This documentation is archived and is not being maintained. Recommended Version

Common Language Specification

Visual Studio 2010

To fully interact with other objects regardless of the language they were implemented in, objects must expose to callers only those features that are common to all the languages they must interoperate with. For this reason, the Common Language Specification (CLS), which is a set of basic language features needed by many applications, has been defined. The CLS rules define a subset of the Common Type System; that is, all the rules that apply to the common type system apply to the CLS, except where stricter rules are defined in the CLS. The CLS helps enhance and ensure language interoperability by defining a set of features that developers can rely on to be available in a wide variety of languages. The CLS also establishes requirements for CLS compliance; these help you determine whether your managed code conforms to the CLS and to what extent a given tool supports the development of managed code that uses CLS features.

If your component uses only CLS features in the API that it exposes to other code (including derived classes), the component is guaranteed to be accessible from any programming language that supports the CLS. Components that adhere to the CLS rules and use only the features included in the CLS are said to be CLS-compliant components.

Most of the members defined by types in the .NET Framework Class Library are CLS-compliant. However, some types in the class library have one or more members that are not CLS-compliant. These members enable support for language features that are not in the CLS. The types and members that are not CLS-compliant are identified as such in the reference documentation, and in all cases a CLS-compliant alternative is available. For more information about the types in the .NET Framework class library, see the .NET Framework Class Library.

The CLS was designed to be large enough to include the language constructs that are commonly needed by developers, yet small enough that most languages are able to support it. In addition, any language construct that makes it impossible to rapidly verify the type safety of code was excluded from the CLS so that all CLS-compliant languages can produce verifiable code if they choose to do so. For more information about verification of type safety, see Managed Execution Process.

The following table summarizes the features that are in the CLS and indicates whether the feature applies to both developers and compilers (All) or only compilers. It is intended to be informative, but not comprehensive. For details, see the specification for the Common Language Infrastructure, Partition I, which is available on the Microsoft Developer Network (MSDN) Web site.

~~**Feature**~~	~~**Applies to**~~	~~**Description**~~
~~**General**~~		
~~Visibility~~	~~All~~	~~CLS rules apply only to those parts of a type that are exposed outside the defining assembly.~~
~~Global members~~	~~All~~	~~Global **static** fields and methods are not CLS-compliant.~~
~~**Naming**~~		
~~Characters and casing~~	~~All~~	~~CLS-compliant language compilers must follow the rules of Annex 7 of Technical Report 15 of the Unicode Standard 3.0, which governs the set of characters that can start and be included in identifiers. This standard is available from the Web site of the Unicode Consortium.~~