

## Registro y Uso de Procedimientos y Funciones .NET con CLR en SQL Server

Usando Visual Studio y cualquiera de los lenguajes de la plataforma .NET (Visual C#, Visual C++, Visual J# y Visual Basic) se pueden crear librerías (ficheros.DLL) que contengan diferentes elementos:

- Procedimientos Almacenados
- Tipos de Datos de Usuario
- Funciones de Usuario
- Agregados de Usuario
- Desencadenadores (Triggers)
- Clases

Estas librerías (.DLL) contienen módulos (**Ensamblados**) de **Código Objeto** llamado código **MSIL** (**M**icrosoft **I**ntermediate **L**anguage). Éste es un código Multiplataforma, independiente del Procesador y Sistema Operativo, que requiere ser compilado “*al vuelo*” cuando se quiera ejecutar. Para esta tarea, SQL Server incluye en su núcleo el **compilador JIT** (**J**ust **i**n-**t**ime), que es la base del **CLR** (**C**ommon **L**anguage **R**untime – Entorno Común de Ejecución). El CLR es el núcleo de la plataforma .NET

El propio Visual Estudio se encarga, después de compilado el proyecto, de implementarlo en la BB.DD. que le hayamos dicho. Pero si queremos implementar (Agregar y Registrar) un Ensamblado que ya exista con todos sus elementos, deberemos realizar los siguientes pasos en la BB.DD. de SQL Server deseada:

### 1.- Agregar o Crear nuevo Ensamblado

Agregarlo:

Picamos en: **BBDD - Programación - Ensamblados - Botón Derecho - Nuevo Ensamblado...**  
Picamos el botón **Examinar...** y seleccionamos el <fichero.DLL> en la ubicación donde lo tengamos.

Otra posibilidad es Crearlo con Transact-SQL:

```
CREATE ASSEMBLY [Nombre_SqlServerProject]
AUTHORIZATION [dbo]
FROM 'Ruta\fichero.DLL'
WITH PERMISSION_SET = SAFE
```

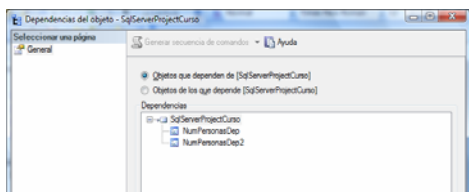
Si ahora comprobamos las dependencias del Ensamblado, veremos que todavía no tiene ninguna:

Picamos en: **BBDD - Programación - Ensamblados - <Nuestro Ensamblado> - Botón Derecho - Ver Dependencias**

### 2.- Registrar los Procedimientos Almacenados con Transact-SQL:

```
CREATE PROCEDURE [dbo].[NumPersonasDep]
    @CodigoDepartamento [nvarchar](3)
WITH EXECUTE AS CALLER
AS EXTERNAL NAME
[Nombre_SqlServerProject].[StoredProcedures].[NumPersonasDep]
```

Una CREATE por cada Procedimiento / Función del Ensamblado.



Si ahora comprobamos de nuevo las dependencias del Ensamblado, veremos que ya tiene:

Picamos en: **BBDD - Programación - Ensamblados - <Nuestro Ensamblado> - Botón Derecho - Ver Dependencias**

Para ver los nuevos Proc. Almacenados tenemos que actualizar con **Botón Derecho – Actualizar**

Si ahora probamos en una nueva Consulta los procedimientos:

```
EXECUTE NumPersonasDep2 MKT
GO
```

Nos informará de que no tenemos habilitada la ejecución de código de usuario en .NET Framework:

La ejecución de código de usuario en .NET Framework está deshabilitada. Habilite la opción de configuración "clr enabled".

### 3.- Habilitar la Integración CLR (ejecución de código de usuario en .NET Framework)

**NOTA:** Este paso sólo hará falta hacerlo una vez en cada servidor de SQL Server.

En una Consulta ejecutamos: `sp_configure 'clr enabled', 1`  
Y nos contestará con el mensaje:

Se ha cambiado la opción de configuración 'clr enabled' de 0 a 1. Ejecute la instrucción RECONFIGURE para instalar.

En la Consulta ejecutamos: `RECONFIGURE`

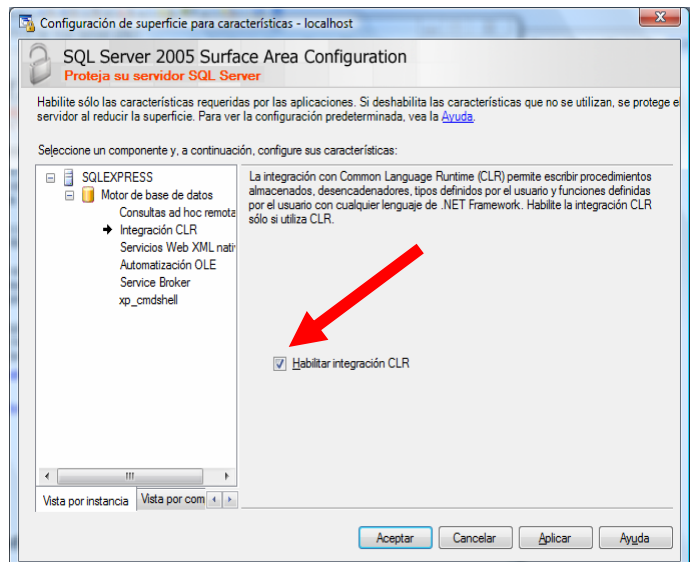
Ahora ya podemos usar los Procedimientos / Funciones que tengamos registrados.

### 3.bis.- También podemos habilitar la Integración CLR con la herramienta gráfica

Abrimos la herramienta de "Configuración de Superficie de SQL Server"

Elegimos "Configuración de Superficie para Características"

En la opción "Integración CLR" activamos el check "Habilitar integración CLR"



## Código en C# para crear el ensamblado con los Procedimientos Almacenados

Con este código en lenguaje C# creamos un Procedimiento Almacenado “NumPersonasDep2” al que pasaremos como único argumento un código de Departamento “CodigoDepartamento”.

El Procedimiento nos informará del número de Personas adscritas a ese Departamento así como el Nombre del Departamento.

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Data.SqlTypes;
using Microsoft.SqlServer.Server;

public partial class StoredProcedures
{
    [Microsoft.SqlServer.Server.SqlProcedure()]
    public static void NumPersonasDep2(String CodigoDepartamento)
    {
        //Establecemos una Conexión con la BBDD Actual
        SqlConnection Conexion = new SqlConnection("context connection=true");

        //Para Crear un Comando
        SqlCommand Consulta1 = Conexion.CreateCommand();
        SqlCommand Consulta2 = Conexion.CreateCommand();

        //Las consultas en SQL (Comandos)
        Consulta1.CommandText = "SELECT COUNT(perCod) FROM Personal WHERE perDep='" +
            CodigoDepartamento + "'";

        Consulta2.CommandText = "SELECT depNom FROM Departamentos WHERE depCod='" +
            CodigoDepartamento + "'";

        //Abrimos la Conexión
        Conexion.Open();

        //Ejecutamos los Comandos
        int NumFilas = (int) Consulta1.ExecuteScalar();
        string NombreDep = (string)Consulta2.ExecuteScalar();

        //Cerramos la Conexión
        Conexion.Close();

        //Preparamos una Fila de Datos con la información que queremos devolver
        SqlDataRecord Resultado = new SqlDataRecord(
            new SqlMetaData("Numero Personas", SqlDbType.Int),
            new SqlMetaData("Departamento", SqlDbType.Text));

        Resultado.SetInt32(0, NumFilas);
        Resultado.SetSqlString(1, NombreDep);

        //Enviamos los resultados como parámetro de salida del Proc. Almacenado
        SqlContext.Pipe.SendResultsStart(Resultado);
        SqlContext.Pipe.SendResultsRow(Resultado);
        SqlContext.Pipe.SendResultsEnd();
    }
};
```