

OLTP	STAGE/ ODS	EDW	DW	CUBES	REPORTING
Daily Business Process Data	Intermediate Data [Direct Load Data]	Intermediate to DWH [Incremental Data (SCD)]	Actual Warehouse Storage [Incremental Data]	Small Storage Areas for Business Analysis [Incremental]	

Direct load [simply source to destination] [No logic load / Fresh load / Truncate Load] Oltp→stage Stage→ods	a) Import Export Wizard b) Data flow task c) Bulk Insert task
Incremental load [stage→EDW, EDW→DW] [logic load / Historical load]	Data flow task Execute SQL Task

Vinay Tech House

REAL-TIME LOADING PROCESS AND LOADING TYPES

Data loading happens between OLTP systems to BI Systems (Analytical systems) with different types of data loading.

There are 2 major types of data loading.

1. Direct load

- a) No logic loads
- b) Truncate and load (**Fresh data loading**)

Note: In Data Insights, between **OLTP → Stage and Stage → ODS**

2. Incremental Load

Incrementing destination with source data is called incremental loading. **[Stage → EDW, EDW → DW, DW → Cube]**

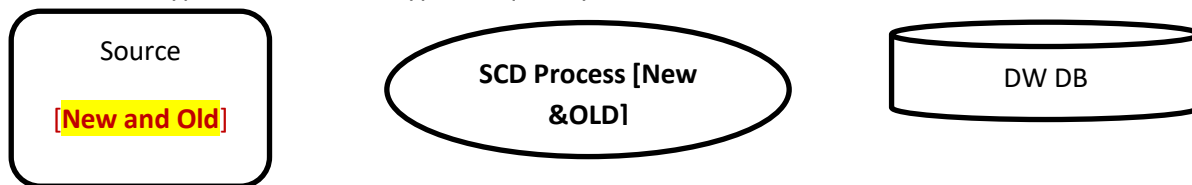
- b) **Delta load / Period direct incremental load**: Difference between two objects (source and destination)
Ex: Every day, the same data load into target without any changes.

Delta → Difference load [**compare source and target, load the differentiated records**]

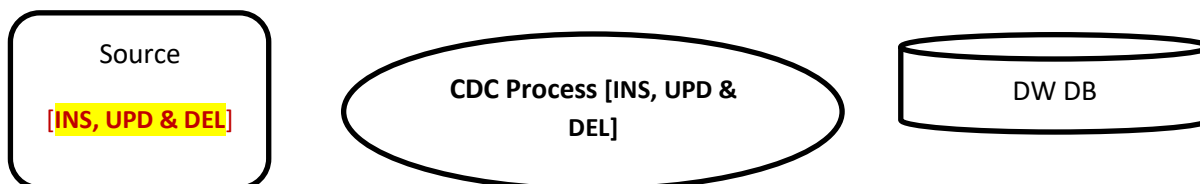
Period → specific date load [**current / previous day/ week/ month**]

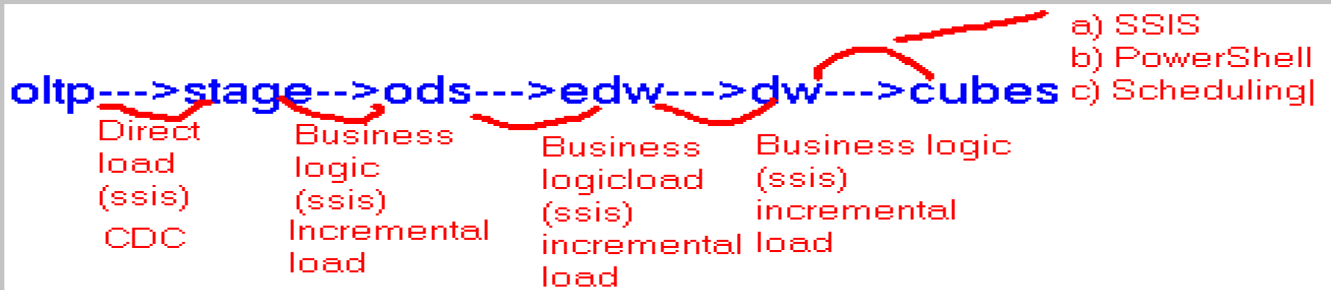


- c) **SCD (Slowly changing dimension load)**: If source has new and old records, we perform action accordingly. The action is of 6 types, but we use 3 types frequently.

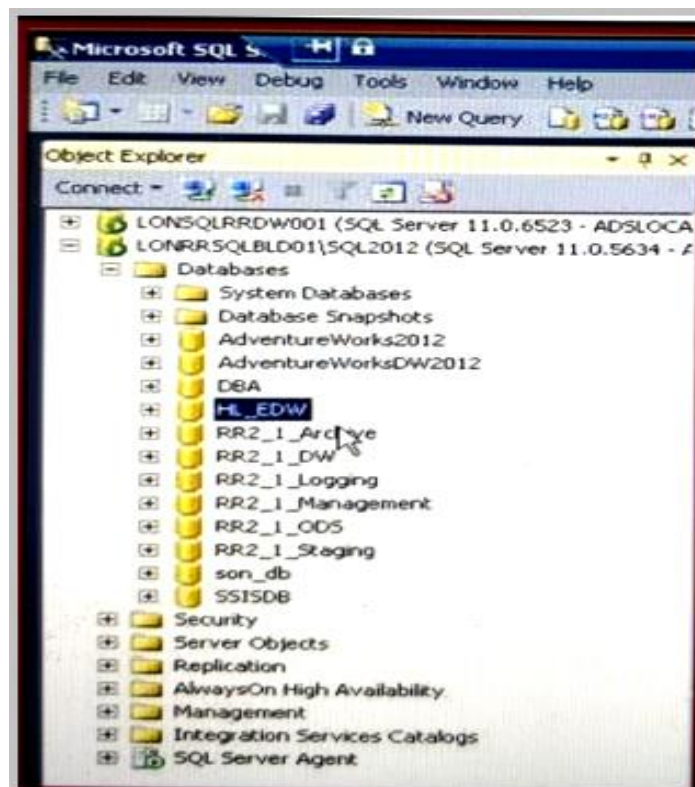
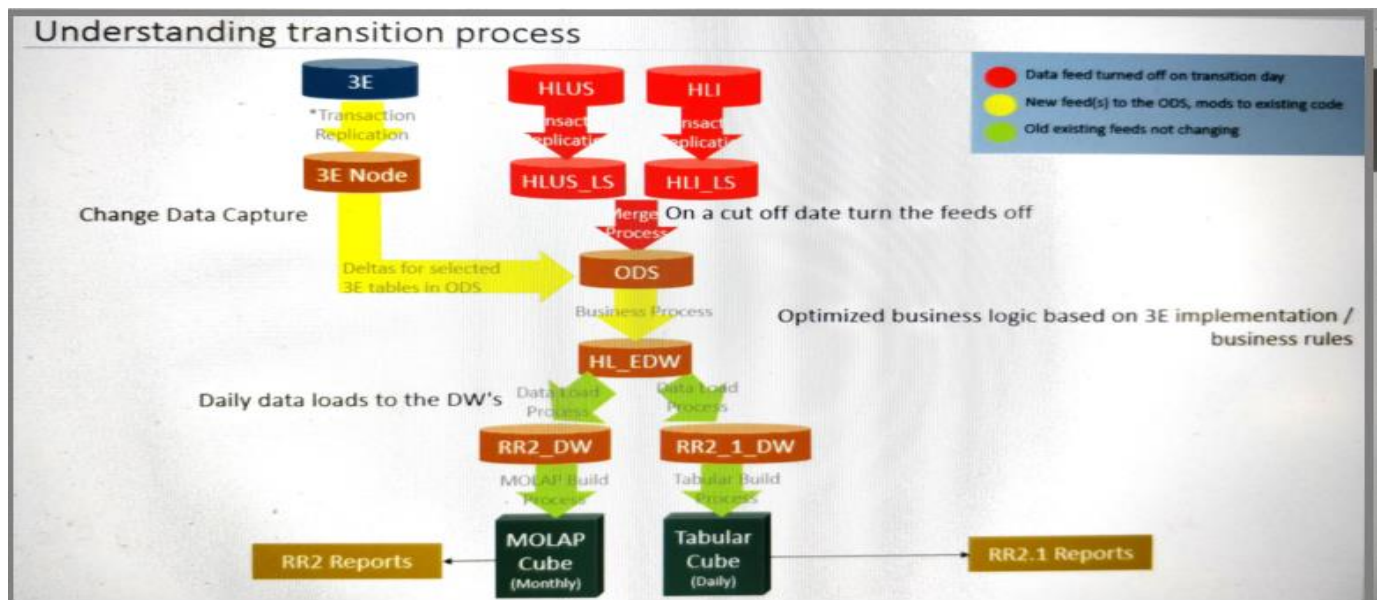


- d) **CDC (Change Data capturing)**: If we want to keep source and destination in sync this process helps. Means the inserts, updates and deletes whatever happened, we will bring those changes in to the target.



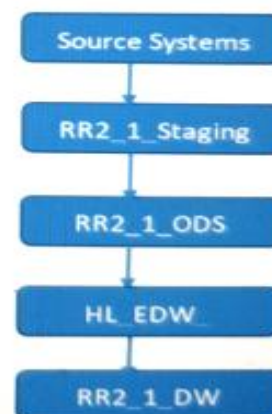


REAL TIME CUSTOMER END-END DATA FLOW [TOP MNC]



This image shows Build environment, which is Dev having the below databases.
For RR2 with appropriate naming conventions and operational names.
Ex: RR2_1_logging for log info storage.

Here the data movement is like below between databases.



REAL-TIME END-END FLOW OF SSIS PACKAGES

DEVELOPMENT	TESTING	PRE-PROD/UAT	PRODUCTION[live]
	Quality Analysts ETL Testers	User Acceptance Testing	24*7
Developers	Testers	Customers	Support Team
Non Shifts			Shifts [3 shifts]
No levels			Levels—1,2,3,4
No issues No severities			Issue Severity [Incident Severity]— P1,P2,P3,P4 Where P1 is high
Unit Testing [White box testing]	Quality Test [Black box testing]	Smoke test	Smoke test
Code test	Functional test	Change test	Change test
Major: Application creation and unit testing Add configurations/ parameters Logging	Application business functionality test Change configurations/ parameters and run the application		Making applications run smoothly Production settings in configurations / parameters and scheduled [SQL Server Agent]
Errors and trouble shooting	Defects raised in Defect Tracking Tools [HP Quality Center, Rational Rose, Bug Zilla etc...]		Issues are incidents raised in portals [Service now, Azure help desk]

Deployment: It is the process of moving packages and relevant files from one environment to other environment [Dev→Test, Test→Prod]

Configurations /parameters: these hold connection strings, userid, passwords, important user interaction values etc...

Logging: Capturing log information

Developer Operations:

- Create a package
- Add parameters (latest-project deployment) / configurations (old-package deployment) to the package [so that other environment people will work on it easily]
- Perform Unit testing
- Deploy the project [latest-project deployment] or package [old-package deployment]

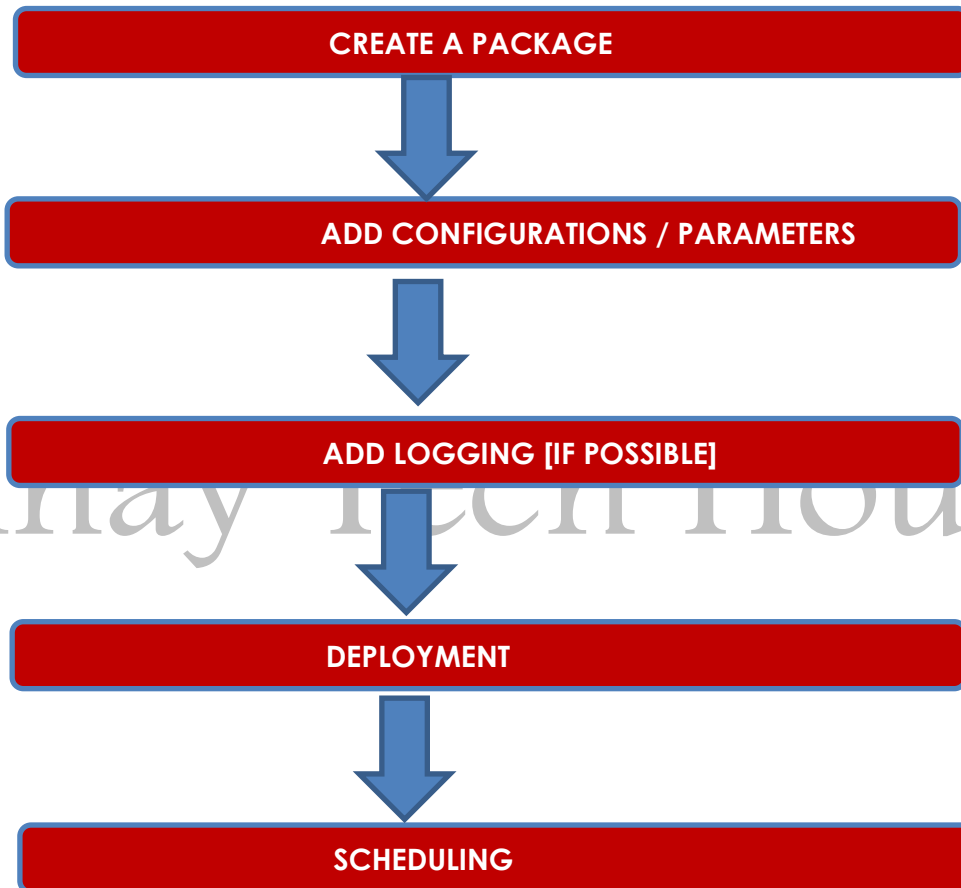
Support Operations:

- Schedule the packages and other relevant applications
- Monitor the scheduled jobs, taking action based on status.

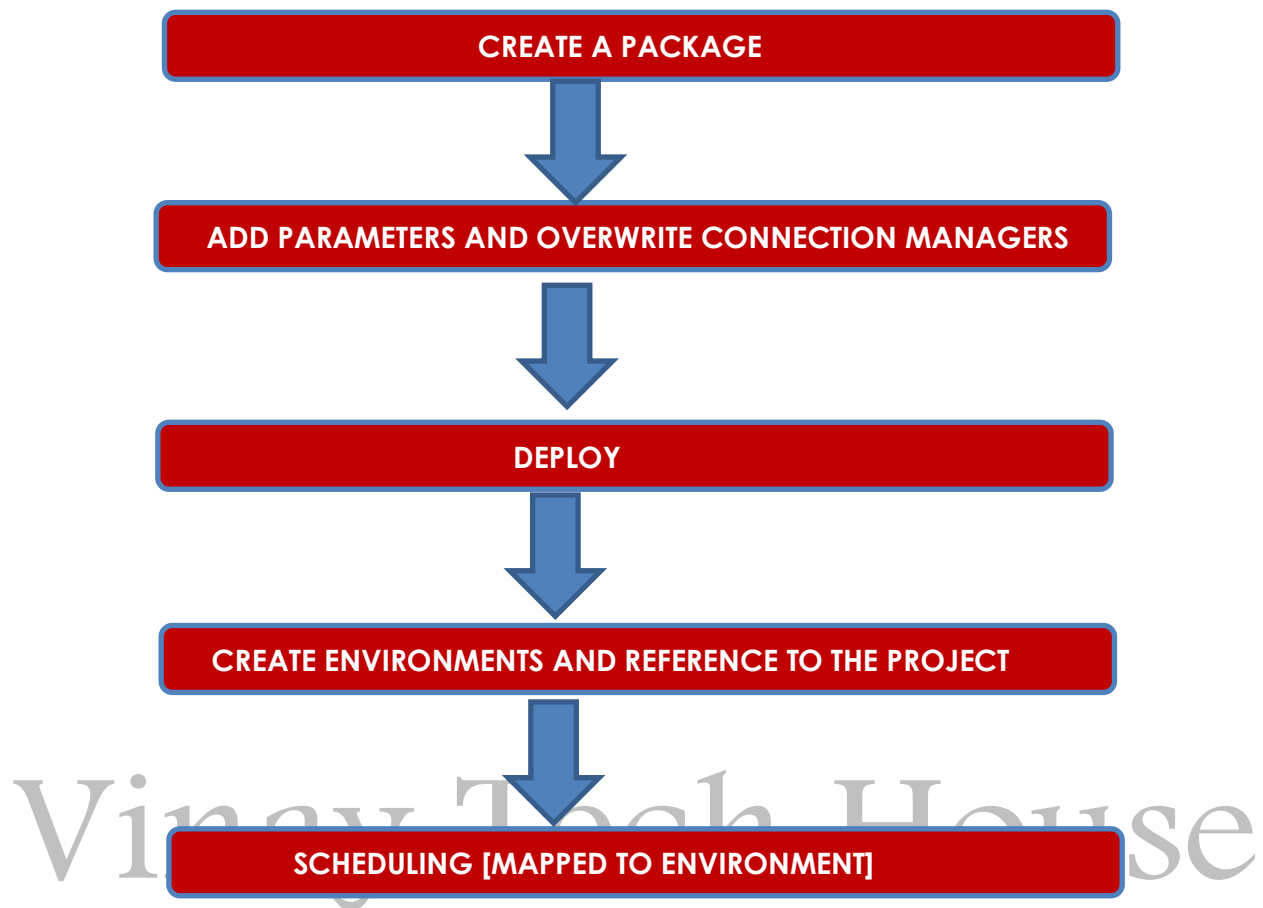
PROJECT DEPLOYMENT MODEL [2012 ONWARDS]	PACKAGE DEPLOYMENT MODEL [2005 ONWARDS]
PROJECT→RIGHT CLICK→DEPLOY	MANIFEST FILE CREATE AND DEPLOY
COMPLETE PROJECT AT ONCE	PACKAGE BY PACKAGE DEPLOYMENT
CATALOG DATABASE DEPLOY	FILE SYSTEM / SQL SERVER DEPLOYMENT
WONDERFUL LOGGING [CATALOG LOGGING] with multiple levels.	LEGACY LOGGING [SQL SERVER, FILE OR ANY]
VERSIONING	NO VERSIONING
PARAMETERS AND ENVIRONMENTS	CONFIGURATIONS

DEPLOYMENT BEST PRACTICE

- a) CREATE A PACKAGE
- b) ADD PARAMETERS TO THE PACKAGE
- c) DEPLOY TO CATALOG DATABASE
- d) CREATE ENVIRONMENTS
- e) SCHEDULE THE PACKAGES IN THE SQL SERVER AGENT



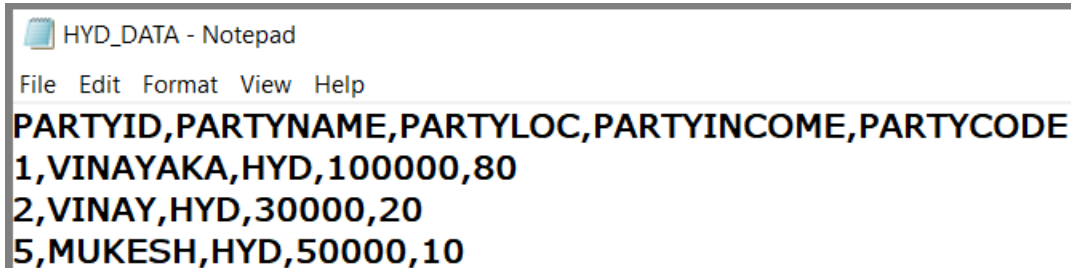
PROJECT DEPLOYMENT PRACTICE



STEP1: CREATE SET UP AND PACKAGE LIKE BELOW**a) Create two folders DEV and TEST under C Drive.**

Keep hyd_data.txt, mum_data.txt, che_data.txt files in both the folders.

Example of hyd_data.txt file



```
HYD_DATA - Notepad
File Edit Format View Help
PARTYID,PARTYNAME,PARTYLOC,PARTYINCOME,PARTYCODE
1,VINAYAKA,HYD,100000,80
2,VINAY,HYD,30000,20
5,MUKESH,HYD,50000,10
```

b) Create two databases VINAYTECH_DEV_DB, VINAYTECH_TEST_DB with a table like below

```
CREATE TABLE [dbo].[PARTY_SRC](
    [PARTYID] [varchar](50) NULL,
    [PARTYNAME] [varchar](50) NULL,
    [PARTYLOC] [varchar](50) NULL,
    [PARTYINCOME] [varchar](50) NULL,
    [PARTYCODE] [varchar](50) NULL
)
```

b) Create a package with a data flow task like below.

a) Create two connection managers like below

1. Flat file connection manager [which is pointing DEV folder and HYD_Data.txt file]

Connection manager → New flat file connection manager →

Connection manager name:

Description:

General
Columns
Advanced
Preview

Select a file and specify the file properties and the file format.

File name:

Locale: ☐ Unicode

Code page:

Format:

Text qualifier:

Header row delimiter:

Header rows to skip:

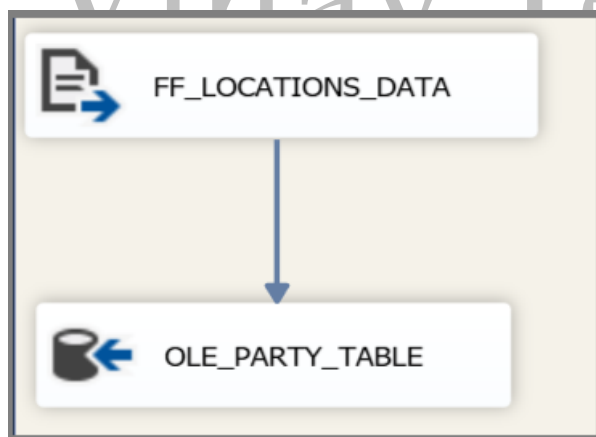
☒ Column names in the first data row

b). Connection manager → New oledb connection [name it: ole_tgt_conn]

create it by pointing to VINAYTECH_DEV_DB

c) Take Flatfile source and map to ff_src_conn

take Oledb destination, connect to flat file, and map to ole_tgt_conn, choose party_src table



d) Create parameters like below [**user values** → **parameters** → **connection strings** → **package run**]

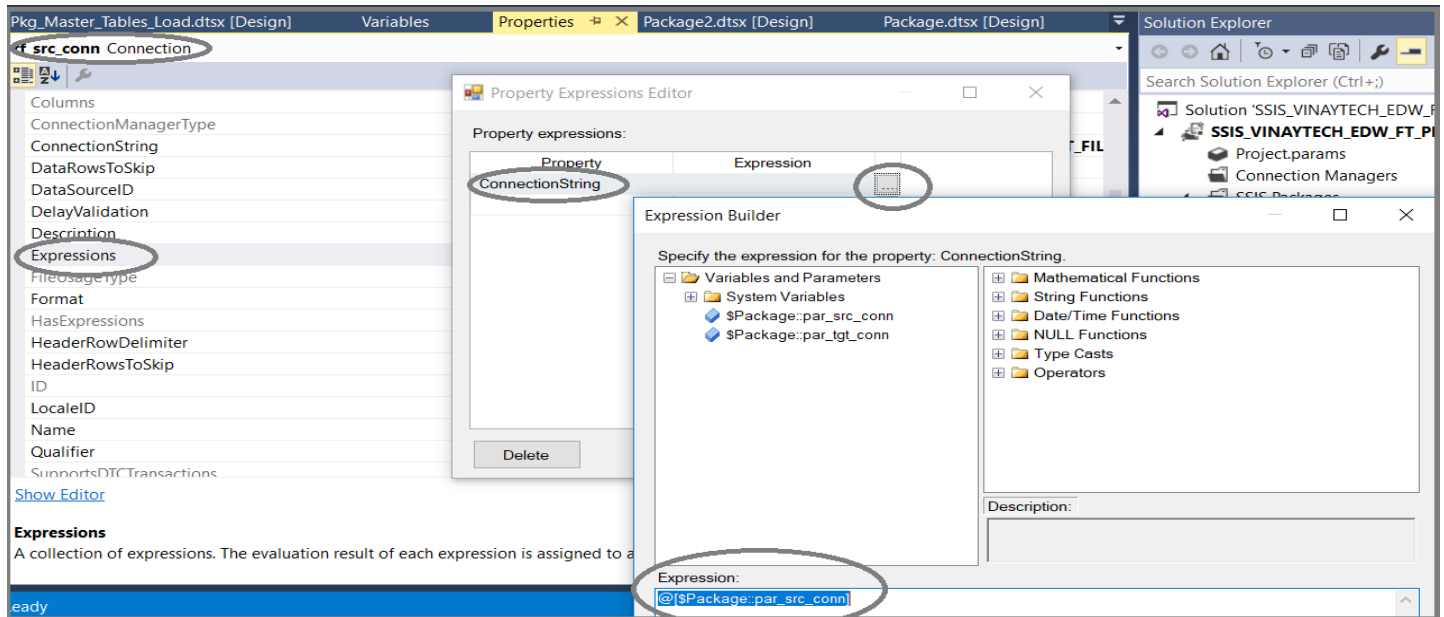
Control Flow Data Flow Parameters Event Handlers Package Explo...			
Name	Data type	Value	Sensitive
par_src_conn	String	C:\Users\Lenovo\Desktop\LAB_BACKUP & study material\INPUT_FILES\DEV\HYD_DATA.txt	False
par_tgt_conn	String	Data Source=DESKTOP-RN4SMHT\VINAYTECH_2017;Initial Catalog=VINAYTECH_DEV_DB;Provider=SQLNCLI11.1;Integrated Security=SSPI;Auto Translate=False;	False

In the above data flow two connection strings available, and I want to allow others to change only those. So, create two parameters.

Connection strings either copy manually, or go to the previous connection managers (ff_src_conn)→right click→properties→ connection string value→copy

Use the parameters for source and target connection string **[overriding connection strings]**

Source connection manager (ff_src_conn)→right click properties→expression→click ellipse→specify like below



Similarly map ole_tgt_conn connection manager connection string to par_tgt_conn
e) Build→Solution

f) Solution explorer→ project→ right click→deploy

Note: There should be a catalog set up to deploy in project deployment model.

Catalog deployment set up creation:

SSMS→Database Engine→Integration Services Catalog→right click

Create Catalog→

☐ Enable automatic execution of Integration Services stored procedure at SQL Server startup.

Name of the catalog database:
SSISDB

The catalog protects data using encryption. A key is needed for this encryption. Enter a password to protect the encryption key, and save the password in a secure location.

Password:

Retype Password:

You can manage the encryption key by creating a backup. If you migrate or move the Integration Services catalog to another SQL Server instance, you can restore the key to regain access to encrypted content.

☒ Enable this server as SSIS scale out master

Create Folder

SSISDB→Create Folder→ Specify Folder Name: VINAYTECH_MEDW_PROJECTS

Deployment from SSDT

Go to SSDT, View->Solution Explorer→MEDW_Project→Right Click→Deploy→Next→Specify Database Engine instance name and specify folder (VINAYTECH_MEDW_PROJECTS) to deploy→Next→Finish.

Select Destination

Introduction
Select Source
Select Destination
Validate
Review
Results

Enter the destination server name and where the project will be located in the Integration Services catalog.

Server name:

Authentication:

User name:

Password:

Path:

Browse for Folder or Project

Select a folder for the project or replace an existing project in an Integration Services catalog:

- SSISDB
 - VINAYTECH_MEDW_PROJECTS**

Integration Services catalog and the path that specifies the location of the project.

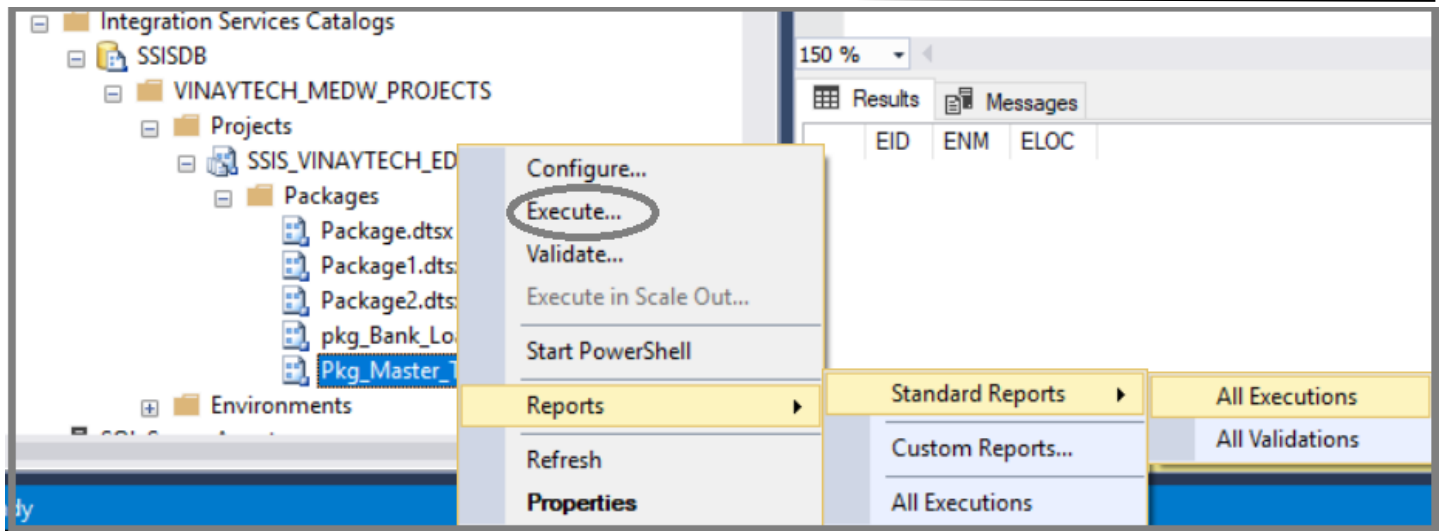
g) A sample run of package

1) Verify the table data in SSMS

```
USE VINAYTECH_DEV_DB;
```

```
SELECT * FROM PARTY_TABLE;
```

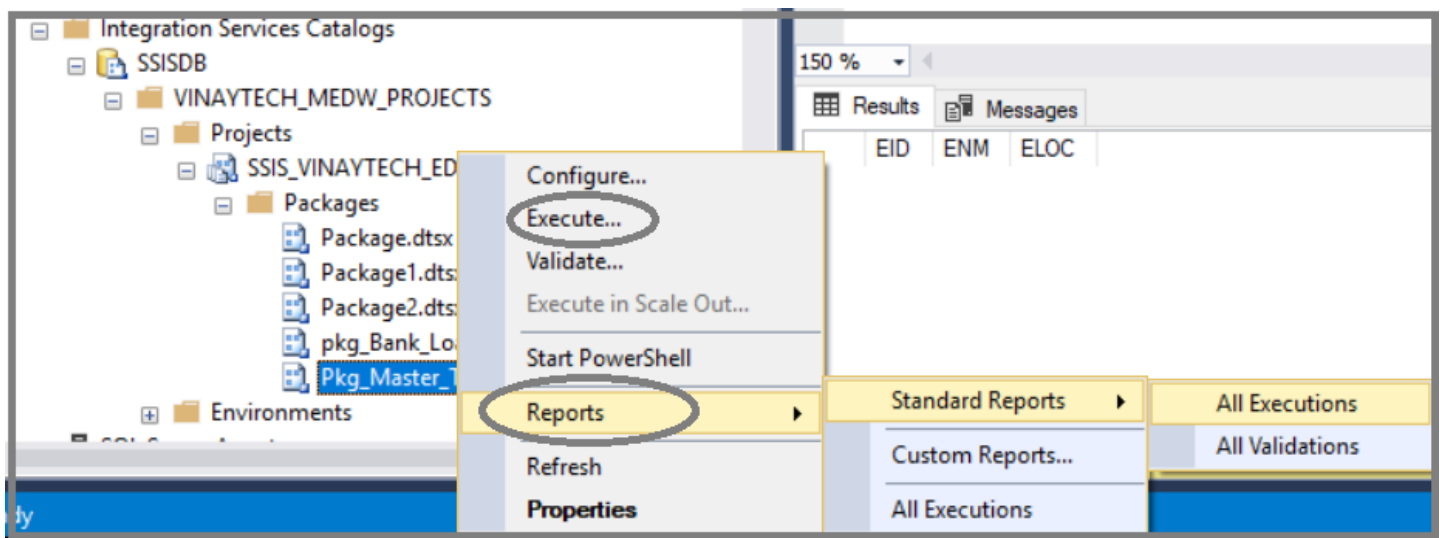
2) Run the package like below



Package→right click→execute

3)See the status of package executed

Package→right Click→reports→standard reports→All execution



on DESKTOP-RN4SMHT\VINAYTECH_2017 at 9/21/2019 12:43:40 PM

This report provides information about the Integration Services package executions that have been performed on the connected SQL Server instance.

Filter: Start time range: 9/15/2019 - 9/21/2019; Status: All; (4 more)

Execution Information

ID	Status	Report	Folder Name	Project Name	Package Name	Start Time
0	Failed					
0	Running					
1	Succeeded					
0	Others					
2	Succeeded	Overview All Messages Execution Performance	VINAYTECH_MEDW_	SSIS_VINAYTECH_EDW_	Pkg_Master_Tables_Load.dtsx	9/21/2019 12:38:36 PM

1 / 1

Click All messages to see the detailed log [Default log –Basic log]

Verify data again

```
USE VINAYTECH_DEV_DB;
```

```
SELECT * FROM PARTY_TABLE;
```

h) Tester or other environment people run

a) create a set up for tester [source and target setup]

Create a folder called TEST and keep the flat files [hyd_data.txt, Chennai_data.txt etc...]

Create a database VINAYTECH_TEST_DB, and create a table like below [PARTY_SRC TABLE]

```
CREATE TABLE PARTY_TABLE ([EID] varchar(50), [ENM] varchar(50), [ELOC] varchar(50))
```

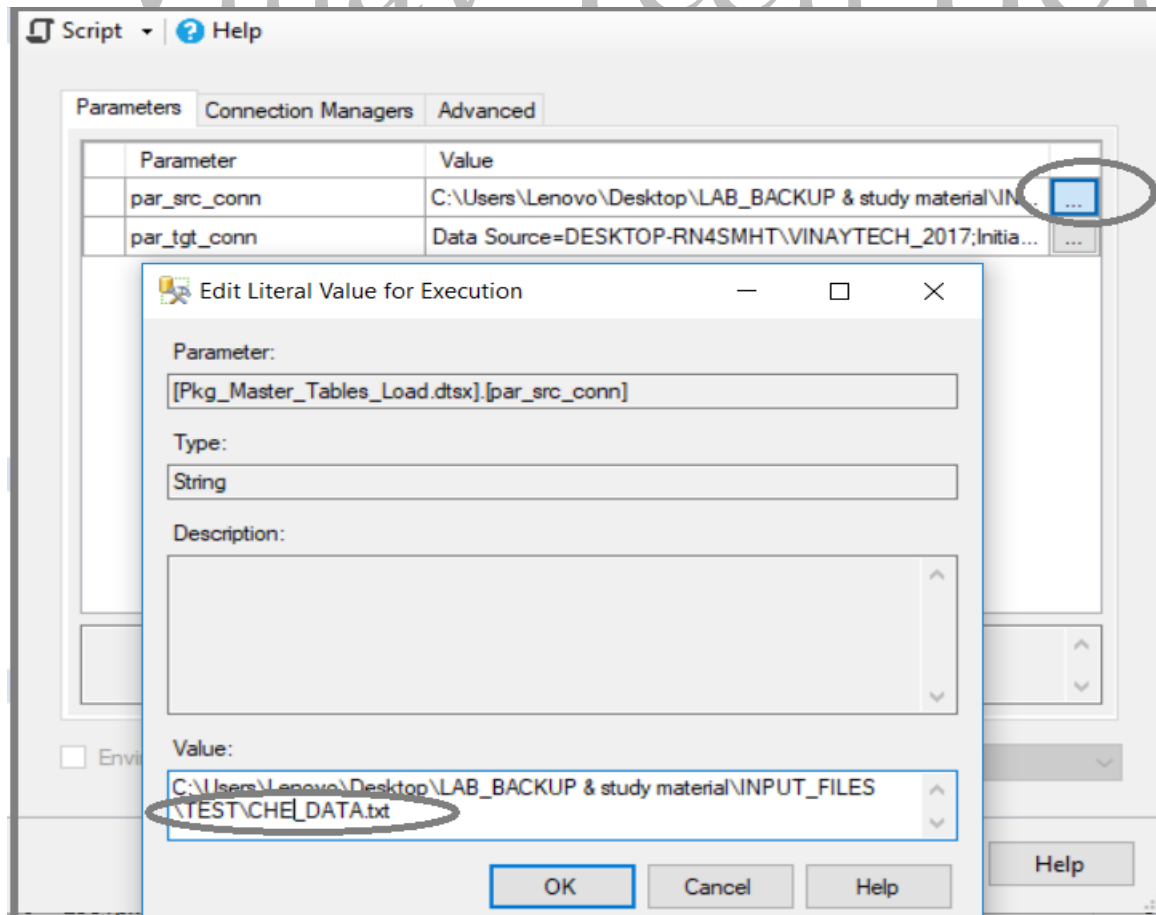
```
USE VINAYTECH_TEST_DB;
```

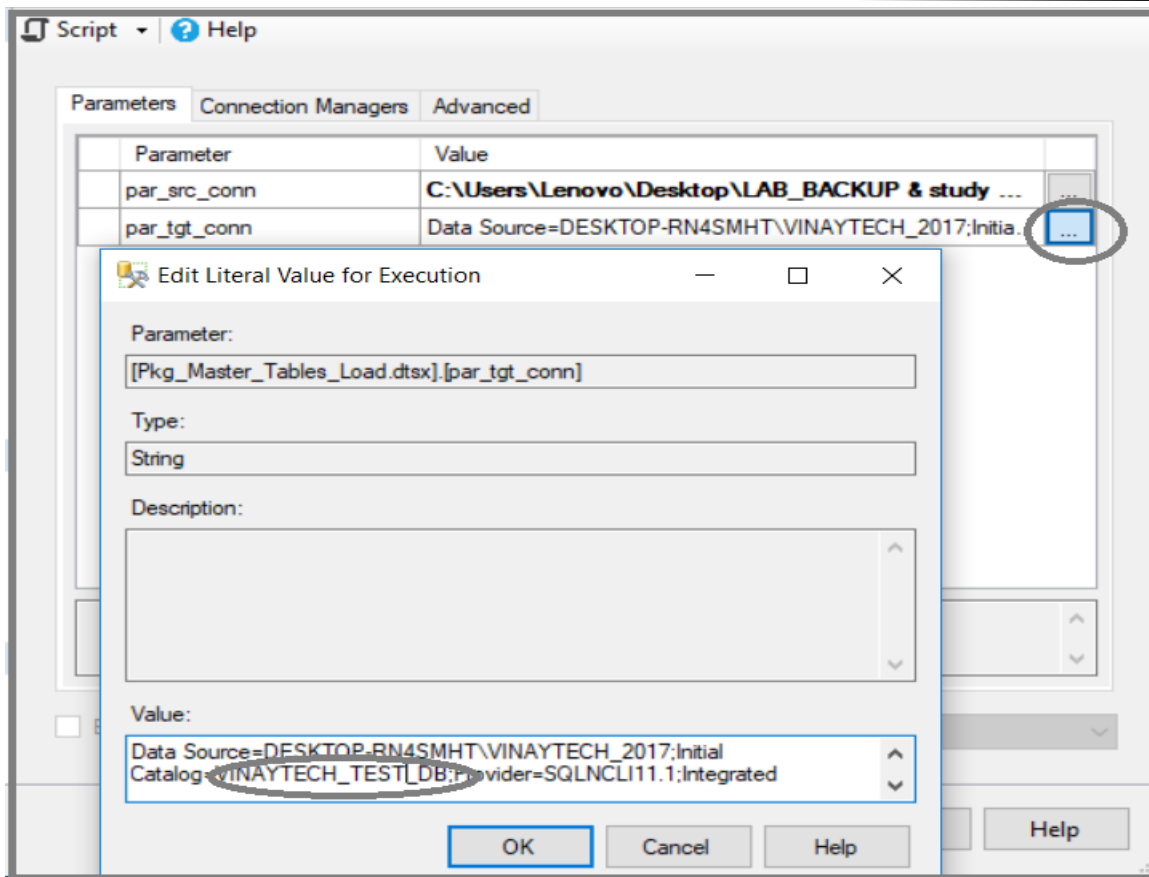
```
SELECT * FROM PARTY_TABLE;
```

b) Go to package, right click → execute → Change par_src_conn to TEST folder ,

par_tgt_conn database point to

VINAYTECH_TEST_DB, click ok





Now the package executed with TEST environment settings and load data into PARTY_SRC under VINAYTECH_TEST_DB

```
USE VINAYTECH_TEST_DB;
SELECT * FROM PARTY_SRC ;
```

Now it will show you the data

i) Scheduling the packages with production settings [production running]

a) create a set up for production [source and target setup]

Create a folder called PROD and keep the flat files [hyd_data.txt, Chennai_data.txt etc...]

Create a database VINAYTECH_PROD_DB, and create a table like below [PARTY_SRC]

```
CREATE TABLE PARTY_TABLE ([EID] varchar(50), [ENM] varchar(50), [ELOC] varchar(50))
```

```
USE VINAYTECH_PROD_DB;
SELECT * FROM PARTY_TABLE;
```

SSMS→SQL SERVER AGENT→START

Jobs→ right click→ New job→

General: provide job name [Ex: Daily_load]

Steps: create two steps like below

Step1: New step

New Job Step

Select a page

- General
- Advanced

Script ? Help

Step name: Truncate_Data

Type: Transact-SQL script (T-SQL)

Run as:

Database: VINAYTECH_PROD_DB

Command: TUNCATE TABLE PARTY_TABLE;

Connection

Server:

New Job Step

Select a page

- General
- Advanced

Script ? Help

Step name: Party_Table-Load

Type: SQL Server Integration Services Package

Run as: vinaytech_jun_proxy

Package Configuration

Server: DESKTOP-RN4SMHT\VINAYTECH_2017

Log on to the server

- ☒ Use Windows Authentication
- ☐ Use SQL Server Authentication

User name:

Password:

Package: \SSISDB\VINAYTECH_MEDW_PROJECTS\SSIS_VINAYTECH_EDW.

Connection

Server: DESKTOP-RN4SMHT\VINAYTEC

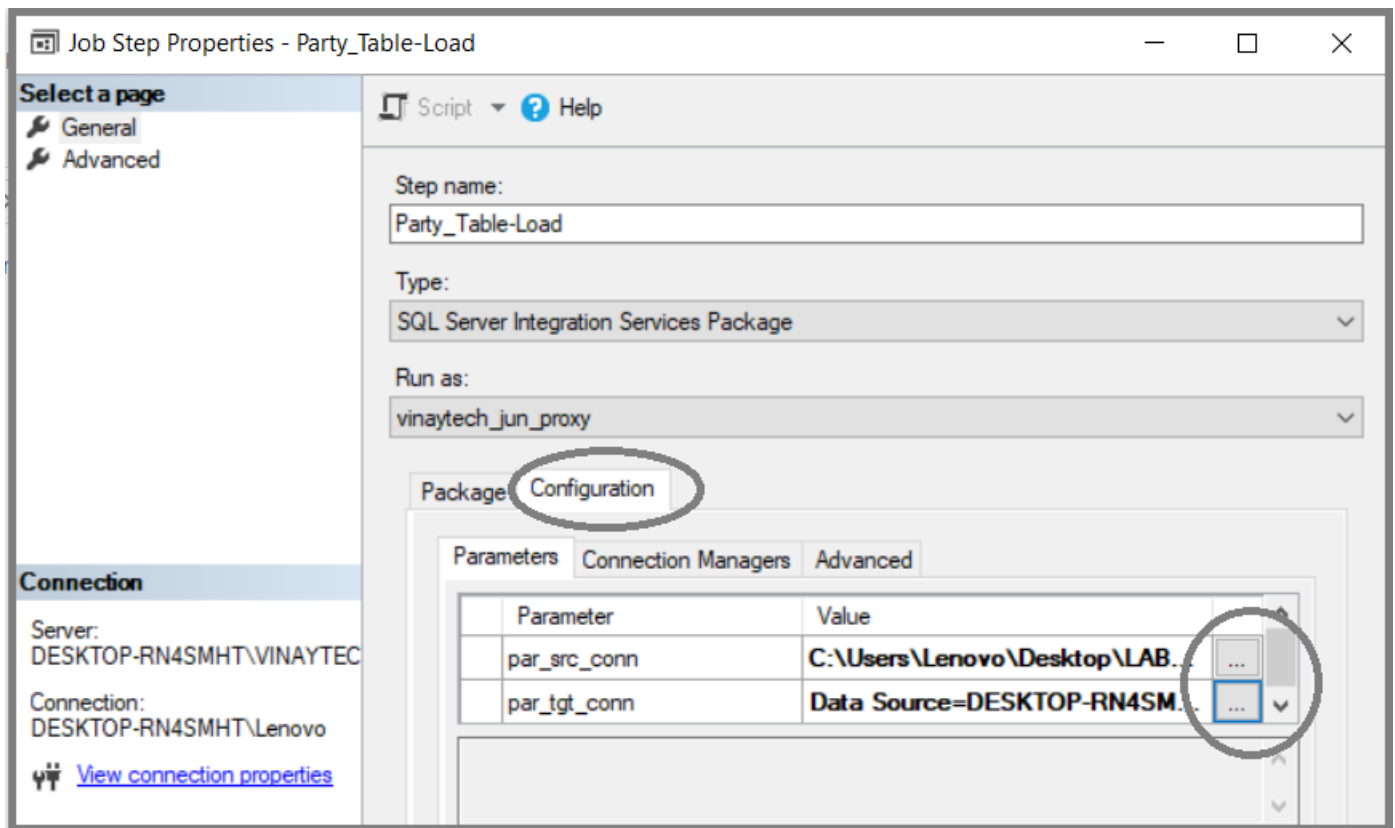
Connection: DESKTOP-RN4SMHT\Lenovo

[View connection properties](#)

Progress

Ready

Go to configurations and specify the settings for the production



Go to Step1: Advanced → On Success → Step2

Schedule

New Job Schedule

Name: Jobs in Schedule

Schedule type: ☒ Enabled

One-time occurrence

Date: Time:

Frequency

Occurs:

Recurs every: day(s)

Daily frequency

☒ Occurs once at:

☐ Occurs every: hour(s)

Starting at:

Ending at:

Duration

Start date: ☐ End date: ☒ No end date:

After scheduled time to see job statistics:

Job(Daily_Load)→right click→View History→see the statistics

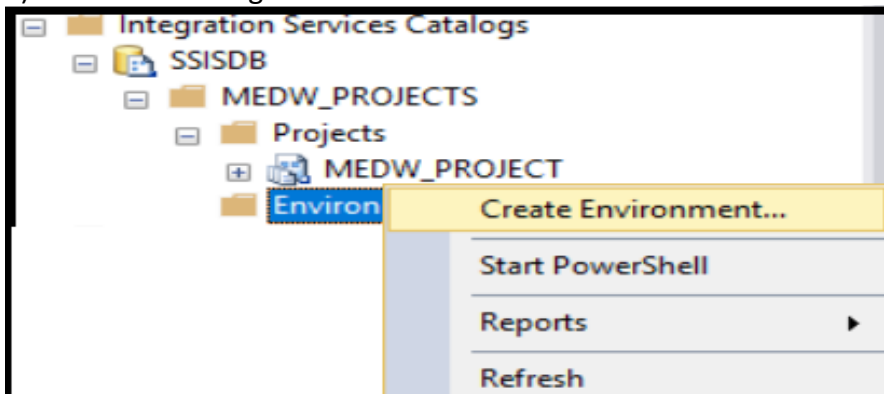
If it is success, then go to the table and see the data.

```
USE VINAYTECH_PROD_DB;  
SELECT * FROM PARTY_TABLE;
```

CREATE ENVIRONMENT WITH DEVELOPMENT SETTINGS:

User choose environment → parameter → package → run with the environment values

a) Environment → Right Click → Create Environment like below

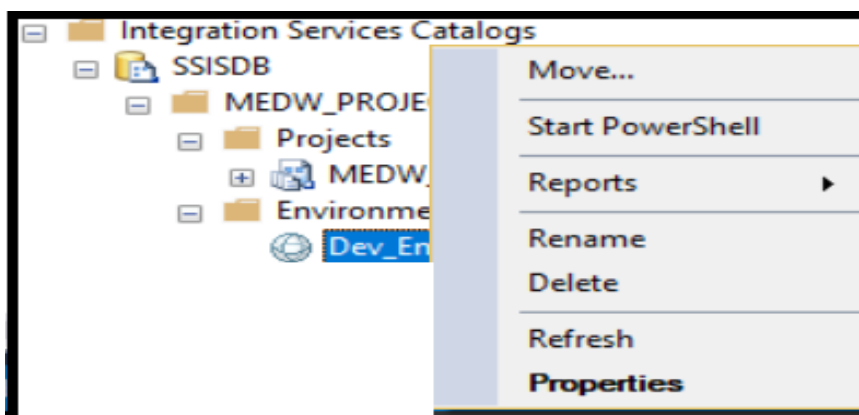


Environment Name: **Dev_Env**

Description: Development settings mentioned

OK

b) Dev_Env → Right Click → Properties



House

Variables section → add variables with source and target connection string values

Variables:			
Name	Type	Descri...	Value
env_src_conn	String		C:\Users\Lenovo\Desktop\LAB_BACKUP & study material\INPUT_FILES\DEV\HYD_DATA.txt
env_tgt_conn	String		Data Source=DESKTOP-RN4SMHT\VINAYTECH_2017;Initial Catalog=VINAYTECH_DEV_DB; provider=...

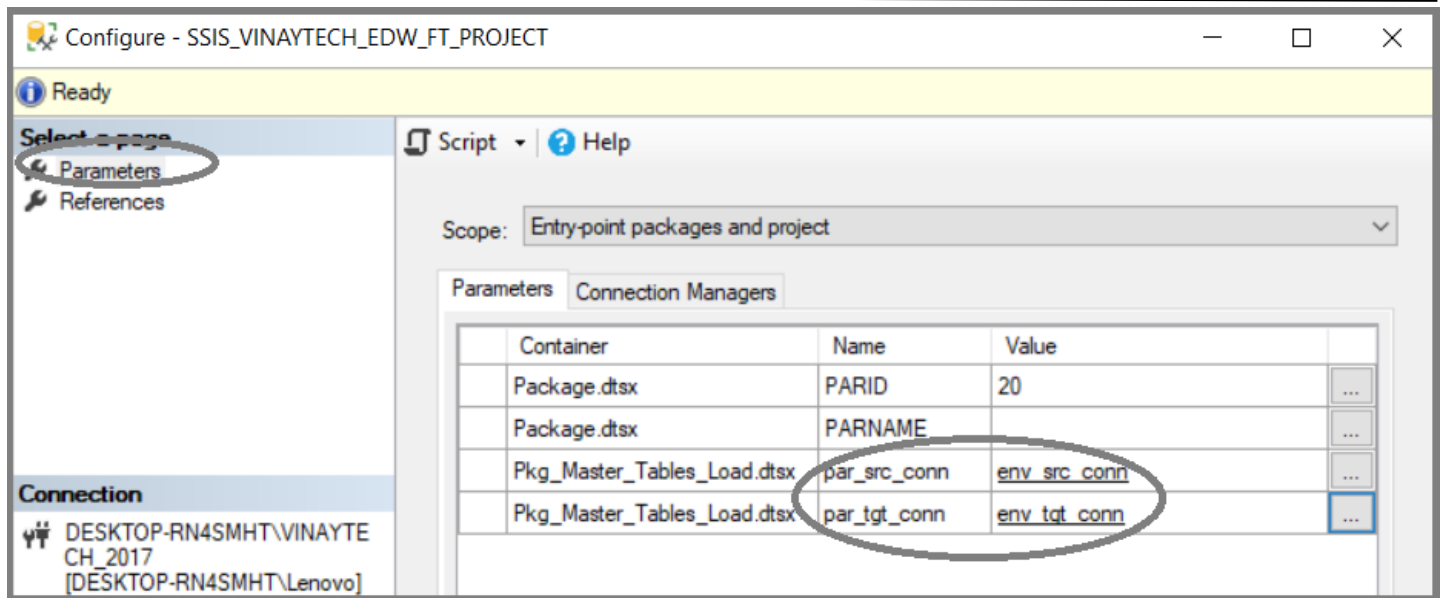
OK

Assigning environment to the package:

Go to SSIS Db → Foldername → Project → right click → Configure → Click Add →

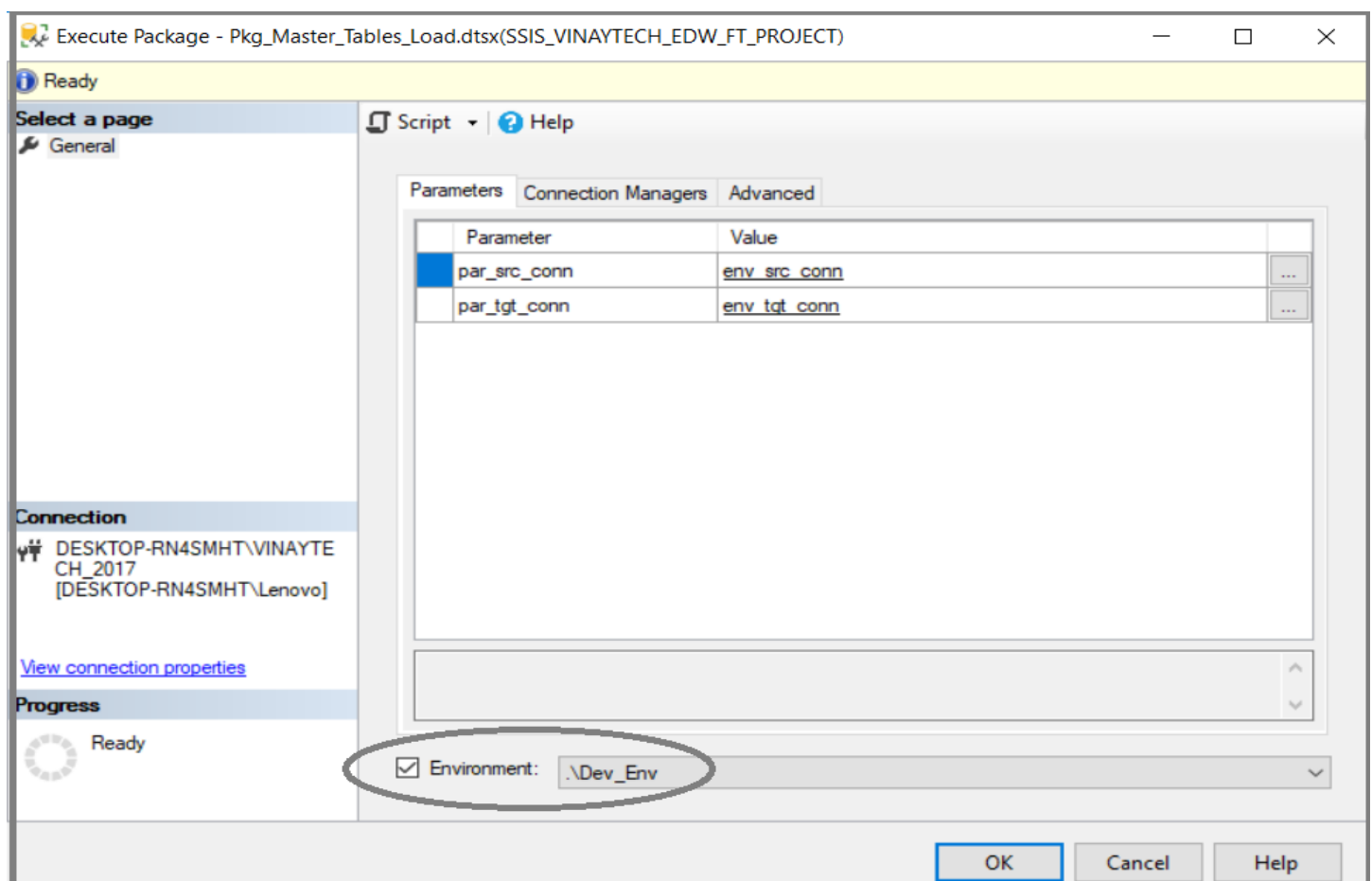
Specify Dev_Env

Go to parameters and map environment variables to parameters like below



Execute package with environment

Go to package → right click → execute → Specify the environment like below



After successful execution of the package, go to the below and see the data

USE VINAYTECH_DEV_DB;

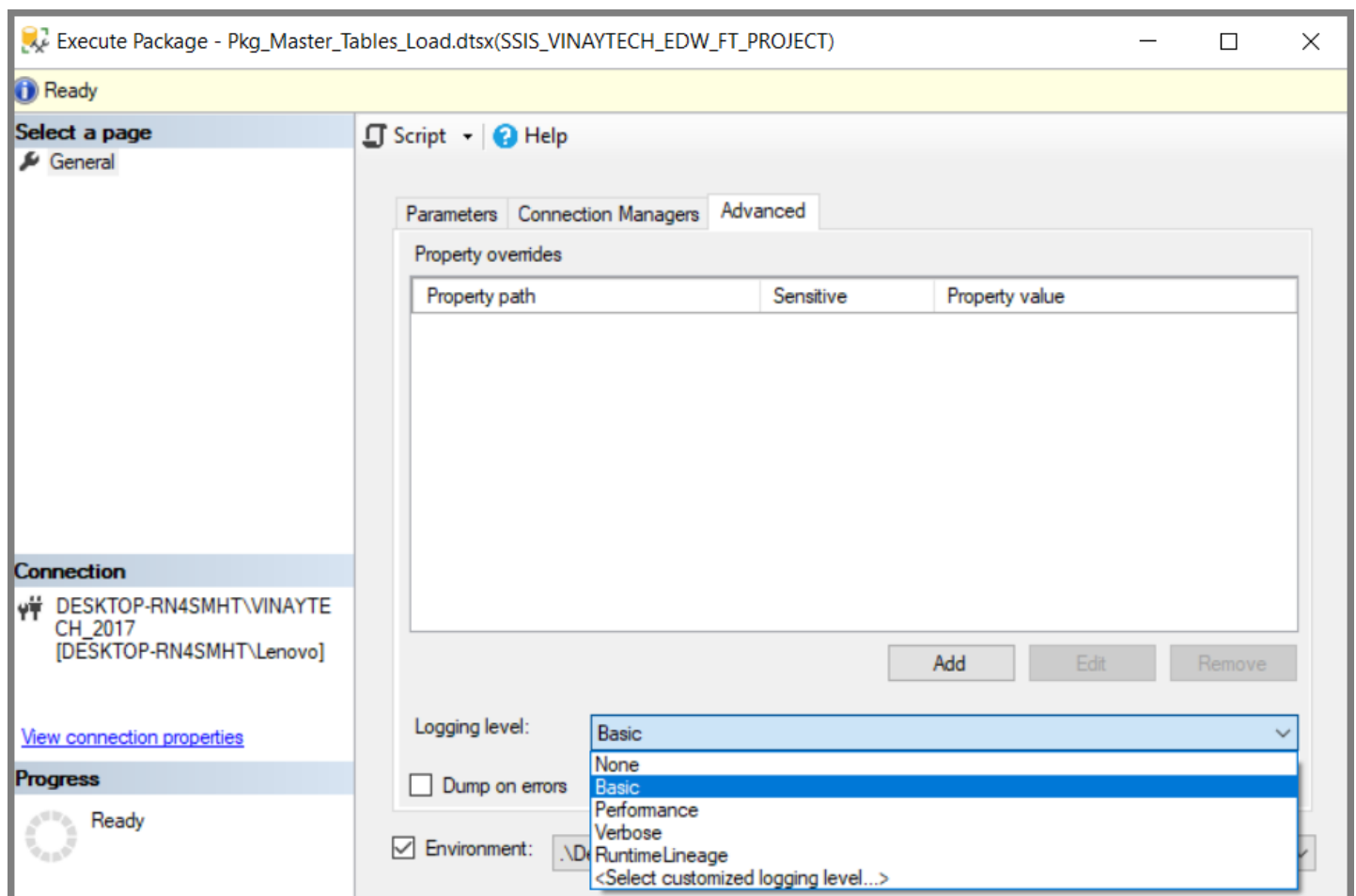
SELECT * FROM PARTY_TABLE

PRACTICE THE ADDITIONAL PROJECT DEPLOYMENT OPTIONS:**Versions observation**

- 1> SSDT→PROJECT→ Add one new package
- 2> SSDT→Project→Deploy→ Specify catalog Server and Database,
Now the project deployed with version 2
- 3> SSMS→ Integration Services catalog→ SSIS Db→ Folder→ Project→right click→ versions
- 4> Choose any version which you are interested and click restore.

Catalog log observation

- a) SSMS→ Integration Services catalog→ SSIS Db→ Folder→ Project→package→right click→
Execute→ Choose advanced table logging level→Basic→ok



- b)Run again the same package with logging level→verbose
- c) Package→ reports→ standard reports→all executions→
Click all messages and observe the log info and number of pages.

Ex: Basic –Will not show you milliseconds information and the complete log in less pages

Verbose—Will show you milliseconds and the complete log in more pages

Real time:

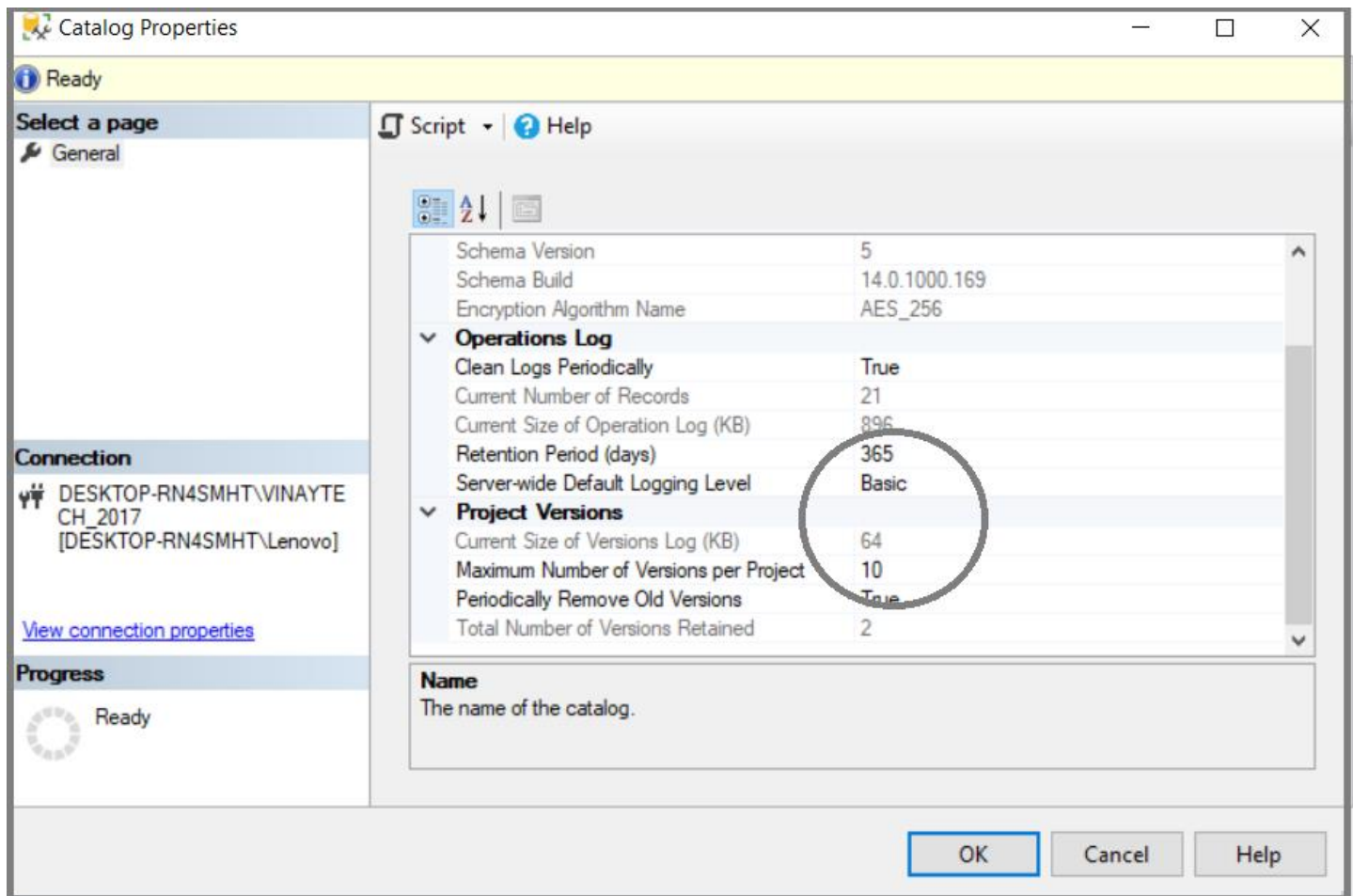
Development and detailed statistics observation level, we take verbose.

Production level, we always use basic logging. Otherwise more log generated and kills the performance.

Controlling the number of versions and log maintain days

Integration Services Catalog→SSIS DB→ right click→properties

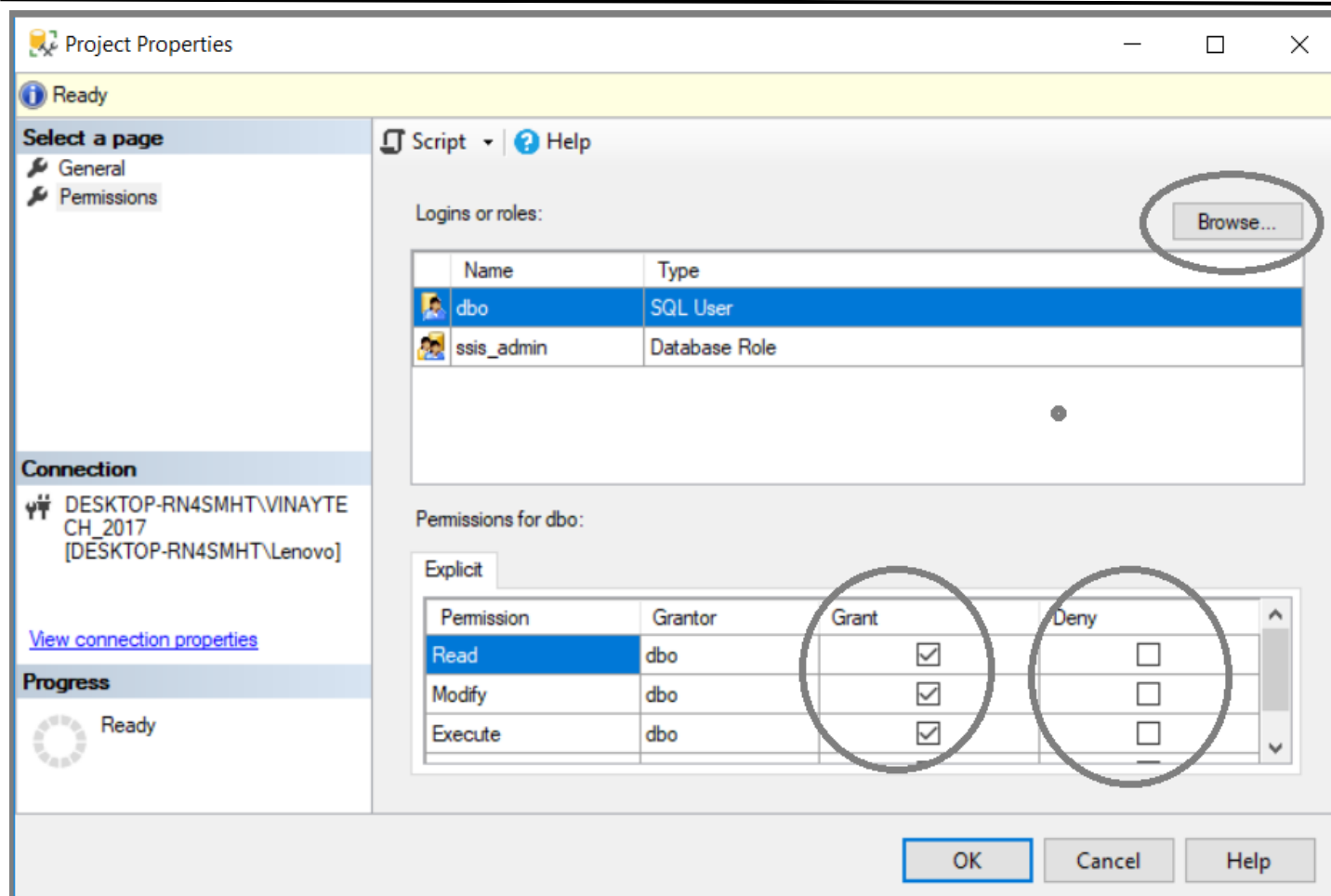
Change the versions values, days log maintained or default logging level based on the project.



SECURITY TO THE PROJECT [ROLE BASED]

SSMS→ Integration Services catalog→ SSIS Db→ Folder→ Project→right click→properties→

Click browse and choose a role / user / group, grant or revoke the permissions [read, write, execute etc...]



Project and package logging, configurations, and deployment comparisons

Package Deployment Model	Project Deployment Model
Logging in multiple providers <div> SSIS log provider for SQL Server SSIS log provider for SQL Server SSIS log provider for Windows Event Log SSIS log provider for Text files SSIS log provider for SQL Server Profiler SSIS log provider for XML files </div>	Logging in Catalog database level [SSISDB] in SQL Server only.
Configurations in multiple areas <div> XML configuration file XML configuration file Environment variable Registry entry Parent package variable SQL Server </div>	Parameters and environments in Catalog database SSISDB
Deployment methods using manifest file a) File System (any folder in a system) b) SQL Server (SSMS→Integration Services)	Deployment to Catalog database (SSISDB)
SSMS→SSIS Security at package level (role based security) File System Package password and protection level is the security	SSMS→Database Engine Security at project level (role based security) File System Project password and protection level is the security

PACKAGE DEPLOYMENT MODEL

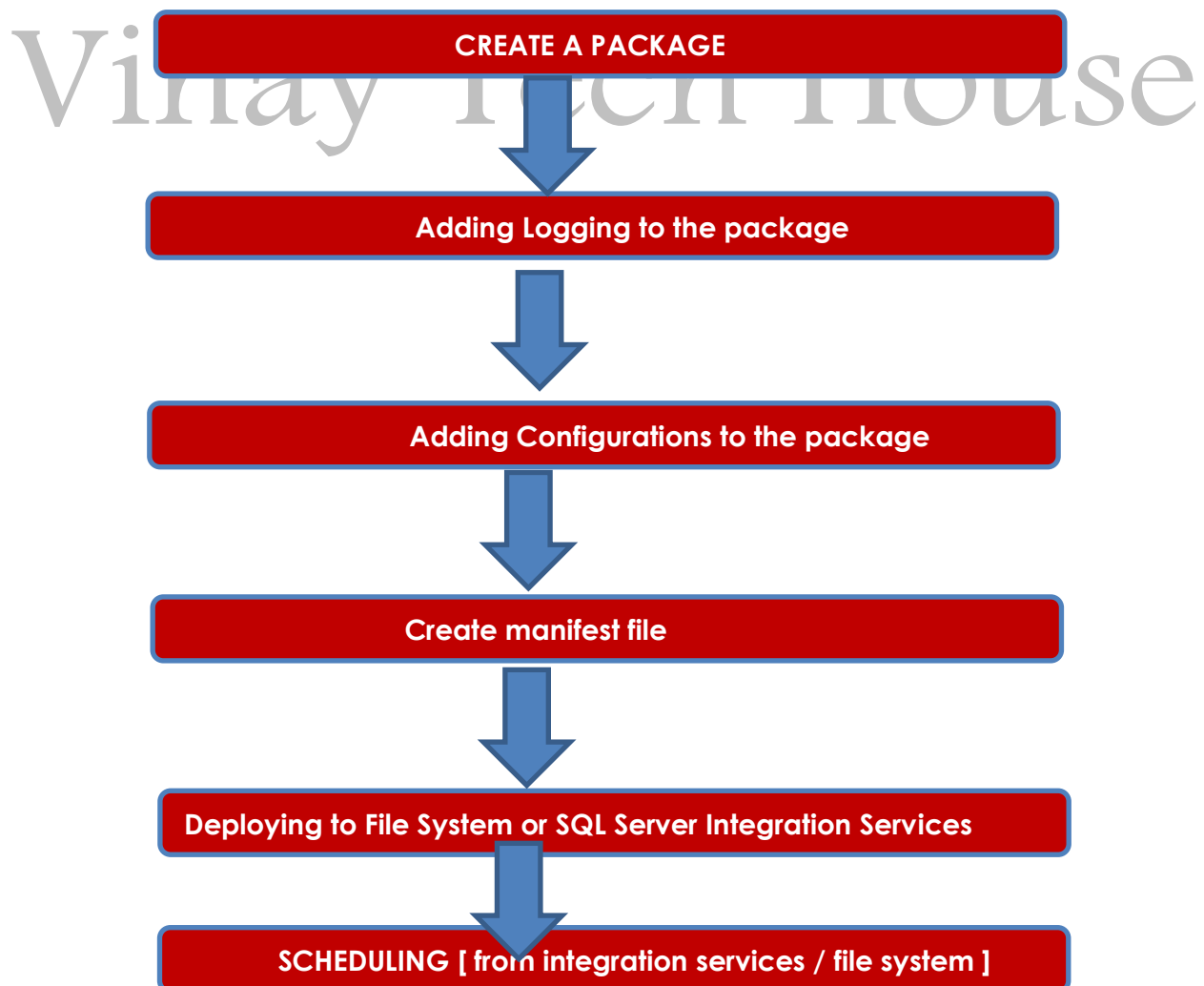
- a) Create package
- b) Add logging [not required for project deployment because we have default logging]
- c) Add configurations [similar to parameters in project deployment]
- d) Create manifest file
- e) Deploy to file system / SQL server database [under integration services]
- f) Schedule package [mostly from SQL Server database] [under integration services]
- g) Security to the packages [optional]

Deploying package to FILE SYSTEM (folder) or Integration Services in SSMS

It requires a manifest file to deploy the packages.

Usually before deploying we will add the below additional information

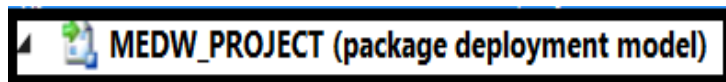
- a) Logging b) Configurations store [XML / SQL SERVER/ ANY OTHER [5 AREAS]]



a) Converting Project from Project Deployment Model to Package Deployment

Open SSDT, View→Solution Explorer

Goto Project MEDW_PROJECT→ right click->Convert to Package Deployment Model

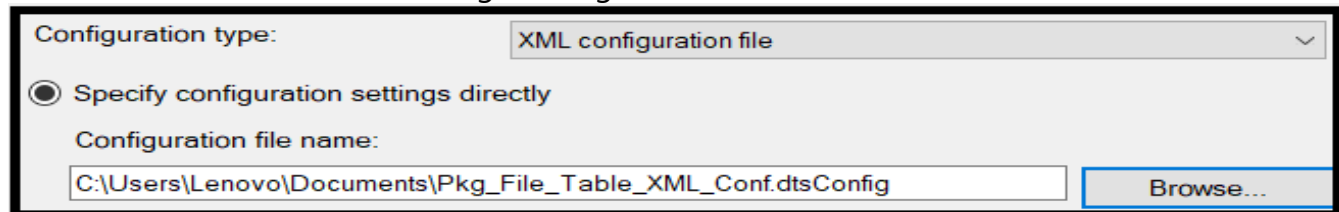


Note: It can't convert packages which use concepts beyond 2012 version.

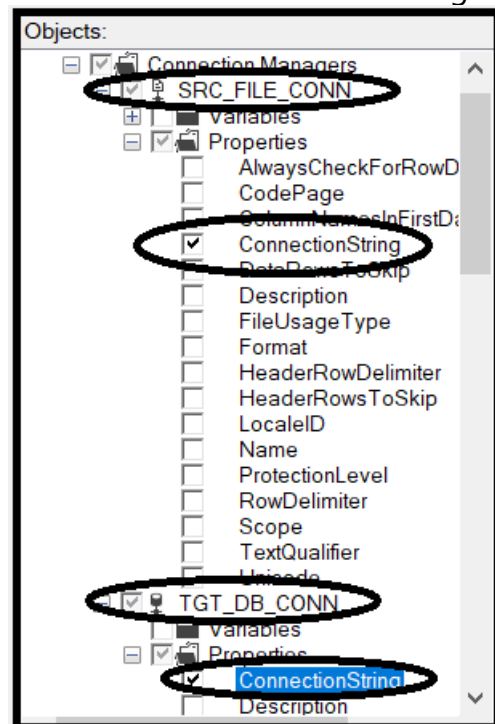
b) Adding configurations to the package [XML configuration here]

Go to package → Control Flow→Properties→Configurations→

Next→Tick mark Enable Package Configurations→ Click Add→



Next→ choose ConnectionString from SRC_FILE_CONN and TGT_DB_CONN



Next→Specify Configuration name (Pkg_XML_Config)→ Preview settings→Close.

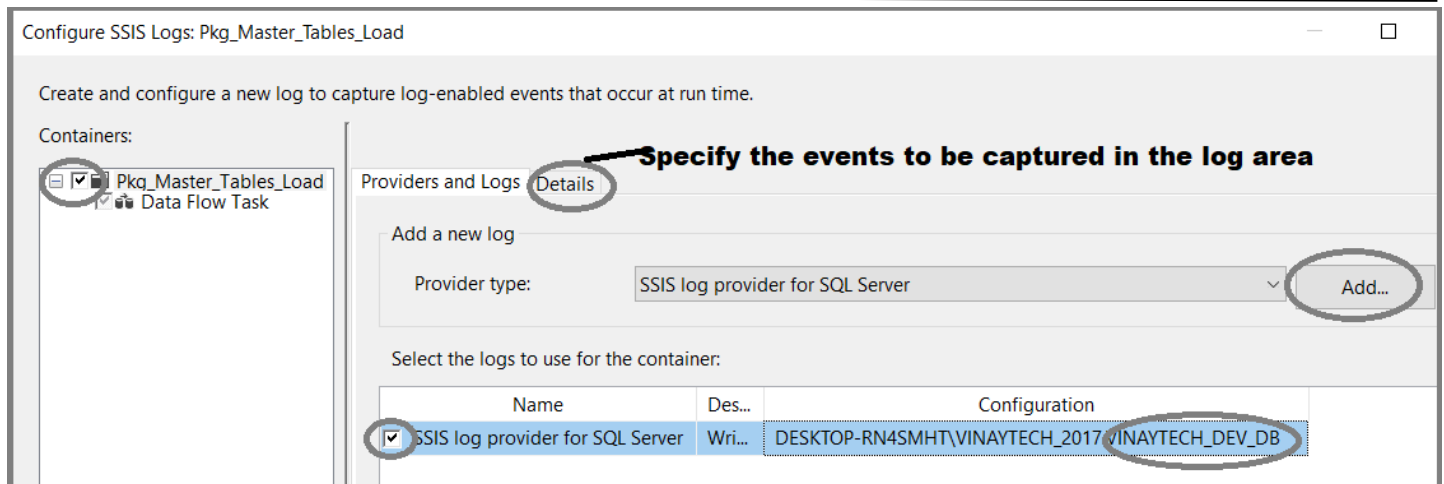
Build->Solution

c) Adding logging

To see the log after execution of the package

SSIS MENU→ Logging→

Tech House



To verify log creation happening, run the package and then go to Vinaytech_Dev_DB
And run the below

Select * from sysssislog

```
SELECT * FROM SYSSSISLOG ORDER BY STARTTIME DESC
```

```
select * from sysssislog where message='' and event='OnPostExecute'
```

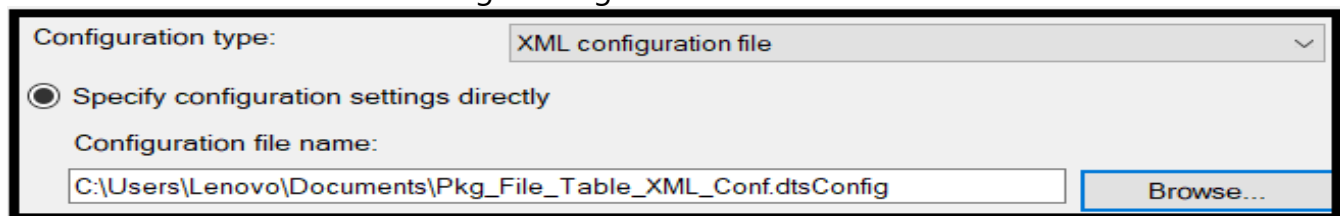
```
--packages successful
```

```
Note: message <>' --package failed
```

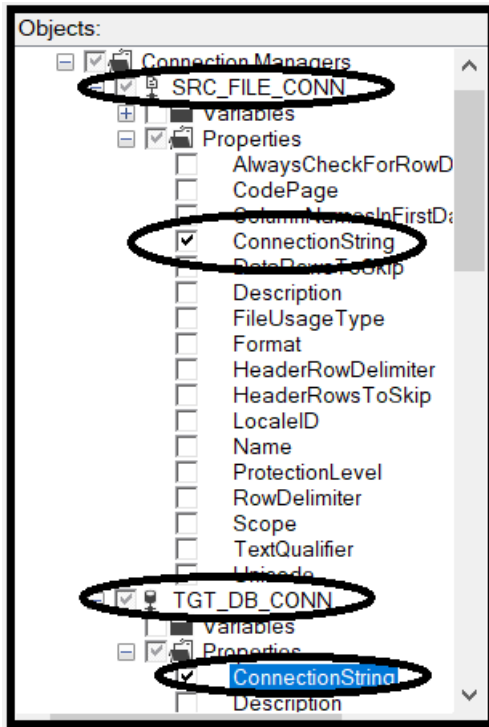
```
select * from sysssislog where source='pkg_multi_transform'
and event='OnPostExecute'
```

d) Adding XML Configuration

a) Go to package → Control Flow → Properties → Configurations →
Next → Tick mark Enable Package Configurations → Click Add →



Next → choose ConnectionString from SRC_FILE_CONN and TGT_DB_CONN



Next→Specify Configuration name (Pkg_XML_Config)→ Preview settings→Close.

b)Build->Solution

c) Go to the Pkg_FILE_TABLE_XML_Config file and do the below

Change source and target connection settings [point to different]

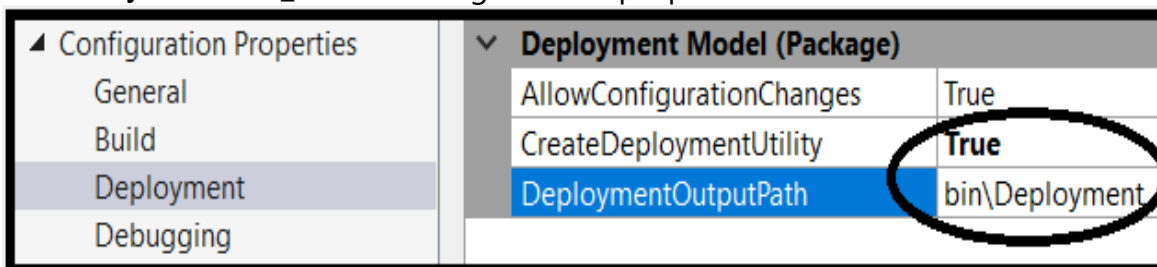
d) Start→Run→DTEXECUI→ Specify package by browsing [File System or SQL Server], configurations→ Browse to configuration file.

Execute

e) Observe the log generated and monitor the output

e)Manifest File Creation






Go to Project MEDW_PROJECT→right click→properties



Build Menu →Solution

Now the deployment folder created under bin folder of the solution with the available packages and manifest files.

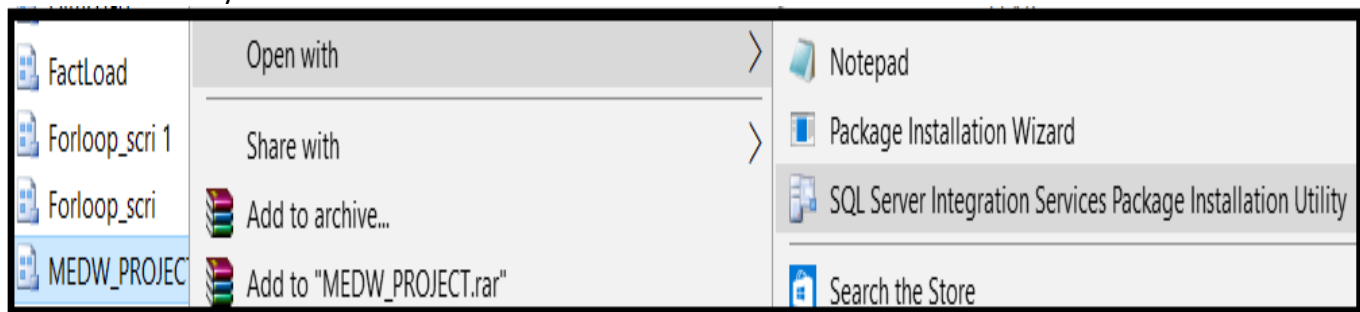
d)Go to Solution Folder→bin folder→ Deployment folder→

MEDW_PROJECT\bin\Deployment			
Name	Date modified	Type	Size
 DimLoad	01/26/2019 8:24 A...	Integration Services Package	11 KB
 FactLoad	01/28/2019 9:02 A...	Integration Services Package	6 KB
 Forloop_scri 1	01/25/2019 6:55 PM	Integration Services Package	47 KB
 Forloop_scri	01/25/2019 6:55 PM	Integration Services Package	35 KB
 MEDW_PROJECT	01/28/2019 10:20 ...	Integration Services Deployment Manifest	1 KB

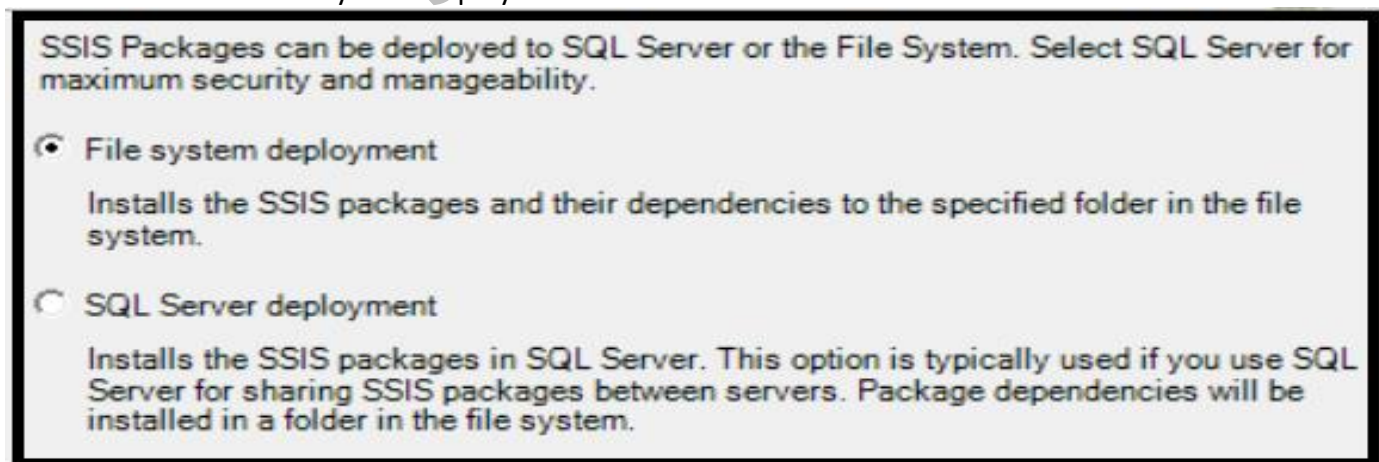
File System Deployment

a) Create a folder named "C:\TEST_ENV"

b) Go to MEDW_PROJECT manifest file → right click → open with → SQL Server Integration Services Packages Installation Utility



Click next → choose File System Deployment



Click Next → Browse to TEST_ENV folder → Click Next

a) Go to TEST_ENV folder and see the packages

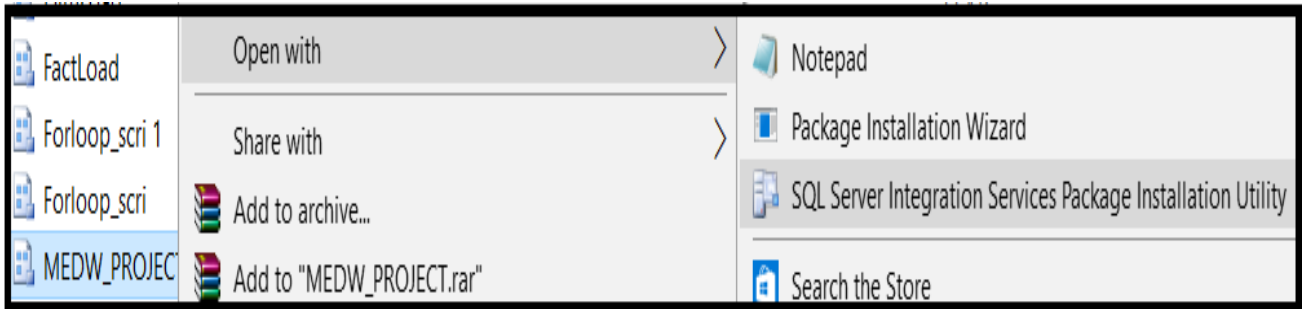
b) To execute any package

Package → right click →

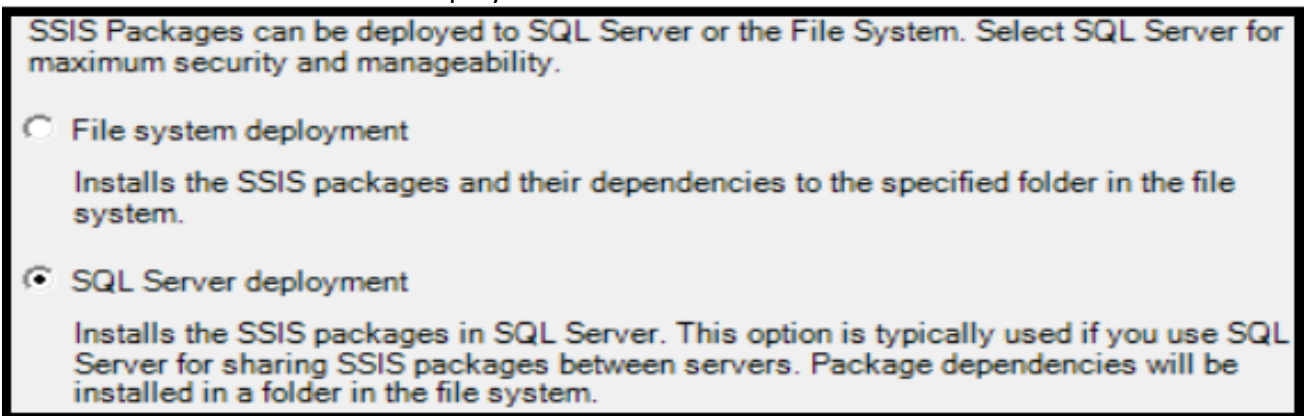


SQL Server Deployment

- a) Go to SSMS-->Integration Services-->Stored Packages-->MSDB-->right click-->New Folder-->MEDW_Project
- b) Go to MEDW_PROJECT manifest file →right click→open with →SQL Server Integration Services Package Installation Utility



- c) Click Next→ choose SQL Server Deployment



- e)Click Next→ Browse to the above MEDW_Project folder→Next→Finish
- f)Go to any package→right click→ Execute Package or Run

Limited Packages Deployment [How do we deploy required packages?]**Three ways**

- a) Remove packages from bin folder and deploy
- b) Open Manifest file, remove the packages and deploy
- c) View->Solution Explorer→SSIS Packages→Package→right click→ Exclude from project

Build solution

Deploy

Note: Exclude will remove from solution visibility, but still available in the folder.

Conclusion:

How many areas packages resided in SSIS?

- a) SSDT—Local
- b) SSMS→SSIS→MSDB
- c) Catalog Database
- d) SQL Store

CONFIGURATIONS USAGE:

Go to the configuration file area [where you saved the xml file in the step d]

Change source file name [may be server name / folder name/ file name]

And target database name [vinaytech_test_db]

Run the package, then you will see the configuration settings data loaded to the respective destination [instead of creation time settings].

Go to the log table and see the below**Use vinaytech_dev_db;**

```
SELECT * FROM SYSSSISLOG ORDER BY STARTTIME DESC
```

```
select * from sysssislog where message='' and event='OnPostExecute'
```

```
--packages successful
```

```
Note: message <>' --paakcge failed
```

```
select * from sysssislog where source='pkg_multi_transform'
and event='OnPostExecute'
```

ADDITIONAL CONFIGURATION TRY

REFER TO MATERIAL [PACKAGE CONDIGURATIONS] AND TRY SQL SERVER CONFIGURATION

ADDITIONAL LOGGING TRY

REFER TO MATERIAL [LOGGING] AND TRY FILE LOGGING.

SECURITY PRACTICE AT PACKAGE DEPLOYMENT MODEL**SSDT LEVEL: PASSWORD / KEY**

- a) SSDT LEVEL, GO TO PACKAGE, CONTROL FLOW→ PROPERTIES, CHHOSE PROTECTION LEVEL: ENCRYPT ALL WITH PASSWORD, AND PROVIDE PASSWORD.

Go to solution at your desktop, click to open, it will prompt for password.

Or

Go to Manifest file, right click→ installation utility→ choose file system/SQL server→ next--

It will prompt you for password.

Note: This password protection will not allow unauthorized modifications and deployment.

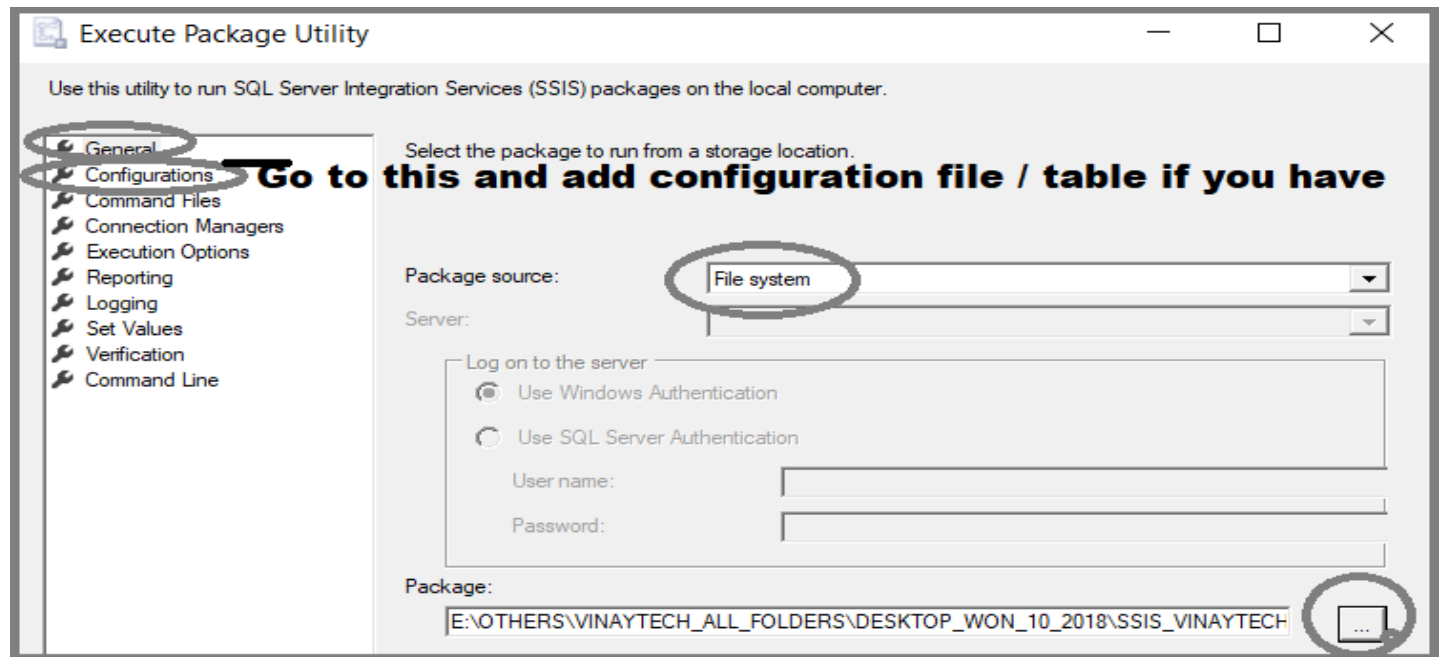
SSMS LEVEL: ROLE BASED SECURITY

SSMS→INTEGRATION SERVICES→ MSDB→ PACKAGE→ RIGHT CLICK→ ADD ROLES AND CHOOSE PERMISSIONS [GRANT / DENY]

PACKAGES RUNNING FROM COMMANDLINE

Without using SSDT running a package quickly

START→RUN→DTEXECUI



NOTE:

READ DETAILED THEORY AT MATERIAL [Where you find other commands used at scheduling and .net script level]