bcs
The Chartered Institute for IT

Enabling the
information society

Removing simple deficiencies in Fortran 2015

David Muxworthy d.muxworthy @ bcs.org.uk 1 October 2015

Revising the Fortran 2008 standard

It was decided in 2012 that the revision of the standard should consist of incorporation into the main standard of:

- Agreed editorial improvements
- Corrigenda
- Technical Specification on Further Interoperability with C
- Technical Specification on Additional Parallel Features
- Removal of simple deficiencies in, and discrepancies between, existing facilities

In fact some minor 'improvements' have also been approved

Selecting deficiencies to address

Proposals for 'wart removal' were made by member countries of WG5, resulting in just over 50% of them (34 items) being approved.

These can be classified as follows:

Types	1
Expressions	1
Execution control	4
I/O and formatting	7
Program units	1
Procedures	8
Intrinsic procedures	10
IEEE compatibility	
New obsolescences and deletions	

Types

- In an array constructor implied-DO, add optional type-spec for the ac-do-variable (to conform to FORALL and DO CONCURRENT)

Proposed but not accepted:

- *Default KINDs for constants and intrinsics*
- *Generalize enumerators to avoid repeating list of names*
- *Provide PROTECTED types to encompass EVENT_TYPE and LOCK_TYPE*
- *Allow a scalar assumed-type actual argument to correspond to an assumed-rank dummy argument*
- *Allow the TARGET attribute for derived-type components*
- *Relax restrictions on arguments to type-bound procedures*

Specifications

No proposals were accepted

Proposed but not accepted:

- *Provide syntax to allow use of an array to specify the bounds of an new array*
- *Decouple allocation and initialization of objects*
- *Allow assumed coshape dummy coarrays*

Expressions

- Allow any constant properties of an object to be referenced within an initialization in its entity declaration
  e.g. allow such as

```
integer :: B = bit_size(B)
real    :: E = sqrt(sqrt(epsilon(E)))
```

Proposed but not accepted:

- *Conditional expressions and conditional arguments*
- *Allow dummy arguments in functions used in specification expressions*
- *Allow concatenation of characters of different kinds*
- *Allow intrinsic assignment to a polymorphic variable that is not allocatable, with certain provisos*

Execution control

- Remove restriction on ERROR STOP and unnecessary constraints on stop codes
- Allow specification of locality within DO CONCURRENT
- New SELECT RANK construct

Proposed but not accepted:

- *Extend SELECT TYPE to avoid having multiple nested blocks*

Input/output and Formatting - 1

- Remove prohibition on a file being connected to more than one unit at one time
- With advancing input, if SIZE= is specified do not require also ADVANCE=
- Specify value returned by INQUIRE for RECL= for unlimited records
- Standardize certain error conditions in numeric formatted input
- Allow w=0 in Ew.d, ESw.d, ENw.d edit descriptors, to conform with G0.d
- Allow the .d in G0.d to be ignored if the corresponding value is not real or complex
- Allow e=0 in Ew.dEe etc, to specify minimal exponent width

Input/output and Formatting - 2

Proposed but not accepted:

- *A0 should be defined and should imply TRIM*
- *Provide output facilities:*
  - *to suppress leading zeroes in exponents*
  - *to suppress the sign for exponents*
  - *to specify the default exponent width*
- *Allow * in a FLUSH statement to indicate OUTPUT_UNIT*

Program Units

- For entities accessed from a module, allow specification of the default accessibility (by permitting a module name in a PUBLIC or PRIVATE statement)

Procedures - 1

- Require diagnosability of nonstandard intrinsic modules et al
- Make RECURSIVE the default for procedures
- Allow a GENERIC statement as alternative to interface block
- Remove anomaly of INTENT(IN) and VALUE for generic interfaces
  (update from Fortran 2003 to 2008 was not applied consistently)
- Extend host association to be able to control both importation and hiding a host entity
- Remove potential failure in generic disambiguation rules
- Allow IMPLICIT NONE to be used to require EXTERNAL to be specified explicitly
- Permit modification of VALUE arguments in a pure function and specify that procedures in ISO_C_Binding, other than C_F_POINTER, are pure.

Procedures - 2

Proposed but not accepted:

- *Allow a dummy argument with the VALUE attribute to have the VOLATILE attribute. (Since the volatility commences after it gets its value.)*

Intrinsic Procedures - 1

- New intrinsic functions:
  - REDUCE
  - COSHAPE
  - OUT_OF_RANGE (Check whether values of one intrinsic type can be converted to another without error)
- Enhanced functions:
  - CMPLX
  - SIGN
  - EXTENDS_TYPE_OF, SAME_TYPE_AS
  - RANDOM_NUMBER (Require the processor to provide independent random-number sequences on different images)
- In general, improve consistency of:
  - specifying KIND for intrinsic arguments
  - specifying DIM in intrinsic argument lists
  - providing optional ERRMSG arguments
- Amend some command line related intrinsic procedures to accommodate teams

Intrinsic Procedures - 2

Proposed but not accepted:

- *Intrinsic functions to test for equality of bit strings*
- *Incorporate some intrinsic functions from the Varying Length String standard*
- *Allow IAND, IEOR, IOR to have any number of arguments*
- *Intrinsic function to find all elements of an array matching a particular value*
- *Intrinsic function to merge pointers*
- *Intrinsic function to query allocatable or associated status of a variable*
- *Allow specification of lower bound in MOVE_ALLOC*
- *Allow a nonoptional argument to PRESENT*
- *Delete the minimum requirement on the length of certain arguments to command line intrinsic functions*
- *Add ORDERED option to FINDLOC*

IEEE conformance

Intrinsic modules IEEE_EXCEPTIONS, IEEE_ARITHMETIC, and IEEE_FEATURES provide support for the facilities commonly known as 'IEEE arithmetic'. Whether the modules are provided is processor dependent.

Fortran 2008 supports the facilities defined by IEC 60559:1989.

Fortran 2015 supports the facilities defined by ISO/IEC/IEEE 60559:2011. There are many new features and extensions to existing ones.

New Obsolescences and Deletions

Deleted:

- Arithmetic IF
- Shared DO termination and termination on a statement other than END DO or CONTINUE

Obsolescent:

- EQUIVALENCE
- COMMON & BLOCK DATA
- Labelled DO loops
- Specific names for intrinsic functions
- FORALL construct

Proposed for obsolescence but not accepted:

- *Implicit interfaces*
- *WHERE statements and constructs*

- *Un-obsolesce CHARACTER* form*

References

WG5 documents are accessible via its Complete Document Register at:
http://www.nag.co.uk/sc22wg5/docs.html

German proposals for the revision: WG5-N2015
UK proposals: WG5-N1975, WG5-N2016 and WG5-N2059
US proposals: (2013) http://j3-fortran.org/doc/meeting/200/13-244r1.txt
(2014) http://j3-fortran.org/doc/meeting/203/14-100r1.txt, also 14-128r1.txt
http://j3-fortran.org/doc/meeting/204/14-165.txt, also 14-204.txt
(2015) WG5-N2064

List of accepted proposals: WG5-N2082
Late (not final) draft of TS 18508 Additional Parallel Features in Fortran: WG5-N2056

Latest (incomplete) draft of the revised language standard:
http://j3-fortran.org/doc/meeting/208/15-007r2.pdf

Minutes of relevant WG5 meetings: WG5-N1977, WG5-N2020 and WG5-N2068

The voting procedure for proposals is described on page 4 of
http://www.fortran.bcs.org/2013/removing_deficiences.pdf