title: "Deploy a project with PowerShell | 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 PowerShell

* This quick start tutorial demonstrates how to use a PowerShell script to connect to an Azure SQL database and deploy an SSIS project.

[!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).

## PowerShell script

Provide appropriate values for the variables at the top of the following script, and then run the script to deploy the SSIS project.

# Variables  
$SSISNamespace = "Microsoft.SqlServer.Management.IntegrationServices"  
$TargetServerName = "localhost"  
$TargetFolderName = "Project1Folder"  
$ProjectFilePath = "C:\Projects\Integration Services Project1\Integration Services Project1\bin\Development\Integration Services Project1.ispac"  
$ProjectName = "Integration Services Project1"  
  
# Load the IntegrationServices assembly  
$loadStatus = [System.Reflection.Assembly]::Load("Microsoft.SQLServer.Management.IntegrationServices, "+  
 "Version=14.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91, processorArchitecture=MSIL")  
  
# Create a connection to the server  
$sqlConnectionString = `  
 "Data Source=" + $TargetServerName + ";Initial Catalog=master;Integrated Security=SSPI;"  
$sqlConnection = New-Object System.Data.SqlClient.SqlConnection $sqlConnectionString  
  
# Create the Integration Services object  
$integrationServices = New-Object $SSISNamespace".IntegrationServices" $sqlConnection  
  
# Get the Integration Services catalog  
$catalog = $integrationServices.Catalogs["SSISDB"]  
  
# Create the target folder  
$folder = New-Object $SSISNamespace".CatalogFolder" ($catalog, $TargetFolderName,  
 "Folder description")  
$folder.Create()  
  
Write-Host "Deploying " $ProjectName " project ..."  
  
# Read the project file and deploy it  
[byte[]] $projectFile = [System.IO.File]::ReadAllBytes($ProjectFilePath)  
$folder.DeployProject($ProjectName, $projectFile)  
  
Write-Host "Done."

## 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)