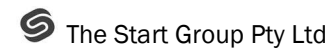


KHRONOS EXCEL PUMP

Examples

V1.0





© Copyright 2017. The Start Group Pty Ltd. All rights reserved.

Throughout this document, wherever reference is made to "The Start Group", it refers to The Start Group Pty Ltd and all of its subsidiaries, both partially or fully owned. All rights claimed on behalf of The Start Group Pty Ltd apply to it and all of its subsidiaries. Start Services Pty Ltd is a fully owned subsidiary of The Start Group Pty Ltd.

Contents

Khronos Excel Pump Examples.....	4
Simple Import.....	6
Import with a common reference cell	7
Import with trimming of text in Excel cells	8
Data Pivot	9
Data Pivot with common reference cells	10
Pivot Block with a multiple header rows	12
Wide Pivot	14
Group and Pivot	16
Recursive Data Block.....	18
Value Substitution	20
Value Lookup from SQL.....	21
Combo Pivot.....	22
Column Filtering in a Pivot Block.....	23
Identity Columns	26
Calling a Stored Procedure.....	28

Khronos Excel Pump Examples

Khronos Excel Pump is a configurable Windows® Service which provides a simple, yet powerful mechanism for moving data from Excel spreadsheets into Microsoft® SQL Servers. This includes pushing data from on-premise spreadsheets to cloud-hosted SQL Servers such as Azure SQL dB.

This document contains a variety of examples for many different scenarios which can be used as reference guides for your own situations. Each example includes the source Excel data, the resultant SQL table schema, and the Excel Pump configuration used to perform the import process.

The table below lists all of the examples, and describes the scenario which they cover. When working on real examples, many of the techniques shown in these examples are combined. For example, you might have an Excel worksheet which requires text Trimming, and Substitution, and Pivoting.

Example	Title	Description
1	Simple Import	Moves a table of data into SQL, with one SQL field for each Excel Column. Data is mapped one-to-one from the Excel table into the SQL table. This is typically used when your Excel data is already formatted very similarly to how you want it stored in SQL.
2	Using a common reference cell for multiple import rows	Moves a table of data into SQL, similarly to the Simple Import, but with one referenced value applied commonly to every SQL row which is inserted. This is typically used when header data from the worksheet is to be applied to all import data.
3	Trimming characters from Excel text values	Removes specific character patterns from the Excel values prior to assessing Row Rules, and again when pushing the values to SQL. This is used to remove unwanted characters from your Excel sheet, such as commas, spaces, semi-colons, or prefixes and suffixes.
4	Data Pivot - Narrow (wide Excel row to narrow SQL table)	Transforms rows of Excel data into columns of SQL data. This is frequently used to transform wide rows of Excel data into multiple, narrow rows of SQL data. This aligns with best-practice when 'normalising' data in SQL for long, narrow tables instead of wide, sparsely populated tables.
5	Data Pivot with multiple reference cells for each imported row	Performs a Data Pivot per above, but includes additional columns which are not part of the pivot operation. This is useful when one section of your Excel block needs to be pivoted from wide rows to narrow rows, but the rest of the data is in line with the pivoted row of Excel data and needs to be associated with it.
6	Data Pivot with multiple header rows	Performs a Data Pivot per above, but pivots multiple Excel header rows into multiple SQL columns beside the pivoted data values.

7	Data Pivot - Wide (wide Excel row to wide SQL table with multiple columns)	This example performs a data pivot, but instead of putting all pivoted data values into a single SQL field, it uses a separate SQL field for each pivoted item. The Excel header row is still transformed into separate SQL values beside each pivoted item. This approach is not commonly used as it results in wide, sparsely populated tables in SQL, which is not best practice.
8	Groups and Subgroups	When the Excel data has a group of values with a header cell (such as an area followed by detailed rows for the area), you may want the header cell's value to be included in every row sent to the SQL table. Group values are identified by the first or last value in a vertical group of Excel cells which contain a value. Group values rely on the Excel worksheet having blank cells between vertical groups to allow delineation of the group header/footer values.
9	Recursive operations	Searches for multiple data blocks in the worksheet by using reference text to identify the start and end of each data block. Each block is processed individually, and the data built up into a single bulk import operation to the SQL table.
10	Value Substitution from predefined lists	Replaces Excel values with substitution values from a pre-defined list in your pump configuration. This is useful when 'normalising' your data to use identity values or enumeration values instead of text values in SQL and assists in adhering to best-practice database design standards.
11	Value Lookup from SQL master data tables	Replaces Excel values with lookup values sourced from a SQL table or view. This is useful when replacing text values in Excel with identity values in SQL and assists in adhering to best-practice database design standards.
12	Combo Pivot	This example normalises a wide Excel data block into a narrow SQL table, replaces product descriptions with product names using Value Substitution, and then replaces those derived product names with product IDs using Value Lookups. This demonstrates the order in which trim, substitution and lookup operations are performed by the pump.
13	Column Filtering	This example shows how to filter the columns being processed by a Pivot Block. This is useful if the number of columns you are processing changes over time (eg. different number of columns for the days in the month).
14	Identity	This example shows how to use an Identity column which is created by the pump but managed by the SQL Server. When pumping the table of data to SQL, the Identity column is removed from the push to allow SQL to handle its self-incrementing value.
15	Calling a Stored Procedure	This example calls a SQL stored procedure once for every row of data processed. The fields identified in the Excel Pump 'Columns' are used as named parameters when passing values into the stored procedure.
16	Data Types	This example has a separate column for every supported data type in the pump.

Simple Import

This example transforms a simple grid of Excel data, 1-to-1 into a corresponding SQL table. For each column in the Excel data, a corresponding SQL field is created. No data transformations or re-shaping is performed. If the Excel sheet is updated over time with new rows of data, the new rows will be inserted into the SQL table. If rows are removed from the Excel sheet, then the SQL data will be **Marked** as deleted, but the data will be left intact.

Excel Source

	A	B	C	D	E	F
2						
3	Standard Import	Barley	Flour	Wheat	Canola	
4	01-Jan-2018	1.1	1.2	1.3	1.4	
5	02-Jan-2018	2.1	2.2	2.3	2.4	
6	03-Jan-2018	3.1	3.2	3.3	3.4	
7	04-Jan-2018	4.1	4.2	4.3	4.4	
8	Totals	10.4	10.8	11.2	11.6	
9						

SQL Destination

Results					
	Date	Barley	Flour	Wheat	Canola
1	2018-01-01	1.10	1.20	1.30	1.40
2	2018-01-02	2.10	2.20	2.30	2.40
3	2018-01-03	3.10	3.20	3.30	3.40
4	2018-01-04	4.10	4.20	4.30	4.40

```
<group name="ExampleData" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="SimpleImport" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Simple" orderByField="Date" mergeFields="Date" transferMode="Basic"
    includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
    <dataBlock name="SimpleImport" worksheet="Standard" startRefSearchCell="A1" endRefSearchCell="A8" startRefType="Absolute" startRef="A3" endRefType="Value" endRef="Totals" maxBlocks="1">
      <rowRules>
        <rowRule name="HeadingRow" columnRefType="RelativeToStart" columnRef="0" ruleType="Value" ruleExpression="Standard Import" ruleAction="Exclude" />
        <rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="1" ruleType="Compare" ruleExpression=" > 0" ruleAction="Include" />
        <rowRule name="TotalsRow" columnRefType="RelativeToStart" columnRef="0" ruleType="Value" ruleExpression="Totals" ruleAction="Exclude" />
      </rowRules>
      <columns>
        <column name="Date" dataType="date" columnRefType="RelativeToStart" columnRef="0" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Barley" dataType="decimal(10,2)" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Flour" dataType="decimal(10,2)" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Wheat" dataType="decimal(10,2)" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Canola" dataType="decimal(10,2)" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
      </columns>
    </dataBlock>
  </importItem>
</group>
```

Import with a common reference cell

This example transforms a simple grid of Excel data, and includes an additional column for a common Excel cell value which is shared with all rows being pumped. This is achieved by referencing the common cell using a fixed offset from the block starting point, rather than relative to each row being processed.

Excel Source

	A	B	C	D	E	F
10						
11	Common Reference Cell	Barley	Flour	Wheat	Canola	
12	01-Feb-2018					
13	Truck 1	1.1	1.2	1.3	1.4	
14	Truck 2	2.1	2.2	2.3	2.4	
15	Truck 3	3.1	3.2	3.3	3.4	
16	Truck 4	4.1	4.2	4.3	4.4	
17	Totals	10.4	10.8	11.2	11.6	
18						

SQL Destination

	Date	Vehicle	Barley	Flour	Wheat	Canola
1	2018-02-01	Truck 1	1.1	1.2	1.3	1.4
2	2018-02-01	Truck 2	2.1	2.2	2.3	2.4
3	2018-02-01	Truck 3	3.1	3.2	3.3	3.4
4	2018-02-01	Truck 4	4.1	4.2	4.3	4.4

```

<group name="ExampleData" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="ReferenceImport" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Reference" orderByField="Date" mergeFields="Date,Vehicle"
    transferMode="Basic" includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
    <dataBlock name="ReferenceImport" worksheet="Standard" startRefSearchCell="A10" endRefSearchCell="A20" startRefType="Value" startRef="Common Reference Cell" endRefType="Value" endRef="Totals" maxBlocks="1">
      <rowRules>
        <rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="0" ruleType="Contains" ruleExpression="Truck 1,Truck 2,Truck 3,Truck 4" ruleAction="Include" />
      </rowRules>
      <columns>
        <!-- The date is the same on every row, but mapping against RelativeToStart offset instead of against each row. -->
        <column name="Date" dataType="date" columnRefType="RelativeToStart" columnRef="0" rowRefType="RelativeToStart" rowRef="1" />
        <column name="Vehicle" dataType="nvarchar(50)" columnRefType="RelativeToStart" columnRef="0" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Barley" dataType="float" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Flour" dataType="float" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Wheat" dataType="float" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Canola" dataType="float" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
      </columns>
    </dataBlock>
  </importItem>
</group>

```

References the same cell each time an Excel row is processed by the pump

Import with trimming of text in Excel cells

This example removes unwanted text from the Excel data, prior to assessing the cell values for Row Rules, Substitution or Lookup operations. A similar operation is performed during the Column transformation for the data to be stored in SQL.

Excel Source

	A	B	C	D
2				
3	Trimming Example		Product	Quantity
4	1/01/2018	Wheat	1.1	
5	1/01/2018	Flour	1.2	
6	1/01/2018	Barley	1.3	
7	2/01/2018	Direct - Wheat	2.1	
8	2/01/2018	Flour (silo delivery)	2.2	
9	2/01/2018	Barley (direct)	2.3	
10	3/01/2018	Flour (direct)	3.1	
11	3/01/2018	Barley, from farm	3.2	
12	Totals		16.5	
13				

SQL Destination

	Date	Product	Quantity
1	2018-01-01	Wheat	1.1
2	2018-01-01	Flour	1.2
3	2018-01-01	Barley	1.3
4	2018-01-02	Wheat	2.1
5	2018-01-02	Flour	2.2
6	2018-01-02	Barley	2.3
7	2018-01-03	Flour	3.1
8	2018-01-03	Barley	3.2

```
<group name="ExampleData" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
```

```
<importItem name="TrimmingImport" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Trimming" orderByField="Date" mergeFields="Date,Product" transferMode="Basic" includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
```

```
<dataBlock name="TrimmingImport" worksheet="Trimming" startRefSearchCell="A1" endRefSearchCell="A20" startRefType="Value" startRef="Trimming Example" endRefType="Value" endRef="Totals" maxBlocks="1">
```

```
<rowRules>
```

```
<rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="1" ruleType="Contains" ruleExpression="Wheat,Flour,Barley,Canola" ruleAction="Include" trimLeadingWhiteSpace="true" trimTrailingWhiteSpace="true" >
```

```
<trimming>
```

```
<trimChars trimAllChars="" trimLeadingChars="Direct-" trimTrailingChars="{silodelivery}" />
```

```
<trimChars trimTrailingChars="{direct}" />
```

```
<trimChars trimTrailingChars=",fromfarm" />
```

```
</trimming>
```

```
</rowRule>
```

```
</rowRules>
```

```
<columns>
```

```
<column name="Date" dataType="date" columnRefType="RelativeToStart" columnRef="0" rowRefType="RelativeToRow" rowRef="0" />
```

```
<column name="Product" dataType="nvarchar(50)" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" trimLeadingWhiteSpace="true" trimTrailingWhiteSpace="true">
```

```
<trimming>
```

```
<trimChars trimAllChars="" trimLeadingChars="Direct-" trimTrailingChars="{silodelivery}" />
```

```
<trimChars trimTrailingChars="{direct}" />
```

```
<trimChars trimTrailingChars=",fromfarm" />
```

```
</trimming>
```

```
</column>
```

```
<column name="Quantity" dataType="float" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
```

```
</columns>
```

```
</dataBlock>
```

```
</importItem>
```

```
</group>
```

Removes characters from Excel cells prior to assessing the row rule

Removes characters from Excel cells before sending to SQL.

Data Pivot

This example transforms a row of Excel header values into a SQL column, and then maps the remaining data in the Excel block against that transformation. This is useful when wide rows of Excel data need to be normalised into narrow SQL tables. This is the most common type of Pivot operation performed by the Khronos Excel Pump.

Excel Source

	A	B	C	D	E	F
2						
3	Narrow Pivot	1/02/2018	2/02/2018	3/02/2018	4/02/2018	
4	Barley	2.1				
5	Flour		2.2			
6	Wheat			2.3		
7	Canola				2.4	
8	Totals	2.1	2.2	2.3	2.4	
9						

SQL Destination

Results			
	Product	Date	Value
1	Barley	2018-02-01	2.1
2	Flour	2018-02-02	2.2
3	Wheat	2018-02-03	2.3
4	Canola	2018-02-04	2.4

```
<group name="ExamplePivots" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="NarrowPivot" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Pivot_Narrow" orderByField="Date" mergeFields="Product,Date"
    transferMode="Basic" includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
```

```
<dataBlock name="NarrowPivot" worksheet="Pivot" startRefSearchCell="A1" endRefSearchCell="B50" startRefType="Value" startRef="Narrow Pivot" endRefType="Value" endRef="Totals" maxBlocks="1">
```

```
<rowRules>
  <rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="0" ruleType="Contains" ruleExpression="Barley,Flour,Wheat,Canola" ruleAction="Include" />
</rowRules>
```

Uses the same cell value for all cells in the same row which are being pivoted

```
<columns>
  <!-- Include the Product as its own column, beside the pivoted data. -->
  <column name="Product" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToRow" rowRef="0" />
</columns>
```

```
<pivots>
  <!-- Translate header row into a Column of Dates -->
  <pivotHeaderRow name="Date" dataType="date" >
    <column name="Date1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToStart" rowRef="0" />
  </pivotHeaderRow>
```

Uses the same header row values from Excel for each pivoted block of data sent to SQL

```
<!-- Translate Excel's Wide list of Value Columns into a single SQL Column of values -->
  <pivotBlock name="Value" dataType="float" ruleType="Compare" ruleExpression="> 0" ruleAction="Include" >
    <column name="Value1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
  </pivotBlock>
```

Pivots a wide set of Excel columns into a narrow SQL table with a SQL row for each Excel column in the block

```
</pivots>
</dataBlock>
</importItem>
</group>
```

Data Pivot with common reference cells

This example performs a normal Pivot operation, and then includes additional columns from Excel in each row of the SQL table. This is useful when your Excel spreadsheet has a number of columns with data which pertains to a wide-row of values.

Excel Source

	A	B	C	D	E	F	G
21							
22	Reference Pivot	Product	1/02/2018	2/02/2018	3/02/2018	4/02/2018	
23	Truck 1	Barley	2.1		4.1		
24	Truck 1	Flour	2.2	3.2		5.1	
25	Truck 2	Wheat	2.3		4.3		
26	Truck 2	Flour		3.4	4.4	5.2	
27	Truck 3	Wheat	2.5	3.5			
28	Truck 1	Canola			4.6		
29	Totals		9.1	10.1	17.4	10.3	
30							

SQL Destination

	Vehicle	Product	Date	Value
1	Truck 1	Barley	2018-02-01	2.1
2	Truck 1	Barley	2018-02-03	4.1
3	Truck 1	Flour	2018-02-01	2.2
4	Truck 1	Flour	2018-02-02	3.2
5	Truck 1	Flour	2018-02-04	5.1
6	Truck 2	Wheat	2018-02-01	2.3
7	Truck 2	Wheat	2018-02-03	4.3
8	Truck 2	Flour	2018-02-02	3.4
9	Truck 2	Flour	2018-02-03	4.4
10	Truck 2	Flour	2018-02-04	5.2
11	Truck 3	Wheat	2018-02-01	2.5
12	Truck 3	Wheat	2018-02-02	3.5
13	Truck 1	Canola	2018-02-03	4.6

```

<group name="ExamplePivots" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="ReferencePivot" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Pivot_Reference" orderByField="Date" mergeFields="Product,Vehicle,Date"
    transferMode="Basic" includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
    <dataBlock name="NarrowPivot" worksheet="Pivot" startRefSearchCell="A1" endRefSearchCell="B50" startRefType="Value" startRef="Reference Pivot" endRefType="Value" endRef="Totals" maxBlocks="1">
      <rowRules>
        <rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="1" ruleType="Contains" ruleExpression="Barley,Flour,Wheat,Canola" ruleAction="Include" />
      </rowRules>
      <columns>
        <!-- Include the and Vehicle Product as their own columns, beside the pivoted data. -->
        <column name="Vehicle" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Product" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="B" rowRefType="RelativeToRow" rowRef="0" />
      </columns>
      <pivots>
        <!-- Translate header row into a Column of Dates -->
        <pivotHeaderRow name="Date" dataType="date" >
          <column name="Date1" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date2" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date3" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date4" columnRefType="RelativeToStart" columnRef="5" rowRefType="RelativeToStart" rowRef="0" />
        </pivotHeaderRow>
        <!-- Translate all Value Columns in Excel into a single SQL Column of values -->
        <pivotBlock name="Value" dataType="numeric(10,2)" ruleType="Compare" ruleExpression="> 0" ruleAction="Include" >
          <column name="Value1" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
          <column name="Value2" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
        </pivotBlock>
      </pivots>
    </dataBlock>
  </importItem>
</group>

```

Uses the same two cell values for all cells in the same row which are being pivoted

```
<column name="Value3" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />  
<column name="Value4" columnRefType="RelativeToStart" columnRef="5" rowRefType="RelativeToRow" rowRef="0" />  
</pivotBlock>  
</pivots>  
</dataBlock>  
</importItem>  
</group>
```

Pivot Block with a multiple header rows

This example performs a normal Pivot operation, but includes two header rows in the transformation. Both header rows end up in columns beside the pivoted values.

Excel Source

	A	B	C	D	E	F
37						
38	Double Pivot	1/02/20 18	2/02/20 18	3/02/20 18	4/02/20 18	
39		Truck 1	Truck 2	Truck 3	Truck 4	
40	Barley	2.1		4.1		
41	Flour	2.2	3.2		5.1	
42	Wheat	2.3		4.3		
43	Flour		3.4	4.4	5.2	
44	Wheat	2.5	3.5			
45	Canola			4.6		
46	Totals	9.1	10.1	17.4	10.3	
47						

SQL Destination

	Product	Date	Vehicle	Value
1	Barley	2018-02-01	Truck 1	2.1
2	Barley	2018-02-03	Truck 3	4.1
3	Flour	2018-02-01	Truck 1	2.2
4	Flour	2018-02-02	Truck 2	3.2
5	Flour	2018-02-04	Truck 4	5.1
6	Wheat	2018-02-01	Truck 1	2.3
7	Wheat	2018-02-03	Truck 3	4.3
8	Flour	2018-02-02	Truck 2	3.4
9	Flour	2018-02-03	Truck 3	4.4
10	Flour	2018-02-04	Truck 4	5.2
11	Wheat	2018-02-01	Truck 1	2.5
12	Wheat	2018-02-02	Truck 2	3.5
13	Canola	2018-02-03	Truck 3	4.6

```
<group name="ExamplePivots" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
```

```
<importItem name="DoublePivot" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Pivot_Double" orderByField="Date" mergeFields="Product,Vehicle,Date" transferMode="Basic" includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
```

```
<dataBlock name="DoublePivot" worksheet="Pivot" startRefSearchCell="A1" endRefSearchCell="B50" startRefType="Value" startRef="Double Pivot" endRefType="Value" endRef="Totals" maxBlocks="1">
<rowRules>
<rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="0" ruleType="Contains" ruleExpression="Barley,Flour,Wheat,Canola" ruleAction="Include" />
</rowRules>
```

```
<columns>
<!-- Include the and Vehicle Product as their own columns, beside the pivoted data. -->
<column name="Product" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToRow" rowRef="0" />
</columns>
```

```
<pivots>
<!-- Translate header row into a Column of Dates -->
<pivotHeaderRow name="date" dataType="Date" >
<column name="Date1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToStart" rowRef="0" />
<column name="Date2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToStart" rowRef="0" />
<column name="Date3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToStart" rowRef="0" />
<column name="Date4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToStart" rowRef="0" />
</pivotHeaderRow>
```

```
<pivotHeaderRow name="Vehicle" dataType="nvarchar(50)" >
<column name="Vehicle1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToStart" rowRef="1" />
<column name="Vehicle2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToStart" rowRef="1" />
<column name="Vehicle3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToStart" rowRef="1" />
```

Pivots two header rows and aligns them with each processed cell in the Pivot Block

```
        <column name="Vehicle4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToStart" rowRef="1" />
    </pivotHeaderRow>

    <!-- Translate all Value Columns in Excel into a single SQL Column of values -->
    <pivotBlock name="Value" dataType="float" ruleType="Compare" ruleExpression="> 0" ruleAction="Include" >
        <column name="Value1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Value2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Value3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Value4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
    </pivotBlock>
</pivots>
</dataBlock>
</importItem>
</group>
```

Wide Pivot

This example transforms a wide table of Excel data into a wide, multi-column table in SQL via a Pivot function, whereby each specific value in one Excel column becomes its own field in the SQL destination table (versus the narrow table which a standard pivot performs). This is an uncommonly used operation, and generally not recommended by SQL best practices.

Excel Source

	A	B	C	D	E	F
10						
11	Wide Pivot	1/02/2018	2/02/2018	3/02/2018	4/02/2018	
12	Barley	1.1			1.4	
13	Flour		2.2			
14	Wheat			3.3		
15	Canola	4.1	4.2	4.3	4.4	
16	Totals	5.2	6.4	7.6	5.8	
17						

SQL Destination

	Date	Barley	Flour	Wheat	Canola
1	2018-02-01	1.1	NULL	NULL	NULL
2	2018-02-01	NULL	NULL	NULL	4.1
3	2018-02-02	NULL	NULL	NULL	4.2
4	2018-02-02	NULL	2.2	NULL	NULL
5	2018-02-03	NULL	NULL	3.3	NULL
6	2018-02-03	NULL	NULL	NULL	4.3
7	2018-02-04	NULL	NULL	NULL	4.4
8	2018-02-04	1.4	NULL	NULL	NULL

```
<group name="ExamplePivots" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="WidePivot" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Pivot_Wide" orderByField="Date" mergeFields="Date" transferMode="Basic"
    includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
    <dataBlock name="WidePivot" worksheet="Pivot" startRefSearchCell="A1" endRefSearchCell="A50" startRefType="Value" startRef="Wide Pivot" endRefType="Value" endRef="Totals" maxBlocks="1">
      <rowRules>
        <rowRule name="ValidRows" columnRefType="RelativeToStart" columnRef="0" ruleType="Contains" ruleExpression="Barley,Wheat,Flour,Canola" ruleAction="Include" />
      </rowRules>
      <pivots>
        <!-- Translate header row into a Column of Dates -->
        <pivotHeaderRow name="Date" dataType="date" >
          <column name="Date1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToStart" rowRef="0" />
        </pivotHeaderRow>
        <!-- Translate Barley row into a Column of Barley values -->
        <pivotBlock name="Barley" dataType="float" ruleType="Compare" ruleExpression="> 0" ruleAction="Include" >
          <rowRules>
            <rowRule name="ValidRows" columnRefType="RelativeToStart" columnRef="0" ruleType="Value" ruleExpression="Barley" ruleAction="Include" />
          </rowRules>
          <column name="Barley1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" />
          <column name="Barley2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
          <column name="Barley3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
          <column name="Barley4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
        </pivotBlock>
        <!-- Translate Wheat row into a Column of Wheat values -->
        <pivotBlock name="Wheat" dataType="float" ruleType="Compare" ruleExpression="> 0" ruleAction="Include" >
          <rowRules>
```

Pivot Block rules control which rows are processed for each destination SQL column

```

        <rowRule name="ValidRows" columnRefType="RelativeToStart" columnRef="0" ruleType="Value" ruleExpression="Wheat" ruleAction="Include" />
    </rowRules>
    <column name="Wheat1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Wheat2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Wheat3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Wheat4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
</pivotBlock>

<!-- Translate Flour row into a Column of Flour values -->
<pivotBlock name="Flour" dataType="float" ruleType="Compare" ruleExpression=" > 0" ruleAction="Include" >
    <rowRules>
        <rowRule name="ValidRows" columnRefType="RelativeToStart" columnRef="0" ruleType="Value" ruleExpression="Flour" ruleAction="Include" />
    </rowRules>
    <column name="Flour1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Flour2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Flour3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Flour4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
</pivotBlock>

<!-- Translate Canola row into a Column of Barley values -->
<pivotBlock name="Canola" dataType="float" ruleType="Compare" ruleExpression=" > 0" ruleAction="Include" >
    <rowRules>
        <rowRule name="ValidRows" columnRefType="RelativeToStart" columnRef="0" ruleType="Value" ruleExpression="Canola" ruleAction="Include" />
    </rowRules>
    <column name="Canola1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Canola2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Canola3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Canola4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
</pivotBlock>
</pivots>
</dataBlock>
</importItem>
</group>

```

Group and Pivot

This example finds values at the top of a data group, and uses those values in every row in that group. It relies on the white space in the blank rows between each group to be able to find the group headers (or footer) values. Group values can be found using:

- **TopOfValues** Finds the top cell in the block prior to any blank cell (white space).
- **TopOfBlanks** Finds the first cell which is not blank, at the top of the data block.
- **BottomOfValues** Finds the bottom cell in the block prior to any blank cell.
- **BottomOfBlanks** Finds the first cell which is not blank, at the bottom of the data block.

TopOfBlanks and **BottomOfBlanks** are typically used in adjoining columns, and not the main reference column.

Excel Source

	A	B	C	D	E	F	G
2							
3	Group Header Example	Schedule	1/02/2018	2/02/2018	3/02/2018	4/02/2018	
4							
5	Inloading	Truck 1					
6		Barley	1.1		3.1		
7		Flour	1.2	2.2			
8		Wheat			3.3		
9		Canola	1.4			4.4	
10							
11	Inloading	Truck 2					
12		Barley	1.5		3.5		
13		Flour		2.6			
14		Wheat			3.7		
15		Canola	1.8	2.8		4.8	
16							
17	Outloading	Truck 3					
18		Barley			-3.1		
19		Flour	-1.2		-3.2		
20		Wheat	-1.3				
21		Canola				-4.4	
22		Totals	4.5	7.6	7.3	4.8	
23							

SQL Destination

	Type	Vehicle	Product	Date	Value
1	Inloading	Truck 1	Barley	2018-02-01	1.1
2	Inloading	Truck 1	Barley	2018-02-03	3.1
3	Inloading	Truck 1	Flour	2018-02-01	1.2
4	Inloading	Truck 1	Flour	2018-02-02	2.2
5	Inloading	Truck 1	Wheat	2018-02-03	3.3
6	Inloading	Truck 1	Canola	2018-02-01	1.4
7	Inloading	Truck 1	Canola	2018-02-04	4.4
8	Inloading	Truck 2	Barley	2018-02-01	1.5
9	Inloading	Truck 2	Barley	2018-02-03	3.5
10	Inloading	Truck 2	Flour	2018-02-02	2.6
11	Inloading	Truck 2	Wheat	2018-02-03	3.7
12	Inloading	Truck 2	Canola	2018-02-01	1.8
13	Inloading	Truck 2	Canola	2018-02-02	2.8
14	Inloading	Truck 2	Canola	2018-02-04	4.8
15	Outloading	Truck 3	Barley	2018-02-03	-3.1
16	Outloading	Truck 3	Flour	2018-02-01	-1.2
17	Outloading	Truck 3	Flour	2018-02-03	-3.2
18	Outloading	Truck 3	Wheat	2018-02-01	-1.3
19	Outloading	Truck 3	Canola	2018-02-04	-4.4


```

<group name="ExamplePivots" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="GroupPivot" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Pivot_Group" orderByField="Date" mergeFields="Product,Date,Vehicle,Type"
    transferMode="Basic" includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
    <dataBlock name="GroupPivot" worksheet="Groups" startRefSearchCell="A1" endRefSearchCell="B50" startRefType="Value" startRef="Group Header Example" endRefType="Value" endRef="Totals" maxBlocks="1">
      <rowRules>
        <rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="1" ruleType="Contains" ruleExpression="Barley,Flour,Wheat,Canola" ruleAction="Include" />
      </rowRules>
      <columns>
        <!-- Include the Type and the Vehicle as their own columns, beside the pivoted data. -->
        <column name="Type" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="A" rowRefType="Group" rowRef="TopOfBlanks" />
        <column name="Vehicle" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="B" rowRefType="Group" rowRef="TopOfValues" />
        <column name="Product" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="B" rowRefType="RelativeToRow" rowRef="0" />
      </columns>
      <pivots>
        <!-- Translate header row into a Column of Dates -->
        <pivotHeaderRow name="Date" dataType="date" >
          <column name="Date1" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date2" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date3" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date4" columnRefType="RelativeToStart" columnRef="5" rowRefType="RelativeToStart" rowRef="0" />
        </pivotHeaderRow>
        <!-- Translate Excel's Wide list of Value Columns into a single SQL Column of values -->
        <pivotBlock name="Value" dataType="float" ruleType="Compare" ruleExpression="&lt;&gt; 0" ruleAction="Include" >
          <column name="Value1" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
          <column name="Value2" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
          <column name="Value3" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
          <column name="Value4" columnRefType="RelativeToStart" columnRef="5" rowRefType="RelativeToRow" rowRef="0" />
        </pivotBlock>
      </pivots>
    </dataBlock>
  </importItem>
</group>

```

Group Headings are located at the top of the data blocks, relative to the current row being processed by the pump.

Recursive Data Block

This example identifies three separate blocks of data to process and import as part of a single **Data Block** definition. The recursion is based on finding the heading “Repeat Example” followed by “Totals”, and it set in motion by having a maxBlocks parameter value greater than 1 (basically all data blocks are recursive, but typically restricted to 1 iteration through this attribute setting).

Excel Source

SQL Destination

	A	B	C	D	E	F
2						
3	Repeat Example	1/03/2018	2/03/2018	3/03/2018	4/03/2018	
4	March					
5	Barley	1.1				
6	Flour		1.2			
7	Wheat			1.3		
8	Canola				1.4	
9	Sub-Totals	1.1	1.2	1.3	1.4	
10						
11	Repeat Example	1/03/2018	2/03/2018	3/03/2018	4/03/2018	
12	April					
13	Barley				2.1	
14	Flour			2.2		
15	Wheat		2.3			
16	Canola	2.4				
17	Sub-Totals	2.4	2.3	2.2	2.1	
18						
19	Repeat Example	1/03/2018	2/03/2018	3/03/2018	4/03/2018	
20	May					
21	Barley		3.1			
22	Flour				3.3	
23	Wheat	3.2				
24	Canola			3.4		
25	Totals	3.2	3.1	3.4	3.3	
26						

Results				
	Month	Product	Date	Value
1	March	Barley	2018-03-01	1.1
2	March	Flour	2018-03-02	1.2
3	March	Wheat	2018-03-03	1.3
4	March	Canola	2018-03-04	1.4
5	April	Barley	2018-03-04	2.1
6	April	Flour	2018-03-03	2.2
7	April	Wheat	2018-03-02	2.3
8	April	Canola	2018-03-01	2.4
9	May	Barley	2018-03-02	3.1
10	May	Flour	2018-03-04	3.3
11	May	Wheat	2018-03-01	3.2
12	May	Canola	2018-03-03	3.4

```

<group name="ExampleRecursion" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="Recursion" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Recursion" orderByField="Date" mergeFields="Date,Month,Product"
    transferMode="Basic" includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
    <dataBlock name="Recursion" worksheet="Recursive" startRefSearchCell="A1" endRefSearchCell="C50" startRefType="Value" startRef="Repeat Example" endRefType="Contains" endRef="Sub-Totals,Totals" maxBlocks="10">
      <rowRules>
        <rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="0" ruleType="Contains" ruleExpression="Barley,Flour,Wheat,Canola" ruleAction="Include" />
      </rowRules>
      <columns>
        <!-- Include the Product as its own column, beside the pivoted data. -->
        <column name="Month" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToStart" rowRef="1" />
        <column name="Product" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToRow" rowRef="0" />
      </columns>
      <pivots>
        <!-- Translate header row into a Column of Dates -->
        <pivotHeaderRow name="Date" dataType="date" >
          <column name="Date1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToStart" rowRef="0" />
          <column name="Date4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToStart" rowRef="0" />
        </pivotHeaderRow>
        <!-- Translate Excel's Wide list of Value Columns into a single SQL Column of values -->
        <pivotBlock name="Value" dataType="real" ruleType="Compare" ruleExpression="> 0" ruleAction="Include" >
          <column name="Value1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" />
          <column name="Value2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
          <column name="Value3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
          <column name="Value4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
        </pivotBlock>
      </pivots>
    </dataBlock>
  </importItem>
</group>

```

Looks for up to 10 instances of the recursion pattern within the overall search frame.

Value Substitution

This example substitutes the text name of the products in Excel for an integer identity value before sending the data to SQL. The substitution is driven by a static list of values stored in the Khronos Excel Pump configuration file. A more advanced option is shown in the next example, whereby the Identity values can be sourced from lookup lists in the SQL Server.

Excel Source

	A	B	C	D
10				
11	Lookup Example	Product	Quantity	
12	1/01/2018	Wheat	1.1	
13	2/01/2018	Flour	2.1	
14	3/01/2018	Barley	3.1	
15	4/01/2018	Canola	4.1	
16	Totals		10.4	
17				

SQL Destination

Results			
	Date	ProductId	Quantity
1	2018-01-01	2	1.1
2	2018-01-02	1	2.1
3	2018-01-03	3	3.1
4	2018-01-04	4	4.1

```

<group name="SubstitutionExample" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="SubstitutionData" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Substitution" orderByField="Date" mergeFields="ProductId,Date"
    transferMode="Basic" includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">

    <dataBlock name="SubstitutionData" worksheet="Lookups" startRefSearchCell="A1" endRefSearchCell="B50" startRefType="Value" startRef="Substitution Example" endRefType="Value" endRef="Totals" maxBlocks="1">
      <rowRules>
        <rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="1" ruleType="Contains" ruleExpression="Barley,Flour,Wheat,Canola" ruleAction="Include" />
      </rowRules>

      <columns>
        <!-- Include the Product as its own column, beside the pivoted data. -->
        <column name="Date" dataType="date" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToRow" rowRef="0" />
        <column name="ProductId" dataType="int" columnRefType="Absolute" columnRef="B" rowRefType="RelativeToRow" rowRef="0" substitutionList="ProductIDs" />
        <column name="Quantity" dataType="float" columnRefType="Absolute" columnRef="C" rowRefType="RelativeToRow" rowRef="0" />
      </columns>
    </dataBlock>
  </importItem>
</group>

<substitutionLists>
  <list name="ProductIDs">
    <listItem sourceValue="Flour" substituteValue="1" />
    <listItem sourceValue="Wheat" substituteValue="2" />
    <listItem sourceValue="Barley" substituteValue="3" />
    <listItem sourceValue="Canola" substituteValue="4" />
  </list>
</substitutionLists>

```

Substitutes the Excel cell text for an ID value.

Value Lookup from SQL

This example exchanges the text name of the products for an integer identity value sourced from the Products master table in SQL. The text value from Excel is never sent to SQL; only the identity value is. This functionality relies on the Lookup source in SQL having corresponding values for all lookup values found in Excel. If any are missing, then the pump operation will fail.

It can be a good idea to process your Excel data twice:

1. Once to import all the Products to your master SQL table of products, and if any new ones appear in Excel then they will automatically be assigned a new identity field in SQL (assuming you use identity fields in your SQL master data tables).
2. A second time to import the main data block, where it can lookup the identity values including for any new products just created.

Excel Source

	A	B	C	D
2				
3				
4	1/01/2018	Wheat	2.1	
5	2/01/2018	Flour	2.1	
6	3/01/2018	Barley	3.1	
7	4/01/2018	Canola	4.1	
8	Totals		10.4	
9				

SQL Destination

	Date	ProductId	Quantity
1	2018-01-01	2	1.1
2	2018-01-02	1	2.1
3	2018-01-03	3	3.1
4	2018-01-04	4	4.1

	Id	Product
1	1	Flour
2	2	Wheat
3	3	Barley
4	4	Canola

```
<group name="SubstitutionExample" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="LookupData" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Lookup" orderByField="Date" mergeFields="ProductId,Date" transferMode="Basic"
    includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
    <dataBlock name="LookupData" worksheet="Lookups" startRefSearchCell="A1" endRefSearchCell="B50" startRefType="Value" startRef="Lookup Example" endRefType="Value" endRef="Totals" maxBlocks="1">
      <rowRules>
        <rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="1" ruleType="Contains" ruleExpression="Barley,Flour,Wheat,Canola" ruleAction="Include" />
      </rowRules>
      <columns>
        <!-- Include the Product as its own column, beside the pivoted data. -->
        <column name="Date" dataType="date" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToRow" rowRef="0" />
        <column name="ProductId" dataType="int" columnRefType="Absolute" columnRef="B" rowRefType="RelativeToRow" rowRef="0"
          lookupDataConnection="TargetServer" lookupTable="tb_ExampleLookup_Products" lookupField="Product" lookupValue="Id" />
        <column name="Quantity" dataType="float" columnRefType="Absolute" columnRef="C" rowRefType="RelativeToRow" rowRef="0" />
      </columns>
    </dataBlock>
  </importItem>
</group>
```

Substitutes the Excel cell text for an ID value which is sourced from a SQL table.

Combo Pivot

This example performs a standard Data Pivot from wide Excel rows to a narrow SQL table, replaces product descriptions from the Excel text values with product names using Value Substitution, and then replaces product names with product IDs from a SQL table using Value Lookups. This demonstrates the order in which trim, substitution and lookup operations are performed by the pump.

Excel Source

	A	B	C	D
18				
19	Combo Example	Product	Quantity	
20	1/02/2018	Grain: Wheat	1.1	
21	2/02/2018	Grain: Flour	2.1	
22	3/02/2018	Grain: Barley	3.1	
23	4/02/2018	Grain: Canola	4.1	
24	Totals		10.4	
25				

SQL Destination

	ProductId	Date	Quantity
1	2	2018-02-01	1.1
2	1	2018-02-02	2.1
3	3	2018-02-03	3.1
4	4	2018-02-04	4.1

	Id	Product
1	1	Flour
2	2	Wheat
3	3	Barley
4	4	Canola

```
<group name="SubstitutionExample" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="ComboPivot" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Pivot_Combo" orderByField="Date" mergeFields="ProductId,Date"
    transferMode="Basic" includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
    <dataBlock name="ComboPivot" worksheet="Lookups" startRefSearchCell="A1" endRefSearchCell="B50" startRefType="Value" startRef="Combo Example" endRefType="Value" endRef="Totals" maxBlocks="1">
      <rowRules>
        <rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="1" ruleType="Contains" ruleExpression="Grain: Barley,Grain: Flour,Grain: Wheat,Grain: Canola" ruleAction="Include" />
      </rowRules>
      <columns>
        <!-- Include the Product as its own column, beside the pivoted data. -->
        <column name="ProductId" dataType="int" columnRefType="Absolute" columnRef="B" rowRefType="RelativeToRow" rowRef="0"
          substitutionList="ProductDescriptions" lookupDataConnection="TargetServer" lookupTable="tb_ExampleLookup_Products" lookupField="Product" lookupValue="Id" >
          <trimming>
            <trimChars trimAllChars=" " />
          </trimming>
        </column>
        <column name="Date" dataType="date" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToRow" rowRef="0" />
        <column name="Quantity" dataType="real" columnRefType="Absolute" columnRef="C" rowRefType="RelativeToRow" rowRef="0" />
      </columns>
    </dataBlock>
  </importItem>
</group>
```

1. Trim option is applied first
2. Substitution list is applied second
3. Lookup list is applied last

Column Filtering in a Pivot Block

This example performs a standard Data Pivot from wide Excel rows to a narrow SQL table, but only for cells which adhere to a set of specified Rules. This allows you to perform a Pivot operation on a variable number of columns, such as when you have one column for each day of the month, and February has less than another month. This example checks that the header row does not contain the word "Totals", and also that the header row is not empty. It attempts to process all other cells in the data block.

Excel Source

Variable Column Example	1/1/2018	2/1/2018	3/1/2018	4/1/2018	5/1/2018	6/1/2018	7/1/2018	8/1/2018	9/1/2018	10/1/2018	11/1/2018	12/1/2018	13/1/2018	14/1/2018	15/1/2018	16/1/2018	17/1/2018	18/1/2018	19/1/2018	20/1/2018	21/1/2018	22/1/2018	23/1/2018	24/1/2018	25/1/2018	26/1/2018	27/1/2018	28/1/2018	29/1/2018	31/1/2018	Totals	
Wheat	1.11	1.12	1.13	1.14	1.15	1.16	1.17	1.18	1.19	1.2	1.21	1.22	1.23	1.24	1.25	1.26	1.27	1.28	1.29	1.3	1.31	1.32	1.33	1.34	1.35	1.36	1.37	1.38	1.39	1.4	1.41	39.06
Flour	2.11	2.12	2.13	2.14	2.15	2.16	2.17	2.18	2.19	2.2	2.21	2.22	2.23	2.24	2.25	2.26	2.27	2.28	2.29	2.3	2.31	2.32	2.33	2.34	2.35	2.36	2.37	2.38	2.39	2.4	2.41	70.06
Barley	3.11	3.12	3.13	3.14	3.15	3.16	3.17	3.18	3.19	3.2	3.21	3.22	3.23	3.24	3.25	3.26	3.27	3.28	3.29	3.3	3.31	3.32	3.33	3.34	3.35	3.36	3.37	3.38	3.39	3.4	3.41	101.1
Canola	4.11	4.12	4.13	4.14	4.15	4.16	4.17	4.18	4.19	4.2	4.21	4.22	4.23	4.24	4.25	4.26	4.27	4.28	4.29	4.3	4.31	4.32	4.33	4.34	4.35	4.36	4.37	4.38	4.39	4.4	4.41	132.1
Totals	10.44	10.48	10.52	10.56	10.6	10.64	10.68	10.72	10.76	10.8	10.84	10.88	10.92	10.96	11	11.04	11.08	11.12	11.16	11.2	11.24	11.28	11.32	11.36	11.4	11.44	11.48	11.52	11.56	11.6	11.64	342.2

Variable Column Example	1/20/2018	2/20/2018	3/20/2018	4/20/2018	5/20/2018	6/20/2018	7/20/2018	8/20/2018	9/20/2018	10/20/2018	11/20/2018	12/20/2018	13/20/2018	14/20/2018	15/20/2018	16/20/2018	17/20/2018	18/20/2018	19/20/2018	20/20/2018	21/20/2018	22/20/2018	23/20/2018	24/20/2018	25/20/2018	26/20/2018	27/20/2018	28/20/2018	Totals
Wheat	5.11	5.12	5.13	5.14	5.15	5.16	5.17	5.18	5.19	5.2	5.21	5.22	5.23	5.24	5.25	5.26	5.27	5.28	5.29	5.3	5.31	5.32	5.33	5.34	5.35	5.36	5.37	5.38	146.9
Flour	6.11	6.12	6.13	6.14	6.15	6.16	6.17	6.18	6.19	6.2	6.21	6.22	6.23	6.24	6.25	6.26	6.27	6.28	6.29	6.3	6.31	6.32	6.33	6.34	6.35	6.36	6.37	6.38	174.9
Barley	7.11	7.12	7.13	7.14	7.15	7.16	7.17	7.18	7.19	7.2	7.21	7.22	7.23	7.24	7.25	7.26	7.27	7.28	7.29	7.3	7.31	7.32	7.33	7.34	7.35	7.36	7.37	7.38	202.9
Canola	8.11	8.12	8.13	8.14	8.15	8.16	8.17	8.18	8.19	8.2	8.21	8.22	8.23	8.24	8.25	8.26	8.27	8.28	8.29	8.3	8.31	8.32	8.33	8.34	8.35	8.36	8.37	8.38	230.9
Totals	26.44	26.48	26.52	26.56	26.6	26.64	26.68	26.72	26.76	26.8	26.84	26.88	26.92	26.96	27	27.04	27.08	27.12	27.16	27.2	27.24	27.28	27.32	27.36	27.4	27.44	27.48	27.52	765.4

SQL Destination

Product	Date	Value	
1	Barley	2018-01-27	3.37
2	Canola	2018-01-27	4.37
3	Flour	2018-01-27	2.37
4	Wheat	2018-01-27	1.37
5	Barley	2018-01-28	3.38
6	Canola	2018-01-28	4.38
7	Flour	2018-01-28	2.38
8	Wheat	2018-01-28	1.38
9	Barley	2018-01-29	3.39
10	Canola	2018-01-29	4.39
11	Flour	2018-01-29	2.39
12	Wheat	2018-01-29	1.39
13	Barley	2018-01-30	3.4
14	Canola	2018-01-30	4.4
15	Flour	2018-01-30	2.4
16	Wheat	2018-01-30	1.4
17	Barley	2018-01-31	3.41
18	Canola	2018-01-31	4.41
19	Flour	2018-01-31	2.41
20	Wheat	2018-01-31	1.41
21	Barley	2018-02-27	7.37
22	Canola	2018-02-27	8.37
23	Flour	2018-02-27	6.37
24	Wheat	2018-02-27	5.37
25	Barley	2018-02-28	7.38
26	Canola	2018-02-28	8.38
27	Flour	2018-02-28	6.38
28	Wheat	2018-02-28	5.38

```
<group name="VariableColumnExample" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="__KhronosExcelStatus" statusConnectionTimeout="60">
  <importItem name="VariableColumnData" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_FilteredColumns" orderByField="Date" mergeFields="Product,Date"
    transferMode="Basic" includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
    <dataBlock name="FilteredColumnData" worksheet="VariableColumns" startRefSearchCell="A1" endRefSearchCell="B20" startRefType="Value" startRef="Variable Column Example" endRefType="Value" endRef="Totals" maxBlocks="2">
      <rowRules>
        <rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="0" rowRefType="RelativeToRow" rowRef="0" ruleType="Contains" ruleExpression="Barley,Flour,Wheat,Canola" ruleAction="Include" />
      </rowRules>
    </dataBlock>
  </importItem>
</group>
```

```

<columns>
  <!-- Include the Product as its own column, beside the pivoted data. -->
  <column name="Product" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToRow" rowRef="0" />
</columns>

<pivots>
  <!-- Filter out all rows which don't have a valid column of data. Base this on the header row contents. -->
  <cellRules>
    <cellRule name="NotTotalsColumn" columnRefType="RelativeToCell" columnRef="0" rowRefType="RelativeToStart" rowRef="0" ruleType="Value" ruleExpression="Totals" ruleAction="Exclude" />
    <cellRule name="NotHeaderEmpty" rowRefType="RelativeToStart" rowRef="0" ruleType="Value" ruleExpression="" ruleAction="Exclude" />
  </cellRules>

  <!-- Translate header row into a Column of Dates -->
  <pivotHeaderRow name="Date" dataType="date" >
    <column name="Date1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date5" columnRefType="RelativeToStart" columnRef="5" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date6" columnRefType="RelativeToStart" columnRef="6" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date7" columnRefType="RelativeToStart" columnRef="7" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date8" columnRefType="RelativeToStart" columnRef="8" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date9" columnRefType="RelativeToStart" columnRef="9" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date10" columnRefType="RelativeToStart" columnRef="10" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date11" columnRefType="RelativeToStart" columnRef="11" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date12" columnRefType="RelativeToStart" columnRef="12" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date13" columnRefType="RelativeToStart" columnRef="13" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date14" columnRefType="RelativeToStart" columnRef="14" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date15" columnRefType="RelativeToStart" columnRef="15" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date16" columnRefType="RelativeToStart" columnRef="16" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date17" columnRefType="RelativeToStart" columnRef="17" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date18" columnRefType="RelativeToStart" columnRef="18" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date19" columnRefType="RelativeToStart" columnRef="19" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date20" columnRefType="RelativeToStart" columnRef="20" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date21" columnRefType="RelativeToStart" columnRef="21" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date22" columnRefType="RelativeToStart" columnRef="22" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date23" columnRefType="RelativeToStart" columnRef="23" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date24" columnRefType="RelativeToStart" columnRef="24" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date25" columnRefType="RelativeToStart" columnRef="25" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date26" columnRefType="RelativeToStart" columnRef="26" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date27" columnRefType="RelativeToStart" columnRef="27" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date28" columnRefType="RelativeToStart" columnRef="28" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date29" columnRefType="RelativeToStart" columnRef="29" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date30" columnRefType="RelativeToStart" columnRef="30" rowRefType="RelativeToStart" rowRef="0" />
    <column name="Date31" columnRefType="RelativeToStart" columnRef="31" rowRefType="RelativeToStart" rowRef="0" />
  </pivotHeaderRow>

  <!-- Translate Excel's Wide list of Value Columns into a single SQL Column of values -->
  <pivotBlock name="Value" dataType="float" ruleType="Compare" ruleExpression="> 0" ruleAction="Include" >
    <column name="Value1" columnRefType="RelativeToStart" columnRef="1" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value2" columnRefType="RelativeToStart" columnRef="2" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value3" columnRefType="RelativeToStart" columnRef="3" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value4" columnRefType="RelativeToStart" columnRef="4" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value5" columnRefType="RelativeToStart" columnRef="5" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value6" columnRefType="RelativeToStart" columnRef="6" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value7" columnRefType="RelativeToStart" columnRef="7" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value8" columnRefType="RelativeToStart" columnRef="8" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value9" columnRefType="RelativeToStart" columnRef="9" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value10" columnRefType="RelativeToStart" columnRef="10" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value11" columnRefType="RelativeToStart" columnRef="11" rowRefType="RelativeToRow" rowRef="0" />
    <column name="Value12" columnRefType="RelativeToStart" columnRef="12" rowRefType="RelativeToRow" rowRef="0" />
  </pivotBlock>

```

Ignores all Pivot Block cells where the header row has the word "Totals", or is empty.


```
<column name="Value13" columnRefType="RelativeToStart" columnRef="13" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value14" columnRefType="RelativeToStart" columnRef="14" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value15" columnRefType="RelativeToStart" columnRef="15" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value16" columnRefType="RelativeToStart" columnRef="16" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value17" columnRefType="RelativeToStart" columnRef="17" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value18" columnRefType="RelativeToStart" columnRef="18" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value19" columnRefType="RelativeToStart" columnRef="19" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value20" columnRefType="RelativeToStart" columnRef="20" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value21" columnRefType="RelativeToStart" columnRef="21" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value22" columnRefType="RelativeToStart" columnRef="22" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value23" columnRefType="RelativeToStart" columnRef="23" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value24" columnRefType="RelativeToStart" columnRef="24" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value25" columnRefType="RelativeToStart" columnRef="25" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value26" columnRefType="RelativeToStart" columnRef="26" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value27" columnRefType="RelativeToStart" columnRef="27" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value28" columnRefType="RelativeToStart" columnRef="28" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value29" columnRefType="RelativeToStart" columnRef="29" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value30" columnRefType="RelativeToStart" columnRef="30" rowRefType="RelativeToRow" rowRef="0" />
<column name="Value31" columnRefType="RelativeToStart" columnRef="31" rowRefType="RelativeToRow" rowRef="0" />
</pivotBlock>
</pivots>
</dataBlock>
</importItem>
</group>
```

Identity Columns

This example defines an Identity column which has no source value in Excel, but instead uses the SQL Server to self-manage the value in that column when new rows are inserted into the table. This has the effect of creating a column in the SQL table with the following properties:

SQL Column Details	Description
Column Name	Specified by the user in the Excel Pump configuration.
Data Type	Bigint
Allow Nulls	False
Identity Specification	Yes
Identity Seed	1
Identity Increment	1
SQL Create Statement	[Id] [bigint] IDENTITY(1,1) NOT NULL

The Excel Pump will not source any values from the worksheet for this column, and will not attempt to insert or merge any values into this column in SQL. Instead it relies on SQL to service the identity attributes as new rows are inserted.

Excel Source

	A	B	C	D
2				
3	Identity Example	Product	Quantity	
4	1/01/2018	Wheat	1.1	
5	2/01/2018	Flour	2.1	
6	3/01/2018	Barley	3.1	
7	4/01/2018	Canola	4.1	
8	Totals		10.4	
9				

SQL Destination

Id	Date	ProductId	Quantity
1	2018-01-01	2	1.1
2	2018-01-02	1	2.1
3	2018-01-03	3	3.1
4	2018-01-04	4	4.1

```
<group name="IdentityExample" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
```

```
<importItem name="IdentityData" file="C:\_Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="tb_Example_Identity" orderByField="Date" mergeFields="ProductId,Date" transferMode="Basic"
includeSystemFields="true" deleteMode="Marked" maxBlockSize="10000" connectionTimeout="60">
```

```
<dataBlock name="IdentityData" worksheet="Identities" startRefSearchCell="A1" endRefSearchCell="B50" startRefType="Value" startRef="Identity Example" endRefType="Value" endRef="Totals" maxBlocks="1">
```

```
<rowRules>
```

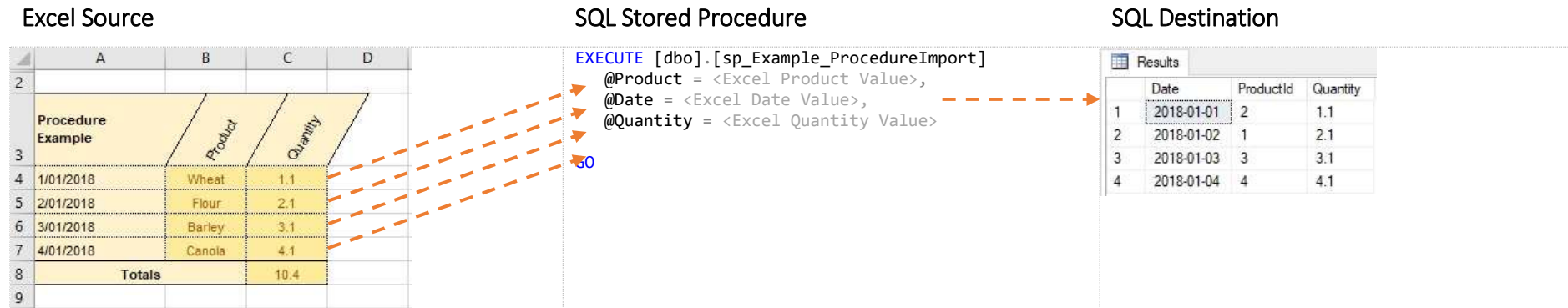
```
<rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="1" ruleType="Contains" ruleExpression="Barley,Flour,Wheat,Canola" ruleAction="Include" />
```

```
</rowRules>
```

```
<columns>
  <!-- Include an Identity column with no source reference. -->
  <column name="Id" dataType="identity" />
  <column name="Date" dataType="date" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToRow" rowRef="0" />
  <column name="ProductId" dataType="int" columnRefType="Absolute" columnRef="B" rowRefType="RelativeToRow" rowRef="0"
    lookupDataConnection="TargetServer" lookupTable="tb_ExampleLookup_Products" lookupField="Product" lookupValue="Id" />
  <column name="Quantity" dataType="float" columnRefType="Absolute" columnRef="C" rowRefType="RelativeToRow" rowRef="0" />
</columns>
</dataBlock>
</importItem>
</group>
```

Calling a Stored Procedure

This example sends the extracted Excel data to MS SQL via a Stored Procedure. Each row of processed data is passed into the Stored Procedure one-by-one, with the cell values from Excel passed into the Stored Procedure as parameter values. This relies on the column names in the configuration file to match exactly with the parameter names required as input parameters to the stored procedure.



```
<group name="ProcedureExample" enabled="true" statusTableEnabled="true" statusTableDestination="TargetServer" statusTableName="_KhronosExcelStatus" statusConnectionTimeout="60">
```

```
<importItem name="ProcedureData" file="C:\Khronos\KhronosExcelPump\Examples\ImportData.xlsx" dataDestination="TargetServer" destinationObject="sp_Example_ProcedureImport" transferMode="Procedure" includeSystemFields="false"
maxBlockSize="10000" connectionTimeout="60">
```

```
<dataBlock name="ProcedureData" worksheet="Procedure" startRefSearchCell="A1" endRefSearchCell="B50" startRefType="Value" startRef="Procedure Example" endRefType="Value" endRef="Totals" maxBlocks="1">
```

```
<rowRules>
```

```
<rowRule name="ValidRow" columnRefType="RelativeToStart" columnRef="1" ruleType="Contains" ruleExpression="Barley,Flour,Wheat,Canola" ruleAction="Include" />
```

```
</rowRules>
```

```
<columns>
```

```
<!-- Include the Product as its own column, beside the pivoted data. -->
```

```
<column name="Date" dataType="date" columnRefType="Absolute" columnRef="A" rowRefType="RelativeToRow" rowRef="0" />
```

```
<column name="Product" dataType="nvarchar(50)" columnRefType="Absolute" columnRef="B" rowRefType="RelativeToRow" rowRef="0" />
```

```
<column name="Quantity" dataType="float" columnRefType="Absolute" columnRef="C" rowRefType="RelativeToRow" rowRef="0" />
```

```
</columns>
```

```
</dataBlock>
```

```
</importItem>
```

```
</group>
```

The stored procedure used in this example is shown below as a reference:

```
-- =====
-- Author:      Start Services Pty Ltd
-- Create date: 2017-01-01
-- Description: Imports data from the Khronos Excel Pump
-- =====
ALTER PROCEDURE [dbo].[sp_Example_ProcedureImport]
/* Receive one row's values from the Excel Pump */
@Product nvarchar(50),
@Date datetime,
@Quantity float
```

```

AS
BEGIN
    SET NOCOUNT ON;

    /* Create the table if it doesn't exist */
    IF NOT EXISTS(SELECT 1 FROM sys.Objects WHERE Object_id = OBJECT_ID(N'dbo.tb_Example_Procedure') AND Type = N'U')
    BEGIN
        CREATE TABLE [dbo].[tb_Example_Procedure]( [Date] date, [ProductId] int, [Quantity] float )
    END

    /* Check if the product referenced in this Excel row already exists in the Products master data table */
    IF NOT EXISTS (SELECT 1 FROM dbo.tb_ExampleLookup_Products WHERE Product LIKE @Product)
    BEGIN
        /* If not, create a new product in the Products master data table */
        INSERT INTO dbo.tb_ExampleLookup_Products (Product)
        VALUES (@Product)
    END

    /* Retrieve the Product ID ready to insert into the data table */
    DECLARE @ProductID int
    SELECT TOP 1 @ProductID = Id FROM dbo.tb_ExampleLookup_Products WHERE Product LIKE @Product

    /* Merge the Excel data into the data table */
    MERGE INTO dbo.tb_Example_ProcedureData AS Target
    USING (SELECT @Date AS Date, @ProductID AS ProductId, @Quantity AS Quantity) AS Source
    ON Target.Date = Source.Date AND Target.ProductId = Source.ProductId
    WHEN NOT MATCHED THEN
        INSERT VALUES (Source.Date, Source.ProductId, Source.Quantity)
    WHEN MATCHED THEN
        UPDATE SET Target.Quantity = Source.Quantity;
END

```