title: "Deploy a project with .NET code (C#) | Microsoft Docs"  
ms.date: "08/21/2017"  
ms.topic: "article"  
ms.prod: "sql-server-2017"  
ms.technology:

* "integration-services"  
  author: "douglaslMS"  
  ms.author: "douglasl"  
  manager: "craigg"  
  ---

# Deploy an SSIS project with C# code in a .NET app

* This quick start tutorial demonstrates how to write C# code to connect to an Azure SQL database and deploy an SSIS project.

You can use Visual Studio, Visual Studio Code, or another tool of your choice to create a C# app.

[!NOTE] Only the project deployment model is supported. For more info about SSIS deployment, and about converting a project to the project deployment model, see [Deploy Integration Services (SSIS) Projects and Packages](https://docs.microsoft.com/en-us/sql/integration-services/packages/deploy-integration-services-ssis-projects-and-packages.md).

## Prerequisites

Before you start, make sure you have Visual Studio or Visual Studio Code installed. Download the free Community edition of Visual Studio, or the free Visual Studio Code, from [Visual Studio Downloads](https://www.visualstudio.com/downloads/).

You also have to have a server-level firewall rule for the public IP address of the computer you use for this quick start tutorial. To create a server-level firewall rule, see [Create a server-level firewall rule](sql-database-get-started-portal.md#create-a-server-level-firewall-rule).

## Get the SSISDB connection information

Get the connection information you need to connect to the SSIS Catalog database (SSISDB) on SQL database. You need the fully qualified server name and login information in the next procedures.

1. Log in to the [Azure portal](https://portal.azure.com/).
2. Select **SQL Databases** from the left-hand menu, and click the SSISDB database on the **SQL databases** page.
3. On the **Overview** page for your database, review the fully qualified server name. To bring up the **Click to copy** option, hover over the server name .
4. If you forget your Azure SQL Database server login information, navigate to the SQL Database server page to view the server admin name. You can reset the password if necessary.
5. Click **Show database connection strings**.
6. Review the complete **ADO.NET** connection string. The sample code uses a SqlConnectionStringBuilder to recreate this connection string with the individual parameter values that you provide.

## Create a new Visual Studio project

1. In Visual Studio, choose **File**, **New**, **Project**.
2. In the **New Project** dialog, and expand **Visual C#**.
3. Select **Console App** and enter *deploy\_ssis\_project* for the project name.
4. Click **OK** to create and open the new project in Visual Studio.

## Add references

1. In Solution Explorer, right-click the **References** folder and select **Add Reference**. The **Reference Manager** dialog box opens.
2. In the **Reference Manager** dialog box, expand **Assemblies** and select **Extensions**.
3. Select the following two references to add:
   * Microsoft.SqlServer.Management.Sdk.Sfc
   * Microsoft.SqlServer.Smo
4. Click the **Browse** button to add a reference to **Microsoft.SqlServer.Management.IntegrationServices**. (This assembly is installed only in the global assembly cache (GAC).) The **Select the files to reference** dialog box opens.
5. In the **Select the files to reference** dialog box, navigate to the GAC folder that contains the assembly. Typically this is C:\Windows\assembly\GAC\_MSIL\Microsoft.SqlServer.Management.IntegrationServices\14.0.0.0\_\_89845dcd8080cc91.
6. Select the assembly (that is, the .dll file) in the folder and click **Add**.
7. Click **OK** to close the **Reference Manager** dialog box and add the three references. Check the **References** list in Solution Explorer to make sure the references are there.

## Add the C# code

1. Open **Program.cs**.
2. Replace the contents of **Program.cs** with the following code. Add the appropriate values for your server, database, user, and password.

using Microsoft.SqlServer.Management.IntegrationServices;  
using System;  
using System.Data.SqlClient;  
using System.IO;  
  
namespace deploy\_ssis\_project  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 // Variables  
 string targetServerName = "localhost";  
 string targetFolderName = "Project1Folder";  
 string projectName = "Integration Services Project1";  
 string projectFilePath = @"C:\Projects\Integration Services Project1\Integration Services Project1\bin\Development\Integration Services Project1.ispac";  
  
 // Create a connection to the server  
 string sqlConnectionString = "Data Source=" + targetServerName +  
 ";Initial Catalog=master;Integrated Security=SSPI;";  
 SqlConnection sqlConnection = new SqlConnection(sqlConnectionString);  
  
 // Create the Integration Services object  
 IntegrationServices integrationServices = new IntegrationServices(sqlConnection);  
  
 // Get the Integration Services catalog  
 Catalog catalog = integrationServices.Catalogs["SSISDB"];  
  
 // Create the target folder  
 CatalogFolder folder = new CatalogFolder(catalog,  
 targetFolderName, "Folder description");  
 folder.Create();  
  
 Console.WriteLine("Deploying " + projectName + " project.");  
  
 byte[] projectFile = File.ReadAllBytes(projectFilePath);  
 folder.DeployProject(projectName, projectFile);  
  
 Console.WriteLine("Done.");  
 }  
 }  
}

## Run the code

1. Press **F5** to run the application.
2. In SSMS, verify that the project has been deployed.

## Next steps

* Run a package. To run a package, you can choose from several tools and languages. For more info, see the following articles:
  + [Run from SSMS](ssis-everest-quickstart-run-ssms.md)
  + [Run with T-SQL from SSMS](ssis-everest-quickstart-run-tsql-ssms.md)
  + [Run with T-SQL from VS Code](ssis-everest-quickstart-run-tsql-vscode.md)
  + [Run from command prompt](ssis-everest-quickstart-run-cmdline.md)
  + [Run from PowerShell](ssis-everest-quickstart-run-powershell.md)
  + [Run from C# app](ssis-everest-quickstart-run-dotnet.md)
* Schedule a package. For more info, see [Schedule page](ssis-everest-howto-schedule-package.md)