

EXP12-C. Do not ignore values returned by functions

Many functions return useful values whether or not the function has side effects. In most cases, this value is used to signify whether the function successfully completed its task or if some error occurred (see [ERR02-C. Avoid in-band error indicators](#)). Other times, the value is the result of some computation and is an integral part of the function's API.

Subclause 6.8.3 of the C Standard [[ISO/IEC 9899:2011](#)] states:

The expression in an expression statement is evaluated as a void expression for its side effects.

All expression statements, such as function calls with an ignored value, are implicitly cast to `void`. Because a return value often contains important information about possible errors, it should always be checked; otherwise, the cast should be made explicit to signify programmer intent. If a function returns no meaningful value, it should be declared with return type `void`.

This recommendation encompasses [ERR33-C. Detect and handle standard library errors](#). Unlike this recommendation, that rule is restricted to functions from the Standard C library.

Compliance with this recommendation is required in order to comply with [ERR00-C. Adopt and implement a consistent and comprehensive error-handling policy](#)

Noncompliant Code Example

This noncompliant code example calls `fputs()` and fails to check whether a write error occurs:

```
FILE* f = /* output file stream */
/* ... */
fputs("foo", f);
```

However, `fputs()` can fail and return `EOF`.

Compliant Solution

This compliant solution checks to make sure no output error occurred (see [ERR33-C. Detect and handle standard library errors](#)).

```
FILE* f = /* output file stream */
/* ... */
if (fputs("foo", f) == EOF) {
    /* Handle error */
}
```

Exceptions

EXP12-C-EX1: If the return value is inconsequential or if any errors can be safely ignored, such as for functions called because of their side effects, the function should be explicitly cast to `void` to signify programmer intent. For an example of this exception, see "Compliant Solution (Remove Existing Destination File)" under the section "Portable Behavior" in [FIO10-C. Take care when using the rename\(\) function](#), or Exception **ERR33-C-EX1** in [ERR33-C. Detect and handle standard library errors](#).

EXP12-C-EX2: If a function cannot fail or if the return value cannot signify an error condition, the return value may be ignored. Such functions should be added to a whitelist when automatic checkers are used.

```
strcpy(dst, src);
```

Risk Assessment

Failure to handle error codes or other values returned by functions can lead to incorrect program flow and violations of data integrity.

Recommendation	Severity	Likelihood	Remediation Cost	Priority	Level
EXP12-C	Medium	Unlikely	Medium	P4	L3

Automated Detection

Tool	Version	Checker	Description
Astrée	19.04	error-information-unused error-information-unused-computed	Fully checked
Axivion Bauhaus Suite	6.9.0	CertC-EXP12	Fully implemented
CodeSonar	5.2p0	LANG.FUNCS.IRV	Ignored return value
Compass /ROSE			
Coverity	2017.07	CHECKED_RETURN	Finds inconsistencies in how function call return values are handled. Coverity Prevent cannot discover all violations of this recommendation, so further verification is necessary
Cppcheck	1.66	leakRetValNotUsed, ignoredReturnValue	Return value of memory allocation function is not used. Ignored return value from function when configuration says it must be used. See the chapter "Library configuration" in the cppcheck manual
ECLAIR	1.2	CC2.EXP12	Fully implemented
Klocwork	2018	MISRA.FUNC. UNUSEDRET.2012 SV.RVT. RETVL_NOTTESTED	
LDRA tool suite	9.7.1	382 S	Fully implemented
Parasoft C /C++test	10.4.2	CERT_C-EXP12-a CERT_C-EXP12-b	The value returned by a function having non-void return type shall be used The value returned by a function having non-void return type shall be used
Polyspace Bug Finder	R2019b	CERT C: Rec. EXP12-C	Checks for situations where returned value of a sensitive function is not checked (rec. fully covered)
PRQA QA-C	9.7	3200	Fully implemented
PVS-Studio	6.23	V530, V698, V757, V797	
RuleChecker	19.04	error-information-unused	Partially checked
Splint	3.1.1		

Related Vulnerabilities

Search for [vulnerabilities](#) resulting from the violation of this rule on the [CERT website](#).

Related Guidelines

SEI CERT C++ Coding Standard	VOID EXP12-CPP. Do not ignore values returned by functions or methods
CERT Oracle Secure Coding Standard for Java	EXP00-J. Do not ignore values returned by methods
ISO/IEC TR 24772:2013	Passing Parameters and Return Values [CSJ]
MITRE CWE	CWE-754, Improper check for unusual or exceptional conditions

Bibliography

[ISO/IEC 9899:2011]	Subclause 6.8.3, "Expression and Null Statements"
-------------------------------------	---

