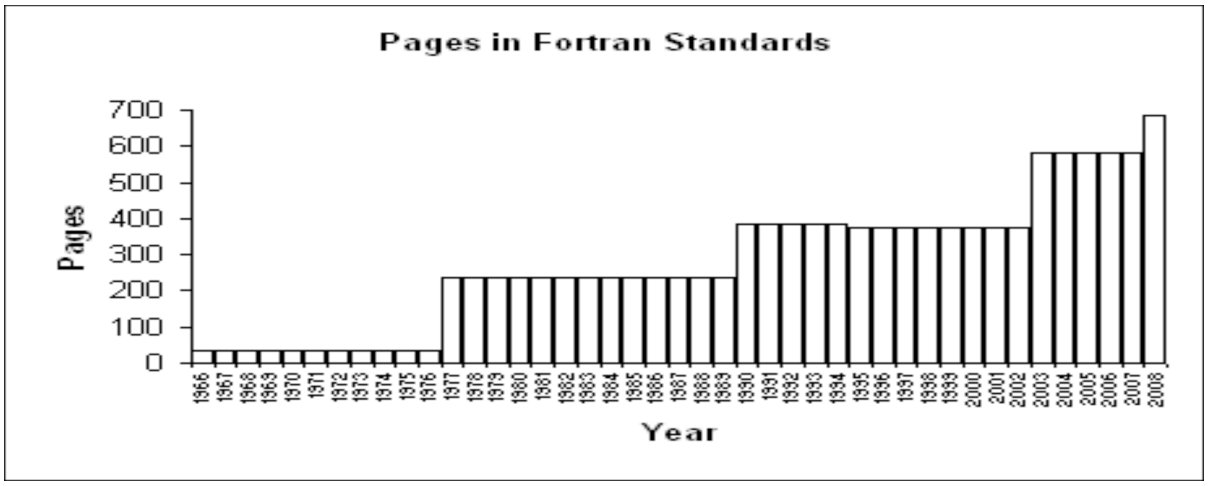
**Discussion paper on the draft of the next Fortran standard**

**David Muxworthy, BSI Fortran Convenor**

This note is prepared for the BCS Fortran Specialist Group AGM on 14 June 2007 and intended to be read in conjunction with John Reid's paper "What will be in Fortran 2008". This is referenced below as JKR.

In May 2005 WG5 considered the requirements put forward by the various member countries and allocated them into three groups: required (i.e. must be done). allowed (second priority - to be done if resources are available) and rejected (at least for this revision). The intention was to produce a minor revision of the language, much as Fortran 95 was a minor revision of Fortran 90. However as the following chart shows, the language, and the draft revised standard, is turning out to be far more than a minor revision.




For reasons of space, JKR covers only a part of the work that has gone into the revision. The full list of requirements and their disposition is shown in the table at the end of this note.

Apart from three items (one first priority and two second priority) that have fallen by the wayside since May 2005, it will be seen that all of the approved items have been progressed and incorporated into the draft standard. All, that is apart from the plugging the gap in interoperability with C which had been not been covered in Fortran 2003. F03 allows the sharing of some forms of data with C but does not cope with procedures that have data pointer, allocatable, assumed-shape array, or optional dummy arguments. Because this feature was taking longer than anticipated to develop it was decided to remove it from the draft standard and to make it the subject of a Technical Report. This means that the facilities described in the report would be expected to be incorporated into a subsequent standard and meanwhile would be available to vendors to implement if they wished. Of the three large items described in JKR, one (co-arrays) was a first priority item and BITS and intelligent macros were second priority; the original intention had been to process at most one large item.

The schedule to which WG5 has been working requires that the first working draft be available in May 2007 and be approved by WG5 by September. After further refinement the document should be technically complete by September 2008 and, following due ISO processing, should be published as an ISO Standard in August 2009.

At the time of writing (prior to the May meeting of J3), edits for the individual requirements have largely been completed but integrating them into the full working draft has revealed 17 unresolved issues. Moreover there remain 50 outstanding interpretation requests relating to F03.

This Fortran Specialist Group meeting gives an opportunity to discuss British opinions on the status of the revision prior to the next WG5 meeting, which is to be held at the BCS offices in London in August 2007. There are two major things to consider: (a) do we approve of the content of the revision, and (b) do we approve of the schedule?

Concern has been expressed that the content of next standard has been fixed before compilers for the full Fortran 2003 language have even been released. Thus there is no body of experience of use of F03 which in itself might be expected to give rise to further requirements from users. F03 was a major revision and compilers have appeared far more slowly than they did for F90 and for F95. While there is considerable merit in setting and adhering to a schedule, which has largely been achieved, it is not now clear that this is necessarily in the best interests of users or vendors.

While there is demand for the new facilities from some users, there is clearly confusion for example amongst many contributors to comp.lang.fortran even about the compatibility between F77 and F90/95. Would the premature appearance of F08 give the impression, albeit grossly mistaken, that Fortran is not a stable base for writing long-term applications?

Concern has also been expressed that the emphases in the revised language have turned out wrongly. For example it has been argued that it was a major mistake to leave interoperability with C unfinished in F03 and that filling this hole, which is now scheduled to be delayed until a later revision, would have been of more value to more users than adding the large new items.

It has been argued that co-arrays 'clutter up' the language and make it larger and more complicated to the benefit of relatively few users, and hence that co-arrays should be made a separate optional part, or the subject of a technical report, rather than part of the base language. It has been argued that the model used to implement type BITS is essentially misconceived and moreover it precludes future generalization. It has been argued that since a minor revision was promised, and that the typical scientific or engineering user would benefit from a period of stability while they became accustomed to using F03, that all three large new items be removed from this revision. This would also facilitate delivery of F08 compilers.

More fundamentally perhaps, given the slow production of F03 processors, would a large extension to the language next year give vendor companies an excuse to abandon Fortran development altogether? In recent years we have seen EPC (now Analog Devices) and Salford Software in the UK give up Fortran production.

A counter-argument to relaxing the schedule is that history shows that this would bring strong pressure from some quarters to add yet more new features to the language. Discussion on the points outlined above, and on any related topic, will be most welcome. For those unable to attend the meeting, BCS members may use the discussion area in the 'Fortran Members' part of the BCS Secure Area website. Contributions by e-mail (to comp-fortran-90@jiscmail.ac.uk) are also welcome.

## REQUIREMENTS CONSIDERED FOR REVISION OF FORTRAN 2003

### First priority items

Mentioned in JKR:
J3-003 EXECUTE_COMMAND_LINE
J3-013 Internal subprograms as actual arguments and procedure pointer targets
J3-019 More mathematical functions
J3-039 Max rank + co-rank .LE. 15
J3-043 Pointers to contiguous memory + more contiguous
J3-046 DO CONCURRENT construct
RU-003 Obsolesce ENTRY
UK-001 Co-array Fortran for parallel programming
UK-005 Long Integers

Minor items not mentioned in JKR:
J3-001 Enhanced STOP
J3-002 Get unused I/O unit somehow
J3-008 Rewrite attribute requirements (initially second priority, promoted in 2006)
J3-010 Allow empty CONTAINS part
J3-020 Allow TYPE ( *intrinsic-type-spec* )
J3-027 ASCII arguments for LGE etc.
UK-002 Decimal floating point arithmetic
UK-007 Pointer function reference as actual argument

### Second priority items

Mentioned in JKR:
J3-014 Intelligent macros
J3-018 Non-null initial targets for pointers
J3-038 Libm: Bessel, erf, gamma, hypot
J3-047 BITS
UK-011 Impure elemental

Minor items not mentioned in JKR:
J3-004 STORAGE_SIZE
J3-005 C_SIZEOF
J3-012 Use ALLOCATABLE and POINTER attributes in generic resolution
J3-015 Updating complex parts
J3-016 Nonpointer nonallocatable optional dummy is not present if corresponding actual is dissociated or deallocated
J3-022 Allow a polymorphic allocatable variable in intrinsic assignment
J3-023 Named array constant's extents from its *initialization-expr*
J3-024 EXIT from any labeled construct
J3-025 SUBROUTINE *name* or FUNCTION *name* optional on END statements
J3-026 ATAN with two arguments works like ATAN2
J3-028 Forward type for allocatable components
J3-030 Simplified means to select the most commonly desired real and integer kinds
J3-032 Findloc
J3-033 Compiler Version etc.
J3-034 Mold on Allocate
J3-035 Proposed f2k+ MTE on semicolons
J3-048 Writing Comma Separated Value files
UK-008 Pointer function reference as asg stmt LHS
UK-009 Use procedureness in generic resolution
UK-012 Recursive I/O to different unit

### Second priority item moved to Technical Report

J3-041 Interoperability of pointers, allocatables, assumed-shape arrays and optional arguments

### Proposals combined with others

J3-006 Find all available logical and character kinds
J3-042 Interoperability of optional arguments
RU-001 Remove restriction on the maximum rank of arrays

RU-002 Extend the semantics of the EXIT statement
RU-005 Extend a set of array intrinsic functions (reduced)
RU-006 Give a table with attribute compatibility

**Rejected items**

J3-007 Construct Name Local to Construct
J3-009 IO_UNIT standard derived type (initially second priority, rejected in 2006)
J3-011 Coroutines
J3-017 Default initial values for absent optional dummy arguments
J3-021 Resolve generic without invoking a procedure or evaluating arguments
J3-029 More info about GET_COMMAND[_ARGUMENT] failure
J3-031 ANDTHEN and ORELSE pseudo-functions
J3-036 Use, Except
J3-037 Pointers and Targets
J3-040 Compute if actual arg is present
J3-044 New Intents (initially first priority, rejected in 2006)
J3-045 Same Assumed Shape declaration
J3-049 Select between expressions
RU-003 Delete statement functions
RU-004 Subset of Fortran Standard which does not include redundant features
UK-003 Conformance to IEEE 754R (initially second priority, rejected in 2006)
UK-004 KIND environment specification
UK-006 Multiple Nonzero-Rank Part References
UK-010 Partial initialization of PARAMETERs

**Links**

The initial requirements for development of the language are shown in more detail in WG5 document N1649 at ftp://ftp.nag.co.uk/sc22wg5/N1601-N1650/N1649.txt.

The development status of each item is in J3 document 010 at http://j3-Fortran.org/doc/standing/links/010.pdf; the references in the tables in 010 are to documents in the J3 repository, which is at http://www.j3-fortran.org/.

The complete working draft of the revised standard, which varies from meeting to meeting, is at the time of writing at http://www.j3-fortran.org/doc/meeting/180/07-007r1.pdf (or ...ps.bz2).