



DRAFT INTERNATIONAL STANDARD ISO/IEC 26926

Attributed to ISO/IEC JTC 1 by the Central Secretariat (see page iii)

Voting begins on
2006-03-09

Voting terminates on
2006-08-09

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОММИСИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

FAST-TRACK PROCEDURE

C++/CLI Language Specification

Specification du langage C++/CLI

ICS 35.060

iTeh STANDARD PREVIEW (standards.iteh.ai)

ISO/IEC FCD 26926

<https://standards.iteh.ai/catalog/standards/sist/f763c1e4-cb7b-446a-a681-438dbdeadafc/iso-iec-fcd-26926>

In accordance with the provisions of Council Resolution 21/1986 this DIS is circulated in the English language only.

Conformément aux dispositions de la Résolution du Conseil 21/1986, ce DIS est distribué en version anglaise seulement.

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC FCD 26926

<https://standards.iteh.ai/catalog/standards/sist/f763c1e4-cb7b-446a-a681-438dbdeadafc/iso-iec-fcd-26926>

Copyright notice

This ISO document is a Draft International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

NOTE FROM ITTF

This draft International Standard is submitted for JTC 1 national body vote under the Fast-Track Procedure.

In accordance with Resolution 30 of the JTC 1 Berlin Plenary 1993, the proposer of this document recommends assignment of ISO/IEC 26926 to JTC 1/SC 22.

“FAST-TRACK” PROCEDURE

1 Any P-member and any Category A liaison organization of ISO/IEC JTC 1 may propose that an existing standard from any source be submitted directly for vote as a DIS. The criteria for proposing an existing standard for the fast-track procedure are a matter for each proposer to decide.

2 The proposal shall be received by the ITTF which will take the following actions.

2.1 To settle the copyright and/or trade mark situation with the proposer, so that the proposed text can be freely copied and distributed within JTC 1 without restriction.

2.2 To assess in consultation with the JTC 1 secretariat which SC is competent for the subject covered by the proposed standard and to ascertain that there is no evident contradiction with other International Standards.

2.3 To distribute the text of the proposed standard as a DIS. In case of particularly bulky documents the ITTF may demand the necessary number of copies from the proposer.

3 The period for combined DIS voting shall be six months. In order to be accepted the DIS must be supported by 75 % of the votes cast (abstention is not counted as a vote) and by two-thirds of the P-members voting of JTC 1.

4 At the end of the voting period, the comments received, whether editorial only or technical, will be dealt with by a working group appointed by the secretariat of the relevant SC.

5 If, after the deliberations of this WG, the requirements of 3 above are met, the amended text shall be sent to the ITTF by the secretariat of the relevant SC for publication as an International Standard.

If it is impossible to agree to a text meeting the above requirements, the proposal has failed and the procedure is terminated.

In either case the WG shall prepare a full report which will be circulated by the ITTF.

6 If the proposed standard is accepted and published, its maintenance will be handled by JTC 1.

iTeh STANDARD PREVIEW **(standards.iteh.ai)**

ISO/IEC FCD 26926

<https://standards.iteh.ai/catalog/standards/sist/f763c1e4-cb7b-446a-a681-438dbdeadafc/iso-iec-fcd-26926>

Standard ECMA-372

1st Edition / December 2005

C++ / CLI Language Specification

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC FCD 26926

<https://standards.iteh.ai/catalog/standards/sist/f763c1e4-cb7b-446a-a681-438dbdeadafc/iso-iec-fcd-26926>

Standard

iTeh STANDARD PREVIEW **(standards.iteh.ai)**

ISO/IEC FCD 26926

<https://standards.iteh.ai/catalog/standards/sist/f763c1e4-cb7b-446a-a681-438dbdeadafc/iso-iec-fcd-26926>

Standard ECMA-372

1st Edition / December 2005

iTeh STANDARD PREVIEW
**C++ / CLI Language
Specification**
(standards.iteh.ai)

ISO/IEC FCD 26926

<https://standards.iteh.ai/catalog/standards/sist/f763c1e4-cb7b-446a-a681-438dbdeadafc/iso-iec-fcd-26926>

iTeh STANDARD PREVIEW **(standards.iteh.ai)**

ISO/IEC FCD 26926

<https://standards.iteh.ai/catalog/standards/sist/f763c1e4-cb7b-446a-a681-438dbdeadafc/iso-iec-fcd-26926>

Table of Contents

Introduction	xii
1. Scope.....	1
2. Conformance	2
3. Normative references	3
4. Definitions	4
5. Notational conventions.....	7
6. Acronyms and abbreviations	8
7. General description	9
8. Language overview.....	10
8.1 Getting started	10
8.2 Types	10
8.2.1 Fundamental types and the CLI	12
8.2.2 Conversions	13
8.2.3 CLI array types	13
8.2.4 Type system unification.....	13
8.2.5 Pointers, handles, and null	14
8.3 Parameters	16
8.4 Automatic memory management.....	17
8.5 Expressions.....	18
8.6 Statements.....	19
8.7 Delegates	19
8.8 Native and ref classes.....	20
8.8.1 Literal fields	20
8.8.2 Initonly fields.....	21
8.8.3 Functions.....	21
8.8.4 Properties	21
8.8.5 Events.....	23
8.8.6 Static operators	24
8.8.7 Instance constructors.....	25
8.8.8 Destructors and finalizers	25
8.8.9 Static constructors.....	26
8.8.10 Inheritance	27
8.9 Value classes	28
8.10 Interfaces	28
8.11 Enums	30
8.12 Namespaces and assemblies	30
8.13 Versioning	31
8.14 Attributes	32
8.15 Generics.....	33
8.15.1 Creating and consuming generics	33
8.15.2 Constraints	34
8.15.3 Generic functions.....	35
9. Lexical structure.....	37
9.1 Tokens	37
9.1.1 Identifiers	37
9.1.2 Keywords.....	38

9.1.3 Literals	39
9.1.4 Operators and punctuators	40
10. Basic concepts	41
10.1 Assemblies.....	41
10.2 Application entry point.....	41
10.3 Importing types from assemblies.....	41
10.4 Reserved names	42
10.5 Members	43
10.5.1 Value class members.....	43
10.5.2 Delegate members.....	43
10.6 Member access	43
10.6.1 Declared accessibility	43
10.7 Name lookup	44
11. Preprocessor	48
11.1 Conditional inclusion.....	48
11.2 Predefined macro names.....	48
12. Types	49
12.1 Value types	50
12.1.1 Fundamental types	50
12.2 Class types.....	51
12.2.1 Value classes.....	51
12.2.2 Ref classes.....	51
12.2.3 Interface classes.....	51
12.2.4 Delegate types.....	51
12.3 Declarator types.....	52
12.3.1 Raw types.....	52
12.3.2 Pointer types.....	52
12.3.3 Handle types.....	52
12.3.4 Null type	53
12.3.5 Reference types.....	53
12.3.6 Interior pointers.....	54
12.3.7 Pinning pointers	55
12.3.8 Native arrays	57
12.4 Top-level type visibility	57
13. Variables	58
13.1 gc-lvalues.....	58
13.1.1 Standard conversions	58
13.1.2 Expressions	59
13.1.3 Reference initializers	60
13.1.4 Temporary objects	60
13.2 File-scope and namespace-scope variables	60
13.3 Direct initialization.....	60
14. Conversions.....	62
14.1 Conversion sequences	62
14.2 Standard conversions.....	62
14.2.1 Handle conversions.....	62
14.2.2 Pointer conversions.....	63
14.2.3 Lvalue conversions	64
14.2.4 Integral promotions.....	64
14.2.5 String literal conversions	65
14.2.6 Boxing conversions.....	66

14.3 Implicit conversions	66
14.3.1 Implicit constant expression conversions	66
14.3.2 User-defined implicit conversions	66
14.3.3 Boolean Equivalence	66
14.4 Explicit conversions	67
14.5 User-defined conversions	67
14.5.1 Constructors	67
14.5.2 Explicit conversion functions	67
14.5.3 Static conversion functions	67
14.6 Parameter array conversions	67
14.7 Naming conventions	68
15. Expressions	70
15.1 Function members	70
15.2 Primary expressions	71
15.3 Postfix expressions	71
15.3.1 Subscripting and indexed access	72
15.3.2 Function call	72
15.3.3 Explicit type conversion (functional notation)	72
15.3.4 Class member access	73
15.3.5 Increment and decrement	73
15.3.6 Dynamic cast	73
15.3.7 Type identification	74
15.3.8 Static cast	75
15.3.9 Reinterpret cast	76
15.3.10 Const cast	76
15.3.11 Safe cast	76
15.4 Unary expressions	77
15.4.1 Unary operators	77
15.4.2 Increment and decrement	79
15.4.3 Sizeof	80
15.4.4 New	80
15.4.5 Delete	80
15.4.6 The genew operator	81
15.4.7 The throw expression	81
15.5 Explicit type conversion (cast notation)	81
15.6 Additive operators	82
15.6.1 Delegate combination	82
15.6.2 Delegate removal	82
15.6.3 String concatenation	82
15.7 Shift operators	83
15.8 Relational operators	83
15.8.1 Handle equality operators	83
15.8.2 Delegate equality operators	84
15.8.3 String equality	85
15.9 Logical AND operator	85
15.10 Logical OR operator	85
15.11 Conditional operator	85
15.12 Assignment operators	85
15.13 Constant expressions	86
15.14 Property and event rewrite rules	86
16. Statements	89
16.1 Selection statements	89
16.1.1 The switch statement	89

16.2 Iteration statements.....	89
16.2.1 The for each statement.....	89
16.3 Jump statements.....	91
16.3.1 The break statement.....	91
16.3.2 The continue statement.....	91
16.3.3 The return statement.....	91
16.3.4 The goto statement.....	91
16.4 The try block.....	91
17. Namespaces.....	93
17.1 Reserved namespaces.....	93
18. Functions.....	94
18.1 <cstdlib>-style variable-argument lists.....	94
18.2 Name lookup.....	94
18.3 Overload resolution.....	94
18.4 Parameter arrays.....	94
18.5 Importing native functions.....	96
18.6 Non-member functions.....	97
18.7 Attributes.....	97
19. Classes and members.....	98
19.1 Class definitions.....	98
19.1.1 Class modifiers.....	99
19.2 Reserved member names.....	100
19.2.1 Member names reserved for properties.....	100
19.2.2 Member names reserved for events.....	101
19.2.3 Member names reserved for functions.....	101
19.2.4 Possible collision with reserved property and event names.....	102
19.3 Data members.....	103
19.4 Functions.....	103
19.4.1 Override functions.....	104
19.4.2 Sealed function modifier.....	107
19.4.3 Abstract function modifier.....	107
19.4.4 New function modifier.....	108
19.5 Properties.....	109
19.5.1 Qualified names of properties and events.....	110
19.5.2 Static and instance properties.....	111
19.5.3 Accessor functions.....	111
19.5.4 Virtual, sealed, abstract, and override accessor functions.....	113
19.5.5 Trivial scalar properties.....	114
19.6 Events.....	115
19.6.1 Static and instance events.....	116
19.6.2 Accessor functions.....	116
19.6.3 Virtual, sealed, abstract, and override accessor functions.....	117
19.6.4 Trivial events.....	117
19.6.5 Event invocation.....	117
19.7 Static operators.....	117
19.7.1 Homogenizing the candidate overload set.....	119
19.7.2 Operators on handles.....	119
19.7.3 Increment and decrement operators.....	120
19.7.4 Operator synthesis.....	123
19.7.5 Naming conventions.....	123
19.8 Non-static operators.....	126
19.9 Instance constructors.....	126
19.10 Static constructors.....	127

19.11 Literal fields.....	128
19.12 Initonly fields.....	129
19.12.1 Using static initonly fields for constants.....	130
19.12.2 Versioning of literal fields and static initonly fields.....	130
19.13 Destructors and finalizers	130
19.13.1 Destructors	131
19.13.2 Finalizers.....	131
20. Native classes	133
20.1 Functions	133
20.2 Properties.....	133
20.3 Static operators	133
20.4 Delegates	133
20.5 Friends	133
20.6 Events	134
20.7 Finalizer.....	134
20.8 Initonly and literal fields.....	134
20.9 Static constructors	134
21. Ref classes	135
21.1 Ref class definitions	135
21.1.1 Ref class base specification	135
21.2 Ref class members	135
21.2.1 Variable initializers.....	135
21.3 Functions	136
21.4 Properties.....	136
21.5 Events	136
21.6 Static operators	137
21.7 Non-static operators.....	137
21.8 Instance constructors.....	137
21.9 Static constructor	137
21.10 Literal fields.....	137
21.11 Initonly fields.....	137
21.12 Destructors and finalizers	137
21.13 Delegates	137
22. Value classes	138
22.1 Value class definitions.....	138
22.1.1 Value class base specification.....	138
22.2 Value class members	138
22.3 Ref class and value class differences.....	139
22.3.1 Inheritance	139
22.3.2 Default values	139
22.3.3 Meaning of this	139
22.3.4 Destructors and finalizers	139
22.4 Simple value classes	140
22.5 Constructors.....	140
22.6 Operators	140
23. Mixed types	141
24. CLI arrays.....	142
24.1 CLI array types	142
24.1.1 The System::Array type	142
24.2 CLI array creation.....	143
24.3 CLI array element access.....	143

ITeH STANDARD PREVIEW
(standards.itech.ai)

ISO/IEC FCD 26926

[https://standards.itech.ai/catalog/standards/sist/1763e1e4-cb7b-446a-a681-](https://standards.itech.ai/catalog/standards/sist/1763e1e4-cb7b-446a-a681-438dbdeada1f/iso-iec-fcd-26926)

[438dbdeada1f/iso-iec-fcd-26926](https://standards.itech.ai/catalog/standards/sist/1763e1e4-cb7b-446a-a681-438dbdeada1f/iso-iec-fcd-26926)

24.4 CLI array members.....	144
24.5 CLI array covariance	144
24.6 CLI array initializers.....	144
25. Interfaces.....	146
25.1 Interface definitions.....	146
25.1.1 Interface base specification.....	146
25.2 Interface members	146
25.2.1 Functions.....	147
25.2.2 Properties	147
25.2.3 Events.....	147
25.2.4 Delegates.....	148
25.2.5 Member access.....	148
25.2.6 Destructors and finalizers	148
25.3 Interface implementations	148
26. Enums.....	150
26.1 Enum definitions	150
26.1.1 Enum base specification	151
26.1.2 Initial enumerator values.....	151
26.1.3 CLI enum values and operations.....	151
26.2 The System::Flags attribute.....	151
27. Delegates.....	153
27.1 Delegate definitions.....	153
27.2 Delegate instantiation.....	155
27.3 Delegate invocation.....	156
28. Exceptions and exception handling	157
28.1 Common exception classes.....	157
28.2 Exception specifications.....	158
29. Attributes	159
29.1 Attribute classes.....	159
29.1.1 Attribute usage.....	159
29.1.2 Positional and named parameters.....	160
29.1.3 Attribute parameter types.....	161
29.2 Attribute specification	161
29.3 Attribute instances	165
29.3.1 Compilation of an attribute	165
29.3.2 Run-time retrieval of an attribute instance.....	166
29.4 Reserved attributes	166
29.4.1 The AttributeUsage attribute.....	166
29.4.2 The Obsolete attribute.....	166
29.4.3 The Conditional attribute	167
29.4.4 Security attributes	167
29.5 Attributes for interoperation	167
29.5.1 Interoperation with other CLI-based languages.....	167
29.5.2 Interoperation with native code	167
30. Templates.....	168
30.1 Template declarations.....	168
30.2 Template specialization	168
30.3 Attributes	168
30.4 Type deduction	169
30.4.1 Template argument deduction.....	169

31. Generics.....	170
31.1 Generic declarations	170
31.1.1 Type parameters.....	171
31.1.2 Referencing a generic type by name	172
31.1.3 The instance type	172
31.1.4 Base classes and interfaces	173
31.1.5 Class members	173
31.1.6 Static members.....	174
31.1.7 Operators.....	175
31.1.8 Member overloading	175
31.1.9 Member overriding	176
31.1.10 Nested types.....	176
31.2 Constructed types	177
31.2.1 Open and closed constructed types	178
31.2.2 Type arguments.....	178
31.2.3 Base classes and interfaces	179
31.2.4 Class members	179
31.2.5 Accessibility.....	180
31.3 Generic functions.....	180
31.3.1 Function signature matching rules	181
31.3.2 Type deduction	182
31.4 Constraints.....	184
31.4.1 Satisfying constraints	185
31.4.2 Member lookup on type parameters.....	187
31.4.3 Type parameters and boxing	188
31.4.4 Conversions involving type parameters	189
32. Standard C and C++ libraries.....	190
33. CLI libraries	191
33.1 Custom modifiers	191
33.1.1 Signature matching	191
33.1.2 modreq vs. modopt.....	192
33.1.3 Modifier syntax.....	192
33.1.4 Types having multiple custom modifiers	193
33.1.5 Standard custom modifiers	194
33.2 Standard attributes	199
33.2.1 NativeCppClass	199
34. Metadata	200
34.1 Basic concepts	200
34.1.1 Importing types from assemblies	200
34.2 Types	200
34.2.1 Reference types.....	200
34.2.2 Interior pointers.....	201
34.2.3 Pinning pointers	201
34.2.4 Native arrays	202
34.3 Variables.....	202
34.3.1 File-scope and namespace-scope variables.....	202
34.4 Conversions	202
34.4.1 String literal conversions	202
34.4.2 Boxing conversions.....	202
34.4.3 Conversion functions	203
34.5 Expressions.....	203
34.5.1 Class member access.....	203