

[Home](#)[Next >](#)

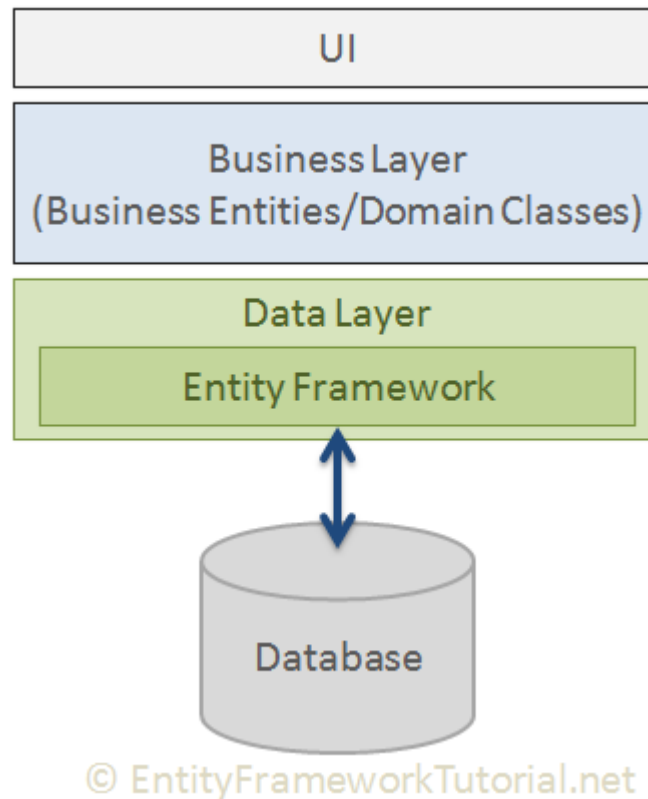
What is Entity Framework?

Prior to .NET 3.5, we (developers) often used to write ADO.NET code or Enterprise Data Access Block to save or retrieve application data from the underlying database. We used to open a connection to the database, create a DataSet to fetch or submit the data to the database, convert data from the DataSet to .NET objects or vice-versa to apply business rules. This was a cumbersome and error prone process. Microsoft has provided a framework called "Entity Framework" to automate all these database related activities for your application.

Entity Framework is an open-source [ORM framework](#) for .NET applications supported by Microsoft. It enables developers to work with data using objects of domain specific classes without focusing on the underlying database tables and columns where this data is stored. With the Entity Framework, developers can work at a higher level of abstraction when they deal with data, and can create and maintain data-oriented applications with less code compared with traditional applications.

Official Definition: "Entity Framework is an object-relational mapper (O/RM) that enables .NET developers to work with a database using .NET objects. It eliminates the need for most of the data-access code that developers usually need to write."

The following figure illustrates where the Entity Framework fits into your application.



As per the above figure, Entity Framework fits between the business entities (domain classes) and the database. It saves data stored in the properties of business entities and also retrieves data from the database and converts it to business entities objects automatically.

Entity Framework Features

- **Cross-platform:** EF Core is a cross-platform framework which can run on Windows, Linux and Mac.
- **Modelling:** EF (Entity Framework) creates an EDM (Entity Data Model) based on POCO (Plain Old CLR Object) entities with get/set properties of different data types. It uses this model when querying or saving entity data to the underlying database.

- **Querying:** EF allows us to use LINQ queries (C#/VB.NET) to retrieve data from the underlying database. The database provider will translate this LINQ queries to the database-specific query language (e.g. SQL for a relational database). EF also allows us to execute raw SQL queries directly to the database.
- **Change Tracking:** EF keeps track of changes occurred to instances of your entities (Property values) which need to be submitted to the database.
- **Saving:** EF executes INSERT, UPDATE, and DELETE commands to the database based on the changes occurred to your entities when you call the `SaveChanges()` method. EF also provides the asynchronous `SaveChangesAsync()` method.
- **Concurrency:** EF uses Optimistic Concurrency by default to protect overwriting changes made by another user since data was fetched from the database.
- **Transactions:** EF performs automatic transaction management while querying or saving data. It also provides options to customize transaction management.
- **Caching:** EF includes first level of caching out of the box. So, repeated querying will return data from the cache instead of hitting the database.
- **Built-in Conventions:** EF follows conventions over the configuration programming pattern, and includes a set of default rules which automatically configure the EF model.
- **Configurations:** EF allows us to configure the EF model by using data annotation attributes or Fluent API to override default conventions.
- **Migrations:** EF provides a set of migration commands that can be executed on the NuGet Package Manager Console or the Command Line Interface to create or manage underlying database Schema.

Entity Framework Latest Versions

Microsoft introduced Entity Framework in 2008 with .NET Framework 3.5. Since then, it released many versions of Entity Framework. Currently, there are two latest versions of Entity Framework: EF 6 and EF Core. The following table lists important difference between EF 6 and EF Core.

EF 6	EF Core
✓ First released in 2008 with .NET Framework 3.5 SP1	✓ First released in June 2016 with .NET Core 1.0
✓ Stable and feature rich	✓ New and evolving
✓ Windows only	✓ Windows, Linux, OSX
✓ Works on .NET Framework 3.5+	✓ Works on .NET Framework 4.5+ and .NET Core
✓ Open-source	✓ Open-source

EF 6 Version History

EF Version	Release Year	.NET Framework
EF 6	2013	.NET 4.0 & .NET 4.5, VS 2012
EF 5	2012	.NET 4.0, VS 2012
EF 4.3	2011	.NET 4.0, VS 2012
EF 4.0	2010	.NET 4.0, VS 2010
EF 1.0 (or 3.5)	2008	.NET 3.5 SP1, VS 2008

Learn more about EF 6 versions history and its features [here](https://www.entityframeworktutorial.net/what-is-entityframework.aspx).

EF Core Version History

EF Core Version	Release Date	.NET Framework
EF Core 2.0	August 2017	.NET Core 2.0, VS 2017

EF Core Version	Release Date	.NET Framework
EF Core 1.1	November 2016	.NET Core 1.1
EF Core 1.0	June 2016	.NET Core 1.0

Learn about the basic workflow while working with Entity Framework in the next chapter.

[Home](#)[Next >](#)

ENTITYFRAMEWORKTUTORIAL

Learn Entity Framework using simple yet practical examples on EntityFrameworkTutorial.net for free. Learn Entity Framework DB-First, Code-First and EF Core step by step. While using this site, you agree to have read and accepted our terms of use and privacy policy.

TUTORIALS

[› EF Basics](#)[› EF Core](#)[› EF 6 DB-First](#)[› EF 6 Code-First](#)

E-MAIL LIST

Subscribe to EntityFrameworkTutorial email list and get EF 6 and EF Core Cheat Sheets, latest updates, tips & tricks about Entity Framework to your inbox.

We respect your privacy.

[HOME](#) [PRIVACY POLICY](#) [ADVERTISE WITH US](#)

© 2020 EntityFrameworkTutorial.net. All Rights Reserved.