

Database First

發行項 • 2023/09/12

這段影片和逐步解說提供使用 Entity Framework 進行 Database First 開發的簡介。

Database First 可讓您從現有的資料庫反向工程模型。模型會儲存在 EDMX 檔案（.edmx 副檔名）中，而且可以在 Entity Framework Designer 中檢視和編輯。您在應用程式中互動的類別會自動從 EDMX 檔案產生。

觀賞影片

這段影片提供使用 Entity Framework 進行 Database First 開發的簡介。Database First 可讓您從現有的資料庫反向工程模型。模型會儲存在 EDMX 檔案（.edmx 副檔名）中，而且可以在 Entity Framework Designer 中檢視和編輯。您在應用程式中互動的類別會自動從 EDMX 檔案產生。

主講人 [Rowan Miller](#)

影片：[WMV](#) | [MP4](#) | [WMV \(ZIP\)](#)

必要條件

您必須至少安裝 Visual Studio 2010 或 Visual Studio 2012，才能完成本逐步解說。

如果您使用 Visual Studio 2010，您也必須 [安裝 NuGet](#)。

1. 建立現有的資料庫

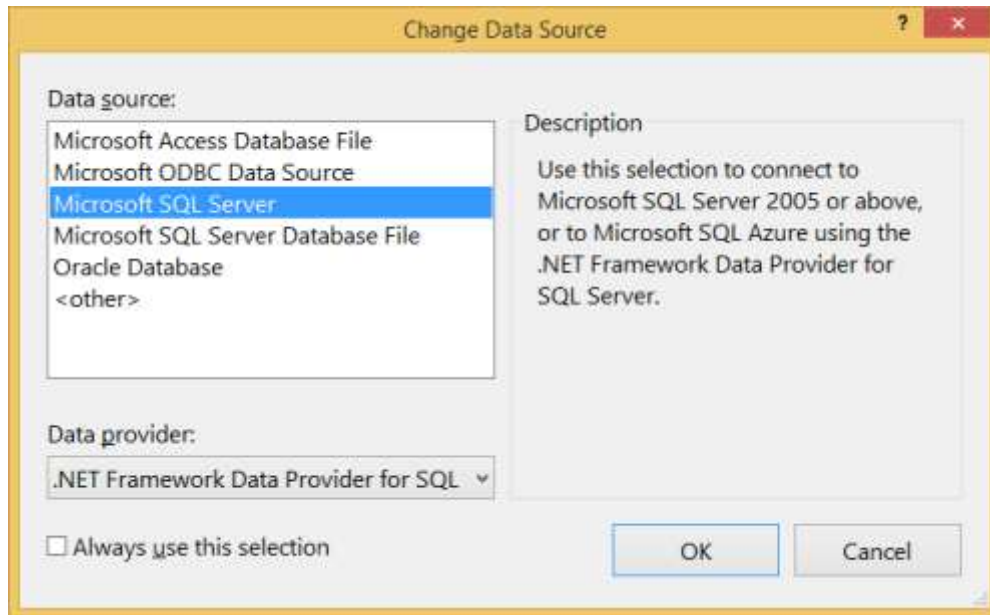
一般而言，當您將目標設為現有的資料庫時，就會建立它，但在本逐步解說中，我們需要建立資料庫才能存取。

隨 Visual Studio 一起安裝的資料庫伺服器會根據您安裝的 Visual Studio 版本而有所不同：

- 如果您使用 Visual Studio 2010，您將建立 SQL Express 資料庫。
- 如果您使用 Visual Studio 2012，您將建立 [LocalDB](#) 資料庫。

讓我們繼續產生資料庫。

- 開啟 Visual Studio
- 檢視 - > 伺服器總管
- 以滑鼠右鍵按一下 [資料連線 - > 新增連線...
- 如果您尚未從 [伺服器總管] 連線到資料庫，則必須先選取 [Microsoft SQL Server] 作為資料來源



- 視您已安裝的本機DB 或 SQL Express 而定，連線至 LocalDB 或 SQL Express，然後輸入 **DatabaseFirst.Blogging** 作為資料庫名稱

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
.\SQLEXPRESS Refresh

Log on to the server

☒ Use Windows Authentication
☐ Use SQL Server Authentication

User name:
Password:
☐ Save my password

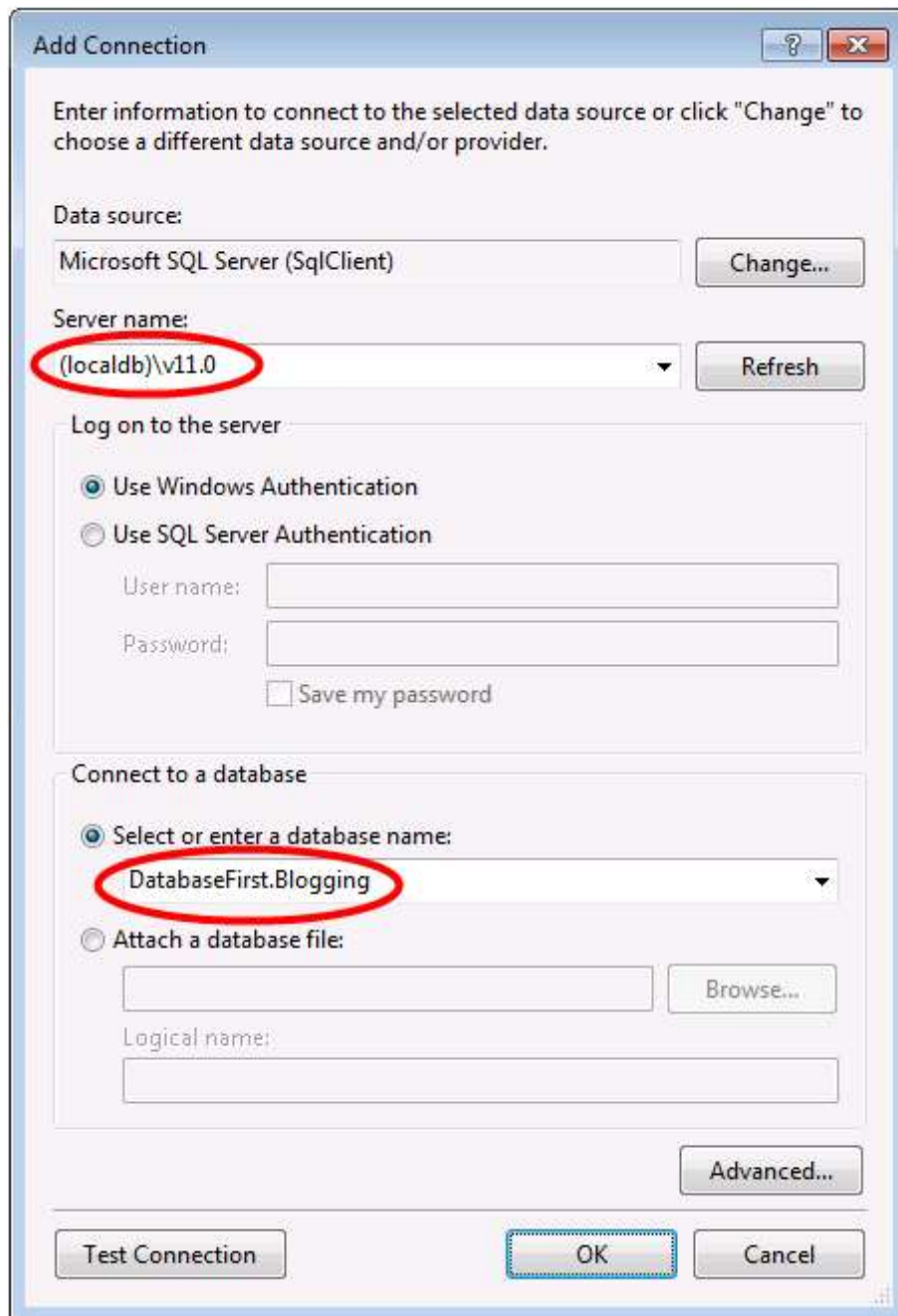
Connect to a database

☒ Select or enter a database name:
DatabaseFirst.Blogging

☐ Attach a database file:
 Browse...
Logical name:

Advanced...

Test Connection OK Cancel



- 選取 [確定]，系統會詢問您是否要建立新的資料庫，選取 [是]



- 新的資料庫現在會出現在 [伺服器總管] 中，以滑鼠右鍵按一下它，然後選取 [新增查詢]

- 將下列 SQL 複製到新的查詢，然後以滑鼠右鍵按一下查詢，然後選取 [執行]

SQL

```
CREATE TABLE [dbo].[Blogs] (  
    [BlogId] INT IDENTITY (1, 1) NOT NULL,  
    [Name] NVARCHAR (200) NULL,  
    [Url] NVARCHAR (200) NULL,  
    CONSTRAINT [PK_dbo.Blogs] PRIMARY KEY CLUSTERED ([BlogId] ASC)  
);  
  
CREATE TABLE [dbo].[Posts] (  
    [PostId] INT IDENTITY (1, 1) NOT NULL,  
    [Title] NVARCHAR (200) NULL,  
    [Content] NTEXT NULL,  
    [BlogId] INT NOT NULL,  
    CONSTRAINT [PK_dbo.Posts] PRIMARY KEY CLUSTERED ([PostId] ASC),  
    CONSTRAINT [FK_dbo.Posts_dbo.Blogs_BlogId] FOREIGN KEY ([BlogId])  
REFERENCES [dbo].[Blogs] ([BlogId]) ON DELETE CASCADE  
);
```

2. 建立應用程式

為了保持簡單，我們將建置使用 Database First 執行資料存取的基本主控台應用程式：

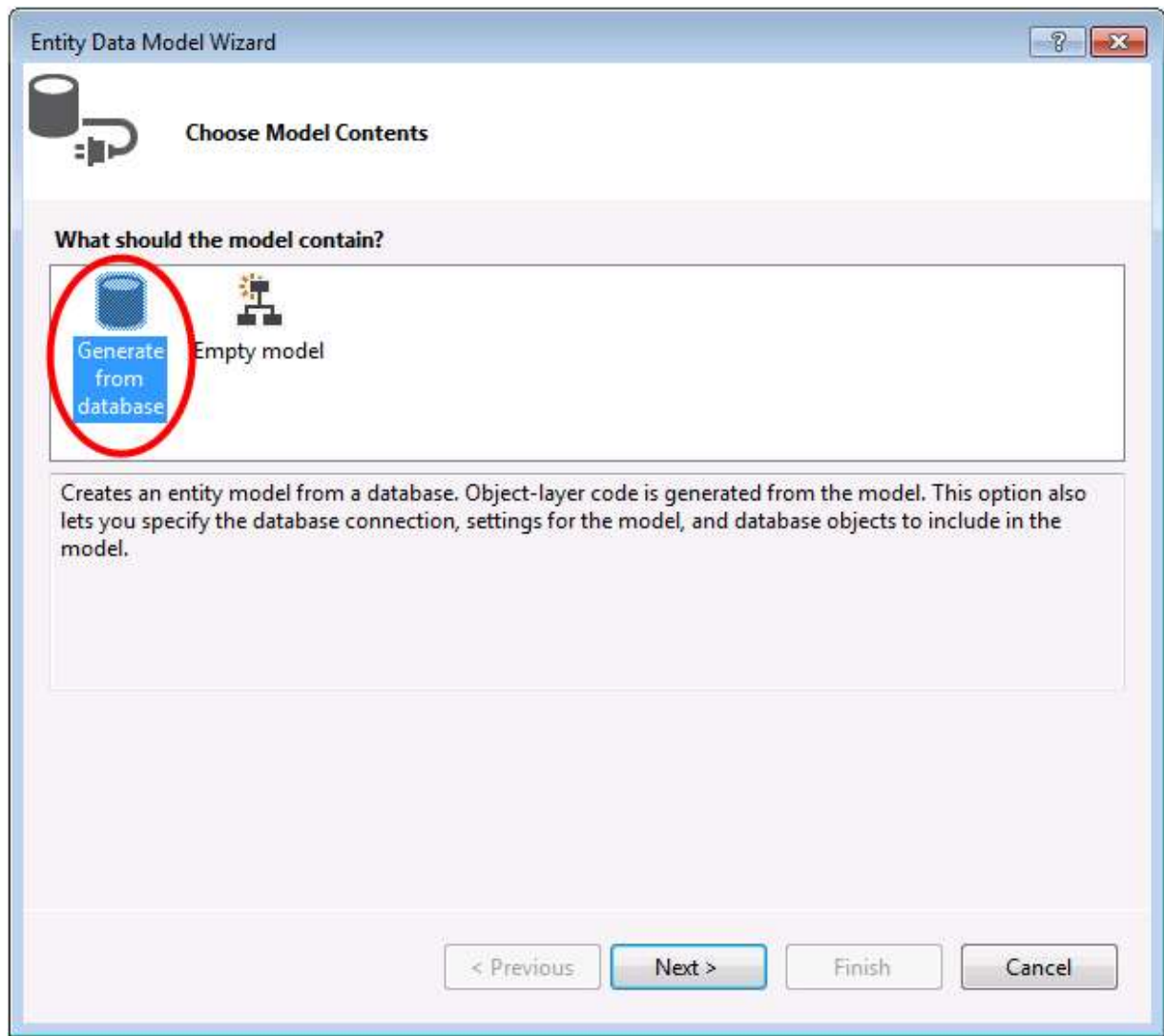
- 開啟 Visual Studio
- 檔案 - > 新增 - > 專案...
- 從左側功能表選取 [Windows] 和 [主控台應用程式]
- 輸入 **DatabaseFirstSample** 作為名稱
- 選取**確定**

3. 反向工程師模型

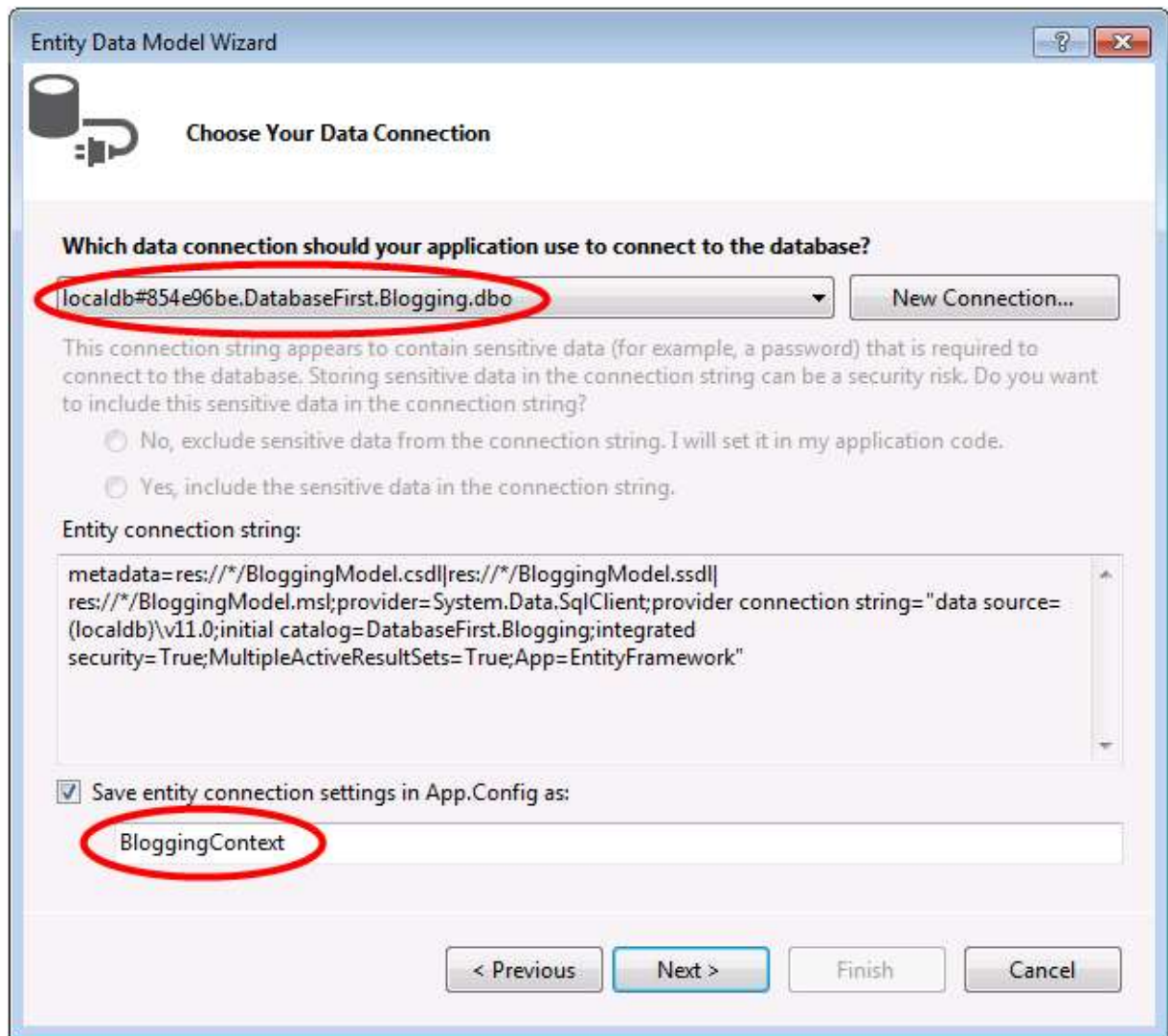
我們將使用包含在 Visual Studio 中的 Entity Framework Designer 來建立模型。

- 專案 - > 新增專案...
- 從左側功能表中選取 [資料]，然後 **ADO.NET 實體資料模型**
- 輸入 **BloggingModel** 作為名稱，然後按一下 [確定]

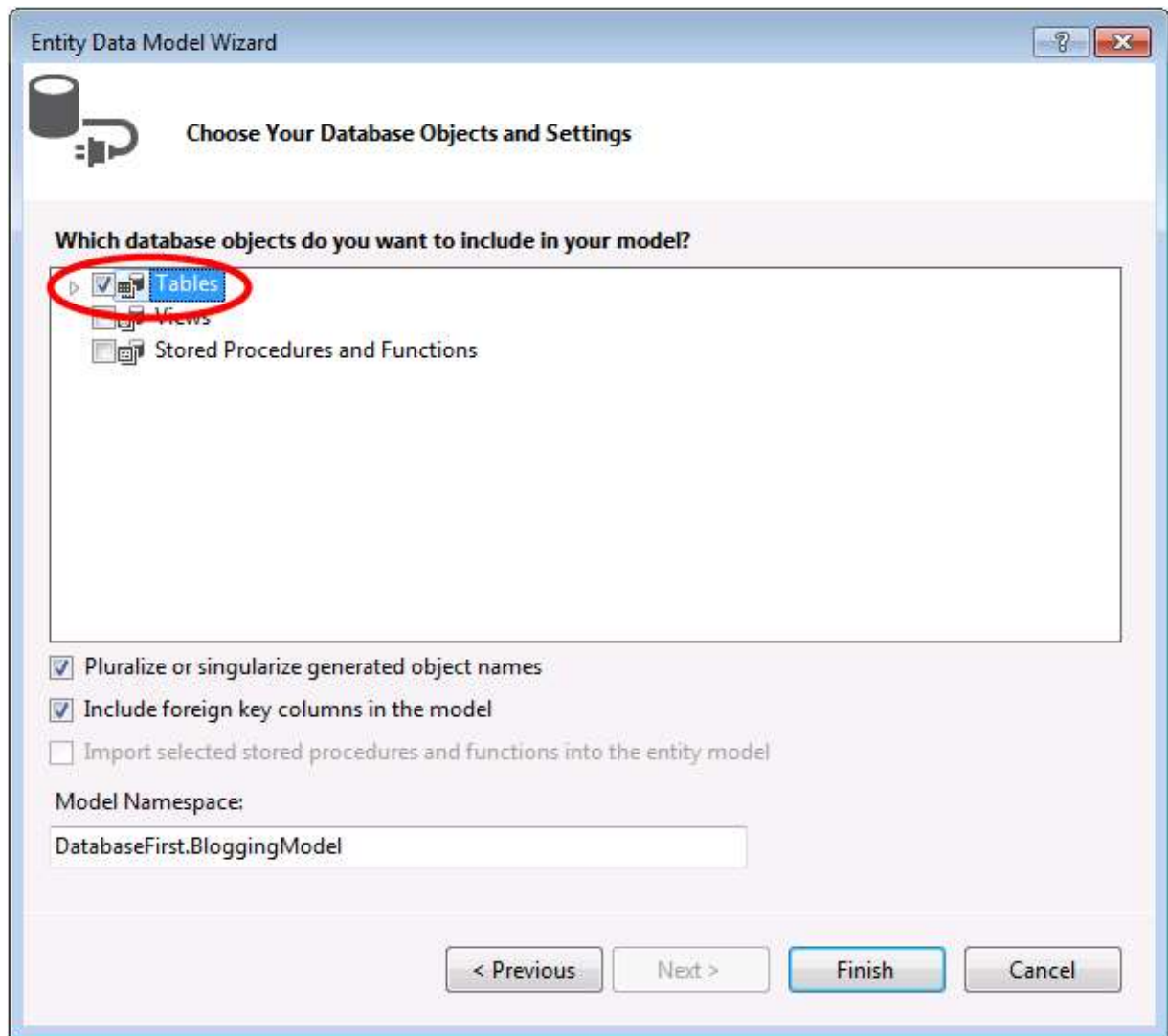
- 這會啟動 **實體資料模型精靈**
- 選取 [從資料庫 產生]，然後按 [下一步]



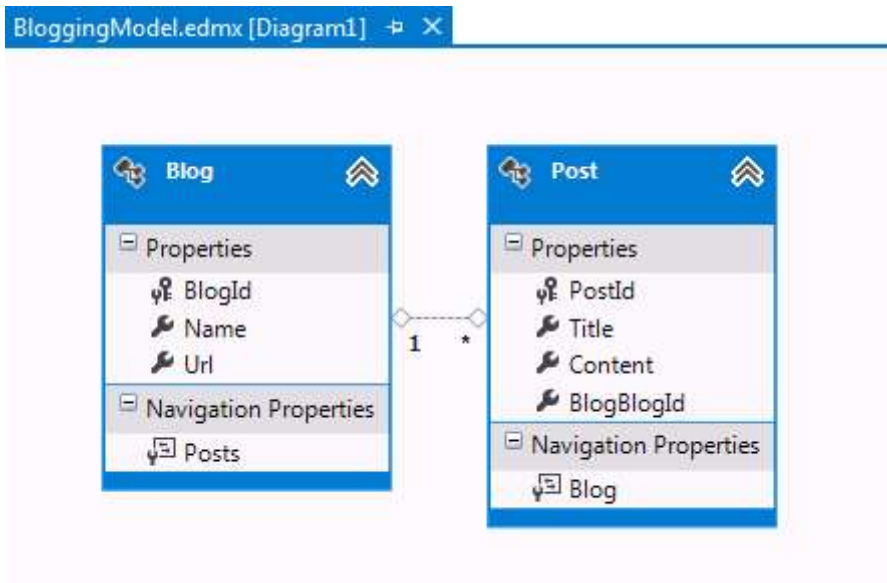
- 選取您在第一節中建立之資料庫的連線，輸入 **BloggingCoNtext** 作為連接字串的名稱，然後按 [下一步]



- 按一下 [資料表] 旁的核取方塊以匯入所有資料表，然後按一下 [完成]



一旦反向工程師程式完成，新的模型就會新增至您的專案，並開啟以供您在 Entity Framework Designer 中檢視。App.config 檔案也已新增至您的專案，其中包含資料庫的連線詳細資料。



Visual Studio 2010 中的其他步驟

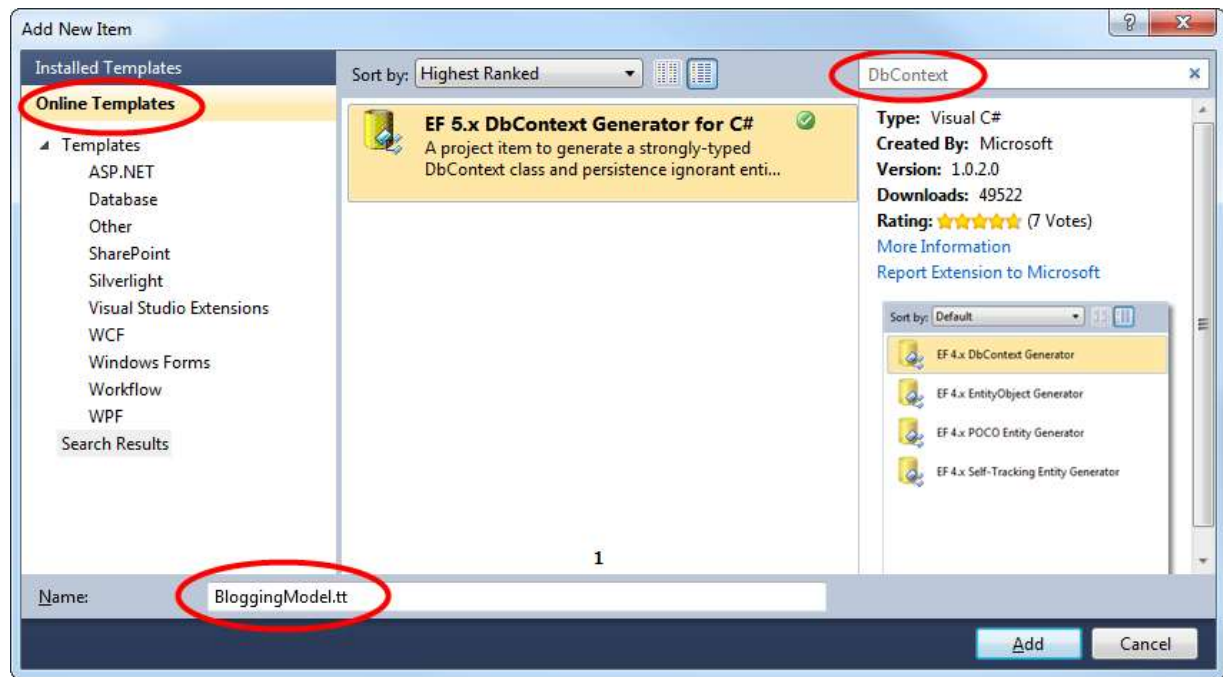
如果您在 Visual Studio 2010 中工作，則需要執行一些額外的步驟，才能升級至最新版本的 Entity Framework。升級很重要，因為它可讓您存取改良的 API 介面，這更容易使用，以及最新的錯誤修正。

首先，我們需要從 NuGet 取得最新版本的 Entity Framework。

- 專案 -> 管理 NuGet 套件... 如果您沒有 [管理 NuGet 套件...] 選項，您應該安裝 [最新版本的 NuGet](#)
- 選取 [線上] 索引 標籤
- 選取 EntityFramework 套件
- 按一下 [安裝]

接下來，我們需要交換模型，以產生使用 DbCoNtext API 的程式碼，而 DbCoNtext API 是在更新版本的 Entity Framework 中引進的。

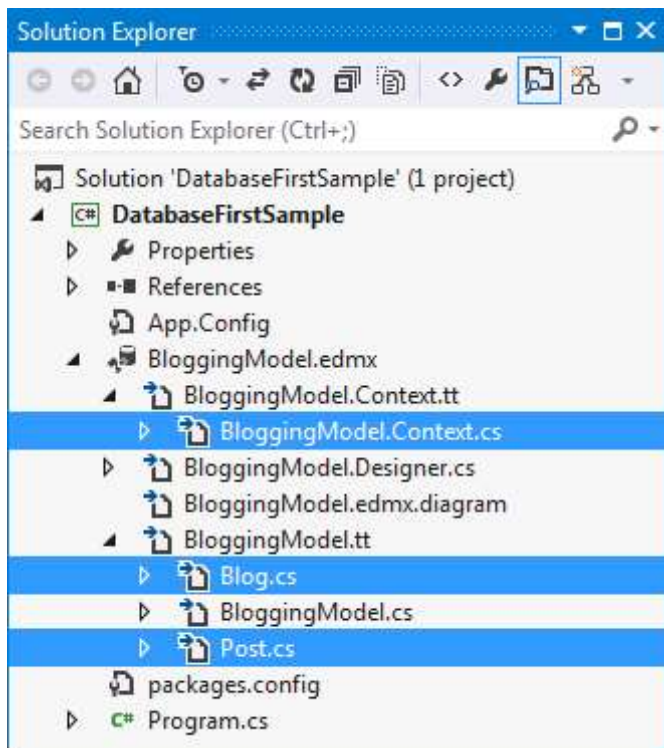
- 以滑鼠右鍵按一下 EF Designer 中模型的空白位置，然後選取 [新增程式碼產生專案...]
- 從左側功能表中選取 [線上範本]，然後搜尋 DbCoNtext
- 選取適用於 C# 的 EF 5.x DbCoNtext 產生器，輸入 BloggngModel 作為名稱，然後按一下 [新增]



4.讀取 & 寫入資料

既然我們有模型，現在是時候使用它來存取某些資料了。我們將用來存取資料的類別會根據 EDMX 檔案自動為您產生。

此螢幕擷取畫面來自 Visual Studio 2012，如果您使用 Visual Studio 2010，*BloggingModel.tt* 和 *BloggingModel.CoNtext.tt* 檔案會直接位於您的專案底下，而不是巢狀在 EDMX 檔案底下。



在 Program.cs 中實作 Main 方法，如下所示。此程式碼會建立內容的新實例，然後使用它來插入新的部落格。然後使用 LINQ 查詢，從依 Title 依字母順序排序的資料庫擷取所有部落格。

C#

```
class Program
{
    static void Main(string[] args)
    {
        using (var db = new BloggingContext())
        {
            // Create and save a new Blog
            Console.WriteLine("Enter a name for a new Blog: ");
            var name = Console.ReadLine();

            var blog = new Blog { Name = name };
            db.Blogs.Add(blog);
            db.SaveChanges();

            // Display all Blogs from the database
            var query = from b in db.Blogs
                        orderby b.Name
                        select b;

            Console.WriteLine("All blogs in the database:");
            foreach (var item in query)
```

```
{
    Console.WriteLine(item.Name);
}

Console.WriteLine("Press any key to exit...");
Console.ReadKey();
}
}
```

您現在可以執行應用程式並進行測試。

主控台

```
Enter a name for a new Blog: ADO.NET Blog
All blogs in the database:
ADO.NET Blog
Press any key to exit...
```

5.處理資料庫變更

現在是時候對資料庫架構進行一些變更，當我們進行這些變更時，我們也需要更新模型以反映這些變更。

第一個步驟是對資料庫架構進行一些變更。我們要將 Users 資料表新增至架構。

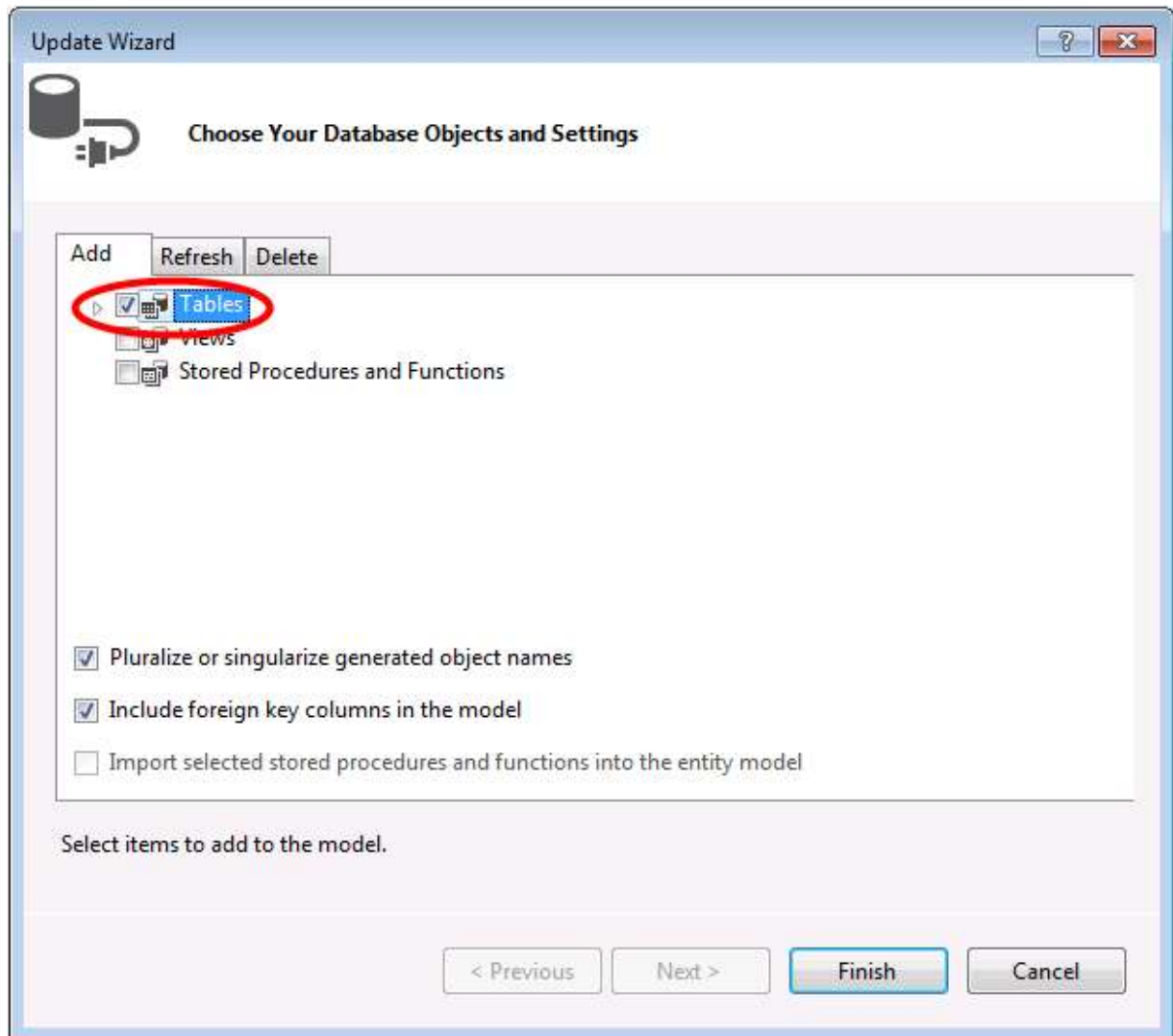
- 以滑鼠右鍵按一下 [伺服器總管] 中的 DatabaseFirst.Blogging 資料庫，然後選取 [新增查詢]
- 將下列 SQL 複製到新的查詢，然後以滑鼠右鍵按一下查詢，然後選取 [執行]

SQL

```
CREATE TABLE [dbo].[Users]
(
    [Username] NVARCHAR(50) NOT NULL PRIMARY KEY,
    [DisplayName] NVARCHAR(MAX) NULL
)
```

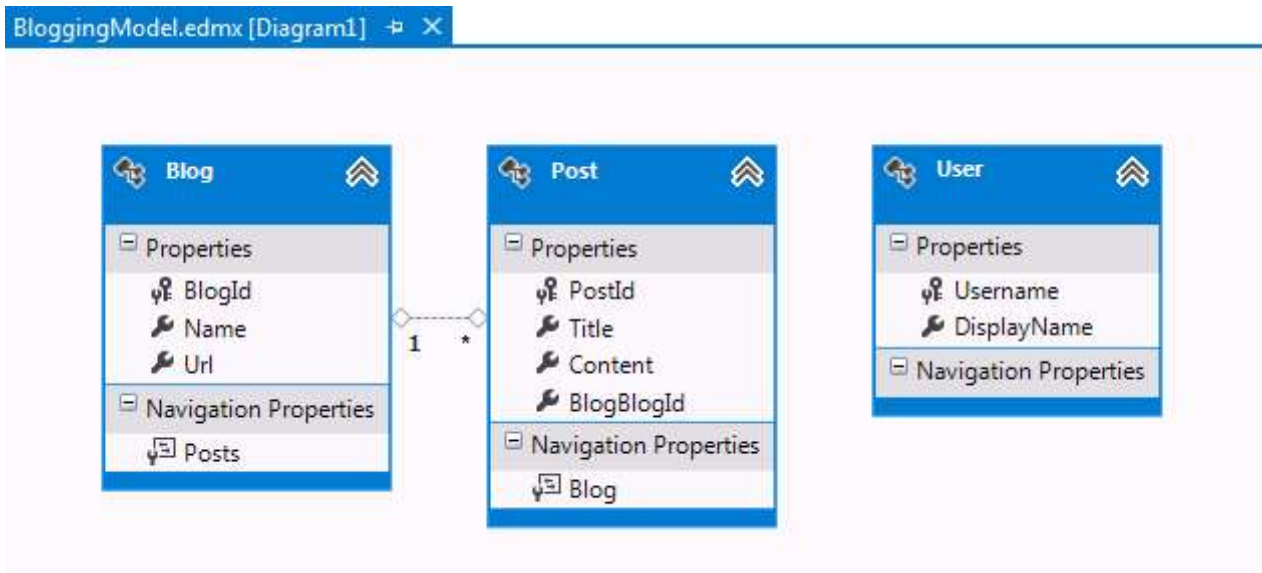
現在架構已更新，現在是時候使用這些變更來更新模型了。

- 以滑鼠右鍵按一下 EF 設計工具中模型的空白位置，然後選取 [從資料庫更新模型...]，這會啟動 [更新精靈]
- 在 [更新精靈] 的 [新增] 索引標籤上，核取 [資料表] 旁的方塊，這表示我們想要從架構新增任何新的資料表。[重新整理] 索引標籤會顯示模型中將檢查更新期間是否有變更的任何現有資料表。[刪除] 索引標籤會顯示已從架構中移除的任何資料表，也會從模型移除，作為更新的一部分。系統會自動偵測這兩個索引標籤上的資訊，而且僅供參考之用，您無法變更任何設定。



- 按一下 [更新精靈] 上的 [完成]

模型現在會更新為包含新的 User 實體，該實體會對應至我們新增至資料庫的 Users 資料表。



摘要

在本逐步解說中，我們已探討 Database First 開發，這可讓我們根據現有的資料庫，在 EF Designer 中建立模型。然後，我們使用該模型從資料庫讀取和寫入一些資料。最後，我們更新了模型，以反映我們對資料庫架構所做的變更。