

# C++/CLI Migration Primer

06/05/2017 2 minutes to read

## In this article

[In This Section](#)

[See Also](#)

The new home for Visual Studio documentation is Visual Studio 2017 Documentation on [docs.microsoft.com](https://docs.microsoft.com).

The latest version of this topic can be found at [C++/CLI Migration Primer](#).

This is a guide to moving your Visual C++ programs from Managed Extensions for C++ to Visual C++. For a checklist summary of syntactic changes, see (NOT IN BUILD) [Managed Extensions for C++ Syntax Upgrade Checklist](#).

C++/CLI extends a dynamic component programming paradigm to the ISO-C++ standard language. The new language offers a number of significant improvements over Managed Extensions. This section provides an enumerated listing of the Managed Extensions for C++ language features and their mapping to Visual C++ where such a mapping exists, and points out those constructs for which no mapping exists.

## In This Section

### [Outline of Changes \(C++/CLI\)](#)

A high-level outline for quick reference, providing a listing of the changes under five general categories.

### [Language Keywords \(C++/CLI\)](#)

Discusses changes in language keywords, including the removal of the double underscore and the introduction of both contextual and spaced keywords.

### [Managed Types \(C++/CLI\)](#)

Looks at syntactic changes in the declaration of the Common Type System (CTS) – this includes changes in the declaration of classes, arrays (including the parameter array), enums, and so on.

### [Member Declarations within a Class or Interface \(C++/CLI\)](#)

Presents the changes involving class members such as scalar properties, index properties, operators, delegates, and events.

### [Value Types and Their Behaviors \(C++/CLI\)](#)

Focuses on value types and the new family of interior and pinning pointers. It also discusses a number of significant semantics changes such as the introduction of implicit boxing, immutability of boxed value types, and the removal of support for default constructors within value classes.

### [General Language Changes \(C++/CLI\)](#)

Details semantic changes such as support for cast notation, string literal behavior, and changes in the semantics between ISO-C++ and C++/CLI.

## See Also

Mixed (Native and Managed) Assemblies

Component Extensions for Runtime Platforms