FIO39-C. Do not alternately input and output from a stream without an intervening flush or positioning call

The C Standard, 7.21.5.3, paragraph 7 [ISO/IEC 9899:2011], places the following restrictions on update streams:

> *When a file is opened with update mode . . ., both input and output may be performed on the associated stream. However, output shall not be directly followed by input without an intervening call to the `fflush` function or to a file positioning function (`fseek`, `fsetpos`, or `rewind`), and input shall not be directly followed by output without an intervening call to a file positioning function, unless the input operation encounters end-of-file. Opening (or creating) a text file with update mode may instead open (or create) a binary stream in some implementations.*

The following scenarios can result in undefined behavior. (See undefined behavior 151.)

- Receiving input from a stream directly following an output to that stream without an intervening call to `fflush()`, `fseek()`, `fsetpos()`, or `rewind()` if the file is not at end-of-file
- Outputting to a stream after receiving input from that stream without a call to `fseek()`, `fsetpos()`, or `rewind()` if the file is not at end-of-file

Consequently, a call to `fseek()`, `fflush()`, or `fsetpos()` is necessary between input and output to the same stream. See ERR07-C. Prefer functions that support error checking over equivalent functions that don't for more information on why `fseek()` is preferred over `rewind()`.

Noncompliant Code Example

This noncompliant code example appends data to a file and then reads from the same file:

```
#include <stdio.h>

enum { BUFFERSIZE = 32 };

extern void initialize_data(char *data, size_t size);

void func(const char *file_name) {
  char data[BUFFERSIZE];
  char append_data[BUFFERSIZE];
  FILE *file;

  file = fopen(file_name, "a+");
  if (file == NULL) {
    /* Handle error */
  }

  initialize_data(append_data, BUFFERSIZE);

  if (fwrite(append_data, 1, BUFFERSIZE, file) != BUFFERSIZE) {
    /* Handle error */
  }
  if (fread(data, 1, BUFFERSIZE, file) < BUFFERSIZE) {
    /* Handle there not being data */
  }

  if (fclose(file) == EOF) {
    /* Handle error */
  }
}
```

Because there is no intervening flush or positioning call between the calls to `fread()` and `fwrite()`, the behavior is undefined.

Compliant Solution

In this compliant solution, `fseek()` is called between the output and input, eliminating the undefined behavior:

```
#include <stdio.h>

enum { BUFFERSIZE = 32 };
extern void initialize_data(char *data, size_t size);

void func(const char *file_name) {
  char data[BUFFERSIZE];
  char append_data[BUFFERSIZE];
  FILE *file;

  file = fopen(file_name, "a+");
  if (file == NULL) {
    /* Handle error */
  }

  initialize_data(append_data, BUFFERSIZE);
  if (fwrite(append_data, BUFFERSIZE, 1, file) != BUFFERSIZE) {
    /* Handle error */
  }

  if (fseek(file, 0L, SEEK_SET) != 0) {
    /* Handle error */
  }

  if (fread(data, BUFFERSIZE, 1, file) != 0) {
    /* Handle there not being data */
  }

  if (fclose(file) == EOF) {
    /* Handle error */
  }
}
```

Risk Assessment

Alternately inputting and outputting from a stream without an intervening flush or positioning call is undefined behavior.

Rule	Severity	Likelihood	Remediation Cost	Priority	Level
FIO39-C	Low	Likely	Medium	P6	L2

Automated Detection

Tool	Version	Checker	Description
Astrée	19.04		Supported, but no explicit checker
Compass/ROSE			Can detect simple violations of this rule
LDRA tool suite	9.7.1	**84 D**	Fully implemented
Parasoft C/C++test	10.4.2	**CERT_C-FIO39-a**	Do not alternately input and output from a stream without an intervening flush or positioning call
Polyspace Bug Finder	R2019a	CERT C: Rule FIO39-C	Checks for alternating input and output from a stream without flush or positioning call (rule fully covered)
PRQA QA-C	9.5	**5029**	

Related Vulnerabilities

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

Related Guidelines

Key here (explains table format and definitions)

Taxonomy	Taxonomy item	Relationship

CERT C	FIO50-CPP. Do not alternately input and output from a file stream without an intervening positioning call	Prior to 2018-01-12: CERT: Unspecified Relationship
ISO/IEC TS 17961: 2013	Interleaving stream inputs and outputs without a flush or positioning call [ioileave]	Prior to 2018-01-12: CERT: Unspecified Relationship
CWE 2.11	CWE-664	2017-07-10: CERT: Rule subset of CWE

CERT-CWE Mapping Notes

Key here for mapping notes

CWE-664 and FIO39-C

CWE-664 = Union(FIO39-C, list) where list =

- Improper use of an object (besides alternating reading/writing a file stream without an intervening flush

This CWE is vague on what constitutes “improper control of a resource”. It could include any violation of an object’s method constraints (whether they are documented or not). Or it could be narrowly interpreted to mean object creation and object destruction (which are covered by other CWEs).

Bibliography

[ISO/IEC 9899:2011]	7.21.5.3, "The `fopen` Function"