Control Flow\_Precedence Constraints

Precedence Constraints

*Precedence constraints* are the connections between the tasks that control

the execution order of each task. After you drag in more than one task in the Control Flow in

SSIS, you can link them together by using these precedence constraints. Click once on a task,

and you see a green arrow pointing down from the task; this is the precedence constraint line

for this task.

**Green** = On Success

➤➤ **Red** = On Failure

➤➤ **Blue** = On Completion

**Step-by-Step**

**1.** Drag four Script Tasks into the Control Flow.

**2.** Drag the precedence constraint from Script Task to Script Task 1.

**3.** Drag the precedence constraint from Script Task 1 to Script Task 2.

**4.** Drag the precedence constraint from Script Task 3 to Script Task 2.

**5.** Run the package; a green check mark indicating success should appear in the top right of

each task, as shown in Figure



**6.** Stop the debugging using the square Stop button on the

toolbar.

**7.** Double-click one of the constraint arrows going into

Script Task 2.

**8.** Change the logical constraint to OR; the two lines in

Script Task 2 change to dotted lines.

**9.** Click Script Task 3.

**10.** In the Properties window, change the

ForceExecutionResult to Failure

****

**11.** Run the package. Script Task 3 should have a red “X” in

the top right indicating failure, and all other tasks should

have a green check mark in the top right. Notice that

Script Task 2 ran even though Script Task 3 failed.

**Execute SQL Task**

When you are creating a SQL Server Integration Services (SSIS) package, you will find that one of the most commonly used tasks is the Execute SQL Task. This task is used to insert, update, select, and truncate data from SQL tables. Any normal SQL commands you would use can be used in this task. You can use parameters just like a stored procedure and can even call stored procedures from the task. A connection to the database must exist in the connection manager

for the Execute SQL Task to reference.

**Step-by-Step**

**1.** Drag in an Execute SQL Task and double-click the task to open the editor.

**2.** Click the connection and select New Connection.

**3.** Create a connection to the AdventureWorks2012 database.

**4.** Select Single Row as the result set.

**5.** Select Direct Input as the SQL type.

**6.** Click the SQL command and enter the following query:

Select Count(\*) as Counter from Production.Product Where ProductID = ?

**7.** In the Parameter Mapping node, click Add and create a parameter with the name of **0**.

While in the Parameter Mapping node, click the Variable Name drop-down menu and select

New Variable.

**9.** Create an integer (int32) variable named **intProductID** and set the value to 316.

**10.** Click the Result Set node and click Add to create a result set with the name of **0**.

**11.** In the Result Set node, click the Variable Name drop-down and select New Variable.

**12.** Create another Int32 variable named **intProductCount**.

**13.** Drag a Script Task into the Control Flow of the package.

**14.** Connect the Execute SQL Task to the Script Task with an On Success Precedence Constraint.

**15.** Double-click the Script Task and select intProductCount in the ReadOnlyVariables of the

Script Task.

**16.** Click the Edit Script button.

**17.** Type the following VB code in the script editor (refer to Lesson 11 for a Script Task

explanation):

Msgbox(DTS.Variables(“intProductCount”).Value)

**18.** Close the script editor.

**19.** Click OK in the Script Task.

**20.** The package should look like Figure 12-6. Click Debug on the toolbar to run the package.

**Figure 12-6**

**21.** A popup message should appear showing the intProductCount variable, which should have a

value of 1, as shown in



A popup message should appear showing the intProductCount variable, which should have a

value of 1



Send Mail Task

**Step-by-Step**

**1.** Drag a Send Mail Task into a blank package.

**2.** Right-click in the connection manager and select New Connection.

**3.** Select the SMTP connection from the list and click Add



Change the SMTP connection name to your company name and SMTP, for example, **Your**

**Server Name**.

**5.** Set the SMTP connection description to **My companys SMTP Server**.

**6.** Set the SMTP connection server to the actual SMTP Server connection.

**7.** Place a check in Windows Authentication if your company uses Windows Authentication to

send SMTP e‑mail.

**8.** Place a check in Enable Secure Sockets Layer (SSL) if your SMTP server requires a secure

connection.

**9.** Once you have completed the previous steps, the SMTP connection should look like



Click OK in both open windows to return to the Control Flow.

**11.** Double-click the Send Mail Task to open the editor.

**12.** Name the Send Mail Task **Send Package Info**.

**13.** Set the Send Mail Task description to **Send email to users containing the package**

**information**.

**14.** Click the Mail node on the left-hand side of the Send Mail Task Editor window.

**15.** Set the SMTPConnection to the SMTP connection you created in Steps 2–9.

**16.** Set the From address to your e‑mail address.

**17.** Set the To address to your e‑mail address. (If you have two e‑mail addresses, you can set

From and To to the two different e‑mail addresses. This is true as long as the SMTP server

allows you to send and receive e‑mail from these e‑mail addresses.)

Set the Subject line to **Email From Package**.

**19.** Set the MessageSourceType to Direct Input.

**20.** Set the MessageSource to **The Send Mail Package Finished**. The Send Mail Task should look

similar to Figure



Click OK.

**22.** Run the package by clicking the green debug arrow on the toolbar; you should receive an

e‑mail from yourself.