Contents

Create a new Entity Data Model (EDM) with the Entity Framework's **EDM Wizard**

DevForce developers begin entity model development by defining an Entity Framework Entity Data Model (EDM). You create a new EDM using the **Entity Framework EDM Wizard**.

In this topic we show most of what you need to use the EDM Wizard but it's also worth looking at Microsoft's documentation in MSDN

Pick the model project

The first question it "where do I put the model?". You have to pick the project that will be home to your entity class model.

The DevForce tutorials teach you to begin with a DevForce n-tier template and add the model to the web application server project. That's a simple way to learn DevForce. Although not ideal in the long term, don't worry about it now; you can break that model into a separate project later if you want to.

More experienced DevForce developers create a dedicated model project up front. Make sure you create the model project as a .NET 4.5 class library.

Either way, you'll add your entity model classes to a **.NET 4.5 project**. Entity models may not be directly defined in Silverlight and mobile projects.

Add an ADO.NET Entity Data Model item

Having decided upon the project that will hold your model, do the following:

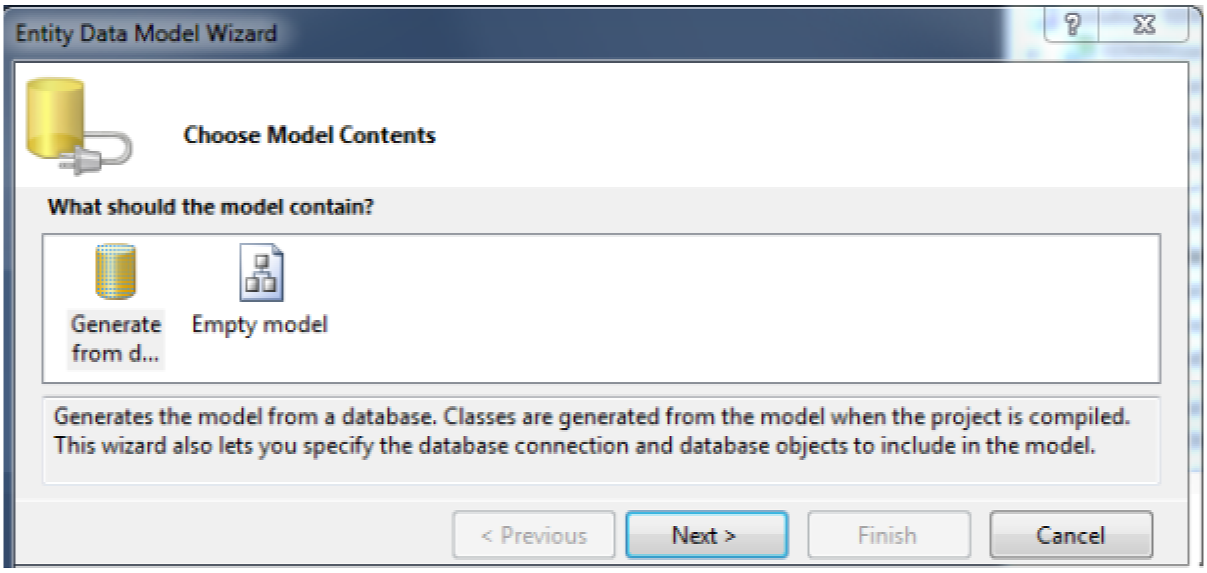
- Select the project in the Visual Studio "Solution Explorer" window
- Select "**Add a new item**" from the context menu (or press **Ctrl-Shift-A**)
- Select the "**ADO.NET Entity Data Model**" template under the Data node
- Enter a name for the model in the "Name:" box (e.g., *DomainModel.edmx*)
- Press **Enter**

You have launched the "**Entity Data Model Wizard**".

Note that the template added the *.edmx* file extension to your model name; you picked the wrong template if it didn't.

The wizard offers two ways forward:

1. "Generate from database" - the **Data First** approach
2. "Empty model" - the **Model First** approach



Generates the model from a database. Classes are generated from the model when the project is compiled. This wizard also lets you specify the database connection and database objects to include in the model.

Data, Model, or Code First?

Do you want to generate the entity model from an existing database? Choose #1 and follow the "Data First" path. Most developers go this way, either because they are committed to a legacy database or because they prefer to define the database and the entity model at the same time.

You might prefer #2 - "Model First" - if you are building the application from scratch, have no legacy database to worry about, and want to delay defining a database until you've worked out your entity model.

Both approaches produce a **Conceptual Entity Model** that defines the shapes and relationships among the entities that will eventually become your entity classes.

"Data First" produces a full EDM that maps the Conceptual Model to the Storage Model. The Storage Model describes the database schema; the mapping correlates the Conceptual Entities with tables and columns in storage. This full EDM is required by the Entity Framework to read and write to the database.

In this topic we consider only the "Data First" and "Model First" approaches. The Entity Framework "Code First" does not use an EDMX at all. See the Code First documentation for how to create a DevForce Code First model.

The "Model First" approach yields an EDMX without mapping and storage descriptions. You can add the Storage Model and Mapping later but until you do you won't be able to access a database with this EDM.

DevForce is OK with that. DevForce only needs the Conceptual Model to generate entity classes. You can develop and test your application quite effectively using an offline entity cache or a fake backing store populated with dummy data.

When you choose "Empty model", the wizard generates a skeletal EDMX file and sends you straight to the EDM Designer with a blank canvas. The EDM wizard has done its job. You proceed to manually add entities and associations to the Conceptual Model.

Create the model from the database

In the "Data First" approach, you point the Wizard at an existing database and tell it to generate entities corresponding to selected tables and views. This is by far the most common approach to entity modeling.

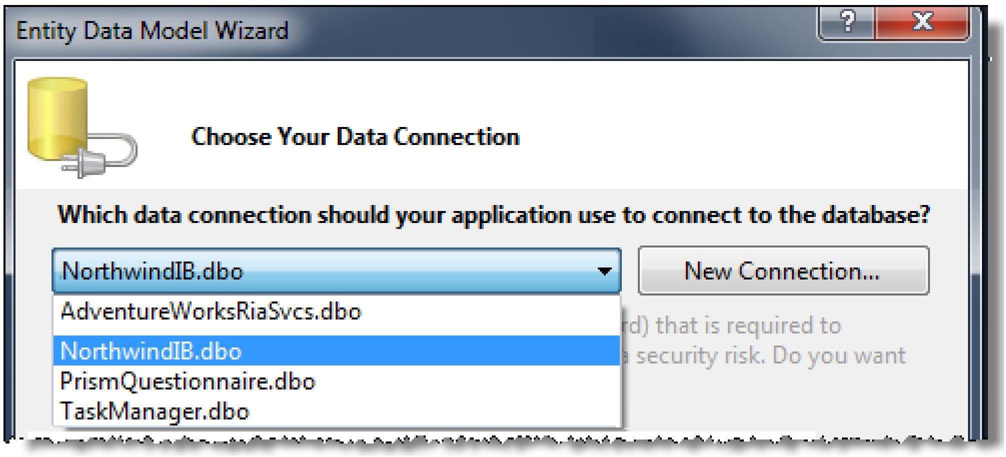
Identify the database

First you must identify the database that will be the schema source.

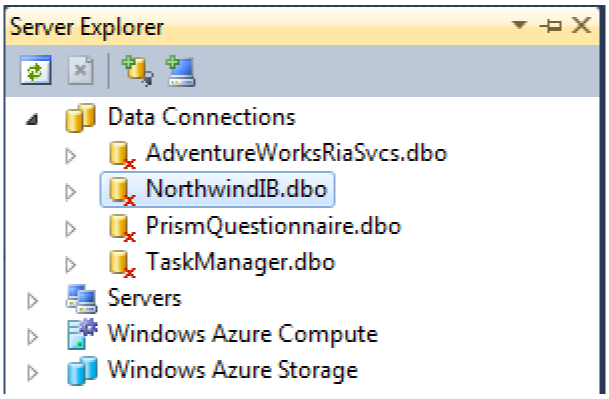
This needn't be the production database; in fact it should **not** be. Pick a version of the database with a suitable schema; it should be a database you can read to and write from during development.

Many developers are comfortable creating and updating a design database using other tools such as SQL Server Management Studio (SMS). They alternate between the EDM Designer and SMS as they evolve the model.

The wizard asks you to pick a database connection as in this example:

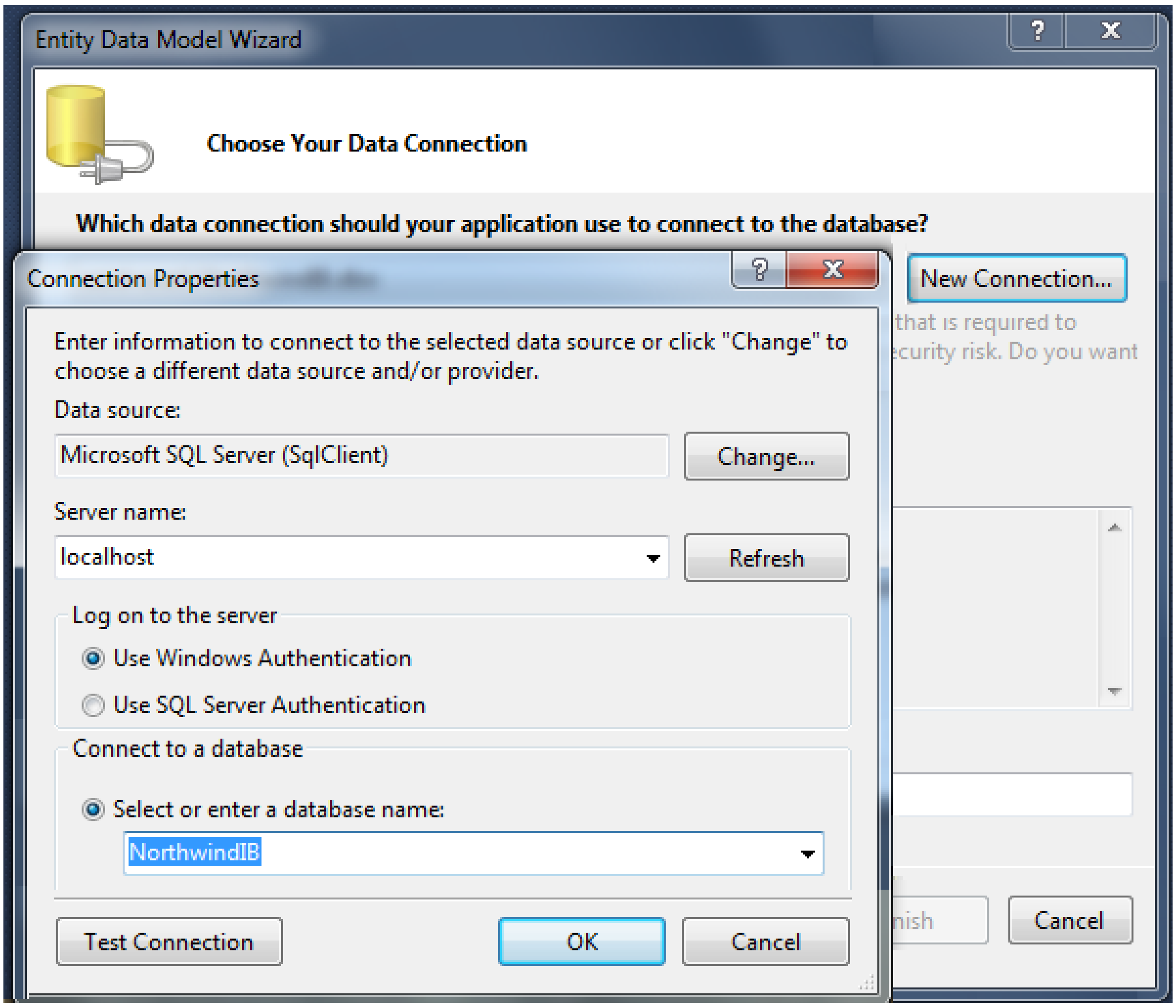

Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
NorthwindIB.dbo
New Connection...
AdventureWorksRiaSvcs.dbo
NorthwindIB.dbo
PrismQuestionnaire.dbo
TaskManager.dbo
d) that is required to
security risk. Do you want

Your target database will be in the drop-down list if you've previously created a connection to it in Visual Studio. The "Server Explorer" window reveals the connections known to Visual Studio at the moment:



You can remove an unwanted connection from the "Server Explorer" - and from the EDM Wizard connection list - by selecting it and pressing the **Delete** key.

If the target database is not in the list, press the **New Connection...** button to open the Visual Studio "Connection Properties" dialog. This screenshot shows the default connection in SQL Server for the "NorthwindIB" tutorial database as it was installed with DevForce.


Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

that is required to

curity risk. Do you want

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

Server name: localhost

Log on to the server

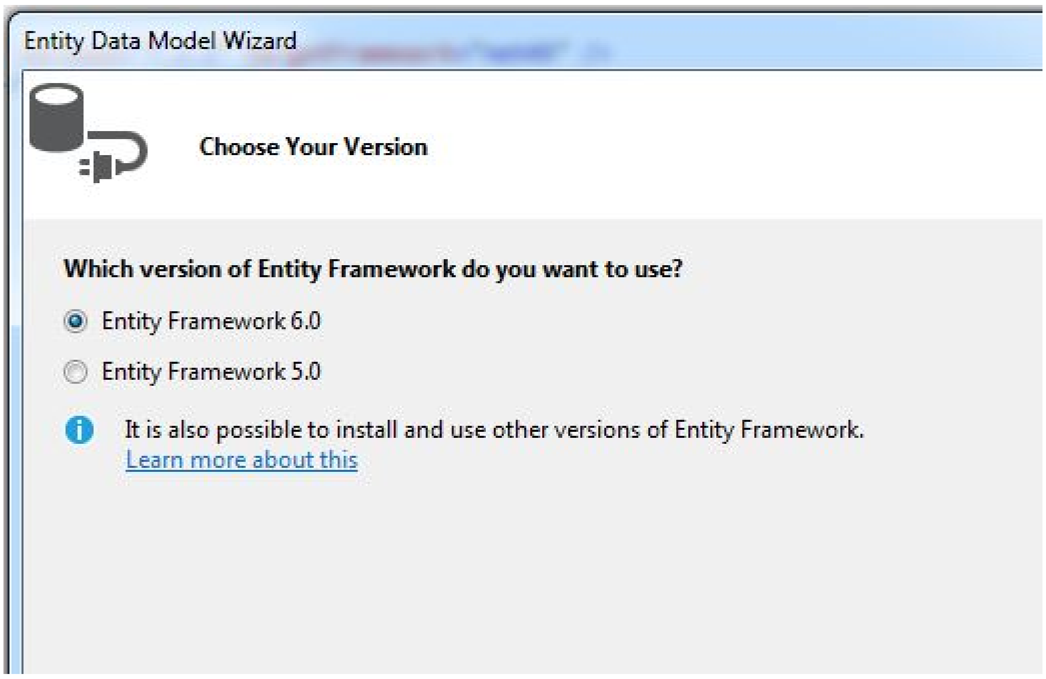
(● Use Windows Authentication) ○ Use SQL Server Authentication

Connect to a database

(● Select or enter a database name:) NorthwindIB

Pick Entity Framework version

If using Visual Studio 2013, or the EF 6 Power Tools are installed with VS2012, you'll also be prompted to choose the EF version:


Entity Data Model Wizard

Choose Your Version

Which version of Entity Framework do you want to use?

[✓ Entity Framework 6.0]

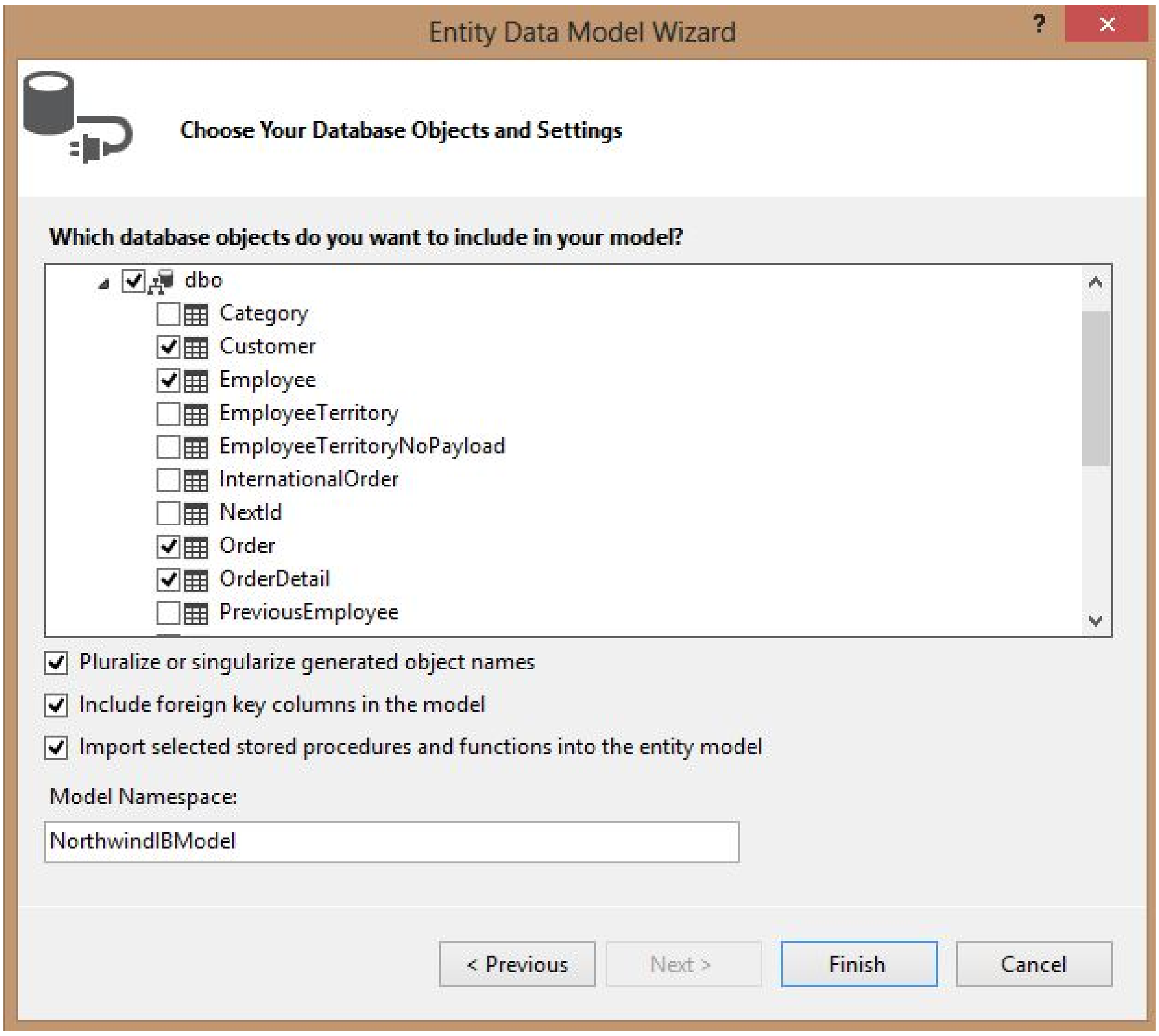
Entity Framework 5.0

It is also possible to install and use other versions of Entity Framework.

As of DevForce version 7.2.2, you may choose either version 5 or 6 for your model. If using an earlier version of DevForce, only EF 5 is supported.

Pick database objects to model

The wizard connects to the database you picked and reads the database schema. This can take from seconds to minutes. Eventually it presents an "Choose objects" dialog with a list of database objects categorized by "Tables", "Views", and "Stored Procedures". In the example shown in the following screenshot, the developer is modeling tables:



Entity Data Model Wizard
Choose Your Database Objects and Settings
Which database objects do you want to include in your model?

- [✓] dbo
  - [ ] Category
  - [✓] Customer
  - [✓] Employee
  - [ ] EmployeeTerritory
  - [ ] EmployeeTerritoryNoPayload
  - [ ] InternationalOrder
  - [ ] NextId
  - [✓] Order
  - [✓] OrderDetail
  - [ ] PreviousEmployee

[✓] Pluralize or singularize generated object names
[✓] Include foreign key columns in the model
[✓] Import selected stored procedures and functions into the entity model

Model Namespace: NorthwindIBModel

< Previous
Next >
Finish
Cancel

Some observations:

1. The "Tables" category shows the database tables that have not yet been modeled. This developer picked four tables.
2. The wizard can use English language pluralization rules to generate entity class and property names. When the pluralization option is checked, the wizard creates a *Customer* entity type mapped to either a "Customer" (singular) or "Customers" (plural) table. The wizard sees that a *Customer* has multiple related orders so it creates a *Customer.Orders* (plural) navigation property. The wizard does a decent job of English pluralization; non-English speakers might turn this off. In any case, you can set these names directly in the designer. You can turn pluralization on and off later as well.
3. You **must check the "Include foreign key columns"** option which tells the EDM designer to create "Foreign Key Associations" and expose foreign key columns as entity properties. DevForce requires this option.
4. The "Model Namespace" value isn't used by DevForce. DevForce generates the entity classes with the project's default namespace ... unless you override that default by setting the "Custom Tool Namespace" property of the DevForce template (the *.tt* file).

When you press **Finish**, the wizard generates the EDM and hands off to the EDM Designer.