title: "Deploy a project with Transact-SQL (SSMS) | 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 from SSMS with Transact-SQL

This quick start demonstrates how to use SQL Server Management Studio (SSMS) to connect to the SSIS Catalog database on an Azure SQL database server, and then use Transact-SQL statements to deploy an SSIS project to the SSIS Catalog.

SQL Server Management Studio is an integrated environment for managing any SQL infrastructure, from SQL Server to SQL Database. For more info about SSMS, see [SQL Server Management Studio (SSMS)](../ssms/sql-server-management-studio-ssms.md).

[!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 the latest version of SQL Server Management Studio. To download SSMS, see [Download SQL Server Management Studio (SSMS)](https://docs.microsoft.com/sql/ssms/download-sql-server-management-studio-ssms).

## Connect to the SSIS Catalog database

Use SQL Server Management Studio to establish a connection to the SSIS Catalog on your Azure SQL Database server.

[!IMPORTANT]  
An Azure SQL Database server listens on port 1433. If you are attempting to connect to an Azure SQL Database server from within a corporate firewall, this port must be open in the corporate firewall for you to connect successfully.

1. Open SQL Server Management Studio.
2. In the **Connect to Server** dialog box, enter the following information:

|  |  |  |
| --- | --- | --- |
| Setting | Suggested value | Description |
| **Server type** | Database engine | This value is required. |
| **Server name** | The fully qualified server name | The name should be something like this: **mysqldbserver.database.windows.net**. |
| **Authentication** | SQL Server Authentication | This quickstart uses SQL authentication. |
| **Login** | The server admin account | This is the account that you specified when you created the server. |
| **Password** | The password for your server admin account | This is the password that you specified when you created the server. |

1. Click **Connect**. The Object Explorer window opens in SSMS.
2. In Object Explorer, expand **Integration Services Catalogs** and then expand **SSISDB** to view the objects in the SSIS Catalog database.

## Run the T-SQL code

Run the following Transact-SQL code to deploy an SSIS project.

1. In SSMS, open a new query window and paste the following code.
2. Update the parameter values in the catalog.deploy\_project stored procedure for your system.
3. Make sure that SSISDB is the current database.
4. Run the script.
5. In Object Explorer, refresh the contents of **SSISDB** if necessary and check for the project that you just deployed.

DECLARE @ProjectBinary AS varbinary(max)  
DECLARE @operation\_id AS bigint  
SET @ProjectBinary = (SELECT \* FROM OPENROWSET(BULK '<project\_file\_path>.ispac', SINGLE\_BLOB) AS BinaryData)  
  
EXEC catalog.deploy\_project @folder\_name = '<target\_folder>', @project\_name = '<project\_name',  
 @Project\_Stream = @ProjectBinary, @operation\_id = @operation\_id out

## 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)