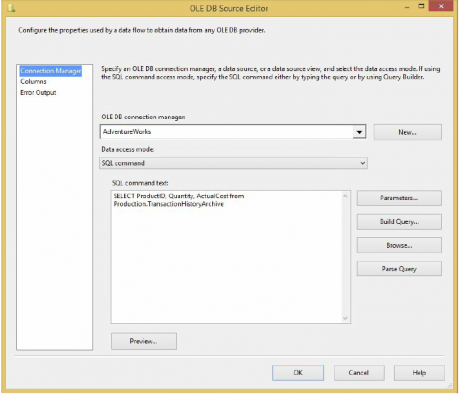
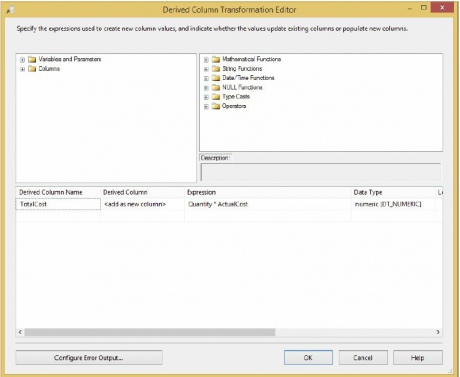
1. Create a new package and rename it AdventureWorksExtract.dtsx. Start by dragging a Data Flow Task onto the Control Flow. Double-click the task to open the Data Flow tab.
2. In the Data Flow tab, drag an OLE DB Source onto the design pane. Right click the source and rename it Transaction History. Double-click it to open the editor. Click the New button next to the OLE DB Connection Manager Dropdown box. The connection to the AdventureWorks database may already be in the Data Connections list on the left. If it is, select it and click OK. Otherwise, click New to add a new connection to the AdventureWorks database on any server.
3. When you click OK, you’ll be taken back to the OLE DB Source Editor. Ensure that the Data Access Mode option is set to SQL Command. Type the following query for the command, as shown in Figure 4-45 or as follows:

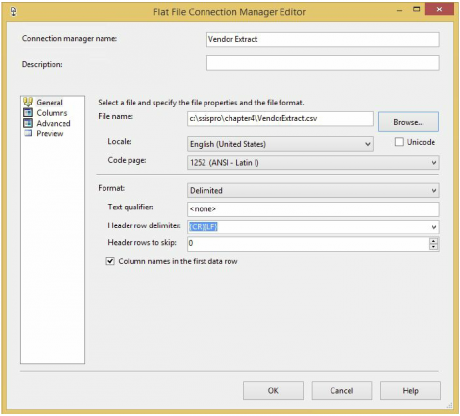


SELECT ProductID, Quantity, ActualCost from Production.TransactionHistoryArchive

1. Drag a Derived Column Transformation onto the Data Flow, right-click it, and select Rename. Rename the transform Calculate Total Cost. Click the TransactionHistory OLE DB Source and drag the green arrow (the data path) onto the Derived Column Transformation.
2. Double-click the Derived Column Transformation to open the editor (shown in Figure 4-46). For the Expression column, type the following code or drag and drop the column names from the upper-left box: [Quantity]∗ [ActualCost]. The Derived Column should have the option selected, and type TotalCost for the Derived Column Name option. Click OK to exit the editor.



1. Drag an Aggregate Transformation onto the Data Flow and rename it Aggregate Data. Drag the blue arrow from the Derived Column Transformation onto this transformation. Double-click the Aggregate Transformation to open its editor (shown in Figure 4-47). Select the ProductID column and note that it is transposed into the bottom section. The ProductID column should have Group By for the Operation column. Next, check the Quantity and TotalCost columns and set the operation of both of these columns to Sum. Click OK to exit the editor.
2. Drag a Sort Transformation onto the Data Flow and rename it Sort by Quantity. Connect the Aggregate Transformation to this transformation by the blue arrow as in the preceding step. Double-click the Sort Transformation to configure it in the editor. You can sort by the most popular products by checking the Quantity column and selecting Descending for the Sort Type dropdown box. Click OK to exit the editor.
3. You have now done enough massaging of the data and are ready to export the data to a flat file that can be consumed by another vendor. Drag a Flat File Destination onto the Data Flow. Connect it to the Sort Transformation by using the blue arrow as shown in the last few steps. Rename the Flat File Destination **Vendor Extract**.
4. Double-click the destination to open the Flat File Destination Editor. You’re going to output the data to a new Connection Manager, so click New. When prompted for the Flat File Format, select Delimited. Name the Connection Manager **Vendor Extract** also, and type whatever description you’d like. If you have the directory, point the File Name option to C:ProSSISData FlowVendorExtract.csv (make sure this directory is created before proceeding). Check the “Column names in the first data row” option. Your final screen should look like Below the screen shoot. Click OK to go back to the Flat File Destination Editor.



1. From the Mappings page, ensure that each column in the Inputs table is mapped to the Destination table. Click OK to exit the editor and go back to the Data Flow.  
   Now your first larger ETL package is complete! This package is very typical of what you’ll be doing daily inside of SSIS, and you will see this expanded on greatly, in Creating an End-to-End Package Topic. Execute the package. You should see the rows flow through the Data Flow, as shown in Figure below. Note that as the data flows from transformation to transformation, you can see how many records were passed through.

