Add the following stored procedure to the database where your pipeline parameter table resides. This procedure simply looks up the destination table name in the pipeline parameter table and updates the status and datetime for each table once the Copy data activity is successful.

SET quoted\_identifier ON  
  
go  
  
CREATE PROCEDURE [dbo].[Sql2adls\_data\_files\_loaded] @dst\_name NVARCHAR(500)  
AS  
    SET nocount ON  
    -- turns off messages sent back to client after DML is run, keep this here  
  
    DECLARE @Currentday DATETIME = Getdate();  
  
    UPDATE [dbo].[pipeline\_parameter]  
    SET    pipeline\_status = 'success',  
           pipeline\_date = @Currentday  
    WHERE  dst\_name = @dst\_name;  
  
go

Here is the corresponding source code used in Figure 8-7.

SELECT '@{pipeline().DataFactory}'                                                               AS datafactory\_name,  
       '@{pipeline().Pipeline}'                                                                  AS pipeline\_name,  
       '@{pipeline().RunId}'                                                                     AS runid,  
       '@{item().src\_name}'                                                                      AS source,  
       '@{item().dst\_name}'                                                                      AS destination,  
       '@{pipeline().TriggerType}'                                                               AS triggertype,  
       '@{pipeline().TriggerId}'                                                                 AS triggerid,  
       '@{pipeline().TriggerName}'                                                               AS triggername,  
       '@{pipeline().TriggerTime}'                                                               AS triggertime,  
       '@{activity('copy-TABLE').output.rowsCopied}'                                             AS rowscopied,  
       '@{activity('copy-TABLE').output.rowsRead}'                                               AS rowsread,  
       '@{activity('copy-TABLE').output.usedParallelCopies}'                                     AS no\_parallelcopies,  
       '@{activity('copy-TABLE').output.copyDuration}'                                           AS copyduration\_in\_secs,  
       '@{activity('copy-TABLE').output.effectiveIntegrationRuntime}'                            AS effectiveintegrationruntime,  
       '@{activity('copy-TABLE').output.executionDetails[0].source.type}'                        AS source\_type,  
       '@{activity('copy-TABLE').output.executionDetails[0].sink.type}'                          AS sink\_type,  
       '@{activity('copy-TABLE').output.executionDetails[0].status}'                             AS execution\_status,  
       '@{activity('copy-TABLE').output.executionDetails[0].start}'                              AS copyactivity\_start\_time,  
       '@{utcnow()}'                                                                             AS copyactivity\_end\_time,  
       '@{activity('copy-TABLE').output.executionDetails[0].detailedDurations.queuingDuration}'  AS copyactivity\_queuingduration\_in\_secs,  
       '@{activity('copy-TABLE').output.executionDetails[0].detailedDurations.timeToFirstByte}'  AS copyactivity\_timetofirstbyte\_in\_secs,  
       '@{activity('copy-TABLE').output.executionDetails[0].detailedDurations.transferDuration}' AS copyactivity\_transferduration\_in\_secs

The following parameterized path will ensure that the file is generated in the correct folder structure. Here is the code shown in Figure 8-9.

@{item().server\_name}/@{item().src\_db}/@{item().src\_schema}/@{item().dst\_name}/metadata/@{formatDateTime(utcnow(),'yyyy-MM-dd')}/@{item().dst\_name}.csv

Create the following table in your ADF\_DB database. This table will store and capture the pipeline and copy activity details.

SET ansi\_nulls ON  
  
go  
  
SET quoted\_identifier ON  
  
go  
  
CREATE TABLE [dbo].[pipeline\_log]  
  (  
     [log\_id]                                [INT] IDENTITY(1, 1) NOT NULL,  
     [parameter\_id]                          [INT] NULL,  
     [datafactory\_name]                      [NVARCHAR](500) NULL,  
     [pipeline\_name]                         [NVARCHAR](500) NULL,  
     [runid]                                 [NVARCHAR](500) NULL,  
     [source]                                [NVARCHAR](500) NULL,  
     [destination]                           [NVARCHAR](500) NULL,  
     [triggertype]                           [NVARCHAR](500) NULL,  
     [triggerid]                             [NVARCHAR](500) NULL,  
     [triggername]                           [NVARCHAR](500) NULL,  
     [triggertime]                           [NVARCHAR](500) NULL,  
     [rowscopied]                            [NVARCHAR](500) NULL,  
     [dataread]                              [INT] NULL,  
     [no\_parallelcopies]                     [INT] NULL,  
     [copyduration\_in\_secs]                  [NVARCHAR](500) NULL,  
     [effectiveintegrationruntime]           [NVARCHAR](500) NULL,  
     [source\_type]                           [NVARCHAR](500) NULL,  
     [sink\_type]                             [NVARCHAR](500) NULL,  
     [execution\_status]                      [NVARCHAR](500) NULL,  
     [copyactivity\_start\_time]               [NVARCHAR](500) NULL,  
     [copyactivity\_end\_time]                 [NVARCHAR](500) NULL,  
     [copyactivity\_queuingduration\_in\_secs]  [NVARCHAR](500) NULL,  
     [copyactivity\_transferduration\_in\_secs] [NVARCHAR](500) NULL,  
     CONSTRAINT [PK\_pipeline\_log] PRIMARY KEY CLUSTERED ( [log\_id] ASC )WITH (  
     statistics\_norecompute = OFF, ignore\_dup\_key = OFF) ON [PRIMARY]  
  )  
ON [PRIMARY]  
  
go