.NET Framework 4

**What is New in ADO.NET**

The following features are new in ADO.NET with the .NET Framework version 4.

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifADO.NET Entity Framework

The Entity Framework decreases the amount of coding and maintenance required for data-driven applications by enabling you to program against data models defined in terms of entities and relationships. For more information, see [Entity Framework Overview](http://msdn.microsoft.com/en-us/library/bb399567.aspx).

The features described below are new to the Entity Framework in .NET Framework 4.

**Foreign Keys in the Conceptual Model**

You can create conceptual models in which foreign key columns in the database correspond to scalar properties on entity types. For more information, see [Defining and Managing Relationships](http://msdn.microsoft.com/en-us/library/ee373856.aspx).

**Self-Tracking Entities for N-Tier Application Development**

You can now use the self-tracking entities when working with N-tier applications. The self-tracking entities can record changes to scalar, complex, and navigation properties. The tracking information in self-tracking objects could be applied to the object context on the service side. For more information, see [Working with Self-Tracking Entities](http://msdn.microsoft.com/en-us/library/ff407090.aspx).

**New Methods for N-Tier Application Development**

New methods on classes in the [System.Data.Objects](http://msdn.microsoft.com/en-us/library/system.data.objects.aspx) namespace make it easier to develop N-Tier applications. For more information, see [N-Tier Applications With Entity Framework](http://msdn.microsoft.com/en-us/library/bb896304.aspx).

**EntityDataSource Support for the QueryExtender Control**

The [EntityDataSource](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.entitydatasource.aspx) control now supports the [QueryExtender](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.queryextender.aspx) control, which is used to create filters for data retrieved from a data source. For more information, see [Applying LINQ Queries to EntityDataSource](http://msdn.microsoft.com/en-us/library/ee404748.aspx).

**Testability with IObjectSet<T>**

The [ObjectContext](http://msdn.microsoft.com/en-us/library/system.data.objects.objectcontext.aspx) class now returns typed [ObjectSet<(Of <(TEntity>)>)](http://msdn.microsoft.com/en-us/library/dd412719.aspx) objects that you can use to perform create, read, update, and delete operations on entity objects. The [ObjectSet<(Of <(TEntity>)>)](http://msdn.microsoft.com/en-us/library/dd412719.aspx) class implements the [IObjectSet<(Of <(TEntity>)>)](http://msdn.microsoft.com/en-us/library/dd642094.aspx) interface. The [IObjectSet<(Of <(TEntity>)>)](http://msdn.microsoft.com/en-us/library/dd642094.aspx) interface may be useful in testing scenarios. For more information, see [Working with ObjectSet](http://msdn.microsoft.com/en-us/library/ee473442.aspx).

**Direct Execution of Store Commands**

You can now directly execute store commands from the [ObjectContext](http://msdn.microsoft.com/en-us/library/system.data.objects.objectcontext.aspx). For more information, see [Directly Executing Store Commands](http://msdn.microsoft.com/en-us/library/ee358769.aspx).

**Persistence-Ignorant Objects**

You can use your own custom data classes together with your data model without making any modifications to the data classes themselves. This means that you can use "plain old" CLR objects (POCO), such as existing domain objects, with your Entity Framework application. For more information, see [Persistence Ignorant Objects (Entity Framework)](http://msdn.microsoft.com/en-us/library/dd456853.aspx).

**Lazy Loading of Related Objects**

With lazy loading, also known as *deferred loading*, related objects are automatically loaded from the data source when you access a navigation property. For more information, see [Shaping Query Results (Entity Framework)](http://msdn.microsoft.com/en-us/library/bb896272.aspx).

**Functions in LINQ to Entities Queries**

The [EntityFunctions](http://msdn.microsoft.com/en-us/library/system.data.objects.entityfunctions.aspx) and [SqlFunctions](http://msdn.microsoft.com/en-us/library/system.data.objects.sqlclient.sqlfunctions.aspx) classes provide access to canonical and database functions from LINQ to Entities queries. The [EdmFunctionAttribute](http://msdn.microsoft.com/en-us/library/system.data.objects.dataclasses.edmfunctionattribute.aspx) allows a CLR method to serve as a proxy for a function defined in the conceptual model or storage model. For more information, see [Calling Functions in LINQ to Entities Queries](http://msdn.microsoft.com/en-us/library/dd456828.aspx).

**OrderBy Improvements in LINQ to Entities**

LINQ to Entities queries that contain an **OrderBy** operator now return results ordered by the argument specified in the operator, regardless of the position of the operator in the query. However, the **OrderBy** operator in Entity SQL and [ObjectQuery<(Of <(T>)>)](http://msdn.microsoft.com/en-us/library/bb345303.aspx) queries is ignored if it is not the last operator in the query.

**Customized Object-Layer Code Generation**

You can configure the [ADO.NET Entity Data Model Designer](http://msdn.microsoft.com/en-us/library/cc716685.aspx) to use text templates to generate customized object-layer code. For more information, see [How to: Customize Object-Layer Code Generation](http://msdn.microsoft.com/en-us/library/dd456821.aspx).

**Model-First Support**

The [Generate Database Wizard](http://msdn.microsoft.com/en-us/library/dd456817.aspx) enables you to do conceptual modeling first, and then create a database that supports the model. For more information, see [How to: Generate a Database from a Conceptual Model](http://msdn.microsoft.com/en-us/library/dd456815.aspx).

**Complex Type Support**

The [ADO.NET Entity Data Model Designer](http://msdn.microsoft.com/en-us/library/cc716685.aspx) now supports complex types. For more information, see the following topics:

* [How to: Create and Modify Complex Types](http://msdn.microsoft.com/en-us/library/dd456820.aspx)
* [How to: Add a Complex Type to an Entity Type](http://msdn.microsoft.com/en-us/library/dd456823.aspx)
* [How to: Map a Function Import to a Complex Type](http://msdn.microsoft.com/en-us/library/dd456824.aspx)
* [How to: Map Complex Type Properties to Table Columns](http://msdn.microsoft.com/en-us/library/dd456822.aspx)

**Naming Service**

The [Entity Data Model Wizard](http://msdn.microsoft.com/en-us/library/bb399247.aspx) and the [Update Model Wizard](http://msdn.microsoft.com/en-us/library/cc716705.aspx) provide the option of using singular or plural forms of **Entity**, **EntitySet**, and **NavigationProperty** names to make application code more readable. For more information, see [Choose Your Database Objects Dialog Box (Entity Data Model Wizard)](http://msdn.microsoft.com/en-us/library/bb399253.aspx) and [Choose Your Database Objects Dialog Box (Update Model Wizard)](http://msdn.microsoft.com/en-us/library/cc716689.aspx).

**Improved Model Browser Functionality**

The **Model Browser** window of the [ADO.NET Entity Data Model Designer](http://msdn.microsoft.com/en-us/library/cc716685.aspx) enables you to delete objects from the storage model and to search the conceptual and storage models for a specified string. For more information, see [Model Browser Window](http://msdn.microsoft.com/en-us/library/bb738483.aspx) and [How to: Delete Objects from the Storage Model](http://msdn.microsoft.com/en-us/library/dd456816.aspx).

**Entity Designer Extensibility**

You can extend the functionality of the [ADO.NET Entity Data Model Tools](http://msdn.microsoft.com/en-us/library/bb399249.aspx) by using classes in the Microsoft.Data.Entity.Design.Extensibility namespace to write Visual Studio extensions. Classes in the Microsoft.Data.Entity.Design.Extensibility namespace allow you to do the following:

* Extend the .edmx file generation process.
* Extend the .edmx file update process.
* Add custom properties to objects in the [Entity Designer](http://msdn.microsoft.com/en-us/library/cc716685.aspx) and [Model Browser](http://msdn.microsoft.com/en-us/library/bb738483.aspx).
* Load and save custom file formats.
* Extend the .edmx file loading and saving processes.

For more information, see [Extending the Entity Data Model Tools](http://msdn.microsoft.com/en-us/library/ee373852.aspx).

**Entity Data Model Documentation**

The documentation for the Entity Data Model has been updated. For more information, see [Entity Data Model](http://msdn.microsoft.com/en-us/library/ee382825.aspx).