Module 7

Implementing Control Flow in an SSIS Package

Zyead Ahmed.
Aspiring To Learning Data Engineer.

Module Overview

- Introduction to Control Flow
- Creating Dynamic Packages
- Using Containers
- Managing Consistency

Lesson 1: Introduction to Control Flow

- Control Flow Tasks
- Precedence Constraints
- Grouping and Annotations
- Demonstration: Implementing Control Flow
- Using Multiple Packages
- Create a Package Template

Control Flow Tasks

- Data Flow tasks
- Database tasks
- File and Internet tasks
- Process Execution tasks
- WMI tasks
- Custom Logic tasks
- Database Transfer tasks
- Analysis Services tasks
- SQL Server Maintenance tasks

Precedence Constraints

Precedence constraints connect sequences of tasks

- Three control flow conditions
  - Success
  - Failure
  - Completion

- Multiple constraints
  - Logical AND
  - Logical OR

Grouping and Annotations

- Group tasks to manage them as a unit at design time
  - Show/Hide
  - Move
- Add annotations to provide documentation

Demonstration: Implementing Control Flow

In this demonstration, you will see how to:

- Add tasks to a control flow
- Use precedence constraints to define a control flow

Using Multiple Packages

With multiple packages you can:

- Create reusable units of workflow
- Run multiple control flows in parallel
- Separate ETL workflows to fit data acquisition windows

Create a Package Template

1. Create a package that contains elements you want to reuse
   - Connection Managers
   - Tasks
   - Event Handlers
   - Parameters and Variables
2. Save the package to the **DataTransformationItems** folder
3. Add the package to a project from the **Add New Item** dialog box
4. Change the package name and ID

Lesson 2: Creating Dynamic Packages

- Variables
- Parameters
- Expressions
- Demonstration: Using Variables and Parameters

Variables

- User Variables:
  - Variables created by an SSIS developer to hold dynamic values
  - Defined in the **User** namespace by default
  - Defined at a specified scope
- System Variables
  - Built-in variables with dynamic system values
  - Defined in the **System** namespace

Parameters

- Project parameters
  - Accessible from any package in the project
- Package parameters
  - Exist only at the package level

Expressions

- Expressions set values dynamically:
  - Properties
  - Conditional split criteria
  - Derived column values
  - Precedence constraints
- Based on SSIS expression syntax
  - Can include variables and parameters
- Type expressions or create using Expression Builder

Demonstration: Using Variables and Parameters

In this demonstration, you will see how to:

- Create a variable
- Create a parameter
- Use variables and parameters in an expression

Introduction to Containers

- Task containers
- Sequence containers
- For Loop containers
- Foreach Loop containers

Sequence Containers

Sequence containers

- Define a control flow subset
- Help you to manage properties for multiple tasks
- Create a scope for variables, transactions, and precedence

Demonstration: Using a Sequence Container

In this demonstration, you will see how to use a Sequence container

For Loop Containers

For Loop containers

- Implement iterative control flow
- Similar to a C# For Loop
  - Initialization expression

    ```
    @Count = 0
    ```

  - Evaluation expression

    ```
    @Count < 4
    ```

  - Iteration expression

    ```
    @Count = @Count + 1
    ```

Demonstration: Using a For Loop Container

In this demonstration, you will see how to use a For Loop container

Foreach Loop Containers

Iterate through an enumerated collection

- ADO
  - Rows in a recordset
- ADO.NET Schema Rowset
  - Objects in a database schema
- File
  - Files in a folder
- Variable
  - Elements in an array variable
- Item
  - Enumerated property values of an item
- Nodelist
  - Nodes in an XML document
- SMO
  - SQL Server Management Objects

Demonstration: Using a Foreach Loop Container

In this demonstration, you will see how to use a Foreach Loop container

Lab A: Implementing Control Flow in an SSIS Package

- Exercise 1: Using Tasks and Precedence in a Control Flow
- Exercise 2: Using Variables and Parameters
- Exercise 3: Using Containers

Logon Information
Virtual machine: **20767C-MIA-SQL**
User name: **ADVENTUREWORKS\Student**
Password: **Pa55w.rd**

Estimated Time: 60 minutes

Lab Scenario

You are implementing an ETL solution for Adventure Works Cycles and must ensure that the data flows you have already defined are executed as a workflow that notifies operators of success or failure by sending an email message. You must also implement an ETL solution that transfers data from text files generated by the company’s financial accounting package to the data warehouse.

Lab Review

Having completed this lab, you will now be able to:

- Use tasks and precedence constraints
- Use variables and parameters
- Use containers

Lesson 4: Managing Consistency

- Configuring Failure Behavior
- Using Transactions
- Demonstration: Using a Transaction
- Using Checkpoints
- Demonstration: Using a Checkpoint

Configuring Failure Behavior

Use properties to control failure propagation:

- FailPackageOnFailure
- FailParentOnFailure
- MaximumErrorCount

Using Transactions

Set the **TransactionOption** property of a task, container, or package:

- Required
- Supported
- NotSupported

Demonstration: Using a Transaction

In this demonstration, you will see how to use a transaction

Using Checkpoints

- Checkpoints allow failed packages to be restarted without repeating previously successful tasks
- Enable checkpoints by setting package properties:
  - CheckpointFileName
  - CheckpointUsage
  - SaveCheckpoints

Demonstration: Using a Checkpoint

In this demonstration, you will see how to use a checkpoint