**Overview of SSIS**

SQL Server Integration Services (SSIS) is a tool that we use to perform ETL operations; i.e. extract, transform and load data.  While ETL processing is common in data warehousing (DW) applications, SSIS is by no means limited to just DW; e.g. when you create a Maintenance Plan using SQL Server Management Studio (SSMS) an SSIS package is created.  At a high level, SSIS provides the ability to:

* retrieve data from just about any source
* perform various transformations on the data; e.g. convert from one type to another, convert to uppercase or lowercase, perform calculations, etc.
* load data into just about any source
* define a workflow

The first version of SSIS was released with SQL Server 2005.  SSIS is a replacement for Data Transformation Services (DTS) which was available with SQL Server 7.0 and SQL Server 2000.  SSIS builds on the capabilities introduced with DTS.

In this tutorial we will step through a number of topics that you need to understand in order to successfully build an SSIS package.  Our high level outline is as follows:

* Creating SSIS packages with SQL Server Management Studio (SSMS)
* Business Intelligence Development Studio (BIDS)
* Creating a simple SSIS package in BIDS
* Deploying SSIS packages
* Executing SSIS packages
* Step 1: Creating a New Integration Services Project
* Step 2: Adding and Configuring a Flat File Connection Manager
* Step 3: Adding and Configuring an OLE DB Connection Manager
* Step 4: Adding a Data Flow Task to the Package
* Step 5: Adding and Configuring the Flat File Source
* Step 6: Adding and Configuring the Lookup Transformations
* Step 7: Adding and Configuring the OLE DB Destination

**To create a new Integration Services project**

1. On the **Start** menu, point to **All Programs**, point to **Microsoft SQL Server**, and click **SQL Server Data Tools**.
2. On the **File** menu, point to **New**, and click **Project** to create a new Integration Services project.
3. In the **New Project** dialog box, expand the **Business Intelligence** node under **Installed Templates**, and select **Integration Services Project** in the **Templates** pane.
4. In the **Name** box, change the default name to **SSIS Tutorial**. Optionally, clear the **Create directory for solution** check box.
5. Accept the default location, or click **Browse** to browse to locate the folder you want to use. In the **Project Location** dialog box, click the folder and click **Select Folder**.
6. Click **OK**.

By default, an empty package, titled **Package.dtsx**, will be created and added to your project under SSIS Packages.

1. In **Solution Explorer** toolbar, right-click **Package.dtsx**, click **Rename**, and rename the default package to **Lesson 1.dtsx**.

# Lesson 1-2 - Adding and Configuring a Flat File Connection Manager

You must create a new Flat File connection manager for each file format that you work with. Because this tutorial extracts data from multiple flat files that have exactly the same data format, you will need to add and configure only one Flat File connection manager for your package.

For this tutorial, you will configure the following properties in your Flat File connection manager:

* **Column names:** Because the flat file does not have column names, the Flat File connection manager creates default column names. These default names are not useful for identifying what each column represents. To make these default names more useful, you need to change the default names to names that match the fact table into which the flat file data is to be loaded.
* **Data mappings:** The data type mappings that you specify for the Flat File connection manager will be used by all flat file data source components that reference the connection manager. You can either manually map the data types by using the Flat File connection manager, or you can use the **Suggest Column Types** dialog box. In this tutorial, you will view the mappings suggested in the **Suggest Column Types** dialog box and then manually make the necessary mappings in the **Flat File Connection Manager Editor** dialog box.

The Flat File connection manager provides locale information about the data file. If your computer is not configured to use the regional option English (United States), you must set additional properties in the **Flat File Connection Manager Editor** dialog box.

**To add a Flat File connection manager to the SSIS package**

1. Right-click anywhere in the **Connection Managers** area, and then click **New Flat File Connection**.
2. In the **Flat File Connection Manager Editor** dialog box, for **Connection manager name**, type **Sample Flat File Source Data**.
3. Click **Browse**.
4. In the **Open** dialog box, locate the SampleCurrencyData.txt file on your machine.

The sample data is included with the SSIS lesson packages. To download the sample data and the lesson packages, do the following.

* 1. Navigate to Integration Services Product Samples
  2. Click the **DOWNLOADS** tab.
  3. Click the SQL2012.Integration\_Services.Create\_Simple\_ETL\_Tutorial.Sample.zip file.

1. Clear the Column names in the first data row checkbox.

**To set locale sensitive properties**

1. In the **Flat File Connection Manager Editor** dialog box, click **General**.
2. Set **Locale** to English (United States) and **Code page** to 1252.

**To rename columns in the Flat File connection manager**

1. In the **Flat File Connection Manager Editor** dialog box, click **Advanced**.
2. In the property pane, make the following changes:
   * Change the **Column 0** name property to **AverageRate**.
   * Change the **Column 1** name property to **CurrencyID**.
   * Change the **Column 2** name property to **CurrencyDate**.
   * Change the **Column 3** name property to **EndOfDayRate**.

**To remap column data types**

1. In the **Flat File Connection Manager Editor** dialog box, click **Suggest Types**.

Integration Services automatically suggests the most appropriate data types based on the first 200 rows of data. You can also change these suggestion options to sample more or less data, to specify the default data type for integer or Boolean data, or to add spaces as padding to string columns.

For now, make no changes to the options in the **Suggest Column Types** dialog box, and click **OK** to have Integration Services suggest data types for columns. This returns you to the **Advanced** pane of the **Flat File Connection Manager Editor** dialog box, where you can view the column data types suggested by Integration Services. (If you click **Cancel**, no suggestions are made to column metadata and the default string (DT\_STR) data type is used.)

In this tutorial, Integration Services suggests the data types shown in the second column of the following table for the data from the SampleCurrencyData.txt file. However, the data types that are required for the columns in the destination, which will be defined in a later step, are shown in the last column of the following table.

| **Flat File Column** | **Suggested Type** | **Destination Column** | **Destination Type** |
| --- | --- | --- | --- |
| AverageRate | float [DT\_R4] | FactCurrency.AverageRate | float |
| CurrencyID | string [DT\_STR] | DimCurrency.CurrencyAlternateKey | nchar(3) |
| CurrencyDate | date [DT\_DATE] | DimDate.FullDateAlternateKey | date |
| EndOfDayRate | float [DT\_R4] | FactCurrency.EndOfDayRate | float |

The data type suggested for the **CurrencyID** column is incompatible with the data type of the field in the destination table. Because the data type of DimCurrency.CurrencyAlternateKey is nchar (3), **CurrencyID** must be changed from string [DT\_STR] to Unicode string [DT\_WSTR]. Additionally, the field DimDate.FullDateAlternateKey is defined as a date data type; therefore, **CurrencyDate** needs to be changed from date [DT\_Date] to database date [DT\_DBDATE].

1. In the list, select the CurrencyID column and in the property pane, change the Data Type of column **CurrencyID** from string [DT\_STR] to Unicode string [DT\_WSTR].
2. In the property pane, change the data type of column **CurrencyDate** from date [DT\_DATE] to database date [DT\_DBDATE].
3. Click **OK**.

# Lesson 1-3 - Adding and Configuring an OLE DB Connection Manager

**Add and configure an OLE DB Connection Manager to the SSIS package**

1. Right-click anywhere in the **Connection Managers** area, and then click **New OLE DB Connection**.
2. In the **Configure OLE DB Connection Manager** dialog box, click **New**.
3. For **Server name**, enter **localhost**.

When you specify localhost as the server name, the connection manager connects to the default instance of SQL Server on the local computer. To use a remote instance of SQL Server, replace localhost with the name of the server to which you want to connect.

1. In the **Log on to the server** group, verify that **Use Windows Authentication** is selected.
2. In the **Connect to a database** group, in the **Select or enter a database name** box, type or select **AdventureWorksDW2012**.
3. Click **Test Connection** to verify that the connection settings you have specified are valid.
4. Click **OK**.
5. Click **OK**.
6. In the **Data Connections** pane of the **Configure OLE DB Connection Manager** dialog box, verify that **localhost.AdventureWorksDW2012** is selected.
7. Click **OK**.

# Lesson 1-4 - Adding a Data Flow Task to the Package

### To add a Data Flow task

1. Click the **Control Flow** tab.
2. In the **SSIS Toolbox**, expand **Favorites**, and drag a **Data Flow Task** onto the design surfaceof the **Control Flow** tab.
3. On the **Control Flow** design surface, right-click the newly added **Data Flow Task**, click **Rename**, and change the name to **Extract Sample Currency Data**.

It is good practice to provide unique names to all components that you add to a design surface. For ease of use and maintainability, the names should describe the function that each component performs. Following these naming guidelines allows your Integration Services packages to be self-documenting. Another way to document your packages is by using annotations. For more information about annotations, see Use Annotations in Packages.

1. Right-click the Data Flow task, click **Properties**, and in the Properties window, verify that the **LocaleID** property is set to **English (United States)**.

# Lesson 1-5 - Adding and Configuring the Flat File Source

**To add a Flat File Source component**

1. Open the **Data Flow** designer, either by double-clicking the **Extract Sample Currency Data**data flow task or by clicking the **Data Flow tab**.
2. In the **SSIS Toolbox**, expand **OtherSources**, and then drag a **Flat File Source** onto the design surface of the **Data Flow** tab.
3. On the **Data Flow** design surface, right-click the newly added **Flat File Source**, click **Rename**, and change the name to **Extract Sample Currency Data**.
4. Double-click the Flat File source to open the Flat File Source Editor dialog box.
5. In the **Flat file connection manager** box, select **Sample Flat File Source Data**.
6. Click **Columns** and verify that the names of the columns are correct.
7. Click **OK**.
8. Right-click the Flat File source and click **Properties**.
9. In the Properties window, verify that the **LocaleID** property is set to **English (United States)**.

# Lesson 1-6 - Adding and Configuring the Lookup Transformations

you will add and configure the following two Lookup transformation components to the package:

* One transformation to perform a lookup of values from the **CurrencyKey** column of the **DimCurrency** dimension table based on matching **CurrencyID** column values from the flat file.
* One transformation to perform a lookup of values from the **DateKey** column of the **DimDate**dimension table based on matching **CurrencyDate** column values from the flat file.

In both cases, the Lookup transformation will utilize the OLE DB connection manager that you previously created.

**To add and configure the Lookup Currency Key transformation**

1. In the **SSIS Toolbox**, expand **Common**, and then drag **Lookup** onto the design surface of the **Data Flow** tab. Place Lookup directly below the **Extract Sample Currency Data** source.
2. Click the **Extract Sample Currency Data** flat file source and drag the blue arrow onto the newly added **Lookup** transformation to connect the two components.
3. On the **Data Flow** design surface, click **Lookup** in the **Lookup** transformation, and change the name to **Lookup Currency Key**.
4. Double-click the **Lookup CurrencyKey** transformation to display the Lookup Transformation Editor.
5. On the **General** page, make the following selections:
   1. Select **Full cache**.
   2. In the **Connection type** area, select **OLE DB connection manager**.
6. On the **Connection** page, make the following selections:
   1. In the **OLE DB connection manager** dialog box, ensure that **localhost.AdventureWorksDW2012** is displayed.
   2. Select **Use results of an SQL query**, and then type or copy the following SQL statement:

SQLCopy

SELECT \* FROM [dbo].[DimCurrency]

WHERE [CurrencyAlternateKey]

IN ('ARS', 'AUD', 'BRL', 'CAD', 'CNY',

'DEM', 'EUR', 'FRF', 'GBP', 'JPY',

'MXN', 'SAR', 'USD', 'VEB')

1. On the **Columns** page, make the following selections:
   1. In the **Available Input Columns** panel, drag **CurrencyID** to the **Available Lookup Columns**panel and drop it on **CurrencyAlternateKey**.
   2. In the **Available Lookup Columns** list, select the check box to the left of **CurrencyKey**.
2. Click **OK** to return to the **Data Flow** design surface.
3. Right-click the Lookup Currency Key transformation, click **Properties**.
4. In the Properties window, verify that the **LocaleID** property is set to **English (United States)**and the **DefaultCodePage** property is set to **1252**.

**To add and configure the Lookup DateKey transformation**

1. In the **SSIS Toolbox**, drag **Lookup** onto the **Data Flow** design surface. Place Lookup directly below the **Lookup Currency Key** transformation.
2. Click the **Lookup Currency Key** transformation and drag the green arrow onto the newly added **Lookup** transformation to connect the two components.
3. In the **Input Output Selection** dialog box, click **Lookup Match Output** in the **Output** list box, and then click **OK**.
4. On the **Data Flow** design surface, click **Lookup** in the newly added **Lookup** transformation, and change the name to **Lookup Date Key**.
5. Double-click the **Lookup Date Key** transformation.
6. On the **General** page, select **Partial cache**.
7. On the **Connection** page, make the following selections:
   1. In the **OLEDB connection manager** dialog box, ensure that **localhost.AdventureWorksDW2012** is displayed.
   2. In the **Use a table or view** box, type or select **[dbo].[DimDate]**.
8. On the **Columns** page, make the following selections:
   1. In the **Available Input Columns** panel, drag **CurrencyDate** to the **Available Lookup Columns** panel and drop it on **FullDateAlternateKey**.
   2. In the **Available Lookup Columns** list, select the check box to the left of **DateKey**.
9. On the **Advanced** page, review the caching options.
10. Click **OK** to return to the **Data Flow** design surface.
11. Right-click the Lookup Date Key transformation and click **Properties.**
12. In the Properties window, verify that the **LocaleID** property is set to **English (United States)**and the **DefaultCodePage** property is set to **1252**.

# Lesson 1-7 - Adding and Configuring the OLE DB Destination

**To add and configure the sample OLE DB destination**

1. In the **SSIS Toolbox**, expand **Other Destinations**, and drag **OLE DB Destination** onto the design surface of the **Data Flow** tab. Place the OLE DB destination directly below the **Lookup Date Key** transformation.
2. Click the **Lookup Date Key** transformation and drag the green arrow over to the newly added **OLE DB Destination** to connect the two components together.
3. In the **Input Output Selection** dialog box, in the **Output** list box, click **Lookup Match Output**, and then click **OK**.
4. On the **Data Flow** design surface, click **OLE DB Destination** in the newly added **OLE DB Destination** component, and change the name to **Sample OLE DB Destination**.
5. Double-click **Sample OLE DB Destination**.
6. In the **OLE DB Destination Editor** dialog box, ensure that **localhost.AdventureWorksDW2012**is selected in the **OLE DB Connection manager** box.
7. In the **Name of the table or the view** box, type or select **[dbo].[FactCurrencyRate]**.
8. Click the **New** button to create a new table. Change the name of the table in the script to read **NewFactCurrencyRate**. Click **OK**.
9. Upon clicking **OK**, the dialog will close and the **Name of the table or the view** will automatically change to **NewFactCurrencyRate**.
10. Click **Mappings**.
11. Verify that the **AverageRate**, **CurrencyKey**, **EndOfDayRate**, and **DateKey** input columns are mapped correctly to the destination columns. If same-named columns are mapped, the mapping is correct.
12. Click **OK**.
13. Right-click the **Sample OLE DB Destination** destination and click **Properties**.
14. In the Properties window, verify that the **LocaleID** property is set to **English (United States)**and the**DefaultCodePage** property is set to **1252**.