Contents

The generated entity model is typically extended with additional business logic such as validation rules, UI hints, calculated properties and workflow methods. This topic introduces some of these **customization techniques**.

Limits of code generation

The generated entity model classes are the product of the DevForce T4 template (the .tt file) and the conceptual entity metadata inscribed in the edmx file. The template prescribes *how* classes and properties are written. The metadata describe *what* classes and properties are written.

The EDM Designer, as extended by DevForce, gives you some control over aspects of the classes and their properties, in particular their accessibility, UI hints and schema-determined validation attributes. But you are limited to describing persistent classes and their data and navigation properties - the values stored in database tables and the pathways to related entities. There is no place in the EDMX to specify constructors, calculations, events, workflow methods or validation rules, no way to define classes or properties that aren't directly translatable to data storage.
You can modify the T4 template (the .tt file) to change how the classes are implemented. That's not an effective instrument for detailing the semantics of individual entity classes. When you need the *Customer* class to be more than a bag of persistent data (*CustomerID*, *CompanyName*, etc.), to have behavior that expresses what it means to be a "customer" in the application domain, you'll extend the *Customer* class with handwritten, custom code.

Customize Entities

You don't customize the entity model by modifying the generated code; you'd lose those changes the next time you ran the code generator.

You cannot extend generated entity classes by subclassing them. Neither DevForce nor Entity Framework will work with derived entity classes.

Instead you customize by writing code in one or all of three places:

1. A partial class file
2. A metadata class file (the "buddy class")
3. A helper class to which the entity delegates, perhaps for validation or property interception.