Using Data from One Source to Report from Another

Overview

With **XLReporter** it is very easy to report on data from many different data sources like live data from PLCs, summary data from historians and alarm data from a relational database.

However, in some cases the data from these different sources needs to be combined together in a report to give a full picture of what is going on. For example, a report may need to show data samples from a historian over the day and also the alarms that occurred over that day as a single data table. Or a report may need to show all the alarms over a period of time and then show a set of process values recorded in the historian at the time of each alarm.
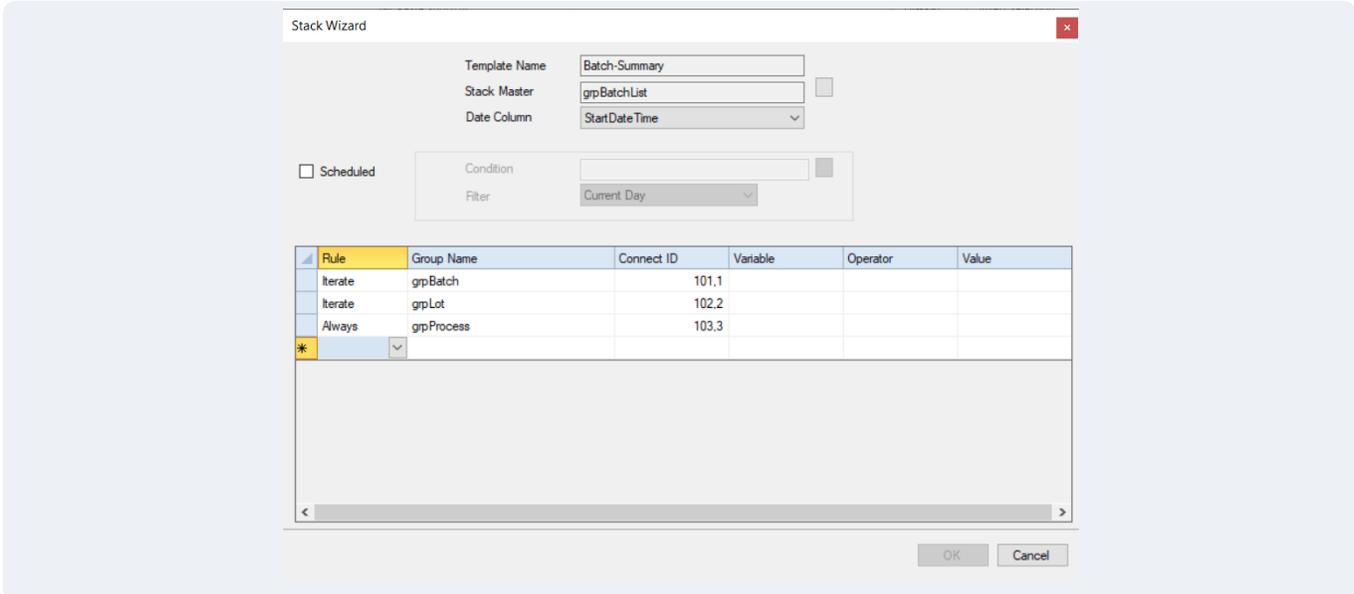
The following spotlight document highlights the features available to combine different data sources in the same report.

Stacked Reports

Stacked reports are a collection of **Sub Reports** that are combined (stacked) together according to a **Stack Rule** to produce the report. Sub reports may contain dependencies between each other.

The two main elements of a stacked report are the workbook template and a list of items which are used to drive the sub reports to the final report. Since there are several parts to stacked reports, a **Stacked Template Wizard** is provided to simplify the process.

Stack Wizard

The **Stack Master** is set to a database group which return a list of items. Each item in the list will be used to drive the rules specified in the grid.

Each row in the grid represents a sub report. When a report is produced, the **Stack Master** list determines the framework of the report. Each item of the list is applied to the rows in the grid top down.

Typical Scenario

Consider the following scenario: A report is required where a user is presented with a list of batches that have been produced and for each selected batch, the report must show the production lot. The list of batches has been recorded in a database table along with the start and end times when the batch ran.

Stack Wizard

Stack Wizard

Template Name	Batch-Summary
Stack Master	grpBatchList
Date Column	StartDateTime

☐ Scheduled

| Condition | |
| Filter | Current Day |

Rule	Group Name	Connect ID	Variable	Operator	Value
Iterate	grpBatch	101,1			
Iterate	grpLot	102,2			
Always	grpProcess	103,3			
*					

OK   Cancel

When configuring the **Stack Wizard**, the **Stack Master** is set to *grpBatchList*.

SQL Statement

```
SELECT Batch, StartDateTime, EndDateTime
FROM tblBatchLot
```

This is a **Database Data Group** that returns a list of *batches* along with the *start* and *end* times of the batch.

For each selected batch, the *grpBatch* group is executed.

SQL Statement

```
SELECT StartDateTime, EndDateTime, Batch
FROM tblBatchLot
WHERE Batch = '{grpBatch_List:Batch}'
AND Lot IS NULL
```

This is a **Database Data Group** that returns a single record for the batch selected from *grpBatchList*. This is used in a sub report to display header information for each selected batch.

The record returned from *grpBatch* is then used to execute *grpLot*.

SQL Statement

```
SELECT StartDateTime, EndDateTime, Lot
FROM tblBatchLot
WHERE Batch = '{grpBatch:Batch}'
AND Lot IS NOT NULL
```

This is a **Database Data Group** that returns a record for each lot run during the batch. This is used in a sub report to display header information for each lot in the batch.

Each lot record returned from *grpLot* is then used to execute *grpProcess*.

This is a **History Data Group** that returns records between the start and end of lot. This is used in a sub report to display process data recorded during the lot.

Template

The *Template* sheet is where the sub reports are laid out.

The sub report for *grpBatch* is set up to show the *Batch Name* along with the *start* and *end* time of the batch.

Grou	Connector	Source	Target	Place
1	Variables	{grpBatch:StartDateTime}	Template!$B$4	Direct
1	Variables	{grpBatch:EndDateTime}	Template!$D$4	Direct
1	Variables	{grpBatch:Batch}	Template!$C$2	Direct

These are set up as **Variable** connections configured for **Group** *1*.

The stacking is done using the **Copy Sub Report Range** management function that copies the sub report range from the template sheet to the **Report** sheet based on a named cell (*lastcell*) that is then redefined after the range is pasted.

Note that the **Group** for this management is *101*.

Rule	Group Name	Connect ID	Va
Iterate	grpBatch	101,1	

In the **Stack Wizard**, for *grpBatch* the **Connect ID** is set to *101, 1*. This means that for this sub report, first group *101* is executed (to copy the sub report) and then group *1* is executed to populate the sub report with the batch name, start and end time.

The same is true for the other 2 sub reports defined for this template.

On-Demand Report

When run on-demand, the user is presented with the list of batches from the *grpBatchList* group.
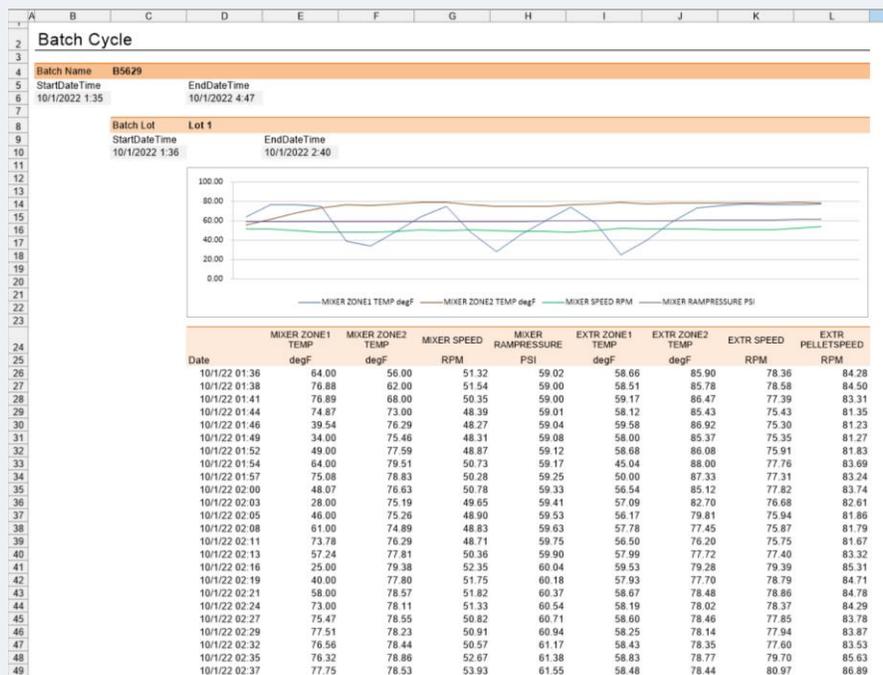
The generated report displays the batch sub report, the batch lot sub report and the lot details from the historian all stacked together.

Historical Data from Database Rows

Data Group Time Period

When configuring a **History Data Group**, under the **Time Period** tab the Period Type can be set to Database. When set, the **Source** provides a list of **Database Data Groups** configured in the project. The **Start** and **End** can be configured to timestamp columns selected in the group.

In the example above the **AlarmTimes** group has *DateAndTime* and *DateAndTimeOut* columns for the timestamp of the start and end of each alarm. These columns contain both the date and time, so the **Date includes Time** option is checked. If there were separate columns for date and time, this would be unchecked, and each column would be specified for the **Date** and **Time** settings.

In some cases, the group may have a column for the **Start** but not one for **End**. In this case, for **End**, **Type** can be set to *Duration*. In this case, the **End** is defined by adding the *Duration* specified to the **Start**.

When this **History Data Group** is executed, the **Database Data Group** set for the **Source** is executed and for each record returned, the History Data Group is executed using the column values for that record. If *10* records are returned from the database group, the history group is run *10* times.

Data Connection

When the **History Data Group** that uses **Database** as the **Time Period** is configured as a **Data Connection**, it is important to set the **Placement Type** to *Append* or one of the *Insert* options. This is because multiple rows of data will be returned as separate updates based on each record of the database group.
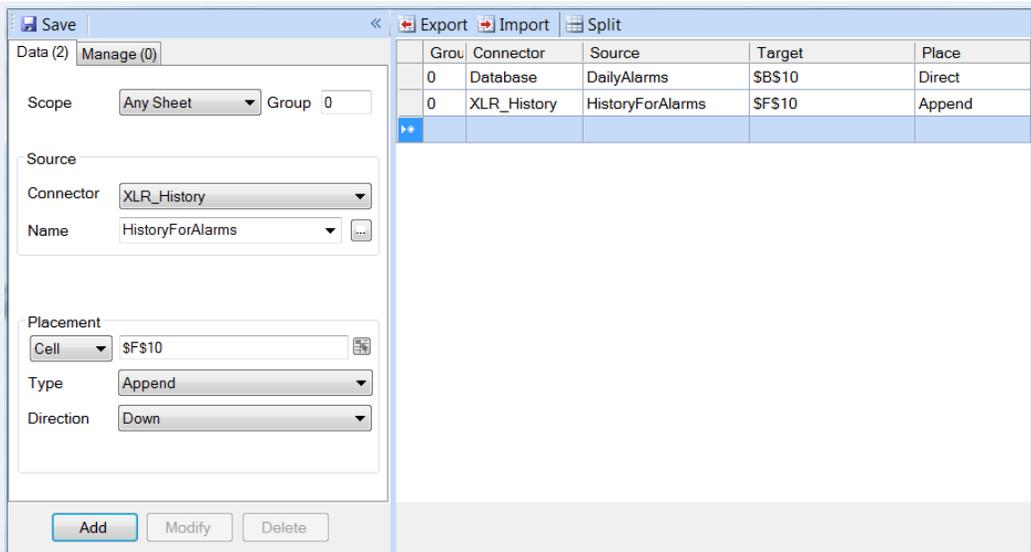
Typical Scenario

Consider the following scenario: A daily report is required to show all the *HI* alarms that occurred over the day and for each alarm in the mixer, the average of each mixer tag recorded in the historian during the alarm.

Database Data Group

The **Database Data Group** is used both to bring the daily alarm data into the report and also to use as the **Time Period** in the **History Data Group**.

Under the **Columns** tab, all the columns to display for the daily alarms are selected. *DateAndTime* is the start timestamp for the alarm and *DateAndTimeOut* is the end timestamp for the alarm.

Under the **Filters** tab there are **Filter Conditions** set to restrict the records to those from the current day, only for tags that start with *MIXER* and where the alarm is *HI* or *HIHI*.

Under the **Order** tab, the **Order By** is configured so alarm records for the day are returned from oldest to newest.

History Data Group

In the **History Data Group**, under the **Columns** tab, the *MIXER* tags logged to the historian are set up with the **Calculation** as *average*.

Also, the **Output Options** are set to *None*. This is because the DailyAlarms database group (the group configured in the previous section) brings in the timestamps for each alarm. This group will be configured adjacent to the database group so there is no reason to double up on the timestamps.

Under the **Time Period** tab the **Period** is set up the *DailyAlarms* database group using the *DateAndTime* column for **Start** and the *DateAndTimeOut* column for **End**.

Since the requirement is to report on the average during the entire duration of the alarm and each alarm can have a different duration, the **Interval** is set to **Count** as *1* to produce a single average over the entire duration.

Data Connections

In the template, the **Data Connections** are set up to bring in both the alarm data and the historical data.

Schedule
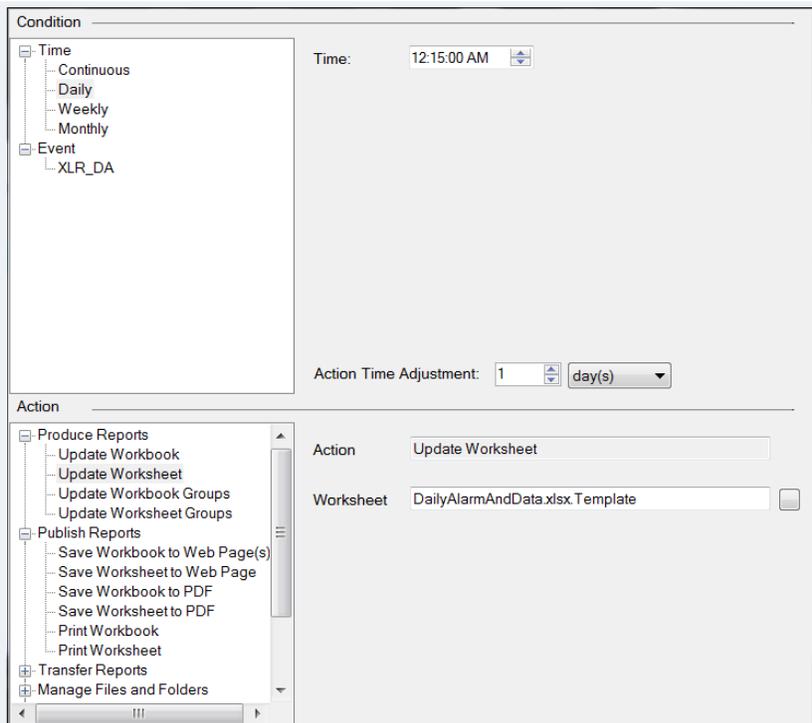
To generate this report, a single *Update Worksheet* **Action** scheduled to run every day.

Because the data is coming from a database and historian, it is a good idea to trigger this report on the next day to ensure all the data has been collected and is available to retrieve. That's why the **Time** is set to *12:15:00 AM*.

However, everything in the report template is configured to work off the current day. To account for this, the **Action Time Adjustment** is set to *1 day* to adjust the current date by 1 day when generating the report.

	A	B	C	D	E	F	G	H	I
7									
8							MIXER		
9		DateAndTime	DateAndTimeOut	TagName	ThreshLabl	ZONE1 TEMP	ZONE2 TEMP	SPEED	RAMPRESSURE
10		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_ZONE2_TEMP	HI	62.50	64.76	45.21	32.12
11		03/17/2020 07:15:00	03/17/2020 07:35:00	MIXER_ZONE1_TEMP	HIHI	59.50	67.76	49.21	33.12
12		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_SPEED	HI	55.50	74.76	54.21	36.12
13		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_ZONE2_TEMP	HI	51.50	77.76	56.21	37.12
14		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_ZONE2_TEMP	HI	44.50	83.76	59.21	40.12
15		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_ZONE1_TEMP	HI	39.50	87.76	64.21	42.12
16		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_SPEED	HIHI	33.50	93.76	69.21	45.12
17		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_ZONE2_TEMP	HI	27.50	98.76	72.21	47.12
18		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_ZONE2_TEMP	HIHI	21.50	103.76	76.21	49.12
19		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_ZONE1_TEMP	HI	16.50	108.76	79.21	52.12
20		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_SPEED	HI	13.50	112.76	84.21	55.12
21		03/17/2020 03:45:00	03/17/2020 04:15:00	MIXER_ZONE2_TEMP	HI	8.50	113.76	89.21	58.12

The resultant report displays all the information from both alarms and history.

Combine Data Into a Single Data Table

In many reports, data from different data sources are written in separate ranges or even separate worksheets. However, sometimes it is nice to be able to view data from disparate data sources in a single view. **XLReporter** has a management function, **Weave Into Range**, that can do just that.

Weave Into Range

The **Weave Into Range** management function is available under the **Category** *Presentation*.

The **Apply To** settings define the range which data will be woven into.

The **Collection** settings define the range of data to weave. This should be set to the top row of where the collection data appears. The bottom row of the **Collection** is determined by finding the first empty cell in the leftmost column of the range. In the example above, the **Collection** range would be determined by finding the first empty cell beneath $J$6.

Rows are woven into the **Apply To** range from the **Collection** range based on the leftmost column of each range. Typically, these are timestamps from the data source.

The **Weave Mode** defines how the **Collection** rows are woven into the **Apply To Range**.

The *Column Insert* mode means that every row in the **Collection** is inserted as its own new row.

The *Row Append* mode means that if the left column of the **Collection** range matches the left column of the **Apply To** range, the remaining columns of the Collection range are appended on the same row (to the right of the rightmost column of the **Apply To** range). Otherwise, a new row is inserted.

To illustrate, consider the following data:

	A	B	C	D	E	F	G
1							
2							
3							
4		0:00	1			1:00	100
5		1:00	2			2:15	101
6		2:00	3			4:00	102
7		3:00	4			7:30	103
8		4:00	5			11:00	104
9		5:00	6			11:45	105
10		6:00	7				
11		7:00	8				
12		8:00	9				
13		9:00	10				
14		10:00	11				
15		11:00	12				
16		12:00	13				

The **Apply To** range is $B$4:$C$16. The **Collection** range is $F$4:$G$9. If **Weave Mode** is *Column Insert*, the results are:

	A	B	C	D
1				
2				
3				
4		0:00	1	
5		1:00	2	
6		1:00		100
7		2:00	3	
8		2:15		101
9		3:00	4	
10		4:00	5	
11		4:00		102
12		5:00	6	
13		6:00	7	
14		7:00	8	
15		7:30		103
16		8:00	9	
17		9:00	10	
18		10:00	11	
19		11:00	12	
20		11:00		104
21		11:45		105
22		12:00	13	

For *Row Append* the results are:

	A	B	C	D
1				
2				
3				
4		0:00	1	
5		1:00	2	100
6		2:00	3	
7		2:15		101
8		3:00	4	
9		4:00	5	102
10		5:00	6	
11		6:00	7	
12		7:00	8	
13		7:30		103
14		8:00	9	
15		9:00	10	
16		10:00	11	
17		11:00	12	104
18		11:45		105
19		12:00	13	

The **Use Collection Format** option applies the formatting along with the values from the **Collection** range.

The **Add Grouping** option adds crosshairs to every row (**Weave Mode** = *Column Insert*) or column (**Weave Mode** = *Row Append*) from the **Collection** range.

If grouping is enabled, the **Initial State** defines if all groups are initially *Expanded* or *Collapsed*.

After the **Collection** is woven, it can be cleared from the report by setting **Clear Collection** to *All* or *All (extended)*. If the **Collection** range has headings on the worksheet use the *All (extended)* setting to clear the headings as well.

Typical Scenario

Consider the following scenario: A report is required to show 15 minute samples of historical data over the day along with all the alarms recorded for that day. The report should show both the historical data and alarm data in a single table of data.

Template Design

	Group	Connector	Source	Target	Place
	0	XLR_History	Alarms with Process Values	$B$6	Direct
▶	0	Access DB	Alarms with Process Values	$O$6	Direct
✳					

The **Data Connections** for the template are configured to retrieve the historical data and the alarm data onto the worksheet at cells $B$6 and $O$6 respectively.

To combine the historical and alarm data the **Weave Into Range** management connection is configured to weave the alarm data into the historical data, inserting new rows for each alarm row. The formatting of the alarm data will be applied as will grouping.

On the template worksheet the alarm data is formatted red to distinguish from the historical data when woven together.

Report Generation

Without the **Weave Into Range**, the report would look like this:

With the **Weave Into Range** applied, the report now looks like:

The group buttons on the left allow the user to expand and collapse to show or hide the alarm data.

Use Data From One Source to Retrieve Data From Another

In the first two parts of this document, it showed using the results of a database data group to drive a stacked report and to use the results of a database group as the time period of a history data group. However, there are scenarios where the data to drive the history data group is not stored in a database but rather in a historian (either the same historian or a different historian). The time period of a history data group does not have an option for this.

However, this can be achieved using the **By Row Iteration** management function.

By Row Iteration

The **By Row Iteration** management function can be used to drive additional data into a report or even to generate new reports. For this document, we are focusing in on using this function to feed time settings into a history data group to get additional data for the report.
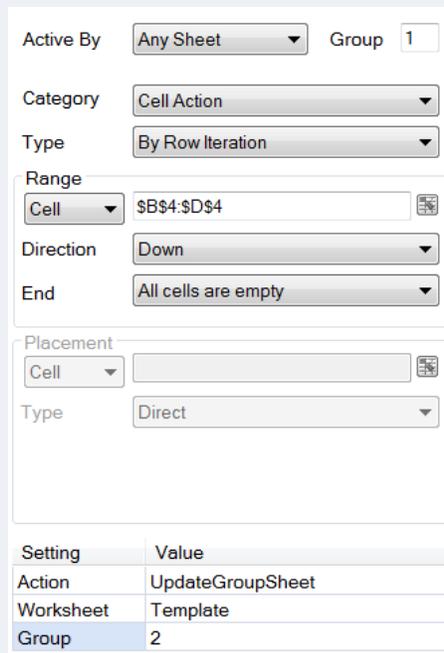
This management function is available under the **Cell Action** category.

The **Range** defines the rows of data that will be iterated over. This range should consist of a row of headings and then one or more rows of values. The headings each column should be variable names. For every row of data, the value in each column is set to the variable in that row.

Consider the following:

When executed, this will iterate three times. The first time, the variables *Start Date*, *End Date* and *Tag 1* will be set with the data in row 5, the second time the data from row 6 and the third with the data from row 7.

As part of the execution, after the variables are set, an **Action** can be triggered to run to update either another worksheet or a specific group within the template.

Note, **Action** can be set to a cell (or range of cells) that specify **Actions** that can trigger other reports to execute. This will not be discussed in this document.

Typical Scenario

Consider the following scenario: A daily report is required that needs to show the time of the maximum speed of the mixer every hour of the day. The report must also show the values of the other mixer parameters at that time recorded in the historian.

Template Design

There are two data connections configured for this template.

The first is configured to retrieve the maximum time of the mixer speed. The maximum value is also retrieved.

This data group is set up to return hourly values over the day.

For the connection, the **Group** is set to *1*. This is done to control the order at which connections are executed. It is also important to note that the cells with the 24-hourly maximum times is formatted to show the date and time all the way to second, e.g., *mm/dd/yyyy HH:mm:ss*.

For the second data group, the other mixer tags are selected.

The **Time Period** is configured for the **Start** as a *Variable*. This *Variable* will be set using the **By Cell Iteration** management connection. The **Interval** ensures that only *1* row is returned from the group.

This data group is configured for **Group** 2. **Group** 2 connections will be used using the **By Cell Iteration** management connection.

This connection is also set up for **Placement** as *Append* so that each row of data returned is appended to the report going down.

5					
6	Speed		Values		
7	Max Time	Max Value	Zone 1 Temp	Zone 2 Temp	Ramp Pressure
8					

On the template worksheet, the heading in cell *$B$7* is *Max Time* which is the name of the variable used in the second data group.

For the **By Row Iteration** management connection, only values in the B column need to be iterated over to set the **Max Time** variable that is configured in the second data group.

For each row of data starting in *$B$7* the **Group** 2 connections on the *Template* **Worksheet** are updated.

Schedule

The schedule to generate this report is a single *Update Worksheet Groups* **Action** to update the **Group** *1* connections in the template.

This will trigger *Daily* at *12:15:00 AM*. The **Action Time Adjustment** is set to *1 day* to generate the report for the previous day.

What will happen when this action triggers is that it will generate a new report and add the hourly maximum times and values for the mixer speed (24 rows). Then **the By Row Iteration** management connection executes. For each of the 24 rows, the *Max Time* variable is set with the time value in the cell in the *B* column and then the group 2 connection is executed.