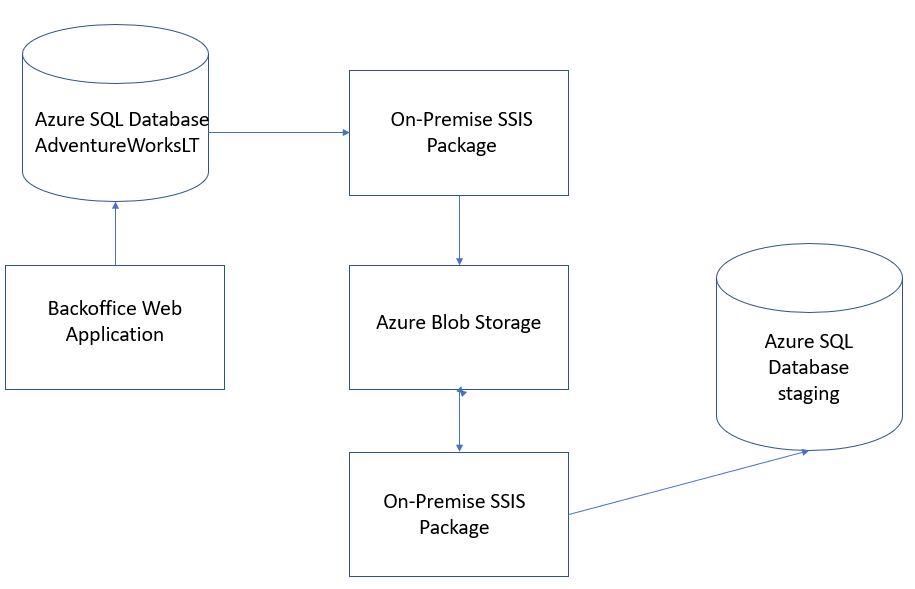
# Create a Hybrid Environment for SSIS Development

##### Problem

I am starting a new data warehousing project for a new client where I will use SQL Server Integration Services (SSIS) for the Extract, Transform and Load (ETL) operations. The client wants to create a hybrid environment for development where I will only install SQL Server Data Tools (SSDT) and SQL Server Management Studio (SSMS) on my laptop. The goal is to leverage Azure for file storage and the SQL Server database. Since this is a development environment, I will use an Azure SQL Database for Backoffice Web Application and staging databases. Ultimately my client wants to transition this ETL application to Azure Data Factory [CHANGE TO LINK] at some point in the future. Can you walk me through the steps to accomplish this?

##### Solution

The following diagram shows the simple solution to be built in this tip:



The main points are:

* An On-Premise SSIS package extracts data from the Backoffice Web Application’s Azure SQL Database and writes pipe-delimited files to Azure Blob Storage on a daily basis for the various types of transactions that are processed and of interest to the data warehouse
* An On-Premise SSIS package reads the pipe-delimited files from Azure Blob Storage and inserts the records into staging tables in the Azure SQL Database

Note that for demonstration purposes, I’m creating two SSIS packages to walk through the scenario where the Backoffice Web Application and the ETL process are loosely integrated. Certainly, I could create a single SSIS package that extracts data from the Backoffice Web Application’s database and inserts that data into the staging database. I want to demonstrate extracting data from an Azure SQL Database and writing to Azure Blob Storage as well as reading data from Azure Blob Storage and inserting into an Azure SQL Database.

## Prerequisites

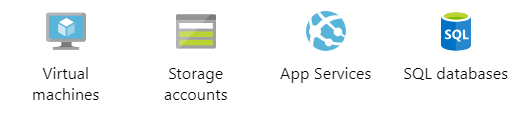
There are a couple of things that we need to take care of before we walk through building the on-premise SSIS package:

* Create Azure Subscription
* Create Azure Storage Account
* Create Azure SQL Database
* Install SQL Server Data Tools
* Install Azure Feature Pack for SQL Server Integration Services (SSIS)
* Install SQL Server Management Studio

I will provide a brief description of the setup steps for each of these assuming that you are starting from scratch.

#### Create Azure Subscription

Go to <https://azure.microsoft.com/> and setup an account. You can probably get a free subscription for a period of time or sign up for a pay-as-you-go plan. After you have an Azure Subscription, go to <https://portal.azure.com/> and sign in. You will see the following on the home page (just the relevant portion of the page is shown):



#### Create Azure Storage Account

Click on the Storage accounts icon on the Azure Home Page shown above to create an Azure Storage Account. Follow the steps in [Create a storage account](https://docs.microsoft.com/en-us/azure/storage/common/storage-quickstart-create-account?tabs=azure-portal) to get your storage account created. Make a note of the Storage account name; you will need it later.

#### Create Azure SQL Database

Click on the SQL databases icon on the Azure Home Page shown above to create an Azure SQL Database. Follow the steps in [Quickstart: Create a single database in Azure SQL Database using the Azure portal](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-single-database-get-started) to get your Azure SQL Database created. Make a note of the Server admin name and password when you create a new server; you will need it later.

After you create your Azure SQL Database, make sure to [Create a server-level firewall rule](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-server-level-firewall-rule) so you can access the database outside of the Azure portal.

Note that when you create a new Azure SQL Database, you have the following options for the Source:

* Blank database
* AdventureWorksLT
* Backup

For this tip I chose AdventureWorksLT for the application database and Blank database for the staging database.

#### Install SQL Server Data Tools (SSDT)

SSDT is a free, stand-alone version of Visual Studio that you can use to develop Analysis Services, SQL Server Integration Services and SQL Server Reporting Services project. Follow this link to [download and install SQL Server Data Tools (SSDT) for Visual Studio](https://docs.microsoft.com/en-us/sql/ssdt/download-sql-server-data-tools-ssdt?view=sql-server-2017).

#### Install Azure Feature Pack for SSIS

Download and install the [Azure Feature Pack for SSIS](https://docs.microsoft.com/en-us/sql/integration-services/azure-feature-pack-for-integration-services-ssis?view=sql-server-2017) which includes a handful of components for working with Azure from SSIS.

#### Install SQL Server Management Studio (SSMS)

SSMS is the tool we use to work with a SQL Server database. For this tip I am using version 14.0.17289.0. Follow this link to [download SQL Server Management Studio](https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017). I like to use the current general availability version. Note that if you have an older version of SSMS installed, it may not have the features for working with Azure SQL Databases.

## Get Azure Values

Before we start creating SSIS packages, we need to gather all of the parameter values from our Azure Storage Account and Azure SQL Database that we will need for this tip.

#### Azure Storage Account Values

Follow these steps to get the Azure Storage Account values we need:

1. Navigate to the [Azure home page](https://portal.azure.com/)
2. Click on Storage accounts
3. Select your storage account
4. Click on Access Keys (under Settings) and copy the key value under Key1
5. Click on Overview (top left)
6. Click on Blobs (under Services)
7. Select your Container name

The following are the parameter values for this tip based on the above steps (sensitive values shown as asterisks):

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Storage Account Name | backofficestaging |
| Access Key | \*\*\*\*\*\*\*\*\*\*\* |
| Container Name | subscriptions |

#### Azure SQL Database Values

I use two Azure SQL Databases in this tip: an application database and a staging database. Follow these steps to get the Azure SQL Database parameter values that we need:

1. Navigate to the [Azure home page](https://portal.azure.com/)
2. Click on SQL databases
3. Select the database server and database name

The following are the parameter values for this tip based on the above steps (sensitive values shown as asterisks):

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Application Database Server | \*\*\*\*\*\*\*\*\*\*\* |
| Application Database Name | AdventureWorksLT |
| User Name | \*\*\*\*\*\*\*\*\*\*\* |
| Password | \*\*\*\*\*\*\*\*\*\*\* |
| Staging Database Server | \*\*\*\*\*\*\*\*\*\*\* |
| Staging Database Name | staging |
| User Name | \*\*\*\*\*\*\*\*\*\*\* |
| Password | \*\*\*\*\*\*\*\*\*\*\* |

Note that the User Name and Password were specified when your created the server.

## Create SSIS Packages

Now we are ready to create the SSIS packages. Open SSDT and create a new SSIS project. Add new SSIS packages named EXTRACT\_SUBSCRIPTIONS SSIS and PROCESS\_SUBSCRIPTIONS SSIS.

I will walk through the following steps to build the SSIS packages:

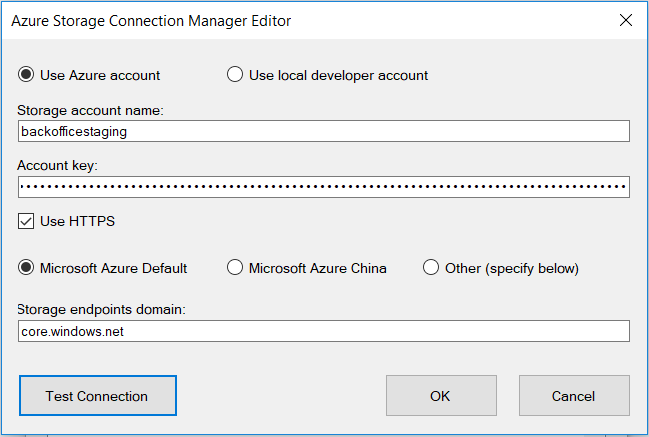
1. Create the Azure Storage Connection Manager
2. Create an OLEDB Connection Manager for an Azure SQL Database
3. Create the SUBSCRIPTIONS table in the staging Azure SQL Database
4. Create the EXTRACT\_SUBSCRIPTIONS SSIS package
5. Create the PROCESS\_SUBSCRIPTIONS SSIS package

#### Create the Azure Storage Connection Manager

We need to create an Azure Storage Connection Manager in order to save a delimited file to Azure Blob Storage and open a delimited file from Azure Blob Storage. Follow these steps to create the Azure Storage Connection Manager:

1. Click on the Control Flow tab in an SSIS package
2. Right-click in the Connection Managers area of the SSIS package designer
3. Select New Connection from the popup menu
4. Select Azure Storage as the Connection Type
5. Click Add

The following form will be displayed (fill in the Storage account name and Account key from the Azure Storage Account Values section above):



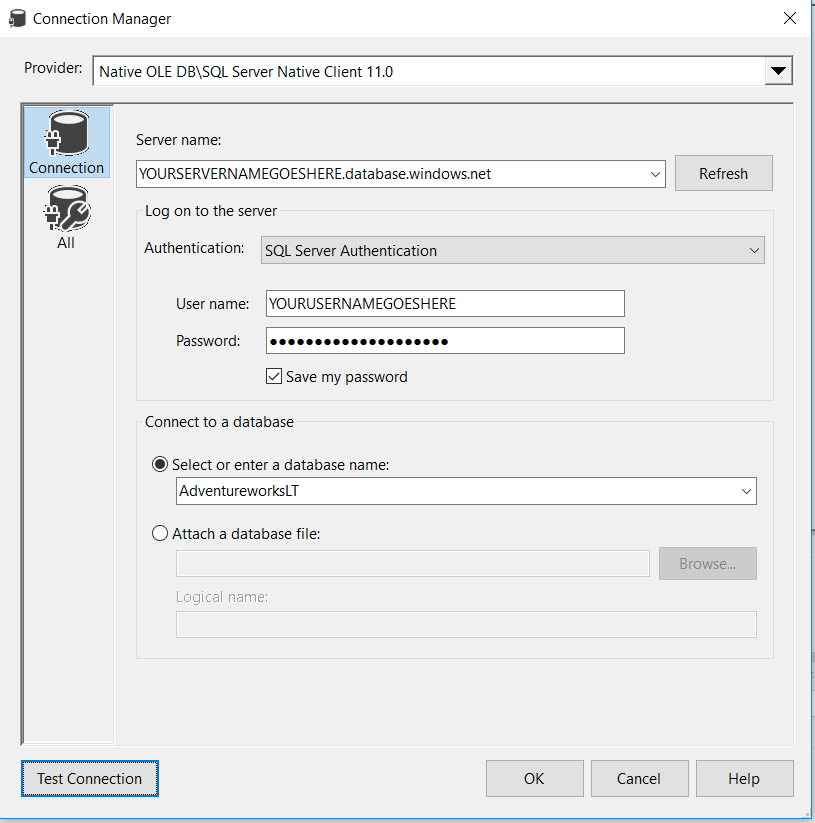
Click Test Connection to make sure you can connect to your Azure Storage Account. Click OK to save the Azure Storage Connection Manager. Finally right click on the connection manager and select Conver to Project Connection. This connection manager will be used in both of the SSIS packages we will create.

#### Create an OLEDB Connection Manager for an Azure SQL Database

There really isn’t much difference between creating an OLEDB connection manager for an Azure SQL Database and an OLEDB connection manager for an on-premise SQL Server database. Follow these steps to create an OLEDB connection manager for an Azure SQL Database:

1. Click on the Control Flow tab in an SSIS package
2. Right-click in the Connection Managers area of the SSIS package designer
3. Select New OLE DB Connection from the popup menu
4. Select New from the OLE DB Connection Manager form

The following form will be displayed (fill in the YOURXXX with your values from the Azure SQL Database Values section above):



Note that the only difference for an Azure SQL Database connection versus an on-premise SQL Server database is that the server name has “.database.windows.net” appended to it.

Click Test Connection to verify that your connection manager works. Click OK to complete the Connection Manager form. Click OK again on the OLE DB Connection Manager form. Right-click on the Connection Manager and select Rename; change to AdventureWorksLT. Right-click on the Connection Manager and select Convert to Project Connection so that the connection manager will be available for use in any SSIS package in the project.

Repeat the above steps to create the staging connection manager.

#### Create the SUBSCRIPTIONS table in the staging Azure SQL Database

#### Create the EXTRACT\_SUBSCRIPTIONS SSIS package

#### Create the PROCESS\_SUBSCRIPTIONS SSIS package

To use this file as a template directly in Word, save the file to Root\Users\Your\_Profile\Documents\Custom Office Templates. Then when you open Word you can access the file as a Personal Theme.

For the file name of your tip, do not save with any spaces. Please use dashes instead of spaces.

For any long scripts, backups, SSRS Reports, Project files, etc. Please save them as an \*.zip file and submit them with this file.

Images no longer need to be saved as separate files.

## Tip Formatting

In terms of formatting, use the Emphasis and Strong Styles in the Home ribbon.

For URLs, highlight the text, right click on the text and select the Hyperlink option. In the interface that loads, enter the URL in the Address field.

## Image Example

Save the image directly into this document, do not save the image files separately.

Click on each image and then click on the Image Styles in the Home ribbon to properly mark the image.

Images need to be 7.5 inches wide or less.

Add a title to each image by right clicking on the image and select Format Picture. On the Format Text Effects interface select the Layout & Properties option. Then click on Alt Text to enter the Title and Description.



## Table Example

If a table is needed in your tip, please use the table below as a starting point and adjust the rows and columns by right clicking on the table then selecting the Insert or Delete options.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Code Example

Once you have your code completed in Management Studio. Copy and paste it into this file and mark the section as a Code Style in the Home ribbon.

For a new line, use the Shift + Enter to go to a new line.

Do not use Font Styles for code colors. This will be handled by the style sheet for T-SQL code.

SELECT T.\*, A.\*   
FROM dbo.TestT T  
INNER JOIN dbo.TestA A  
ON T.ID = A.ID  
WHERE T.Column1 = 'MSSQLTips'  
GO

##### Next Steps

* Please outline the next steps for the reader to implement the tip.
* Reference as many existing tips as possible that the reader should review.