

ISO/IEC 1539-1:2004-11 (E)

Information technology - Programming languages - Fortran - Part 1: Base language

Contents		Page
1	Overview	1
1.1	Scope	1
1.2	Processor	1
1.3	Inclusions	1
1.4	Exclusions	1
1.5	Conformance	2
1.6	Compatibility	3
1.6.1	Fortran 95 compatibility	3
1.6.2	Fortran 90 compatibility	3
1.6.3	FORTTRAN 77 compatibility	4
1.7	Notation used in this standard	4
1.7.1	Informative notes	4
1.7.2	Syntax rules	4
1.7.3	Constraints	5
1.7.4	Assumed syntax rules	6
1.7.5	Syntax conventions and characteristics	6
1.7.6	Text conventions	6
1.8	Deleted and obsolescent features	7
1.8.1	Nature of deleted features	7
1.8.2	Nature of obsolescent features	7
1.9	Normative references	7
2	Fortran terms and concepts	9
2.1	High level syntax	9
2.2	Program unit concepts	11
2.2.1	Program	12
2.2.2	Main program	12
2.2.3	Procedure	12
2.2.4	Module	13
2.3	Execution concepts	13
2.3.1	Executable/nonexecutable statements	13
2.3.2	Statement order	13
2.3.3	The END statement	14
2.3.4	Execution sequence	15
2.4	Data concepts	15
2.4.1	Type	15
2.4.2	Data value	16
2.4.3	Data entity	16
2.4.4	Scalar	17
2.4.5	Array	18
2.4.6	Pointer	18
2.4.7	Storage	18
2.5	Fundamental terms	18
2.5.1	Name and designator	19
2.5.2	Keyword	19
2.5.3	Association	19
2.5.4	Declaration	19
2.5.5	Definition	19
2.5.6	Reference	20
2.5.7	Intrinsic	20
2.5.8	Operator	20
2.5.9	Sequence	20

2.5.10	Companion processors	21
3	Characters, lexical tokens, and source form	23
3.1	Processor character set	23
3.1.1	Letters	23
3.1.2	Digits	24
3.1.3	Underscore	24
3.1.4	Special characters	24
3.1.5	Other characters	24
3.2	Low-level syntax	25
3.2.1	Names	25
3.2.2	Constants	25
3.2.3	Operators	25
3.2.4	Statement labels	26
3.2.5	Delimiters	27
3.3	Source form	27
3.3.1	Free source form	27
3.3.2	Fixed source form	29
3.4	Including source text	30
4	Types	33
4.1	The concept of type	33
4.1.1	Set of values	33
4.1.2	Constants	33
4.1.3	Operations	34
4.2	Type parameters	34
4.3	Relationship of types and values to objects	35
4.4	Intrinsic types	35
4.4.1	Integer type	36
4.4.2	Real type	37
4.4.3	Complex type	39
4.4.4	Character type	40
4.4.5	Logical type	43
4.5	Derived types	44
4.5.1	Derived-type definition	45
4.5.2	Derived-type parameters	48
4.5.3	Components	49
4.5.4	Type-bound procedures	56
4.5.5	Final subroutines	58
4.5.6	Type extension	60
4.5.7	Derived-type values	62
4.5.8	Derived-type specifier	62
4.5.9	Construction of derived-type values	63
4.5.10	Derived-type operations and assignment	65
4.6	Enumerations and enumerators	66
4.7	Construction of array values	67
5	Data object declarations and specifications	71
5.1	Type declaration statements	71
5.1.1	Declaration type specifiers	75
5.1.2	Attributes	76
5.2	Attribute specification statements	85
5.2.1	Accessibility statements	86
5.2.2	ALLOCATABLE statement	86
5.2.3	ASYNCHRONOUS statement	86
5.2.4	BIND statement	87
5.2.5	DATA statement	87
5.2.6	DIMENSION statement	90
5.2.7	INTENT statement	90
5.2.8	OPTIONAL statement	90
5.2.9	PARAMETER statement	90
5.2.10	POINTER statement	91

5.2.11	PROTECTED statement	91
5.2.12	SAVE statement	91
5.2.13	TARGET statement	92
5.2.14	VALUE statement	92
5.2.15	VOLATILE statement	92
5.3	IMPLICIT statement	92
5.4	NAMELIST statement	95
5.5	Storage association of data objects	95
5.5.1	EQUIVALENCE statement	95
5.5.2	COMMON statement	98
6	Use of data objects	103
6.1	Scalars	104
6.1.1	Substrings	104
6.1.2	Structure components	104
6.1.3	Type parameter inquiry	106
6.2	Arrays	106
6.2.1	Whole arrays	107
6.2.2	Array elements and array sections	107
6.3	Dynamic association	110
6.3.1	ALLOCATE statement	110
6.3.2	NULLIFY statement	113
6.3.3	DEALLOCATE statement	114
7	Expressions and assignment	117
7.1	Expressions	117
7.1.1	Form of an expression	117
7.1.2	Intrinsic operations	121
7.1.3	Defined operations	122
7.1.4	Type, type parameters, and shape of an expression	123
7.1.5	Conformability rules for elemental operations	125
7.1.6	Specification expression	125
7.1.7	Initialization expression	126
7.1.8	Evaluation of operations	128
7.2	Interpretation of operations	132
7.2.1	Numeric intrinsic operations	133
7.2.2	Character intrinsic operation	133
7.2.3	Relational intrinsic operations	134
7.2.4	Logical intrinsic operations	135
7.3	Precedence of operators	136
7.4	Assignment	138
7.4.1	Assignment statement	138
7.4.2	Pointer assignment	142
7.4.3	Masked array assignment - WHERE	145
7.4.4	FORALL	148
8	Execution control	155
8.1	Executable constructs containing blocks	155
8.1.1	Rules governing blocks	155
8.1.2	IF construct	156
8.1.3	CASE construct	158
8.1.4	ASSOCIATE construct	160
8.1.5	SELECT TYPE construct	162
8.1.6	DO construct	164
8.2	Branching	169
8.2.1	GO TO statement	169
8.2.2	Computed GO TO statement	170
8.2.3	Arithmetic IF statement	170
8.3	CONTINUE statement	170
8.4	STOP statement	170
9	Input/output statements	171

9.1	Records	171
9.1.1	Formatted record	171
9.1.2	Unformatted record	172
9.1.3	Endfile record	172
9.2	External files	172
9.2.1	File existence	173
9.2.2	File access	173
9.2.3	File position	175
9.2.4	File storage units	177
9.3	Internal files	177
9.4	File connection	178
9.4.1	Connection modes	179
9.4.2	Unit existence	179
9.4.3	Connection of a file to a unit	179
9.4.4	Preconnection	180
9.4.5	The OPEN statement	180
9.4.6	The CLOSE statement	184
9.5	Data transfer statements	186
9.5.1	Control information list	186
9.5.2	Data transfer input/output list	191
9.5.3	Execution of a data transfer input/output statement	193
9.5.4	Termination of data transfer statements	205
9.6	Waiting on pending data transfer	205
9.6.1	WAIT statement	205
9.6.2	Wait operation	206
9.7	File positioning statements	207
9.7.1	BACKSPACE statement	207
9.7.2	ENDFILE statement	208
9.7.3	REWIND statement	208
9.8	FLUSH statement	208
9.9	File inquiry	209
9.9.1	Inquiry specifiers	210
9.9.2	Restrictions on inquiry specifiers	216
9.9.3	Inquire by output list	216
9.10	Error, end-of-record, and end-of-file conditions	216
9.10.1	Error conditions and the ERR= specifier	217
9.10.2	End-of-file condition and the END= specifier	217
9.10.3	End-of-record condition and the EOR= specifier	217
9.10.4	IOSTAT= specifier	218
9.10.5	IOMSG= specifier	218
9.11	Restrictions on input/output statements	219
10	Input/output editing	221
10.1	Explicit format specification methods	221
10.1.1	FORMAT statement	221
10.1.2	Character format specification	221
10.2	Form of a format item list	222
10.2.1	Edit descriptors	222
10.2.2	Fields	224
10.3	Interaction between input/output list and format	224
10.4	Positioning by format control	225
10.5	Decimal symbol	226
10.6	Data edit descriptors	226
10.6.1	Numeric editing	226
10.6.2	Logical editing	232
10.6.3	Character editing	232
10.6.4	Generalized editing	233
10.6.5	User-defined derived-type editing	235
10.7	Control edit descriptors	235
10.7.1	Position editing	235
10.7.2	Slash editing	236
10.7.3	Colon editing	236

10.7.4	SS, SP, and S editing	237
10.7.5	P editing	237
10.7.6	BN and BZ editing	237
10.7.7	RU, RD, RZ, RN, RC, and RP editing	238
10.7.8	DC and DP editing	238
10.8	Character string edit descriptors	238
10.9	List-directed formatting	238
10.9.1	List-directed input	239
10.9.2	List-directed output	241
10.10	Namelist formatting	242
10.10.1	Namelist input	243
10.10.2	Namelist output	246
11	Program units	249
11.1	Main program	249
11.2	Modules	250
11.2.1	The USE statement and use association	251
11.3	Block data program units	253
12	Procedures	255
12.1	Procedure classifications	255
12.1.1	Procedure classification by reference	255
12.1.2	Procedure classification by means of definition	255
12.2	Characteristics of procedures	256
12.2.1	Characteristics of dummy arguments	256
12.2.2	Characteristics of function results	257
12.3	Procedure interface	257
12.3.1	Implicit and explicit interfaces	257
12.3.2	Specification of the procedure interface	258
12.4	Procedure reference	266
12.4.1	Actual arguments, dummy arguments, and argument association	268
12.4.2	Function reference	276
12.4.3	Subroutine reference	276
12.4.4	Resolving named procedure references	276
12.4.5	Resolving type-bound procedure references	278
12.5	Procedure definition	279
12.5.1	Intrinsic procedure definition	279
12.5.2	Procedures defined by subprograms	279
12.5.3	Definition and invocation of procedures by means other than Fortran	285
12.5.4	Statement function	285
12.6	Pure procedures	286
12.7	Elemental procedures	287
12.7.1	Elemental procedure declaration and interface	287
12.7.2	Elemental function actual arguments and results	288
12.7.3	Elemental subroutine actual arguments	289
13	Intrinsic procedures and modules	291
13.1	Classes of intrinsic procedures	291
13.2	Arguments to intrinsic procedures	291
13.2.1	The shape of array arguments	292
13.2.2	Mask arguments	292
13.3	Bit model	292
13.4	Numeric models	293
13.5	Standard generic intrinsic procedures	294
13.5.1	Numeric functions	294
13.5.2	Mathematical functions	294
13.5.3	Character functions	295
13.5.4	Kind functions	295
13.5.5	Miscellaneous type conversion functions	295
13.5.6	Numeric inquiry functions	295
13.5.7	Array inquiry functions	296
13.5.8	Other inquiry functions	296

13.5.9	Bit manipulation procedures	296
13.5.10	Floating-point manipulation functions	296
13.5.11	Vector and matrix multiply functions	297
13.5.12	Array reduction functions	297
13.5.13	Array construction functions	297
13.5.14	Array location functions	297
13.5.15	Null function	297
13.5.16	Allocation transfer procedure	297
13.5.17	Random number subroutines	297
13.5.18	System environment procedures	298
13.6	Specific names for standard intrinsic functions	298
13.7	Specifications of the standard intrinsic procedures	300
13.8	Standard intrinsic modules	359
13.8.1	The IEEE modules	359
13.8.2	The ISO FORTRAN ENV intrinsic module	359
13.8.3	The ISO C BINDING module	360
14	Exceptions and IEEE arithmetic	361
14.1	Derived types and constants defined in the modules	362
14.2	The exceptions	363
14.3	The rounding modes	365
14.4	Underflow mode	365
14.5	Halting	366
14.6	The floating point status	366
14.7	Exceptional values	366
14.8	IEEE arithmetic	366
14.9	Tables of the procedures	367
14.9.1	Inquiry functions	367
14.9.2	Elemental functions	368
14.9.3	Kind function	368
14.9.4	Elemental subroutines	368
14.9.5	Nonelemental subroutines	368
14.10	Specifications of the procedures	369
14.11	Examples	384
15	Interoperability with C	389
15.1	The ISO C BINDING intrinsic module	389
15.1.1	Named constants and derived types in the module	389
15.1.2	Procedures in the module	390
15.2	Interoperability between Fortran and C entities	393
15.2.1	Interoperability of intrinsic types	394
15.2.2	Interoperability with C pointer types	395
15.2.3	Interoperability of derived types and C struct types	396
15.2.4	Interoperability of scalar variables	397
15.2.5	Interoperability of array variables	397
15.2.6	Interoperability of procedures and procedure interfaces	398
15.3	Interoperation with C global variables	400
15.3.1	Binding labels for common blocks and variables	401
15.4	Interoperation with C functions	401
15.4.1	Binding labels for procedures	401
15.4.2	Exceptions and IEEE arithmetic procedures	402
16	Scope, association, and definition	403
16.1	Scope of global identifiers	403
16.2	Scope of local identifiers	404
16.2.1	Local identifiers that are the same as common block names	405
16.2.2	Function results	405
16.2.3	Restrictions on generic declarations	405
16.2.4	Components, type parameters, and bindings	406
16.2.5	Argument keywords	407
16.3	Statement and construct entities	407
16.4	Association	408

16.4.1	Name association	408
16.4.2	Pointer association	412
16.4.3	Storage association	413
16.4.4	Inheritance association	416
16.4.5	Establishing associations	416
16.5	Definition and undefinition of variables	417
16.5.1	Definition of objects and subobjects	417
16.5.2	Variables that are always defined	417
16.5.3	Variables that are initially defined	417
16.5.4	Variables that are initially undefined	418
16.5.5	Events that cause variables to become defined	418
16.5.6	Events that cause variables to become undefined	419
16.5.7	Variable definition context	421
Annex A (informative)Glossary of technical terms		423
Annex B (informative)Decremental features		435
B.1	Deleted features	435
B.2	Obsolescent features	436
B.2.1	Alternate return	436
B.2.2	Computed GO TO statement	436
B.2.3	Statement functions	436
B.2.4	DATA statements among executables	437
B.2.5	Assumed character length functions	437
B.2.6	Fixed form source	437
B.2.7	CHARACTER* form of CHARACTER declaration	437
Annex C (informative)Extended notes		439
C.1	Section 4 notes	439
C.1.1	Intrinsic and derived types (4.4, 4.5)	439
C.1.2	Selection of the approximation methods (4.4.2)	440
C.1.3	Type extension and component accessibility (4.5.1.1, 4.5.3)	440
C.1.4	Abstract types	441
C.1.5	Pointers (4.5.1)	442
C.1.6	Structure constructors and generic names	443
C.1.7	Generic type-bound procedures	445
C.1.8	Final subroutines (4.5.5, 4.5.5.1, 4.5.5.2, 4.5.5.3)	446
C.2	Section 5 notes	448
C.2.1	The POINTER attribute (5.1.2.11)	448
C.2.2	The TARGET attribute (5.1.2.14)	449
C.2.3	The VOLATILE attribute (5.1.2.16)	449
C.3	Section 6 notes	450
C.3.1	Structure components (6.1.2)	450
C.3.2	Allocation with dynamic type (6.3.1)	451
C.3.3	Pointer allocation and association	452
C.4	Section 7 notes	453
C.4.1	Character assignment	453
C.4.2	Evaluation of function references	453
C.4.3	Pointers in expressions	453
C.4.4	Pointers on the left side of an assignment	453
C.4.5	An example of a FORALL construct containing a WHERE construct	454
C.4.6	Examples of FORALL statements	455
C.5	Section 8 notes	456
C.5.1	Loop control	456
C.5.2	The CASE construct	456
C.5.3	Examples of DO constructs	456
C.5.4	Examples of invalid DO constructs	458
C.6	Section 9 notes	459
C.6.1	External files (9.2)	459
C.6.2	Nonadvancing input/output (9.2.3.1)	461

C.6.3	Asynchronous input/output	462
C.6.4	OPEN statement (9.4.5)	463
C.6.5	Connection properties (9.4.3)	464
C.6.6	CLOSE statement (9.4.6)	465
C.7	Section 10 notes	465
C.7.1	Number of records (10.3, 10.4, 10.7.2)	465
C.7.2	List-directed input (10.9.1)	466
C.8	Section 11 notes	466
C.8.1	Main program and block data program unit (11.1, 11.3)	466
C.8.2	Dependent compilation (11.2)	467
C.8.3	Examples of the use of modules	469
C.9	Section 12 notes	475
C.9.1	Portability problems with external procedures (12.3.2.2)	475
C.9.2	Procedures defined by means other than Fortran (12.5.3)	476
C.9.3	Procedure interfaces (12.3)	476
C.9.4	Abstract interfaces (12.3) and procedure pointer components (4.5)	476
C.9.5	Argument association and evaluation (12.4.1.2)	478
C.9.6	Pointers and targets as arguments (12.4.1.2)	479
C.9.7	Polymorphic Argument Association (12.4.1.3)	480
C.10	Section 15 notes	482
C.10.1	Runtime environments	482
C.10.2	Examples of Interoperation between Fortran and C Functions	482
C.11	Section 16 notes	487
C.11.1	Examples of host association (16.4.1.3)	487
C.11.2	Rules ensuring unambiguous generics (16.2.3)	488
C.12	Array feature notes	493
C.12.1	Summary of features	493
C.12.2	Examples	495
C.12.3	FORmula TRANslation and array processing	499
C.12.4	Sum of squared residuals	500
C.12.5	Vector norms: infinity-norm and one-norm	500
C.12.6	Matrix norms: infinity-norm and one-norm	500
C.12.7	Logical queries	500
C.12.8	Parallel computations	501
C.12.9	Example of element-by-element computation	501
C.12.10	Bit manipulation and inquiry procedures	502
Annex D (informative)Syntax rules		503
D.1	Extract of all syntax rules	503
D.2	Syntax rule cross-reference	541
Annex E (informative)Index		553
List of Tables 2.1 Requirements on statement ordering		14
2.2	Statements allowed in scoping units	14
3.1	Special characters	24
6.1	Subscript order value	108
7.1	Type of operands and results for intrinsic operators	121
7.2	Interpretation of the numeric intrinsic operators	133
7.3	Interpretation of the character intrinsic operator //	134
7.4	Interpretation of the relational intrinsic operators	134
7.5	Interpretation of the logical intrinsic operators	135
7.6	The values of operations involving logical intrinsic operators	135
7.7	Categories of operations and relative precedence	136
7.8	Type conformance for the intrinsic assignment statement	138
7.9	Numeric conversion and the assignment statement	141
13.1	Characteristics of the result of NULL ()	341
15.1	Names of C characters with special semantics	390
15.2	Interoperability between Fortran and C types	394