

Foreword

This fifth edition cancels and replaces the fourth edition. Changes from the previous edition (and thus also ISO/IEC 23271:2006) were made to enhance and update this Standard.

The following features have been added, extended or clarified in the Standard:

- The presentation of the rules for assignment compatibility (§1.8.7, §III.1.8.1.2.3) has been extensively revised to a more precise and clearer relation-based format.
- The presentation of the verification rules for many IL instructions has been revised to be more precise and clearer by building upon the revisions to the presentation of assignment compatibility.
- The presentation of delegate signature compatibility has been revised along the same lines as assignment compatibility.
- The verification rules for the IL newobj instruction have been extended to cover general delegate creation.
- The dispatch rules for variance (§II.12.2) have been extended to define resolutions for the ambiguities that can arise.
- Type forwarders have been added to support the relocation of types between libraries (§II.6.8)

The following changes of behavior have been made to the Standard:

- The semantics of variance has been redefined making it a core feature of the CLI. In the previous edition of the Standard variance could be ignored by languages not wishing to support it (§1.8); as exact type matches always took precedence over matches-by-variance. In this edition the dispatch rules for interfaces §II.12.2) allow a match-by-variance to take precedence over an exact match, so all language implementation targeting the CLI must be aware of the behavior even if it is not support in the language (§1.8).
- Additional requirements on ilasm to metadata conversion. The left-to-right order of interfaces listed in a type header (§II.10.2) must now be preserved as a top-to-bottom order in the InterfaceImpl table (§II.22.23); and the top-to-bottom of method definitions (§II.10.2, §II.15) must now be preserved as a top-to-bottom order in the MethodDef table (§II.22.26). Both these additional requirements are required to support the revised variance semantics.
- System.Math and System.Double have been modified to better conform to IEEE (insert correct ref to IEEE standard here, and ref lib XML)

The following types have been added to the Standard or have been significantly updated (* represents an update).

Type	Library
System.Action	BCL
System.Action`1<-T>* ... System.Action`8<-T1..-T8>	BCL
System.Comparison`1<-T>*	BCL
System.Converter`2<-T,+U>*	BCL

System.IComparable<-T>*	BCL
System.Predicate<-T>*	BCL
System.Collections.Generic.IComparer`1<-T>*	BCL
System.Collections.Generic.IEnumerable`1<+T>*	BCL
System.Collections.Generic.IEqualityComparer`1<-T>*	BCL
System.Guid	BCL
System.MulticastDelegate	BCL
System.Reflection.CallingConventions	Runtime Infrastructure
System.Runtime.InteropServices.GuidAttribute	Runtime Infrastructure
System.Func`1<+TResult>...System.Func`9<-T1...-T8, +TResult>	BCL
System.Collections.Generic.Comparer`1<T>	BCL
System.Collections.Generic.EqualityComparer`1<T>	BCL
System.Collections.Generic.ISet`1<T>	BCL
System.Collections.Generic.LinkedList`1<T>	BCL
System.Collections.Generic.LinkedList`1<T>.Enumerator	BCL
System.Collections.Generic.LinkedListNode`1<T>	BCL
System.Collections.Generic.Queue`1<T>	BCL
System.Collections.Generic.Stack`1<T>	BCL
System.Collections.Generic.Stack`1<T>.Enumerator	BCL
System.Collections.Stack	BCL
System.DBNull	BCL
System.Runtime.InteropServices.Marshal	Runtime Infrastructure
System.Runtime.InteropServices.SafeBuffer	Runtime Infrastructure
System.Runtime.InteropServices.SafeHandle	Runtime Infrastructure
System.Threading.AutoResetEvent	BCL
System.Threading.EventWaitHandle	BCL
System.Threading.ManualResetEvent	BCL
System.WeakReference	BCL
System.Runtime.CompilerServices.TypeForwardedToAttribute	BCL
System.Runtime.CompilerServices.TypeForwardedFromAttribute	BCL
System.Threading.EventResetMode	BCL
System.Runtime.InteropServices.DllAttribute*	Runtime Infrastructure
System.Math*	BCL

Technical Report 89 (TR89), which was submitted during the third edition of this Ecma standard, will no longer be part of the submission. TR89 specified a collection of generic types, to help enhance inter-language interoperability, under consideration for inclusion in a future version of the standard. That consideration has

now occurred and TR89 has fulfilled its role. A selection of the types covered in TR89 has been introduced into this edition of the standard. An archive version of TR89 will continue to be available from Ecma.

The following companies and organizations have participated in the development of this standard, and their contributions are gratefully acknowledged: Eiffel Software, Kahu Research, Microsoft Corporation, Novell Corporation, Twin Roots. For previous editions, the following companies and organizations are also acknowledged: Borland, Fujitsu Software Corporation, Hewlett-Packard, Intel Corporation, IBM Corporation, IT University of Copenhagen, Jagger Software Ltd., Monash University, Netscape, Phone.Com, Plum Hall, and Sun Microsystems.