

Error Codes

This page provides an overview of each error code. You can suppress errors using the Suppressions Control Panel (see [Customizing Options on Windows](#)). The following sections provide a detailed description of each error including:

- A brief explanation of what problem has been detected.
- An example program that generates a similar error.
- Output that would be generated by running the example, with annotations indicating what the various pieces of the diagnostic mean and how they should be interpreted in identifying your own problems. *The exact appearance of the error message may depend on how Insure++ is currently configured.*
- A brief description of ways in which the problem might be eliminated.

Sometimes you will see values identified as `<argument #>` or `<return>` instead of names from your program. In this case, `<argument n>` refers to the `n`th argument passed to the current function (i.e., the argument where the error was detected), and `<return>` refers to a value returned from the function indicated.

ALLOC_CONFLICT

Code	Description	Enabled	Reported	Platform
ALLOC_CONFLICT	Mixing malloc/free with new/ delete	✓	Runtime	Windows/Unix
(badfree)	Memory was allocated with <code>new</code> or <code>new[]</code> and an attempt was made to free it with <code>free</code> .	✓	Runtime	Windows/Unix
(baddelete)	Memory was allocated with <code>malloc</code> and an attempt was made to free it with <code>delete</code> or <code>delete[]</code> .	✓	Runtime	Windows/Unix
(badhandle)	Memory is allocated by one heap and freed by a different heap	✓	Runtime	Windows

BAD_CAST

Code	Description	Enabled	Reported	Platform
BAD_CAST	Cast of pointer loses precision	✓	Compilation	Windows/Unix








BAD_DECL

Code	Description	Enabled	Reported	Platform
BAD_DECL	Incompatible global declarations	✓	Runtime	Windows/Unix


BAD_FORMAT

Code	Description	Enabled	Reported	Platform
BAD_DECL	Incompatible global declarations	✗	Compilation Runtime	Windows/Unix
(sign)	Types differ only by sign, e.g., <code>int</code> vs. <code>unsigned int</code>	✗	Compilation	Windows/Unix
(compatible)	Fundamental types are different but they happen to have the same representation on the particular hardware in use, e.g., <code>int</code> vs. <code>long</code> on machines where both are 32-bits, or <code>int *</code> vs. <code>long</code> where both are 32-bits.	✗	Compilation	Windows/Unix
(incompatible)	Fundamental types are different, e.g., <code>int</code> vs. <code>double</code>	✓	Runtime	Windows/Unix
(other)	A problem other than an argument type mismatch is detected, such as passing the wrong number of arguments. Error messages are classified according to this scheme and can be selectively enabled or disabled	✓	Compilation	Windows/Unix


BAD_PARM

Code	Description	Enabled	Reported	Platform
BAD_PARM	Mismatch in argument type.		Compilation	Windows/Unix
(sign)	Types differ only by sign, e.g., <code>int</code> vs. <code>unsigned int</code>		Compilation	Windows/Unix
(compatible)	Fundamental types are different but they happen to have the same representation on the particular hardware in use, e.g., <code>int</code> vs. <code>long</code> on machines where both are 32-bits.		Compilation	Windows/Unix
(incompatible)	Fundamental types are different, e.g., <code>int</code> vs. <code>double</code>		Compilation	Windows/Unix
(pointer)	This is not an error class, but a keyword used to suppress messages about mismatched pointer types, such as <code>int *</code> vs. <code>char *</code> .		Compilation	Windows/Unix
(union)	Forces a declared union argument to match only a similar union as an actual argument. If this is suppressed, you may pass any of the individual union elements to the routine, rather than the union type, or pass a union to a routine which expects one of the union-elements as an argument.		Compilation	Windows/Unix
(other)	A problem other than an argument type mismatch is detected, such as passing the wrong number of arguments. Error messages are classified according to this scheme and can be selectively enabled or disabled		Compilation	Windows/Unix


BOGUS_LEAK

Code	Description	Enabled	Reported	Platform
BOGUS_LEAK	Runtime leak not found during execution		Runtime	Windows/Unix


COPY_BAD_RANGE

Code	Description	Enabled	Reported	Platform
COPY_BAD_RANGE	Attempt to copy out-of-range pointer		Runtime	Windows/Unix

COPY_DANGLING

Code	Description	Enabled	Reported	Platform
COPY_DANGLING	Attempt to copy dangling pointer		Runtime	Windows/Unix

COPY_UNINIT_PTR

Code	Description	Enabled	Reported	Platform
COPY_UNINIT_PTR	Attempt to copy uninitialized pointer		Runtime	Windows/Unix

COPY_WILD

Code	Description	Enabled	Reported	Platform
COPY_UNINIT_PTR	Attempt to copy a wild pointer		Runtime	Windows/Unix

DEAD_CODE

Code	Description	Enabled	Reported	Platform
DEAD_CODE	Code is not evaluated, has no effect, or is unreachable	✘	Compilation	Windows/Unix
(emptyloopbody)	Loop body is empty.	✘	Compilation	Windows/Unix
(emptystmt)	The statement is empty.	✘	Compilation	Windows/Unix
(noeffect)	Code has no effect.	✘	Compilation	Windows/Unix
(notevaluated)	Code is not evaluated.	✘	Compilation	Windows/Unix

DELETE_MISMATCH

Code	Description	Enabled	Reported	Platform
DELETE_MISMATCH	Mismatch between new/new[] and delete/delete[]	✘	Runtime	Windows/Unix
(bracket)	new, delete[]	✔	Runtime	Windows/Unix
(nobracket)	new[], delete	✔	Runtime	Windows/Unix

EXPR_BAD_RANGE

Code	Description	Enabled	Reported	Platform
EXPR_BAD_RANGE	Expression exceeded range	✘	Runtime	Windows/Unix

EXPR_DANGLING

Code	Description	Enabled	Reported	Platform
EXPR_DANGLING	Expression uses a dangling pointer	✘	Runtime	Windows/Unix

EXPR_NULL

Code	Description	Enabled	Reported	Platform
EXPR_NULL	Expression uses NULL pointer	✔	Runtime	Windows/Unix

EXPR_UNINIT_PTR

Code	Description	Enabled	Reported	Platform
EXPR_UNINIT_PTR	Expression uses uninitialized pointer	✔	Runtime	Windows/Unix

EXPR_UNRELATED_PTRCMP

Code	Description	Enabled	Reported	Platform
EXPR_UNRELATED_PTRCMP	Expression compares unrelated pointers	✔	Runtime	Windows/Unix

EXPR_UNRELATED_PTRDIFF

Code	Description	Enabled	Reported	Platform
EXPR_UNRELATED_PTRDIFF	Expression subtracts unrelated pointers	✓	Runtime	Windows/Unix

EXPR_WILD

Code	Description	Enabled	Reported	Platform
EXPR_WILD	Expression uses a wild pointer	✗	Runtime	Windows/Unix

FREE_BODY

Code	Description	Enabled	Reported	Platform
FREE_BODY	Freeing memory block from body	✓	Runtime	Windows/Unix

FREE_DANGLING

Code	Description	Enabled	Reported	Platform
FREE_DANGLING	Freeing dangling pointer	✓	Runtime	Windows/Unix

FREE_GLOBAL

Code	Description	Enabled	Reported	Platform
FREE_GLOBAL	Freeing global memory	✓	Runtime	Windows/Unix

FREE_LOCAL

Code	Description	Enabled	Reported	Platform
FREE_LOCAL	Freeing local memory	✓	Runtime	Windows/Unix

FREE_NULL

Code	Description	Enabled	Reported	Platform
FREE_NULL	Freeing null pointer	✓	Runtime	Windows/Unix

FREE_UNINIT_PTR

Code	Description	Enabled	Reported	Platform
FREE_UNINIT_PTR	Freeing uninitialized pointer		Runtime	Windows/Unix

FREE_WILD

Code	Description	Enabled	Reported	Platform
FREE_WILD	Freeing wild pointer		Runtime	Windows/Unix

FUNC_BAD

Code	Description	Enabled	Reported	Platform
FUNC_BAD	Function pointer is not a function		Runtime	Windows/Unix

FUNC_NULL

Code	Description	Enabled	Reported	Platform
FUNC_NULL	Function pointer is NULL		Runtime	Windows/Unix

FUNC_UNINIT_PTR

Code	Description	Enabled	Reported	Platform
FUNC_UNINIT_PTR	Function pointer is uninitialized		Runtime	Windows/Unix

HEAP_CORRUPT

Code	Description	Enabled	Reported	Platform
HEAP_CORRUPT	Dynamic memory heap is corrupt		Runtime	Windows/Unix

INSURE_ERROR

Code	Description	Enabled	Reported	Platform
INSURE_ERROR	Various internal errors		Runtime, Compilation	Windows/Unix

INSURE_WARNING

Code	Description	Enabled	Reported	Platform
INSURE_WARNING	Errors from <code>iic_warning</code> calls		Compilation	Windows/Unix

LEAK_ASSIGN

Code	Description	Enabled	Reported	Platform
LEAK_ASSIGN	Memory leaked as a result of pointer reassignment	✓	Runtime	Windows/Unix

LEAK_FREE

Code	Description	Enabled	Reported	Platform
LEAK_ASSIGN	Memory leaked freeing block	✓	Runtime	Windows/Unix

LEAK_RETURN

Code	Description	Enabled	Reported	Platform
LEAK_RETURN	Memory leaked by ignoring returned value	✓	Runtime	Windows/Unix

LEAK_SCOPE

Code	Description	Enabled	Reported	Platform
LEAK_RETURN	Memory leaked leaving scope	✓	Runtime	Windows/Unix

PARM_BAD_RANGE

Code	Description	Enabled	Reported	Platform
PARM_BAD_RANGE	Array parameter exceeded range	✓	Runtime	Windows/Unix

PARM_DANGLING

Code	Description	Enabled	Reported	Platform
PARM_DANGLING	Array parameter is a dangling pointer	✓	Runtime	Windows/Unix

PARM_NULL

Code	Description	Enabled	Reported	Platform
PARM_NULL	Array parameter is NULL	✓	Runtime	Windows/Unix

PARM_UNINIT_PTR

Code	Description	Enabled	Reported	Platform
PARM_UNINIT_PTR	Array parameter is an uninitialized pointer	✓	Runtime	Windows/Unix

PARM_WILD

Code	Description	Enabled	Reported	Platform
PARM_WILD	Array parameter is wild	✓	Runtime	Windows/Unix

PARSE_ERROR

Code	Description	Enabled	Reported	Platform
PARSE_ERROR	Unidentified error	✓	Compilation	Windows/Unix

PARSE_WARNING

Code	Description	Enabled	Reported	Platform
PARSE_WARNING	Variable declared but never used	✗	Compilation	Windows/Unix

PCH_ERROR

Code	Description	Enabled	Reported	Platform
PCH_ERROR	Unsupported PCH configuration	✓	Compilation	Windows

READ_BAD_INDEX

Code	Description	Enabled	Reported	Platform
READ_BAD_INDEX	Reading array out of range	✓	Runtime	Windows/Unix

READ_DANGLING

Code	Description	Enabled	Reported	Platform
READ_DANGLING	Reading from a dangling pointer	✓	Runtime	Windows/Unix

READ_NULL

Code	Description	Enabled	Reported	Platform
READ_NULL	Reading NULL pointer	✓	Runtime	Windows/Unix

READ_OVERFLOW

Code	Description	Enabled	Reported	Platform
------	-------------	---------	----------	----------

READ_OVERFLOW			Runtime	Windows/Unix
(normal)	Reading overflow's memory		Runtime	Windows/Unix
(nonnull)	String is not NULL-terminated within range		Runtime	Windows/Unix
(string)	Alleged string does not begin within legal range		Runtime	Windows/Unix
(struct)	Structure reference out of range		Runtime	Windows/Unix

READ_UNINIT_MEM

Code	Description	Enabled	Reported	Platform
READ_UNINIT_MEM	Reading uninitialized memory		Runtime	Windows/Unix
(copy)	Copy from uninitialized region		Runtime	Windows/Unix
(read)	Use of uninitialized value		Runtime	Windows/Unix

READ_UNINIT_PTR

Code	Description	Enabled	Reported	Platform
READ_UNINIT_PTR	Reading from uninitialized pointer		Runtime	Windows/Unix

READ_WILD

Code	Description	Enabled	Reported	Platform
READ_WILD	Reading wild pointer		Runtime	Windows/Unix

RETURN_DANGLING

Code	Description	Enabled	Reported	Platform
RETURN_DANGLING	Returning pointer to local variable		Compilation	Windows/Unix

RETURN_FAILURE

Code	Description	Enabled	Reported	Platform
RETURN_FAILURE	Function call returned an error		Runtime	Windows/Unix

RETURN_INCONSISTENT

Code	Description	Enabled	Reported	Platform
RETURN_INCONSISTENT	Function has inconsistent return type		Compilation	Windows/Unix
(level 1)	No declaration, returns nothing		Compilation	Windows/Unix
(level 2)	Declared <code>int</code> , returns nothing		Compilation	Windows/Unix

(level 3)	Declared non- <code>int</code> , returns nothing	✓	Compilation	Windows/Unix
(level 4)	Returns different types at different statements	✓	Compilation	Windows/Unix

STL_ERROR

Insure++ can detect several STL usage errors on Unix systems:

- [Dereferencing an Unbound Iterator](#)
- [Dereferencing an Invalid Iterator](#)
- [Dereferencing an Iterator at End of Container](#)
- [Iterator Bound to Wrong Container](#)
- [Iterator Refers to Different Containers](#)
- [Iterator Does Not Constitute Valid Range](#)

UNUSED_VAR

Code	Description	Enabled	Reported	Platform
UNUSED_VAR	Unused variables	✗	Compilation	Windows/Unix
(assigned)	Variable is assigned but never used	✗	Compilation	Windows/Unix
(unassigned)	Variable is never used	✗	Compilation	Windows/Unix

USER_ERROR

Code	Description	Enabled	Reported	Platform
USER_ERROR	User-generated error message	✓	Runtime	Windows/Unix

WRITE_BAD_INDEX

Code	Description	Enabled	Reported	Platform
WRITE_BAD_INDEX	Writing array out of range	✓	Runtime	Windows/Unix




WRITE_DANGLING

Code	Description	Enabled	Reported	Platform
WRITE_DANGLING	Writing to a dangling pointer	✓	Runtime	Windows/Unix


WRITE_NULL

Code	Description	Enabled	Reported	Platform
WRITE_NULL	Writing to a <code>NULL</code> pointer	✓	Runtime	Windows/Unix


WRITE_OVERFLOW

Code	Description	Enabled	Reported	Platform
WRITE_OVERFLOW			Runtime	Windows/Unix
(normal)	Writing overflows memory		Runtime	Windows/Unix
(struct)	Structure references out of range		Runtime	Windows/Unix

WRITE_UNINIT_PTR

Code	Description	Enabled	Reported	Platform
WRITE_UNINIT_PTR	Writing to an uninitialized pointer		Runtime	Windows/Unix

WRITE_WILD

Code	Description	Enabled	Reported	Platform
WRITE_WILD	Writing to a wild pointer		Runtime	Windows/Unix