TECHNICAL REPORT

ISO/IEC TR 25438

First edition
2006-08-01

Information technology — Common Language Infrastructure (CLI) — Technical Report: Common Generics

Technologies de l'information — Infrastructure commune de langage (ICL) — Rapport technique: Génériques communs

Reference number
ISO/IEC TR 25438:2006(E)

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

Published in Switzerland

Contents

Page

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

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.

ISO/IEC TR 25438 was prepared by Ecma International (as Ecma TR/89) and was adopted, under a special "fast-track procedure", by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

Introduction

This Technical Report defines a collection of types that are intended to enhance the common language nature of the CLI, by facilitating language inter-operation. The collection includes generic tuples, functions, actions, optional value representation, a type that can contain a value of one of two different types, and a utility filler type.

These types are experimental and will be considered for inclusion in a future version of the CLI International Standard. A reference implementation, written in C#, is included (see the accompanying file CommonGenericsLibrary.cs). This implementation source is also available from http://kahu.zoot.net.nz/ecma. A binary version is also available from that site, along with any updates to the proposal.

Feedback on these types is encouraged. (Please send comments to ecmacli@zoot.net.nz.)

Information technology — Common Language Infrastructure (CLI) — Technical Report: Common Generics

1 Scope

The CLI standard libraries (ISO/IEC 23271) provide a collection of common types that can be used by multiple languages. With the addition of generics to the CLI, the standard libraries have been extended to include a number of common generic types, in particular, collections. However, at present, these libraries do not include many simple generic types found in a number of different languages. Any language which uses these common types must implement them rather than deferring to the CLI library, thereby reducing language inter-operability. This Technical Report addresses this issue by providing a number of these common types.

Generic tuples (product types) are standard in a number of languages: C++ (`template Pair`), Ada, Haskell, and Standard ML (SML). However, languages differ in the number of pre-defined tuple sizes supported by their standard libraries; e.g. C++ provides just one (Pair) while Haskell provides eight (sizes 2 to 9) and SML allows any size of tuple. This Technical Report provides nine (sizes 2 to 10).

Generic programming encourages “higher order” programming where generic functions (methods) take function (delegate) type arguments that have generic types. Examples include Ada’s `with` and generic constraints, and function arguments in Haskell and SML. In the CLI, function values are provided in the form of delegates, so this proposal defines standard generic delegate types for functions (which return a value) and procedures (which do not).

Another two types that occur in a number of languages are an *optional* type, which either contains a value of some other type or an indication that such a value is not present; and an *either* type, which holds a value of one of two possible types and an indication of which one is present. This proposal provides both of these.

Note The optional type is similar to, but different from, the type `System.Nullable`.

Finally, in existing generic languages, a need has been found for a *filler* type to be used when a particular generic parameter is not required for a particular use of the generic type. A standard one-value type is often provided for this purpose, often called `Unit` or `Void`. This Technical Report includes such a type.

2 Rationale

2.1 Reference vs. value tuples

In some languages (such as C++ and Ada) tuple types can be value or reference types, while in others (such as Haskell and SML) they are always reference types. This implementation provides only value type tuples (which can contain value or reference types). Boxed versions of these types can be used when reference versions are required. Named boxed types would, of course, help greatly here, but these are not currently provided by the CLI Standard.

A simple generic type, such as `Boxed<T>`, can be written to address this issue. Such a type has been made available in the reference implementation of this proposal. However it is not being considered for standardization at this time.