# Creating a new Entity Data Model using VS 2013 to work with Entity Framework 6.1.3

by **Paul Chang** on May 1, 2017 in Tutorials, **Technology**, **UniData**, **Universe**

Entity Framework 4.0 or 5.0 is part of old Visual Studio (VS) versions. When upgrading to VS 2013 or later versions, it creates a .NET Framework 4.5 application to work with the newer Entity Framework 6.1.x version. It requires the Manage NuGet Packages tool to install it and it also requires an additional registering process to recognize the Rocket U2 Toolkit driver.

Before you can begin using your U2 accounts in SQL-based applications, you must make the U2 data accessible to those applications. It requires Visual Schema Generator (VSG) to create new sub-table or view schema on UniData accounts. On UniVerse, it needs to run the HS.ADMIN tool to activate a UniVerse account for dynamic normalization or to create a new schema on the account. The HS.SALES and HS.SERVICE accounts have been activated during UniVerse installation by default.

## Requirements

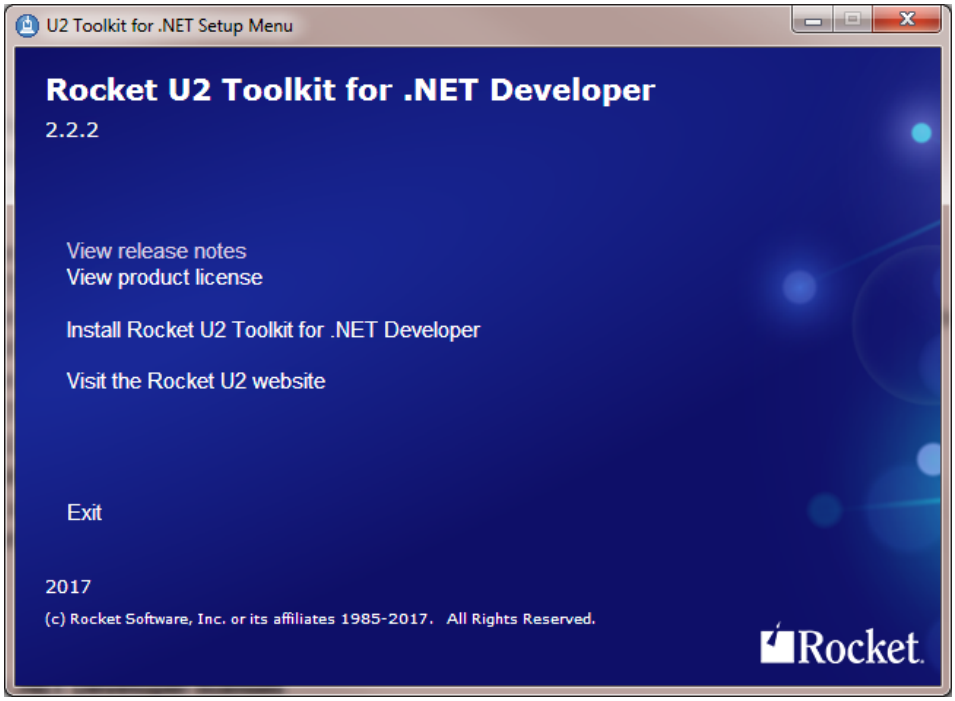
U2 Toolkit for .NET Developer 2.2.2
U2 Toolkit Developer license
UniVerse 11.1 or UniData 7.3 or later
Visual Studio 2013, 2015, or 2017
Microsoft.NET Framework 4.5 or later
ADO.NET Entity Framework 6.1.x
Preparing U2 accounts using UniData VSG or UniVerse HS.ADMIN tool

## Installing U2 Toolkit for .NET Developer and Licensing

The Rocket U2 Toolkit for .NET Developer software cannot be found on the Rocket Product availability web site. You must order the product with the U2 Toolkit Developer license via RBC  or through your value-added-reseller (VAR).  When you install Rocket U2 Toolkit

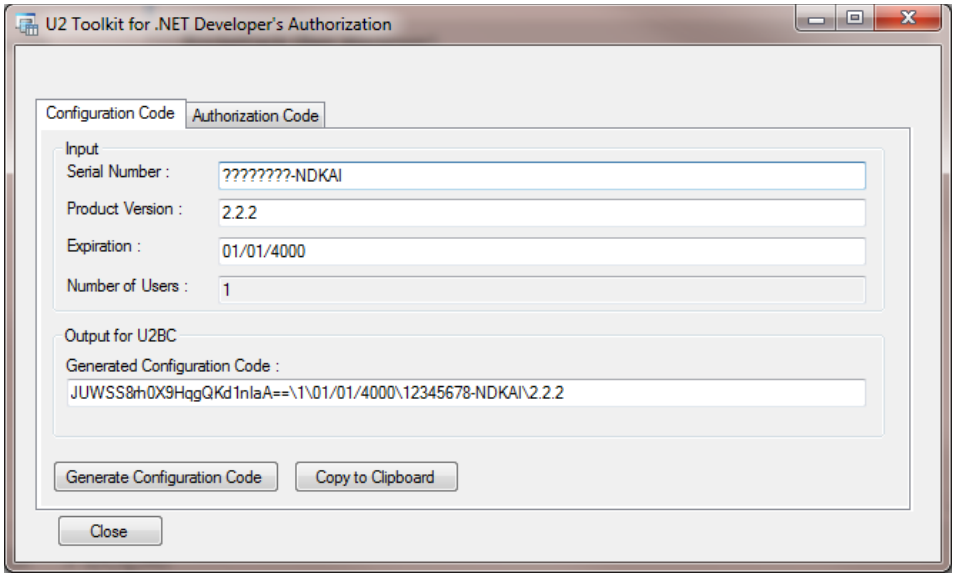
for .NET Developer software, it also installs 32-bit and 64-bit Rocket U2 Toolkit for .NET Provider software.

The software installation and licensing information can be found in the U2 Toolkit manual. The installation setup menu looks like the following:




The U2 Toolkit for .NET Developer's Authorization tool can be found in the Rocket U2 sub folder. You must have a U2 Toolkit developer serial number for authorization from the RBC team. The authorization code can be generated on the Rocket Business Connect web site - https://rbc.rocketsoftware.com/

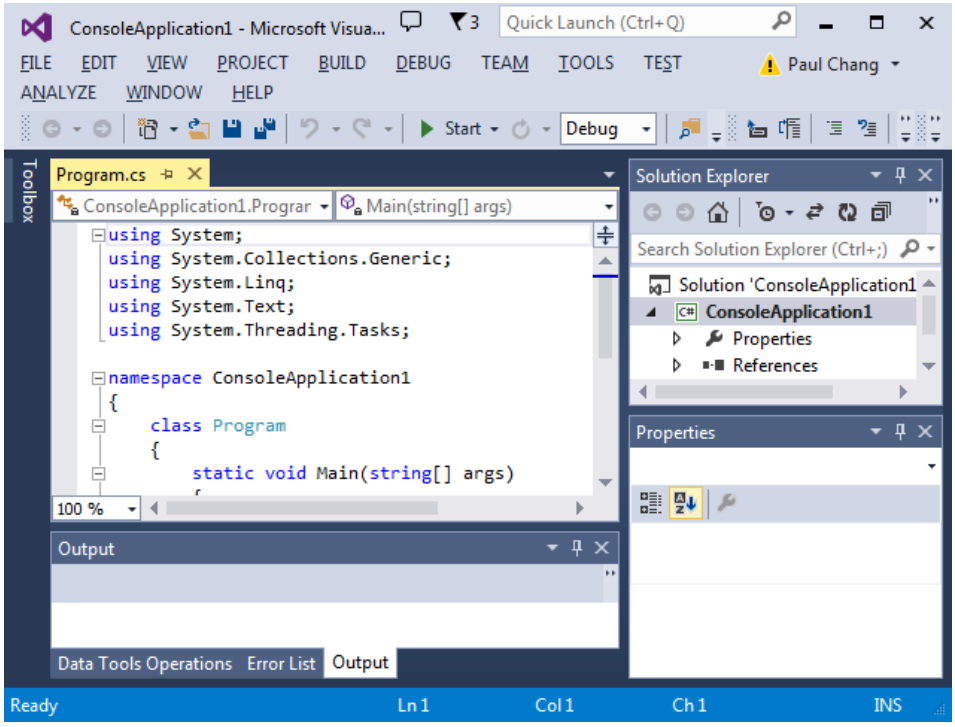
The U2 Toolkit for .NET Developer's Authorization tool is run as the following output.




# Procedure

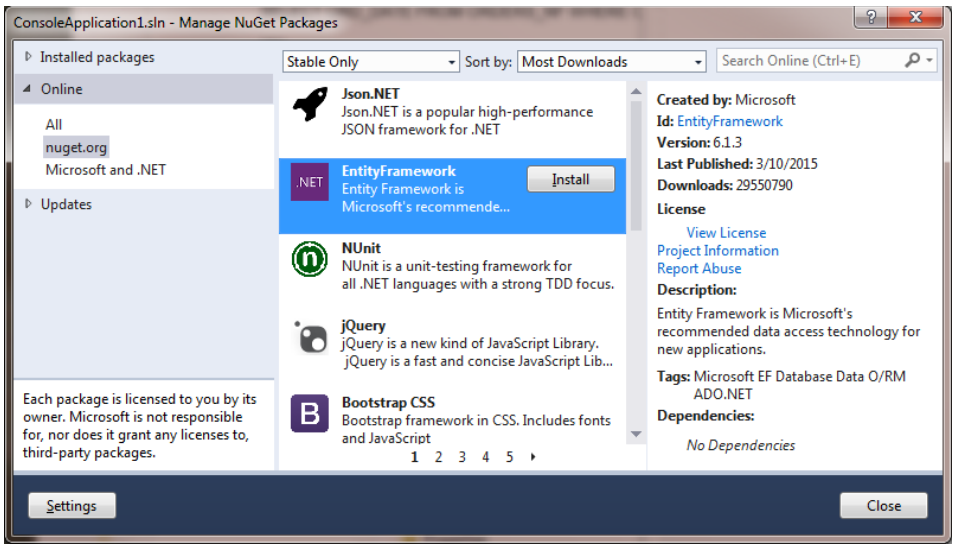
When you use Visual Studio 2013 to create a new C# application, it will use a target .NET Framework 4.5. On newer VS 2015 and VS 2017, it generates a .NET Framework 4.6 application by default that needs a different U2 Toolkit 4.6 driver to work with. This blog provides simple steps to create a C# console program on a VS 2013 environment.

1. Open a project in Visual Studio. This project was created in Visual Studio 2013.
2. Select the programming language you want to work in. The examples in this document are all created using C#.
3. Select File > New Project.
4. When the New Project dialog box opens, select Windows / Console Application.
5. In the name field, enter a name for the project. The project name in this example is ConsoleApplication1.
6. In the location field, enter the location where the project will reside. The location in this example is C:\temp.
7. Click OK. It will create a .NET Framework 4.5 application by default.




8. On new Entity Framework, 6.1.3 started shipping independently. In the Visual Studio 2013 project, it requires the Manage NuGet Packages tool. From the TOOLS menu choose the Manage NuGet Packages for solution to EntityFramework 6.1.3 package to the project. New entityFramework information will be added to the App.config file.

**Commented [KL1]:** This section needs additional punctuation and is possibly missing some words – it need to be more clear. I'm not sure how to make it more clear since I don't understand – sorry!




9. Add a new Class object to the project and put the “Customer.cs” in the Name field and update the Customer.cs file with the sample code.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel.DataAnnotations;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
namespace ConsoleApplication1
{
    public class Customer
    {
        Customer()
        {
        }
        [Key]
        public int CUSTID { get; set; }
        public string FNAME { get; set; }
        public string LNAME { get; set; }
        public string FULLADDR { get; set; }
    }
    public class CustomerContext : DbContext
    {
        public CustomerContext()
        {
        }
        public DbSet<Customer> Customers { get; set; }
        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            modelBuilder.Entity<Customer>()
                .Property(s => s.FNAME)
                .IsRequired();
            modelBuilder.Conventions.Remove<IncludeMetadataConvention>();
        }
    }
}
```

Note: You must add the CustomerContext connection string with U2 server credential information in the app.config configuration file.

10. Update the App.config file for the CutomerContext connection string to find the Customer entity. In the connection string, you need to specify the provider name “U2.Data.Client.4.5” so the entry is defined in the providers list.

**Commented [KL2]:** I made edits to this sentence to make it more clear. Did I make it more clear or did I change the meaning? I do not want to change the meaning.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection,
EntityFramework, Version=6.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" requirePermission="false"/>
  </configSections>
  <connectionStrings>
    <add name="CustomerContext" connectionString="Database=HS.SALES;User
ID=administrator;Password=password;Server=localhost;Pooling=false;ServerType=unive
rse;ConnectTimeout=360;PersistSecurityInfo=true"
providerName="U2.Data.Client.4.5"/>
  </connectionStrings>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5"/>
  </startup>
  <entityFramework>
    <providers>
      <provider invariantName="U2.Data.Client.4.5"
type="U2.Data.Client.Entity.U2ProviderServices, U2.Data.Client.Entity,
Version=2.2.2.0, Culture=neutral, PublicKeyToken=883335d992998a08"/>
    </providers>
  </entityFramework>
</configuration>
```

Note: For the U2.Data.Client.Entity 4.6 driver, the PublicKeyToken is “7f1dc11a3fe611eb”. You can use the VS 2013 x86 native sn command with the “-T” option to find the public token information.

11. Add the SqlQuery statement to the Customer entity in the Program.cs file.

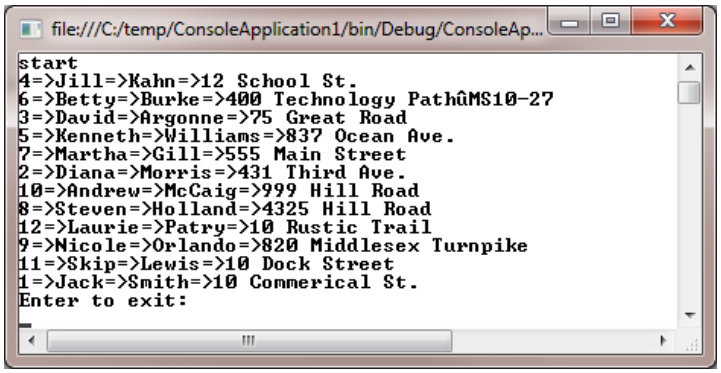
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                Console.WriteLine("start");
                CustomerContext ctx = new CustomerContext();
                var t = ctx.Database.SqlQuery<Customer>("SELECT
CUSTID,FNAME,LNAME,FULLADDR FROM CUSTOMER");
                foreach (Customer item in t)
                {
                    Console.WriteLine(item.CUSTID + "=>" + item.FNAME +
"=>" + item.LNAME + "=>" + item.FULLADDR);
                }
            }
            catch (Exception e)
            {
                if (e.InnerException != null)

               {Console.WriteLine(e.InnerException.Message);}
                else
                {Console.WriteLine(e.Message);}
            }
            finally
            {
                Console.WriteLine("Enter to exit:");
                string line = Console.ReadLine();
            }
        }
    }
}
```

**Formatted:** German (Germany)

# Console application result

The sample program should return the entity query as the following output.




## Paul Chang

Paul Chang is a Senior Technical Support Engineer at Rocket Software. He is responsible for supporting all U2 common middleware client products, Java products and .NET clients. He has 35 years of experience in computing and holds a master's degree in Computer Science with an emphasis on Relational Database Design.