PRE08-C. Guarantee that header file names are unique

Make sure that included header file names are unique. According to the C Standard, subclause 6.10.2, paragraph 5 [ISO/IEC 9899:2011],

> *The implementation shall provide unique mappings for sequences consisting of one or more nondigits or digits (6.4.2.1) followed by a period (.) and a single nondigit. The first character shall not be a digit. The implementation may ignore distinctions of alphabetical case and restrict the mapping to eight significant characters before the period.*

This means that

- Only the first eight characters in the file name are guaranteed to be significant.
- The file has only one nondigit character after the period in the file name.
- The case of the characters in the file name is not guaranteed to be significant.

To guarantee that header file names are unique, all included files should differ (in a case-insensitive manner) in their first eight characters or in their (one-character) file extension.

Note that compliance with this recommendation does not require that short file names are used, only that the file names are unique.

Noncompliant Code Example

This noncompliant code example contains references to headers that may exist independently in various environments but can be ambiguously interpreted by a C-compliant compiler:

```
#include "Library.h"
#include <stdio.h>
#include <stdlib.h>
#include "library.h"

#include "utilities_math.h"
#include "utilities_physics.h"

#include "my_library.h"

/* ... */
```

`Library.h` and `library.h` may refer to the same file. Also, because only the first eight characters are guaranteed to be significant, it is unclear whether `utilities_math.h` and `utilities_physics.h` are parsed. Finally, if a file such as `my_libraryOLD.h` exists, it may inadvertently be included instead of `my_library.h`.

Compliant Solution

This compliant solution avoids the ambiguity by renaming the associated files to be unique under the preceding constraints:

```
#include "Lib_main.h"
#include <stdio.h>
#include <stdlib.h>
#include "lib_2.h"

#include "util_math.h"
#include "util_physics.h"

#include "my_library.h"

/* ... */
```

The only solution for mitigating ambiguity of a file, such as `my_libraryOLD.h`, is to rename old files with either a prefix (that would fall within the first eight characters) or add an extension (such as `my_library.h.old`).

Exceptions

PRE08-C-EX1: Although the C Standard requires only the first eight characters in the file name to be significant, most modern systems have long file names, and compilers on such systems can typically differentiate them. Consequently, long file names in headers may be used, provided that all the implementations to which the code is ported can distinguish between these file names.

Risk Assessment

Failing to guarantee uniqueness of header files may result in the inclusion of an older version of a header file, which may include incorrect macro definitions or obsolete function prototypes or result in other errors that may or may not be detected by the compiler. Portability issues may also stem from the use of header names that are not guaranteed to be unique.

Recommendation	Severity	Likelihood	Remediation Cost	Priority	Level
PRE08-C	Low	Unlikely	Medium	P2	L3

Automated Detection

Tool	Version	Checker	Description
Axivion Bauhaus Suite	6.9.0	CertC-PRE08	
ECLAIR	1.2	CC2.PRE08	Fully implemented
PRQA QA-C	9.5	5002	Fully implemented

Related Vulnerabilities

Search for vulnerabilities resulting from the violation of this rule on the CERT website.

Related Guidelines

SEI CERT C++ Coding Standard	VOID PRE08-CPP. Guarantee that header file names are unique

Bibliography

[ISO/IEC 9899:2011]	Subclause 6.10.2, "Source File Inclusion"