SQL Server Integration Services Chalk-Talk

Module 01: Create Catalog

1. Launch SSMS
2. Integration Services Catalog → right-click → Create Catalog.
3. To create or use SSIS catalog server must have CLR enabled.
4. Enable Automatic Execution of SSIS stored procedure on SQL Server startup.
5. Define password for catalog. This is used to encrypt connection strings.
6. Show configuration settings – don't dive deep into them. Will talk about it in later modules.

Module 02: Demo 01: SSDT Walkthrough

1. Adding Visual Studio Extensions
2. Solution & Projects
3. SSIS Toolbox
4. SSIS Menus – Variables, Build, Debug, Extensions
5. Navigation
6. Properties Pane

Module 02: SSDT Walkthrough

1. Launch SSDT.
2. Explain Recent/Open/New Projects on Start Page.
3. Create a new Project, review various BI project types.
4. Explain the three work spaces (SSIS Toolbox, Designer, Solution Explore/Properties).
5. How to access the toolbox (Menu View > Other > SSIS Toolbox) or the small icon in top right design pane.
6. Center pane (Control Flow, Data Flow, Parameters, Event Handler, Package Explorer)
7. Explain properties pane (F4).
8. Package explore (CTRL + W,S)
9. How to see the variables in the package.
10. Create SQL Task select * from dimAccount
11. Create a basic package with Script Task object attach it to event handler. Force a message "MessageBox.Show("This is a test","Test")".
12. Run to show it is failing so we can see the "Execution Results" & "Output".
13. Fix the script and run again add a ";" at the end.
14. Explain how to bring each of them into view.
15. Explain SSIS menu.
16. Show how to target SQL Server 2016.
17. Deploy simple package to SQL Server.

Module 02: Demo 02: SSMS Walkthrough

1. SSISDB Packages & Dashboard
2. MSDB Packages

3. Connecting to Integration Services

Module 03: Projects vs Packages

1. Create new empty project.
2. Add a Project Connection.
3. Add a Package Connection.
4. Explain the difference.
5. Add a Project Parameter, string and set it to a value[1].
6. Add a Package Parameter, string and set it to a value.
7. Deploy a script object and pass in both variables as read-only.
8. Write following script.. fixing parameter names:
   ```
   string ProjectMessage = Dts.Variables["$Project::ProjectMessage"].Value.ToString();
   string PackageMessage = Dts.Variables["$Package::PackageMessage"].Value.ToString();
   MessageBox.Show("Message From Project Parameter is:" + ProjectMessage + Environment.NewLine +
   Environment.NewLine + "Message From Package Parameter is:" + PackageMessage);
   ```

9. Demonstrate executing sql task with expressions.

Module 03: Connection Manager

1. Create a new empty project.
2. Create a connection for dummy FTP server.
3. Create data flow, show how these connections are visible. However they are not useable.
4. Show the destination and source targets in data flow.
5. Promote connection and de-promote a connection.

Module 03: Variables, Expressions & Parameters

1. Create a new empty project.
2. Create package parameter.
3. Create project parameter.
4. Create a Package Parameter init
5. Create Variable counter, equal
6. Create a Expression Task : equal = init +counter
7. Create a Script Task

```
MessageBox.Show(Dts.Variables["User::equal"].Value.ToString());
```

[1] Example from https://www.tutorialgateway.org/ssis-project-parameters-vs-ssis-package-parameters/

Module 03: Connection String

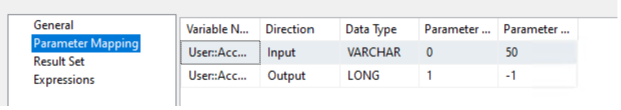
1. Create a new package
2. Create parameters username / password
3. Connection String expression

Module 03: Control Flow: SQL Execute Task (SP)

1. create a new package
2. create a sp

```
ALTER PROCEDURE [dbo].[usp_AccountTypeCounter]
       @accounttype varchar(16), @counter int OUTPUT
AS
BEGIN
       -- SET NOCOUNT ON added to prevent extra result sets from
       -- interfering with SELECT statements.
       SET NOCOUNT ON;
       set @counter = (
   select count(*)
  FROM [AdventureWorksDW2012].[dbo].[DimAccount]
  where AccountType =@accounttype)
END
```

3. create a SQL Task `exec usp_AccountTypeCounter 'Expenditures' ,@counter output`
4. Create Var “AccountCount”, “AccountType”
5. SQL Task para input and output mapping



Variable N...	Direction	Data Type	Parameter ...	Parameter ...
User::Acc...	Input	VARCHAR	0	50
User::Acc...	Output	LONG	1	-1

6. Create a Script task

MessageBox.Show(Dts.Variables["User::AccountCount"].Value.ToString());

Module 03: Control Flow: SQL Execute Task(Single row)

1. Create a new package
2. Create a SQL Task select count(*) from DimAccount where AccountType = ?
3. Create a var “AccountCount”, “AccountType” = Expenditures
4. Asign to resulteset
5. Create a script task

MessageBox.Show(Dts.Variables["User::AccountCount"].Value.ToString());

6. Change to This to show the information

```
bool fireAgain = false;
```

```
string description = "Counter Value= " + Dts.Variables["User::AccountCount"].Value.ToString();
Dts.Events.FireInformation(0, null, description, null, 0, ref fireAgain);
```

Module3 Bright it Together

1. Create a Report Error Script Task

```
MessageBox.Show("Uhmm! I think something went wrong!", "Help!")
```

2. Change AccountType to int
3. Create a variable AccountTypeint int
4. Create a fx Convert string to int
5. Edit constrain to fail and or
6. Run the package

Module Data Viewer: part 1

1. Create a new empty project.
2. Create OLEDB Connection Manager to AdventureWorks2016.
3. Create Variable called Counter
4. Create Data Flow Task
5. Add OLEDB Source to Data Flow tab
6. Set source to sql command:
   a. SELECT a.* from Person.Person a
7. Add Row Count Transform
8. Add Data Viewer
9. Run and package is paused until you press Play, and that there is only one buffer.

Module Data Viewer: part 2

6. modify source sql, add CROSS JOIN Person.Person b.
7. show multiple buffers, attach, detach, reattach after some time .

Module4: Transaction

1. create table with PK
2. create a new Packages
3. create time sequence
4. create delete

5. create insert duplicate data to pk column
6. change TransactionOption and View Data

- CREATE TABLE [dbo].[Person](

[id] [bigint] NOT NULL,

CONSTRAINT [PK_Persion] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

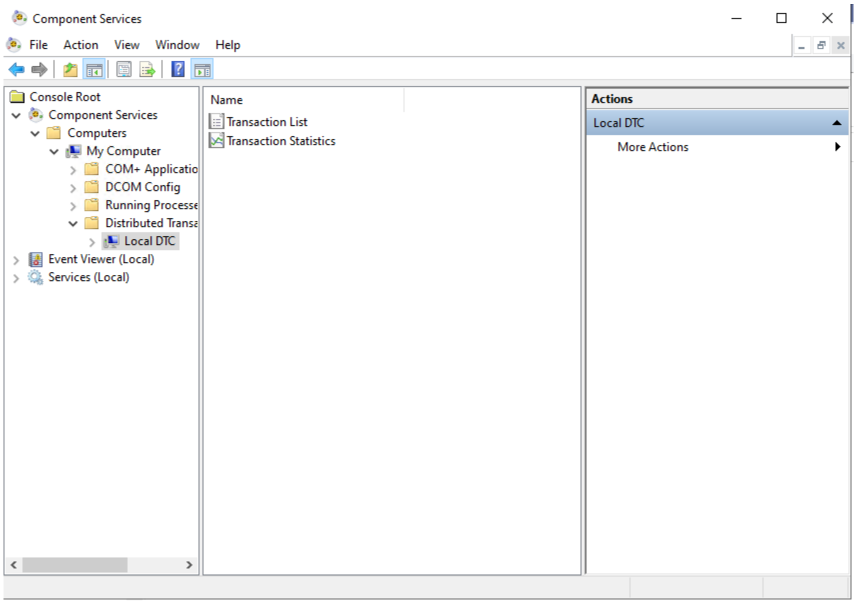
) ON [PRIMARY]

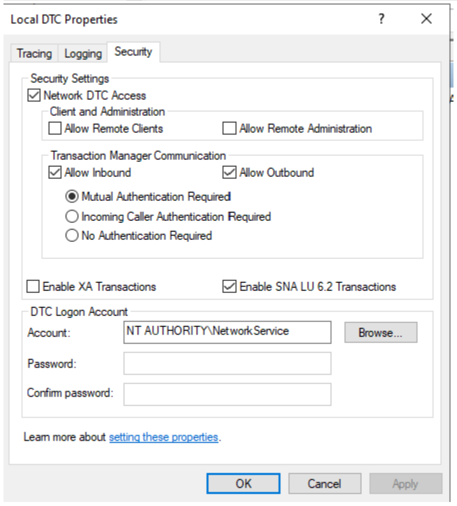
- insert into person (id) values(1),(2)

SSIS Paclage

- delete person where id =2
- select 2 id
- union all select 2

https://www.ge.com/digital/documentation/meridium/V36160/Help/Master/Subsystems/Installation/Content/Setting_the_Local_DTC_Property_Settings_-_Database_Server.htm




Local DTC Properties

Tracing | Logging | Security

Security Settings

☑ Network DTC Access

Client and Administration

☐ Allow Remote Clients ☐ Allow Remote Administration

Transaction Manager Communication

☑ Allow Inbound ☑ Allow Outbound

● Mutual Authentication Required

○ Incoming Caller Authentication Required

○ No Authentication Required

☐ Enable XA Transactions ☑ Enable SNA LU 6.2 Transactions

DTC Logon Account

Account:	NT AUTHORITY\NetworkService
Password:	
Confirm password:	

Learn more about setting these properties.