**SQL Server SSISDB Catalog Project Migration:**

1. Server

2. Migrating Catalog Project

3. Migrating Catalog Environment

4. Service Account

5. Migrating SQL Agent Job

**SQL Server and SSISDB Summary:**

a. On-Premise Server – Source Server

b. Azure VM Server – Target Server

c. Over view of SSISDB setup in Source Server

1. Server:

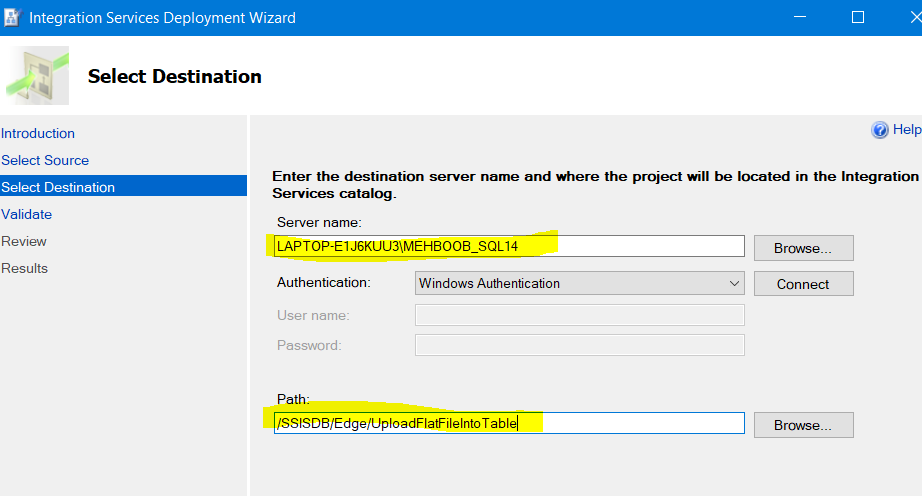
Source: lda-pdb01 – LDA\gcoopman

Target: devsqli01.lda.local – LDA\gcoopman

2. Migrating Catalog Project

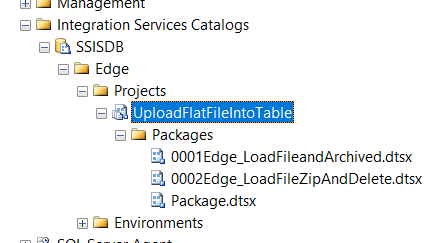
a. Open Source Project in Visual Studio

b. Under [Solution Explorer] right click on Project and Select [Deploy]



c. change the Server name to Target Server

d. On Target Server using SQL Server Management Studio create folder same as Source folder under SSISDB Catalog



If you don’t have rights on Target server you will get error

Now Deploy the project and if everything is OK you will see your project on Target Server

ENJOY!!!

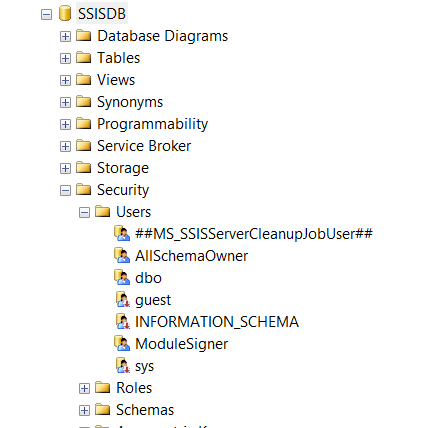
Extra:

To run SSIS Service you have to have two account

a. User/group

b. service account svcSQLBI account

This is my test server and there is no service account – I have to investigate on it….



**Migrating SSISDB Catalog Environment:**

a. Using Script Stored Procedures

<http://www.sqlservercentral.com/articles/Integration+Services+(SSIS)/135173/>

1. Create Stored Procedure <dbo.usp_SSIS_ScriptEnvironment.sql> for testing an environment that has at least one variable for each type.

2. EXEC dbo.usp\_SSIS\_ScriptEnvironment @folder = 'Test', @env = 'Test1'

<GenerateSSISEnvironmentandVariables.sql>

If I want to use this script to create a duplicate environment named ‘Test2’ to a new ‘Test2’  folder on my dev box.  I need to change the @folder and @env variables, provide values for any sensitive values, and make any other changes to the variables.

<ScriptForCreatingDuplicateEnvironment.sql>

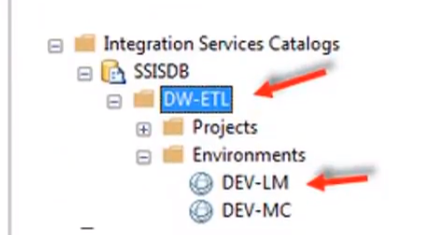
**Note:**

To Create Environment on Source Server and Target Server

1. run [dbo.usp\_SSIS\_ScriptEnvironment.sql](file:///C:\TechnicalExp\SSIS\Edge\Documents\dbo.usp_SSIS_ScriptEnvironment.sql) on source Server using EXEC dbo.usp\_SSIS\_ScriptEnvironment @folder = 'Test', @env = 'Test1'

2. Now Go to Target Server and run this script

EXEC dbo.usp\_SSIS\_ScriptEnvironment @folder = 'DW-ETL', @env = 'DEV-LM' (LM is for Target Server)



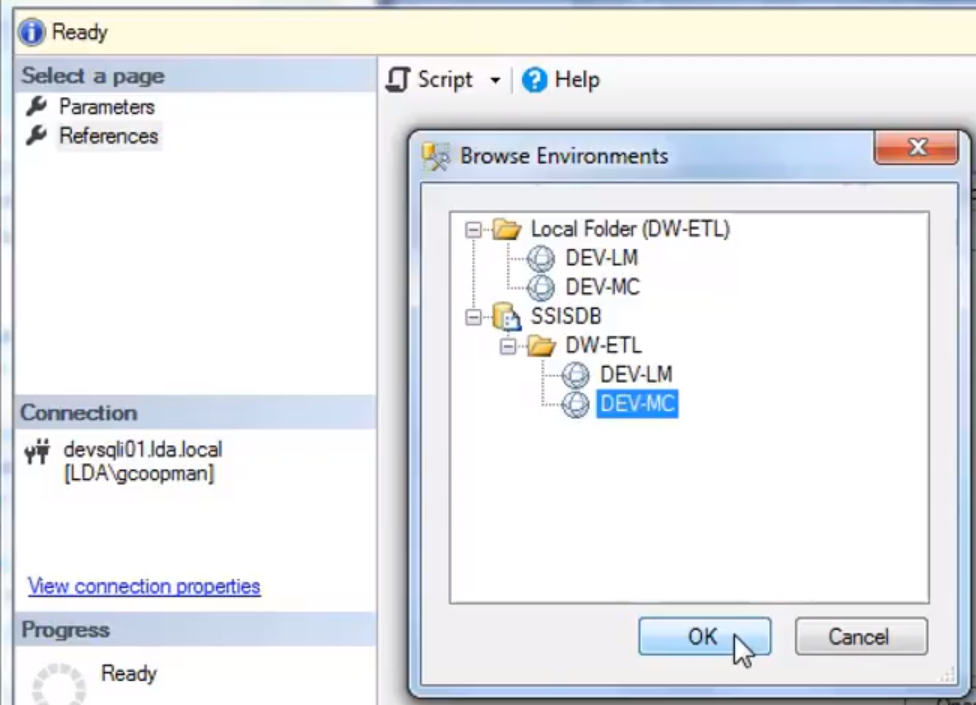
EXEC dbo.usp\_SSIS\_ScriptEnvironment @folder = 'DW-ETL', @env = 'DEV-MC' (Mc is another Environment on Target server)

**Note: LM and MC are two different Projects for Finance (Finance\_LM, Finance\_MC)**

In our example we have DW-ETL folder under SSISDB that has Projects for those Projects we have created Environment “DEV-LM” and then we are migrating on Azure Machin for that we have created “DEV-MC” environment

Now link Environment variables with the Project, you need to configure and provide reference to the Environment

a. Go to your Target server SSISDB folder and provide references and assigned Parameters (LM and MC are two different environments)



b. Understanding and Running Script

c. Add Environment Reference to Project

d. Map Project Parameters to Environment variables

**Service Account:**

1. Assign to local databases

2. Assign Permissions and roles in local databases

3. Create logins and Permissions to external sources

4. Create Credential with Service Account

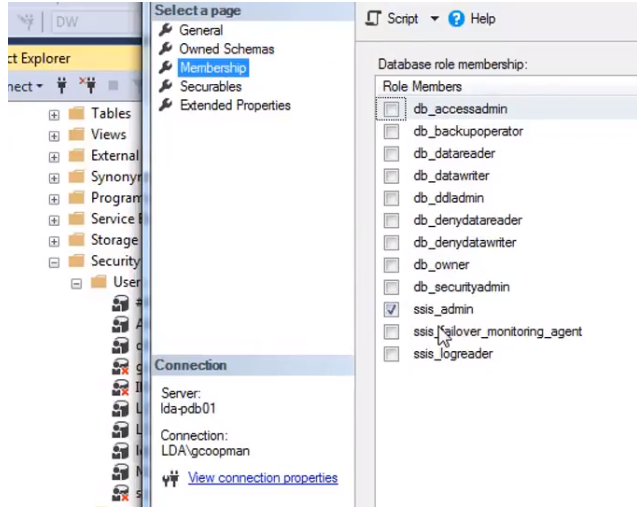
5. Create SQL Agent Proxy from Credential

**Create Credential-Proxi: Credential**

Create Job under Proxi, to do Proxi you have to create Credential

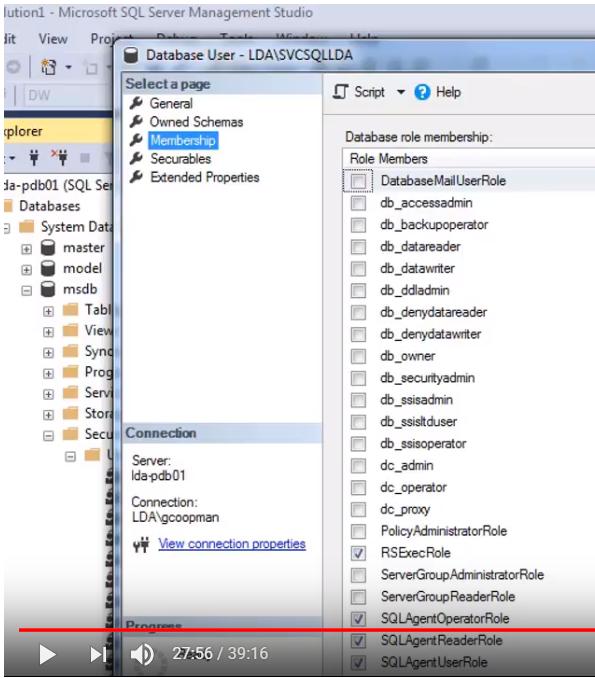
Security – Credential – Create one as ETL Credential and use **service account** that will create Proxi and Proxi will run the Job (Packages)

* Service Account: must be in SSISDB-Security-Users (lda\svcSQLBI)



And must be member of SSIS\_admin

* Service Account must also be member of MSDB- Security - Users – it run SQL Agent Job

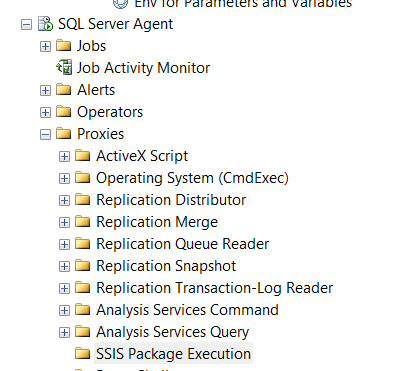


(This is on Source Server)

SSISDB use setup as Admin

Service Account : must have domain permission

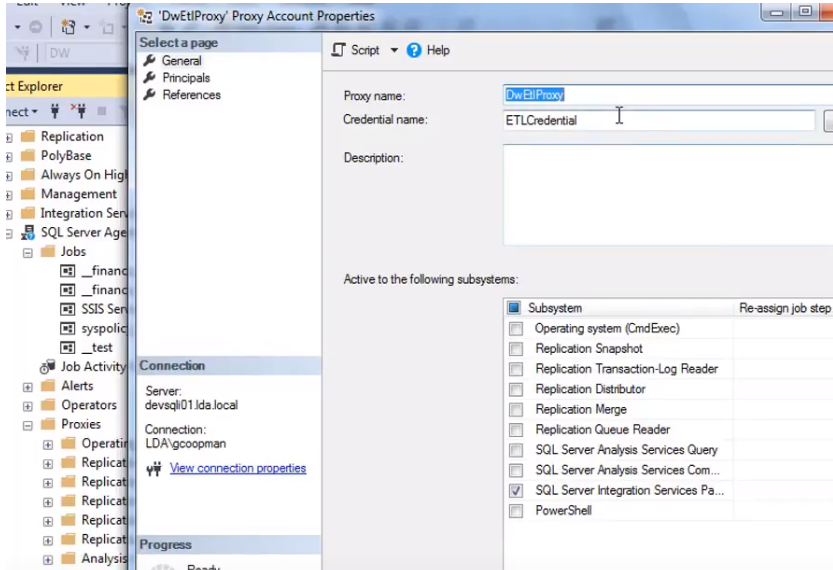
**SETUP Proxi:** To run the Packages you need to create Proxi under SQL Server Agent – Proxies – SSIS Package Execution



**Proxi:** DwEtlProxy

**Credential: ETLCredential**

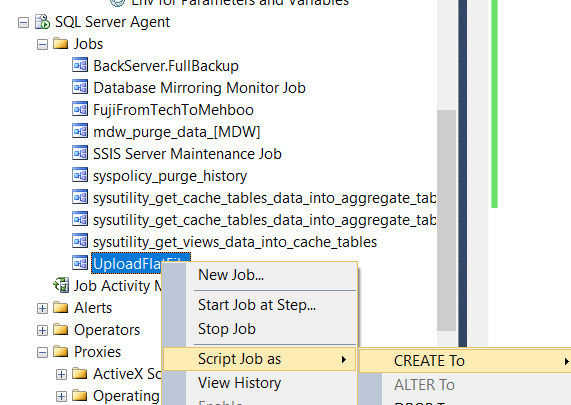
Proxi point to credential and Credential point to user that runs the Job



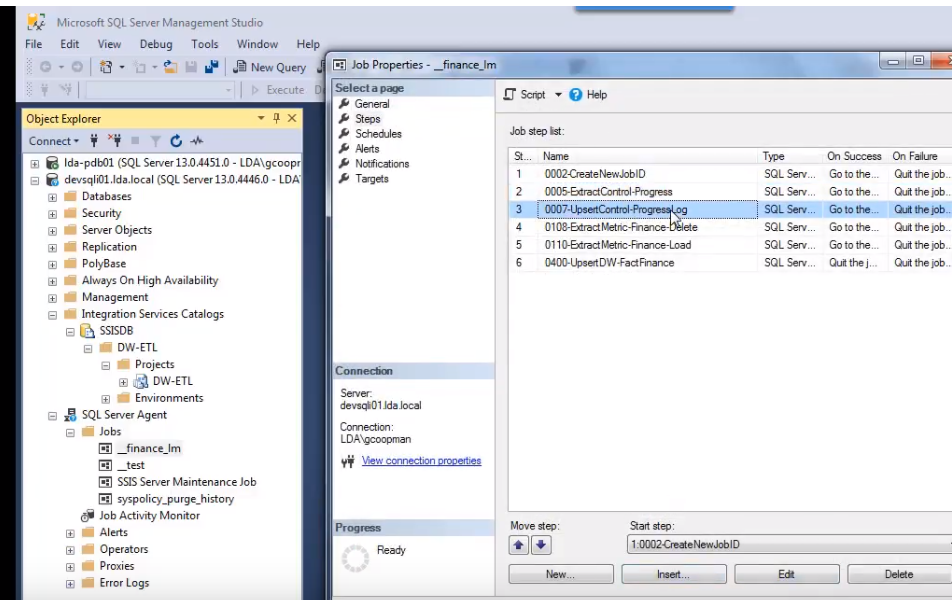
**Migrating SQL Server Agent Job:**

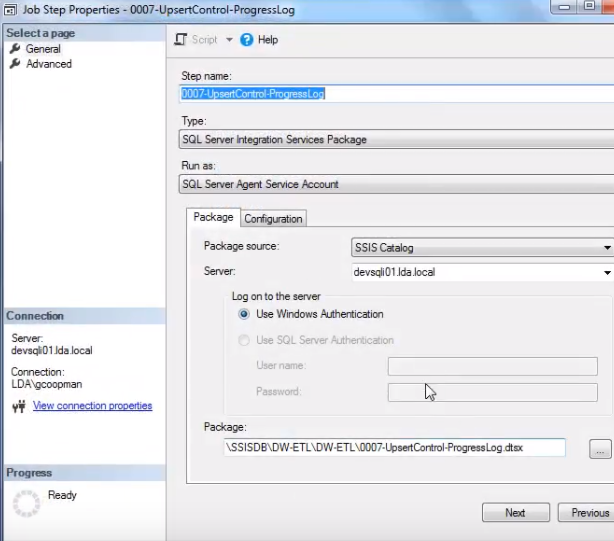
1. – Script “Create Script” from Source job:

Create Script and replace Source Server to Target Server, Job use MSDB database so just change server name and run it.



Once Job is setup check Steps and make sure that all Parameters mapped properly





Run the Project if everything is OK it will execute..

2. – Modify “Script” from Source Job

3. – Test and Debug Job Creation/ Execution