

The new features of Fortran 2015

**John Reid, JKR Associates and
Rutherford Appleton Laboratory**

BCS Fortran Specialist Group
London, 28 September 2017

Abstract

I will provide a quick overview of the new features of Fortran 2015.

For details, see N2127 on the WG5 site,
`https://wg5-fortran.org/`

Further interoperability with C

Fortran 2003 provides for interoperability of procedures with arguments that are scalars, explicit-shape arrays, or assumed-size arrays, and are not optional.

It does not provide for arguments that are **assumed shape, allocatable, pointer, or optional**.

Fortran 2015 fills this gap by defining C descriptors for such arguments. They are provided by the system when calling C from Fortran and must be constructed by C code for calling Fortran.

Fortran 2015 also allows C functions to accept arguments of **any rank or any type**.

Further coarray features: Teams

Needed for independent computations on subsets of images.

change team construct defines division into subteams and back.

Code that has been written and tested on whole machine should run on a team, so image indices are within team.

Collective activities, including syncs and allocations, are relative to team.

Further coarray features: Collectives

The collective subroutines are:

co_broadcast,

co_max, co_min, co_sum,

co_reduce .

Invoked by the same statement on all images of the team and involve synchronization within them.

Further coarray features: Image failure

There is provision for image failure because of the huge numbers of images likely to be in use.

Once failed, an image remains so.

Can test by adding **stat**=variable to image control statement or remote reference.

failed_images intrinsic lists failed images in a team.

New IEEE standard

A large number of detailed changes have been made for conformance with the new IEEE standard for floating-point arithmetic.

Minor changes

- Specification of locality within **do concurrent**
- For entities accessed from a module, allow specification of the default for **public/private**
- **random_init (repeatable, image_distinct)** controls initialization of random number generator.
- Procedures are recursive by default
- Many more

New Obsolescences and Deletions

Deleted:

- Arithmetic IF
- Shared DO termination and DO termination other than **end do** or **continue**

Obsolescent:

- **equivalence, common & block data**
- Labelled DO loops
- Specific names for intrinsic functions
- **forall** construct

Timetable

(ISO Limit: Sept 2019)

May 2017	First CD ballot comments available
July 2017	Second CD ballot initiated
Sept 2017	Second CD ballot comments available
Oct 2017	DIS constructed
Nov 2017	DIS ballot initiated
Apr 2018	DIS ballot results available
May 2018	DIS revised
Aug 2018	Standard published