



عنوان مقاله: گام به گام با قابلیت CLR در SQL Server

نویسنده مقاله: تیم فنی نیک آموز

تاریخ انتشار: ۱ خرداد ۱۴۰۳

منبع: <https://nikamooz.com/clr-functionality-in-sql-server>

قابلیت CLR در SQL Server مثل یک ماشین اسپرت با موتور جت است؛ به عبارتی، سرعت و قدرت را یکجا باهم دارد. CLR دروازه‌ای به سوی دنیای برنامه‌نویسی .NET در SQL Server می‌گشاید. این قابلیت به شما امکان می‌دهد تا با استفاده از زبان‌های قدرتمند مانند سی پلاس و VB.NET، افق‌های تازه‌ای در زمینه مدیریت و تحلیل داده‌ها تجربه کنید. در این مقاله، شما را با صفر تا صد قابلیت CLR آشنا خواهیم کرد. پس از خواندن این مقاله، به راحتی می‌توانید لذت کدنویسی را با قابلیت CLR برای خود چندین برابر کنید. در این مقاله، از زبان T-SQL برای کدنویسی استفاده شده است.

قابلیت CLR در SQL Server چیست؟

CLR مخفف Common Language Runtime، محیطی را برای اجرای کدهای .NET، مانند سی پلاس و VB.NET در [SQL Server](#) فراهم می‌کند. این قابلیت به توسعه‌دهندگان اجازه می‌دهد تا از قدرت و انعطاف‌پذیری زبان‌های .NET برای ایجاد انواع داده سفارشی در SQL Server استفاده کنند و محدودیت‌های T-SQL را دور بزنند. محیط CLR قلب تپنده [فریم ورک دات نت](#) محسوب می‌شود و تمام کدهای نوشته‌شده با این فریم‌ورک اجرا می‌شود.



مزایای استفاده از CLR در SQL Server

مزایای متعدد CLR برای کوئری‌های پیچیده، توانسته توجه‌ها را به سمت خود بکشاند. قابلیت این محیط برای کوئری‌های ساده و ابتدایی، چندان توصیه نمی‌شود؛ زیرا در اینصورت، انگار یک متخصص ناسا را برای تعمیر یک دوچرخه ساده به کار گرفتید. در مقابل، تصور کنید به جای نوشتن کدهای پیچیده Transact-SQL برای حل یک مشکل خاص، می‌توانید از قدرت زبان‌های برنامه‌نویسی NET. مانند **سی شارپ** یا VB.NET در SQL Server استفاده کنید؛ این همان جادوی CLR است. CLR در SQL Server حکم یک جعبه ابزار جادویی را دارد که به شما مزایای زیر را ارائه می‌دهد:

مدیریت خودکار حافظه و ایمنی نوع داده

قابلیت CLR با کد مدیریت‌شده NET. موجب مدیریت خودکار حافظه شده و از خطاهای ناشی از مدیریت دستی جلوگیری می‌کند. CLR ها از سیستم ایمنی نوع داده بهره می‌برند؛ با این قابلیت، شما به عنوان توسعه‌دهنده می‌توانید روی منطق اصلی برنامه تمرکز کنید و نگرانی از بابت سازگاری **انواع داده** نداشته باشید.

استفاده از کلاس NET. و پیاده سازی آسان منطق پیچیده

قابلیت CLR در SQL Server به شما امکان می‌دهد تا از کلاس قدرتمند فریم‌ورک NET. استفاده کنید؛ در نتیجه، با چند خط کد ساده، می‌توانید وظایف پیچیده و زمان‌بر در SQL Server را به طور خودکار انجام دهید. کتابخانه‌های NET. امکان دسترسی به قابلیت‌های مختلف را فراهم می‌کند؛ قابلیت‌هایی که به طور مستقیم در CLR قابل استفاده است.

بهره مندی از قابلیت برنامه نویسی شی‌گرا (object-oriented)

پشتیبانی CLR از قابلیت برنامه‌نویسی شی‌گرا مانند Polymorphism، Encapsulation و Inheritance باعث می‌شود تا کدنویسی راحت‌تر شود. علاوه بر این، امکان استفاده مجدد از کد را برایتان فراهم می‌کند.

قابلیت توسعه با زبان های پشتیبانی شده توسط NET.

برخلاف رویه سنتی و محدودیت‌های Transact-SQL، محیط CLR را با هر زبان برنامه‌نویسی که توسط NET. پشتیبانی شود، می‌توان توسعه داد؛ این انعطاف‌پذیری، دستتان را برای کدنویسی با زبان ترجیحی خودتان باز می‌گذارد.

سازگار با نسخه های مختلف دیتابیس اوراکل

اگر از دیتابیس اوراکل استفاده می‌کنید، خبر خوب اینکه CLR با ورژن 10g R2 و بالاتر آن سازگار است؛ در نتیجه، می‌توانید از قابلیت‌های آن در محیط دیتابیس اوراکل نیز بهره ببرید.

کاربردهای CLR در SQL Server

CLR حکم یک ابرقدرت برای SQL Server دارد و به شما این امکان را می‌دهد که فراتر از مرزهای عادی پا بگذارید و از محدودیت‌های T-SQL بگذرید، اما برای همه سناریوها مناسب نیست. در اینجا به موقعیت‌هایی اشاره می‌کنیم که قابلیت CLR در SQL Server می‌درخشد:

۱. پیاده سازی منطق پیچیده بیزینس

زمانی که با سناریویی روبه‌رو هستید که نیازمند پیاده‌سازی منطق پیچیده است، CLR به کمک شما می‌آید. توانایی استفاده از کتابخانه‌های قدرتمند NET و امکانات برنامه‌نویسی شی‌گرا باعث می‌شود توسعه این منطق، به‌سادگی و با سازماندهی بهتر انجام شود.

۲. بهبود عملکرد برای محاسبات سنگین

فرآیندهای مبتنی بر قابلیت‌های CLR در NET به دلیل ماهیت کامپایل و مدیریت‌شده، عملکرد بهتری نسبت به T-SQL سنتی در سناریوهایی با محاسبات بسیار از خود نشان می‌دهند. به عبارت دیگر، این پلتفرم می‌تواند حجم بیشتری از محاسبات را در زمان کوتاه‌تری انجام دهد.

۳. انجام امور غیرممکن با T-SQL

برخی از کارها مانند دسترسی به منابع سیستم، پیاده‌سازی رمزنگاری و برقراری ارتباط با وب‌سرویس به‌طور مستقیم با استفاده از T-SQL قابل انجام نیست. CLR SP این محدودیت را برطرف می‌کند و با قابلیت خود به شما امکان می‌دهد تا چنین وظایفی را به‌طور مستقیم درون SP (مخفف [Stored Procedure](#)) پیاده‌سازی کنید.

۴. عملیات با امنیت بالا

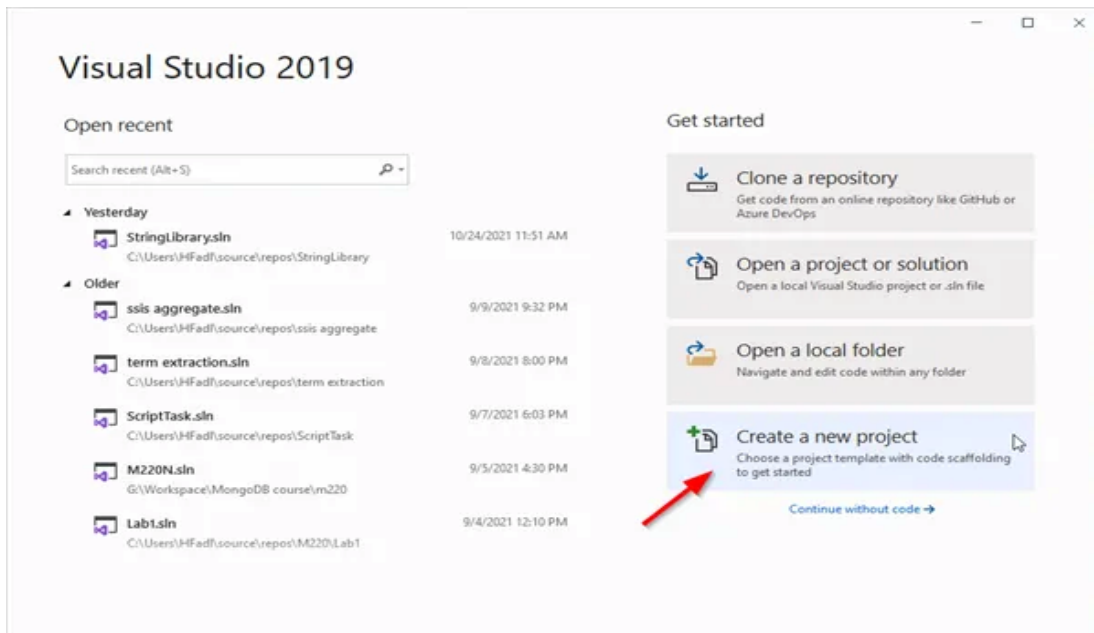
در برخی از موارد، نیازمند امنیت بالا در انجام عملیات هستیم. CLR SP با برخورداری از قابلیت‌های امنیتی NET، گزینه مناسبی برای چنین سناریوهایی است. در نظر داشته باشید استفاده از CLR SP همیشه بهترین انتخاب نیست. در صورتی که با کوئری‌های ساده و ابتدایی سروکار دارید، T-SQL SP به دلیل سادگی و کارایی بالاتر، گزینه مناسب‌تری خواهد بود. همچنین، پیاده‌سازی CLR SP نیازمند دانش برنامه‌نویسی NET است.



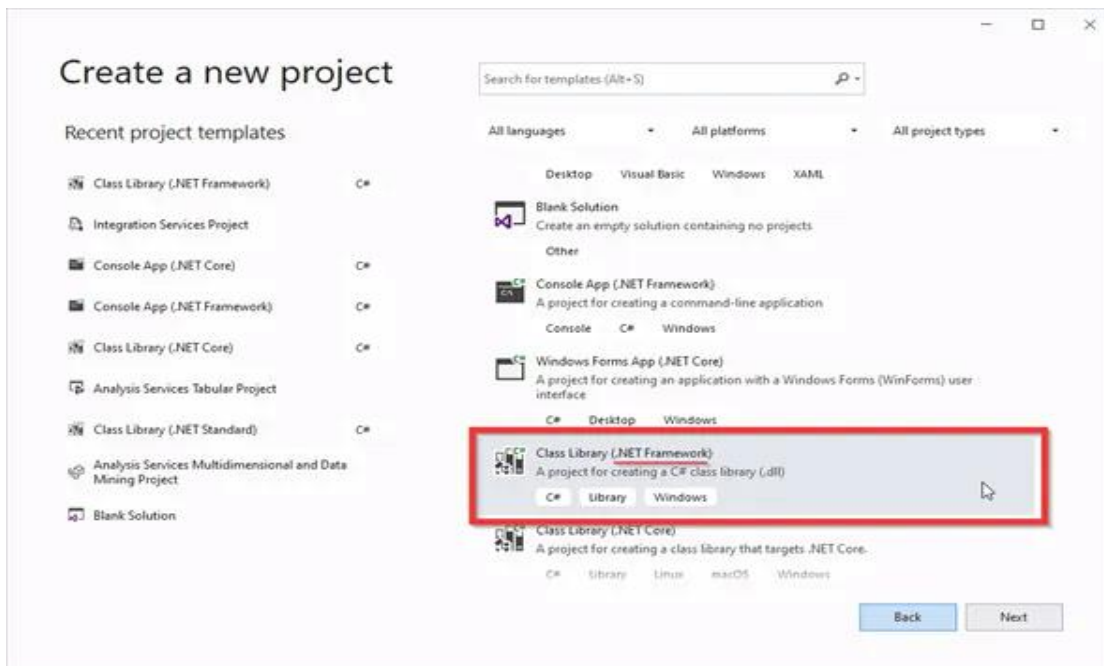
مراحل ایجاد اولین پروژه CLR در SQL Server

برای استفاده از توابع دلخواه C# در SQL Server، نیاز به یک کتابخانه کلاس .NET داریم. پس بیاید باهم مراحل ساخت این کتابخانه را قدم به قدم طی کنیم:

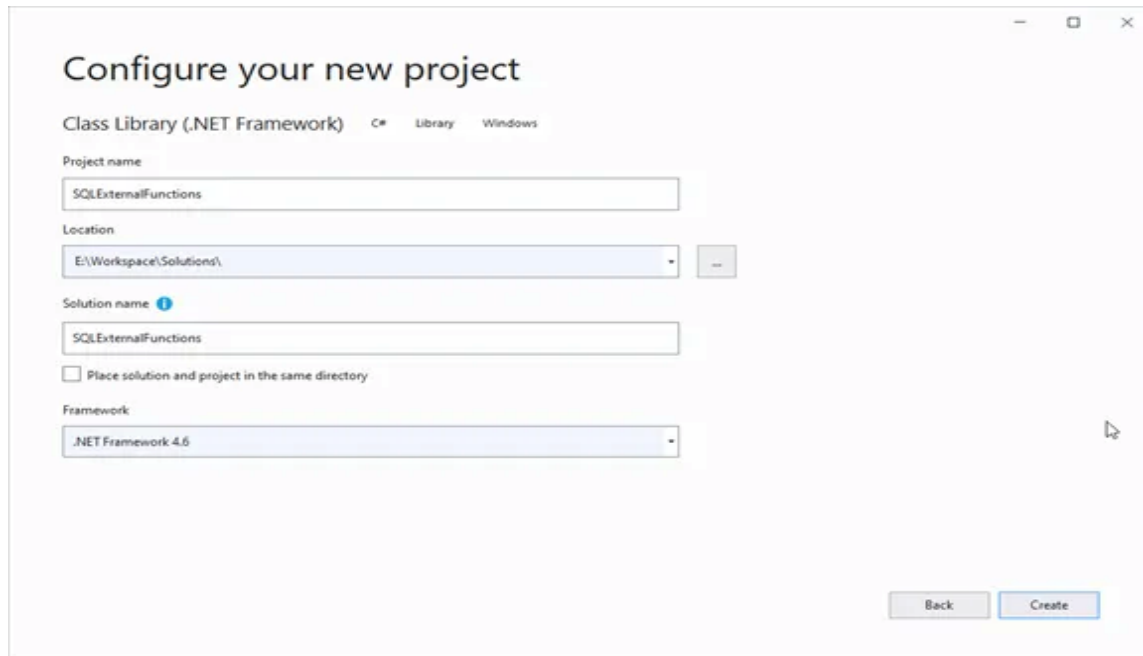
- **گام اول:** اولین کار، ایجاد یک پروژه جدید در نرم افزار Microsoft Visual Studio است.



- **گام دوم:** برای ساخت یک کتابخانه کلاس CLR برای SQL Server، باید از نوع پروژه «.NET Framework Class Library» استفاده کنیم.



- **گام سوم:** در این مثال، ما از فریم ورک .NET Framework نسخه ۴.۶ برای ساخت کتابخانه استفاده کردیم و نام پروژه را «SQL External Functions» در نظر گرفتیم.



- **گام چهارم:** پس از باز شدن کامل محیط ویژوال استودیو، به تب «Solution Explorer» در سمت راست صفحه بروید و نام فایل «Class1.cs» را به «SQL External Functions.cs» تغییر دهید.
- **گام پنجم:** حالا نوبت نوشتن تابعی است که در SQL Server می‌خواهیم استفاده کنیم. ابتدا بخش‌های غیرضروری فایل «SQL External Functions.cs» را حذف می‌کنیم. روی این فایل دابل کلیک کنید تا وارد ویرایشگر کد C# شوید.
- **گام ششم:** در ویرایشگر، آکولادهای مربوط به «namespace SQL External Functions» و کتابخانه‌های غیرضروری را حذف کنید. فقط System.Linq و System.Linq را نگه دارید. همانطور که در تصویر زیر مشاهده می‌کنید:

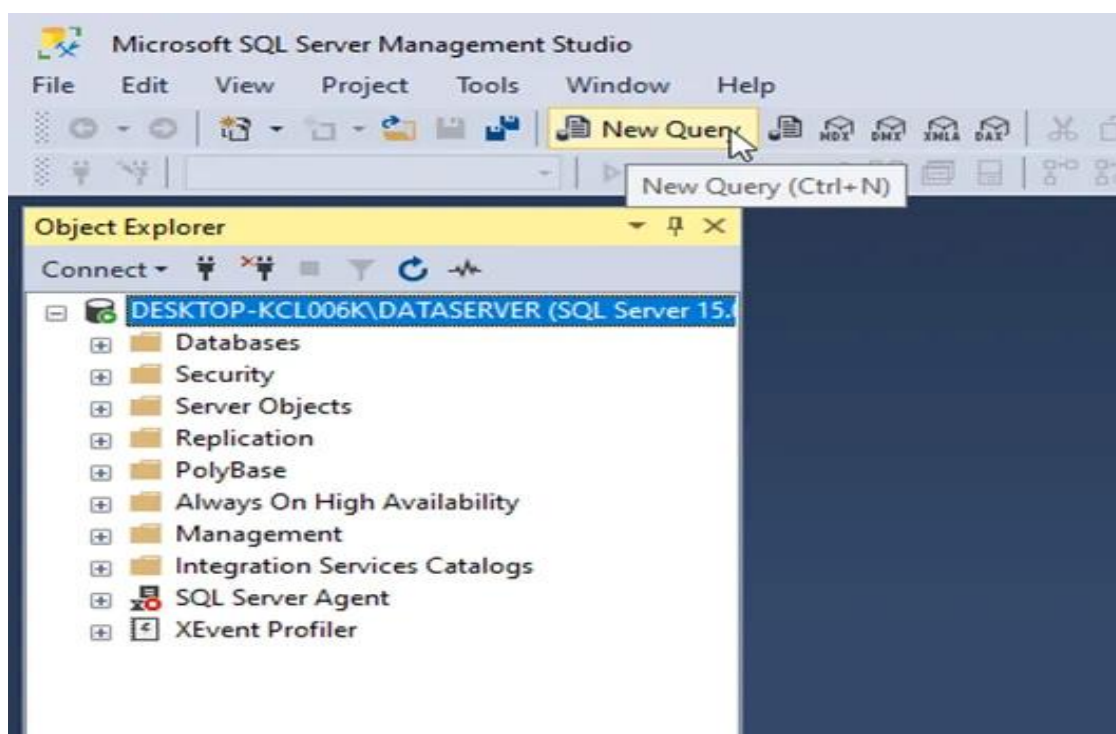


- **گام هفتم:** از آنجایی که توابع CLR در SQL Server باید درون یک کلاس استاتیک تعریف شوند، نیاز است کلیدواژه «static» را قبل از تعریف کلاس اضافه کنیم.

```
public static class SQLExternalFunctions
```

با دنبال کردن این مراحل، حالا یک کتابخانه کلاس .NET. اولیه برای استفاده از توابع C# در SQL Server در اختیار داریم. تا اینجا، کتابخانه کلاس .NET. را برای استفاده در SQL Server ایجاد کردیم. حالا نوبت به ساختن یک تابع کاربر-تعریف شده CLR در SQL Server می‌رسد که از توابع موجود در کتابخانه کلاس شما استفاده کند.

- **گام هشتم:** در ادامه، قصد ایجاد یک کوئری جدید را داریم. برای این کار، [Studio SQL Server Management](#) را باز می‌کنیم.



- **گام نهم:** قبل از تعریف تابع CLR، باید سرور را برای استفاده از قابلیت یکپارچه‌سازی CLR پیکربندی کنیم. این کار را در بخش بعدی انجام خواهیم داد.

نصب و پیکربندی CLR در SQL Server

قابلیت CLR در SQL Server به طور پیش فرض، غیرفعال است. فعال بودن CLR در SQL Server به این معنی است که شما می‌توانید از تمام توانایی‌های دات نت (.NET). برای توسعه قسمت‌های مختلف پایگاه داده خود استفاده کنید. این قابلیت به شما اجازه می‌دهد تا رویه‌های ذخیره شده (stored procedures)، **تریگرها** (triggers)، انواع کاربر-تعریف شده (user-defined types)، توابع تک‌مقدار و چند مقداره کاربر-تعریف شده (user-defined functions) و توابع جمع کاربر-تعریف شده (user-defined aggregate functions) را به زبان‌های مختلف دات نت، از جمله ویژوال بیسیک، بنویسید.

در ادامه مقاله گام به گام با قابلیت CLR در SQL Server ، به سادگی مراحل فعال سازی CLR را بررسی می کنیم:

گام اول: روشن کردن قابلیت ادغام CLR

برای فعال کردن CLR، از دستور sp_configure در Transact-SQL استفاده می کنیم. کافی است کد زیر را در SQL Management Studio اجرا کنید:

```
sp_configure 'clr enabled', 1
GO
RECONFIGURE
GO
```

با اجرای این کد، CLR در سرور SQL شما روشن می شود. به همین سادگی.

اگر بعداً تصمیم به غیرفعال کردن CLR گرفتید، از دستور مشابه با مقدار ۰ برای گزینه clr enabled می توانید استفاده کنید:

```
sp_configure 'clr enabled', 0
GO
RECONFIGURE
GO
```

گام دوم: فعال کردن نمایش تنظیمات پیشرفته

با استفاده از دستور زیر، امکان مشاهده و تغییر تنظیمات پیشرفته SQL Server را فعال کنید:

```
EXEC sp_configure 'show advanced options' , 1;
RECONFIGURE;
```

مدیریت ماژول های CLR در SQL Server

در این قسمت از قابلیت CLR در SQL Server ، برای استفاده از کتابخانه برنامه های کاربردی CLR در SQL Server ، باید یک ماژول در SQL Server ایجاد کرده و آن را به مسیر فایل کتابخانه وصل کنید. این کار به شما امکان می دهد تا از توابع و کلاس های موجود در کتابخانه کلاس SQL Server بهره ببرید. برای این کار، دستور زیر را پیاده سازی کنید:

```
CREATE ASSEMBLY SQLCLRDemo
FROM 'D:\CLR Functions\SQLExternalFunctions.dll';
GO.
```

در این مثال، یک ماژول به نام SQLCLR Demo ایجاد می شود. سپس به فایل DLL SQL External Functions.dll که در دایرکتوری D:\CLR Functions قرار دارد، متصل خواهد شد.

ایجاد توابع CLR در SQL Server

پس از ایجاد ماژول می‌توانید از کلاس و توابع موجود در کتابخانه CLR در T-SQL استفاده کنید. توابع CLR به برنامه‌نویس این امکان را می‌دهند تا به منابع خارجی مثل فایل‌ها، منابع شبکه‌ای، سرویس‌های وب و پایگاه‌های داده دیگر (شامل سایر نسخه‌های SQL Server) دسترسی پیدا کند. این کار با استفاده از کلاس‌های مختلفی در فریم‌ورک دات نت مانند System.IO ، System.Web Services ، و System.Data.SqlClient انجام می‌شود.

ایجاد توابع CLR کار ساده‌ای است. برای هر تابعی که در کتابخانه کلاس C# تعریف کردید، باید یک تابع CLR مرتبط در SQL Server ایجاد و پارامترهای ورودی و خروجی مشابه با تابع سی شارپ را مشخص کنید. برای این کار، دستور زیر را به کار ببرید:

```
DECLARE @result INT;
```

```
EXEC SQLCLRDemo.SQLExternalFunctions.Multiply @result OUTPUT, 10, 20;
```

```
SELECT @result;
```

برای مثال، فرض کنید سه تابع در کتابخانه کلاس C# به نام "SQLCLR Demo.SQLExternalFunctions" و با نام‌های "First Character Occurrence" ، "Get Character Occurrence" و "Get Current Date ISO" تعریف کرده‌ایم. در SQL Server، توابع CLR مرتبط با آن‌ها را به صورت زیر ایجاد می‌کنیم:

```
CREATE FUNCTION GetFirstCharacterOccurence(@str nvarchar(255)) RETURNS INT
EXTERNAL NAME SQLCLRDemo.SQLExternalFunctions.FirstCharacterOccurence;
GO
```

```
CREATE FUNCTION GetCharacterOccurence(@str nvarchar(255), @chr nchar(1))
RETURNS INT
EXTERNAL NAME SQLCLRDemo.SQLExternalFunctions.GetCharacterOccurence;
GO
```

```
CREATE FUNCTION GetIsoDate() RETURNS nvarchar(255)
EXTERNAL NAME SQLCLRDemo.SQLExternalFunctions.GetCurrentDateISO;
GO
```

تکنیک های امنیتی مورد نیاز برای استفاده از CLR در SQL Server

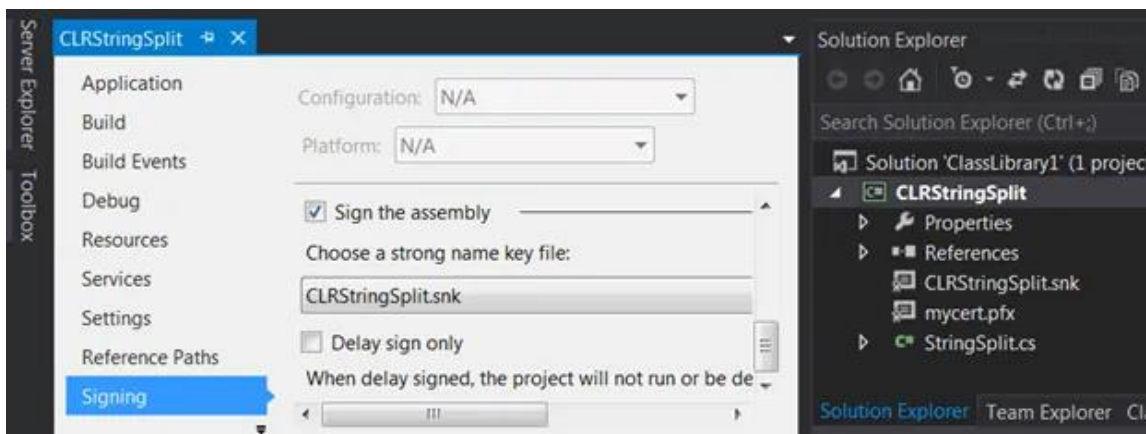
بعد از فعال‌سازی قابلیت یکپارچه سازی CLR در SQL Server ، زمان آن رسیده است که روی نحوه استفاده امن از آن تمرکز کنید. خبر خوب این که از نسخه ۲۰۱۷ به بعد، SQL Server یک ویژگی امنیتی جدید به نام «clr strict security» را به صورت پیش‌فرض در اختیار شما قرار می‌دهد.

این ویژگی در قابلیت CLR در SQL Server به چه معناست؟ فرض کنید یکسری ماژول برای استفاده در SQL Server نوشتید. این کدها قبلاً بدون هیچ امضایی بارگذاری می‌شدند. اما حالا با فعال بودن «clr strict security»، هر کدی که امضاء نداشته باشد، قابل بارگذاری نیست. برای رفع این مشکل، می‌توانید ماژول را با یک امضای دیجیتال معتبر امضا کنید. این امضا را می‌توانید از طریق یک گواهینامه معتبر (Certificate) یا یک کلید نامتقارن

(Asymmetric Key) انجام دهید. حسابی که این امضا را انجام می‌دهد، باید در SQL Server از سطح دسترسی «UNSAFE ASSEMBLY» برخوردار باشد.

امضای ماژول CLR با کلید نامتقارن

برای امضای ماژول CLR با استفاده از کلید نامتقارن، باید به پروژه ویژوال استودیویی برگردیم که برای ساخت ماژول استفاده کردیم. در این پروژه، به بخش تنظیمات (Properties) رفته و ماژول را مطابق شکل زیر امضا کنیم.



بعد از اینکه پروژه ماژول CLR خود را دوباره ساختید، باید به SSMS برگردید و مراحل زیر را برای امضای ماژول با کلید نامتقارن دنبال کنید:

- **مرحله ۱:** ایجاد کلید نامتقارن از فایل ماژول

```
USE master;
GO
CREATE ASYMMETRIC KEY CLRStringSplitKey FROM EXECUTABLE FILE =
'C:\CLRStringSplit.dll';
GO
```

- **مرحله ۲:** ساختن یک حساب کاربری (Login) در SQL Server مرتبط با کلید نامتقارن

```
USE master;
GO
CREATE LOGIN CLRStringSplitKeyLogin FROM ASYMMETRIC KEY CLRStringSplitKey;
GO
```

- **مرحله ۳:** اعطای مجوز «اجرای کدهای غیر ایمن» (UNSAFE assembly) به حساب کاربری ساخته شده در مرحله ۲

```
USE master;
GO
GRANT UNSAFE ASSEMBLY TO CLRStringSplitKeyLogin;
GO
```

- **مرحله ۴:** ایجاد یک کاربر پایگاه داده SQL Server برای حساب کاربری ساخته شده در مرحله ۲

```
USE SampleTest2017;
GO
CREATE USER CLRStringSplitKeyLogin FOR LOGIN CLRStringSplitKeyLogin;
GO
```

- **مرحله ۵:** ساختن ماژول CLR

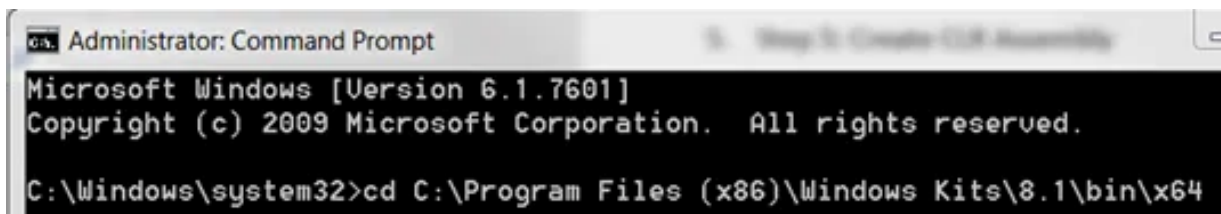
```
USE SampleTest2017;
GO
CREATE ASSEMBLY CLRStringSplit FROM 'C:\CLRStringSplit.dll' WITH
PERMISSION_SET = SAFE;
GO
```

با استفاده از روش امضای کد و کنترل سطح دسترسی، می‌توانید از قابلیت CLR در SQL Server به شکلی امن و مطمئن بهره‌مند شوید.

امضای ماژول CLR با کلید گواهینامه

- **مرحله ۱:** باز کردن Command Prompt با دسترسی مدیر (Administrator)

ابتدا برنامه Command Prompt ویندوز را با دسترسی مدیر اجرا کنید. سپس به پوشه فرعی Windows Kits در دایرکتوری مربوط به کیت‌های توسعه‌دهنده ویندوز بروید. مسیر این پوشه در مثال ما به صورت زیر است:

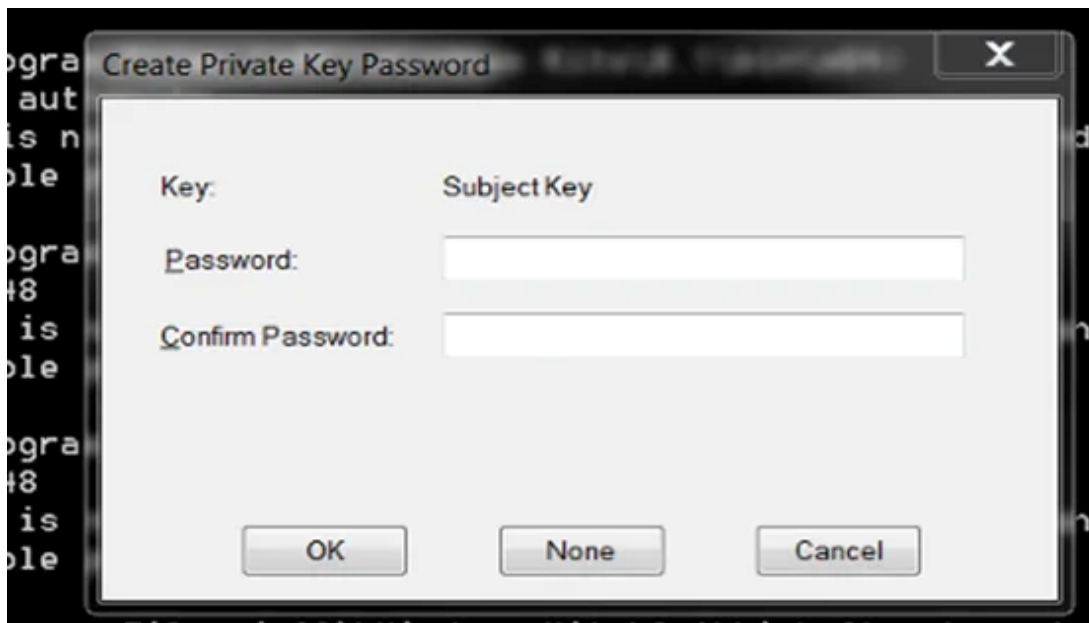


- **مرحله ۲:** ایجاد گواهینامه

با استفاده از برنامه makecert.exe (که باید در پوشه Windows Kits باشد) یک گواهینامه جدید ایجاد کنید. دستور زیر را در Command Prompt بنویسید:

```
C:\Program Files (x86)\Windows Kits\8.1\bin\x64>makecert -r -pe -n
"CN=CLRString Split Test Root Authority" -a sha256 -sky signature -cy
authority -sv CLRStringSplit.pvk -len 2048 -m 144 CLRStringSplit.cer
```

در حین اجرای این کد، پنجره زیر برای ایجاد رمز عبور برای کلید خصوصی ظاهر می‌شود.



• **مرحله ۳: ایجاد فایل PFX**

پس از ایجاد موفقیت‌آمیز گواهینامه، با استفاده از برنامه PVK2PFX.exe یک فایل PFX بسازید. دستور زیر را اجرا کنید:

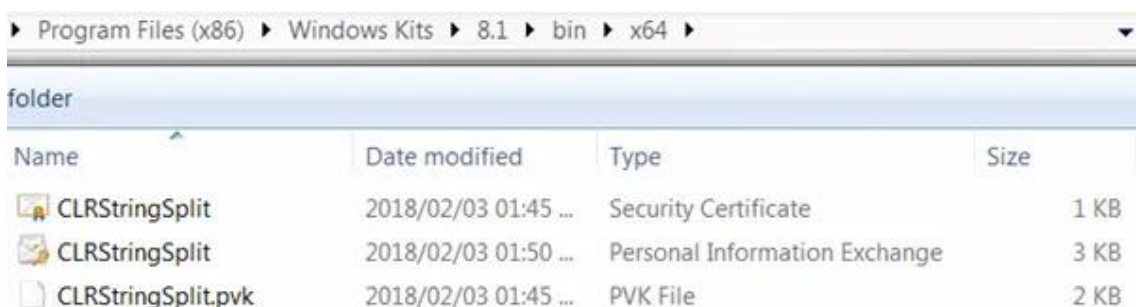
```
C:\Program Files (x86)\Windows Kits\8.1\bin\x64>PVK2PFX -pvk
CLRStringSplit.pvk -spc CLRStringSplit.cer -pfx CLRStringSplit.pfx -pi
P@ssw0rd1 -po P@ssw0rd1
```

• **مرحله ۴: امضای ماژول CLR**

درنهایت، با استفاده از برنامه signtool.exe ماژول CLR خود را امضا کنید:

```
C:\Program Files (x86)\Windows Kits\8.1\bin\x64>signtool sign /f
CLRStringSplit.pfx /p P@ssw0rd1 CLRStringSplit.dll
```

در دستور ۳ و ۴ P@ssw0rd1 را با رمزعبور دلخواه خود جایگزین کنید. با اجرای مراحل بالا، باید سه فایل جدید در پوشه Windows Kits ایجاد شده است:



• **مرحله ۵:** ایجاد گواهینامه و ورود به SQL Server Management Studio (SSMS)

اکنون به نرم‌افزار SSMS بروید و مراحل زیر را دنبال کنید:

۱. ایجاد گواهینامه

```
USE master;
GO
CREATE CERTIFICATE CLRStringSplitCert FROM FILE =
'C:\sqlclr\CLRStringSplit.cer';
GO
```

۲. ایجاد لاگین SQL Server برای گواهینامه

```
USE master;
GO
CREATE LOGIN CLRStringSplitCertLogin FROM CERTIFICATE CLRStringSplitCert;
GO
```

۳. اعطای مجوز «UNSAFE ASSEMBLY» به ورود ایجادشده

```
USE master;
GO
GRANT UNSAFE ASSEMBLY TO CLRStringSplitCertLogin;
GO
```

۴. ساخت کاربر برای پایگاه داده

```
USE SampleTest2017;
GO
CREATE USER CLRStringSplitCertLogin FOR LOGIN CLRStringSplitCertLogin;
GO
```

۵. ایجاد ماژول CLR امضا شده

```
USE SampleTest2017;
GO
CREATE ASSEMBLY CLRStringSplit FROM 'C:\CLRStringSplit.dll' WITH
PERMISSION_SET = SAFE;
GO
```

مسیر فایل DLL خود را در دستور بالا می‌توانید جایگزین C:\CLR String Split.dll کنید.

کلام پایانی: بررسی جامع قابلیت CLR در SQL Server

قابلیت CLR در SQL Server به توسعه‌دهندگان کمک می‌کند تا از قدرت و انعطاف‌پذیری زبان‌های .NET برای ایجاد توابع و رویه‌های ذخیره‌شده در دیتابیس SQL Server استفاده کنند. این قابلیت‌های قدرتمند، امکان انجام وظایف پیچیده را ساده کرده و به شما کمک می‌کند برنامه‌های یکپارچه با T-SQL و .NET ایجاد کنید.