

## DCL09-C. Declare functions that return `errno` with a return type of `errno_t`

When developing new code, declare functions that return `errno` with a return type of `errno_t`. Many existing functions that return `errno` are declared as returning a value of type `int`. It is semantically unclear by inspecting the function declaration or prototype if these functions return an error status or a value or, worse, some combination of the two. (See [ERR02-C. Avoid in-band error indicators.](#))

C11 Annex K introduced the new type `errno_t` that is defined to be type `int` in `errno.h` and elsewhere. Many of the functions defined in C11 Annex K return values of this type. The `errno_t` type should be used as the type of an object that may contain only values that might be found in `errno`. For example, a function that returns the value of `errno` should be declared as having the return type `errno_t`.

This recommendation depends on C11 Annex K being implemented. The following code can be added to remove this dependency:

```
#ifndef __STDC_LIB_EXT1__
    typedef int errno_t;
#endif
```

### Noncompliant Code Example

This noncompliant code example shows a function called `opener()` that returns `errno` error codes. However, the function is declared as returning an `int`. Consequently, the meaning of the return value is not readily apparent.

```
#include <errno.h>
#include <stdio.h>

enum { NO_FILE_POS_VALUES = 3 };

int opener(
    FILE *file,
    size_t *width,
    size_t *height,
    size_t *data_offset
) {
    size_t file_w;
    size_t file_h;
    size_t file_o;
    fpos_t offset;

    if (file == NULL) { return EINVAL; }
    errno = 0;
    if (fgetpos(file, &offset) != 0) { return errno; }
    if (fscanf(file, "%zu %zu %zu", &file_w, &file_h, &file_o)
        != NO_FILE_POS_VALUES) {
        return -1;
    }

    errno = 0;
    if (fsetpos(file, &offset) != 0) { return errno; }

    if (width != NULL) { *width = file_w; }
    if (height != NULL) { *height = file_h; }
    if (data_offset != NULL) { *data_offset = file_o; }

    return 0;
}
```

This noncompliant code example nevertheless complies with [ERR30-C. Set `errno` to zero before calling a library function known to set `errno`, and check `errno` only after the function returns a value indicating failure.](#)

### Compliant Solution (POSIX)

In this compliant solution, the `opener()` function returns a value of type `errno_t`, providing a clear indication that this function returns an error code:

```

#define __STDC_WANT_LIB_EXT1__ 1

#include <errno.h>
#include <stdio.h>
#include <stdlib.h>

enum { NO_FILE_POS_VALUES = 3 };

errno_t opener(
    FILE *file,
    size_t *width,
    size_t *height,
    size_t *data_offset
) {
    size_t file_w;
    size_t file_h;
    size_t file_o;
    fpos_t offset;

    if (NULL == file) { return EINVAL; }
    errno = 0;
    if (fgetpos(file, &offset) != 0 ) { return errno; }
    if (fscanf(file, "%zu %zu %zu", &file_w, &file_h, &file_o)
        != NO_FILE_POS_VALUES) {
        return EIO;
    }

    errno = 0;
    if (fsetpos(file, &offset) != 0 ) { return errno; }

    if (width != NULL) { *width = file_w; }
    if (height != NULL) { *height = file_h; }
    if (data_offset != NULL) { *data_offset = file_o; }

    return 0;
}

```

This compliant solution is categorized as a POSIX solution because it returns `EINVAL` and `EIO`, which are defined by POSIX (IEEE Std 1003.1, 2013 Edition) but not by the C Standard.

## Risk Assessment

Failing to test for error conditions can lead to [vulnerabilities](#) of varying severity. Declaring functions that return an `errno` with a return type of `errno_t` will not eliminate this problem but may reduce errors caused by programmers' misunderstanding the purpose of a return value.

Recommendation	Severity	Likelihood	Remediation Cost	Priority	Level
DCL09-C	Low	Unlikely	Low	<b>P3</b>	<b>L3</b>

## Automated Detection

Tool	Version	Checker	Description
<a href="#">Axivion Bauhaus Suite</a>	6.9.0	<b>CertC-DCL09</b>	
<a href="#">LDRA tool suite</a>	9.7.1	<b>634 S</b>	Partially Implemented

## Related Vulnerabilities

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

## Related Guidelines

<a href="#">SEI CERT C++ Coding Standard</a>	<a href="#">VOID DCL09-CPP. Declare functions that return errno with a return type of errno_t</a>
<a href="#">ISO/IEC TR 24772:2013</a>	Ignored Error Status and Unhandled Exceptions [OYB]

# Bibliography

[IEEE Std 1003.1:2013]

