



Entity Framework

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Entity framework is an Object Relational Mapping (ORM) framework that offers an automated mechanism to developers for storing and accessing the data in the database. This tutorial covers the features of Entity Framework using Code First approach. It also explains the new features introduced in Entity Framework 6.

Audience

This tutorial is designed for those who want to learn how to start the development of Entity Framework in their application.

Prerequisites

You should have a basic knowledge of Visual Studio, C#, and MS SQL Server to make the most of this tutorial.

Disclaimer & Copyright

© Copyright 2018 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com.

Table of Contents

About the Tutorial.....	i
Audience	i
Prerequisites	i
Table of Contents	ii
1. ENTITY FRAMEWORK — OVERVIEW	1
What is Entity Framework?	1
Why Entity Framework?.....	1
Conceptual Model.....	2
2. ENTITY FRAMEWORK — ARCHITECTURE	5
Data Providers	5
Entity Client	5
Object Service	6
3. ENTITY FRAMEWORK — ENVIRONMENT SETUP	7
What's New in Entity Framework 6?	7
Installation	9
4. ENTITY FRAMEWORK — DATABASE SETUP.....	16
One-to-Many Relationship	16
Many-to-Many Relationship.....	17
One-to-One Relationship.....	17
5. ENTITY FRAMEWORK — ENTITY DATA MODEL.....	18
The Storage Schema Model.....	18
The Conceptual Model	18
The Mapping Model.....	18
Schema Definition Language	19
Entity Type.....	19
	ii

Association Type	20
Property	20
6. ENTITY FRAMEWORK — DBCONTEXT	21
Defining a DbContext Derived Class	22
Queries	23
Adding New Entities	23
Changing Existing Entities.....	23
Deleting Existing Entities	24
7. ENTITY FRAMEWORK — ENTITY TYPES.....	25
Dynamic Proxy	29
8. ENTITY FRAMEWORK — ENTITY RELATIONSHIPS.....	31
One-to-Many Relationship	31
One-to-One Relationship.....	33
9. ENTITY FRAMEWORK — ENTITY LIFECYCLE	35
Lifetime.....	35
Entity Lifecycle	36
State Changes in the Entity Lifecycle	36
10. ENTITY FRAMEWORK — CODE FIRST APPROACH	40
11. ENTITY FRAMEWORK — MODEL FIRST APPROACH.....	42
12. ENTITY FRAMEWORK — DATABASE FIRST APPROACH.....	63
13. ENTITY FRAMEWORK — DEV APPROACHES.....	71
14. ENTITY FRAMEWORK — DATABASE OPERATIONS	73
Create Operation	76
Update Operation	76
Delete Operation	77

Read Operation	77
15. ENTITY FRAMEWORK — CONCURRENCY.....	79
16. ENTITY FRAMEWORK — TRANSACTION.....	85
17. ENTITY FRAMEWORK — VIEWS.....	87
18. ENTITY FRAMEWORK — INDEX.....	95
19. ENTITY FRAMEWORK — STORED PROCEDURES	97
20. ENTITY FRAMEWORK — DISCONNECTED ENTITIES.....	109
21. ENTITY FRAMEWORK — TABLE-VALUED FUNCTION.....	114
22. ENTITY FRAMEWORK — NATIVE SQL.....	125
SQL Query on Existing Entity	125
SQL Query for Non-entity Types	125
SQL Commands to the Database	126
23. ENTITY FRAMEWORK — ENUM SUPPORT	128
24. ENTITY FRAMEWORK — ASYNCHRONOUS QUERY.....	143
25. ENTITY FRAMEWORK — PERSISTENCE	148
Creating Persistent Ignorant Entities	148
Connected Scenarios	148
Disconnected Scenarios	150
26. ENTITY FRAMEWORK — PROJECTION QUERIES.....	152
LINQ to Entities	152
LINQ to Entities Essential Keywords	152
Single Object	154
List of Objects	155
Order	155

Standard Vs Projection Entity Framework Query	156
27. ENTITY FRAMEWORK — COMMAND LOGGING	157
28. ENTITY FRAMEWORK — COMMAND INTERCEPTION.....	160
Registering Interceptors	161
29. ENTITY FRAMEWORK — SPATIAL DATA TYPE.....	163
30. ENTITY FRAMEWORK — INHERITANCE.....	167
31. ENTITY FRAMEWORK — MIGRATION	176
32. ENTITY FRAMEWORK — EAGER LOADING	182
33. ENTITY FRAMEWORK — LAZY LOADING	185
34. ENTITY FRAMEWORK — EXPLICIT LOADING	189
35. ENTITY FRAMEWORK — ENTITY VALIDATION.....	191
36. ENTITY FRAMEWORK — TRACK CHANGES.....	194
37. ENTITY FRAMEWORK — COLORED ENTITIES	198
38. ENTITY FRAMEWORK — CODE FIRST APPROACH	205
Why Code First?	206
Environment Setup	207
Install EF via NuGet Package	207
39. ENTITY FRAMEWORK — FIRST EXAMPLE.....	211
Create Model	211
Create Database Context.....	213
Database Initialization	217
No Parameter.....	218
Database Name.....	219

Connection String Name	219
Domain Classes	220
Data Annotations	221
Fluent API	221
40. ENTITY FRAMEWORK — DATA ANNOTATIONS	223
Key	223
Timestamp	225
ConcurrencyCheck	226
Required Annotation	227
MaxLength	228
StringLength	230
Table	230
Column	232
Index	233
ForeignKey	235
NotMapped	237
InverseProperty	238
41. ENTITY FRAMEWORK — FLUENT API	242
Entity Mapping	245
Default Schema	245
Map Entity to Table	246
Entity Splitting (Map Entity to Multiple Table)	247
Properties Mapping	249
Configuring a Primary Key	249
Configure Column	250
Configure MaxLength Property	250
Configure Null or NotNull Property	251

Configuring Relationships.....	251
Configure One-to-One Relationship	252
Configure One-to-Many Relationship	254
Configure Many-to-Many Relationship	256
42. ENTITY FRAMEWORK — SEED DATABASE.....	260
43. ENTITY FRAMEWORK — CODE FIRST MIGRATION	264
Automated Migration	264
Code Based Migration	270
44. ENTITY FRAMEWORK — MULTIPLE DBCONTEXT	274
45. ENTITY FRAMEWORK — NESTED ENTITY TYPES.....	280

1. Entity Framework — Overview

What is Entity Framework?

Entity Framework was first released in 2008, Microsoft's primary means of interacting between .NET applications and relational databases. Entity Framework is an Object Relational Mapper (ORM) which is a type of tool that simplifies mapping between objects in your software to the tables and columns of a relational database.

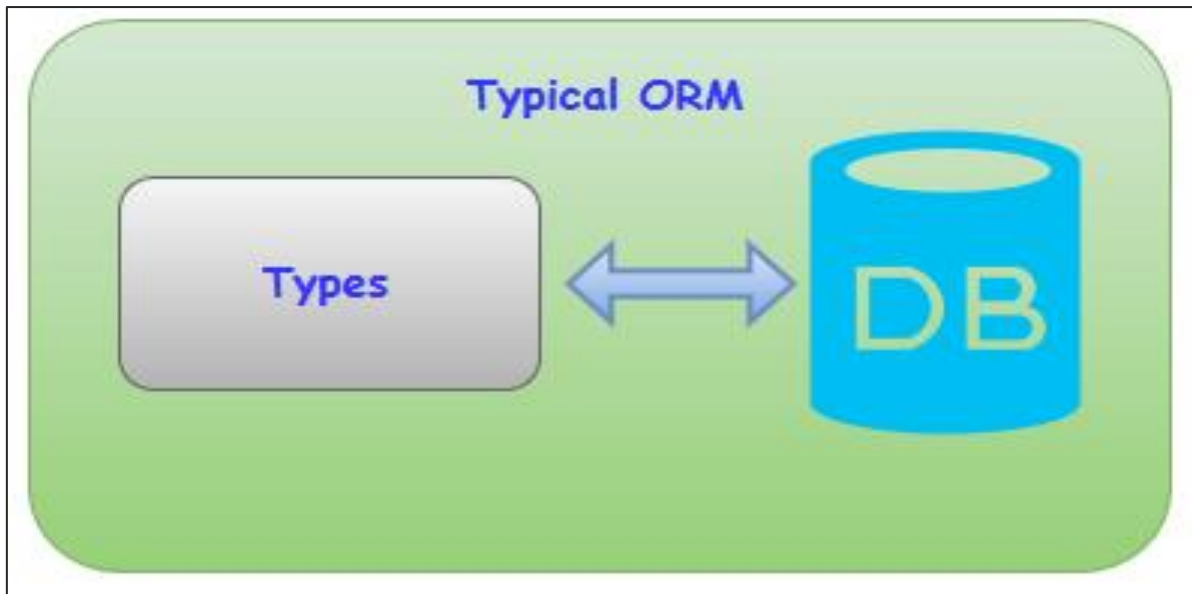
- Entity Framework (EF) is an open source ORM framework for ADO.NET which is a part of .NET Framework.
- An ORM takes care of creating database connections and executing commands, as well as taking query results and automatically materializing those results as your application objects.
- An ORM also helps to keep track of changes to those objects, and when instructed, it will also persist those changes back to the database for you.

Why Entity Framework?

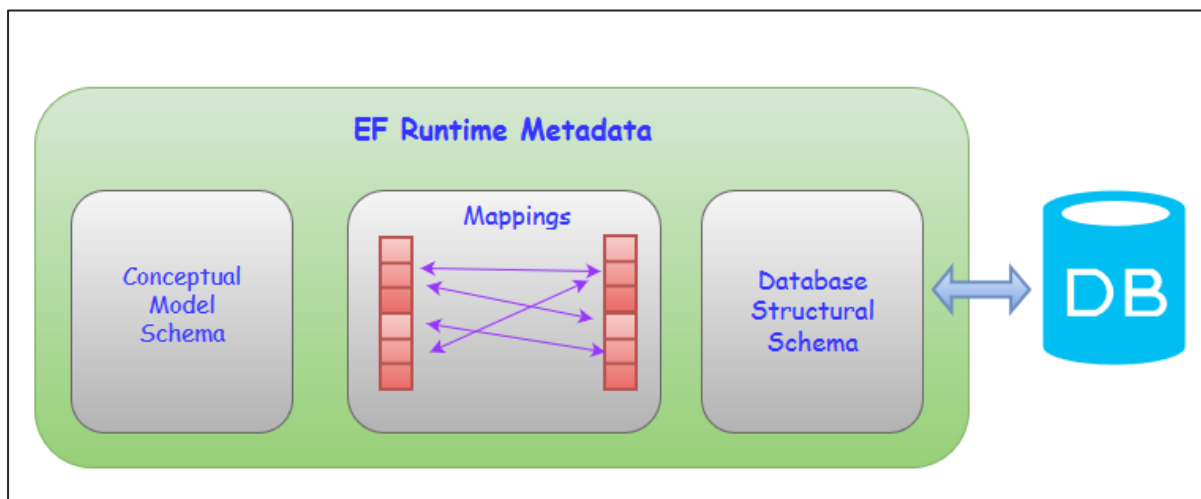
Entity Framework is an ORM and ORMs are aimed to increase the developer's productivity by reducing the redundant task of persisting the data used in the applications.

- Entity Framework can generate the necessary database commands for reading or writing data in the database and execute them for you.
- If you're querying, you can express your queries against your domain objects using LINQ to entities.
- Entity Framework will execute the relevant query in the database and then materialize results into instances of your domain objects for you to work within your app.

There are other ORMs in the marketplace such as NHibernate and LLBLGen Pro. Most ORMs typically map domain types directly to the database schema.



Entity Framework has a more granular mapping layer so you can customize mappings, for example, by mapping the single entity to multiple database tables or even multiple entities to a single table.

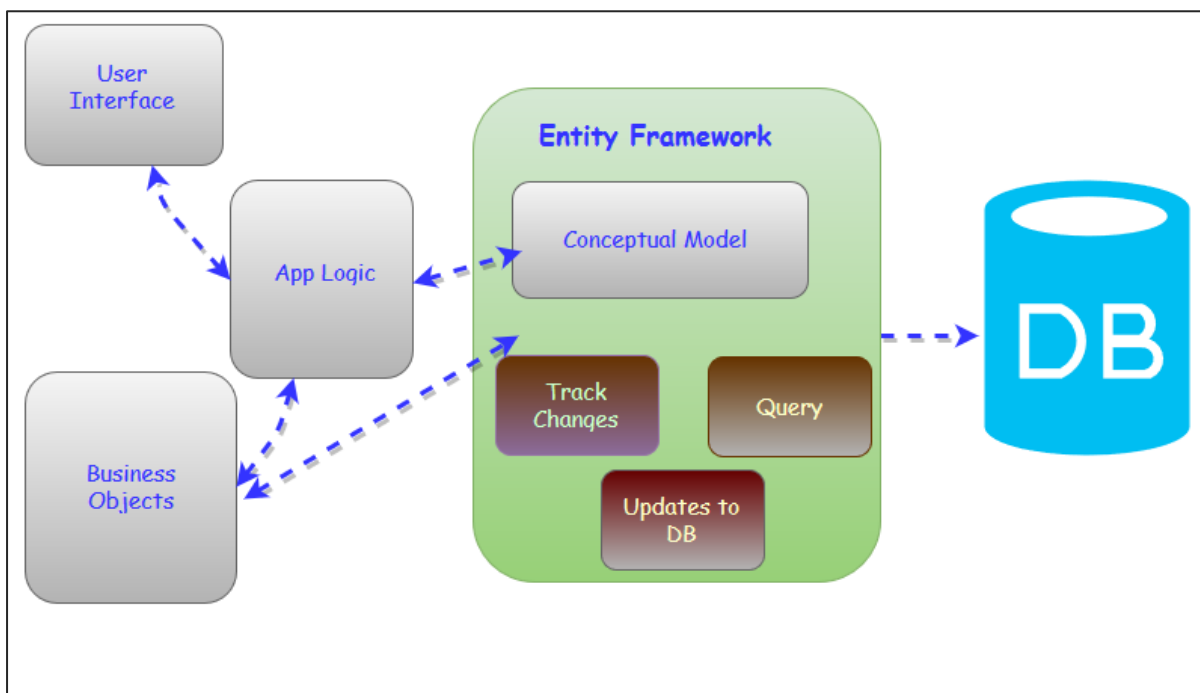


- Entity Framework is Microsoft's recommended data access technology for new applications.
- ADO.NET seems to refer directly to the technology for data sets and data tables.
- Entity Framework is where all of the forward moving investment is being made, which has been the case for a number of years already.
- Microsoft recommends that you use Entity Framework over ADO.NET or LINQ to SQL for all new development.

Conceptual Model

For developers who are used to database focused development, the biggest shift with Entity Framework is that it lets you focus on your business domain. What it is that you want your application to do without being limited by what the database is able to do?

- With Entity Framework, the focal point is referred to as a conceptual model. It's a model of the objects in your application, not a model of the database you use to persist your application data.
- Your conceptual model may happen to align with your database schema or it may be quite different.
- You can use a Visual Designer to define your conceptual model, which can then generate the classes you will ultimately use in your application.
- You can just define your classes and use a feature of Entity Framework called Code First. And then Entity Framework will comprehend the conceptual model.



Either way, Entity Framework works out how to move from your conceptual model to your database. So, you can query against your conceptual model objects and work directly with them.

Features

Following are the basic features of Entity Framework. This list is created based on the most notable features and also from frequently asked questions about Entity Framework.

- Entity Framework is a Microsoft tool.
- Entity Framework is being developed as an Open Source product.
- Entity Framework is no longer tied or dependent to the .NET release cycle.
- Works with any relational database with valid Entity Framework provider.
- SQL command generation from LINQ to Entities.
- Entity Framework will create parameterized queries.
- Tracks changes to in-memory objects.

- Allows to insert, update and delete command generation.
- Works with a visual model or with your own classes.
- Entity Framework has stored Procedure Support.

2. Entity Framework — Architecture

The architecture of Entity Framework, from the bottom up, consists of the following:

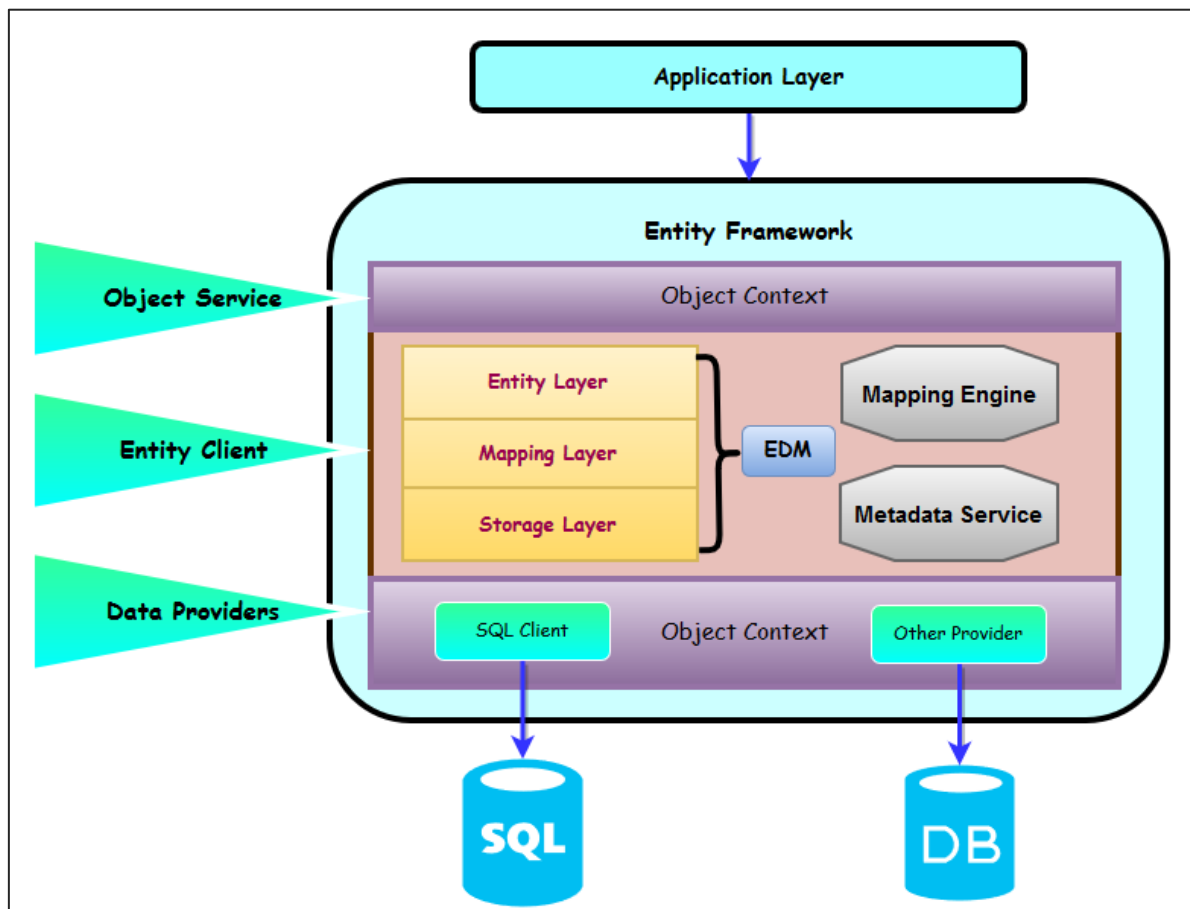
Data Providers

These are source specific providers, which abstract the ADO.NET interfaces to connect to the database when programming against the conceptual schema.

It translates the common SQL languages such as LINQ via command tree to native SQL expression and executes it against the specific DBMS system.

Entity Client

This layer exposes the entity layer to the upper layer. Entity client provides the ability for developers to work against entities in the form of rows and columns using entity SQL queries without the need to generate classes to represent conceptual schema. Entity Client shows the entity framework layers, which are the core functionality. These layers are called as Entity Data Model.



- The **Storage Layer** contains the entire database schema in XML format.

- The **Entity Layer** which is also an XML file defines the entities and relationships.
- The **Mapping layer** is an XML file that maps the entities and relationships defined at conceptual layer with actual relationships and tables defined at logical layer.
- The **Metadata services** which is also represented in Entity Client provides centralized API to access metadata stored Entity, Mapping and Storage layers.

Object Service

Object Services layer is the Object Context, which represents the session of interaction between the applications and the data source.

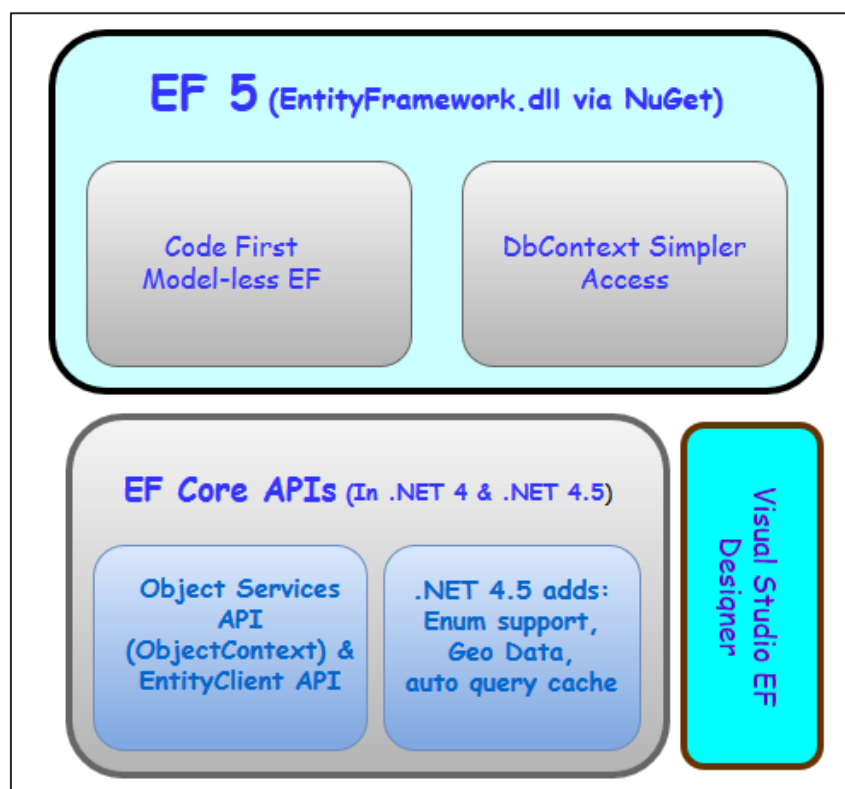
- The main use of the Object Context is to perform different operations like add, delete instances of entities and to save the changed state back to the database with the help of queries.
- It is the ORM layer of Entity Framework, which represents the data result to the object instances of entities.
- This services allow developer to use some of the rich ORM features like primary key mapping, change tracking, etc. by writing queries using LINQ and Entity SQL.

3. Entity Framework — Environment Setup

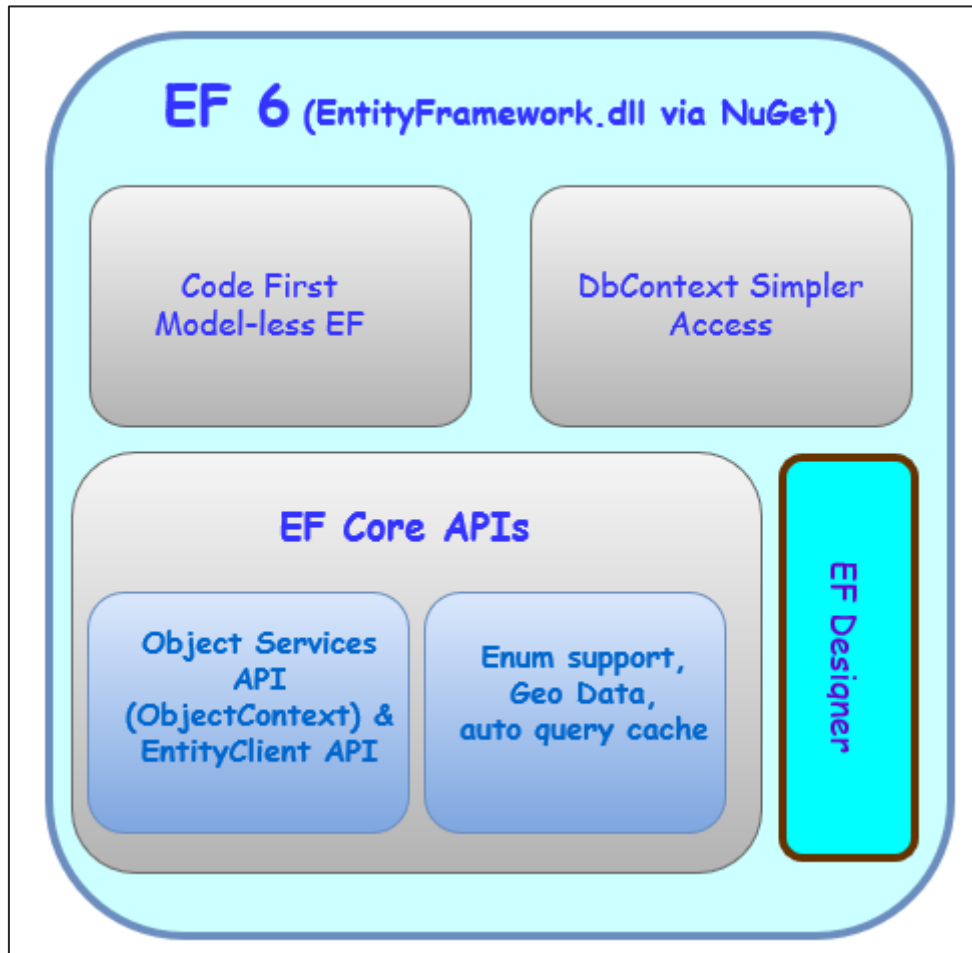
What's New in Entity Framework 6?

Entity Framework has a complex API that lets you have granular control over everything from its modeling to its runtime behavior. Part of Entity Framework 5 lives inside of .NET. And another part of it lives inside of an additional assembly that's distributed using NuGet.

- The core functionality of Entity Framework is built into the .NET Framework.
- The Code First support, that's what lets Entity Framework use classes in lieu of a visual model, and a lighter way API for interacting with EF are in the NuGet package.
- The core is what provides the querying, change tracking and all of the transformation from your queries to SQL queries as well as from data return into the objects.
- You can use the EF 5 NuGet package with both .NET 4 and with .NET 4.5.
- One big point of confusion - .NET 4.5 added support for enums and spatial data to the core Entity Framework APIs, which means if you're using EF 5 with .NET 4, you won't get these new features. You'll only get them when combining EF5 with .NET 4.5.



Let us now take a look at Entity Framework 6. The core APIs which were inside of .NET in Entity Framework 6 are now a part of NuGet package.



It means:

- All of the Entity Framework lives inside this assembly that's distributed by NuGet.
- You won't be dependent on .NET to provide specific features like the Entity Framework enum support and special data support.
- You'll see that one of the features of EF6 is that it supports enums and spatial data for .NET 4

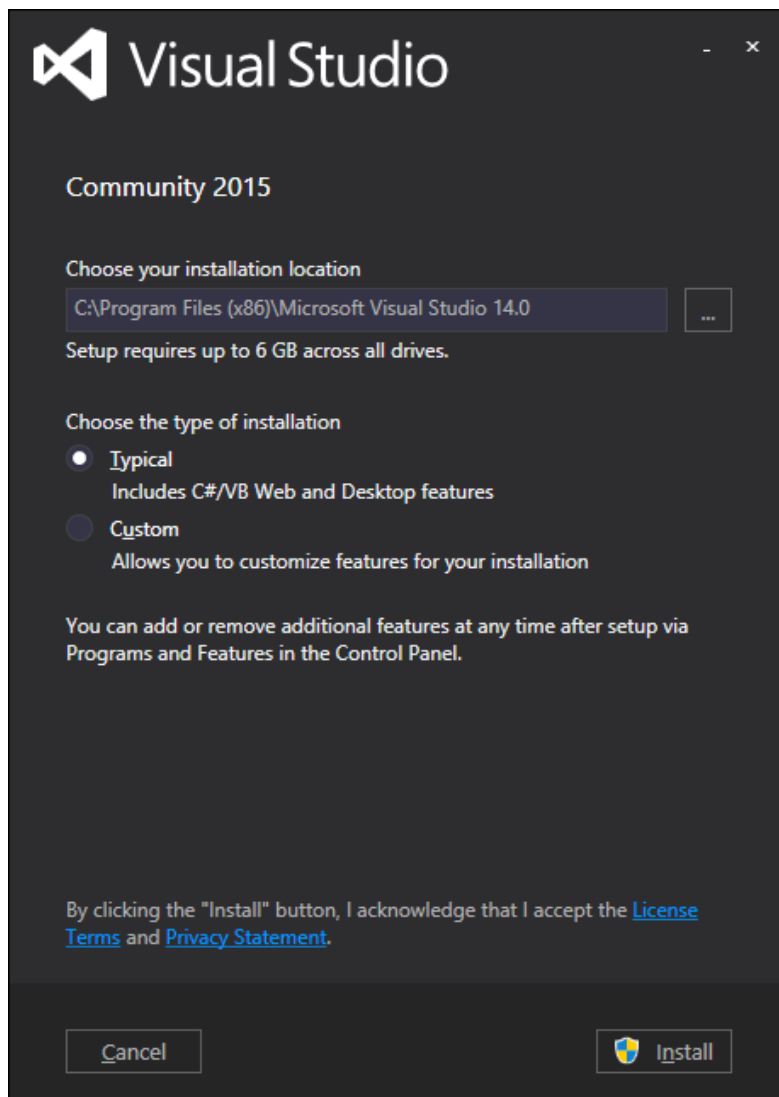
To start working on Entity Framework you need to install the following development tools:

- Visual Studio 2013 or above
- SQL Server 2012 or above
- Entity Framework updates from NuGet Package

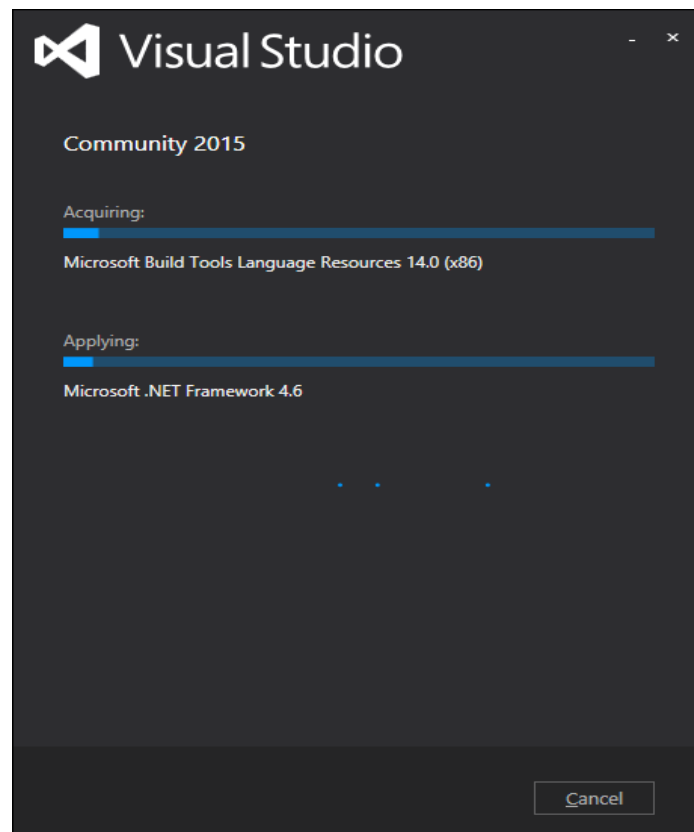
Microsoft provides a free version of visual studio which also contains SQL Server and it can be downloaded from <https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx>.

Installation

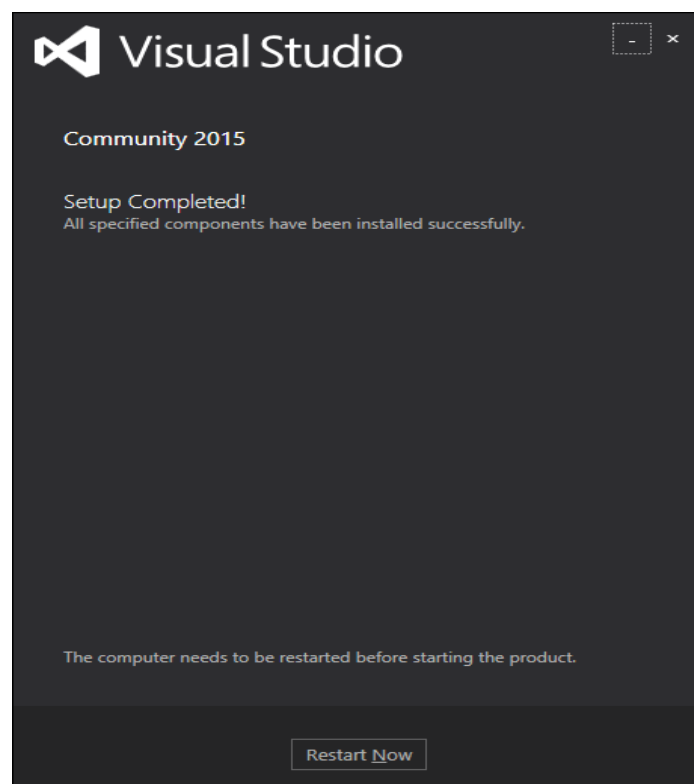
Step 1: Once downloading is complete, run the installer. The following dialog will be displayed.



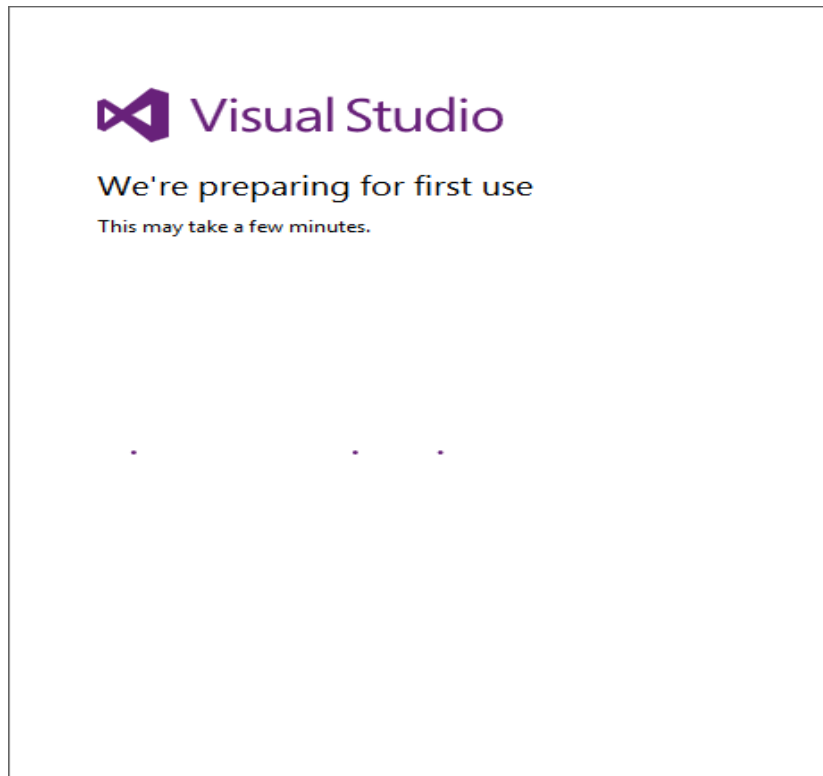
Step 2: Click on the Install button and it will start the installation process.



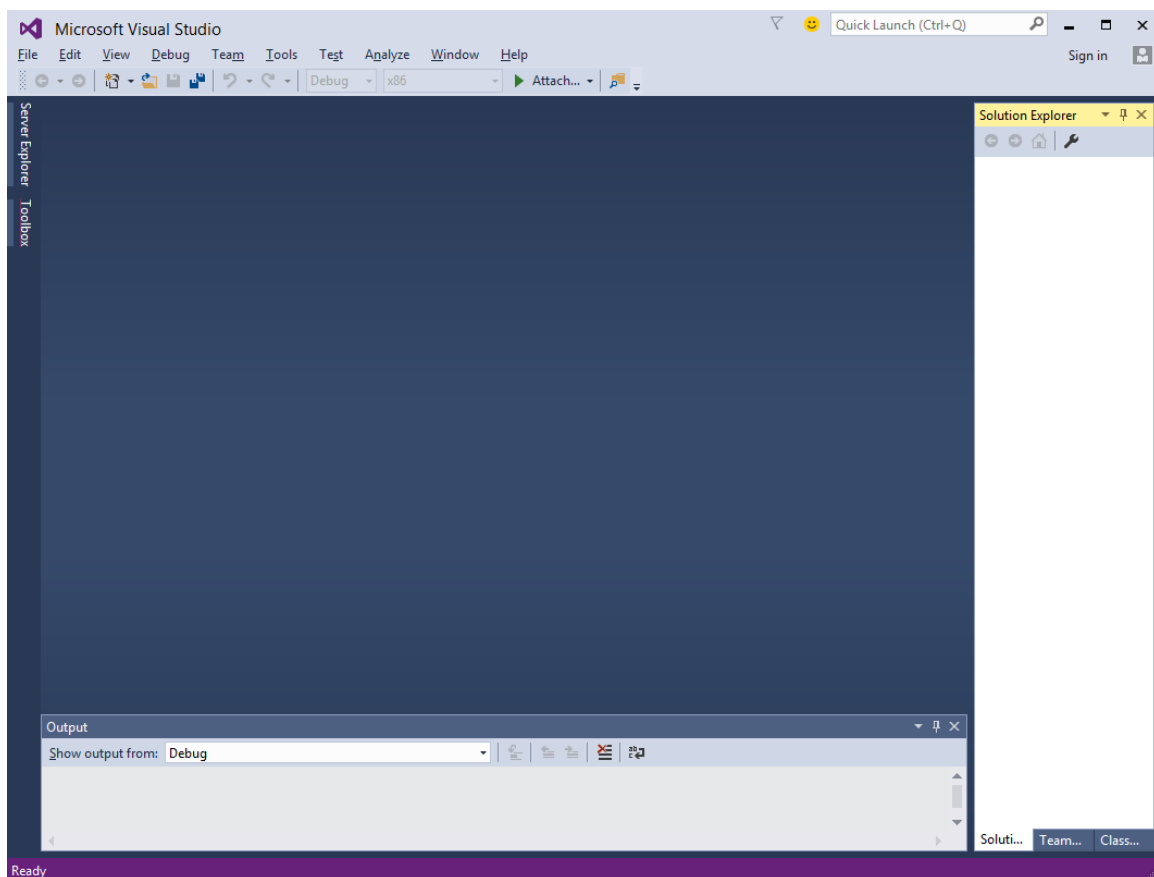
Step 3: Once the installation process is completed successfully, you will see the following dialog. Close this dialog and restart your computer if required.



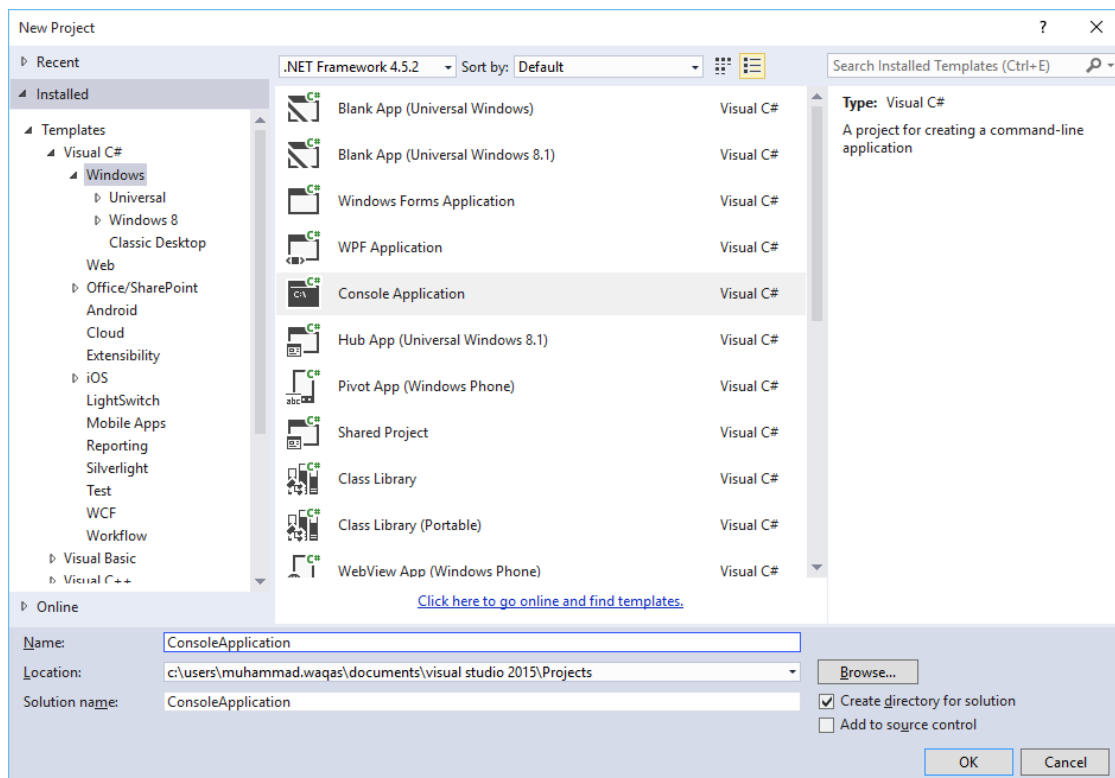
Step 4: Open Visual Studio from start Menu which will open the following dialog. It will be a while for the first time for preparation.



Step 5: Once all is done you will see the main window of Visual studio.

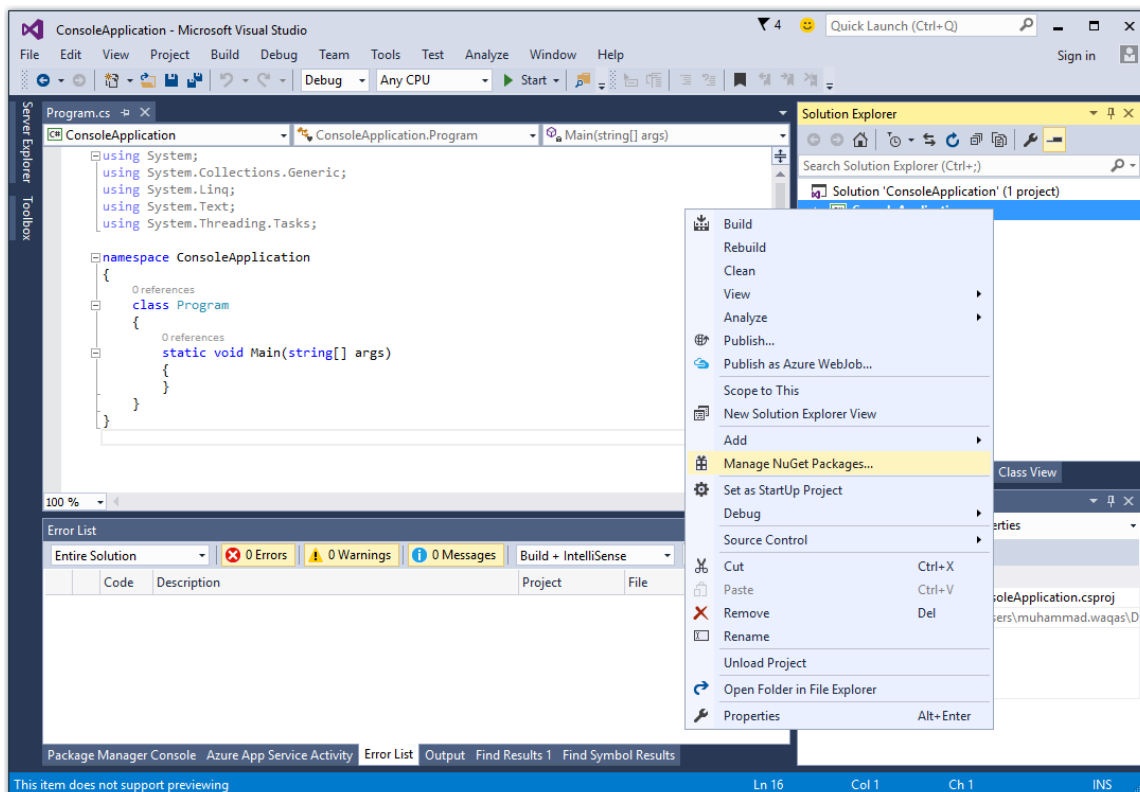


Let's create a new project from File -> New -> Project

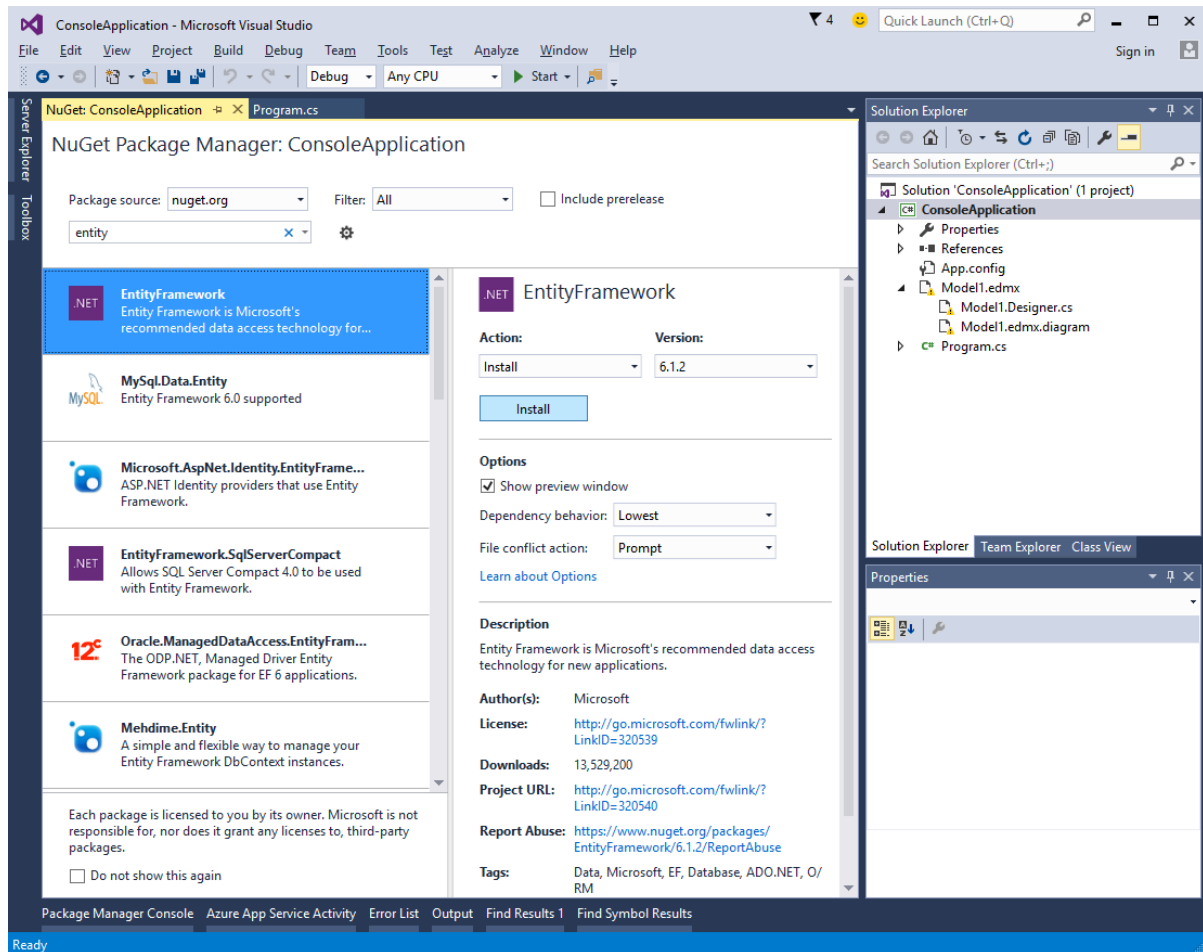


Step 1: Select Console Application and click OK button.

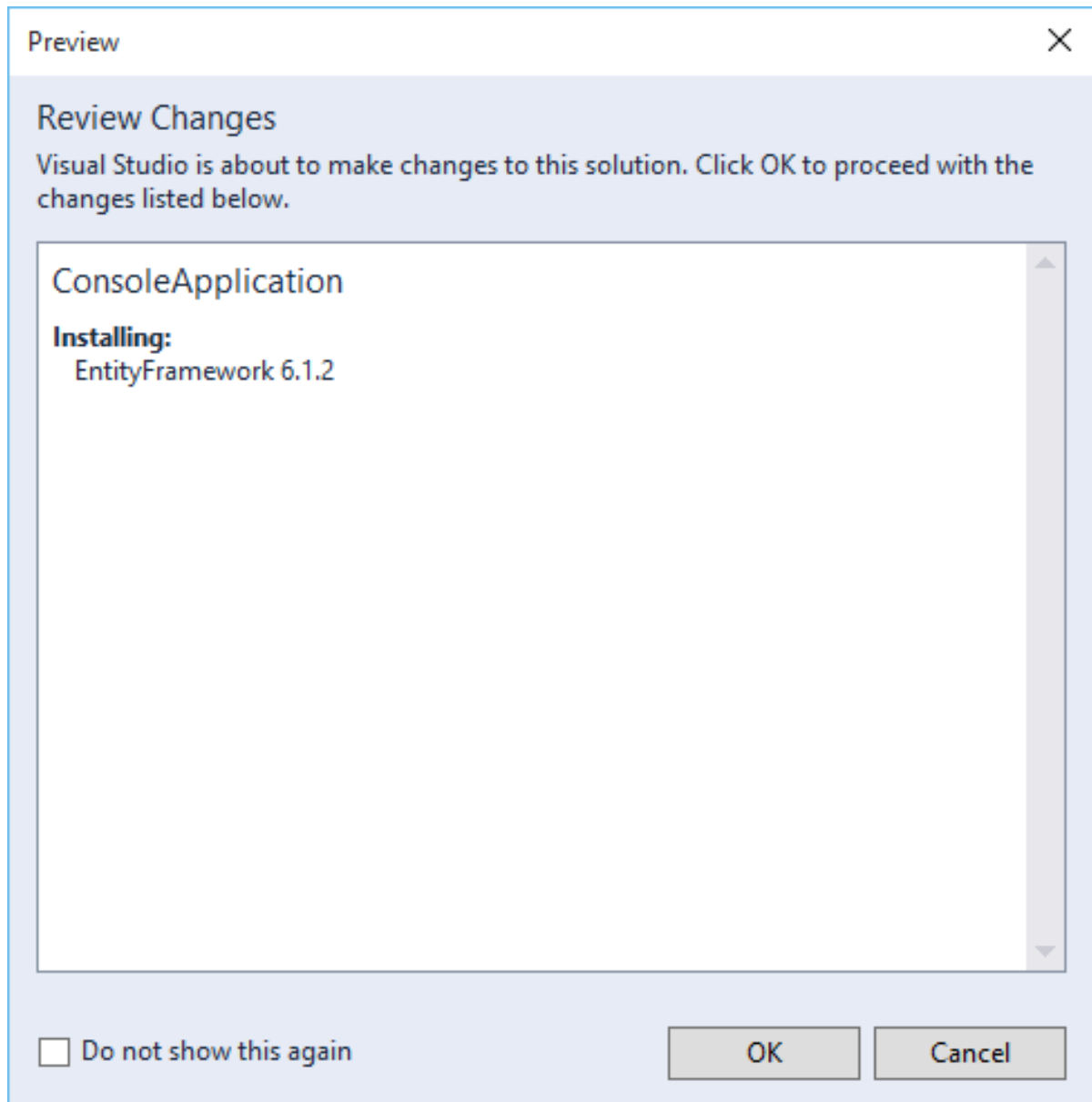
Step 2: In solution Explorer, right-click on your project.



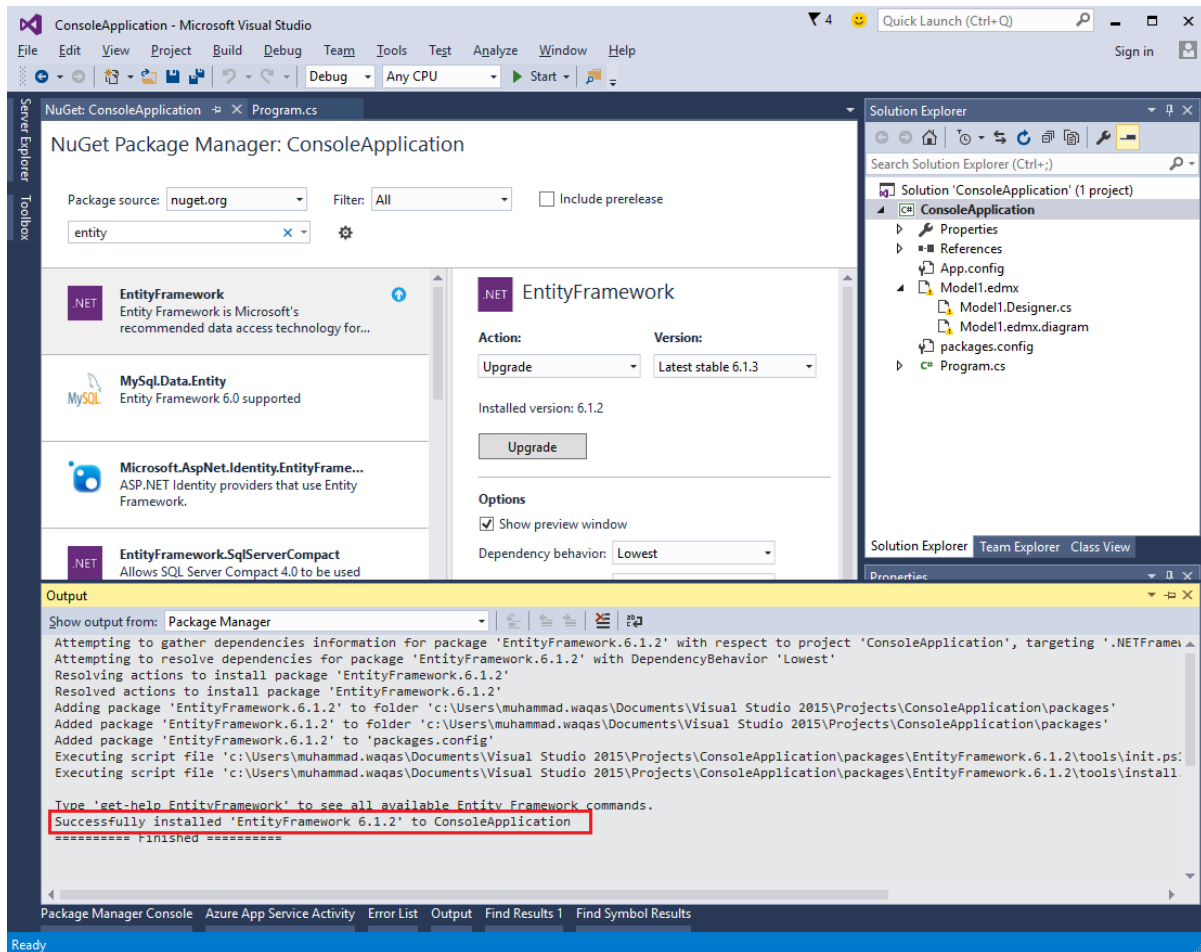
Step 3: Select Manage NuGet Packages as shown in the above image, which will open the following window in Visual Studio.



Step 4: Search for Entity Framework and install the latest version by pressing the install button.



Step 5: Click Ok. Once installation is done, you will see the following message in your output Window.

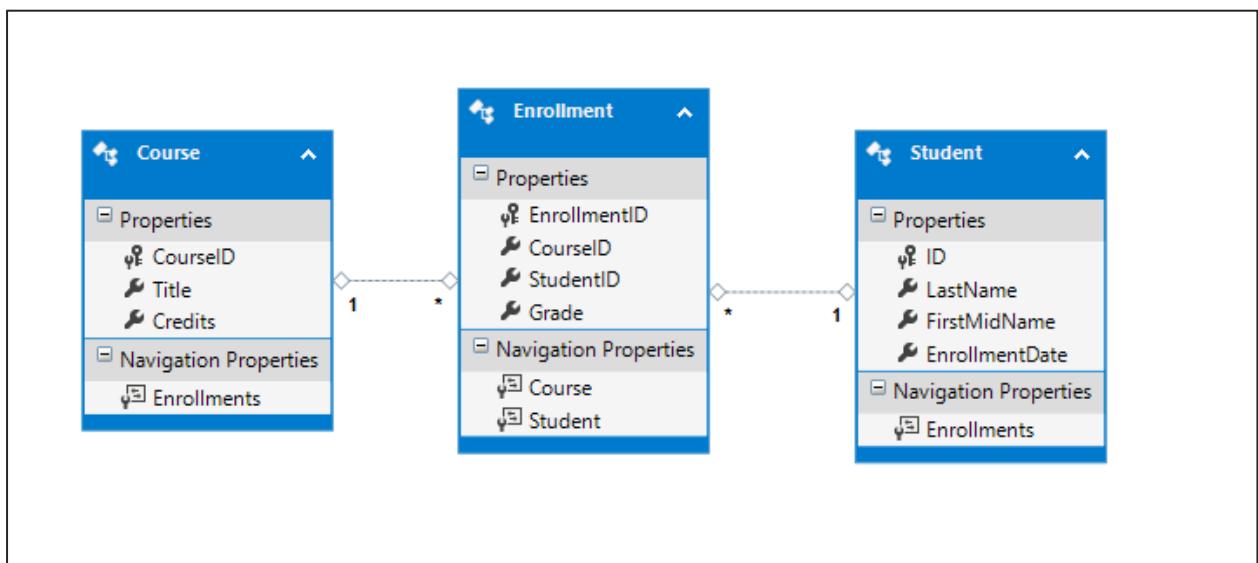


You are now ready to start your application.

4. Entity Framework — Database Setup

In this tutorial, we will be using a simple University database. A University database can be much more complex as a whole but for demo and learning purpose, we are using the simplest form of this database. The following diagram contains three tables.

- Student
- Course
- Enrollment



Whenever a term database is used one thing comes directly to our mind and that is different kind of tables which has some sort of relationship. There are three types of relationships between tables and the relationship between different tables depends on how the related columns are defined.

- One-to-Many Relationship
- Many-to-Many Relationship
- One-to-One Relationship

One-to-Many Relationship

One-to-many relationship is the most common type of relationship. In this type of relationship, a row in table A can have many matching rows in table B, but a row in table B can have only one matching row in table A. For example, in the above diagram, Student and Enrollment table have one-to-many relationship, each student may have many enrollments, but each enrollment belongs to only one student.

Many-to-Many Relationship

In a many-to-many relationship, a row in table A can have many matching rows in table B, and vice versa. You create such a relationship by defining a third table, called a junction table, whose primary key consists of the foreign keys from both table A and table B. For example, Student and Course table have many-to-many relationship that is defined by a one-to-many relationship from each of these tables to the Enrollment table.

One-to-One Relationship

In one-to-one relationship, a row in table A can have no more than one matching row in table B, and vice versa. A one-to-one relationship is created if both of the related columns are primary keys or have unique constraints.

This type of relationship is not common because most information related in this way would be all-in-one table. You might use a one-to-one relationship to:

- Divide a table with many columns.
- Isolate part of a table for security reasons.
- Store data that is short-lived and could be easily deleted by simply deleting the table.
- Store information that applies only to a subset of the main table.

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://store.tutorialspoint.com>