**Entity Framework Core Developer’s Manual**

* Install the Package Manager Console tools by running the following command in **Package Manager Console**:

**Install-Package Microsoft.EntityFrameworkCore.Tools**

* Update the tools by running the following command in **Package Manager Console**.

**Update-Package Microsoft.EntityFrameworkCore.Tools**

* Verify that the tools are installed by running this command:

**Get-Help about\_EntityFrameworkCore**

* You can specify [the environment](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/environments) for ASP.NET Core projects on the command-line. This and any additional arguments are passed into Program.CreateHostBuilder.

**Update-Database -Args '--environment Production'**

| * **Parameter** | * **Description** |
| --- | --- |
| * -Context <String> | * The DbContext class to use. Class name only or fully qualified with namespaces. If this parameter is omitted, EF Core finds the context class. If there are multiple context classes, this parameter is required. |
| * -Project <String> | * The target project. If this parameter is omitted, the **Default project** for **Package Manager Console** is used as the target project. |
| * -StartupProject <String> | * The startup project. If this parameter is omitted, the **Startup project** in **Solution properties** is used as the target project. |
| * -Args <String> | * Arguments passed to the application. |
| * -Verbose | * Show verbose output. |

**Add-Migration**

| **Parameter** | **Description** |
| --- | --- |
| -Name <String> | The name of the migration. This is a positional parameter and is required. |
| -OutputDir <String> | The directory use to output the files. Paths are relative to the target project directory. Defaults to "Migrations". |
| -Namespace <String> | The namespace to use for the generated classes. Defaults to generated from the output directory. |

**Bundle-Migration:**

| **Parameter** | **Description** |
| --- | --- |
| -Output <String> | The path of executable file to create. |
| -Force | Overwrite existing files. |
| -SelfContained | Also bundle the .NET runtime so it doesn't need to be installed on the machine. |
| -TargetRuntime <String> | The target runtime to bundle for. |
| -Framework <String> | The target framework. Defaults to the first one in the project. |

* You are free to move Migrations files and change their namespace manually. New migrations are created as siblings of the last migration. Alternatively, you can specify the directory at generation time as follows:

**Add-Migration InitialCreate -OutputDir Your\Directory**

* Sometimes you add a migration and realize you need to make additional changes to your EF Core model before applying it. To remove the last migration, use this command.

**Remove-Migration**

* To roll back your database to the previous migration:

**Update-Database PreviousMigrationName**

* You can list all existing migrations as follows:

**Get-Migration**

* This command **won’t create a migration**, but it will show you **what would change** if you did

**Add-Migration DetectChanges -WhatIf -Context YourDbContextName**

 The -WhatIf flag **simulates** the migration process.

 If there are changes in your **Entity Framework Core model** (e.g., class or property changes, navigation updates, etc.), it will **output a migration preview** without writing any files.

 If there are **no changes**, it will say something like:

pgsql

CopyEdit

No changes were found in your model for the current context.

**Data Migration on KenHRApp.Infrastructure Project:**

* Data Migration PMC Command #1

Add-Migration InitialCreate -Context AppDbContext -Project KenHRApp.Infrastructure -StartupProject KenHRApp.Web -Verbose

Update-Database -Context AppDbContext

Get-Migration -Context AppDbContext

* Data Migration PMC Command #2

Add-Migration RenameMaritalStatusToMaritalStatusCode -Context AppDbContext -Project KenHRApp.Infrastructure -StartupProject KenHRApp.Web -Verbose

Update-Database -Context AppDbContext

* Data Migration PMC Command #3

Add-Migration RefactorEmployeeEntitySchema -Context AppDbContext -Project KenHRApp.Infrastructure -StartupProject KenHRApp.Web -Verbose

Update-Database -Context AppDbContext

* Data Migration PMC Command #4

Add-Migration UpdateDataModel -Context AppDbContext -Project KenHRApp.Infrastructure -StartupProject KenHRApp.Web -Verbose

Update-Database -Context AppDbContext

* Data Migration PMC Command #5

Add-Migration AddUserDefinedCode -Context AppDbContext -Project KenHRApp.Infrastructure -StartupProject KenHRApp.Web -Verbose

Update-Database -Context AppDbContext

* Data Migration PMC Command #6

Add-Migration ModifyEmployeeModel -Context AppDbContext -Project KenHRApp.Infrastructure -StartupProject KenHRApp.Web -Verbose

Update-Database -Context AppDbContext

**To remove previous migration without updating the database:**

Remove-Migration -Context AppDbContext

**To get the list of all existing migrations:**

Get-Migration -Context AppDbContext

**AI Prompts**

I have successfully executed 5 migrations for my data model and have updated the database. However, I need to execute the same number of migrations but to a different database. Tell me how to run the migrations to a new database in Entity Framework Core.

I have created a new key called “AppSettings” in the appsettings.json file. This key contains and object value where I defined a property called “Environment”. Now, tell me how to get the value of the “Environment” property under “AppSettings” key in Program.cs

I have successfully executed the SQL migration script on my new database. Now, I will create views and table view function database objects using SQL Server scripts. Tell me how to synchronize the database objects that I will create in SQL Server with my Entity Framework data model.