extension (all) on page Specify file extension for assembly listing.
- Xlit-marg... (all) on page Set delay of literal generation.
- Xllen (all) on page Set line length of listing file.
- Xmacro-arg-space-... (all) on page Enable blanks in macro arguments.
- Xmmu-68xxx, -Xmmu-all, -Xmmu-none (ColdFire) on page Set MMU type.
- Xmnem-all, -Xmnem-emb, -Xmnem-mit, -Xmnem-moto (ColdFire) on page Set mnemonics type.
- Xmnem-mit, -Xmnem-intel (Intelx86_Pentium) on page Set mnemonics type.
- Xnoexp (RH850) on page -Xnoexp option.
- Xobject-format (all) on page Set output object format.
- Xoptim-... (ARM) on page Select branch size optimizations.
- Xoptim-... (ColdFire) on page Select branch size optimizations.
- Xoptim-... (Intelx86_Pentium) on page Select branch size optimizations.
- Xoptim-... (PowerPC) on page Select branch size optimizations.
- Xoptim-... (TriCore-PCP) on page Select branch size optimizations.
- Xpage-skip (all) on page Set page break margin.
- Xplen (all) on page Set lines per page.
- Xprepare-compress (PowerPC) on page Limit length of conditional branch.

- Xrel-entry... (all) on page Specify type of relocation entry.
- Xsemi-is-newline (ARM) on page Treat semicolons as statement separators.
- Xsemi-is-newline (ColdFire) on page Treat semicolons as statement separators.
- Xsemi-is-newline (Intelx86_Pentium) on page Treat semicolons as statement separators.
- Xsemi-is-newline (MIPS) on page Treat semicolons as statement separators.
- Xsemi-is-newline (PowerPC) on page Treat semicolons as statement separators.
- Xset-reorder, -Xset-noreorder (all) on page Fill delay slots with NOPs.
- Xspace-... (all) on page Enable spaces between operands.
- Xstrip-locals..., -Xstrip-temps... (all) on page Delete local symbols.
- Xsubtitle (all) on page Set subtitle.
- Xsymbol-debug (all) on page
- Xtab-size (all) on page Set tab size.
- Xtitle (all) on page Set title.
- Xvle-opt-off (PowerPC) on page Disable automatic instruction optimization from BookE to VLE and from long VLE to short VLE.
- Xwarn-use-greg (MIPS) on page Warn when specified registers are accessed.
- Xwarn-use-greg (PowerPC) on page Warn when specified registers are accessed.

3.2. Alphabetical Compiler Option Index

- A (all) on page Define assertion.
- A- (all) on page Ignore predefined macros and assertions.
- @, -@@ (all) on page Read command-line options from file or variable.
- #, ##, ### (all) on page Print subprograms with arguments.
- balanced (all) on page Optimize code and balance between size and speed optimization.
- balanced-debug (all) on page Optimize code and balance between size and speed optimization, and enable debugging.
- b (all) on page Inform the c++ compiler about the compilation of C++ runtime libraries.
- c (all) on page Stop after assembly, produce object file.
- C (all) on page Pass along comments.
- D (all) on page Define preprocessor macro name.
- @E (all) on page Redirect standard error to file.
- e (all) on page Change diagnostic severity level.

- E (all) on page Stop after preprocessor, write source to standard output.
- g (all) on page Generate symbolic debugger information.
- , -??, -h, -help, -?X, -?W (all) on page Show information about compiler options.
- I (all) on page Specify directory for header files.
- i (all) on page Modify header file processing.
- I@ (all) on page Control search for user-defined header files.
- M (all) on page Specify pathname of target-spec file.
- mfix-vr4133 (MIPS) on page Work Around VR4133 Bug.
- mfix-vr4133 (MIPS16) on page Work Around VR4133 Bug.
- @O (all) on page Redirect standard output to file.
- O (all) on page Optimize code.
- o (all) on page Specify output file.
- P (all) on page Stop after preprocessor, produce source file.
- S (all) on page Stop after compilation, produce assembly file.
- size (all) on page Optimize code for size.
- size-debug (all) on page Optimize code for size and enable debugging.
- speed (all) on page Optimize code for speed.
- speed-debug (all) on page Optimize code for speed and enable debugging.
- t (all) on page Select the target processor.
- U (all) on page Undefine preprocessor macro name.
- V (all) on page Display current driver version number.
- v (all) on page Run driver in verbose mode.
- VV (all) on page Display current program version numbers.
- W a,-W :as: (all) on page Pass arguments to the assembler.
- W D (all) on page Define configuration variable.
- Wl, -W:ld: (all) on page Pass arguments to linker.
- W m (all) on page Specify linker command file.
- W s (all) on page Specify startup (crt0.o) module.
- W x (all) on page Substitute program or file for default.
- W x,arguments (all) on page Pass arguments to subprogram.

- W x.ext (all) on page Associate source file extension.
- Xabsolute18-const, -Xabsolute18-data (all) on page Take advantage of the 18-bit absolute addressing mode supported on TriCore hardware.
- Xaddr-code (all) on page Set addressing mode for code sections.
- Xaddr-const (all) on page Set addressing mode for const sections.
- Xaddr-data (all) on page Set addressing mode for data sections.
- Xaddr-sconst (all) on page Set addressing mode for sconst sections.
- Xaddr-sdata (all) on page Set addressing mode for sdata sections.
- Xaddr-string (all) on page Set addressing mode for string sections.
- Xaddr-user (all) on page Set addressing mode for user-defined sections.
- Xadd-underscore (ARM) on page Prefix function identifiers with underscore.
- Xadd-underscore (ARM-T) on page Prefix function identifiers with underscore.
- Xadd-underscore (Intelx86_Pentium) on page Prefix function identifiers with underscore.
- Xadd-underscore (MIPS) on page Prefix function identifiers with underscore.
- Xadd-underscore (MIPS16) on page Prefix function identifiers with underscore.
- Xadd-underscore (PowerPC) on page Prefix function identifiers with underscore.
- Xadd-underscore (PowerPCVLE) on page Prefix function identifiers with underscore.
- Xadd-underscore (RH850) on page Prefix function identifiers with underscore.
- Xadd-underscore (TriCore) on page Prefix function identifiers with underscore.
- Xaggressive-static-live (all) on page Optimize static and global variable access aggressively.
- Xalign-functions (all) on page Align functions on n-byte boundaries.
- Xalign-int-by2 (TriCore) on page Change alignment of long integer types to two bytes.
- Xalign-loop-to-cache (all) on page Optimize, using instruction-cache alignment.
- Xalign-min (all) on page Specify minimum alignment for single memory access to multi-byte values.
- Xalign-min-hardware-float (all) on page Specify minimum alignment for memory access to float registers.
- Xalign-off (ColdFire) on page Do not generate .align directive
- Xalternate-coloring (all) on page (PPC only) Use alternate frame-packing algorithm for register allocation
- Xapu-ibm-mac (PowerPC) on page Enable multiply and accumulate instructions.
- Xapu-lsp (PowerPC) on page Enable LSP instructions and intrinsic functions.
- Xapu-lsp (PowerPCVLE) on page Enable LSP instructions and intrinsic functions.
- Xapu-sie (PowerPC) on page Enable saturation instruction extension (SIE) instructions and intrinsic fuctions.

- Xapu-sie (PowerPCVLE) on page Enable saturation instruction extension (SIE) instructions and intrinsic functions.
- Xargs-in-regs (all) on page Pass argument in register.
- Xargs-not-aliased (all) on page Assume no aliasing of pointer arguments.
- Xarm-fdiv-exc-hook (ARM) on page Allow users to customize the behavior of floating point division by zero.
- Xarray-align-min (all) on page Specify minimum array alignment.
- Xasm-const-pound... (ColdFire) on page Disable '::' prefix for assembly numeric constants.
- Xbig-switch-table (ColdFire) on page Specify jump-table for switch statements.
- Xbit-fields-access-as-byte, -Xbit-fields-access-as-type (all) on page Disable bit-field access optimization.
- Xbit-fields-compress (all) on page Control bit-field storage.
- Xbit-fields-signed, -Xbit-fields-unsigned (all) on page Specify sign of plain bit-field.
- Xblock-count (all) on page Insert profiling code.
- Xbool-... (all) on page Control use of bool, true, and false keywords.
- Xbool-is-... (all) on page Set type for bool.
- Xbottom-up-init (all) on page Parse initial values bottom-up.
- Xbss-off, -Xbss-common-off (all) on page Control allocation of uninitialized variables in "COMMON" and bss sections.
- Xbss-zero-init-const-on, -Xbss-zero-init-data-off (all) on page Control movement of variables initialized to zero to BSS.
- Xc++-abr (all) on page Use abridged C++ libraries.
- Xcga-min-use (all) on page Optimize global assignments in conditionals.
- Xcharset-ascii (all) on page Generate code using ASCII character set.
- Xchar-signed, -Xchar-unsigned (all) on page Specify sign of plain char.
- Xclass-type-name-visible (all) on page Use old for scope rules.
- Xclib-optim-off (all) on page Disregard ANSI C library functions.
- Xc-new (all) on page Use the 'new' compiler frontend.
- Xcode-absolute... (all) on page Use absolute addressing for code.
- Xcode-factor (all) on page Share common code sequences.
- Xcode-model-absolute (PowerPC) on page select corresponding code model for PPC64 targets.
- Xcode-model-medium (PowerPC) on page select corresponding code model for PPC64 targets.
- Xcode-model-small (PowerPC) on page select corresponding code model for PPC64 targets.
- Xcode-relative... (all) on page Generate position-independent code (PIC).
- Xc-old (all) on page Use legacy DIAB front end (ctoa).

- Xcomdat (all) on page Mark Sections as COMDAT for Linker Collapse
- Xcomdat-info-file (all) on page Maintain project-wide COMDAT list.
- Xcompress-prolog-epilog (all) on page Reduce size of function prologue and epilogue.
- Xconst-in-... (all) on page Locate constants with "text" or "data".
- Xconventions-eabi (MIPS) on page Select EABI calling conventions.
- Xconventions-eabi (PowerPC) on page Select EABI calling conventions.
- Xconventions-eabi (PowerPCVLE) on page Select EABI calling conventions.
- Xconventions-n32 (MIPS) on page Select n32 calling conventions.
- Xcpp-cr-line-terminator (all) on page Enable line terminator behavior of carriage return (CR) characters.
- Xcpp-dump-symbols (all) on page Dump symbol information for macros or assertions.
- Xcpp-no-space (all) on page Suppress preprocessor spacing.
- Xcrb6-always (PowerPC) on page Always set CR bit 6 for an unprototyped function call.
- Xcrb6-always (PowerPCVLE) on page Always set CR bit 6 for an unprototyped function call.
- Xcrb6-float (PowerPC) on page Set CR bit 6 for an unprototyped function call using float.
- Xcrb6-float (PowerPCVLE) on page Set CR bit 6 for an unprototyped function call using float.
- Xcrb6-never (PowerPC) on page Never set CR bit 6 for an unprototyped function call.
- Xcrb6-never (PowerPCVLE) on page Never set CR bit 6 for an unprototyped function call.
- Xcritical-interrupts (RH850) on page Use critical interrupt return instruction if applicable.
- Xc++-template-name-lookup-old (all) on page Disable standards-conformant dependent name processing.
- Xdata-absolute... (all) on page Use absolute addressing for data.
- Xdata-relative... (all) on page Generate position-independent data (PID).
- Xdebug-align (all) on page Align .debug sections
- Xdebug-dwarf1 (all) on page Select DWARF 1 format.
- Xdebug-dwarf2 (all) on page Select DWARF 2 format.
- Xdebug-dwarf2-extensions-off (all) on page Suppress vendor-specific extensions.
- Xdebug-dwarf3 (all) on page Select DWARF 3 format.
- Xdebug-inline-on (all) on page Generate debug information for inlined functions.
- Xdebug-interrupt (PowerPC) on page Generate rfdi instruction for interrupt functions.
- Xdebug-interrupt (PowerPCVLE) on page Generate rfdi instruction for interrupt functions.
- Xdebug-local-all (all) on page Emit debug information for unused local variables.

- Xdebug-local-cie (all) on page Generate local CIE for each unit.
- Xdebug-mode (all) on page Disable debugging information extensions.
- Xdebug-source-line-barriers-off (all) on page Do not restrict scheduling optimization.
- Xdebug-source-line-barriers-on (all) on page Restrict scheduling optimization by source line.
- Xdebug-struct-all (all) on page Disable debug optimization of type information.
- Xdebug-struct-compact (all) on page Enable debug optimization of type information.
- Xdefer-pop (Intelx86_Pentium) on page Defer popping stack on function return.
- Xdiagnose-inline... (all) on page List all functions that have not been inlined.
- Xdialect... (all) on page Specify C dialect.
- Xdialect-c++... (all) on page Specify C++ dialect.
- Xdigraphs-... (all) on page Disable digraphs.
- Xdisable-keywords (all) on page Disable keywords selectively.
- Xdollar-in-ident (all) on page Allow dollar signs in identifiers.
- Xdouble-avoid (all) on page Force single precision.
- Xdouble-error (all) on page Generate error for double precision operations.
- Xdouble-warning (all) on page Warn about double precision operations.
- Xdsync (TriCore) on page Synchronize volatile data.
- Xdump-literals (MIPS16) on page Dump literals at the end of each function.
- Xdynamic-init (all) on page Generate initializers for static variables.
- Xeieio (PowerPC) on page Prevent reordering of memory access.
- Xeieio (PowerPCVLE) on page Prevent reordering of memory access.
- Xendian-little (all) on page Compile in little-endian mode.
- Xenum-bitfields (all) on page Do not treat bitfields of type enum the same as bitfields of type int.
- Xenum-is-... (all) on page Specify enum type.
- Xerrata-mask (TriCore) on page Use workarounds for TriCore hardware errors.
- Xexceptions-... (all) on page Enable exceptions.
- Xexplicit-inline-factor (all) on page Control inlining expansion.
- Xextend-args (all) on page Force precision of real arguments.
- Xfeedback (all) on page Optimize using profile data.
- Xfeedback-frequent (all) on page Set upper feedback limit for optimization parameters used with profile data.

- [-Xfeedback-seldom \(all\) on page](#) Set lower feedback limit for optimization parameters used with profile data.
- [-Xfloats-as-gnu \(ColdFire\) on page](#) Use GNU convention for returning floating point values.
- [-Xfloats-in-d0 \(ColdFire\) on page](#) Use d0 for returning floating point values.
- [-Xforce-declarations \(all\) on page](#) Generate warnings on undeclared functions.
- [-Xforce-prototypes \(all\) on page](#) Generate warnings on functions without previous prototype.
- [-Xforeign-as-ld \(all\) on page](#) Suppress assembler and linker parameters.
- [-Xfor-init-scope-... \(all\) on page](#) Use old for scope rules.
- [-Xfp-fast, -Xfp-normal, -Xfp-pedantic \(all\) on page](#) Specify degree of conformance to the IEEE754 standard.
- [-Xfp-long-double-off, -Xfp-float-only \(all\) on page](#) Convert double and long double.
- [-Xfp-min-prec... \(all\) on page](#) Specify minimum floating point precision.
- [-Xframe-align \(PowerPC\) on page](#) Change the stack frame alignment.
- [-Xframe-align \(PowerPCVLE\) on page](#) Change the stack frame alignment.
- [-Xframe-info \(all\) on page](#) Generate .frame_info for C functions.
- [-Xframe-ptr \(ColdFire\) on page](#) Generate link instruction.
- [-Xfull-pathname \(all\) on page](#) Include filename path in debug information.
- [-Xgcc-options-off \(all\) on page](#) Disable GNU option translator.
- [-Xgcc-options-on \(all\) on page](#) Enable GNU option translator.
- [-Xgcc-options-verbose \(all\) on page](#) Display GNU option translations.
- [-Xgnu-asm-usage \(all\) on page](#) generate warning or error for GNU style inline asm.
- [-Xhardware-divide... \(ColdFire\) on page](#) Control use of hardware divide instructions.
- [-Xhardware-divide... \(RH850\) on page](#) Control use of hardware divide instructions.
- [-Xhardware-divide... \(TriCore\) on page](#) Enable hardware divide instructions.
- [-Xhll-symbol \(all\) on page](#)
- [-Xident-off \(all\) on page](#) Do not pass #ident strings.
- [-Xieee754-pedantic \(all\) on page](#) Enable strict implementation of IEEE754 floating point standard.
- [-Xif-conversion \(all\) on page](#) Enable optimization of conditional jumps.
- [-Ximplicit-templates... \(all\) on page](#) Control template instantiation.
- [-Ximport \(all\) on page](#) Treat #include as #import.
- [-Xincfile-missing-ignore \(all\) on page](#) Ignore missing include files.
- [-Xinit-locals \(all\) on page](#) Initialize local variables.

- Xinit-section (all) on page Control generation of initialization and finalization sections.
- Xinit-section-default-pri (all) on page Control default priority for initialization and finalization sections.
- Xinit-value (all) on page Define initial value for -Xinit-locals.
- Xinline (all) on page Inline functions with fewer than n nodes.
- Xinline-asm-off (all) on page Disable inlining for functions with ASMs.
- Xinline-explicit-force (all) on page Allow inlining of recursive function calls.
- Xinline-single-use-statics, -Xinline-single-use-statics-off (all) on page Enable or disable inlining of single-use static functions.
- Xinline-volatile-off (all) on page Inhibit inlining of functions that access volatiles.
- Xinline-without-body-warn (all) on page Warn about inlined functions with no definition.
- Xinsert-nop-end (all) on page Insert nops into end of functions.
- Xinsert-nop-init (all) on page Insert nops into beginning of functions.
- Xinstr-00, -Xinstr-20 (ColdFire) on page Permit 68020 instructions.
- Xintc-eoir (PowerPC) on page Write end of interrupt register.
- Xintc-eoir (PowerPCVLE) on page Write end of interrupt register.
- Xinterrupt-funcs-use-rfci (PowerPC) on page Return from interrupt function using rfci.
- Xinterrupt-funcs-use-rfci (PowerPCVLE) on page Return from interrupt function using rfci.
- Xinterwork (ARM) on page Enable interworking.
- Xinterwork (ARM-T) on page Enable interworking.
- Xint-reciprocal (all) on page Allow division by reciprocal-multiply when optimizing.
- Xintrinsic-mask, -Xdisable-intrinsic-mask (all) on page Enable and disable specific intrinsic functions.
- Xjmpbuf-size (all) on page Set longjmp buffer size.
- Xkeep-assembly-file (all) on page Create and keep assembly file.
- Xkeep-function-asm (all) on page Generate function body if it contains an assembly string or calls an assembly macro.
- Xkeep-object-file (all) on page Create and keep object file.
- Xkeywords (all) on page Enable extended keywords.
- Xkill-opt (all) on page Disable target-dependent optimizations.
- Xkill-reorder (all) on page Disable target-independent optimizations.
- Xlibc-... (all) on page Use alternative C99 libraries.
- Xlicense-proxy-path (all) on page Change license proxy path.
- Xlicense-proxy-use (all) on page Turn license proxy off.

- Xlicense-wait (all) on page Wait for license.
- Xlimit-reaching (all) on page Disable reaching analysis optimizations.
- Xlink-time-lint (all) on page Perform link-time lint.
- Xlint (all) on page Generate warnings on suspicious/non-portable code.
- Xlno (all) on page Enable loop nest optimizations.
- Xlocal-data-area (all) on page Allocate static and global variables to local data area.
- Xlocal-data-area-static-only (all) on page Restrict local data area optimization to static variables.
- Xlocals-on-stack (all) on page Do not assign locals to registers.
- Xlsp-int (PowerPC) on page Enable use of LSP instructions for integer data types.
- Xlsp-int (PowerPCVLE) on page Enable use of LSP instructions for integer data types.
- Xlto-group (all) on page
- Xmac-conventions (all) on page Use Macintosh calling conventions for pascal functions.
- Xmachine-check-interrupt (PowerPC) on page Generate return for machine-check interrupt.
- Xmachine-check-interrupt (PowerPCVLE) on page Generate return for machine-check interrupt.
- Xmacro-in-pragma (all) on page Expand macros in pragmas.
- Xmacro-patterns (all) on page Use extended preprocessor syntax.
- Xmacro-undefined-warn (all) on page Warn on undefined macro in #if statement.
- Xmake-dependency (all) on page Show make rules.
- Xmake-dependency-canonicalize-path-off (all) on page Disable translation of path separators.
- Xmake-dependency-savefile (all) on page Specify dependency output file.
- Xmake-dependency-target (all) on page Specify dependency name.
- Xmax-inst-level (all) on page Set template instantiation recursion limit.
- Xmember-max-align (all) on page Set maximum structure member alignment.
- Xmemory-is-volatile, -Xglobals-volatile, -Xstatics-volatile, -Xpointers-volatile (all) on page Treat all variables as volatile.
- Xmetadata (all) on page Activate or deactivate metadata generation.
- Xmips16-hw-fp-call (MIPS) on page Pass floating point parameters to MIPS16 functions.
- Xmips-mad, -Xmips-movc, -Xmips-mul (MIPS) on page Enable specific MIPS instructions.
- Xmismatch-warning (all) on page Warn on type and argument mismatch.
- Xmode-64bit (MIPS) on page Allow MIPS64 assembler syntax.
- Xmode-64bit (MIPS16) on page Allow MIPS64 assembler syntax.

- Xname-... (all) on page Specify section name.
- Xnamespace-... (all) on page Disable C++ keywords namespace and using.
- Xnested-interrupts (MIPS) on page Save registers for nested interrupts.
- Xnested-interrupts (PowerPC) on page Save registers for nested interrupts.
- Xnested-interrupts (PowerPCVLE) on page Save registers for nested interrupts.
- Xnested-interrupts (RH850) on page Save registers for nested interrupts.
- Xno-builtin (all) on page Disable Diab built-in functions.
- Xno-gnu-inline-asm (all) on page (ARM, PPC, TC, RH850 only) Turn off extended GNU inline assembly extension use
- Xno-if-conversion (all) on page Disable optimization of conditional jumps.
- Xno-movw (ARM) on page Don't generate movw and movt instructions.
- Xno-reloc-movw (ARM) on page Don't generate movw and movt instructions in case of source operand isn't of type constant.
- Xno-wpo-dead (all) on page Disable whole program dead code and data elimination.
- XO (all) on page Enable extra optimizations.
- Xold-inline-asm-casting (all) on page Use old inline assembly casting.
- Xopt-count (all) on page Execute the compiler's optimizing stage n times.
- Xoptimistic-aliasing (all) on page Permit more optimizations.
- Xoptimized-debug-off (all) on page Disable most optimizations with -g.
- Xoptimized-debug-on (all) on page Allow optimizations with -g.
- Xparse-count (all) on page Specify optimization buffer size.
- Xpass-source (all) on page Output source as comments.
- Xpch-... (all) on page Use precompiled headers.
- Xpic (all) on page Generate position-independent code for shared libraries.
- Xpipeline-short-circuit (TriCore) on page Optimize evaluation of logical operators && and ||.
- Xpragma-section-first, -Xpragma-section-last (all) on page Control interpretation of multiple section pragmas.
- Xpreprocess-assembly (all) on page Preprocess assembly files.
- Xpreprocessor-lineno-off (all) on page Suppress line numbers in preprocessor output.
- Xpreprocessor-old (all) on page Use old preprocessor.
- Xptr-values-in-... (ColdFire) on page Select convention for returning pointer values from functions.
- Xreduce-reaching (all) on page Reduce reaching analysis optimizations.
- Xrelax-alias (all) on page Control aliasing rules applied by the compiler.

- Xrestart (all) on page Restart optimization from scratch.
- Xreverse-inline (all) on page Optimize code by moving some code to new functions.
- Xrh850-interrupt-use-reg (RH850) on page Generate save/restore code for system registers.
- Xrtc (all) on page Generate code for the run-time error checker (RTEC).
- Xrts-nil (ColdFire) on page Add null bytes after rts or rte for MCF5307 prefetch errata
- Xrtti, -Xrtti-off (all) on page Enable run-time type information.
- Xsavefpr-avoid (PowerPC) on page Save floating point registers without calling libimpl.a
- Xsavefpr-avoid (PowerPCVLE) on page Save floating point registers without calling libimpl.a
- Xsection-pad (all) on page Pad sections for optimized loading.
- Xsection-split (all) on page Generate data or functions in separate section classes.
- Xsection-split-name (all) on page Specify name pattern for split sections.
- Xsect-pri-off (all) on page Disable generation of priority section names.
- Xsect-pri-on (all) on page Enable generation of priority section names.
- Xsemi-is-comment (all) on page Treat semicolons as comment instead of a statement separators.
- Xsemi-is-newline (all) on page Treat semicolons as statement separators.
- Xshow-configuration (all) on page Control listing of -X options in assembly output.
- Xshow-inst (all) on page Print instantiations.
- Xshow-target (all) on page Show target.
- Xsingle-fp (ARM) on page internal option
- Xsingle-fp (PowerPC) on page internal option
- Xsingle-fp (PowerPCVLE) on page internal option
- Xsingle-fp (RH850) on page internal option
- Xsingle-fp (TriCore) on page internal option
- Xsize-opt (all) on page Optimize for size rather than speed.
- Xsmall-const (all) on page Set size limit for "small const" variable.
- Xsmall-const-registers (all) on page Extend small constant areas.
- Xsmall-data (all) on page Set size limit for "small data" variables.
- Xsmall-data-registers (all) on page Extend small data areas.
- Xsoft-float (ARM) on page internal option exclusively: Select software floating point emulation.
- Xsoft-float (ARM-T) on page internal option exclusively: Select software floating point emulation.

- Xsoft-float (MIPS) [on page](#) internal option exclusively: Select software floating point emulation.
- Xsoft-float (MIPS16) [on page](#) internal option exclusively: Select software floating point emulation.
- Xsoft-float (PowerPC) [on page](#) internal option exclusively: Select software floating point emulation.
- Xsoft-float (PowerPCVLE) [on page](#) internal option exclusively: Select software floating point emulation.
- Xsoft-float (RH850) [on page](#) internal option exclusively: Select software floating point emulation.
- Xsoft-float (TriCore) [on page](#) internal option exclusively: Select software floating point emulation.
- Xspe-fpmac (PowerPC) [on page](#) Accept SPE multiply-accumulate instructions.
- Xspe-fpmac (PowerPCVLE) [on page](#) Accept SPE multiply-accumulate instructions.
- Xspe-int (PowerPC) [on page](#) Enable support for SPE integer instructions
- Xspe-int (PowerPCVLE) [on page](#) Enable support for SPE integer instructions
- Xsplit-llong (ARM) [on page](#) Split long long expressions.
- Xstack-delay, -Xstack-delay-off (all) [on page](#) Delay popping stack after function call.
- Xstack-probe (all) [on page](#) Enable stack checking.
- Xstack-protection (all) [on page](#) Protects functions that have local arrays or that use alloca
- Xstack-protection-all (all) [on page](#) Protects all functions
- Xstack-protection-off (all) [on page](#) Turns stack protection off
- Xstack-protection-strong (all) [on page](#) Protects functions that have local arrays of any type, or local variables that have their address taken, or that use alloca
- Xstack-protection-verbose (all) [on page](#) see which functions are protected by SSP
- Xstatic-addr-error (all) [on page](#) Diagnose static initialization using address.
- Xstatic-addr-warning (all) [on page](#) Diagnose static initialization using address.
- Xstderr-fully-buffered (all) [on page](#) Buffer stderr.
- Xstmw-fast, -Xstmw-ok, -Xstmw-slow (PowerPC) [on page](#) Select stmw and lmw instructions.
- Xstmw-fast, -Xstmw-ok, -Xstmw-slow (PowerPCVLE) [on page](#) Select stmw and lmw instructions.
- Xstop-on-warning (all) [on page](#) Terminate compilation on warning.
- Xstrict-ansi (all) [on page](#) Compile C/C++ in pedantic mode.
- Xstrict-bitfield-promotions (all) [on page](#) Ignore sign when promoting bit-fields.
- Xstrict-eabi (all) [on page](#) Disable non-EABI conversion functions.
- Xstring-align (all) [on page](#) Align strings on n-byte boundaries.
- Xstruct-arg-warning (all) [on page](#) Warn on large structure.
- Xstruct-as-arg (all) [on page](#) Select convention for returning structures and unions.

- Xstruct-as-gnu (ColdFire) [on page](#) Select convention for returning structures and unions. (ColdFire only)
- Xstruct-assign-split-... (all) [on page](#) Control optimization of structure member assignments.
- Xstruct-as-static (all) [on page](#) Select convention for returning structures and unions.
- Xstruct-best-align (all) [on page](#) Align data on “natural” boundaries.
- Xstruct-min-align (all) [on page](#) Set minimum structure member alignment.
- Xstsw-fast, -Xstsw-ok, -Xstsw-slow (PowerPC) [on page](#) Select stswi and lswi instructions.
- Xstsw-fast, -Xstsw-ok, -Xstsw-slow (PowerPCVLE) [on page](#) Select stswi and lswi instructions.
- Xsuppress-warnings (all) [on page](#) Suppress warnings.
- Xswap-cr-nl (all) [on page](#) Swap ‘\n’ and ‘\r’ in constants.
- Xswitch-array-in-function-section (all) [on page](#) Put any compiler-generated switch tables in the same section as the parent function.
- Xswitch-array-off (all) [on page](#) Disables conversion of switch to look-up array
- Xswitch-table (all) [on page](#) Set threshold for a switch statement table.
- Xswitch-table-in-function-section (all) [on page](#) Put any compiler-generated switch tables in the same section as the parent function.
- Xswitch-table-off (all) [on page](#) Disable jump tables for switch statements.
- Xsyntax-warning-... (all) [on page](#) Disable certain syntax warnings.
- Xsystem-headers-warn (all) [on page](#) Disable certain system header warnings.
- Xtc-fdiv-exc-hook (TriCore) [on page](#) Allow users to customize the behavior of single precision floating point division by zero.
- Xtest-at-both (all) [on page](#) Loop tests at top and bottom.
- Xtest-at-bottom (all) [on page](#) Loop test at bottom.
- Xtest-at-top (all) [on page](#) Loop test at top.
- Xtrace-table (PowerPC) [on page](#) Generate trace table.
- Xtrace-table (PowerPCVLE) [on page](#) Generate trace table.
- Xtruncate (all) [on page](#) Truncate all identifiers after m characters.
- Xunderscore-leading (all) [on page](#) Prefix identifiers with underscore.
- Xunderscore-surround (all) [on page](#) Add underscores to identifier prefix and suffix.
- Xunderscore-trailing (all) [on page](#) Add underscore to identifier suffix.
- Xunroll (all) [on page](#) Set loop unrolling count.
- Xunroll-register-rename (all) [on page](#) Allow scalar variable to occupy multiple registers for unrolled loops.
- Xunroll-size (all) [on page](#) Set loop unrolling size.
- Xupdate-slow (PowerPC) [on page](#) Avoid pre-increment and pre-decrement instructions.

- Xupdate-slow (PowerPCVLE) [on page](#) Avoid pre-increment and pre-decrement instructions.
- Xuse-fpsimd (RH850) [on page](#) Enable vector data types and intrinsics for RH850G4MH
- Xuser-metadata (all) [on page](#) Add key-value pairs to the metadata of an object file.
- Xusing-std-... (all) [on page](#) Runtime declarations in standard namespace.
- Xvec-aligned (all) [on page](#) (MIPS64GLRN only) Inform compiler that arrays are properly aligned for vectorization.
- Xvec (all) [on page](#) (MIPS64GLRN only) Enable auto-vectorization.
- Xvec-args-aligned (all) [on page](#) (MIPS64GLRN only) Assume that function array/pointer arguments are aligned for vector memory access.
- Xvector-... (PowerPC) [on page](#) Vector code generation.
- Xvector-... (PowerPCVLE) [on page](#) Vector code generation.
- Xvec-verbose (all) [on page](#) (MIPS64GLRN only) Control diagnostic output for auto-vectorization.
- Xvoid-ptr-arith-ok (all) [on page](#) Void pointer arithmetic.
- Xvrsave-... (PowerPC) [on page](#) VRSAGE special purpose register.
- Xvrsave-... (PowerPCVLE) [on page](#) VRSAGE special purpose register.
- Xwchar (all) [on page](#) Define type for wchar.
- Xwchar_t-off (all) [on page](#) Disable use of wchar_t keyword.
- Xwchar_t-on (all) [on page](#) Enable use of wchar_t keyword.
- Xwhole-program-diagnostics (all) [on page](#) Set whole-program optimization diagnostics.
- Xwhole-program-optim (all) [on page](#) Use whole-program optimization.
- Xwpo-const (all) [on page](#) Enable whole program auto-constant optimization.
- Xwpo-dead (all) [on page](#) Enable whole program dead code and data elimination.
- Xwpo-delete-return-value (all) [on page](#) Enable whole program return value value deletion optimization.
- Xwpo-inline (all) [on page](#) Enable cross-module inlining.
- Xwpo-inline-single-calls (all) [on page](#) Always inline functions that are only called once (regardless of size).
- Xwpo-no-alias (all) [on page](#) Enable whole program unaliased variable optimization.
- Y I,dir (all) [on page](#) Specify default header file search path.

3.3. C Plus Plus Index

- Xbool-... (all) [on page](#) Control use of bool, true, and false keywords.
- Xbool-is-... (all) [on page](#) Set type for bool.
- Xc++-abr (all) [on page](#) Use abridged C++ libraries.

- Xclass-type-name-visible (all) [on page](#) Use old for scope rules.
- Xcomdat (all) [on page](#) Mark Sections as COMDAT for Linker Collapse
- Xcomdat-info-file (all) [on page](#) Maintain project-wide COMDAT list.
- Xc++-template-name-lookup-old (all) [on page](#) Disable standards-conformant dependent name processing.
- Xdialect-c++... (all) [on page](#) Specify C++ dialect.
- Xdigraphs-... (all) [on page](#) Disable digraphs.
- Xexceptions-... (all) [on page](#) Enable exceptions.
- Xfor-init-scope-... (all) [on page](#) Use old for scope rules.
- Xframe-info (all) [on page](#) Generate .frame_info for C functions.
- Ximplicit-templates... (all) [on page](#) Control template instantiation.
- Xjmpbuf-size (all) [on page](#) Set longjmp buffer size.
- Xmax-inst-level (all) [on page](#) Set template instantiation recursion limit.
- Xnamespace-... (all) [on page](#) Disable C++ keywords namespace and using.
- Xpch-... (all) [on page](#) Use precompiled headers.
- Xrtti, -Xrtti-off (all) [on page](#) Enable run-time type information.
- Xshow-inst (all) [on page](#) Print instantiations.
- Xstrict-ansi (all) [on page](#) Compile C/C++ in pedantic mode.
- Xsyntax-warning-... (all) [on page](#) Disable certain syntax warnings.
- Xusing-std-... (all) [on page](#) Runtime declarations in standard namespace.
- Xwchar (all) [on page](#) Define type for wchar.
- Xwchar_t-off (all) [on page](#) Disable use of wchar_t keyword.
- Xwchar_t-on (all) [on page](#) Enable use of wchar_t keyword.

3.4. Checking and Profiling Index

- Xblock-count (all) [on page](#) Insert profiling code.
- Xfeedback (all) [on page](#) Optimize using profile data.
- Xfeedback-frequent (all) [on page](#) Set upper feedback limit for optimization parameters used with profile data.
- Xfeedback-seldom (all) [on page](#) Set lower feedback limit for optimization parameters used with profile data.
- Xrtc (all) [on page](#) Generate code for the run-time error checker (RTEC).

3.5. Compiler Command-Line Options Index

- [-A \(all\) on page](#) Define assertion.
- [-A- \(all\) on page](#) Ignore predefined macros and assertions.
- [-balanced \(all\) on page](#) Optimize code and balance between size and speed optimization.
- [-balanced-debug \(all\) on page](#) Optimize code and balance between size and speed optimization, and enable debugging.
- [-C \(all\) on page](#) Pass along comments.
- [-D \(all\) on page](#) Define preprocessor macro name.
- [-e \(all\) on page](#) Change diagnostic severity level.
- [-g \(all\) on page](#) Generate symbolic debugger information.
- [-?, -??, -h, -help, -?X, -?W \(all\) on page](#) Show information about compiler options.
- [-I \(all\) on page](#) Specify directory for header files.
- [-i \(all\) on page](#) Modify header file processing.
- [-I@ \(all\) on page](#) Control search for user-defined header files.
- [-M \(all\) on page](#) Specify pathname of target-spec file.
- [-mfix-vr4133 \(MIPS\) on page](#) Work Around VR4133 Bug.
- [-mfix-vr4133 \(MIPS16\) on page](#) Work Around VR4133 Bug.
- [-O \(all\) on page](#) Optimize code.
- [-o \(all\) on page](#) Specify output file.
- [-size \(all\) on page](#) Optimize code for size.
- [-size-debug \(all\) on page](#) Optimize code for size and enable debugging.
- [-speed \(all\) on page](#) Optimize code for speed.
- [-speed-debug \(all\) on page](#) Optimize code for speed and enable debugging.
- [-t \(all\) on page](#) Select the target processor.
- [-U \(all\) on page](#) Undefine preprocessor macro name.
- [-W D \(all\) on page](#) Define configuration variable.
- [-Xgnu-asm-usage \(all\) on page](#) generate warning or error for GNU style inline asm.
- [-Y I,dir \(all\) on page](#) Specify default header file search path.

3.6. Debugging Index

- [-Xdebug-dwarf1 \(all\) on page](#) Select DWARF 1 format.

- [-Xdebug-dwarf2 \(all\) on page](#) Select DWARF 2 format.
- [-Xdebug-dwarf2-extensions-off \(all\) on page](#) Suppress vendor-specific extensions.
- [-Xdebug-dwarf3 \(all\) on page](#) Select DWARF 3 format.
- [-Xdebug-inline-on \(all\) on page](#) Generate debug information for inlined functions.
- [-Xdebug-local-all \(all\) on page](#) Emit debug information for unused local variables.
- [-Xdebug-local-cie \(all\) on page](#) Generate local CIE for each unit.
- [-Xdebug-mode \(all\) on page](#) Disable debugging information extensions.
- [-Xdebug-source-line-barriers-off \(all\) on page](#) Do not restrict scheduling optimization.
- [-Xdebug-source-line-barriers-on \(all\) on page](#) Restrict scheduling optimization by source line.
- [-Xdebug-struct-all \(all\) on page](#) Disable debug optimization of type information.
- [-Xdebug-struct-compact \(all\) on page](#) Enable debug optimization of type information.
- [-Xenum-bitfields \(all\) on page](#) Do not treat bitfields of type enum the same as bitfields of type int.
- [-Xfull-pathname \(all\) on page](#) Include filename path in debug information.
- [-Xhll-symbol \(all\) on page](#) [-Xinit-locals \(all\) on page](#) Initialize local variables.
- [-Xinit-value \(all\) on page](#) Define initial value for [-Xinit-locals](#).
- [-Xoptimized-debug-off \(all\) on page](#) Disable most optimizations with [-g](#).
- [-Xoptimized-debug-on \(all\) on page](#) Allow optimizations with [-g](#).
- [-Xstack-probe \(all\) on page](#) Enable stack checking.
- [-Xsymbol-debug \(all\) on page](#)

3.7. Diagnostic and Lint Index

- [-Xdiagnose-inline... \(all\) on page](#) List all functions that have not been inlined.
- [-Xforce-declarations \(all\) on page](#) Generate warnings on undeclared functions.
- [-Xforce-prototypes \(all\) on page](#) Generate warnings on functions without previous prototype.
- [-Xlink-time-lint \(all\) on page](#) Perform link-time lint.
- [-Xlint \(all\) on page](#) Generate warnings on suspicious/non-portable code.
- [-Xmacro-undefined-warn \(all\) on page](#) Warn on undefined macro in [#if](#) statement.
- [-Xmismatch-warning \(all\) on page](#) Warn on type and argument mismatch.
- [-Xstatic-addr-error \(all\) on page](#) Diagnose static initialization using address.
- [-Xstatic-addr-warning \(all\) on page](#) Diagnose static initialization using address.

- Xstderr-fully-buffered (all) on page Buffer stderr.
- Xstop-on-warning (all) on page Terminate compilation on warning.
- Xstruct-arg-warning (all) on page Warn on large structure.
- Xsuppress-warnings (all) on page Suppress warnings.

3.8. Driver Index

- Xforeign-as-ld (all) on page Suppress assembler and linker parameters.
- Xgcc-options-off (all) on page Disable GNU option translator.
- Xgcc-options-on (all) on page Enable GNU option translator.
- Xgcc-options-verbose (all) on page Display GNU option translations.
- Xkeep-assembly-file (all) on page Create and keep assembly file.
- Xkeep-object-file (all) on page Create and keep object file.
- Xlibc-... (all) on page Use alternative C99 libraries.
- Xlicense-proxy-path (all) on page Change license proxy path.
- Xlicense-proxy-use (all) on page Turn license proxy off.
- Xlicense-wait (all) on page Wait for license.
- Xpreprocess-assembly (all) on page Preprocess assembly files.
- Xshow-target (all) on page Show target.

3.9. Instruction Index

- Xadd-underscore (ARM) on page Prefix function identifiers with underscore.
- Xadd-underscore (ARM-T) on page Prefix function identifiers with underscore.
- Xadd-underscore (Intelx86_Pentium) on page Prefix function identifiers with underscore.
- Xadd-underscore (MIPS) on page Prefix function identifiers with underscore.
- Xadd-underscore (MIPS16) on page Prefix function identifiers with underscore.
- Xadd-underscore (PowerPC) on page Prefix function identifiers with underscore.
- Xadd-underscore (PowerPCVLE) on page Prefix function identifiers with underscore.
- Xadd-underscore (RH850) on page Prefix function identifiers with underscore.
- Xadd-underscore (TriCore) on page Prefix function identifiers with underscore.
- Xapu-ibm-mac (PowerPC) on page Enable multiply and accumulate instructions.

- Xapu-lsp (PowerPC) on page Enable LSP instructions and intrinsic functions.
- Xapu-lsp (PowerPCVLE) on page Enable LSP instructions and intrinsic functions.
- Xbig-switch-table (ColdFire) on page Specify jump-table for switch statements.
- Xconventions-eabi (MIPS) on page Select EABI calling conventions.
- Xconventions-eabi (PowerPC) on page Select EABI calling conventions.
- Xconventions-eabi (PowerPCVLE) on page Select EABI calling conventions.
- Xconventions-n32 (MIPS) on page Select n32 calling conventions.
- Xcrb6-always (PowerPC) on page Always set CR bit 6 for an unprototyped function call.
- Xcrb6-always (PowerPCVLE) on page Always set CR bit 6 for an unprototyped function call.
- Xcrb6-float (PowerPC) on page Set CR bit 6 for an unprototyped function call using float.
- Xcrb6-float (PowerPCVLE) on page Set CR bit 6 for an unprototyped function call using float.
- Xcrb6-never (PowerPC) on page Never set CR bit 6 for an unprototyped function call.
- Xcrb6-never (PowerPCVLE) on page Never set CR bit 6 for an unprototyped function call.
- Xcritical-interrupts (RH850) on page Use critical interrupt return instruction if applicable.
- Xdebug-interrupt (PowerPC) on page Generate rfdi instruction for interrupt functions.
- Xdebug-interrupt (PowerPCVLE) on page Generate rfdi instruction for interrupt functions.
- Xdsync (TriCore) on page Synchronize volatile data.
- Xdump-literals (MIPS16) on page Dump literals at the end of each function.
- Xeieio (PowerPC) on page Prevent reordering of memory access.
- Xeieio (PowerPCVLE) on page Prevent reordering of memory access.
- Xendian-little (all) on page Compile in little-endian mode.
- Xerrata-mask (TriCore) on page Use workarounds for TriCore hardware errors.
- Xfloats-as-gnu (ColdFire) on page Use GNU convention for returning floating point values.
- Xfloats-in-d0 (ColdFire) on page Use d0 for returning floating point values.
- Xframe-ptr (ColdFire) on page Generate link instruction.
- Xhardware-divide... (ColdFire) on page Control use of hardware divide instructions.
- Xhardware-divide... (RH850) on page Control use of hardware divide instructions.
- Xhardware-divide... (TriCore) on page Enable hardware divide instructions.
- Xinsert-nop-end (all) on page Insert nops into end of functions.
- Xinsert-nop-init (all) on page Insert nops into beginning of functions.

- Xinstr-00, -Xinstr-20 (ColdFire) on page Permit 68020 instructions.
- Xintc-eoir (PowerPC) on page Write end of interrupt register.
- Xintc-eoir (PowerPCVLE) on page Write end of interrupt register.
- Xinterrupt-funcs-use-rfci (PowerPC) on page Return from interrupt function using rfci.
- Xinterrupt-funcs-use-rfci (PowerPCVLE) on page Return from interrupt function using rfci.
- Xinterwork (ARM) on page Enable interworking.
- Xinterwork (ARM-T) on page Enable interworking.
- Xlsp-int (PowerPC) on page Enable use of LSP instructions for integer data types.
- Xlsp-int (PowerPCVLE) on page Enable use of LSP instructions for integer data types.
- Xmac-conventions (all) on page Use Macintosh calling conventions for pascal functions.
- Xmachine-check-interrupt (PowerPC) on page Generate return for machine-check interrupt.
- Xmachine-check-interrupt (PowerPCVLE) on page Generate return for machine-check interrupt.
- Xmips16-hw-fp-call (MIPS) on page Pass floating point parameters to MIPS16 functions.
- Xmips-mad, -Xmips-movc, -Xmips-mul (MIPS) on page Enable specific MIPS instructions.
- Xnested-interrupts (MIPS) on page Save registers for nested interrupts.
- Xnested-interrupts (PowerPC) on page Save registers for nested interrupts.
- Xnested-interrupts (PowerPCVLE) on page Save registers for nested interrupts.
- Xnested-interrupts (RH850) on page Save registers for nested interrupts.
- Xptr-values-in-... (ColdFire) on page Select convention for returning pointer values from functions.
- Xrh850-interrupt-use-reg (RH850) on page Generate save/restore code for system registers.
- Xrts-nil (ColdFire) on page Add null bytes after rts or rte for MCF5307 prefetch errata
- Xsavefpr-avoid (PowerPC) on page Save floating point registers without calling libimpl.a
- Xsavefpr-avoid (PowerPCVLE) on page Save floating point registers without calling libimpl.a
- Xsingle-fp (ARM) on page internal option
- Xsingle-fp (PowerPC) on page internal option
- Xsingle-fp (PowerPCVLE) on page internal option
- Xsingle-fp (RH850) on page internal option
- Xsingle-fp (TriCore) on page internal option
- Xsoft-float (ARM) on page internal option exclusively: Select software floating point emulation.
- Xsoft-float (ARM-T) on page internal option exclusively: Select software floating point emulation.

- Xsoft-float (MIPS) on page internal option exclusively: Select software floating point emulation.
- Xsoft-float (MIPS16) on page internal option exclusively: Select software floating point emulation.
- Xsoft-float (PowerPC) on page internal option exclusively: Select software floating point emulation.
- Xsoft-float (PowerPCVLE) on page internal option exclusively: Select software floating point emulation.
- Xsoft-float (RH850) on page internal option exclusively: Select software floating point emulation.
- Xsoft-float (TriCore) on page internal option exclusively: Select software floating point emulation.
- Xspe-fpmac (PowerPC) on page Accept SPE multiply-accumulate instructions.
- Xspe-fpmac (PowerPCVLE) on page Accept SPE multiply-accumulate instructions.
- Xspe-int (PowerPC) on page Enable support for SPE integer instructions
- Xspe-int (PowerPCVLE) on page Enable support for SPE integer instructions
- Xsplit-llong (ARM) on page Split long long expressions.
- Xstmw-fast, -Xstmw-ok, -Xstmw-slow (PowerPC) on page Select stmw and lmw instructions.
- Xstmw-fast, -Xstmw-ok, -Xstmw-slow (PowerPCVLE) on page Select stmw and lmw instructions.
- Xstrict-eabi (all) on page Disable non-EABI conversion functions.
- Xstruct-as-arg (all) on page Select convention for returning structures and unions.
- Xstruct-as-gnu (ColdFire) on page Select convention for returning structures and unions. (ColdFire only)
- Xstruct-as-static (all) on page Select convention for returning structures and unions.
- Xstsw-fast, -Xstsw-ok, -Xstsw-slow (PowerPC) on page Select stswi and lswi instructions.
- Xstsw-fast, -Xstsw-ok, -Xstsw-slow (PowerPCVLE) on page Select stswi and lswi instructions.
- Xtrace-table (PowerPC) on page Generate trace table.
- Xtrace-table (PowerPCVLE) on page Generate trace table.
- Xupdate-slow (PowerPC) on page Avoid pre-increment and pre-decrement instructions.
- Xupdate-slow (PowerPCVLE) on page Avoid pre-increment and pre-decrement instructions.
- Xvector-... (PowerPC) on page Vector code generation.
- Xvector-... (PowerPCVLE) on page Vector code generation.
- Xvrsave-... (PowerPC) on page VRSAVE special purpose register.
- Xvrsave-... (PowerPCVLE) on page VRSAVE special purpose register.

3.10. Memory Index

- Xalign-int-by2 (TriCore) on page Change alignment of long integer types to two bytes.

- Xalign-min (all) on page Specify minimum alignment for single memory access to multi-byte values.
- Xalign-min-hardware-float (all) on page Specify minimum alignment for memory access to float registers.
- Xalign-off (ColdFire) on page Do not generate .align directive
- Xarray-align-min (all) on page Specify minimum array alignment.
- Xdynamic-init (all) on page Generate initializers for static variables.
- Xframe-align (PowerPC) on page Change the stack frame alignment.
- Xframe-align (PowerPCVLE) on page Change the stack frame alignment.
- Xinit-section (all) on page Control generation of initialization and finalization sections.
- Xinit-section-default-pri (all) on page Control default priority for initialization and finalization sections.
- Xmember-max-align (all) on page Set maximum structure member alignment.
- Xmemory-is-volatile, -Xglobals-volatile, -Xstatics-volatile, -Xpointers-volatile (all) on page Treat all variables as volatile.
- Xstring-align (all) on page Align strings on n-byte boundaries.
- Xstruct-best-align (all) on page Align data on “natural” boundaries.
- Xstruct-min-align (all) on page Set minimum structure member alignment.

3.11. Numerical Compiler Option Index

- X100: -Xaddr-data (all) on page Set addressing mode for data sections.
- X101: -Xaddr-sdata (all) on page Set addressing mode for sdata sections.
- X102: -Xaddr-const (all) on page Set addressing mode for const sections.
- X103: -Xaddr-sconst (all) on page Set addressing mode for sconst sections.
- X104: -Xaddr-string (all) on page Set addressing mode for string sections.
- X105: -Xaddr-code (all) on page Set addressing mode for code sections.
- X106: -Xaddr-user (all) on page Set addressing mode for user-defined sections.
- X10: -Xstack-probe (all) on page Enable stack checking.
- X114: -Xrelax-alias (all) on page Control aliasing rules applied by the compiler.
- X115: -Xlocal-data-area (all) on page Allocate static and global variables to local data area.
- X116: -Xdebug-struct-all (all) on page Disable debug optimization of type information.
- X116: -Xdebug-struct-compact (all) on page Enable debug optimization of type information.
- X117: -Xcpp-no-space (all) on page Suppress preprocessor spacing.
- X118: -Xbit-fields-access-as-byte, -Xbit-fields-access-as-type (all) on page Disable bit-field access optimization.

- X11: -Xpass-source (all) on page Output source as comments.
- X120: -Xcomdat (all) on page Mark Sections as COMDAT for Linker Collapse
- X121: -Xdynamic-init (all) on page Generate initializers for static variables.
- X122: -Xsect-pri-off (all) on page Disable generation of priority section names.
- X122: -Xsect-pri-on (all) on page Enable generation of priority section names.
- X125: -Xfull-pathname (all) on page Include filename path in debug information.
- X127: -Xstack-delay, -Xstack-delay-off (all) on page Delay popping stack after function call.
- X129: -Xsection-split (all) on page Generate data or functions in separate section classes.
- X12: -Xbit-fields-signed, -Xbit-fields-unsigned (all) on page Specify sign of plain bit-field.
- X135: -Xbit-fields-compress (all) on page Control bit-field storage.
- X136: -Xexplicit-inline-factor (all) on page Control inlining expansion.
- X137: -Xold-inline-asm-casting (all) on page Use old inline assembly casting.
- X138: -Xlicense-wait (all) on page Wait for license.
- X139: -Xaggressive-static-live (all) on page Optimize static and global variable access aggressively.
- X13: -Xswap-cr-nl (all) on page Swap '\n' and '\r' in constants.
- X143: -Xswitch-table (all) on page Set threshold for a switch statement table.
- X147: -Xstruct-assign-split-... (all) on page Control optimization of structure member assignments.
- X149: -Xalign-min-hardware-float (all) on page Specify minimum alignment for memory access to float registers.
- X14: -Xsuppress-warnings (all) on page Suppress warnings.
- X152: -Xsection-pad (all) on page Pad sections for optimized loading.
- X153: -Xdebug-dwarf1 (all) on page Select DWARF 1 format.
- X153: -Xdebug-dwarf2 (all) on page Select DWARF 2 format.
- X153: -Xdebug-dwarf3 (all) on page Select DWARF 3 format.
- X154: -Xintrinsic-mask, -Xdisable-intrinsic-mask (all) on page Enable and disable specific intrinsic functions.
- X155: -Xpreprocessor-old (all) on page Use old preprocessor.
- X156: -Xmake-dependency (all) on page Show make rules.
- X157: -Xmacro-in-pragma (all) on page Expand macros in pragmas.
- X158: -Xcpp-dump-symbols (all) on page Dump symbol information for macros or assertions.
- X15: -Xunroll (all) on page Set loop unrolling count.
- X161: -Xarray-align-min (all) on page Specify minimum array alignment.

- X163: [-Xinline-explicit-force \(all\) on page](#) Allow inlining of recursive function calls.
- X165: [-Xpreprocessor-lineno-off \(all\) on page](#) Suppress line numbers in preprocessor output.
- X166: [-Xlocal-data-area-static-only \(all\) on page](#) Restrict local data area optimization to static variables.
- X167: [-Xvoid-ptr-arith-ok \(all\) on page](#) Void pointer arithmetic.
- X16: [-Xunroll-size \(all\) on page](#) Set loop unrolling size.
- X170: [-Xdebug-align \(all\) on page](#) Align .debug sections
- X171: [-Xmacro-undefined-warn \(all\) on page](#) Warn on undefined macro in #if statement.
- X172: [-Xincfile-missing-ignore \(all\) on page](#) Ignore missing include files.
- X173: [-Xstderr-fully-buffered \(all\) on page](#) Buffer stderr.
- X175: [-Xinit-section-default-pri \(all\) on page](#) Control default priority for initialization and finalization sections.
- X176: [-Xdebug-local-cie \(all\) on page](#) Generate local CIE for each unit.
- X179: [-Xshow-configuration \(all\) on page](#) Control listing of -X options in assembly output.
- X17: [-Xstruct-best-align \(all\) on page](#) Align data on “natural” boundaries.
- X180: [-Xcga-min-use \(all\) on page](#) Optimize global assignments in conditionals.
- X181: [-Xdebug-local-all \(all\) on page](#) Emit debug information for unused local variables.
- X18: [-Xstring-align \(all\) on page](#) Align strings on n-byte boundaries.
- X191: [-Xlicense-proxy-use \(all\) on page](#) Turn license proxy off.
- X194: [-Xgcc-options-off \(all\) on page](#) Disable GNU option translator.
- X194: [-Xgcc-options-on \(all\) on page](#) Enable GNU option translator.
- X195: [-Xgcc-options-verbose \(all\) on page](#) Display GNU option translations.
- X196: [-X448: -Xabsolute18-const, -Xabsolute18-data \(all\) on page](#) Take advantage of the 18-bit absolute addressing mode supported on TriCore hardware.
- X19: [-Xinline \(all\) on page](#) Inline functions with fewer than n nodes.
- X200: [-Xexceptions-... \(all\) on page](#) Enable exceptions.
- X201: [-Xjmpbuf-size \(all\) on page](#) Set longjmp buffer size.
- X202: [-Xdigraphs-... \(all\) on page](#) Disable digraphs.
- X205: [-Xrtti, -Xrtti-off \(all\) on page](#) Enable run-time type information.
- X207: [-Ximplicit-templates... \(all\) on page](#) Control template instantiation.
- X212: [-Xshow-inst \(all\) on page](#) Print instantiations.
- X213: [-Xbool-... \(all\) on page](#) Control use of bool, true, and false keywords.
- X214: [-Xwchar_t-off \(all\) on page](#) Disable use of wchar_t keyword.

- X214: -Xwchar_t-on (all) on page Enable use of wchar_t keyword.
- X215: -Xsyntax-warning-... (all) on page Disable certain syntax warnings.
- X216: -Xmax-inst-level (all) on page Set template instantiation recursion limit.
- X217: -Xfor-init-scope-... (all) on page Use old for scope rules.
- X219: -Xnamespace-... (all) on page Disable C++ keywords namespace and using.
- X21: -Xbottom-up-init (all) on page Parse initial values bottom-up.
- X22: -Xtruncate (all) on page Truncate all identifiers after m characters.
- X23: -Xchar-signed, -Xchar-unsigned (all) on page Specify sign of plain char.
- X24: -Xblock-count (all) on page Insert profiling code.
- X25: -Xopt-count (all) on page Execute the compiler's optimizing stage n times.
- X26: -XO (all) on page Enable extra optimizations.
- X27: -Xkill-opt (all) on page Disable target-dependent optimizations.
- X28: -Xkill-reorder (all) on page Disable target-independent optimizations.
- X29: -Xrestart (all) on page Restart optimization from scratch.
- X2: -Xmismatch-warning (all) on page Warn on type and argument mismatch.
- X31: -Xconventions-eabi (MIPS) on page Select EABI calling conventions.
- X31: -Xconventions-eabi (PowerPC) on page Select EABI calling conventions.
- X31: -Xconventions-eabi (PowerPCVLE) on page Select EABI calling conventions.
- X31: -Xconventions-n32 (MIPS) on page Select n32 calling conventions.
- X31: -Xrh850-interrupt-use-reg (RH850) on page Generate save/restore code for system registers.
- X32: -Xptr-values-in-... (ColdFire) on page Select convention for returning pointer values from functions.
- X32: -Xstmw-fast, -Xstmw-ok, -Xstmw-slow (PowerPC) on page Select stmw and lmw instructions.
- X32: -Xstmw-fast, -Xstmw-ok, -Xstmw-slow (PowerPCVLE) on page Select stmw and lmw instructions.
- X34: -Xadd-underscore (ARM) on page Prefix function identifiers with underscore.
- X34: -Xadd-underscore (ARM-T) on page Prefix function identifiers with underscore.
- X34: -Xadd-underscore (Intelx86_Pentium) on page Prefix function identifiers with underscore.
- X34: -Xadd-underscore (MIPS) on page Prefix function identifiers with underscore.
- X34: -Xadd-underscore (MIPS16) on page Prefix function identifiers with underscore.
- X34: -Xadd-underscore (PowerPC) on page Prefix function identifiers with underscore.
- X34: -Xadd-underscore (PowerPCVLE) on page Prefix function identifiers with underscore.

- X34: -Xadd-underscore (RH850) [on page](#) Prefix function identifiers with underscore.
- X34: -Xadd-underscore (TriCore) [on page](#) Prefix function identifiers with underscore.
- X35: -Xbig-switch-table (ColdFire) [on page](#) Specify jump-table for switch statements.
- X35: -Xhardware-divide... (RH850) [on page](#) Control use of hardware divide instructions.
- X35: -Xhardware-divide... (TriCore) [on page](#) Enable hardware divide instructions.
- X35: -Xstsw-fast, -Xstsw-ok, -Xstsw-slow (PowerPC) [on page](#) Select stswi and lswi instructions.
- X35: -Xstsw-fast, -Xstsw-ok, -Xstsw-slow (PowerPCVLE) [on page](#) Select stswi and lswi instructions.
- X36: -Xcrb6-always (PowerPC) [on page](#) Always set CR bit 6 for an unprototyped function call.
- X36: -Xcrb6-always (PowerPCVLE) [on page](#) Always set CR bit 6 for an unprototyped function call.
- X36: -Xcrb6-float (PowerPC) [on page](#) Set CR bit 6 for an unprototyped function call using float.
- X36: -Xcrb6-float (PowerPCVLE) [on page](#) Set CR bit 6 for an unprototyped function call using float.
- X36: -Xcrb6-never (PowerPC) [on page](#) Never set CR bit 6 for an unprototyped function call.
- X36: -Xcrb6-never (PowerPCVLE) [on page](#) Never set CR bit 6 for an unprototyped function call.
- X36: -Xframe-ptr (ColdFire) [on page](#) Generate link instruction.
- X37: -Xalign-int-by2 (TriCore) [on page](#) Change alignment of long integer types to two bytes.
- X37: -Xalign-off (ColdFire) [on page](#) Do not generate .align directive
- X38: -Xtrace-table (PowerPC) [on page](#) Generate trace table.
- X38: -Xtrace-table (PowerPCVLE) [on page](#) Generate trace table.
- X3: -Xfp-min-prec... (all) [on page](#) Specify minimum floating point precision.
- X405: -Xlink-time-lint (all) [on page](#) Perform link-time lint.
- X407: -Xint-reciprocal (all) [on page](#) Allow division by reciprocal-multiply when optimizing.
- X40: -Xhardware-divide... (ColdFire) [on page](#) Control use of hardware divide instructions.
- X40: -Xinterwork (ARM) [on page](#) Enable interworking.
- X40: -Xinterwork (ARM-T) [on page](#) Enable interworking.
- X40: -Xmips16-hw-fp-call (MIPS) [on page](#) Pass floating point parameters to MIPS16 functions.
- X40: -Xnested-interrupts (PowerPC) [on page](#) Save registers for nested interrupts.
- X40: -Xnested-interrupts (PowerPCVLE) [on page](#) Save registers for nested interrupts.
- X410: -Xsystem-headers-warn (all) [on page](#) Disable certain system header warnings.
- X418: -Xdisable-keywords (all) [on page](#) Disable keywords selectively.
- X41: -Xcritical-interrupts (RH850) [on page](#) Use critical interrupt return instruction if applicable.

- X41: -Xinterrupt-funcs-use-rfci (PowerPC) on page Return from interrupt function using rfci.
- X41: -Xinterrupt-funcs-use-rfci (PowerPCVLE) on page Return from interrupt function using rfci.
- X41: -Xmachine-check-interrupt (PowerPC) on page Generate return for machine-check interrupt.
- X41: -Xmachine-check-interrupt (PowerPCVLE) on page Generate return for machine-check interrupt.
- X41: -Xnested-interrupts (MIPS) on page Save registers for nested interrupts.
- X424: -Xparse-count (all) on page Specify optimization buffer size.
- X426: -Xwhole-program-diagnostics (all) on page Set whole-program optimization diagnostics.
- X42: -Xdump-literals (MIPS16) on page Dump literals at the end of each function.
- X42: -Xfloats-in-d0 (ColdFire) on page Use d0 for returning floating point values.
- X43: -Xdsync (TriCore) on page Synchronize volatile data.
- X43: -Xeieio (PowerPC) on page Prevent reordering of memory access.
- X43: -Xeieio (PowerPCVLE) on page Prevent reordering of memory access.
- X43: -Xfloats-as-gnu (ColdFire) on page Use GNU convention for returning floating point values.
- X443: -Xinline-asm-off (all) on page Disable inlining for functions with ASMs.
- X448: -Xabsolute18-const, -Xabsolute18-data (all) on page Take advantage of the 18-bit absolute addressing mode supported on TriCore hardware.
- X449: -Xreverse-inline (all) on page Optimize code by moving some code to new functions.
- X44: -Xerrata-mask (TriCore) on page Use workarounds for TriCore hardware errors.
- X44: -Xinstr-00, -Xinstr-20 (ColdFire) on page Permit 68020 instructions.
- X44: -Xnested-interrupts (RH850) on page Save registers for nested interrupts.
- X44: -Xvector-... (PowerPC) on page Vector code generation.
- X44: -Xvector-... (PowerPCVLE) on page Vector code generation.
- X451: -Xalign-loop-to-cache (all) on page Optimize, using instruction-cache alignment.
- X454: -Xsmall-data-registers (all) on page Extend small data areas.
- X455: -Xsmall-const-registers (all) on page Extend small constant areas.
- X457: -Xmacro-patterns (all) on page Use extended preprocessor syntax.
- X45: -Xasm-const-pound... (ColdFire) on page Disable ':::' prefix for assembly numeric constants.
- X45: -Xvrsave-... (PowerPC) on page VRSAVE special purpose register.
- X45: -Xvrsave-... (PowerPCVLE) on page VRSAVE special purpose register.
- X462: -Xcpp-cr-line-terminator (all) on page Enable line terminator behavior of carriage return (CR) characters.

- X472: -Xswitch-array-in-function-section (all) on page Put any compiler-generated switch tables in the same section as the parent function.
- X472: -Xswitch-table-in-function-section (all) on page Put any compiler-generated switch tables in the same section as the parent function.
- X47: -Xrts-nil (ColdFire) on page Add null bytes after rts or rte for MCF5307 prefetch errata
- X48: -Xintc-eoir (PowerPC) on page Write end of interrupt register.
- X48: -Xintc-eoir (PowerPCVLE) on page Write end of interrupt register.
- X49: -Xframe-align (PowerPC) on page Change the stack frame alignment.
- X49: -Xframe-align (PowerPCVLE) on page Change the stack frame alignment.
- X49: -Xlicense-proxy-path (all) on page Change license proxy path.
- X49: -Xpipeline-short-circuit (TriCore) on page Optimize evaluation of logical operators && and ||.
- X4: -Xmemory-is-volatile, -Xglobals-volatile, -Xstatics-volatile, -Xpointers-volatile (all) on page Treat all variables as volatile.
- X51: -Xinsert-nop-init (all) on page Insert nops into beginning of functions.
- X52: -Xinsert-nop-end (all) on page Insert nops into end of functions.
- X53: -Xstrict-eabi (all) on page Disable non-EABI conversion functions.
- X54: -Xalign-functions (all) on page Align functions on n-byte boundaries.
- X56: -Xsoft-float (ARM) on page internal option exclusively: Select software floating point emulation.
- X56: -Xsoft-float (ARM-T) on page internal option exclusively: Select software floating point emulation.
- X56: -Xsoft-float (MIPS) on page internal option exclusively: Select software floating point emulation.
- X56: -Xsoft-float (MIPS16) on page internal option exclusively: Select software floating point emulation.
- X56: -Xsoft-float (PowerPC) on page internal option exclusively: Select software floating point emulation.
- X56: -Xsoft-float (PowerPCVLE) on page internal option exclusively: Select software floating point emulation.
- X56: -Xsoft-float (RH850) on page internal option exclusively: Select software floating point emulation.
- X56: -Xsoft-float (TriCore) on page internal option exclusively: Select software floating point emulation.
- X58: -Xcode-relative... (all) on page Generate position-independent code (PIC).
- X59: -Xdata-absolute... (all) on page Use absolute addressing for data.
- X59: -Xdata-relative... (all) on page Generate position-independent data (PID).
- X5: -Xlocals-on-stack (all) on page Do not assign locals to registers.
- X60: -Xcharset-ascii (all) on page Generate code using ASCII character set.
- X61: -Xargs-in-regs (all) on page Pass argument in register.
- X62: -Xpic (all) on page Generate position-independent code for shared libraries.

- X63: -Xident-off (all) on page Do not pass #ident strings.
- X64: -Xrtc (all) on page Generate code for the run-time error checker (RTEC).
- X65: -Xargs-not-aliased (all) on page Assume no aliasing of pointer arguments.
- X66: -Xclib-optim-off (all) on page Disregard ANSI C library functions.
- X67: -Xdollar-in-ident (all) on page Allow dollar signs in identifiers.
- X68: -Xfeedback-frequent (all) on page Set upper feedback limit for optimization parameters used with profile data.
- X69: -Xfeedback-seldom (all) on page Set lower feedback limit for optimization parameters used with profile data.
- X6: -Xtest-at-both (all) on page Loop tests at top and bottom.
- X6: -Xtest-at-bottom (all) on page Loop test at bottom.
- X6: -Xtest-at-top (all) on page Loop test at top.
- X70: -Xfp-long-double-off, -Xfp-float-only (all) on page Convert double and long double.
- X71: -Xunderscore-leading (all) on page Prefix identifiers with underscore.
- X71: -Xunderscore-surround (all) on page Add underscores to identifier prefix and suffix.
- X71: -Xunderscore-trailing (all) on page Add underscore to identifier suffix.
- X73: -Xsize-opt (all) on page Optimize for size rather than speed.
- X74: -Xconst-in-... (all) on page Locate constants with "text" or "data".
- X75: -Ximport (all) on page Treat #include as #import.
- X76: -Xstruct-min-align (all) on page Set minimum structure member alignment.
- X77: -Xextend-args (all) on page Force precision of real arguments.
- X78: -Xkeywords (all) on page Enable extended keywords.
- X79: -Xmac-conventions (all) on page Use Macintosh calling conventions for pascal functions.
- X7: -Xstrict-ansi (all) on page Compile C/C++ in pedantic mode.
- X80: -Xstruct-as-arg (all) on page Select convention for returning structures and unions.
- X80: -Xstruct-as-gnu (ColdFire) on page Select convention for returning structures and unions. (ColdFire only)
- X80: -Xstruct-as-static (all) on page Select convention for returning structures and unions.
- X81: -Xstatic-addr-error (all) on page Diagnose static initialization using address.
- X81: -Xstatic-addr-warning (all) on page Diagnose static initialization using address.
- X82: -Xfp-fast, -Xfp-normal, -Xfp-pedantic (all) on page Specify degree of conformance to the IEEE754 standard.
- X82: -Xieee754-pedantic (all) on page Enable strict implementation of IEEE754 floating point standard.
- X83: -Xbss-off, -Xbss-common-off (all) on page Control allocation of uninitialized variables in "COMMON" and bss sections.

- X84: -Xlint (all) on page Generate warnings on suspicious/non-portable code.
- X85: -Xstop-on-warning (all) on page Terminate compilation on warning.
- X86: -Xwchar (all) on page Define type for wchar.
- X87: -Xinit-locals (all) on page Initialize local variables.
- X88: -Xmember-max-align (all) on page Set maximum structure member alignment.
- X89: -Xoptimized-debug-off (all) on page Disable most optimizations with -g.
- X89: -Xoptimized-debug-on (all) on page Allow optimizations with -g.
- X8: -Xenum-is-... (all) on page Specify enum type.
- X90: -Xinit-value (all) on page Define initial value for -Xinit-locals.
- X91: -Xinit-section (all) on page Control generation of initialization and finalization sections.
- X92: -Xstruct-arg-warning (all) on page Warn on large structure.
- X93: -Xalign-min (all) on page Specify minimum alignment for single memory access to multi-byte values.
- X94: -Xendian-little (all) on page Compile in little-endian mode.
- X96: -Xdoube-avoid (all) on page Force single precision.
- X96: -Xdoube-error (all) on page Generate error for double precision operations.
- X96: -Xdoube-warning (all) on page Warn about double precision operations.
- X97: -Xsmall-data (all) on page Set size limit for "small data" variables.
- X98: -Xsmall-const (all) on page Set size limit for "small const" variable.
- X99: -Xdebug-mode (all) on page Disable debugging information extensions.
- X9: -Xforce-declarations (all) on page Generate warnings on undeclared functions.
- X9: -Xforce-prototypes (all) on page Generate warnings on functions without previous prototype.

3.12. Optimization Index

- Xaggressive-static-live (all) on page Optimize static and global variable access aggressively.
- Xalign-loop-to-cache (all) on page Optimize, using instruction-cache alignment.
- Xalternate-coloring (all) on page (PPC only) Use alternate frame-packing algorithm for register allocation
- Xapu-sie (PowerPC) on page Enable saturation instruction extension (SIE) instructions and intrinsic fuctions.
- Xapu-sie (PowerPCVLE) on page Enable saturation instruction extension (SIE) instructions and intrinsic fuctions.
- Xargs-not-aliased (all) on page Assume no aliasing of pointer arguments.
- Xcga-min-use (all) on page Optimize global assignments in conditionals.

- Xclib-optim-off (all) on page Disregard ANSI C library functions.
- Xcode-factor (all) on page Share common code sequences.
- Xcompress-prolog-epilog (all) on page Reduce size of function prologue and epilogue.
- Xdefer-pop (Intelx86_Pentium) on page Defer popping stack on function return.
- Xexplicit-inline-factor (all) on page Control inlining expansion.
- Xif-conversion (all) on page Enable optimization of conditional jumps.
- Xinline (all) on page Inline functions with fewer than n nodes.
- Xinline-asm-off (all) on page Disable inlining for functions with ASMs.
- Xinline-explicit-force (all) on page Allow inlining of recursive function calls.
- Xinline-single-use-statics, -Xinline-single-use-statics-off (all) on page Enable or disable inlining of single-use static functions.
- Xinline-volatile-off (all) on page Inhibit inlining of functions that access volatiles.
- Xint-reciprocal (all) on page Allow division by reciprocal-multiply when optimizing.
- Xkeep-function-asm (all) on page Generate function body if it contains an assembly string or calls an assembly macro.
- Xkill-opt (all) on page Disable target-dependent optimizations.
- Xkill-reorder (all) on page Disable target-independent optimizations.
- Xlimit-reaching (all) on page Disable reaching analysis optimizations.
- Xlno (all) on page Enable loop nest optimizations.
- Xlocals-on-stack (all) on page Do not assign locals to registers.
- Xno-builtin (all) on page Disable Diab built-in functions.
- Xno-if-conversion (all) on page Disable optimization of conditional jumps.
- Xno-wpo-dead (all) on page Disable whole program dead code and data elimination.
- XO (all) on page Enable extra optimizations.
- Xopt-count (all) on page Execute the compiler's optimizing stage n times.
- Xoptimistic-aliasing (all) on page Permit more optimizations.
- Xparse-count (all) on page Specify optimization buffer size.
- Xpipeline-short-circuit (TriCore) on page Optimize evaluation of logical operators && and ||.
- Xreduce-reaching (all) on page Reduce reaching analysis optimizations.
- Xrelax-alias (all) on page Control aliasing rules applied by the compiler.
- Xrestart (all) on page Restart optimization from scratch.
- Xreverse-inline (all) on page Optimize code by moving some code to new functions.

- Xsize-opt (all) on page Optimize for size rather than speed.
- Xstack-delay, -Xstack-delay-off (all) on page Delay popping stack after function call.
- Xstruct-assign-split... (all) on page Control optimization of structure member assignments.
- Xswitch-array-in-function-section (all) on page Put any compiler-generated switch tables in the same section as the parent function.
- Xswitch-array-off (all) on page Disables conversion of switch to look-up array
- Xswitch-table (all) on page Set threshold for a switch statement table.
- Xswitch-table-in-function-section (all) on page Put any compiler-generated switch tables in the same section as the parent function.
- Xswitch-table-off (all) on page Disable jump tables for switch statements.
- Xtest-at-both (all) on page Loop tests at top and bottom.
- Xtest-at-bottom (all) on page Loop test at bottom.
- Xtest-at-top (all) on page Loop test at top.
- Xunroll (all) on page Set loop unrolling count.
- Xunroll-register-rename (all) on page Allow scalar variable to occupy multiple registers for unrolled loops.
- Xunroll-size (all) on page Set loop unrolling size.
- Xvec-aligned (all) on page (MIPS64GLRN only) Inform compiler that arrays are properly aligned for vectorization.
- Xvec (all) on page (MIPS64GLRN only) Enable auto-vectorization.
- Xvec-args-aligned (all) on page (MIPS64GLRN only) Assume that function array/pointer arguments are aligned for vector memory access.
- Xvec-verbose (all) on page (MIPS64GLRN only) Control diagnostic output for auto-vectorization.
- Xwhole-program-diagnostics (all) on page Set whole-program optimization diagnostics.
- Xwhole-program-optim (all) on page Use whole-program optimization.
- Xwpo-const (all) on page Enable whole program auto-constant optimization.
- Xwpo-dead (all) on page Enable whole program dead code and data elimination.
- Xwpo-delete-return-value (all) on page Enable whole program return value value deletion optimization.
- Xwpo-inline (all) on page Enable cross-module inlining.
- Xwpo-inline-single-calls (all) on page Always inline functions that are only called once (regardless of size).
- Xwpo-no-alias (all) on page Enable whole program unaliased variable optimization.

3.13. Output Index

- Xasm-const-pound... (ColdFire) on page Disable '::' prefix for assembly numeric constants.
- Xcode-relative... (all) on page Generate position-independent code (PIC).

- Xcpp-cr-line-terminator (all) on page Enable line terminator behavior of carriage return (CR) characters.
- Xcpp-dump-symbols (all) on page Dump symbol information for macros or assertions.
- Xdata-relative... (all) on page Generate position-independent data (PID).
- Xident-off (all) on page Do not pass #ident strings.
- Xincfile-missing-ignore (all) on page Ignore missing include files.
- Xmacro-patterns (all) on page Use extended preprocessor syntax.
- Xmake-dependency (all) on page Show make rules.
- Xmake-dependency-canonicalize-path-off (all) on page Disable translation of path separators.
- Xmake-dependency-savefile (all) on page Specify dependency output file.
- Xmake-dependency-target (all) on page Specify dependency name.
- Xmetadata (all) on page Activate or deactivate metadata generation.
- Xpass-source (all) on page Output source as comments.
- Xpic (all) on page Generate position-independent code for shared libraries.
- Xpreprocessor-lineno-off (all) on page Suppress line numbers in preprocessor output.
- Xsection-split (all) on page Generate data or functions in separate section classes.
- Xsection-split-name (all) on page Specify name pattern for split sections.
- Xsect-pri-off (all) on page Disable generation of priority section names.
- Xsect-pri-on (all) on page Enable generation of priority section names.
- Xshow-configuration (all) on page Control listing of -X options in assembly output.
- Xunderscore-leading (all) on page Prefix identifiers with underscore.
- Xunderscore-surround (all) on page Add underscores to identifier prefix and suffix.
- Xunderscore-trailing (all) on page Add underscore to identifier suffix.
- Xuser-metadata (all) on page Add key-value pairs to the metadata of an object file.

3.14. Sections Index

- Xabsolute18-const, -Xabsolute18-data (all) on page Take advantage of the 18-bit absolute addressing mode supported on TriCore hardware.
- Xaddr-code (all) on page Set addressing mode for code sections.
- Xaddr-const (all) on page Set addressing mode for const sections.
- Xaddr-data (all) on page Set addressing mode for data sections.
- Xaddr-sconst (all) on page Set addressing mode for sconst sections.

- [-Xaddr-sdata \(all\) on page](#) Set addressing mode for sdata sections.
- [-Xaddr-string \(all\) on page](#) Set addressing mode for string sections.
- [-Xaddr-user \(all\) on page](#) Set addressing mode for user-defined sections.
- [-Xalign-functions \(all\) on page](#) Align functions on n-byte boundaries.
- [-Xargs-in-regs \(all\) on page](#) Pass argument in register.
- [-Xbss-off, -Xbss-common-off \(all\) on page](#) Control allocation of uninitialized variables in "COMMON" and bss sections.
- [-Xbss-zero-init-const-on, -Xbss-zero-init-data-off \(all\) on page](#) Control movement of variables initialized to zero to BSS.
- [-Xcode-absolute... \(all\) on page](#) Use absolute addressing for code.
- [-Xconst-in-... \(all\) on page](#) Locate constants with "text" or "data".
- [-Xdata-absolute... \(all\) on page](#) Use absolute addressing for data.
- [-Xdebug-align \(all\) on page](#) Align .debug sections
- [-Xlocal-data-area \(all\) on page](#) Allocate static and global variables to local data area.
- [-Xlocal-data-area-static-only \(all\) on page](#) Restrict local data area optimization to static variables.
- [-Xname-... \(all\) on page](#) Specify section name.
- [-Xpragma-section-first, -Xpragma-section-last \(all\) on page](#) Control interpretation of multiple section pragmas.
- [-Xsection-pad \(all\) on page](#) Pad sections for optimized loading.
- [-Xsmall-const \(all\) on page](#) Set size limit for "small const" variable.
- [-Xsmall-const-registers \(all\) on page](#) Extend small constant areas.
- [-Xsmall-data \(all\) on page](#) Set size limit for "small data" variables.
- [-Xsmall-data-registers \(all\) on page](#) Extend small data areas.

3.15. Syntax Index

- [-Xbottom-up-init \(all\) on page](#) Parse initial values bottom-up.
- [-Xc-new \(all\) on page](#) Use the 'new' compiler frontend.
- [-Xc-old \(all\) on page](#) Use legacy DIAB front end (ctoa).
- [-Xcpp-no-space \(all\) on page](#) Suppress preprocessor spacing.
- [-Xdialect... \(all\) on page](#) Specify C dialect.
- [-Xdisable-keywords \(all\) on page](#) Disable keywords selectively.
- [-Xdollar-in-ident \(all\) on page](#) Allow dollar signs in identifiers.
- [-Ximport \(all\) on page](#) Treat #include as #import.

- Xinline-without-body-warn (all) on page Warn about inlined functions with no definition.
- Xintrinsic-mask, -Xdisable-intrinsic-mask (all) on page Enable and disable specific intrinsic functions.
- Xkeywords (all) on page Enable extended keywords.
- Xmacro-in-pragma (all) on page Expand macros in pragmas.
- Xmode-64bit (MIPS) on page Allow MIPS64 assembler syntax.
- Xmode-64bit (MIPS16) on page Allow MIPS64 assembler syntax.
- Xno-gnu-inline-asm (all) on page (ARM, PPC, TC, RH850 only) Turn off extended GNU inline assembly extension use
- Xpreprocessor-old (all) on page Use old preprocessor.
- Xswap-cr-nl (all) on page Swap '\n' and '\r' in constants.
- Xsystem-headers-warn (all) on page Disable certain system header warnings.
- Xtruncate (all) on page Truncate all identifiers after m characters.
- Xuse-fpsimd (RH850) on page Enable vector data types and intrinsics for RH850G4MH
- Xvoid-ptr-arith-ok (all) on page Void pointer arithmetic.

3.16. Type Index

- Xbit-fields-access-as-byte, -Xbit-fields-access-as-type (all) on page Disable bit-field access optimization.
- Xbit-fields-compress (all) on page Control bit-field storage.
- Xbit-fields-signed, -Xbit-fields-unsigned (all) on page Specify sign of plain bit-field.
- Xcharset-ascii (all) on page Generate code using ASCII character set.
- Xchar-signed, -Xchar-unsigned (all) on page Specify sign of plain char.
- Xdouble-avoid (all) on page Force single precision.
- Xdouble-error (all) on page Generate error for double precision operations.
- Xdouble-warning (all) on page Warn about double precision operations.
- Xenum-is-... (all) on page Specify enum type.
- Xextend-args (all) on page Force precision of real arguments.
- Xfp-fast, -Xfp-normal, -Xfp-pedantic (all) on page Specify degree of conformance to the IEEE754 standard.
- Xfp-long-double-off, -Xfp-float-only (all) on page Convert double and long double.
- Xfp-min-prec... (all) on page Specify minimum floating point precision.
- Xieee754-pedantic (all) on page Enable strict implementation of IEEE754 floating point standard.
- Xold-inline-asm-casting (all) on page Use old inline assembly casting.

`-Xstrict-bitfield-promotions` (all) on page Ignore sign when promoting bit-fields.