
Information technology — C++ extensions for reflection

Technologies de l'information — Extensions C++ pour la réflexion

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC TS 23619:2021](https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021)

<https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021>



Reference number
ISO/IEC TS 23619:2021(E)

© ISO/IEC 2021

iTeh STANDARD PREVIEW (standards.iteh.ai)

ISO/IEC TS 23619:2021

<https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2021

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Foreword	v
1 Scope	1
2 Normative references	2
3 Terms and definitions	3
4 General	4
4.1 Implementation compliance	4
4.2 Namespaces and headers	4
4.3 Feature-testing recommendations	4
5 Lexical conventions	5
5.12 Keywords	5
6 Basic concepts	6
6.2 One definition rule	6
6.9 Types	6
7 Standard conversions	7
8 Expressions	8
8.1 Primary expressions	8
8.2 Postfix expressions	8
9 Statements	9
10 Declarations	10
10.1 Specifiers	10
11 Declarators	14
11.1 Type names	14
12 Classes	15
13 Derived classes	16
14 Member access control	17
15 Special member functions	18
16 Overloading	19
17 Templates	20
17.6 Name resolution	20
18 Exception handling	21
19 Preprocessing directives	22

20 Library introduction	23
20.6 Library-wide requirements	23
21 Language support library	24
21.11 Static reflection	24
A Compatibility	45
A.1 C++ extensions for Concepts with Reflection and ISO/IEC 14882:2020	45
Index	46

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC TS 23619:2021](https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021)
<https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021>

Foreword

[foreword]

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

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

ISO/IEC TS 23619:2021

<https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021>

1 Scope

[scope]

- ¹ This document describes extensions to the C++ Programming Language (Clause 2) that enable operations on source code. These extensions include new syntactic forms and modifications to existing language semantics, as well as changes and additions to the existing library facilities.
- ² Instructions to modify or add paragraphs are written as explicit instructions. Modifications made directly to existing text from ISO/IEC 14882:2020 use underlining to represent added text and ~~strikethrough~~ to represent deleted text.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC TS 23619:2021
<https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021>

2 Normative references

[refs]

- ¹ The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- (1.1) — ISO/IEC 14882:2020, *Programming Languages — C++*
- (1.2) — ISO/IEC TS 19217:2015, *C++ Extensions for Concepts*

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC TS 23619:2021
<https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021>

3 Terms and definitions [defs]

¹ No terms and definitions are listed in this document. ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- (1.1) — IEC Electropedia: available at <https://www.electropedia.org/>
- (1.2) — ISO Online browsing platform: available at <https://www.iso.org/obp>

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC TS 23619:2021](https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021)
<https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021>

4 General

[general]

4.1 Implementation compliance

[general.compliance]

- ¹ Conformance requirements for this document are those defined in ISO/IEC 14882:2020, 4.1. Similarly, all references to ISO/IEC 14882:2020 in the resulting document shall be taken as referring to the resulting document itself. [Note: Conformance is defined in terms of the behavior of programs. — end note]

4.2 Namespaces and headers

[general.namespaces]

- ¹ Whenever a name *x* declared in subclause 21.11 at namespace scope is mentioned, the name *x* is assumed to be fully qualified as `::std::experimental::reflect::v1::x`, unless otherwise specified. The header described in this document (see Table 1) shall import the contents of `::std::experimental::reflect::v1` into `::std::experimental::reflect` as if by:

```
namespace std::experimental::reflect {
    inline namespace v1 {}
}
```

- ² Whenever a name *x* declared in the standard library at namespace scope is mentioned, the name *x* is assumed to be fully qualified as `::std::x`, unless otherwise specified.

Table 1 — Reflection library headers

<experimental/reflect>

4.3 Feature-testing recommendations

[general.features]

- ¹ An implementation that provides support for this document shall define each feature test macro defined in Table 2 if no associated headers are indicated for that macro, and if associated headers are indicated for a macro, that macro is defined after inclusion of one of the corresponding headers specified in the table.

Table 2 — Feature-test macros

Macro name	Value	Header
<code>__cpp_reflection</code>	201902	none
<code>__cpp_lib_reflection</code>	201902	<experimental/reflect>

5 Lexical conventions

[lex]

5.12 Keywords

[lex.key]

- ¹ In ISO/IEC 14882:2020 [lex.key], add the keyword reflexpr to the list of keywords in Table 4. Refer to Annex A for compatibility with the original features of ISO/IEC 14882:2020.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC TS 23619:2021
<https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021>

6 Basic concepts

[basic]

- ¹ The set of changes proposed in this document, as applied to the specification of ISO/IEC 14882:2020, shall be as per C++ Extensions for Concepts, ISO/IEC TS 19217:2015.
- ² In ISO/IEC 14882:2020 [basic], add the following last paragraph:

An *alias* is a name introduced by a *typedef* declaration, an *alias-declaration*, or a *using-declaration*.

6.2 One definition rule

[basic.def.odr]

- ¹ In ISO/IEC 14882:2020 [basic.def.odr], insert a new paragraph after the existing paragraph 3:
- A function or variable of static storage duration reflected by T (10.1.6.5) is odr-used by the specialization `std::experimental::reflect::get_pointer<T>` (21.11.4.9, 21.11.4.17), as if by taking the address of an *id-expression* nominating the function or variable.
- ² In ISO/IEC 14882:2020 [basic.def.odr], apply the following changes to the second bullet within paragraph 6:

and the object has the same value in all definitions of D, or a type implementing `std::experimental::reflect::Object` (21.11.3.1), as long as all operations (21.11.4) on this type yield the same constant expression results; and

6.9 Types

[basic.types]

6.9.1 Fundamental types

[basic.fundamental]

- ¹ In ISO/IEC 14882:2020 [basic.fundamental], apply the following change to paragraph 9:

An expression of type `void` shall be used only as an expression statement (6.2), as an operand of a comma expression (5.18), as a second or third operand of `?:` (5.16), as the operand of `typeid`, `noexcept`, `constexpr`, or `decltype`, as the expression in a return statement (6.6.3) for a function with the return type `void`, or as the operand of an explicit conversion to type `cv void`.

7 Standard conversions

[conv]

For Standard conversions, ISO/IEC 14882:2020, Clause 7 shall be used.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC TS 23619:2021
<https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021>

8 Expressions

[expr]

8.1 Primary expressions

[expr.prim]

8.1.2 Lambda expressions

[expr.prim.lambda]

- ¹ In ISO/IEC 14882:2020 [expr.prim.lambda], apply the following change to the second bullet in paragraph 12:

- names the entity in a potentially-evaluated expression (3.2) where the enclosing full-expression depends on a generic lambda parameter declared within the reaching scope of the *lambda-expression*;

where, for the process of this determination, `reflexpr` operands are not considered to be unevaluated operands.

- ² Also apply the following change to paragraph 18:

Every *id-expression* within the *compound-statement* of a *lambda-expression* that is an odr-use (6.2) of an entity captured by copy, as well as every use of an entity captured by copy in a *reflexpr-operand*, is transformed into an access to the corresponding unnamed data member of the closure type.

8.2 Postfix expressions

[expr.post]

- ¹ In ISO/IEC 14882:2020 [expr.post], apply the following change:

postfix-expression:

primary-expression
postfix-expression [*expression*]
postfix-expression [*braced-init-list*]
~~*postfix-expression* (*expression-list*_{opt})~~
function-call-expression
~~*simple-type-specifier* (*expression-list*_{opt})~~
~~*typename-specifier* (*expression-list*_{opt})~~
~~*simple-type-specifier* *braced-init-list*~~
~~*typename-specifier* *braced-init-list*~~
functional-type-conv-expression
postfix-expression . **template**_{opt} *id-expression*
postfix-expression -> **template**_{opt} *id-expression*
postfix-expression . *pseudo-destroyer-name*
postfix-expression -> *pseudo-destroyer-name*
postfix-expression ++
postfix-expression --
dynamic_cast < *type-id* > (*expression*)
static_cast < *type-id* > (*expression*)
reinterpret_cast < *type-id* > (*expression*)
const_cast < *type-id* > (*expression*)
typeid (*expression*)
typeid (*type-id*)

function-call-expression:

postfix-expression (*expression-list*_{opt})

functional-type-conv-expression:

simple-type-specifier (*expression-list*_{opt})
typename-specifier (*expression-list*_{opt})
~~*simple-type-specifier* *braced-init-list*~~
~~*typename-specifier* *braced-init-list*~~

expression-list:

initializer-list

9 Statements

[stmt.stmt]

For Statements, ISO/IEC 14882:2020, Clause 9 shall be used.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC TS 23619:2021
<https://standards.iteh.ai/catalog/standards/sist/00b300af-2ce5-4442-8ad5-1aab4b5b6172/iso-iec-ts-23619-2021>