

Contents

- [Add entities and associations](#)
- [Foreign Key Associations only](#)
- [Mapping to a Database](#)

This topic introduces techniques for defining a Conceptual Entity Model in the [Model First](#) style using the Visual Studio EDM Designer.

You can define a Conceptual Model for your Entity Data Model (EDM) without referring to a database using the "Model First" approach. DevForce can generate entity classes from the Conceptual Model alone. You can develop and test the application without a database using an [offline entity cache](#) or a [fake backing store](#) populated with dummy data.

Add entities and associations

You begin "Model First" development with the [EDM Wizard, choosing the "Empty model" option](#).

Then you add entities and relationships directly to the designer canvas. The details are out-of-scope for DevForce documentation. The techniques involved are covered in the standard Entity Framework literature. You might start with these two resources and search the web for more:

- [Model-First in the Entity Framework 4](#)
- [Creating an Entity Data Model the Model-First Way](#)

You work more extensively with the [Properties window](#) in "Model First" than you do in "Data First". In "Data First" most of the properties are filled-in for you based on database schema information and there is rarely cause to change those values. In "Model First" there is no database schema; you specify the details of every property and association according to your plan for the entity.

Foreign Key Associations only

You must create "[Foreign Key Associations](#)" when adding associations to your model.

"Data First" developers don't worry about this because the EDM Designer, once properly configured, ensures that all associations added from the database are "Foreign Key" associations.

A "Model First" developer must be alert to the issue potential for the EDM Designer to create an "Independent" association unexpectedly.

The designer creates an "Independent" association if you *uncheck* the "Add foreign key properties..." CheckBox in the "Add Association" dialog, something you might do if the foreign key property happened to be already defined.

You must convert an "Independent" association to a "Foreign Key" association by adding a [Referential Constraint](#).

Mapping to a Database

When you are ready to associate the application with a real database, you map the Conceptual Model to that database. Right-mouse-click on the canvas after which you select "Generate Database from Model...". The resources cited above walk you through the process.

The EDM Designer doesn't actually generate the database. It produces a "DDL script" (Data Definition script) that can generate the database. You can play this script from within Visual Studio or run it separately.

Beware! the generated DDL script deletes all objects from the existing database; all of the database data will be lost. If you are using "Model First" to add new entities and tables to an existing database and you want to use the DDL script, be sure to edit it carefully to ensure it does no harm.

Above all else, **make sure you backup the database before running any scripts!**

Database generation works well in the beginning when you have don't have a real database with real data and are less inclined to care about an optimal database structure. As the application matures you won't want to use that DDL script. You are more likely to create the table in the database in your own way with your own script even if you've separately designed the corresponding entity in your EDM in "Model First" style.

You can still proceed "Model First". You "reconcile" with the database in a different manner.

You don't generate DDL Script. You "**Update Model from Database...**" instead and add the new tables that correspond to the new entities as you would for "Data First". The designer won't recognize the correspondence with your "Model First"

entities. if you had created a *Customer* entity in "Model First" and subsequently added a *Customer* table to your database, the EDM Designer will add that table as *Customer1*.

Use this behavior to your advantage. Compare *Customer* and *Customer1*. When you are convinced that the generated *Customer1* is correct, (1) delete your "Model First" *Customer* entity and (2) rename *Customer1* to *Customer*.

We suggest you proceed deliberately, adding only one or two tables at a time.