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: Demo 02: SSMS Walkthrough

1. SSISDB Packages & Dashboard
2. MSDB Packages
3. Connecting to Integration Services

Module 03: Demo 01: Projects vs Packages

1. Create a project and explore the differences.
   1. Package vs Project
   2. Parameters
   3. Connections
   4. Design Layout

Module 03: Demo 02: Connection Manager

1. Create an explore different types of connections for various data sources in control flow.
   1. Convert package to project.
   2. Convert project to package.
   3. Duplicate name issue.
   4. Missing connection information.
   5. Why would you consider package vs project connection?
2. Data flow
   1. Connection mapping to input and output.

Showing Difference between PER\_PACKAGE and PER\_EXECUTION encryption level.

1. Show the current list of Certificates and Symmetric Keys in SSISDB.
2. Deploy a demo project with at least two package.
3. View Cert and Keys (before execution)
4. Execute Package
5. View Cert and Keys (after 1st execution)
6. Execute Package
7. View Cert and Keys (notice only one set for package regardless of number of execution.

Now lets change setting from 2 to 1 (PER\_EXECUTION)

EXEC SSISDB.internal.cleanup\_server\_log

ALTER DATABASE SSISDB SET SINGLE\_USER

--change SEVER\_OPERATION\_ENCRYPTION\_LEVEL to PER\_PROJECT (2) or PER\_EXECUTION (1)

EXEC SSISDB.catalog.configure\_catalog @property\_name='SERVER\_OPERATION\_ENCRYPTION\_LEVEL', @property\_value='1'

ALTER DATABASE SSISDB SET MULTI\_USER

--clean up transaction level keys.

EXEC SSISDB.internal.Cleanup\_Server\_execution\_keys @cleanup\_flag = 1

1. View Cert and Keys (before execution)
2. Execute Package
3. View Cert and Keys (after 1st execution)
4. Execute Package
5. View Cert and Keys (notice set certs and sym keys create each time package is executed. )

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 a basic package with SQL Task object attach it to event handler. Force a message “MessageBox.Show(“This is a test”,”Test”)”.
11. Run to show it is failing so we can see the “Execution Results” & “Output”.
12. Fix the script and run again add a “;” at the end.
13. Explain how to bring each of them into view.
14. Explain SSIS menu.
15. Show how to target SQL Server 2016.
16. Deploy simple package to SQL Server.

Module 02: SSMS Walkthrough

1. Go execute package we just deployed in Demo 1.
2. Show how to look at the dashboard reports.
3. Explain the various aspects of the reports.
4. Show how to schedule a package.
5. Create another SSIS package that has “WAITFOR DELAY (’00:05:00’)” and deploy the package.
6. Show how to monitor execution.
7. Show how to stop current running packages.

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]](#footnote-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 two packages, 1st package only have package parameter. 2nd package uses project parameter. Show the configured value vs execution value.
5. Create two variables.
6. Assign one package level and one task level scope.
7. Create table task.
8. Create expression to insert into table from two tasks.
9. Show data in database.
10. Also demonstrate how to set expressions for Data Flow Tasks.

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:
   1. 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

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

1. Example from https://www.tutorialgateway.org/ssis-project-parameters-vs-ssis-package-parameters/ [↑](#footnote-ref-1)