



# Fast-track SSIS

Sample manual - first two chapters



## TABLE OF CONTENTS (1 of 7)

<b>1</b>	<b>INTRODUCING SSIS</b>	<b>Page</b>
1.1	Overview of SSIS	9
	<i>SSIS Packages and Solution Explorer</i>	9
	<i>Control Flow</i>	9
	<i>Data Flow</i>	10

<b>2</b>	<b>USING VISUAL STUDIO</b>	<b>Page</b>
2.1	Starting Visual Studio	11
	<i>Visual Studio and SQL Server Data Tools</i>	11
	<i>Running Visual Studio</i>	11
2.2	Creating a Project	12
2.3	Visual Studio Windows	13
	<i>The Solution Explorer, Properties and SSIS Toolbox Windows</i>	13
	<i>Floating Windows</i>	14
	<i>Auto-hiding Windows</i>	14

<b>3</b>	<b>PACKAGES</b>	<b>Page</b>
3.1	Working with Packages	15
	<i>Creating a Package</i>	15
	<i>Renaming Packages</i>	15
	<i>Opening or Editing Packages</i>	16
	<i>Viewing a Package's XML</i>	16
	<i>Saving a Package</i>	17
	<i>Closing Individual Packages</i>	17
	<i>Closing Multiple Packages</i>	17
	<i>Copying Packages</i>	17
3.2	The Package Tabs	18
3.3	Running a Package	19
	<i>Executing a Single Package</i>	19
	<i>Making a Package the Start-up Object and Debugging a Project</i>	19
	<i>Monitoring Package Execution</i>	20
	<i>Using the Progress Window</i>	20
3.4	Stopping Running a Package	21

<b>4</b>	<b>TASKS</b>	<b>Page</b>
4.1	Working with Tasks	22
	<i>Adding a Task</i>	22
	<i>Renaming a Task</i>	22
	<i>Editing a Task</i>	23
	<i>Executing an Individual Task</i>	23
	<i>Deleting and Suspending (Disabling) Tasks</i>	24
4.2	Working with Task Diagrams	25
	<i>Selecting Tasks</i>	25
	<i>Moving Tasks</i>	25
	<i>Auto-layout of Diagrams</i>	26
	<i>Zooming In and Out</i>	26
4.3	Annotations	27
4.4	Grouping Tasks	28
	<i>Grouping Tasks</i>	28
	<i>Grouping using Sequence Containers</i>	29

<b>5</b>	<b>DATA FLOW</b>	<b>Page</b>
5.1	Introducing Data Flow	30
	<i>Sources, Transforms and Destinations This and Subsequent Chapters</i>	30
5.2	Data Flow Tasks	31
	<i>Creating Data Flow Tasks</i>	31
	<i>Switching to Data Flow Tasks</i>	31
5.3	Connections	32
	<i>Types of Connection</i>	32
	<i>Project-Level and Package-Level Connections</i>	32
	<i>Creating Connections in Advance</i>	33
	<i>Creating Connections as Part of Sources/Destinations</i>	33
5.4	Creating Our Package	34
5.5	The Advanced Editor	35

## TABLE OF CONTENTS (2 of 7)

<b>6</b>	<b>WORKING WITH SQL SERVER</b>	<b>Page</b>
6.1	Creating a SQL Server Connection	36
6.2	Creating a SQL Server Source/Destination	37
	<i>Step 1 – Creating the SQL Server Source/Destination</i>	37
	<i>Step 2 – Choosing the Table or Query</i>	38
	<i>Step 3 – Mapping Columns</i>	38
6.3	Creating SQL Server Tables	39
6.4	Execute SQL Tasks	40
	<i>Step 1 - Adding an Execute SQL Task</i>	40
	<i>Step 2 – Assign a Connection</i>	40
	<i>Step 3 – Enter a SQL Statement</i>	41
	<i>Step 4 – Optionally, Parse your SQL Command</i>	41

<b>7</b>	<b>WORKING WITH FLAT FILES</b>	<b>Page</b>
7.1	Examples for this Chapter	42
7.2	Existing Flat File - Source	43
	<i>Connecting to a Flat File</i>	43
	<i>Configuring Rows and Columns</i>	44
	<i>Creating the Flat File Source</i>	44
7.3	Existing Flat File – Destination	45
7.4	Creating New Flat Files	45
	<i>Step 1 – Creating a Destination</i>	46
	<i>Step 2 – Starting a New Connection</i>	46
	<i>Step 3 – Choose a File Format</i>	47
	<i>Step 4 – Browsing to a Folder</i>	47
	<i>Step 5 – Choosing a File Name</i>	48
	<i>Step 6 – Configuring the New File (Rows)</i>	48
	<i>Step 7 – Configure the New File (Columns)</i>	49
	<i>Step 8 – Configuring Mappings</i>	49
	<i>Step 9 – Running your Package</i>	49
7.5	Using Multiple Flat Files	50
	<i>Our Example</i>	50
	<i>Adding the MultiFlatFile Connector</i>	51
	<i>Using a MultiFlatFile Connector</i>	52

<b>8</b>	<b>WORKING WITH EXCEL</b>	<b>Page</b>
8.1	Excel Workbooks – Overview	53
	<i>The Example Used in this Chapter</i>	53
	<i>What Can Go Wrong</i>	53
8.2	Importing/Exporting using Excel	54
	<i>Step 1 – Creating an Excel Connection</i>	54
	<i>Step 2 – Creating a Source or Destination</i>	55
	<i>Step 3 – Mapping Columns</i>	55
8.3	The 32-Bit Problem	56
	<i>The Problem</i>	56
	<i>The Solution</i>	57

<b>9</b>	<b>DATA VIEWERS</b>	<b>Page</b>
9.1	Using Data Viewers	58
9.2	Creating Data Viewers	59
	<i>The Data Viewer Buffer</i>	59

<b>10</b>	<b>TRANSFORMS</b>	<b>Page</b>
10.1	Adding Transforms	60
10.2	Sorting / Getting Unique Rows	61
10.3	Sampling	62
	<i>The Sampling Transforms</i>	62
	<i>Incorporating Sampling Transforms into Data Flow</i>	63
10.4	Combining and Splitting Data	64
	<i>Introduction to the Transforms</i>	64
	<i>An Example using Multicast and Union All Transforms</i>	65
	<i>Configuring Union All Transforms</i>	65
10.5	Aggregating Data	66

<b>11</b>	<b>DATA TYPES</b>	<b>Page</b>
11.1	The Data Types in SSIS	67
	<i>The Three Data Type Families in SSIS</i>	67
	<i>Data Types Listed in this Chapter</i>	67
11.2	SSIS Data Types	68
	<i>String Data Types</i>	68
	<i>Integer Data Types</i>	68
	<i>Boolean or Logical Data Types</i>	68
	<i>Non-Integer Data Types</i>	69
	<i>Date/time Data Types</i>	69
	<i>Time Data Types</i>	69

## TABLE OF CONTENTS (3 of 7)

12	DATA CONVERSION TRANSFORMS	Page
12.1	Overview of Data Conversion	70
12.2	Creating Data Conversion Transforms	71
	<i>Our Example: Importing Excel Unicode Data to Varchar Columns</i>	71
	<i>Adding a Data Conversion Transform</i>	71
	<i>Configuring a Data Conversion Transform</i>	72
	<i>Mapping Data Conversion Transform Columns</i>	73
	<i>The Complete Package</i>	73

13	VARIABLES	Page
13.1	Overview of Variables	74
	<i>bles</i>	74
13.2	Working with Variables	75
	<i>The Variables Window</i>	75
	<i>Creating a Variable</i>	75
	<i>Deleting Variables</i>	75
	<i>Changing the Scope of Variables</i>	76
	<i>Choosing What to See</i>	76
13.3	User and System Variables	77

14	USING VARIABLES	Page
14.1	Example for this Chapter	78
	<i>Creating the Variables Needed</i>	78
14.2	Creating Row Count Transforms	79
	<i>Adding a Row Count Transform</i>	79
14.3	Creating Expression Tasks	80
14.4	Debugging	81
	<i>Setting and Removing Breakpoints</i>	81
	<i>Debugging using Breakpoints</i>	82

15	DERIVED COLUMN TRANSFORMS	Page
15.1	Overview of Derived Column Transforms	83
	<i>Our Example – Calculating Film Length Statistics</i>	83
	<i>Adding a Derived Column Transform</i>	83
15.2	Simple Column Expressions	84
15.3	Concatenating, Casting and Conditions	84
	<i>Simple Concatenation Won't Work</i>	85
	<i>Casting</i>	86
	<i>The Ternary or Conditional Operator</i>	86
	<i>The Double = Sign When Testing Conditions</i>	88
	<i>The Relational Operators Allowed</i>	88
15.4	More Examples and Syntax	89
	<i>Example of a Mathematical Function</i>	89
	<i>Dealing with Nulls</i>	90
	<i>Example of a String Function</i>	90

16	SSIS FUNCTIONS	Page
16.1	Working with Strings of Text	91
	<i>Functions to Find and Replace Text</i>	91
	<i>Getting Extracts from a String of Text</i>	91
	<i>Other Text Functions</i>	92
	<i>New Line and Other Special Characters</i>	92
16.2	Date Functions	93
	<i>The DATEPART Function</i>	93
	<i>Using DATEADD to Add Dates</i>	94
	<i>Using DATEDIFF to Take the Difference between Two Dates</i>	94
16.3	Casting Data	95

17	ERROR-HANDLING	Page
17.1	Our Example	96
17.2	Configuring Error-Handling	97
17.3	Tracking Error Rows	98

18	CONDITIONAL SPLITS	Page
18.1	Our Example	99
18.2	Creating a Conditional Split	100
	<i>Beginning a Conditional Split</i>	100
	<i>The Operators Allowed</i>	101
	<i>Typing Conditions</i>	101
	<i>Spotting Errors (Red Text)</i>	101
18.3	Directing Conditional Split Output	102

## TABLE OF CONTENTS (4 of 7)

<b>19</b>	<b>LOOKUP TRANSFORMS</b>	<b>Page</b>
19.1	Our Example	103
19.2	Creating a Lookup Transform	104
	<i>Step 1 – Creating the Transform</i>	104
	<i>Step 2 – Choose the Lookup Table</i>	104
	<i>Step 3 – Redirect Non-Matching Rows</i>	105
	<i>Step 4 – Choosing a Cache Mode</i>	106
	<i>Step 5 – Choosing Columns to Output</i>	107
	<i>Step 6 – Choosing Columns to Capture</i>	107
19.3	Dealing with Matched Rows	108
19.4	Dealing with Unmatched Rows	109
	<i>Strategy 1 – Store Unmatched Rows for Inspection</i>	109
	<i>Strategy 2 – Add Unmatched Shop Names as Unknown</i>	110
	<i>Strategy 3 – Add Unmatched Rows to the Lookup Table</i>	111
19.5	Caching Lookup Tables	112
	<i>Benefits of using Cache Transforms</i>	112
	<i>Our Example – Looking Up Property Types</i>	112
	<i>Filling the Cache</i>	113
	<i>Using the Cache</i>	114

<b>20</b>	<b>FILE SYSTEM TASKS</b>	<b>Page</b>
20.1	Adding File System Tasks	115
20.2	Working with Files	116
	<i>Creating a File Connection</i>	116
20.3	Working with Folders	117
	<i>Creating a Folder Connection</i>	117
	<i>Case Study: Creating a Folder</i>	118

<b>21</b>	<b>PRECEDENCE CONSTRAINTS</b>	<b>Page</b>
21.1	Simple Constraints	119
21.2	Expression Constraints	120
	<i>Automatically Annotating Expression Constraints</i>	121
21.3	Combining Constraints	122

<b>22</b>	<b>LOOPING OVER FILES</b>	<b>Page</b>
22.1	Our Example	123
22.2	Setting up the Package	124
	<i>Emptying the SQL Server Table</i>	124
	<i>Creating a String Variable to Hold Each File</i>	124
22.3	Looping Over Files	125
	<i>Step 1 – Adding the Foreach Loop Container Task</i>	125
	<i>Step 2 – Configuring the Loop Task</i>	126
	<i>Step 3 – Capturing the File Path in a Variable</i>	127
	<i>Step 4 – Checking the Package Works</i>	127
22.4	Omitting Certain Files	128
22.5	The Data Import Task	129
	<i>The Steps to Follow</i>	129
	<i>Step 1 – Create an Excel Connection</i>	130
	<i>Step 2 – Make this Excel Connection Dynamic</i>	131
	<i>Step 3 – Initialise the File Path Variable Value</i>	132
	<i>Step 4 – Configure and Finish the Data Flow Task</i>	132
22.6	Running the Package	133

## TABLE OF CONTENTS (5 of 7)

23	LOOPING OVER ADO ROWS	Page
23.1	How ADO Enumerators Work	134
23.2	Generating Multiple Files from a Single Table	135
	<i>The Details of this Example</i>	135
	<i>Summary of the Steps to Follow</i>	135
	<i>Step 1 – Creating the Variables</i>	136
	<i>Step 2 – Create an Execute SQL Task to get Unique Shop Names</i>	136
	<i>Step 3 – Store the Results Set in the Object Variable</i>	137
	<i>Step 4 – Create the ADO Enumerator Loop</i>	137
	<i>Step 5 – Configure the ADO Enumerator Loop</i>	138
	<i>Step 6 – Add a Fixed Data Flow Task</i>	139
	<i>Step 7 – Make the Flat File Connection Dynamic</i>	140
	<i>Step 8 – Run your Package</i>	140
23.3	Importing a Workbook's Worksheets	141
	<i>Creating the Variable</i>	142
	<i>Create the Outline Package</i>	142
	<i>The Foreach Loop Type – Creating a Connection</i>	143
	<i>Choosing to Return the Workbook's Worksheet Names</i>	144
	<i>Creating the Data Flow Task (Source)</i>	144
	<i>Adding a Data Flow Destination</i>	145
	<i>Running the Package</i>	145

24	SQL PARAMETERS	Page
24.1	Passing Parameters by Position	146
	<i>Our Example</i>	146
	<i>Creating the Variables</i>	146
	<i>Creating the Loop over Files</i>	147
	<i>Creating the Execute SQL Task using Parameters</i>	148
	<i>The Parameter Mapping Choices</i>	149
	<i>Running the Package</i>	149
24.2	Passing Parameters by Name	150
	<i>Creating the Stored Procedure</i>	150
	<i>Changing the Execute SQL Task</i>	150

25	MERGE JOINS	Page
25.1	What Merge Joins Do	151
25.2	How Merge Joins Work	152
25.3	Creating the Package	153
	<i>Step 1 – Loading the Data</i>	153
	<i>Step 2 – Sorting the Data</i>	153
	<i>Step 3 – Merging the Data</i>	154
	<i>Step 4 – Picking out the Differences</i>	155
	<i>Step 5 – Finishing the Package</i>	155

26	DEPLOYMENT	Page
26.1	Overview of Deployment	156
26.2	Preparing to Deploy	157
	<i>Creating a Catalog</i>	157
	<i>Creating Folders</i>	157
26.3	Deploying Projects and Packages	158
	<i>Starting Project Deployment</i>	158
	<i>Finishing Deployment</i>	159
	<i>Deploying Individual Packages</i>	159

27	EXECUTING DEPLOYED PACKAGES	Page
27.1	Executing Reports from the SSMS Menu	160
27.2	Viewing Reports	161
	<i>Viewing Package Reports</i>	161
	<i>Viewing Catalog Reports</i>	161
27.3	Executing Packages from SQL	162
27.4	Scheduling Packages	163
	<i>Step 1 - Run SQL Server Agent</i>	163
	<i>Step 2 – Add a Job</i>	164
	<i>Step 3 – Adding Steps to the Job</i>	164
	<i>Step 4 – Choosing a Schedule</i>	165
	<i>Step 5 – Test your Job</i>	165

28	PARAMETERS	Page
28.1	Overview of Parameters	166
	<i>Our Example</i>	166
	<i>Variables and Parameters Needed</i>	166
28.2	Creating Parameters	167
	<i>Creating Project-Level Parameters</i>	167
	<i>Creating Package-Level Parameters</i>	167
28.3	Setting up the Example	168
	<i>The Basic Control Flow Tasks</i>	168
	<i>Starting to Configure the Loop over Files</i>	168
	<i>Making the Loop Folder Dynamic</i>	169
	<i>The Data Flow Task Excel Source</i>	169
	<i>Finishing the Data Flow Task</i>	170
	<i>Running the Package</i>	170
28.4	Deploying Packages using Parameters	171
	<i>Deploying Package-Level Parameters</i>	171
	<i>Deploying Project-Level Parameters</i>	171
28.5	Running Deployed Packages with Parameters	172

## TABLE OF CONTENTS (6 of 7)

<b>29</b>	<b>SCRIPT TASKS</b>	<b>Page</b>
29.1	Overview	173
29.2	Adding a Script Task	174
	<i>Step 1 – Creating the Script Task</i>	174
	<i>Step 2 – Choose a Language</i>	175
	<i>Step 3 – Choosing the Start Program Name</i>	175
	<i>Step 4 – Editing your Script</i>	175
	<i>Step 5 – Understanding (and Tidying Up) the Code Generated</i>	176
	<i>Step 6 – Writing your Program</i>	177
	<i>Step 7 – Correcting any Errors</i>	177
	<i>Step 8 – Saving and Running your Code</i>	178
29.3	Passing Variables to Script Tasks	179
	<i>Passing the Variables to your Script</i>	179
	<i>Referencing Variables in Script</i>	180
	<i>A Short-Cut for Experienced Programmers</i>	180
	<i>The Most Common Mistake with Variables</i>	181
29.4	Debugging Script	182
	<i>Setting and Removing Breakpoints</i>	182
	<i>Removing All Breakpoints</i>	182
	<i>Stepping through Code</i>	183
	<i>Displaying Variable Values</i>	184

<b>30</b>	<b>PROGRAMMING (VARIABLES)</b>	<b>Page</b>
30.1	Why Use Variables in Script?	185
30.2	Declaring Variables	185
	<i>Declaring Variables</i>	186
	<i>Creating Nullable Variables</i>	186
	<i>Using Modified Hungarian Notation</i>	186
	<i>Default Values for Variables</i>	187
	<i>Problems with Declaring Variables within Clauses</i>	187
30.3	Setting Values in Variables	187
	<i>Declaring Integer Variables and Adding/Subtracting</i>	188
	<i>Accumulating Text in String Variables</i>	188
30.4	Variable Data Types	188
	<i>Mapping C# Data Types to the CLR Runtime</i>	189
	<i>A Lazy Person's Data Types</i>	189
	<i>Logical Values</i>	190
	<i>Integers</i>	190
	<i>Decimal (Floating Point) Numbers</i>	190
	<i>Strings and Text</i>	190
	<i>Dates and Times</i>	191
	<i>Objects</i>	191
30.5	Converting Variables	191
	<i>Conversion Using Convert.To</i>	192
	<i>ToString() – Special Case for String Conversions</i>	192
	<i>Casting Data Types</i>	192
30.6	Notes on Working with Specific Data Types	193
	<i>Working with Characters</i>	193
	<i>Working with Strings</i>	193
	<i>Escape Characters</i>	194
	<i>Verbatim Strings</i>	194
	<i>Splitting Strings</i>	194
	<i>Formatting Dates</i>	195
	<i>Working with Numbers – Possible Operations</i>	195

<b>31</b>	<b>CONDITIONS</b>	<b>Page</b>
31.1	Using IF for Conditions	196
31.2	Operators	196
31.3	The SWITCH statement	197
	<i>The Syntax of SWITCH</i>	198
	<i>Limitations of SWITCH</i>	199
31.4	Ternary and Coalesce Operators	199
	<i>The Conditional Operator</i>	200
	<i>The Null Coalesce Operator</i>	200

## TABLE OF CONTENTS (7 of 7)

<b>32</b>	<b>LOOPS</b>	<b>Page</b>
32.1	Looping in C#	201
	<i>Looping a Given Number of Times</i>	201
	<i>Looping While a Condition is True (While/Do)</i>	201
32.2	Breaking Out of Loops	202

<b>33</b>	<b>SCRIPT COMPONENTS</b>	<b>Page</b>
33.1	Our Example	204
33.2	Starting the Package	205
	<i>Removing Old Rows</i>	205
	<i>Creating a Flat File Data Connection</i>	205
	<i>Creating a Data Flow to Import Data</i>	206
33.3	Creating your Script Component	207
	<i>Step 1 – Add the Component</i>	207
	<i>Step 2 – Set the Columns Feeding into the Script Component</i>	208
	<i>Step 3 – Set the Columns Coming out of the Script Component</i>	209
	<i>Step 4 – Starting to Create Script</i>	210
	<i>Step 5 – Writing the Script Itself</i>	211
33.4	Finishing the Package	212
	<i>The Conditional Split Task</i>	212
	<i>The Valid Data Destination</i>	213
	<i>The Invalid Data Destination</i>	213

<b>34</b>	<b>SCRIPTING FILES AND FOLDERS</b>	<b>Page</b>
34.1	Solutions Covered	214
34.2	Checking a Folder Exists	215
	<i>Step 1 – Create the Variables</i>	215
	<i>Step 2 – Create the Script Task</i>	215
	<i>Step 3 – Tidy your Code and Reference System.IO</i>	216
	<i>Step 4 – Get a Reference to the Folder</i>	216
	<i>Step 5 – Creating the Expression Constraints</i>	217
34.3	Checking a File Exists	218
	<i>Step 1 – Create Two Versions of the Source Workbook</i>	218
	<i>Step 2 – Create the Variables that you will Need</i>	219
	<i>Step 3 – Create a Static Package</i>	219
	<i>Step 4 – Create a Script Task</i>	220
	<i>Step 5 – Write your Script</i>	220
	<i>Step 6 – Create Two Expression Constraints</i>	221

<b>35</b>	<b>SENDING MAIL</b>	<b>Page</b>
35.1	The Send Mail Task	222
35.2	Sending Mail in Script	223
	<i>Creating and Passing Variable Values</i>	223
	<i>Referencing the Mail Namespace</i>	224
	<i>Writing your Script</i>	224

<b>36</b>	<b>EVENTS</b>	<b>Page</b>
36.1	Overview of Events	225
	<i>Seeing Events on the Progress Tab</i>	225
	<i>Our Example</i>	225
36.2	The List of Events	226
36.3	Using Event-Handlers	227
	<i>Creating (and Deleting) Event-Handlers</i>	227
	<i>The Event-Handlers for our Example</i>	228

<b>37</b>	<b>AUDIT TRANSFORMS</b>	<b>Page</b>
37.1	Using Audit Transforms	229
	<i>What Audit Transforms Do</i>	229
	<i>Adding Audit Transforms</i>	229
	<i>Choosing Columns to Include</i>	230
	<i>Running an Audit Transform</i>	230

<b>38</b>	<b>LOGGING</b>	<b>Page</b>
38.1	Overview of Logging	231
	<i>Showing the SYSSISLOG Table in SQL Server</i>	231
	<i>A Typical Text Log</i>	231
38.2	Starting to Log	232
	<i>Beginning a Log</i>	232
	<i>Choosing a Logging Provider</i>	232
	<i>Specific Notes on Creating a SQL Server Log</i>	233
	<i>Specific Notes on Creating a Text File Log</i>	233
38.3	Configuring your Log	234
	<i>Choosing which Tasks to Log</i>	234
	<i>Choosing which Events to Log</i>	234
38.4	Catalog Logging	235



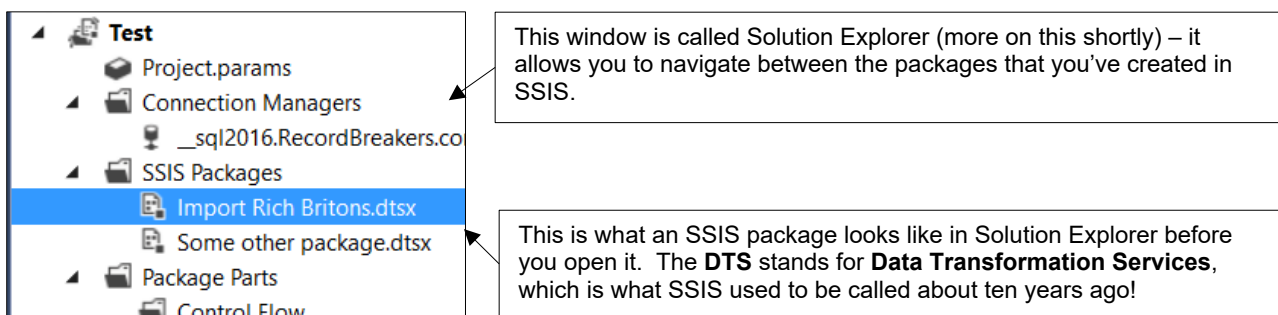
## CHAPTER 1 - INTRODUCING SSIS

### 1.1 Overview of SSIS

SQL Server Integration Services (SSIS) is a program which allows you to build packages to Extract, Transform and Load data (it's often called an ETL application for this reason).

#### SSIS Packages and Solution Explorer

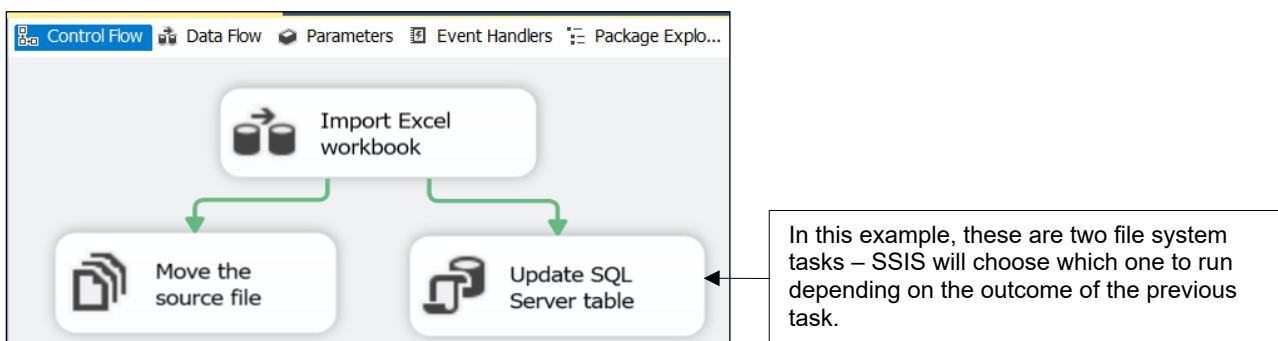
A package in SSIS is a file with the extension **.dtsx**:



A package consists of two main parts: *control flow* and *data flow*, as shown under separate headings below and overleaf.

#### Control Flow

The *control flow* part of a package consists of a series of instructions you ask SSIS to execute:



*When you tell your child to tidy their room, do their homework and then come and help make dinner, this is a series of control flow tasks (as well as being a tad optimistic).*

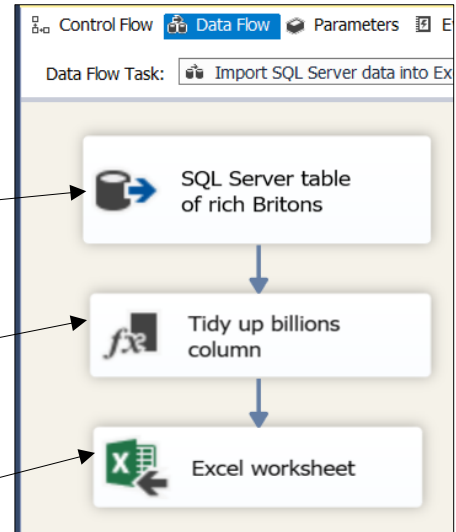
## Data Flow

You can open any single data flow task to reveal instructions on how to load, transform and store data:

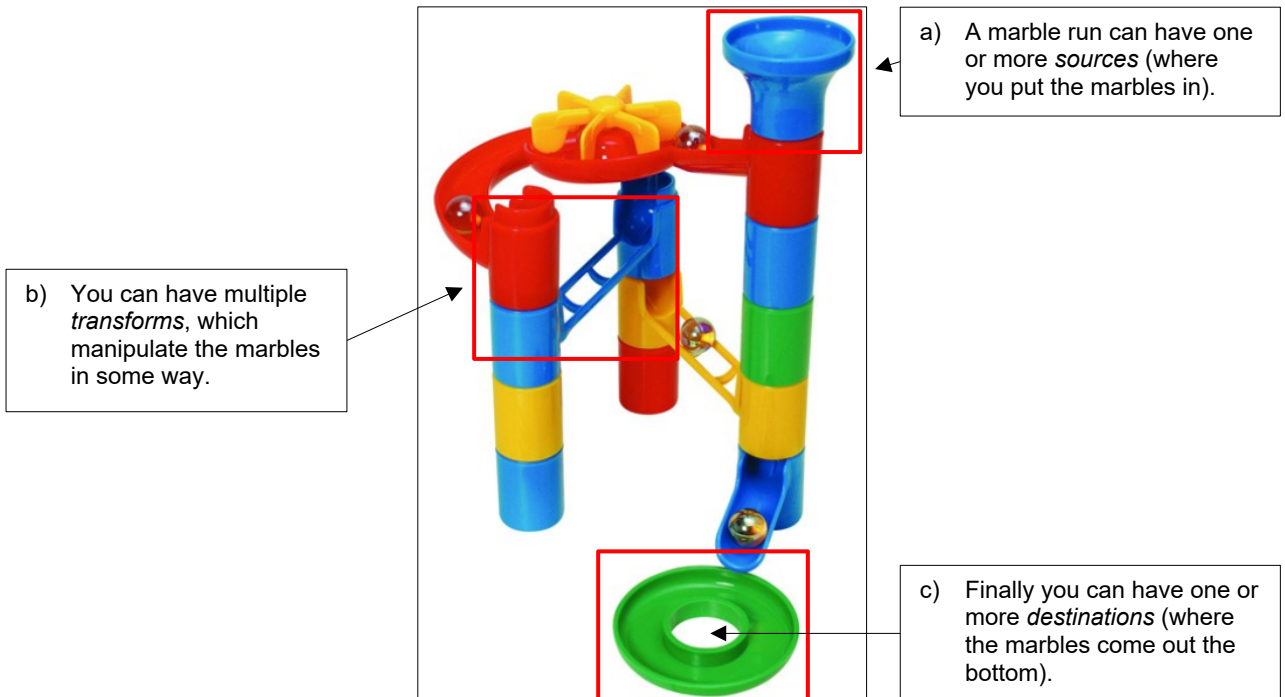
This is a *source* (where the data comes from – in this case a SQL Server table).

This is a *transform* (in this case, it's tidying up a numerical column).

This is a *destination* (where the data ends up – in this case in an Excel workbook).



A good analogy for data flow is a marble run:



*At a rough guess, you typically will spend about 70% of your time in Integration Services inside data flow tasks, and 30% in control flow.*

## CHAPTER 2 - USING VISUAL STUDIO

### 2.1 Starting Visual Studio

#### Visual Studio and SQL Server Data Tools

Visual Studio is Microsoft's flagship development application – you can use it to create websites, mobile phone apps, SSIS packages and Windows applications, among other things:

The part of Visual Studio which you use to create BI tools (in SSIS, Reporting Services and Analysis Services) also goes under the name *SQL Server Data Tools*, or *SSDT*.

*Business Intelligence tools  
(including SSIS)*

*ASP.NET  
(websites)*

*Windows applications  
(Visual Basic and C#)*

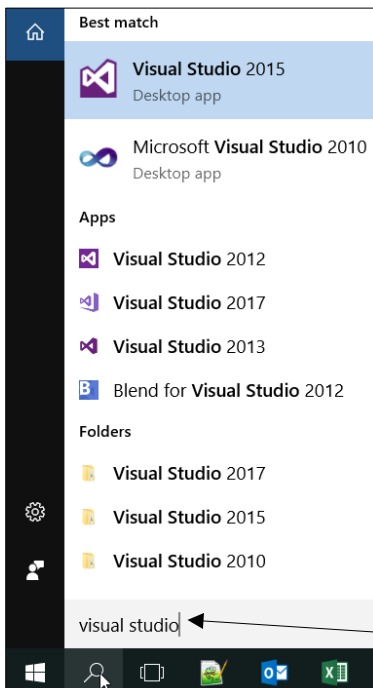
*Mobile phone apps  
(Apple and Android)*



*What the above shows is that it doesn't matter whether you run Visual Studio or SQL Server Data Tools – the second is just a subset of the first.*

#### Running Visual Studio

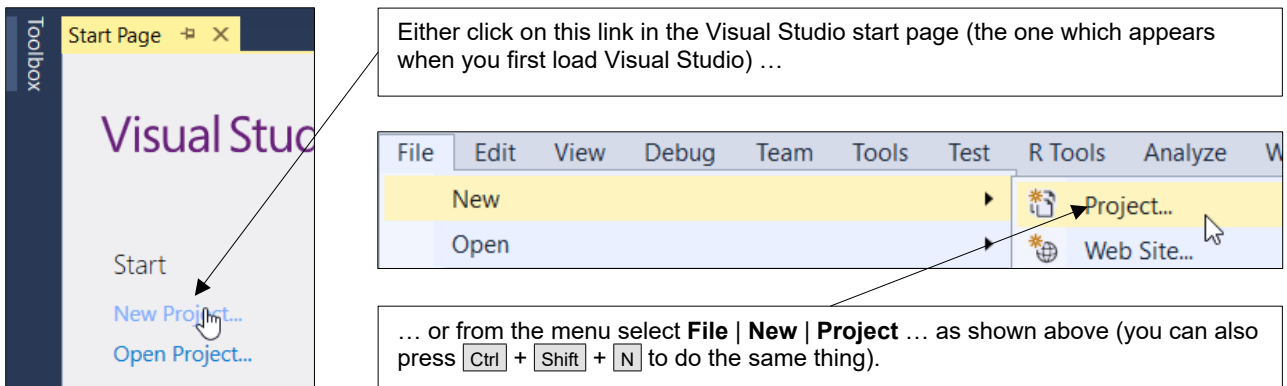
You can start Visual Studio in many ways – one method for Windows 10 is shown here:



Click on the magnifying glass icon, then type in the program you want to run. This Wise Owl has got four different versions of Visual Studio on his computer! The one used in writing this manual was Visual Studio 2015.

## 2.2 Creating a Project

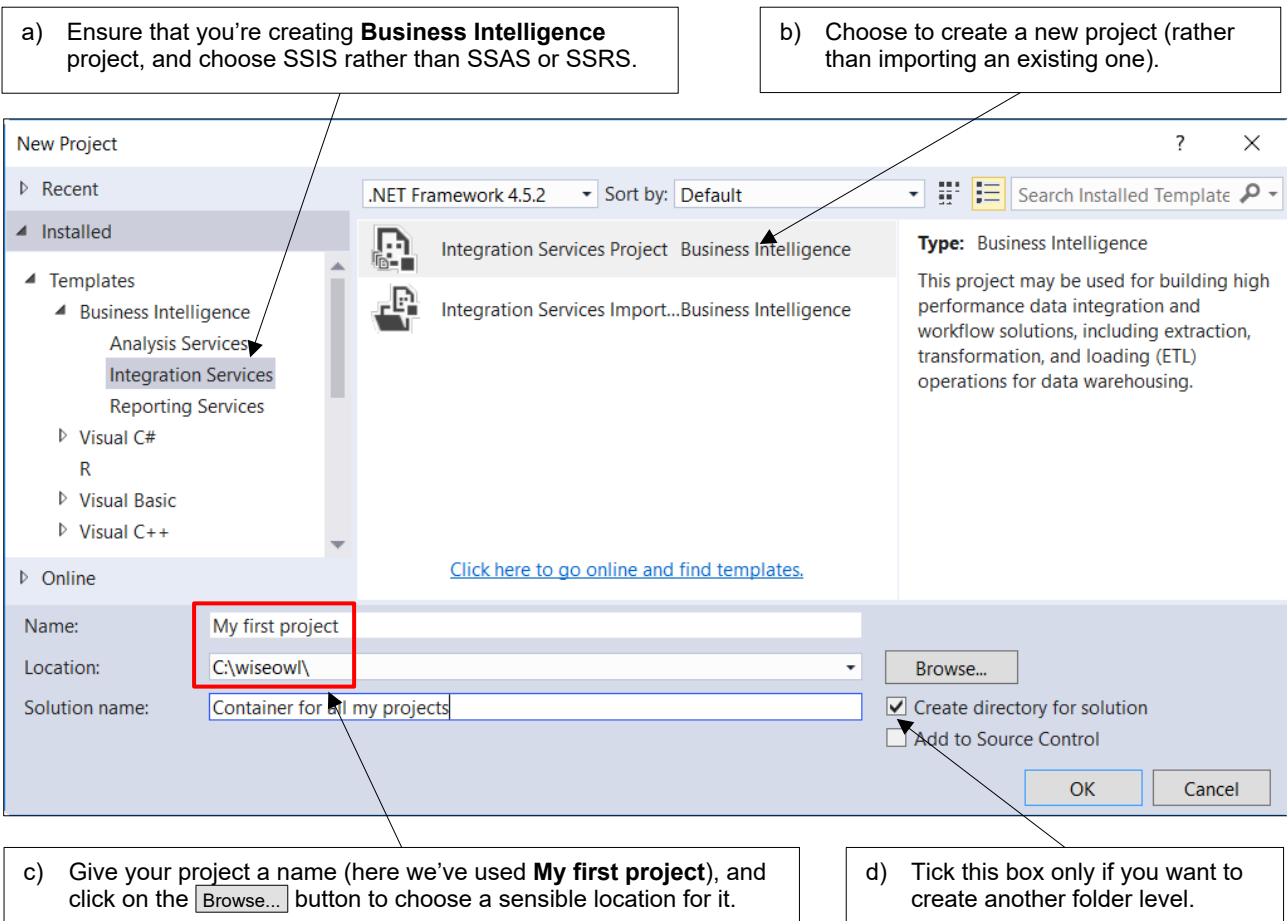
You store SSIS packages in a container called a *project*. Here's how to create one:



Either click on this link in the Visual Studio start page (the one which appears when you first load Visual Studio) ...

... or from the menu select **File | New | Project ...** as shown above (you can also press **Ctrl + Shift + N** to do the same thing).

You can now give your project a name, and choose where to put it:



a) Ensure that you're creating **Business Intelligence** project, and choose SSIS rather than SSAS or SSRS.

b) Choose to create a new project (rather than importing an existing one).

c) Give your project a name (here we've used **My first project**), and click on the **Browse...** button to choose a sensible location for it.

d) Tick this box only if you want to create another folder level.

If you choose to create a directory for your solution as above, you'll end up with a long path!



This PC > OS (C:) > wiseowl > 
 Container for all my projects > 
 My first project

*Base location*
*Folder for solution*
*Project folder*

## 2.3 Visual Studio Windows

### The Solution Explorer, Properties and SSIS Toolbox Windows


When using SSIS, there are three main windows that you will use:

Use *Solution Explorer* to add, view and edit packages and other files.

Use the *SSIS Toolbox* to add control or data flow tasks.

Use the *Properties* window to set selected items' behaviour.

Here's how to display these three windows:

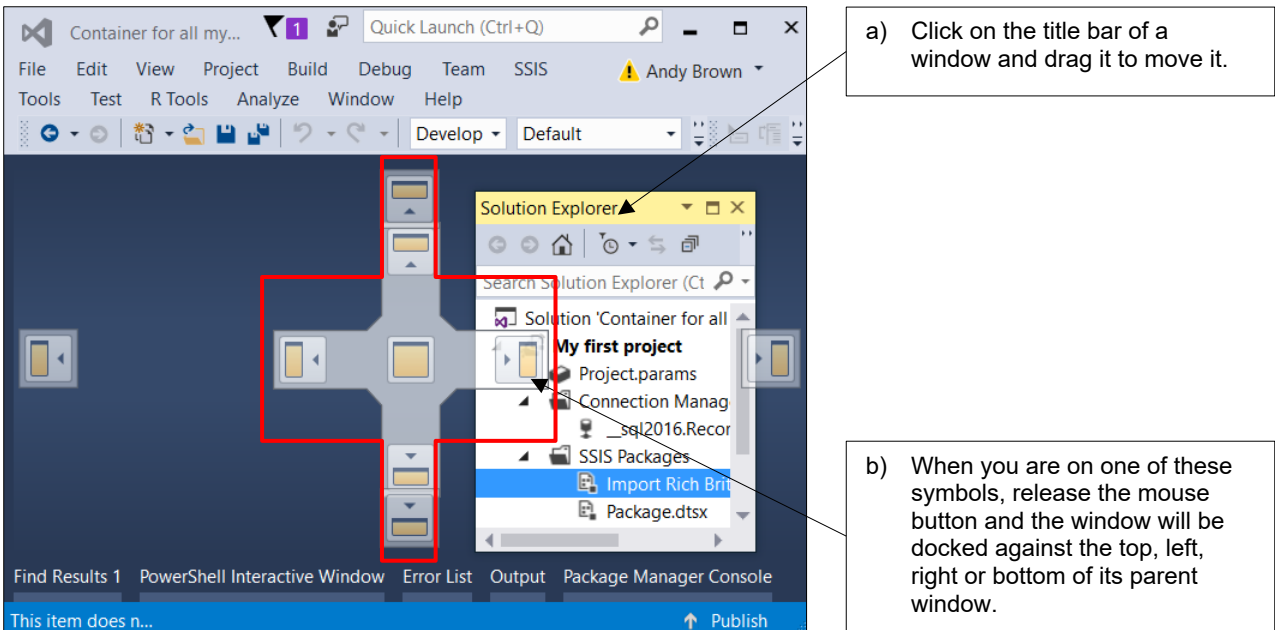
Window	Top menu	Keyboard	Other method
<i>Solution Explorer</i>	<b>View → Solution Explorer</b>	Ctrl + Alt + L	
<i>SSIS Toolbox</i>	<b>SSIS → SSIS Toolbox</b>		
<i>Properties</i>	<b>View → Properties Windows</b>	F4	



To get the SSIS toolbox menu to appear you often have to open a package and click on it first. Don't confuse the (invaluable) SSIS toolbox with the (useless, in this context) standard Visual Studio toolbox.

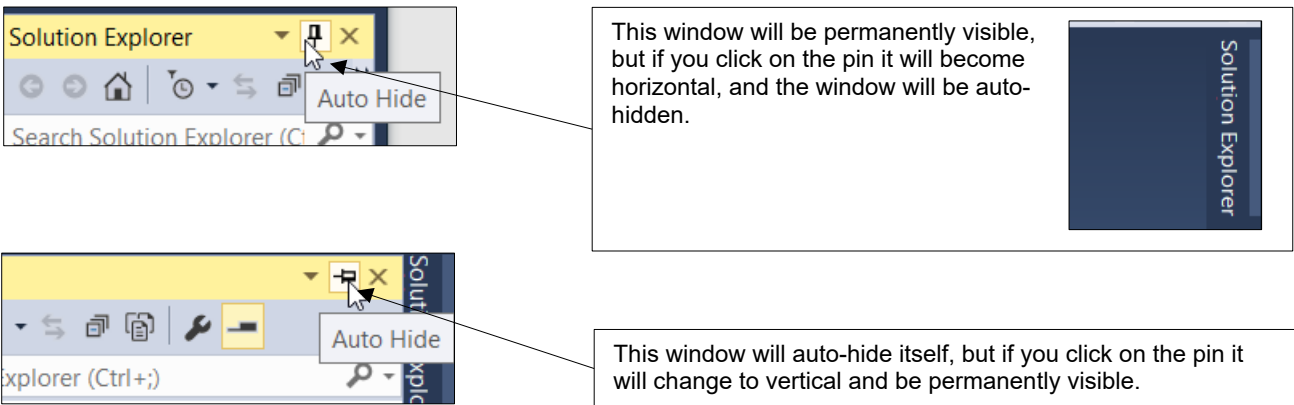
## Floating Windows

You can click and drag on the title bar of any window to reposition it:










































## Auto-hiding Windows

You can click on the pin at the top right of any docked window to make it shrink when you're not using it:



## What we do!

		Basic training	Advanced training	Systems / consultancy
<b>Office</b>	Microsoft Excel			
	VBA macros			
	Office Scripts			
	Microsoft Access			
<b>Business Intelligence</b>	Power BI			
	Power Apps			
	Power Automate / PAD			
<b>SQL Server</b>	SQL			
	Reporting Services			
	Report Builder			
	Integration Services			
	Analysis Services			
<b>Coding</b>	Visual C# programming			
	VB programming			
	DAX			
	Python			



**WiseOwl**  
Training

