Compiler Support for the Fortran 2003 and 2008 Standards
Revision 7

Ian D Chivers & Jane Sleightholme
Ian Chivers: Rhymney Consulting, London.
Jane Sleightholme: FortranPlus, London.
ian@rhymneyconsulting.co.uk
jane@fortranplus.co.uk

Introduction

This is a repeating article in Fortran Forum. The first version appeared in Fortran Forum in April 2007. The basis for the entries in the original list of features was a report by John Reid. An electronic version can be found at:

`ftp://ftp.nag.co.uk/sc22wg5/N1601-N1650/N1648.pdf`

If you are a compiler vendor and would like to be included in future versions of this table please email one of us with details and they will be added to the table and published in Fortran Forum.

Acknowledgements for the original article

An email was sent to the J3 list asking for information about compiler support for the new features of the Fortran 2003 standard. The following people have contributed to the original article:

- Bill Long, Cray

- Joost VandeVondele

- Van Snyder

- Tobias Burnus and Brooks Moses, gfortran

- Andy Vaught, g95

- Robert Holmes, NAG

Thanks.

Revision 1

Two new compiler vendors were added. The information on the Intel compiler has been taken from the release notes that came with release 10 of the compiler. The information on the IBM entry has been taken from their web site. Ian Bush posted an article to comp.lang.fortran regarding this release (IBM XL Fortran Enterprise Edition for AIX, V11.1). Thanks Ian. See

`http://publib.boulder.ibm.com/infocenter/comphelp/v9v111/index.jsp?topic=/com.`
`ibm.xlf111.aix.doc/getstart/new_features.htm`

for more information.

Intel and IBM were contacted to ask them to verify the information.

- Jim Xia of IBM corrected their entry, thanks Jim.

- Stan Whitlock of Intel corrected their entry, thanks Stan.

If there are any errors please notify us and we will correct them in the next version of this article.

Revision 2

Sun has been added and there have been a few corrections and updates to some of the other entries.

- Michael Ingrassia of Sun corrected their entry, thanks Michael.

Revision 3

The entries for Cray, gfortran (11 changes) , Intel (18 changes) and NAG (9 changes) have been updated.

Revision 4

We've had replies from Cray (Bill Long) , gfortran (Tobias Burnus), g95 (Andy Vaught), Intel (Stan Whitlock), Nag (Malcolm Cohen) and Sun (Robert Corbett).

We've also added two entries suggested by Richard Maine. Here is the text of the message we received from Richard.

I just got the latest Fortran Forum and noticed two somewhat related Fortran 2003 features that I personally think are important, but aren't reflected in your table of features. If convenient, they might be useful to add to the table.

1. Allocatable scalars. To me, this is an important feature for object orientation, and in particular for polymorphism. Basically, a polymorphic object has to be either a pointer or an allocatable (or a dummy argument, which is a bit restrictive). In my experiments with polymorphism, the polymorphic objects pretty much always naturally "wanted" to be allocatable. But the NAG compiler (which I was using at the time) didn't yet support allocatable scalars. This meant that I either needed to make all the polymorphic objects pointers or make them arrays (possibly of size 1). Neither of these alternatives was attractive at all. I found this a significant enough shortcoming to keep me from using the polymorphic features. Thus, I'd think this would be something people would want to know about a compiler if they planned to use polymorphism.

2. Allocatable character length. I think that allocatable character length is one of the biggest "sleeper" features of f2003. It wasn't even on the list of f2003 requirements, and thus sometimes doesn't show up in lists of new features. It just naturally arise from allocatable length parameters for parameterized derived types. It seemed like one should allow the same thing for the one intrinsic type with a length type parameter. And lo, when it was all put together, it seemed like this was finally a good way to do variable length strings in Fortran. It integrates with the rest of the language immensely better than iso_varying_string has any hope of doing. In fact, as I said, it integrates so well that it came about as a consequence of the integration of other features. Allocatable-length character strings act like so many people intuitively think of character strings, unlike the fixed-length character strings that we've had since f77.

Although this is related to allocatable scalars, in that you certainly want to be able to have allocatable character strings that are scalar, it is also a separate feature in that you can have allocatable scalars without necessarily allowing character length to be allocatable. It is also different in application, in that I see the main other usage of allocatable scalars as being for polymorphism, whereas allocatable character strings are not much related to polymorphism. It is also useful independent of parameterized derived types. I personally expect to see allocatable character strings used far more than parameterized derived types, even though it was the requirement for parameterized derived types that lead to allocatable character strings. I could almost see allocatable character strings as becoming the "normal" way that most character string variables are done.

Thanks Richard.

We've also added entries for the Fortran 2008 standard. The entries are based on the following document:

`ftp://ftp.nag.co.uk/sc22wg5/N1701-N1750/N1729.pdf`

The last change are entries for compilers that support the Fortran 95 standard, and a list of compilers that are no longer under development but did support Fortran 90, and finally compilers which are available but we have no information on.

We have included the above for completeness. Given the widely differing levels of compiler standard conformance today we wanted to make this information available to people choosing a compiler.

Thanks to everyone who has provided the data.

Revision 5

The IBM entry has been updated. See

`http://www-01.ibm.com/common/ssi/rep_ca/3/897/ENUS210-103/ENUS210-103.PDF`

The entry for gfortran has been updated. There is an entry for HP. The entry refers to the March 2010 release.

John Reid has also updated N1729.pdf and the latest version can be found at

`ftp://ftp.nag.co.uk/sc22wg5/N1801-N1850/`

Revision 6

The gfortran entry has been updated. Here is part of an email we received from Tobias Burnus.

- Hi, as the development of GCC has almost reached the end of Stage 1, I thought I could already update the F2003/F2008 conformance status for the December issue of ACM Fortran Forum. (Stage 1 allows for larger changes; it is followed by Stage 3 (!) which allows only smaller bugfixes, regression fixes and documentation updates.) Past experience suggests that 4.6.0 will be released next March as it won't be ready before Christmas - and it takes a while to fix the new issues reported during the Christmas break. (It is really a break as most developers are paid for GCC work (C, Ada, middle-end, target parts) and take off - only gfortran is purely developed in the spare time.).....gfortran 4.6 will presumably also allow to use REAL(16) (128 bit floating-point numbers) on x86, x86-64, and ia64 systems, which are emulated in software; so far only the real kinds 4, 8 and 10 (80bit FP) were supported on those systems. (This library inclusion had to be approved by the Free Software Foundation - but that problem seems to be mostly solved.)

For a complete list of what's new in 4.6 visit:

`http://gcc.gnu.org/gcc-4.6/changes.html`

Thanks Tobias.

Revision 7

The main driving force for the changes in this revision was an email from Stan Whitlock at Intel. Here is an extract from Stan's email.

I hope you will be updating the F2008 features list based on John Reid's updated article: ISO/N1828 - Features of F2008 - John Reid - latest adds features.

The feature list for Fortran 2008 used in the original article were taken from John Reid's earlier paper:

`ftp://ftp.nag.co.uk/sc22wg5/N1701-N1750/N1729.pdf`

The entries in the table are now taken from the contents of the N1828.pdf document. This is available at

`ftp://ftp.nag.co.uk/sc22wg5/N1801-N1850/N1828.pdf`

We have also added entries for two more Fortran compiler companies, Absoft (thanks Wood Lotz) and PGI (thanks Pat Brooks, Dave Norton and Brent Leback).

Revision 8

Corrections from Malcolm Cohen. Here is the email from Malcolm.

You are probably aware of these, but anyway...

In the F2008 table, the subheading "Input/Output" has a spurious "N" in the PGI column.

Near the end of the table, there are two entries

  "Null pointer or unallocated allocatable as an absent dummy argument"

and

"Null pointer as a missing dummy argument"; these would appear to be the same feature.

Also, the "generic resolution" line says "by pointer or allocatable attribute", it is missing any mention of "data object vs. procedure". (I think there is an argument for these two different extensions to generic resolution being mentioned separately.)

Thanks Malcolm, these have now been incorporated into the table.

Fortran 2003 Features	Absoft	Cray	g95	gfortran	HP	IBM	Intel	NAG	Oracle	PGI
Compiler version number				4.6			12.0	5.2		
ISO TR 15580 IEEE Arithmetic	Y	Y	P	N	Y	Y	Y	Y	Y	Y
ISO TR 15581 Allocatable Enhancements	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Data enhancements and object orientation										
Parameterized derived types	N	Y	N	N	N	Y	N	N	N	N
Procedure pointers	N	Y	Y	Y	Y	Y	Y	Y	N	Y
Finalization	N	Y	N	N	N	Y	Y	N	N	N
Procedures bound by name to a type	N	Y	N	Y	N	Y	Y	Y	N	Y
The PASS attribute	N	Y	N	Y	Y	Y	Y	Y	N	Y
Procedures bound to a type as operators	N	Y	N	Y	N	Y	Y	Y	N	Y
Type extension	N	Y	N	Y	N	Y	Y	Y	N	Y
Overriding a type-bound procedure	N	Y	N	Y	N	Y	Y	Y	N	Y
Enumerations	N	Y	Y	Y	N	Y	Y	Y	N	Y
ASSOCIATE construct	N	Y	N	P	N	Y	Y	Y	N	Y
Polymorphic entities	N	Y	N	P, 1	N	Y	Y	Y	N	Y
SELECT TYPE construct	N	Y	N	P	N	Y	Y	Y	N	Y
Deferred bindings and abstract types	N	Y	N	Y	N	Y	Y	Y	N	Y
Allocatable scalars, 12		Y		Y	N		Y	Y		Y
Allocatable character length, 12		Y		P	N		Y	Y		Y
Miscellaneous enhancements										
Structure constructors	N	Y	Y	Y	N	Y	Y	N	N	Y
The allocate statement	N	Y	P	Y	N	Y	Y	Y	N	Y
Assignment to an allocatable array	N	Y, 2	N	P	Y	Y	Y, 2	Y	N	Y,2
Transferring an allocation	N	Y	N	Y	N	Y	Y	Y	N	Y
More control of access from a module	N	Y	N	Y	N	Y	Y	Y	Y	Y
Renaming operators on the USE statement	Y	Y	Y	P	N	Y	Y	Y	Y	Y
Pointer assignment	N	Y	Y	Y	N	Y	Y	Y	N	Y
Pointer INTENT	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
The VOLATILE attribute	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
The IMPORT statement	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Intrinsic modules	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
Access to the computing environment	N	Y	Y	Y	Y	Y	Y	Y	Y	Y

Fortran 2003 Features	Absoft	Cray	g95	gfortran	HP	IBM	Intel	NAG	Oracle	PGI
Compiler version number				4.6			12.0	5.2		
Support for international character sets	N	P, 19	Y	Y	N	P	P, 19	P	N	N
Lengths of names and statements	N	Y	?	Y	Y	Y	Y	Y	Y	Y
Binary, octal and hex constants	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Array constructor syntax	N	Y	Y	Y	Y	Y	Y	Y	N	Y
Specification and initialization expressions	N	Y	Y	P	Y	Y	P	P	N	Y
Complex constants	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Changes to intrinsic functions	N	Y	Y	P, 9	Y	Y	Y	Y	N	Y
Controlling IEEE underflow	Y	Y	N	N	Y	Y	Y	P	Y	Y
Another IEEE class value	Y	Y	N	N	Y	Y	Y	Y	Y	Y
Input/output enhancements										
Derived type input/output	N	Y	N	N	N	Y	N	N	N	N
Asynchronous input/output	N	Y	Y	Y,10	N	Y	Y	Y	Y	Y
FLUSH statement	N	Y	Y	Y	N	Y	Y	Y	Y	Y
IOMSG= specifier	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
Stream access input/output	N	Y	Y	Y	N	Y	Y	Y	Y	Y
ROUND= specifier	Y	Y	P	P, 30	Y	Y	Y	Y	Y	Y,20, 30
DECIMAL= specifier	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y,21
SIGN= specifier	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y,22
Kind type parameters of integer specifiers	N	Y	?	N	N	Y	Y	Y	N	Y
Recursive input/output	N	Y	Y	Y	N	Y	Y	Y	Y	N
Intrinsic function for newline character	N	Y	Y	Y	Y	Y	Y	Y	N	Y
Input and output of IEEE exceptional values	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Comma after a P edit descriptor	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
Interoperability with C										Y
Interoperability of intrinsic types	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Interoperability with C pointers	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Interoperability of derived types	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Interoperability of variables	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Interoperability of procedures	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Interoperability of global data	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Notes

1	No unlimited polymorphic
2	Optional under flag
9	kind= of maxloc, minloc, shape missing
10	implemented as synchronous i/o
12	Suggested by Richard Maine
18	MOVE_ALLOC
19	SELECTED_CHAR_KIND only
20	plus RC,RD,RN,RP,RU,RZ
21	plus BLANK=,DELIM=,PAD=,SIZE=
22	plus DC,DP
30	only for output

Fortran 2008 Features	Absoft	Cray	g95	gfortran	HP	IBM	Intel	NAG	Oracle	PGI
Compiler version number				4.6			12.0	5.2		
Submodules	N	Y		N	N	N	N	N	N	N
Coarrays	N	Y	P	P, 200	N	N	Y	N	N	N
Performance enhancements										
do concurrent	N	N		N	N	N	Y	N	N	N
Contiguous attribute	N	Y		Y	N	N	Y	N	N	N
Simply contiguous arrays	N	Y		Y	N	N	Y	N	N	N
Data Declaration	N									N
Maximum rank	N	N		N	N	Y	Y	N	N	N
Long integers	Y, 100	Y		Y	N	Y	Y, 100	Y	Y	Y
Allocatable components of recursive type	N	N		N	N	N	P	N	N	N
Implied-shape array	N	N		Y	N	N	Y	N	N	N
Pointer initialization	N	N		P	N	N	Y	N	N	N
Data statement restrictions lifted							N	N		N
Kind of a forall index	N	N		N	N	N	Y	N	N	N
Type statement for intrinsic types							N	N		N
Declaring type-bound procedures							Y	N		N
Allocating a polymorphic variable	N	Y		P, 202	N	N	Y	N	N	
Accessing data objects										
Omitting an allocatable component in a structure component							Y			
Copying bounds of source array in allocate							Y			
Allocating a polymorphic variable							Y			
Accessing real and imaginary parts	N	N		N	N	N	N	N	N	N
Pointer functions	N	N		N	N	N	N	N	N	N
Elemental dummy argument restrictions lifted							N			
Input/Output										
Finding a unit when opening a file	N	N		Y	N	N	Y	N	N	N
g0 edit descriptor	N	Y		Y	N	N	Y	N	N	N
Unlimited format item	N	Y		Y	N	N	Y	N	N	N
Recursive input/output	N	Y		Y	N	N	Y	Y	Y	N
Execution control										
The block construct	N	Y		Y	N	N	N	N	N	N

Fortran 2008 Features	Absoft	Cray	g95	gfortran	HP	IBM	Intel	NAG	Oracle	PGI
Compiler version number				4.6			12.0	5.2		
Exit statement	N	N		Y	N	N	N	N	N	N
Stop code	N	Y		Y	N	N	Y	N	N	N
Intrinsic procedures and modules										
Bit processing										N
Bit sequence comparison	N	Y		Y	N	N	Y	N	N	N
Combined shifting	N	Y		Y	N	N	Y	N	N	N
Counting bits	N	Y		Y	N	Y	Y	N	N	Y
Masking bits	N	Y		Y	N	N	Y	N	N	N
Shifting bits	N	Y		Y	N	N	Y	N	N	N
Merging bits	N	N		Y	N	N	Y	N	N	N
Bit transformational functions	N	N		Y	N	N	Y	N	N	N
Intrinsic procedures and modules										
Storage size	N	N		N	N	N	Y	N	N	N
Taking radix into account when Selecting a real kind	N	Y		N	N	N	Y	N	N	N
Extensions to trigonometric and Hyperbolic intrinsic functions	?	N		Y	N	P	Y	N	N	Y
Bessel functions	N	N		Y	N	N	Y	N	N	N
Arc tangent function	?	Y		Y	N	N	Y	N	N	
Error and gamma functions	N	P		Y	N	P	Y	N	N	N
Euclidean vector norms	N	N		Y	N	N	Y	N	N	N
Parity	N	N		Y	N	N	Y	N	Y	N
Execute command line	N	N		Y	N	N	N	N	N	N
Location of max or min value in an array	N	Y		N	N	N	N	N	N	N
Find location in an array	?	Y		N	N	N	N	N	N	N
String comparison							Y	Y		N
Constants	N	Y		P, 203	N	N	Y	N	N	N
Module procedures	N	N		N	N	N	N	N	N	N
Programs and procedures										N
Save attribute for module and submodule data							Y	Y		N
Empty contains section	N	Y		Y	N	N	Y	N	N	N
Form of the end statement for an internal or module procedure							N	N		N
Internal procedure as an actual argument	N	N		Y	N	N	Y	N	N	N

Fortran 2008 Features	Absoft	Cray	g95	gfortran	HP	IBM	Intel	NAG	Oracle	PGI
Compiler version number				4.6			12.0	5.2		
Null pointer or unallocated allocatable as an absent dummy argument	N	N		Y	N	N	N	N	N	N
Pointer dummy arguments								N		N
Generic resolution by pointer or allocatable attribute - procedure	N	N		N	N	N	N	N	N	N
Generic resolution by pointer or allocatable attribute - data object	N	N		N	N	N	N	N	N	N
Elemental procedures that are not pure	N	N		Y	N	N	N	N	N	N
Entry statement becomes obsolescent	N	Y		Y	N	N	N	N	N	N
Source form										
Semicolon at line start							N	Y		Y

Notes

100   INTEGER (KIND=8)

200   Only for a single image

202   Only MOLD=

203   int and real, and coarray

Fortran 95 and TR 15581

The following companies also make Fortran compilers and we include details of their degree of standard support.

	Lahey	Salford ftn95 Silverfrost	Path Scale	AMD x86 open64
Windows				
Fortran 95	Yes	Yes		NA
TR 15581	Yes	No		NA
Linux				
Fortran 95	Yes	NA	Yes	Yes
TR 15581	Yes	NA	Yes	No
Others				
Apogee	No longer available			
Compaq	No longer available			
Fortran Company	F			
Fujitsu	No information at this time			
NA Software	No longer available			
NEC	No information at this time			