

[Home](#)[Publicaciones y Cursos](#)[CompartiMOSS](#)[Acerca de Mí](#)[Contacto](#)

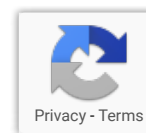
Entity Framework: Como solucionar los problemas en la carga de la metadata!

By Juan Carlos González Jul-17-2014 Sin categoría 4 Comments.

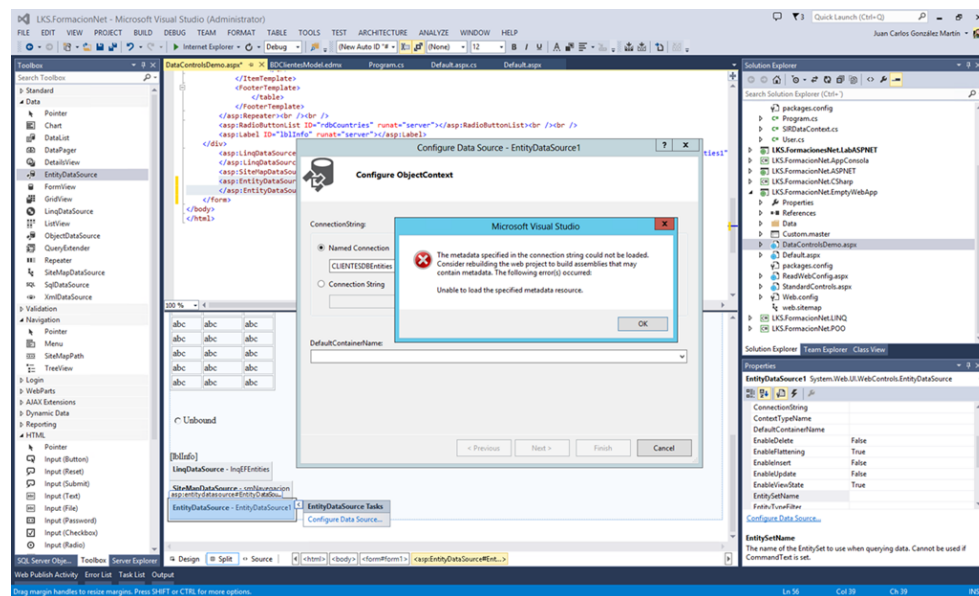
Hacía tiempo que no “jugaba” con las últimas versiones de Entity Framework y esta tarde mientras estaba usando un control de tipo EntityDataSource me encontré con que al intentar usar el EDM de EF que tenía definido en el proyecto de Visual Studio el asistente de configuración de la fuente de datos



julio 2014						
L	M	X	J	V	S	D
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
« Jun				Ago »		



subyacente me soltaba un error la mar de majo: ***"The metadata specified in the connection string could not be loaded. Consider rebuilding the web project to build assemblies that may contain metadata. The following error(s) occurred: Unable to load the specified metadata resource"***. Lógicamente, tras decir WTF me puse a tratar de solucionar el asunto porque necesitaba poder configurar ese EntityDataSource para una pequeña prueba de concepto...



Tras revisar que las cadenas de conexión estaban correctas en el Web.Config y que el problema no se debía al modelo de EF generado en Visual Studio 2013, me puse a bucear por la red en busca de este error y similares, y por suerte (y también por

L	M	X	J	V	S	D
---	---	---	---	---	---	---

Categorías

Azure Azure AD

CompartiMOSS Comunidades

Técnicas Dynamics 365 **Eventos**

Herramientas Hyper-V Internet

Explorer Libros Microsoft Flow

Office 365 Office 2013

Office 2016 OneDrive Power Apps

Product Reviews Recursos

SharePoint SharePoint

2010 SharePoint

2013 sharepoint

2016 SharePoint

Foundation 2010

SharePoint

Foundation 2013

SharePoint

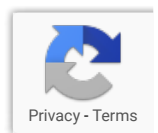
Online SQL Server SQL

Server 2016 Visual Studio Visual

Studio 2013 Visual Studio 2015 Visual

Studio Online Windows 10 Windows

Server Windows Server 2012 R2



desgracia), me encontré con varias entradas reportando el problema para versiones anteriores a la 6.0 de EF...tras varias pruebas sin resultados, finalmente encontré un par de referencias que me llevaron a la solución al problema

-

<http://social.msdn.microsoft.com/Forums/en-US/577ccb2d-1340-4191-b484-3f66ea4c8f07/vs-2013-and-unable-to-load-the-specified-metadata-resource?forum=adodotnetentityframework>

-

<http://social.msdn.microsoft.com/Forums/en-US/8c428649-5b4d-4404-856f-f72dee178bc9/providermanifesttoken2012-in-edmx-causes-metadata-load-errors?forum>

En particular, el segundo enlace es el que contiene la descripción del problema y la solución dada por Diego de La Vega de Microsoft:

Certainly, the EntityDataSource that is part of .NET framework does not support EF6. When you get the error it is because the control is trying to load the model in the EF5 runtime, which does not support 2012 as the value for ProviderManifestToken. We have an updated version of the EntityDataSource in the works that supports the EF6 runtime, and the current plan is to make it available in a NuGet package.

-

Archivos

» abril 2017

» marzo 2017

» febrero 2017

» enero 2017

» diciembre 2016

» noviembre 2016

» octubre 2016

» septiembre 2016

» agosto 2016

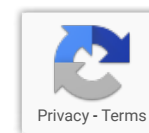
» julio 2016

» junio 2016

» mayo 2016

» abril 2016

» marzo 2016



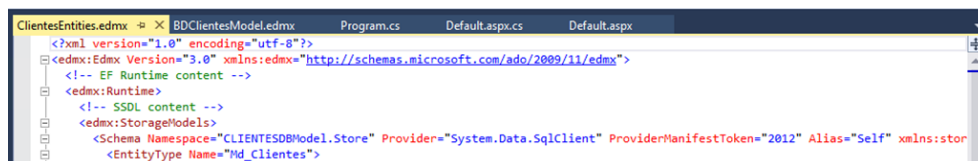
In the meanwhile, it seems that there are two workarounds: – Editing your edmx file and replacing 2012 with 2008 then using EF5 instead of EF6

-

Using model binding for WebForms instead of a DataSource control.

Tras esto, me puse manos a la obra para probar si alguno de los dos workaround funcionaba empezando por el primero:

- En Visual Studio 2013, edité el archivo .edmx de mi modelo y efectivamente, aparece la propiedad ProverManifestToken con el valor 2012.



```
<?xml version="1.0" encoding="utf-8"?>
<edmx:Edmx Version="3.0" xmlns:edmx="http://schemas.microsoft.com/ado/2009/11/edmx">
  <!-- EF Runtime content -->
  <edmx:Runtime>
    <!-- SSDL content -->
    <edmx:StorageModels>
      <Schema Namespace="CLIENTESDBModel.Store" Provider="System.Data.SqlClient" ProviderManifestToken="2012" Alias="Self" xmlns:stor=
      <EntityType Name="Md_Clientes">
```

-

Tras cambiar el valor de ese atributo de 2012 a 2008 y recompilar el proyecto, ya pude configurar sin problemas el control EntityDataSource y usarlo en mi POC...por lo que con el primer workaround, se soluciona el problema.

» febrero 2016

» enero 2016

» diciembre 2015

» noviembre 2015

» octubre 2015

» septiembre 2015

» agosto 2015

» julio 2015

» junio 2015

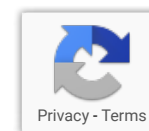
» mayo 2015

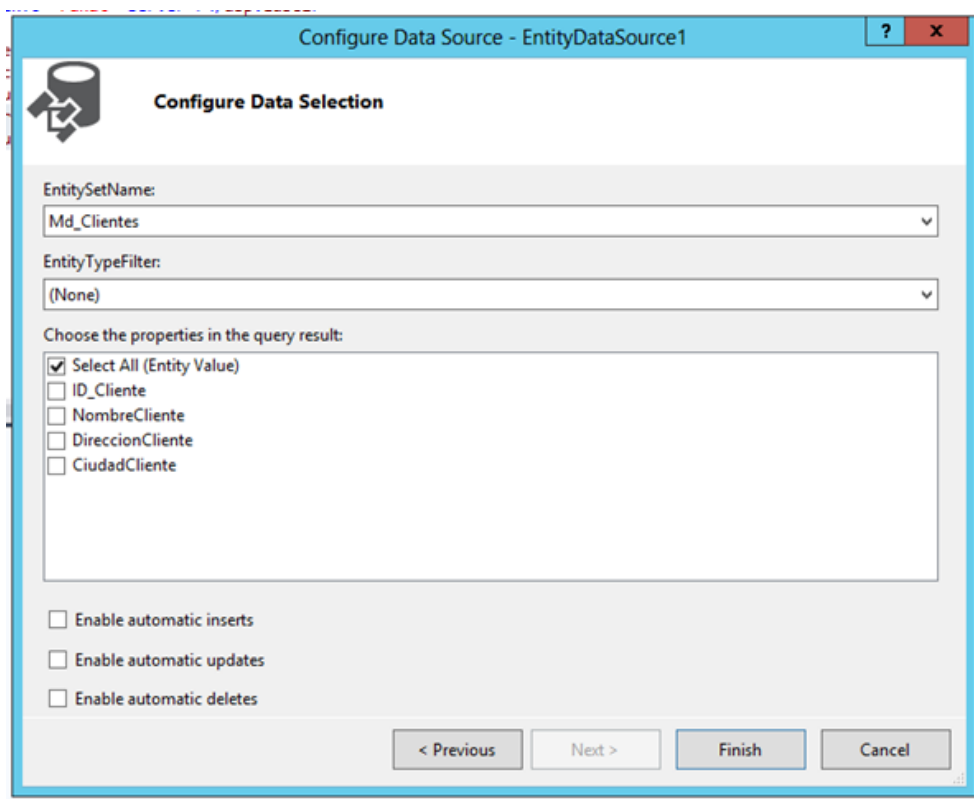
» abril 2015

» marzo 2015

» febrero 2015

» enero 2015



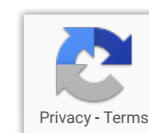


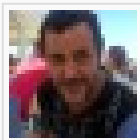
De acuerdo al mismo mensaje de Diego de La Vega, se estaba trabajando en una nueva versión de EntityDataSource que solucione este problema...desconozco si la nueva versión está disponible o no.

Comments

4 Responsesso far

- » diciembre 2014
- » noviembre 2014
- » octubre 2014
- » septiembre 2014
- » agosto 2014
- » julio 2014
- » junio 2014
- » mayo 2014
- » abril 2014
- » marzo 2014
- » febrero 2014
- » enero 2014
- » diciembre 2013
- » noviembre 2013
- » octubre 2013

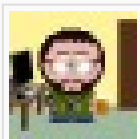




Pedrohurtado dice:
18 julio, 2014 a las 13:44

La verdad que merece la pena tu esfuerzo, pero mi pregunta a estas alturas es para que? Creo que EF es un moribundo y que tras muchos años incordiando está cerca el día de su entierro definitivo. Nos quejamos durante tiempo de DataTable y demás y esto con diferencia es el peor de todos los intentos. Te invito que imagino que ya lo has hecho a leer este post de mi compañero Xavi <http://geeks.ms/blogs/xavipaper/archive/2014/07/18/mythbuster-velocidad-de-acceso-a-datos.aspx>. Seguro que alguno me puede decir pero es que desarrollo más rápido, si pero una prueba de concepto, porque una app de verdad ya te digo yo que no:)

Responder



Jcgonzalez dice:
18 julio, 2014 a las 13:53

Buenas Pedro,
En mi caso para lo que necesitaba tener esto funcionando si me merece la pena para una formación interna que tengo que dar la semana que viene...la verdad es que no tengo claro si EF está condenado a desaparecer o no...parecer parece que no porque Microsoft lo sigue evolucionando y dotando de más capacidad con cada nueva reléase. En el caso de mi empresa, EF lo usamos mucho porque nos facilita

» septiembre 2013

» agosto 2013

» julio 2013

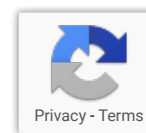
Meta

» Acceder

» [RSS de las entradas](#)

» [RSS de los comentarios](#)

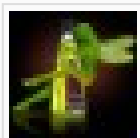
» [WordPress.org](#)



mucho el desarrollo...y efectivamente, puede suceder que en ciertos proyectos no se pueda usar porque limitaciones / problemas / Etc.

Un saludo

Responder



Etomas dice:

18 julio, 2014 a las 13:55

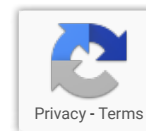
La verdad es que EF tiene sus “cosas” pero muchas de ellas se solucionan usando Code First y olvidándose del EDMX. La verdad es que toda la visión de EF como un proveedor adicional de datos se ha visto muy limitada.

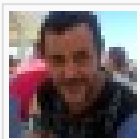
Sobre el tema de carga de datos... pues estamos en lo de siempre. Si solo vas a cargar datos, olvídate de un ORM. Para qué lo quieres? Si es para ayudarte con 4 mapeos básicos estás matando moscas a cañonazos: usa un micro ORM o ADO.NET a pelo.

Pero cuando tienes realmente un modelo de entidades de verdad es cuando aprecias un ORM (ya sea EF o NHibernate). Valorar un ORM solo por la velocidad en la carga de datos me parece que es errar el tiro y por mucho

Saludos!

Responder





Pedrohurtado dice:
19 julio, 2014 a las 01:02

Hola Eduard,

No estoy en nada de acuerdo en que CodeFirst no utilice edmx, es más lo hace aún peor.

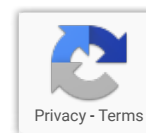
1. Genera un edmx y lo guarda en la tabla dbo.__MigrationHistory
2. Se guarda una columna como VarBinary(max) creo y no es otra cosa que un Xml comprimido pasado a base64.
3. Cada vez que accedes por primera vez recupera el modelo y lo vuelve a pasar a Xml

Así que amigo Eduard lo veo peor que sacarlo de un Metadata/Resource de una dll.

Con lo cual creo que ahora CodeFirst/ModelFirst/DataBaseFirst utilizan lo mismo, y el más costoso para recuperar el modelo es CodeFirst, puesto que tiene que acceder a bb.dd para eso.

En estos pos creo que lo dejo claro.

<http://geeks.ms/blogs/phurtado/archive/2012/06/29/desmitificando-codefirst-1-2.aspx>



<http://geeks.ms/blogs/phurtado/archive/2012/06/29/desmitificando-codefirst-2-2-v-4-3.aspx>

Y desde la versión 4.3 no ha evolucionado, es más para peor con la introducción de Migrations.

Dicho todo esto, tengo claro que en un ORM jamás voy a medir velocidad, sino más bien productividad y si coincido que un ORM me hace ser más productivo en determinados sitios.

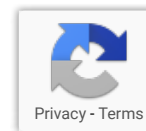
El problema no es otro que cuando me dicen que un ORM es para todo y es en ese punto donde se ha intentado vender #EF como la solución de acceso a datos, hasta que la gente se ha dado cuenta.

Problema que como siempre somos pocos los que lo dejamos tan claro y los demás y ojo no os incluyo a ninguno de los dos. Conociendo el problema sigue en el más de los profundos silencios.

Gracias!!!

Responder

Deja un comentario



Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

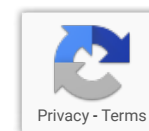
Comentario

Nombre *

Correo electrónico *

Web

Publicar comentario



[← Previous Post](#)

[Next Post →](#)

About This Site

A cras tincidunt, ut tellus et. Gravida scel ipsum sed iaculis, nunc non nam. Placerat sed phase llus, purus purus elit.

Archives Widget

- [January 2010](#)
- [December 2009](#)
- [November 2009](#)
- [October 2009](#)

Categories

- [Entertainment](#)
- [Technology](#)
- [Sports & Recreation](#)
- [Jobs & Lifestyle](#)

Search



Powered by WordPress | Business Directory by InkThemes.

