

MULTI: C and C++ Compiler Error Messages



Green Hills Software
30 West Sola Street
Santa Barbara, California 93101
USA
Tel: 805-965-6044
Fax: 805-965-6343
www.ghs.com

DISCLAIMER

GREEN HILLS SOFTWARE MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. Further, Green Hills Software reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Green Hills Software to notify any person of such revision or changes.

Copyright © 1983-2013 by Green Hills Software. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission from Green Hills Software.

Green Hills, the Green Hills logo, CodeBalance, GMART, GSTART, INTEGRITY, MULTI, and Slingshot are registered trademarks of Green Hills Software. AdaMULTI, Built with INTEGRITY, EventAnalyzer, G-Cover, GHnet, GHnetLite, Green Hills Probe, Integrate, ISIM, u-velOSity, PathAnalyzer, Quick Start, ResourceAnalyzer, Safety Critical Products, SuperTrace Probe, TimeMachine, TotalDeveloper, DoubleCheck, and velOSity are trademarks of Green Hills Software.

All other company, product, or service names mentioned in this book may be trademarks or service marks of their respective owners.

For a partial listing of Green Hills Software and periodically updated patent marking information, please visit http://www.ghs.com/copyright_patent.html.

PubID: c_error_ref-496213

Branch: <http://toolsvc/branches/release-branch-2013-5-bto>

Date: October 4, 2013

Contents

1. C and C++ Compiler Error Messages	1
Error Message Codes	2
Errors 1 to 99	2
Errors 100 to 199	6
Errors 200 to 299	10
Errors 300 to 399	13
Errors 400 to 499	16
Errors 500 to 599	19
Errors 600 to 699	23
Errors 700 to 799	26
Errors 800 to 899	30
Errors 900 to 999	34
Errors 1000 to 1099	38
Errors 1100 to 1199	42
Errors 1200 to 1299	45
Errors 1300 to 1399	49
Errors 1400 to 1499	54
Errors 1500 to 1599	58
Errors 1600 to 1699	63
Errors 1700 to 1799	64
Errors 1800 to 1899	69
Errors 1900 to 1999	74
Errors 2000 to 2099	79

Chapter 1

C and C++ Compiler Error Messages

Contents

Error Message Codes	2
---------------------------	---

This chapter lists error messages which you may encounter when compiling C or C++ code, listed by error number. For information about how to suppress these messages, see “Set Message to Silent (**--diag_suppress**)” in Chapter 3, “Builder and Driver Options” in the *MULTI: Building Applications* book.

Error Message Codes

The following list contains all the C and C++ error messages:

Errors 1 to 99

1	last line of file ends without a newline
2	last line of file ends with a backslash
3	#include file "xxxx" includes itself For more information, see “C++ Implementation Defined Features” in “Green Hills C++” in the <i>MULTI: Building Applications</i> book.
4	out of memory
5	could not open source file "xxxx"
6	comment unclosed at end of file
7	unrecognized token
8	missing closing quote
9	nested comment is not allowed GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the <i>MULTI: Building Applications</i> book. You cannot nest comments inside one another. For example, the following line will cause the compiler to issue a warning: <pre>/* this is /*a comment*/ not a comment */</pre> Because the first instance of */ closes the comment, the compiler parses not a comment as C code.
10	"#" not expected here

11	unrecognized preprocessing directive GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book. The compiler issues this warning when it does not recognize a preprocessing directive. For example: <pre>#foo</pre>
12	parsing restarts here after previous syntax error
13	expected a file name
14	extra text after expected end of preprocessing directive
16	"xxxx" is not a valid source file name
17	expected a "]"
18	expected a ")"
19	extra text after expected end of number
20	identifier "xxxx" is undefined
21	type qualifiers are meaningless in this declaration
22	invalid hexadecimal number
23	integer constant is too large
24	invalid octal digit
25	quoted string should contain at least one character
26	too many characters in character constant
27	character value is out of range
28	expression must have a constant value
29	expected an expression
30	floating constant is out of range
31	expression must have integral type
32	expression must have arithmetic type
33	expected a line number
34	invalid line number
35	#error directive: xxxx

36	the #if for this directive is missing GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
37	the #endif for this directive is missing GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
38	directive is not allowed -- an #else has already appeared
39	division by zero
40	expected an identifier
41	expression must have arithmetic or pointer type
42	operand types are incompatible ("type" and "type")
44	expression must have pointer type
45	#undef may not be used on this predefined name "entity"
46	"entity" is predefined; attempted redefinition ignored
47	incompatible redefinition of macro "entity" (declared at line xxxx)
49	duplicate macro parameter name
50	"##" may not be first in a macro definition
51	"##" may not be last in a macro definition
52	expected a macro parameter name
53	expected a ":"
54	too few arguments in macro invocation GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
55	too many arguments in macro invocation
56	operand of sizeof may not be a function
57	this operator is not allowed in a constant expression
58	this operator is not allowed in a preprocessing expression
59	function call is not allowed in a constant expression
60	this operator is not allowed in an integral constant expression
61	integer operation result is out of range
62	shift count is negative

63	shift count is too large GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
64	declaration does not declare anything
65	expected a ";"
66	enumeration value is out of "int" range GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
67	expected a "}"
68	integer conversion resulted in a change of sign
69	integer conversion resulted in truncation GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
70	incomplete type is not allowed
71	operand of sizeof may not be a bit field
75	operand of "*" must be a pointer
76	argument to macro is empty
77	this declaration has no storage class or type specifier GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
78	a parameter declaration may not have an initializer
79	expected a type specifier
80	a storage class may not be specified here
81	more than one storage class may not be specified
82	storage class is not first
83	type qualifier specified more than once
84	invalid combination of type specifiers
85	invalid storage class for a parameter
86	invalid storage class for a function
87	a type specifier may not be used here
88	array of functions is not allowed
89	array of void is not allowed

90	function returning function is not allowed
91	function returning array is not allowed
92	identifier-list parameters may only be used in a function definition
93	function type may not come from a typedef
94	the size of an array must be greater than zero
95	array is too large
96	a translation unit must contain at least one declaration
97	a function may not return a value of this type
98	an array may not have elements of this type
99	a declaration here must declare a parameter

Errors 100 to 199

100	duplicate parameter name
101	"xxxx" has already been declared in the current scope
102	forward declaration of enum type is nonstandard
103	class is too large
104	struct or union is too large
105	invalid size for bit field
106	invalid type for a bit field
107	zero-length bit field must be unnamed
108	signed bit field of length 1 GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
109	expression must have (pointer-to-) function type
110	expected either a definition or a tag name
111	statement is unreachable
112	expected "while"
114	entity-kind "entity" was referenced but not defined
115	a continue statement may only be used within a loop
116	a break statement may only be used within a loop or switch

117	non-void entity-kind "entity" should return a value GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
118	a void function may not return a value
119	cast to type "type" is not allowed
120	return value type does not match the function type
121	a case label may only be used within a switch
122	a default label may only be used within a switch
123	case label value has already appeared in this switch at line xxxx
124	default label has already appeared in this switch at line xxxx
125	expected a "("
126	expression must be an lvalue
127	expected a statement
128	loop is not reachable from preceding code
129	a block-scope function may only have extern storage class
130	expected a "{"
131	expression must have pointer-to-class type
132	expression must have pointer-to-struct-or-union type
133	expected a member name
134	expected a field name
135	entity-kind "entity" has no member "xxxx"
136	entity-kind "entity" has no field "xxxx"
137	expression must be a modifiable lvalue
138	taking the address of a register variable is not allowed
139	taking the address of a bit field is not allowed
140	too many arguments in function call
141	unnamed prototyped parameters not allowed when body is present
142	expression must have pointer-to-object type
143	program too large or complicated to compile
144	a value of type "type" cannot be used to initialize an entity of type "type"

145	entity-kind "entity" may not be initialized
146	too many initializer values
147	declaration is incompatible with entity-kind "entity" (declared at line xxxx)
148	entity-kind "entity" has already been initialized
149	a global-scope declaration may not have this storage class
150	a type name may not be redeclared as a parameter
151	a typedef name may not be redeclared as a parameter
152	conversion of nonzero integer to pointer
153	expression must have class type
154	expression must have struct or union type
155	old-fashioned assignment operator
156	old-fashioned initializer
157	expression must be an integral constant expression
158	expression must be an lvalue or a function designator
159	declaration is incompatible with previous "entity" (declared at line xxxx)
160	external name conflicts with external name of entity-kind "entity" (declared at line xxxx)
161	unrecognized #pragma
163	could not open temporary file "xxxx"
164	name of directory for temporary files is too long ("xxxx")
165	too few arguments in function call
166	invalid floating constant
167	argument of type "type" is incompatible with parameter of type "type"
168	a function type is not allowed here
169	expected a declaration
170	pointer points outside of underlying object
171	invalid type conversion

172	external/internal linkage conflict with previous declaration GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
173	floating-point value does not fit in required integral type
174	expression has no effect GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
175	subscript out of range
177	entity-kind "entity" was declared but never referenced
178	"&" applied to an array has no effect
179	right operand of "%" is zero
180	argument is incompatible with formal parameter
181	argument is incompatible with corresponding format string conversion (expected "type", given "type") GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
182	could not open source file "xxxx" (no directories in search list)
183	type of cast must be integral
184	type of cast must be arithmetic or pointer
185	dynamic initialization in unreachable code
186	pointless comparison of unsigned integer with zero
187	use of "=" where "==" may have been intended GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
188	enumerated type mixed with another type
189	error while writing xxxx file
190	invalid intermediate language file
191	type qualifier is meaningless on cast type
192	unrecognized character escape sequence GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
193	zero used for undefined preprocessing identifier "xxxx"
194	expected an asm string

195	an asm function must be prototyped
196	an asm function may not have an ellipsis

Errors 200 to 299

219	error while deleting file "xxxx"
220	integral value does not fit in required floating-point type
221	floating-point value does not fit in required floating-point type
222	floating-point operation result is out of range
223	function xxxx declared implicitly GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
224	the format string requires additional arguments GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
225	the format string ends before this argument GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
226	invalid format string conversion GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
227	macro recursion
228	trailing comma is nonstandard
229	bit field cannot contain all values of the enumerated type
230	nonstandard type for a bit field
231	declaration is not visible outside of function
232	old-fashioned typedef of "void" ignored
233	left operand is not a struct or union containing this field
234	pointer does not point to struct or union containing this field
235	variable "xxxx" was declared with a never-completed type
236	controlling expression is constant

237	selector expression is constant GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
238	invalid specifier on a parameter
239	invalid specifier outside a class declaration
240	duplicate specifier in declaration
241	a union is not allowed to have a base class
242	multiple access control specifiers are not allowed
243	class or struct definition is missing
244	qualified name is not a member of class "type" or its base classes
245	a nonstatic member reference must be relative to a specific object
246	a nonstatic data member may not be defined outside its class
247	entity-kind "entity" has already been defined
248	pointer to reference is not allowed
249	reference to reference is not allowed
250	reference to void is not allowed
251	array of reference is not allowed
252	reference entity-kind "entity" requires an initializer
253	expected a ",", "
254	type name is not allowed
255	type definition is not allowed
256	invalid redeclaration of type name "entity" (declared at line xxxx)
257	const entity-kind "entity" requires an initializer
258	"this" may only be used inside a nonstatic member function
259	constant value is not known
260	explicit type is missing ("int" assumed) GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
261	access control not specified ("xxxx" by default)
262	not a class or struct name

263	duplicate base class name
264	invalid base class
265	entity-kind "entity" is inaccessible
266	"entity" is ambiguous
267	old-style parameter list (anachronism)
268	declaration may not appear after executable statement in block
269	conversion to inaccessible base class "type" is not allowed
274	improperly terminated macro invocation
276	name followed by "::" must be a class or namespace name
277	invalid friend declaration
278	a constructor or destructor may not return a value
279	invalid destructor declaration
280	declaration of a member with the same name as its class
281	global-scope qualifier (leading "::") is not allowed
282	the global scope has no "xxxx"
283	qualified name is not allowed
284	NULL reference is not allowed
285	initialization with "{...}" is not allowed for object of type "type"
286	base class "type" is ambiguous
287	derived class "type" contains more than one instance of class "type"
288	cannot convert pointer to base class "type" to pointer to derived class "type" -- base class is virtual
289	no instance of constructor "entity" matches the argument list
290	copy constructor for class "type" is ambiguous
291	no default constructor exists for class "type"
292	"xxxx" is not a nonstatic data member or base class of class "type"
293	indirect nonvirtual base class is not allowed
294	invalid union member -- class "type" has a disallowed member function

296	invalid use of non-lvalue array
297	expected an operator
298	inherited member is not allowed
299	cannot determine which instance of entity-kind "entity" is intended

Errors 300 to 399

300	a pointer to a bound function may only be used to call the function
301	typedef name has already been declared (with same type)
302	entity-kind "entity" has already been defined
304	no instance of entity-kind "entity" matches the argument list
305	type definition is not allowed in function return type declaration
306	default argument not at end of parameter list
307	redefinition of default argument
308	more than one instance of entity-kind "entity" matches the argument list:
309	more than one instance of constructor "entity" matches the argument list:
310	default argument of type "type" is incompatible with parameter of type "type"
311	cannot overload functions distinguished by return type alone
312	no suitable user-defined conversion from "type" to "type" exists
313	type qualifier is not allowed on this function
314	only nonstatic member functions may be virtual
315	the object has cv-qualifiers that are not compatible with the member function
316	program too large to compile (too many virtual functions)
317	return type is neither identical to nor covariant with return type "type" of overridden virtual function entity-kind "entity"
318	override of virtual entity-kind "entity" is ambiguous
319	pure specifier ("= 0") allowed only on virtual functions

320	badly-formed pure specifier (only "= 0" is allowed)
321	data member initializer is not allowed
322	object of abstract class type "type" is not allowed:
323	function returning abstract class "type" is not allowed:
324	duplicate friend declaration
325	inline specifier allowed on function declarations only
326	"inline" is not allowed
327	invalid storage class for an inline function
328	invalid storage class for a class member
329	local class member entity-kind "entity" requires a definition
330	entity-kind "entity" (declared at line xxxx) is inaccessible
332	class "type" has no copy constructor to copy a const object
333	defining an implicitly declared member function is not allowed
334	class "type" has no suitable copy constructor
335	linkage specification is not allowed
336	unknown external linkage specification
337	linkage specification is incompatible with previous "entity" (declared at line xxxx)
338	more than one instance of overloaded function "entity" has "C" linkage
339	class "type" has more than one default constructor
340	value copied to temporary, reference to temporary used
341	"operatorxxxx" must be a member function
342	operator may not be a static member function
343	no arguments allowed on user-defined conversion
344	too many parameters for this operator function
345	too few parameters for this operator function
346	nonmember operator requires a parameter with class type
347	default argument is not allowed
348	more than one user-defined conversion from "type" to "type" applies:
349	no operator "xxxx" matches these operands

350	more than one operator "xxxx" matches these operands:
351	first parameter of allocation function must be of type "size_t"
352	allocation function requires "void *" return type
353	deallocation function requires "void" return type
354	first parameter of deallocation function must be of type "void *"
356	type must be an object type
357	base class "type" has already been initialized
358	base class name required -- "type" assumed (anachronism)
359	entity-kind "entity" has already been initialized
360	name of member or base class is missing
361	assignment to "this" (anachronism)
362	"overload" keyword used (anachronism)
363	invalid anonymous union -- nonpublic member is not allowed
364	invalid anonymous union -- member function is not allowed
365	anonymous union at global or namespace scope must be declared static
366	entity-kind "entity" provides no initializer for:
367	implicitly generated constructor for class "type" cannot initialize:
368	entity-kind "entity" defines no constructor to initialize the following:
369	entity-kind "entity" has an uninitialized const or reference member
370	entity-kind "entity" has an uninitialized const field
371	class "type" has no assignment operator to copy a const object
372	class "type" has no suitable assignment operator
373	ambiguous assignment operator for class "type"
375	declaration requires a typedef name
377	"virtual" is not allowed
378	"static" is not allowed
379	cast of bound function to normal function pointer (anachronism)

380	expression must have pointer-to-member type
381	extra ";" ignored
382	in-class initializer for nonstatic member is nonstandard
384	no instance of overloaded "entity" matches the argument list
386	no instance of entity-kind "entity" matches the required type
387	delete array size expression used (anachronism)
389	a cast to abstract class "type" is not allowed:
390	function "main" may not be called or have its address taken
391	a new-initializer may not be specified for an array
392	member function "entity" may not be redeclared outside its class
393	pointer to incomplete class type is not allowed
394	reference to local variable of enclosing function is not allowed
395	single-argument function used for postfix "xxxx" (anachronism)
397	implicitly generated assignment operator cannot copy:
398	cast to array type is nonstandard (treated as cast to "type")
399	entity-kind "entity" has an operator new() but no default operator delete()

Errors 400 to 499

400	entity-kind "entity" has a default operator delete() but no operator new()
401	destructor for base class "entity" (declared at line xxxx) is not virtual
403	entity-kind "entity" has already been declared
404	function "main" may not be declared inline
405	member function with the same name as its class must be a constructor
406	using nested entity-kind "entity" (anachronism)
407	a destructor may not have parameters
408	copy constructor for class "type" may not have a parameter of type "type"
409	entity-kind "entity" returns incomplete type "type"

410	protected entity-kind "entity" (declared at line xxxx) is not accessible through a "type" pointer or object
411	a parameter is not allowed
412	an "asm" declaration is not allowed here
413	no suitable conversion function from "type" to "type" exists
414	delete of pointer to incomplete class
415	no suitable constructor exists to convert from "type" to "type"
416	more than one constructor applies to convert from "type" to "type":
417	more than one conversion function from "type" to "type" applies:
418	more than one conversion function from "type" to a built-in type applies:
424	a constructor or destructor may not have its address taken
426	temporary used for initial value of reference to non-const (anachronism)
427	qualified name is not allowed in member declaration
428	enumerated type mixed with another type (anachronism)
429	the size of an array in "new" must be non-negative
430	returning reference to local temporary
432	"enum" declaration is not allowed
433	qualifiers dropped in binding reference of type "type" to initializer of type "type"
434	a reference of type "type" (not const-qualified) cannot be initialized with a value of type "type"
435	a pointer to function may not be deleted
436	conversion function must be a nonstatic member function
437	template declaration is not allowed here
438	expected a "<"
439	expected a ">"
440	template parameter declaration is missing
441	argument list for entity-kind "entity" is missing
442	too few arguments for entity-kind "entity"
443	too many arguments for entity-kind "entity"

445	entity-kind "entity" is not used in declaring the parameter types of entity-kind "entity"
446	two nested types have the same name: "entity" and "entity" (declared at line xxxx) (cfront compatibility)
447	global "entity" was declared after nested "entity" (declared at line xxxx) (cfront compatibility)
449	more than one instance of entity-kind "entity" matches the required type
450	the type "long long" is nonstandard
451	omission of "xxxx" is nonstandard
452	return type may not be specified on a conversion function
456	excessive recursion at instantiation of entity-kind "entity" For more information, see “C++ Implementation Defined Features” in “Green Hills C++” in the <i>MULTI: Building Applications</i> book.
457	"xxxx" is not a function or static data member
458	argument of type "type" is incompatible with template parameter of type "type"
459	initialization requiring a temporary or conversion is not allowed
460	declaration of "xxxx" hides function parameter GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the <i>MULTI: Building Applications</i> book.
461	initial value of reference to non-const must be an lvalue
463	"template" is not allowed
464	"type" is not a class template
466	"main" is not a valid name for a function template
467	invalid reference to entity-kind "entity" (union/nonunion mismatch)
468	a template argument may not reference a local type
469	tag kind of xxxx is incompatible with declaration of entity-kind "entity" (declared at line xxxx)
470	the global scope has no tag named "xxxx"
471	entity-kind "entity" has no tag member named "xxxx"
472	member function typedef (allowed for cfront compatibility)

473	entity-kind "entity" may be used only in pointer-to-member declaration
475	a template argument may not reference a non-external entity
476	name followed by "::~" must be a class name or a type name
477	destructor name does not match name of class "type"
478	type used as destructor name does not match type "type"
479	entity-kind "entity" redeclared "inline" after being called
481	invalid storage class for a template declaration
482	entity-kind "entity" (declared at line xxxx) is an inaccessible type (allowed for cfront compatibility)
484	invalid explicit instantiation declaration
485	entity-kind "entity" is not an entity that can be instantiated
486	compiler generated entity-kind "entity" cannot be explicitly instantiated
487	inline entity-kind "entity" cannot be explicitly instantiated
489	entity-kind "entity" cannot be instantiated -- no template definition was supplied
490	entity-kind "entity" cannot be instantiated -- it has been explicitly specialized
493	no instance of entity-kind "entity" matches the specified type
494	declaring a void parameter list with a typedef is nonstandard
495	global entity-kind "entity" used instead of entity-kind "entity" (cfront compatibility)
496	template parameter "xxxx" may not be redeclared in this scope
497	declaration of "xxxx" hides template parameter
498	template argument list must match the parameter list

Errors 500 to 599

500	extra parameter of postfix "operator++" must be of type "int"
501	an operator name must be declared as a function
502	operator name is not allowed
503	entity-kind "entity" cannot be specialized in the current scope

504	nonstandard form for taking the address of a member function
505	too few template parameters -- does not match previous declaration
506	too many template parameters -- does not match previous declaration
507	function template for operator delete(void *) is not allowed
508	class template and template parameter may not have the same name
510	a template argument may not reference an unnamed type
511	enumerated type is not allowed
512	type qualifier on a reference type is not allowed
513	a value of type "type" cannot be assigned to an entity of type "type"
514	pointless comparison of unsigned integer with a negative constant
515	cannot convert to incomplete class "type"
516	const object requires an initializer
517	object has an uninitialized const or reference member
518	nonstandard preprocessing directive
519	entity-kind "entity" may not have a template argument list
520	initialization with "{...}" expected for aggregate object
521	pointer-to-member selection class types are incompatible ("type" and "type")
522	pointless friend declaration
523	"." used in place of "::" to form a qualified name
524	non-const function called for const object (anachronism)
525	a dependent statement may not be a declaration
526	a parameter may not have void type
529	this operator is not allowed in a template argument expression
530	try block requires at least one handler
531	handler requires an exception declaration
532	handler is masked by default handler
533	handler is potentially masked by previous handler for type "type"
534	use of a local type to specify an exception

535	redundant type in exception specification
536	exception specification is incompatible with that of previous entity-kind "entity" (declared at line xxxx)
540	support for exception handling is disabled
541	omission of exception specification is incompatible with previous entity-kind "entity" (declared at line xxxx)
542	could not create instantiation request file "xxxx"
543	non-arithmetic operation not allowed in nontype template argument
544	use of a local type to declare a nonlocal variable
545	use of a local type to declare a function
546	transfer of control bypasses initialization of:
548	transfer of control into an exception handler
549	entity-kind "entity" is used before its value is set GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
550	entity-kind "entity" was set but never used
551	entity-kind "entity" cannot be defined in the current scope
552	exception specification is not allowed
553	external/internal linkage conflict for entity-kind "entity" (declared at line xxxx)
554	entity-kind "entity" will not be called for implicit or explicit conversions
555	tag kind of xxxx is incompatible with template parameter of type "type"
556	function template for operator new(size_t) is not allowed
558	pointer to member of type "type" is not allowed
559	ellipsis is not allowed in operator function parameter list
560	"entity" is reserved for future use as a keyword
561	invalid macro definition:
562	invalid macro undefinition:
563	invalid preprocessor output file
564	cannot open preprocessor output file
565	IL file name must be specified if input is

566	invalid IL output file
567	cannot open IL output file
568	invalid C output file
569	cannot open C output file
570	error in debug option argument
571	invalid option:
572	back end requires name of IL file
573	could not open IL file
574	invalid number:
575	incorrect host CPU id
576	invalid instantiation mode:
578	invalid error limit:
579	invalid raw-listing output file
580	cannot open raw-listing output file
581	invalid cross-reference output file
582	cannot open cross-reference output file
583	invalid error output file
584	cannot open error output file
585	virtual function tables can only be suppressed when compiling C++
586	anachronism option can be used only when compiling C++
587	instantiation mode option can be used only when compiling C++
588	automatic instantiation mode can be used only when compiling C++
589	implicit template inclusion mode can be used only when compiling C++
590	exception handling option can be used only when compiling C++
591	strict mode is incompatible with K&R mode
592	strict mode is incompatible with cfront mode
593	missing source file name
594	output files may not be specified when compiling several input files
595	too many arguments on command line

596	an output file was specified, but none is needed
597	IL display requires name of IL file
598	a template parameter may not have void type
599	excessive recursive instantiation of entity-kind "entity" due to instantiate-all mode

Errors 600 to 699

600	strict mode is incompatible with allowing anachronisms
601	a throw expression may not have void type
602	local instantiation mode is incompatible with automatic instantiation
603	parameter of abstract class type "type" is not allowed:
604	array of abstract class "type" is not allowed:
605	floating-point template parameter is nonstandard
606	this pragma must immediately precede a declaration
607	this pragma must immediately precede a statement
608	this pragma must immediately precede a declaration or statement
609	this kind of pragma may not be used here
611	overloaded virtual function "entity" is only partially overridden in entity-kind "entity"
612	specific definition of inline template function must precede its first use
613	invalid error tag in diagnostic control option:
614	invalid error number in diagnostic control option:
615	parameter type involves pointer to array of unknown bound
616	parameter type involves reference to array of unknown bound
617	pointer-to-member-function cast to pointer to function
618	struct or union declares no named members
619	nonstandard unnamed field GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.

620	nonstandard unnamed member GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
624	"xxxx" is not a type name
631	unable to obtain mapped memory
641	"xxxx" is not a valid directory
642	cannot build temporary file name
643	"restrict" is not allowed
644	a pointer or reference to function type may not be qualified by "restrict"
645	"xxxx" is an unrecognized __declspec attribute
646	a calling convention modifier may not be specified here
647	conflicting calling convention modifiers
648	strict mode is incompatible with Microsoft mode
649	cfront mode is incompatible with Microsoft mode
650	calling convention specified here is ignored
651	a calling convention may not be followed by a nested declarator
652	calling convention is ignored for this type
654	declaration modifiers are incompatible with previous declaration
655	the modifier "xxxx" is not allowed on this declaration
656	transfer of control into a try block
657	inline specification is incompatible with previous "entity" (declared at line xxxx)
658	closing brace of template definition not found
659	wchar_t keyword option can be used only when compiling C++
660	invalid packing alignment value
661	expected an integer constant
662	call of pure virtual function
663	invalid source file identifier string
664	a class template cannot be defined in a friend declaration
665	"asm" is not allowed
666	"asm" must be used with a function definition

667	<code>"asm"</code> function is nonstandard
668	ellipsis with no explicit parameters is nonstandard
669	<code>"&..."</code> is nonstandard
670	invalid use of <code>"&..."</code>
672	temporary used for initial value of reference to <code>const volatile</code> (anachronism)
673	a reference of type <code>"type"</code> cannot be initialized with a value of type <code>"type"</code>
674	initial value of reference to <code>const volatile</code> must be an lvalue
675	SVR4 C compatibility option can be used only when compiling ANSI C
676	using out-of-scope declaration of entity-kind <code>"entity"</code> (declared at line <code>xxxx</code>)
677	strict mode is incompatible with SVR4 C mode
678	call of entity-kind <code>"entity"</code> (declared at line <code>xxxx</code>) cannot be inlined
679	entity-kind <code>"entity"</code> cannot be inlined
681	expected <code>__except</code> or <code>__finally</code>
682	a <code>__leave</code> statement may only be used within a <code>__try</code>
688	<code>"xxxx"</code> not found on pack alignment stack
689	empty pack alignment stack
690	RTTI option can be used only when compiling C++
691	entity-kind <code>"entity"</code> (declared at line <code>xxxx</code>), required for copy that was eliminated, is inaccessible
692	entity-kind <code>"entity"</code> (declared at line <code>xxxx</code>), required for copy that was eliminated, is not callable because reference parameter cannot be bound to rvalue
693	<code><typeinfo></code> must be included before <code>typeid</code> is used
694	<code>xxxx</code> cannot cast away <code>const</code> or other type qualifiers
695	the type in a <code>dynamic_cast</code> must be a pointer or reference to a complete class type, or <code>void *</code>
696	the operand of a pointer <code>dynamic_cast</code> must be a pointer to a complete class type

697	the operand of a reference <code>dynamic_cast</code> must be an lvalue of a complete class type
698	the operand of a runtime <code>dynamic_cast</code> must have a polymorphic class type
699	<code>bool</code> option can be used only when compiling C++

Errors 700 to 799

701	an array type is not allowed here
702	expected an "="
703	expected a declarator in condition declaration
704	"xxxx", declared in condition, may not be redeclared in this scope
705	default template arguments are not allowed for function templates
706	expected a ",", or ">"
707	expected a template parameter list
708	incrementing a <code>bool</code> value is deprecated
709	<code>bool</code> type is not allowed
710	offset of base class "entity" within class "entity" is too large
711	expression must have <code>bool</code> type (or be convertible to <code>bool</code>)
712	array new and delete option can be used only when compiling C++
713	entity-kind "entity" is not a variable name
714	<code>__based</code> modifier is not allowed here
715	<code>__based</code> does not precede a pointer operator, <code>__based</code> ignored
716	variable in <code>__based</code> modifier must have pointer type
717	the type in a <code>const_cast</code> must be a pointer, reference, or pointer to member to an object type
718	a <code>const_cast</code> can only adjust type qualifiers; it cannot change the underlying type
719	<code>mutable</code> is not allowed
720	redeclaration of entity-kind "entity" is not allowed to alter its access
721	nonstandard format string conversion

722	use of alternative token "<:" appears to be unintended
723	use of alternative token "%:" appears to be unintended
724	namespace definition is not allowed
725	name must be a namespace name
726	namespace alias definition is not allowed
727	namespace-qualified name is required
728	a namespace name is not allowed
729	invalid combination of DLL attributes
730	entity-kind "entity" is not a class template
731	array with incomplete element type is nonstandard
732	allocation operator may not be declared in a namespace
733	deallocation operator may not be declared in a namespace
734	entity-kind "entity" conflicts with using-declaration of entity-kind "entity"
735	using-declaration of entity-kind "entity" conflicts with entity-kind "entity" (declared at line xxxx)
736	namespaces option can be used only when compiling C++
737	using-declaration ignored -- it refers to the current namespace
738	a class-qualified name is required
742	entity-kind "entity" has no actual member "xxxx"
744	incompatible memory attributes specified
745	memory attribute ignored
746	memory attribute may not be followed by a nested declarator
747	memory attribute specified more than once
748	calling convention specified more than once
749	a type qualifier is not allowed
750	entity-kind "entity" (declared at line xxxx) was used before its template was declared
751	static and nonstatic member functions with same parameter types cannot be overloaded
752	no prior declaration of entity-kind "entity"
753	a template-id is not allowed

754	a class-qualified name is not allowed
755	entity-kind "entity" may not be redeclared in the current scope
756	qualified name is not allowed in namespace member declaration
757	entity-kind "entity" is not a type name
758	explicit instantiation is not allowed in the current scope
759	entity-kind "entity" cannot be explicitly instantiated in the current scope
760	entity-kind "entity" explicitly instantiated more than once
761	typename may only be used within a template
762	special_subscript_cost option can be used only when compiling C++
763	typename option can be used only when compiling C++
764	implicit typename option can be used only when compiling C++
765	nonstandard character at start of object-like macro definition
766	exception specification for virtual entity-kind "entity" is incompatible with that of overridden entity-kind "entity"
767	conversion from pointer to smaller integer
768	exception specification for implicitly declared virtual entity-kind "entity" is incompatible with that of overridden entity-kind "entity"
769	"entity", implicitly called from entity-kind "entity", is ambiguous
770	option "explicit" can be used only when compiling C++
771	"explicit" is not allowed

772	declaration conflicts with "xxxx" (reserved class name) The compiler emits this non-discretionary diagnostic whenever it encounters a definition of class <code>std::type_info</code> outside the standard header <code><typeinfo></code>. Instances of this class are created and exposed to the programmer via the C++ language operator <code>typeid</code>; therefore, a program that defines its own, incompatible version of class <code>std::type_info</code> may exhibit undefined behavior. To define your own version of class <code>std::type_info</code>, but suppress this diagnostic, precede the class definition with <code>#pragma define_type_info</code>. A correct definition of this class must preserve the member and vtable layout assumed by the compiler's <code>typeid</code> machinery. The following example is an acceptable definition: <pre>namespace std { #pragma define_type_info class type_info { virtual ~type_info(); }; }</pre> We recommend that you do not attempt to provide your own definition of this class. However, you might see this error if you are using standard C++ libraries other than the ones provided with your Green Hills compiler distribution.
773	only "()" is allowed as initializer for array entity-kind "entity"
774	"virtual" is not allowed in a function template declaration
775	invalid anonymous union -- class member template is not allowed
776	template nesting depth does not match the previous declaration of entity-kind "entity"
777	this declaration cannot have multiple "template <...>" clauses
778	option to control the for-init scope can be used only when compiling C++
779	"xxxx", declared in for-loop initialization, may not be redeclared in this scope
780	reference is to entity-kind "entity" (declared at line xxxx) -- under old for-init scoping rules it would have been entity-kind "entity" (declared at line xxxx)
781	option to control warnings on for-init differences can be used only when compiling C++
782	definition of virtual entity-kind "entity" is required here
783	empty comment interpreted as token-pasting operator "##"

784	a storage class is not allowed in a friend declaration
785	template parameter list for "entity" is not allowed in this declaration
786	entity-kind "entity" is not a valid member class or function template
787	not a valid member class or function template declaration
788	a template declaration containing a template parameter list may not be followed by an explicit specialization declaration
789	explicit specialization of entity-kind "entity" must precede the first use of entity-kind "entity"
790	explicit specialization is not allowed in the current scope
791	partial specialization of entity-kind "entity" is not allowed
792	entity-kind "entity" is not an entity that can be explicitly specialized
793	explicit specialization of entity-kind "entity" must precede its first use (at line xxxx)
794	template parameter "xxxx" may not be used in an elaborated type specifier
795	specializing entity-kind "entity" requires "template<>" syntax
798	option "old_specializations" can be used only when compiling C++
799	specializing entity-kind "entity" without "template<>" syntax is nonstandard

Errors 800 to 899

800	this declaration may not have extern "C" linkage
801	"xxxx" is not a class or function template name in the current scope
802	specifying a default argument when redeclaring an unreferenced function template is nonstandard
803	specifying a default argument when redeclaring an already referenced function template is not allowed
804	cannot convert pointer to member of base class "type" to pointer to member of derived class "type" -- base class is virtual
805	exception specification is incompatible with that of entity-kind "entity" (declared at line xxxx):

806	omission of exception specification is incompatible with entity-kind "entity" (declared at line xxxx)
807	unexpected end of default argument expression
808	default-initialization of reference is not allowed
809	uninitialized entity-kind "entity" has a const member
810	uninitialized base class "type" has a const member
811	const entity-kind "entity" requires an initializer -- class "type" has no explicitly declared default constructor
812	const object requires an initializer -- class "type" has no explicitly declared default constructor
813	option "implicit_extern_c_type_conversion" can be used only when compiling C++
814	strict mode is incompatible with long preserving rules
815	type qualifier on return type is meaningless
816	in a function definition a type qualifier on a "void" return type is not allowed
817	static data member declaration is not allowed in this class
818	template instantiation resulted in an invalid function declaration
819	"..." is not allowed
820	option "extern_inline" can be used only when compiling C++
821	extern inline entity-kind "entity" was referenced but not defined
822	invalid destructor name for type "type"
824	destructor reference is ambiguous -- both entity-kind "entity" and entity-kind "entity" could be used
825	virtual inline entity-kind "entity" was never defined
826	entity-kind "entity" was never referenced
827	only one member of a union may be specified in a constructor initializer list
828	support for "new[]" and "delete[]" is disabled
829	"double" used for "long double" in generated C code
830	entity-kind "entity" has no corresponding operator delete (to be called if an exception is thrown during initialization of an allocated object)

831	support for placement delete is disabled
832	no appropriate operator delete is visible
833	pointer or reference to incomplete type is not allowed
834	invalid partial specialization -- entity-kind "entity" is already fully specialized
835	incompatible exception specifications
836	returning reference to local variable
837	omission of explicit type is nonstandard ("int" assumed) GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
838	more than one partial specialization matches the template argument list of entity-kind "entity"
840	a template argument list is not allowed in a declaration of a primary template
841	partial specializations may not have default template arguments
842	entity-kind "entity" is not used in or cannot be deduced from the template argument list of entity-kind "entity"
844	the template argument list of the partial specialization includes a nontype argument whose type depends on a template parameter
845	this partial specialization would have been used to instantiate entity-kind "entity"
846	this partial specialization would have made the instantiation of entity-kind "entity" ambiguous
847	expression must have integral or enum type
848	expression must have arithmetic or enum type
849	expression must have arithmetic, enum, or pointer type
850	type of cast must be integral or enum
851	type of cast must be arithmetic, enum, or pointer
852	expression must be a pointer to a complete object type
854	a partial specialization nontype argument must be the name of a nontype parameter or a constant
855	return type is not identical to return type "type" of overridden virtual function entity-kind "entity"
856	option "guiding_decls" can be used only when compiling C++

857	a partial specialization of a class template must be declared in the namespace of which it is a member
858	entity-kind "entity" is a pure virtual function
859	pure virtual entity-kind "entity" has no overrider
860	<code>__declspec</code> attributes ignored
861	invalid character in input line
862	function returns incomplete type "type"
863	effect of this <code>"#pragma pack"</code> directive is local to entity-kind "entity"
864	xxxx is not a template
865	a friend declaration may not declare a partial specialization
866	exception specification ignored
867	declaration of "size_t" does not match the expected type "type"
868	space required between adjacent ">" delimiters of nested template argument lists (" $>>$ " is the right shift operator)
869	could not set locale "xxxx" to allow processing of multibyte characters
870	invalid multibyte character sequence
871	template instantiation resulted in unexpected function type of "type" (the meaning of a name may have changed since the template declaration -- the type of the template is "type")
872	ambiguous guiding declaration -- more than one function template "entity" matches type "type"
873	non-integral operation not allowed in nontype template argument
874	option "embedded_c++" can be used only when compiling C++
875	Embedded C++ does not support templates
876	Embedded C++ does not support exception handling
877	Embedded C++ does not support namespaces
878	Embedded C++ does not support run-time type information
879	Embedded C++ does not support the new cast syntax
880	Embedded C++ does not support using-declarations
881	Embedded C++ does not support "mutable"
882	Embedded C++ does not support multiple or virtual inheritance

883	invalid Microsoft version number:
884	pointer-to-member representation "xxxx" has already been set for entity-kind "entity"
885	"type" cannot be used to designate constructor for "type"
886	invalid suffix on integral constant
887	operand of __uidof must have a class or enum type for which __declspec(uid("...")) has been specified
888	invalid GUID string in __declspec(uid("..."))
889	option "vla" can be used only when compiling C
890	variable length array with unspecified bound is not allowed
891	an explicit template argument list is not allowed on this declaration
892	an entity with linkage cannot have a type involving a variable length array
893	a variable length array cannot have static storage duration
894	entity-kind "entity" is not a template
896	expected a template argument
898	nonmember operator requires a parameter with class or enum type
899	option "enum_overloading" can be used only when compiling C++

Errors 900 to 999

901	qualifier of destructor name "type" does not match type "type"
902	type qualifier ignored
903	option "nonstd_qualifier_deduction" can be used only when compiling C++
904	a function declared "dllimport" may not be defined
905	incorrect property specification; correct form is __declspec(property(get=name1,put=name2))
906	property has already been specified
907	__declspec(property) is not allowed on this declaration
908	member is declared with __declspec(property), but no "get" function was specified
909	the __declspec(property) "get" function "xxxx" is missing

910	member is declared with <code>__declspec(property)</code> , but no "put" function was specified
911	the <code>__declspec(property)</code> "put" function "xxxx" is missing
912	ambiguous class member reference -- entity-kind "entity" (declared at line xxxx) used in preference to entity-kind "entity" (declared at line xxxx)
913	missing or invalid segment name in <code>__declspec(allocate("..."))</code>
914	<code>__declspec(allocate)</code> is not allowed on this declaration
915	a segment name has already been specified
916	cannot convert pointer to member of derived class "type" to pointer to member of base class "type" -- base class is virtual
917	invalid directory for instantiation files:
918	option "one_instantiation_per_object" can be used only when compiling C++
919	invalid output file: "xxxx"
920	cannot open output file: "xxxx"
921	an instantiation information file name may not be specified when compiling several input files
922	option "one_instantiation_per_object" may not be used when compiling several input files
923	more than one command line option matches the abbreviation "--xxxx":
925	type qualifiers on function types are ignored
926	cannot open definition list file: "xxxx"
927	late/early tiebreaker option can be used only when compiling C++
928	incorrect use of <code>va_start</code>
929	incorrect use of <code>va_arg</code>
930	incorrect use of <code>va_end</code>
931	pending instantiations option can be used only when compiling C++
932	invalid directory for #import files:
933	an import directory can be specified only in Microsoft mode
934	a member with reference type is not allowed in a union
935	"typedef" may not be specified here

936	redeclaration of entity-kind "entity" alters its access
937	a class or namespace qualified name is required
938	return type "int" omitted in declaration of function "main" GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
939	pointer-to-member representation "xxxx" is too restrictive for entity-kind "entity"
940	missing return statement at end of non-void entity-kind "entity" GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
941	duplicate using-declaration of "entity" ignored
942	enum bit-fields are always unsigned, but enum "type" includes negative enumerator
943	option "class_name_injection" can be used only when compiling C++
944	option "arg_dep_lookup" can be used only when compiling C++
945	option "friend_injection" can be used only when compiling C++
946	name following "template" must be a template
948	nonstandard local-class friend declaration -- no prior declaration in the enclosing scope
949	specifying a default argument on this declaration is nonstandard
950	option "nonstd_using_decl" can be used only when compiling C++
951	return type of function "main" must be "int"
952	a nontype template parameter may not have class type
953	a default template argument cannot be specified on the declaration of a member of a class template outside of its class
954	a return statement is not allowed in a handler of a function try block of a constructor
955	ordinary and extended designators cannot be combined in an initializer designation
956	the second subscript must not be smaller than the first
957	option "designators" can be used only when compiling C
958	option "extended_designators" can be used only when compiling C

959	declared size for bit field is larger than the size of the bit field type; truncated to xxxx bits
960	type used as constructor name does not match type "type"
961	use of a type with no linkage to declare a variable with linkage
962	use of a type with no linkage to declare a function
963	return type may not be specified on a constructor
964	return type may not be specified on a destructor
965	incorrectly formed universal character name
966	universal character name specifies an invalid character
967	a universal character name cannot designate a character in the basic character set
968	this universal character is not allowed in an identifier
969	the identifier <code>__VA_ARGS__</code> can only appear in the replacement lists of variadic macros
970	the qualifier on this friend declaration is ignored
971	array range designators cannot be applied to dynamic initializers
972	property name cannot appear here
973	"inline" used as a function qualifier is ignored
974	option "compound_literals" can be used only when compiling C
975	a variable-length array type is not allowed
976	a compound literal is not allowed in an integral constant expression
977	a compound literal of type "type" is not allowed
978	a template friend declaration cannot be declared in a local class
979	ambiguous "?" operation: second operand of type "type" can be converted to third operand type "type", and vice versa
980	call of an object of a class type without appropriate operator() or conversion functions to pointer-to-function type
982	there is more than one way an object of type "type" can be called for the argument list:
983	typedef name has already been declared (with similar type)
984	operator new and operator delete cannot be given internal linkage
985	storage class "mutable" is not allowed for anonymous unions

987	abstract class type "type" is not allowed as catch type:
988	a qualified function type cannot be used to declare a nonmember function or a static member function
989	a qualified function type cannot be used to declare a parameter
990	cannot create a pointer or reference to qualified function type
991	extra braces are nonstandard GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
992	invalid macro definition:
993	subtraction of pointer types "type" and "type" is nonstandard
994	an empty template parameter list is not allowed in a template template parameter declaration
995	expected "class"
996	the "class" keyword must be used when declaring a template template parameter
997	entity-kind "entity" is hidden by "entity" -- virtual function override intended?
998	a qualified name is not allowed for a friend declaration that is a function definition
999	entity-kind "entity" is not compatible with entity-kind "entity"

Errors 1000 to 1099

1000	a storage class may not be specified here
1001	class member designated by a using-declaration must be visible in a direct base class
1003	Sun mode is incompatible with cfront mode
1004	strict mode is incompatible with Sun mode
1005	Sun mode is only allowed when compiling C++
1006	a template template parameter cannot have the same name as one of its template parameters
1007	recursive instantiation of default argument
1009	entity-kind "entity" is not an entity that can be defined
1010	destructor name must be qualified

1011	friend class name may not be introduced with "typename"
1012	a using-declaration may not name a constructor or destructor
1013	a qualified friend template declaration must refer to a specific previously declared template
1014	invalid specifier in class template declaration
1015	argument is incompatible with formal parameter
1016	option "dep_name" can be used only when compiling C++
1017	loop in sequence of "operator->" functions starting at class "type"
1018	entity-kind "entity" has no member class "xxxx"
1019	the global scope has no class named "xxxx"
1020	recursive instantiation of template default argument
1021	access declarations and using-declarations cannot appear in unions
1022	"entity" is not a class member
1023	nonstandard member constant declaration is not allowed
1024	option "ignore_std" can be used only when compiling C++
1025	option "parse_templates" can be used only when compiling C++
1026	option "dep_name" cannot be used with "no_parse_templates"
1027	language modes specified are incompatible
1028	invalid redeclaration of nested class
1029	type containing an unknown-size array is not allowed
1030	a variable with static storage duration cannot be defined within an inline function
1031	an entity with internal linkage cannot be referenced within an inline function with external linkage
1032	argument type "type" does not match this type-generic function macro
1034	friend declaration cannot add default arguments to previous declaration
1035	entity-kind "entity" cannot be declared in this scope
1036	the reserved identifier "xxxx" may only be used inside a function
1037	this universal character cannot begin an identifier

1038	expected a string literal
1039	unrecognized STDC pragma
1040	expected "ON", "OFF", or "DEFAULT"
1041	a STDC pragma may only appear between declarations in the global scope or before any statements or declarations in a block scope
1042	incorrect use of va_copy
1043	xxxx can only be used with floating-point types
1044	complex type is not allowed
1045	invalid designator kind
1046	floating-point value cannot be represented exactly
1047	complex floating-point operation result is out of range
1048	conversion between real and imaginary yields zero
1049	an initializer cannot be specified for a flexible array member
1050	imaginary *= imaginary sets the left-hand operand to zero
1051	standard requires that entity-kind "entity" be given a type by a subsequent declaration ("int" assumed)
1052	a definition is required for inline entity-kind "entity"
1053	conversion from integer to smaller pointer
1054	a floating-point type must be included in the type specifier for a _Complex or _Imaginary type
1055	types cannot be declared in anonymous unions
1056	returning pointer to local variable GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1057	returning pointer to local temporary
1058	option "export" can be used only when compiling C++
1059	option "export" cannot be used with "no_dep_name"
1060	option "export" cannot be used with "implicit_include"
1061	declaration of entity-kind "entity" is incompatible with a declaration in another translation unit
1062	the other declaration is at line xxxx
1065	a field declaration cannot have a type involving a variable length array

1066	declaration of entity-kind "entity" had a different meaning during compilation of "xxxx"
1067	expected "template"
1068	"export" cannot be used on an explicit instantiation
1069	"export" cannot be used on this declaration
1070	a member of an unnamed namespace cannot be declared "export"
1071	a template cannot be declared "export" after it has been defined
1072	a declaration cannot have a label
1073	support for exported templates is disabled
1074	cannot open exported template file: "xxxx"
1075	entity-kind "entity" already defined during compilation of "xxxx"
1076	entity-kind "entity" already defined in another translation unit
1077	a nonstatic local variable may not be used in a <code>__based</code> specification
1078	the option to list makefile dependencies may not be specified when compiling more than one translation unit
1080	the option to generate preprocessed output may not be specified when compiling more than one translation unit
1081	a field with the same name as its class cannot be declared in a class with a user-declared constructor
1082	"implicit_include" cannot be used when compiling more than one translation unit
1083	exported template file "xxxx" is corrupted
1084	entity-kind "entity" cannot be instantiated -- it has been explicitly specialized in the translation unit containing the exported definition
1086	the object has cv-qualifiers that are not compatible with the member entity-kind "entity"
1087	no instance of entity-kind "entity" matches the argument list and object (the object has cv-qualifiers that prevent a match)
1088	an attribute specifies a mode incompatible with "type"
1089	there is no type with the width specified
1090	invalid alignment value specified by attribute
1091	invalid attribute for "type"

1092	invalid attribute for entity-kind "entity"
1093	invalid attribute for parameter
1094	attribute "xxxx" does not take arguments
1096	expected an attribute name
1097	attribute "xxxx" ignored
1098	attributes may not appear here
1099	invalid argument to attribute "xxxx"

Errors 1100 to 1199

1100	the "packed" attribute is ignored in this context
1101	in "goto *expr", expr must have type "void *"
1102	"goto *expr" is nonstandard
1103	taking the address of a label is nonstandard
1104	file name specified more than once:
1105	#warning directive: xxxx
1106	attribute "xxxx" is only allowed in a function definition
1107	the "transparent_union" attribute only applies to unions, and "type" is not a union
1108	the "transparent_union" attribute is ignored on incomplete types
1109	"type" cannot be transparent because entity-kind "entity" does not have the same size as the first field
1110	"type" cannot be transparent because it has a field of type "type" which is not the same size as the first field
1111	only parameters can be transparent
1112	the "xxxx" attribute does not apply to local variables
1113	attributes are not permitted in a function definition
1114	declarations of local labels should only appear at the start of statement expressions
1115	the second constant in a case range must be larger than the first
1116	an asm name is not permitted in a function definition
1117	an asm name is ignored in a typedef
1118	unknown register name "xxxx"

1119	modifier letter "xxxx" ignored in asm operand
1120	unknown asm constraint modifier "xxxx"
1121	unknown asm constraint letter "xxxx"
1122	asm operand has no constraint letter
1123	an asm output operand must have one of the '=' or '+' modifiers
1124	an asm input operand may not have the '=' or '+' modifiers
1125	too many operands to asm statement (maximum is 100)
1126	too many colons in asm statement
1127	register "xxxx" used more than once
1128	register "xxxx" is both used and clobbered
1129	register "xxxx" clobbered more than once
1130	register "xxxx" has a fixed purpose and may not be used in an asm statement
1131	register "xxxx" has a fixed purpose and may not be clobbered in an asm statement
1132	an empty clobbers list must be omitted entirely
1133	expected an asm operand
1134	expected a register to clobber
1135	"format" attribute applied to entity-kind "entity" which does not have variable arguments
1136	first substitution argument is not the first variable argument
1137	format argument index is greater than number of parameters
1138	format argument does not have string type
1139	the "template" keyword used for syntactic disambiguation may only be used within a template
1142	attribute does not apply to non-function type "type"
1143	arithmetic on pointer to void or function type
1144	storage class must be auto or register
1145	"type" would have been promoted to "type" when passed through the ellipsis parameter; use the latter type instead
1146	"xxxx" is not a base class member
1147	__super cannot appear after "::"

1148	<code>__super</code> may only be used in a class scope
1149	<code>__super</code> must be followed by <code>::</code>
1151	mangled name is too long
1152	declaration aliased to undefined entity "xxxx"
1153	declaration does not match its alias entity-kind "entity"
1154	entity declared as alias cannot have definition
1155	variable-length array field type will be treated as zero-length array field type
1156	nonstandard cast on lvalue ignored
1157	unrecognized flag name
1158	void return type cannot be qualified
1159	the <code>auto</code> specifier is ignored here (invalid in standard C/C++)
1160	a reduction in alignment without the "packed" attribute is ignored
1161	a member template corresponding to "entity" is declared as a template of a different kind in another translation unit
1162	excess initializers are ignored GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1163	<code>va_start</code> should only appear in a function with an ellipsis parameter
1164	the "short_enums" option is only valid in GNU C and GNU C++ modes
1165	invalid export information file "xxxx" at line number xxxx
1166	statement expressions are only allowed in block scope
1168	an <code>asm</code> name is ignored on a non-register automatic variable
1169	inline function also declared as an alias; definition ignored
1170	unrecognized UPC pragma
1171	shared block size does not match one previously specified
1172	bracketed expression is assumed to be a block size specification rather than an array dimension
1173	the block size of a shared array must be greater than zero
1174	multiple block sizes not allowed
1175	strict or relaxed requires shared

1176	THREADS not allowed in this context
1177	block size specified exceeds the maximum value of xxxx
1178	function returning shared is not allowed
1180	one dimension of an array of a shared type must be a multiple of THREADS when the number of threads is nonconstant
1181	shared type inside a struct or union is not allowed
1182	parameters may not have shared types
1183	a dynamic THREADS dimension requires a definite block size
1184	shared variables must be static or extern
1185	argument of upc_blocksizeof is a pointer to a shared type (not shared type itself)
1186	affinity expression ignored in nested upc_forall
1187	branching into or out of a upc_forall loop is not allowed
1188	affinity expression must have a shared type or point to a shared type
1189	affinity has shared type (not pointer to shared)
1190	shared void* types can only be compared for equality
1191	UPC mode is incompatible with C++ and K&R modes
1192	null (zero) character in input line ignored
1193	null (zero) character in string or character constant
1194	null (zero) character in header name
1195	declaration in for-initializer hides a declaration in the surrounding scope
1196	the hidden declaration is at line xxxx
1197	the prototype declaration of entity-kind "entity" (declared at line xxxx) is ignored after this unprototyped redeclaration
1199	entity-kind "entity" (declared at line xxxx) must have external C linkage

Errors 1200 to 1299

1200	variable declaration hides declaration in for-initializer
1201	typedef "xxxx" may not be used in an elaborated type specifier

1202	call of zero constant ignored
1203	parameter "xxxx" may not be redeclared in a catch clause of function try block
1204	the initial explicit specialization of entity-kind "entity" must be declared in the namespace containing the template
1205	"cc" clobber ignored
1206	"template" must be followed by an identifier
1207	MYTHREAD not allowed in this context
1208	layout qualifier cannot qualify pointer to shared
1209	layout qualifier cannot qualify an incomplete array
1210	declaration of "xxxx" hides handler parameter
1211	nonstandard cast to array type ignored
1212	this pragma cannot be used in a _Pragma operator (a #pragma directive must be used)
1213	field uses tail padding of a base class
1214	GNU C++ compilers may use bit field padding
1215	entity-kind "entity" (declared at line xxxx) was declared "deprecated"
1216	an asm name is not allowed on a nonstatic member declaration
1217	unrecognized format function type "xxxx" ignored
1218	base class "entity" uses tail padding of base class "entity"
1219	the "init_priority" attribute can only be used for definitions of static data members and namespace scope variables of class types
1220	requested initialization priority is reserved for internal use
1221	this anonymous union/struct field is hidden by entity-kind "entity" (declared at line xxxx)
1222	invalid error number
1223	invalid error tag
1224	expected an error number or error tag
1225	size of class is affected by tail padding
1226	labels can be referenced only in function definitions
1227	transfer of control into a statement expression is not allowed

1229	this statement is not allowed inside of a statement expression
1230	a non-POD class definition is not allowed inside of a statement expression
1231	destructible entities are not allowed inside of a statement expression
1232	a dynamically-initialized local static variable is not allowed inside of a statement expression
1233	a variable-length array is not allowed inside of a statement expression
1234	a statement expression is not allowed inside of a default argument
1235	nonstandard conversion between pointer to function and pointer to data
1236	interface types cannot have virtual base classes
1237	interface types cannot specify "private" or "protected"
1238	interface types can only derive from other interface types
1239	"type" is an interface type
1240	interface types cannot have typedef members
1241	interface types cannot have user-declared constructors or destructors
1242	interface types cannot have user-declared member operators
1243	interface types cannot be declared in functions
1244	cannot declare interface templates
1245	interface types cannot have data members
1246	interface types cannot contain friend declarations
1247	interface types cannot have nested classes
1248	interface types cannot be nested class types
1249	interface types cannot have member templates
1250	interface types cannot have static member functions
1251	this pragma cannot be used in a <code>__pragma</code> operator (a <code>#pragma</code> directive must be used)
1252	qualifier must be base class of "type"
1253	declaration must correspond to a pure virtual member function in the indicated base class

1254	integer overflow in internal computation due to size or complexity of "type" The compiler displays this error if you are using multiple or virtual inheritance, and the offset from the beginning of the derived class <i>type</i> to a base class within <i>type</i> exceeds the 32 kilobytes. If you are using multiple inheritance, rearrange the inheritance order by putting smaller classes first. With the new ordering, the offset of larger classes is less likely to exceed this limit. If you are using virtual inheritance, make the derived class members smaller, because the compiler always places members of a derived class before virtual base classes. If you do not need your code to be backwards-compatible with existing code, you can remove this limit by passing the --large_vtbl_offsets option. For more information, see “Virtual Function Table Offset Size” in the chapter “Builder and Driver Options” in <i>MULTI: Building Applications</i> .
1255	integer overflow in internal computation
1256	<code>__w64</code> can only be specified on int, long, and pointer types
1257	potentially narrowing conversion when compiled in an environment where int, long, or pointer types are 64 bits wide
1258	current value of pragma pack is xxxx
1259	arguments for pragma pack(show) are ignored
1260	invalid alignment specifier value
1261	expected an integer literal
1262	earlier <code>__declspec(align(...))</code> ignored
1263	expected an argument value for the "xxxx" attribute parameter
1264	invalid argument value for the "xxxx" attribute parameter
1265	expected a boolean value for the "xxxx" attribute parameter
1266	a positional argument cannot follow a named argument in an attribute
1267	attribute "xxxx" has no parameter named "xxxx"
1268	expected an argument list for the "xxxx" attribute
1269	expected a ",", or "]"
1270	attribute argument "xxxx" has already been given a value
1271	a value cannot be assigned to the "xxxx" attribute
1272	a throw expression may not have pointer-to-incomplete type
1273	alignment-of operator applied to incomplete type
1274	"xxxx" may only be used as a standalone attribute

1275	"xxxx" attribute cannot be used here
1276	unrecognized attribute "xxxx"
1277	attributes are not allowed here
1278	invalid argument value for the "xxxx" attribute parameter
1279	too many attribute arguments
1280	conversion from inaccessible base class "type" is not allowed
1281	option "export" requires distinct template signatures
1282	string literals with different character kinds cannot be concatenated
1283	GNU layout bug not emulated because it places virtual base "entity" outside "entity" object boundaries
1284	virtual base "entity" placed outside "entity" object boundaries
1285	nonstandard qualified name in namespace member declaration
1286	reduction in alignment ignored
1287	const qualifier ignored
1289	invalid GNU asm qualifiers
1290	non-POD class type passed through ellipsis
1291	a non-POD class type cannot be fetched by va_arg
1292	the 'u' or 'U' suffix must appear before the 'l' or 'L' suffix in a fixed-point literal
1293	option "fixed_point" can be used only when compiling C
1294	integer operand may cause fixed-point overflow
1295	fixed-point constant is out of range
1296	fixed-point value cannot be represented exactly
1297	constant is too large for long long; given unsigned long long type (nonstandard)
1298	layout qualifier cannot qualify pointer to shared void
1299	duplicate THREADS in multidimensional array type

Errors 1300 to 1399

1300	a strong using-directive may only appear in a namespace scope
-------------	---

1301	entity-kind "entity" declares a non-template function -- add <> to refer to a template instance
1302	operation may cause fixed-point overflow
1303	expression must have integral, enum, or fixed-point type
1304	expression must have integral or fixed-point type
1305	function declared with "noreturn" does return
1306	asm name ignored because it conflicts with a previous declaration
1307	class member typedef may not be redeclared
1308	taking the address of a temporary
1309	attributes are ignored on a class declaration that is not also a definition
1310	fixed-point value implicitly converted to floating-point type
1311	fixed-point types have no classification
1312	a template parameter may not have fixed-point type
1313	hexadecimal floating-point constants are not allowed
1314	option "named_address_spaces" can be used only when compiling C
1315	floating-point value does not fit in required fixed-point type
1316	value cannot be converted to fixed-point value exactly
1317	fixed-point conversion resulted in a change of sign
1318	integer value does not fit in required fixed-point type
1319	fixed-point operation result is out of range
1320	multiple named address spaces
1321	variable with automatic storage duration cannot be stored in a named address space
1322	type cannot be qualified with named address space
1323	function type cannot be qualified with named address space
1324	field type cannot be qualified with named address space
1325	fixed-point value does not fit in required floating-point type
1326	fixed-point value does not fit in required integer type
1327	value does not fit in required fixed-point type
1328	option "named_registers" can be used only when compiling C
1329	a named-register storage class is not allowed here

1330	entity-kind "entity" (declared at line xxxx) redeclared with incompatible named-register storage class
1331	named-register storage class cannot be specified for aliased variable
1332	named-register storage specifier is already in use
1333	option "embedded_c" cannot be combined with options to control individual Embedded C features
1334	invalid EDG_BASE directory:
1335	cannot open predefined macro file: "xxxx"
1336	invalid predefined macro entry at line xxxx: xxxx
1337	invalid macro mode name "xxxx"
1338	incompatible redefinition of predefined macro "xxxx"
1339	redeclaration of entity-kind "entity" (declared at line xxxx) is missing a named-register storage class
1340	named register is too small for the type of the variable
1341	arrays cannot be declared with named-register storage class
1342	const_cast to enum type is nonstandard
1343	option "embedded_c" can be used only when compiling C
1344	a named address space qualifier is not allowed here
1345	an empty initializer is invalid for an array with unspecified bound
1346	function returns incomplete class type "type"
1347	entity-kind "entity" has already been initialized; the out-of-class initializer will be ignored
1348	declaration hides entity-kind "entity" (declared at line xxxx)
1349	a parameter cannot be allocated in a named address space
1350	invalid suffix on fixed-point or floating-point constant
1351	a register variable cannot be allocated in a named address space
1352	expected "SAT" or "DEFAULT"
1353	entity-kind "entity" has no corresponding member operator delete (to be called if an exception is thrown during initialization of an allocated object)
1354	a thread-local variable cannot be declared with "dllimport" or "dllexport"

1355	a function return type cannot be qualified with a named address space
1356	an initializer cannot be specified for a flexible array member whose elements have a nontrivial destructor
1357	an initializer cannot be specified for an indirect flexible array member
1358	invalid GNU version number:
1359	variable attributes appearing after a parenthesized initializer are ignored
1360	the result of this cast cannot be used as an lvalue
1361	negation of an unsigned fixed-point value
1362	this operator is not allowed at this point; use parentheses
1363	flexible array member initializer must be constant
1364	register names can only be used for register variables
1365	named-register variables cannot have void type
1366	__declspec modifiers not valid for this declaration
1367	parameters cannot have link scope specifiers
1368	multiple link scope specifiers
1369	link scope specifiers can only appear on functions and variables with external linkage
1370	a redeclaration cannot weaken a link scope
1371	link scope specifier not allowed on this declaration
1372	nonstandard qualified name in global scope declaration
1373	implicit conversion of a 64-bit integral type to a smaller integral type (potential portability problem)
1374	explicit conversion of a 64-bit integral type to a smaller integral type (potential portability problem)
1375	conversion from pointer to same-sized integral type (potential portability problem)
1376	alias creates cycle of aliased entities
1377	field name resolves to more than one offset -- see "entity" (declared at line xxxx) and "entity" (declared at line xxxx)
1378	"xxxx" is not a field name
1379	invalid symbolic operand name "xxxx"

1380	a symbolic match constraint must refer to one of the first ten operands
1381	friend specifier is not allowed in a class definition; friend specifier is ignored
1382	only static and extern variables can use thread-local storage
1383	multiple thread-local storage specifiers
1384	virtual entity-kind "entity" was not defined (and cannot be defined elsewhere because it is a member of an unnamed namespace)
1385	carriage return character in source line outside of comment or character/string literal
1386	expression must have fixed-point type
1387	invalid use of access specifier is ignored
1388	pointer converted to bool
1389	pointer-to-member converted to bool
1390	storage specifier ignored
1391	dllexport and dllimport are ignored on class templates
1392	base class dllexport/dllimport specification differs from that of the derived class
1393	redeclaration cannot add dllexport/dllimport to "entity" (declared at line xxxx)
1394	dllexport/dllimport conflict with "entity" (declared at line xxxx); dllexport assumed
1395	cannot define dllimport entity
1396	dllexport/dllimport requires external linkage
1397	a member of a class declared with dllexport/dllimport cannot itself be declared with such a specifier
1398	field of class type without a DLL interface used in a class with a DLL interface
1399	parenthesized member declaration is nonstandard

Errors 1400 to 1499

1400	white space between backslash and newline ignored for GNU compatibility -- may cause unintentional line continuation In most C dialects, the compiler does not parse backslash (\) as a line continuation character if it is followed by a whitespace. In the GNU C dialect, the compiler parses backslash as a line continuation character whether or not it is followed by a whitespace. In versions of the MULTI compiler earlier than 5.2, the GNU C and C++ dialects behaved the same way as other dialects. This diagnostic indicates that your code has a line that contains a backslash followed by a whitespace, and therefore may contain an unintentional line continuation. The following example shows one of these cases (the space is indicated by an underscore, because it does not display in print): <pre>#include <stdio.h> int main(void) { int i = 42; // use space after backslash, as in C:_ i = 0; // ignored in GNU if (i != 0) { printf("i = %d, should be 0\n", i); } }</pre>
1401	dllexport/dllimport conflict with "entity" (declared at line xxxx); dllimport/dllexport dropped
1402	invalid member for anonymous member class -- class "type" has a disallowed member function
1403	nonstandard reinterpret_cast
1404	positional format specifier cannot be zero
1405	a local class cannot reference a variable-length array type from an enclosing function
1406	member entity-kind "entity" (declared at line xxxx) already has an explicit dllexport/dllimport specifier
1407	a variable-length array is not allowed in a function return type
1408	variable-length array type is not allowed in pointer to member of type "type"
1409	the result of a statement expression cannot have a type involving a variable-length array
1410	support for trigraphs is disabled
1411	the "xxxx" attribute can only appear on functions and variables with external linkage

1412	strict mode is incompatible with treating namespace <code>std</code> as an alias for the global namespace
1413	use of <code>__if_exists</code> is not supported in this context
1414	<code>__if_exists</code> block not closed in the same scope in which it was opened
1415	thread-local variable cannot be dynamically initialized
1416	conversion drops " <code>__unaligned</code> " qualifier
1417	some enumerator values cannot be represented by the integral type underlying the enum type
1418	default argument is not allowed on a friend class template declaration
1419	value of multicharacter character literal is implementation-defined
1420	expected a class, struct, or union type
1421	second operand of <code>offsetof</code> must be a field
1422	second operand of <code>offsetof</code> may not be a bit field
1423	cannot apply <code>offsetof</code> to a member of a virtual base
1424	<code>offsetof</code> applied to non-POD types is nonstandard
1425	default arguments are not allowed on a friend declaration of a member function
1426	default arguments are not allowed on friend declarations that are not definitions
1427	redeclaration of entity-kind "entity" (declared at line xxxx) previously declared as a friend with default arguments is not allowed
1428	invalid qualifier for "type" (a derived class is not allowed here)
1429	invalid qualifier for definition of class "type"
1430	no prior <code>push_macro</code> for "xxxx"
1431	wide string literal not allowed
1433	"xxxx" is only allowed in C
1434	<code>__ptr32</code> and <code>__ptr64</code> must follow a "*"
1435	<code>__ptr32</code> and <code>__ptr64</code> cannot both apply
1436	template argument list of "xxxx" must match the parameter list

1437	an incomplete class type is not allowed
1438	complex integral types are not supported
1439	<code>__real</code> and <code>__imag</code> can only be applied to complex values
1440	<code>__real</code> / <code>__imag</code> applied to real value
1441	entity-kind "entity" (declared at line xxxx) was declared "deprecated ("xxxx")"
1442	invalid redefinition of entity-kind "entity" (declared at line xxxx)
1443	<code>dllimport</code> / <code>dllexport</code> applied to a member of an unnamed namespace
1444	<code>__thiscall</code> can only appear on nonstatic member function declarations
1445	<code>__thiscall</code> not allowed on function with ellipsis parameter
1446	explicit specialization of entity-kind "entity" must precede its first use
1447	a sealed class type cannot be used as a base class
1448	duplicate class modifier
1449	a member function cannot have both the "abstract" and "sealed" modifiers
1450	a sealed member cannot be pure virtual
1451	nonvirtual function cannot be declared with "abstract" or "sealed" modifier
1452	member function declared with "override" modifier does not override a base class member
1453	cannot override sealed entity-kind "entity" (declared at line xxxx)
1454	entity-kind "entity" (declared at line xxxx) was declared with the class modifier "abstract"
1455	option "--uliterals" can be used only when compiling C
1456	attribute "xxxx" not allowed on parameter declarations
1457	underlying type of enum type must be an integral type other than bool
1458	some enumerator constants cannot be represented by "type"
1459	"xxxx" not allowed in current mode

1460	option "--type_traits_helpers" can be used only when compiling C++
1461	attribute "sentinel" requires an ellipsis parameter
1462	argument must be a constant null pointer value
1463	insufficient number of arguments for sentinel value
1464	sentinel argument must correspond to an ellipsis parameter
1465	__declspec(implementation_key(...)) can appear only between #pragma start_map_region and #pragma stop_map_region
1466	#pragma start_map_region already active: pragma ignored
1467	no #pragma start_map_region is currently active: pragma ignored
1468	entity-kind "entity" cannot be used to name a destructor (a type name is required)
1469	nonstandard empty wide character literal treated as L'\0'
1470	"typename" may not be specified here
1471	a non-placement operator delete must be visible in a class with a virtual destructor
1472	name linkage conflicts with previous declaration of entity-kind "entity" (declared at line xxxx)
1473	subscript must be constant
1474	a variable with static storage duration allocated in a specific register cannot be declared with an initializer
1475	a variable allocated in a specific register must have POD type
1476	predefined meaning of "entity" discarded
1477	the "xxxx" attribute can only appear on functions and variables with internal linkage
1478	designator may not specify a non-POD subobject
1479	enum qualified name is nonstandard
1480	anonymous union qualifier is nonstandard
1481	anonymous union qualifier is ignored
1482	__declspec ignored (no variable or function declared)
1483	__declspec(xxxx) ignored (it has no meaning for a C struct)
1484	specifiers after comma between declarations are nonstandard
1485	nonstandard specifier ignored

1486	attributes are ignored on an enum declaration that is not also a definition
1487	declaring a reference with "mutable" is nonstandard
1488	a condition declaration for an array is always true
1489	static assertion failed with "xxxx"
1490	visibility attribute ignored because it conflicts with a previous declaration
1491	a member function cannot have internal linkage
1492	declaration hides built-in entity-kind "entity"
1493	declaration overloads built-in entity-kind "entity"
1494	the option to list macro definitions may not be specified when compiling more than one translation unit
1495	unexpected parenthesis after declaration of entity-kind "entity" (malformed parameter list or invalid initializer?)
1496	parentheses around a string initializer are nonstandard
1497	a variable declared with an auto type specifier cannot appear in its own initializer
1498	cannot deduce "auto" type
1499	initialization with "{...}" is not allowed for "auto" type

Errors 1500 to 1599

1501	invalid align value
1502	effect of this "#pragma align" directive is local to entity-kind "entity"
1503	pragmas which specify data areas cannot be nested
1504	"#pragma asm/endasm" cannot be placed xxxx
1505	interrupt specifier allowed on function declarations only
1506	interrupt function must return void
1507	interrupt function may not have parameters
1508	interrupt functions not supported in this compiler
1509	__interrupt specification is incompatible with previous "entity" (declared at line xxxx)

1510	This pragma is not implemented for this configuration The compiler displays this error if you use a <code>#pragma</code> directive that is not implemented for your processor or target operating system, or if you set an option that disables the directive. For more information, see the documentation for the <code>#pragma</code> directive reported in the error.
1511	invalid alignvar value
1512	incorrect section <code>#pragma: xxxx</code>
1513	GHS <code>#pragma</code> with incorrect keyword "xxxx"
1514	incorrect GHS <code>#pragma: xxxx</code>
1515	the end data section pragma does not match the start data section pragma
1517	this preprocessing directive is not permitted within a <code>#pragma asm/endasm</code> block
1518	a questionable cast from type "type" to type "type" causes ANSI aliasing rules to be turned off for this function This warning indicates that the compiler has detected that two entities of incompatible type might alias each other at function scope, and therefore the compiler will not perform any optimizations whose correctness depends on aliasing information when compiling the current function. This warning is issued only when those optimizations have been enabled. When compiling without optimizations, remark 1798 is issued instead. To disable ANSI aliasing rules and suppress this warning, pass the -no_ansi_alias option.
1519	a questionable cast from type "type" to type "type" -- please recompile with ANSI aliasing turned off This error indicates that the compiler has detected that two entities of incompatible type might alias each other at file scope, and therefore the compiler cannot perform any optimizations whose correctness depends on aliasing information. This error is issued only when those optimizations have been enabled. When compiling without optimizations, remark 1798 is issued instead. To disable ANSI aliasing rules and suppress this error, pass the -no_ansi_alias option.
1520	unsupported GNU language extension: xxxx
1521	use <code>-gnu</code> for full GNU support
1522	global register variable cannot have an initializer
1523	This macro definition will be commented-out in <code>asmsym</code> output files since it may have adverse effects

1524	Incorrect usage of <code>__ASMSYM_INTERNAL()</code>
1527	The name specified in the 'intvect' must be a declared interrupt routine
1528	An asmsym region ended before the xxxx definition containing field "xxxx" was complete
1529	asm statement incompatible with binary code generation. Try command line option -noobj or builder menu File Options->Advanced->Compilation = Assembly
1530	this builtin function may not have its address taken
1531	Builtin function xxxx is not xxxx
1532	overloaded builtin function may not be declared
1533	defining builtin function; may be called only via pointer to function
1534	expression must be a vector type or have a constant value
1535	builtin function may not be implicitly declared
1536	use of obsolete feature: xxxx
1537	too few initializer values
1538	type must be compatible with builtin function
1539	vector argument requires a prototype
1540	vector type specifier is not first
1541	operand of vec_step must be a vector type
1542	floating point types have been disabled
1543	floating point constants have been disabled
1544	incompatibility associated with redeclaration of entity-kind "entity": xxxx
1545	address taken of a packed structure member with insufficient alignment
1546	asm constructs are non-portable
1547	call is made to a non-prototyped function GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1550	"auto" type cannot appear in top-level array type
1551	"auto" type cannot appear in top-level function type

1552	a member of type "type" cannot have an in-class initializer
1553	a member with an in-class initializer must be const
1554	cannot deduce "auto" type (initializer required)
1555	"auto" type is "type" for this entity, but was previously implied to be "type"
1556	invalid constructor declaration
1557	invalid use of a type qualifier
1558	a union cannot be abstract or sealed
1559	"auto" is not allowed here
1560	definition of base class type not completed yet
1561	"extern template" cannot refer to a specialization of static entity-kind "entity" (declared at line xxxx)
1562	"extern template" cannot follow explicit instantiation of entity-kind "entity"
1563	<code>__declspec(restrict)</code> requires a function returning a pointer type
1564	the "report_gnu_extensions" option is only valid in GNU C and GNU C++ modes
1565	variable-length array types are nonstandard
1566	designators are nonstandard
1567	this designator syntax is a GNU extension
1568	compound literals are nonstandard
1569	statement expressions are a GNU extension
1570	asm name ignored for previously defined entity
1571	attributes are a GNU extension
1572	extended asm syntax is a GNU feature
1573	volatile asm declarations are a GNU extension
1574	asm name specifiers are a GNU extension
1575	the " <code>__restrict</code> " qualifier is nonstandard
1576	"typeof" is a GNU extension
1577	modifying the size or signedness of a typedef is nonstandard
1578	zero-length arrays are a GNU extension

1579	flexible array members are nonstandard
1580	attribute "nonnull" references nonpointer parameter
1581	argument for attribute "nonnull" is larger than number of parameters
1582	no parameter has pointer type
1583	null argument provided for parameter marked with attribute "nonnull" This warning is issued in connection with the nonnull attribute.
1584	the destructor for "type" has been suppressed because the destructor for "type" is inaccessible
1585	the suppressed destructor for "type" is needed
1586	routine is both "inline" and "noinline" ("noinline" assumed)
1587	invalid cleanup routine
1588	attribute "cleanup" requires automatic storage duration
1589	attribute "cleanup" does not apply to parameters
1590	cleanup routine has invalid type
1591	call of cleanup routine requires suspect conversion
1592	__sptr and __uptr must follow a "*"
1593	__sptr and __uptr cannot both be specified
1594	widening pointer conversion from "type" to "type" extends sign bit
1595	__sptr and __uptr don't apply to pointer-to-member types
1596	the declaration of the copy assignment operator for "type" has been suppressed because entity-kind "entity" is const
1597	the declaration of the copy assignment operator for "type" has been suppressed because entity-kind "entity" has reference type
1598	the declaration of the copy assignment operator for "type" has been suppressed because that of "type" was suppressed
1599	the declaration of the copy assignment operator for "type" has been suppressed because that of "type" is ambiguous

Errors 1600 to 1699

1600	the declaration of the copy assignment operator for "type" has been suppressed because that of "type" is inaccessible
1601	the declaration of the copy constructor for "type" has been suppressed because that of "type" was suppressed
1602	the declaration of the copy constructor for "type" has been suppressed because that of "type" is ambiguous
1603	the declaration of the copy constructor for "type" has been suppressed because that of "type" is inaccessible
1604	the destructor for "type" will not be called because it is inaccessible and the destructor for "type" was suppressed
1605	definition at end of file not followed by a semicolon or a declarator
1606	first argument must be a pointer to integer or enumeration type
1607	synchronized operations are valid only on objects of size 1, 2, 4, or 8
1608	extra arguments ignored
1609	'=' assumed following macro name "xxxx" in command-line definition
1610	white space is required between the macro name "xxxx" and its replacement text
1611	result of call is not used
1612	attribute is ignored for void return type
1613	attribute requires pointer-to-function type
1614	dllimport/dllexport is ignored on redeclaration using a qualified name
1615	too many characters in character literal -- extra leading characters ignored
1616	entity-kind "entity" cannot be declared inline after its definition at line xxxx
1617	a statement expression is not allowed in a template-dependent typeof specifier
1618	a statement expression is not allowed in a decltype specifier
1619	duplicate function modifier
1620	invalid character for char16_t literal

1621	<code>__LPREFIX</code> cannot be applied to <code>char16_t</code> or <code>char32_t</code> literals
1622	unrecognized calling convention <code>xxxx</code> , must be one of:
1632	<code>__assume</code> expression with side effects discarded
1633	attributes ignored here
1634	type qualifiers are meaningless here
1657	unable to create precompiled header file: <code>xxxx</code>
1658	directory of source file not the same as when precompiled header file " <code>xxxx</code> " was created
1695	trigraphs not allowed
1696	register storage class not allowed
1697	variable argument list not allowed
1698	basic types may not be used -- use <code>typedef</code>
1699	<code>char</code> without signed or unsigned specifier not allowed

Errors 1700 to 1799

1700	" <code>xxxx</code> " is not allowed in C++ -- use " <code>inline</code> "
1701	" <code>xxxx</code> " is not allowed
1702	unary minus may not be applied to an unsigned expression GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1703	cast from type " <code>type</code> " to type " <code>type</code> " not allowed
1704	floating point variable may not be tested for exact equality or inequality GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1705	the <code>goto</code> statement is not allowed
1706	the <code>continue</code> statement is not allowed
1707	the <code>break</code> statement may only be used to terminate the case of a <code>switch</code> statement
1708	label not allowed
1709	dependent statement in conditional or loop construct must be enclosed in braces

1710	final else clause required for if, else if construct
1711	previous non-empty case clause must be terminated with a break statement GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1712	switch statement must contain a default case GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1713	switch statement must contain at least one case GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1714	call mode specifier allowed on function declarations only
1715	#undef may not be used
1716	only one # or ## operator per macro allowed
1717	bit-fields should be explicitly "signed" or "unsigned"
1718	octal constant (other than zero) not allowed
1719	comma operator only allowed in the control expression of a for loop
1720	wide string literal not allowed
1721	identifier shadows entity-kind "entity" (declared at line xxxx) GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1722	identifier redefines typedef entity-kind "entity" (declared at line xxxx)
1723	identifier conflicts with entity-kind "entity" (declared at line xxxx) in separate namespace
1724	right hand operand of && or operator may not contain side effects
1725	side effects not allowed in operand of sizeof operator GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1726	only the first or all members of an enumeration may be explicitly initialized GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.

1727	identifiers must be given for all of the declared parameters
1728	function with no parameters must be declared with parameter type void GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1729	operands of logical && or must be primary expressions
1730	bitwise operation may not be performed on signed integer types
1731	function must be declared at file scope GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1732	#define and #undef may not be used within a block GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1733	switch expression may not represent a boolean value GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1734	function must have a single exit point
1735	constant unsigned integer expression may not wrap around
1736	initializer must use braces to match structure of object GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1737	expression may not depend on operator precedence rules
1738	assignment operator not allowed in boolean operation
1739	implicit conversion may result in loss of information when converting from "type" to "type"
1740	redundant explicit cast not allowed
1741	no more than 2 levels of pointer indirection may be used
1742	all members of structure or union must be named
1743	definition and each parameter of function-like macro must be enclosed in parentheses
1744	the type "long long" is not supported for this configuration
1745	parameter identifiers in declaration different from prototype entity-kind "entity" (declared at line xxxx)

1746	null statement must occur on line by itself
1747	non-standard character in header file name not allowed GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1748	#include statements may only be preceded by preprocessor directives and comments
1749	in-line assembly must be embedded in dedicated assembly procedures only
1750	floating point expression not allowed in condition expression of for loop GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1751	mixed precision arithmetic must use explicit cast
1752	pointer arithmetic not allowed
1753	fixed-point value does not fit in required fixed-point type
1754	fixed-point value does not fit in required floating-point type
1755	fixed-point value does not fit in required integer type
1756	floating-point value does not fit in required fixed-point type
1757	integer value does not fit in required fixed-point type
1758	fixed-point type not supported by this target
1759	fixed-point value out of range
1760	__sat type specifier is not first
1761	__vector requires an underlying element type
1762	vector of "type" not allowed
1763	explicit dimension of vector of "type" is too large
1764	vector subscript out of range
1766	badly placed #pragma: xxxx
1767	result of call is not used (cast to void)
1768	result of call is not used
1770	incorrect #pragma: xxxx
1771	possible occurrence of 'commented out' code
1772	identifier too long - truncated to xxxx significant characters
1773	"__noinline" is not allowed

1774	function address expression - function call without arguments needs "()" as the parameter list
1775	numeric constant should be suffixed to indicate type
1776	macro definition can only be used for symbolic constants, function-like macros, or type and storage class qualifiers
1777	detected usage of a non-constant pointer to function
1778	function pointer prototypes must be identical
1779	address of a position independent object cannot be an initial value; detected inside the routine at the given position
1780	address of an automatic variable potentially used out of scope GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1781	--config option error:
1782	a questionable cast has been found in a routine at the given position - the ANSI aliasing rules are turned off for that routine
1786	argument must be an integer literal in range [0,64)
1787	argument must be an integer literal in range [-32,32)
1788	call to intrinsic function "xxxx" without a prior declaration
1789	bit-field base type must not be larger than 32 bits
1790	failed to vectorize the loop: xxxx
1791	call is made to a non-prototyped function, known to use variable arguments: xxxx GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1792	fixed-point result out of range
1793	operator not allowed on type "type"
1794	extension cannot be used with this target configuration
1795	C++ symbols with the sequence `__' in their names might not be debuggable
1796	use of section "xxxx" not compatible with its other uses -- explicit section assignment ignored
1797	external identifier shall have exactly one definition GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.

1798	a questionable cast from type "type" to type "type" may indicate illegal references See diagnostics 1518 and 1519.
1799	.function declared without a prototype - consider `void' for the parameter list

Errors 1800 to 1899

1800	function defined without a prototype
1801	unexpected character in weak identifier sequence
1803	builtin va_list operators not supported
1804	implicit conversion from type "type" to type "type"
1805	cannot have `#pragma pack' and `#pragma pad' active at the same time
1806	`#pragma align' lowers the alignment below alignment specified by type
1807	taking the address of a register is not allowed
1808	calling convention specification not allowed here
1809	conflicting calling convention attribute
1810	packed floating point types not allowed
1811	byte ordering qualifiers only apply to pointer and integral types larger than char, not "type"
1812	implicit const-cast of string literal
1813	argument must be an integer literal in range xxxx
1814	argument will be truncated to range xxxx
1815	Dwarf 1 cannot represent functions that are defined in distinct .text sections in the same compile unit
1816	external declaration should be in header file
1817	linkonce specifier allowed on function declarations only
1818	floating point division by zero
1819	bit-fields within union probably used to access sub-parts of larger data type

1820	floating point types within union depend on underlying bit representation
1821	unreachable code at top level of switch statement not allowed
1822	value of multicharacter literal is implementation-defined
1823	multibyte character detected
1824	incomplete array type with external linkage not allowed for entity-kind "entity" (declared at line xxxx)
1825	comma operator not allowed
1826	compound statement required for if construct
1827	union not allowed
1828	static function defined without a prototype
1829	parameter of function-like macro must be enclosed in parentheses
1830	# or ## operator not allowed
1831	token "defined" may not be generated during #if or #elif expansion
1832	conversion from type "type" to type "type" not allowed
1833	conversion from type "type" to type "type" not allowed
1834	cast from type "type" to type "type" not allowed
1835	cast from type "type" to type "type" not allowed
1836	cast from type "type" to type "type" that removes underlying const or volatile qualifier not allowed
1837	default clause must be final clause of switch statement GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1838	case or default label not scoped in top level compound statement of switch statement GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1839	entity-kind "entity" (declared at line xxxx) mixes in-line assembly language and C/C++
1840	identifier reused from declaration of entity-kind "entity" (declared at line xxxx)

1841	typedef name conflict between entity-kind "entity" (declared at line xxxx) and identifier
1842	tag name conflict between entity-kind "entity" (declared at line xxxx) and identifier
1843	static storage name conflict between entity-kind "entity" (declared at line xxxx) and identifier
1844	parameter or return type in declaration different from prototype entity-kind "entity" (declared at line xxxx) GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1845	multiple break statements per iteration statement not allowed
1846	object or function defined in header file
1847	cast of complex integer expression with underlying type "type" to non-integer type "type" not allowed
1848	cast of complex floating point expression from type "type" to type "type" not allowed
1849	bitwise ~ or << expression must be immediately cast back to underlying type, "type"
1850	plain char type can only be used for storage of character values
1851	signed or unsigned char type can only be used for storage of numeric values
1852	only assignment and equality operators may be used on plain char operand
1853	effectively boolean expression required
1854	effectively boolean expression not allowed
1855	explicit test against zero required
1856	result of increment or decrement expression may not be used
1857	result of boolean operation is invariant
1858	subscript operator may only be applied to objects declared as an array type
1859	macro definition can only be used for a braced initializer, constant, parenthesized expression, type qualifier, storage class specifier, or do-while-zero construct
1860	macro definition must balance all braces, parentheses, and brackets

1861	reserved identifier, macro, or function name "xxxx" cannot be defined, redefined, or undefined
1862	function-like macro not allowed -- use function instead
1863	implicit conversion from underlying type, "type", to non-integer underlying type, "type", not allowed
1864	implicit conversion from underlying type, "type", to narrower underlying type, "type", not allowed
1865	implicit conversion from underlying type, "type", to same width underlying type, "type", not allowed
1866	implicit conversion from underlying type, "type", to underlying type, "type", changes signedness
1867	implicit conversion of complex expression from underlying type, "type", to underlying type, "type", not allowed
1868	implicit conversion of non-constant function argument from underlying type, "type", to underlying type, "type", not allowed
1869	implicit conversion of non-constant return expression from underlying type, "type", to underlying type, "type", not allowed
1870	implicit conversion of boolean expression from underlying type, "type", to underlying type, "type", not allowed
1871	implicit conversion from floating type, "type", to non-floating type, "type", not allowed
1872	implicit conversion from floating type, "type", to narrower type, "type", not allowed
1873	implicit conversion of complex floating point expression from type, "type", to type, "type", not allowed
1874	implicit conversion of function argument from floating type, "type", to type, "type", not allowed
1875	implicit conversion of return expression from floating type, "type", to type, "type", not allowed
1876	// style comment not allowed
1877	cast of 0 from type "type" to type "type" not allowed
1878	cast of complex integer expression with underlying type "type" to wider type "type" not allowed
1879	cast of complex integer expression with underlying type "type" to same width type "type" not allowed

1880	cast of complex integer expression with underlying type "type" to type "type" changes signedness
1881	explicit test of constant against zero required
1882	header file needs guard to prevent multiple inclusion
1883	call of virtual function inside constructor or destructor when extract vptrs is used
1884	previous non-empty case clause doesn't fall through but must be terminated with a break statement
1885	reserved identifier, macro, or function name "xxxx" beginning with an underscore cannot be defined, redefined, or undefined
1886	command-line macro definition can only be used for a braced initializer, constant, parenthesized expression, type qualifier, storage class specifier, or do-while-zero construct
1887	TDA is disabled for this configuration
1888	#define or #undef of keyword "xxxx"
1889	invalid alignfunc value
1890	initialized non-const entity-kind "entity" was used but never modified
1891	invalid section renaming option:
1892	comparison operator has higher precedence than bitwise operator; parentheses recommended GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1893	entity-kind "entity" will be initialized before the previous constructor initializer, entity-kind "entity"
1894	"restrict" qualifier in declaration different from prototype entity-kind "entity" (declared at line xxxx)
1895	implicit conversion from bool to pointer type
1896	use of reserved trace register
1897	incorrect #pragma double_check xxxx
1898	if statement does not affect control flow
1899	use of gets is unsafe; use fgets instead GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.

Errors 1900 to 1999

1900	use of %s in scanf is unsafe; specify a maximum field width GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1901	assignment of constant address to bool
1902	defined must be followed by an identifier
1903	"xxxx" cannot be used as a macro
1904	sizeof pointer parameter yields size of pointer-to-type where size of array-of-type may have been intended
1905	shift operator has higher precedence than bitwise operator; parentheses suggested
1906	call to setjmp() and longjmp() is not allowed GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1907	call to atof(), atoi(), and atol() is not allowed -- use strtod() or strtol() instead GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1908	initialization of array of type "type" discards terminating null character GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1909	comma operator only allowed in the first or third expression of a for loop
1910	part of expression has no effect GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1911	this expression cannot be used as the initializer for a bit field
1912	global register variable must have integral or pointer type GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1913	variable's type is too wide for a single machine register GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1914	declared size for boolean bit-field is greater than 1

1915	cast involving pointer to incomplete class type "type" will not adjust for inheritance GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1916	<code>__alignof__</code> operator cannot be applied to a bit-field
1917	subtraction of pointers to zero-sized type "type"
1918	out of resources while compiling function "xxxx" (internal variable limit exceeded) The compiler limits the number of variables you can have per function to 65536, which is what causes this error. This limit includes both explicitly declared variables and internal variables the compiler produces. There is no way to increase this limit. One common cause of this error is run-time error checking, which instruments the C code and creates many internal variables. You might want to work around the limitation by selectively disabling run-time error checks.
1922	comparison with string literal has unspecified result
1923	unusual comparison of non-integer literal to pointer
1924	unusual assignment of non-integer literal to pointer
1925	type "type" must define a default constructor to initialize its members lacking default constructors
1926	entity-kind "entity" does not initialize the following fields:
1927	initialized argument must be a pointer or reference to non-const GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1928	argument index is greater than number of parameters
1929	bit-field of same width as underlying type should be explicitly "signed" or "unsigned"
1930	member entity-kind "entity" is marked "final" and cannot be overridden
1931	operand of sizeof is not a type, variable, or dereferenced pointer expression GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1932	format string parameter is not a compile-time constant GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1933	"va_list" passed through ellipsis

1934	aggregate type passed through ellipsis
1935	one-argument constructor should be declared "explicit" This remark is issued whenever you declare a one-argument constructor without the <code>explicit</code> keyword. A non-explicit constructor participates in implicit type conversions. For example: <pre>struct Integer { int value; Integer(int v): value(v) {} }; void foo(const Integer &i); int main() { foo(0); } //Implicit conversion to Integer</pre> This program compiles because 0 may be implicitly converted to <code>Integer</code> via the non-explicit constructor. By adding the <code>explicit</code> keyword, you prevent such conversions; to pass 0 to <code>foo()</code>, you would have to write: <pre>struct Integer { int value; explicit Integer(int v): value(v) {} }; void foo(const Integer &i); int main() { foo(Integer(0)); }</pre>
1936	comparison of comparison where two comparison expressions may have been intended
1937	possibly unintentional comparison of <code>bool</code> to non-<code>bool</code> GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1938	byte ordering qualifiers do not apply to bit-fields
1939	byte ordering qualifiers do not apply to parameters
1940	byte-reversed objects cannot be statically initialized
1941	restrict qualifier is ignored
1942	attribute "xxxx" may only be applied to a typedef
1943	strong typedef rules forbid implicit conversion from type "type" to type "type"
1944	switch statement must contain at least two cases
1945	the "xxxx" attribute is ignored in this context
1946	type attributes should follow the class/struct/union/enum keyword, not the closing brace

1947	struct field is being placed at offset xxxx instead of requested offset This error is issued in connection with the <code>at</code> attribute.
1948	a member function with a non-default calling convention may not have its address taken
1949	placement <code>new[]</code> may invoke undefined behavior and should not be used
1950	multiple definitions of symbol "xxxx"
1951	virtual entity-kind "entity" should be declared "virtual"
1952	+ or - operator has higher precedence than shift operator; parentheses recommended
1953	adjacent string literals will be concatenated
1954	third argument to <code>strncmp</code> does not match the length of the string literal argument
1955	third argument to <code>memset</code> is zero; second and third arguments may have been swapped accidentally
1956	<code>sizeof</code> argument xxxx to xxxx does not divide value of argument xxxx
1957	<code>strlen</code> used in memory allocation; <code>strlen+1</code> was probably intended
1958	implausible operator at top level of boolean expression
1959	uninitialized local variable
1960	apparently redundant comparisons may indicate a mistake
1961	pointer comparison to <code>NULL</code> depends on implementation-defined pointer signedness
1962	case range syntax is non-standard
1963	label not followed by statement is non-standard GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.
1964	cannot open coding standard file
1965	error in coding standard file:
1966	unused coding standard tag or malformed error number
1967	trigraph in source code has possibly unintended effect on semantics GHS Standard Mode enables this diagnostic. For more information about GHS Standard Mode, see the MULTI: Building Applications book.

1968	uninitialized local variable of struct or union type
1969	statement begins in column different from first statement in its block
1970	call tabled functions not supported in this compiler
1971	call tabled interrupt functions are not supported
1972	base specifier list contains more than one "non-interface" class Java and some other object-oriented languages support multiple inheritance of "interfaces", but only single inheritance of "implementation". In C++ terms, an "interface" is a class that contains only pure virtual member functions and static member data (in other words, it does not contain code or non-static data members). This diagnostic is given whenever a class specifies two or more "non-interface" classes in its list of base classes.
1973	conversion drops "__packed" qualifier
1974	overloading the &&, , or comma operators is dangerous
1975	dynamic initialization causes code to be executed before main()
1976	"long" or "long long" constant should be suffixed to indicate type
1977	strong fptr rules forbid explicit conversion from type "type" to type "type"
1978	strong fptr rules forbid implicit conversion from type "type" to type "type"
1979	strong fptr rules prohibit type "type" from being a member of a union
1980	not enough initializers; trailing elements will be default-initialized
1981	empty initializer is non-standard
1982	target stack alignment is insufficient to guarantee alignment of entity-kind "entity"
1983	character escape sequence "\e" is non-standard
1984	octal character escape sequences are not allowed
1985	hexadecimal character escape sequences are not allowed
1986	shift count is too large for underlying type "type"
1987	argument of type "type" is incompatible with ptr_ref parameter
1988	format argument has incorrect type "type"

1989	arguments xxxx and xxxx to xxxx have pointer types whose targets differ in size
1990	argument xxxx to xxxx must be of the form n*sizeof(a)
1991	argument xxxx to xxxx has a type inconsistent with the sizeof-expression in argument xxxx
1996	a __linkonce function cannot be inline
1997	a __linkonce function cannot be static
1998	
1999	function "xxxx" contains possible recursion

Errors 2000 to 2099

2000	attribute "xxxx" is not implemented and will be ignored
2001	invocation of function-like macro not allowed
2002	character greater than 127 not allowed
2003	alignment of type pointed to by "type" is smaller than the one pointed to by "type", which may lead to a misaligned memory access
2004	operation promotes signed type "type" to unsigned type "type", which may lead to unintended results
2005	line splicing inhibited by whitespace after backslash
2006	__[no]sideeffect must be preceded by volatile, otherwise it is ignored
2007	'__ghs_eagle_tag' may only be used as a simple assignment inside of 'if' statement conditional; 'if' statement may not have 'else'
2008	potentially unintended initializer with no named variable
2009	mixing __sideeffect and __nosideeffect qualifiers not allowed
2010	'strong_fptr' attribute invalid for non-function pointer typedef
2011	'strong_fptr' attribute invalid for unnamed struct/class
2012	Use of static assertions is nonstandard
2013	identifier "xxxx" reserved for future library use cannot be defined, redefined, or undefined
2014	an expression is not allowed in a template-dependent typeof specifier

2015	type "type" has non-strong function pointers -- it cannot be a member or base class of strong_fptr type "type"
2016	non-static local variable may not be qualified by, or contain field qualified by, '__sideeffect'