

Common Language Runtime (CLR) Overview - SQL Server

By rwestMSFT

Archived: 2026-04-05 17:07:39 UTC

Applies to:  [SQL Server](#)  [Azure SQL Managed Instance](#)

SQL Server and [Azure SQL Managed Instance](#) enable you to implement some of the functionalities with .NET languages using the native common language runtime (CLR) integration as SQL Server server-side modules (procedures, functions, and triggers). The CLR supplies managed code with services such as cross-language integration, code access security, object lifetime management, and debugging and profiling support.

For SQL Server users and application developers, CLR integration means that you can write stored procedures, triggers, user-defined types, user-defined functions (scalar and table valued), and user-defined aggregate functions using any .NET Framework language, including C# Visual Basic .NET. SQL Server includes the .NET Framework version 4 preinstalled.

This 6-minute video shows you how to use CLR in Azure SQL Managed Instance:

Code access security no longer supported

CLR uses Code Access Security (CAS) in the .NET Framework, which is no longer supported as a security boundary. A CLR assembly created with `PERMISSION_SET = SAFE` might be able to access external system resources, call unmanaged code, and acquire sysadmin privileges. In SQL Server 2017 (14.x) and later versions, the `sp_configure` option, [clr strict security](#), enhances the security of CLR assemblies. `clr strict security` is enabled by default, and treats `SAFE` and `EXTERNAL_ACCESS` assemblies as if they were marked `UNSAFE`. The `clr strict security` option can be disabled for backward compatibility, but isn't recommended.

We recommend that you sign all assemblies by a certificate or asymmetric key, with a corresponding login that has been granted `UNSAFE ASSEMBLY` permission in the `master` database. SQL Server administrators can also add assemblies to a list of assemblies, which the Database Engine should trust. For more information, see [sys.sp_add_trusted_assembly](#).

When to use CLR modules

CLR Integration enables you to implement complex features that are available in .NET Framework such as regular expressions, code for accessing external resources (servers, web services, databases), custom encryption, etc. Some of the benefits of the server-side CLR integration are:

- **A better programming model.** The .NET Framework languages are in many respects richer than Transact-SQL, offering constructs and capabilities previously not available to SQL Server developers. Developers can also use the power of the .NET Framework Library, which provides an extensive set of classes that can be used to quickly and efficiently solve programming problems.

- **Improved safety and security.** Managed code runs in a common language run-time environment, hosted by the Database Engine. SQL Server uses this to provide a safer and more secure alternative to the extended stored procedures available in earlier versions of SQL Server.
- **Ability to define data types and aggregate functions.** User-defined types and user-defined aggregates are two new managed database objects that expand the storage and querying capabilities of SQL Server.
- **Streamlined development through a standardized environment.** Database development is integrated into future releases of the Visual Studio .NET development environment. Developers use the same tools for developing and debugging database objects and scripts as they use to write middle-tier or client-tier .NET Framework components and services.
- **Potential for improved performance and scalability.** In many situations, the .NET Framework language compilation and execution models deliver improved performance over Transact-SQL.

[SQL Server Language Extensions](#) provide an alternative execution environment for runtimes close to the Database Engine. For a discussion of the differences between SQL CLR and SQL language extensions, see [Compare SQL Server Language Extensions to SQL CLR](#).

The following table lists the articles in this section.

Related content

- [.NET Framework installation guide](#)
- [Performance of CLR integration architecture](#)

Source: <https://docs.microsoft.com/en-us/sql/relational-databases/clr-integration/common-language-runtime-integration-overview?view=sql-server-2017>