Use of .net Assemblies in SQL Server 2005

Asked 16 years, 3 months ago   Modified 15 years, 5 months ago

Viewed 908 times

I've recently discovered that it's possible to place .net assemblies on SQL Server >=2005 servers so that .net functions can be called in T/SQL statements.

I wondered what uses people found for these and how they perform?

.net   sql-server

Share

Improve this question

Follow

5 Answers

Sorted by: Highest score (default)

3

The first general purpose use for the CLR in SQL 2005 I created was a SQL 2005 assembly that has a variety of functions that perform string operations and pattern matches using regular expressions. The native string functions in SQL 2005 can be augmented so that you can

validate common formats like phone numbers or credit card numbers or perform ad-hoc regular expressions within stored procedures.

For deterministic user defined functions, I have found the SQL CLR support to be very performant.

Share Improve this answer

Follow

Oppositional
11.2k ● 6 ● 51 ● 63

1   Rather than writing your own library for this you can also use the SQL# library (both a free and a pay version) sqlsharp.com . It provides many features right out of the box. I'm a happy user of the free version :) – Joe Kuemerle Jul 17, 2009 at 19:58

I found it to be very useful.

0

I used this possibility to extend MSSQL2005 XML related functions.

If I remember correctly you can even introduce your own data types.
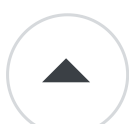
Share Improve this answer

Follow

aku
124k ● 33 ● 176 ● 203

This is normally used if you need to interact with the operating system in some way, for example, to place a

0

message in MSMQ or write to a file. It is also useful if you have some complex mathematical or financial calculations that are already implemented in .NET, and you don't want to re-write them in T-SQL.

Share   Improve this answer

Follow

answered Sep 2, 2008 at 14:36

Eric Z Beard
38.4k ● 27 ● 101 ● 147

0

CLR integration with SQL 2005 is particularly useful for user-defined functions/stored procs written in .NET and also user-defined data types. e.g. You could write a heavy duty data type that allows SQL to define objects and reference properties. You could write some super-duper datetime variation, for instance.

Share   Improve this answer

Follow

answered Sep 2, 2008 at 14:56

spoulson
21.6k ● 16 ● 78 ● 102

0

Oppositional, I agree with you. Assemblies are wonderful for all of those things that not at all attainable or barely attainable in T-SQL (via in hackish methods).

Simplicity of use is the key. Please be aware of the following implications...

1. The use of an assembly introduces trust/security/permission configuration issues.

An assembly will have to be defined into the database. If the assembly is not signed, the database will have to allow UNTRUSTED code. This database setting is reset when a database is detached/reattached.

2. The use of an assembly affects the SQL Server configuration (not just a database).

To run an assembly, you must first insure that SQL Server lightweight pooling is disabled.

3. For the Assemblies functions, remember to be conservative with data manipulations; especially, on busy tables.

If you can, avoid the use of an Assembly method that attempts to affect too many rows at a time. Take care for those used within table triggers..as some of these will degrade performance sharply.

Share   Improve this answer

Follow

answered Jul 17, 2009 at 19:41

CMB
456 ● 3 ● 6