

Clause 4.8 . I think this needs to be subdivided into subclauses as follows. The para on collectives was deleted by mistake last time.

4.8.1 Images and coarrays

Paras 1 & 2.

4.8.2 Locks

Para 3.

4.8.3 Teams

Para 4.

4.8.4 Segmentts

Para 5.

4.8.5 Asynchronous variables

Para 6.

4.8.6 Volatile variables

Para 7

4.8.7 Collective subroutines

There are several intrinsic subroutines that are *collective* in the sense that the images of the current team collaborate to perform an action, such as summation.

4.8.8 Image failure

Para 8.

4.8.9 Do concurrent

Para 9.

Clause 4.8. Remove the comments SM1, SM2, and SM3 (or explain what is needed).

Clause 6.61. Change the text to

6.61.1 Applicability to language

The vulnerability as described in ISO/IEC 24772-1 clause 6.61 applies to Fortran. Coarray data are accessible to all images by using image selectors in square brackets. There are several mechanisms, see clause 4.8, for separating the alteration of the value of a coarray variable on one image from its access by another image. To ensure correct execution, it is essential to use one or more of these mechanisms.

6.61.2 Guidance to language users

To ensure correct execution when executing on more than one image, use one or more of the following mechanisms.

- The `sync_all` statement may be used to separate the alteration of the value of a coarray variable on one image from its access by any other image.
- The `sync_images` statement may be used to separate the alteration of the value of a coarray variable on one image from its access by an image in a specified set of images.
- A collective subroutine should be used whenever it is suitable.
- Integer variables of kind `atomic_int_kind` and logical variables of kind `atomic_logical_kind` may be referenced and defined in unordered segments by atomic intrinsic subroutines including `atomic_define`, `atomic_ref`, and `atomic_or`. The system ensures that for each such variable all such actions occur sequentially.
- An `event post` statement may be used on one image and a corresponding `event wait` statement may be used on another. Statements executed on the first image ahead of its execution of the `event post` statement precede statements executed on the other image after its execution of the corresponding `event wait` statement.
- A critical section may be used to limit execution to one image at a time; if performance using critical sections is unacceptable, use locks and use analysis to show correct lock behaviour.

Avoid

- The use of the `volatile` attribute.
- The use of the `asynchronous` attribute except for use with a parallel-processing package such as MPI for nonblocking data transfer.
- The use of the `sync memory` statement for defining and ordering segments.

Clause 6.62.1. Remove the comment SM21 (moved to part 1).

Clause 6.64. Change the text to

6.64.1 Applicability to language

Most of the vulnerability as described in ISO/IEC 24772-1 clause 6.64 does not apply to Fortran. Fortran provides the ability to control input or output via format strings and mistakes in format strings may cause serious program errors. However, the format string cannot affect the access of memory beyond the data items being referenced.

6.64.2 Guidance to language users

- Wherever possible, use format strings that are constants.
- Where a variable string is needed, check that its value is within expectations.

Clause 6.65.1. In para 2.

In sentence 1, change “prevent” to “prevents”. In sentence 2, change “compilers usually do not prevent” to “permits”. Add sentence 3 “Compilers prevent the alteration of the value of a constant.” Remove comment [no other way to modify constant found].