

Make Reports Convey Their Content

Overview

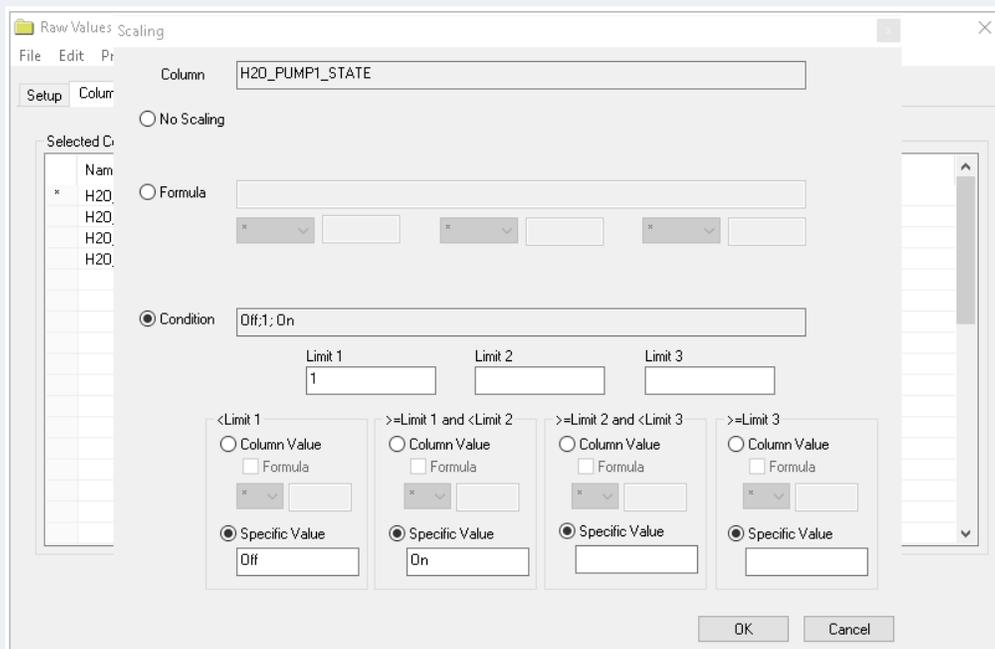
Sometimes, SCADA systems manage data in formats that aren't easily interpreted in the context of a report. For example, a machine's state can be indicated by a numeric 0 or 1, but in the report, a better interpretation would be "On" or "Off". Or, the numeric value may be an integer that represents one of multiple stages of a cycle. **XLReporter** offers numerous functions that transform data from formats used by machines and software into formats easily readable by people, or compatible with compliance or operational requirements.

Feature: Scaling

Scaling allows the value returned from a **Data Group** to be modified before it is added into a report, based on either arithmetic formulas, value ranges, or a combination of the two.

This feature is available in a **Data Group** and applicable to every real time, historical and analytic data.

Open a **Data Group** and from the **Columns** tab select the **Scaling** column for a row with a **Name** specified.



Scaling can use either a **Formula**, or a **Condition**.

With **Formula** scaling, an arithmetic operation is performed on the value before it is returned to the report, e.g., turn watts to kilowatts or Fahrenheit to centigrade.

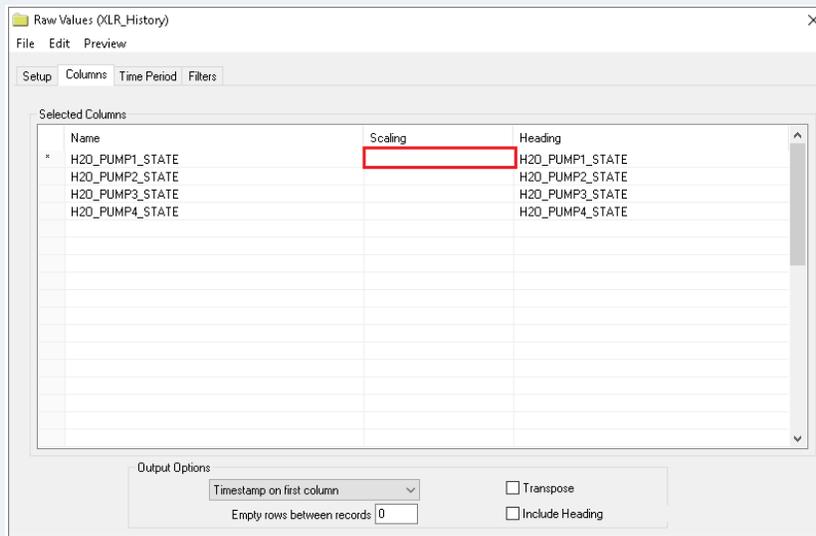
With **Condition** scaling, the process value is modified based on 4 conditions, each applied based on the value compared to 3 specified **Limits**.

For **Column Value** with a set **Formula**, the value is conditioned based on the formula otherwise it is not altered in value.

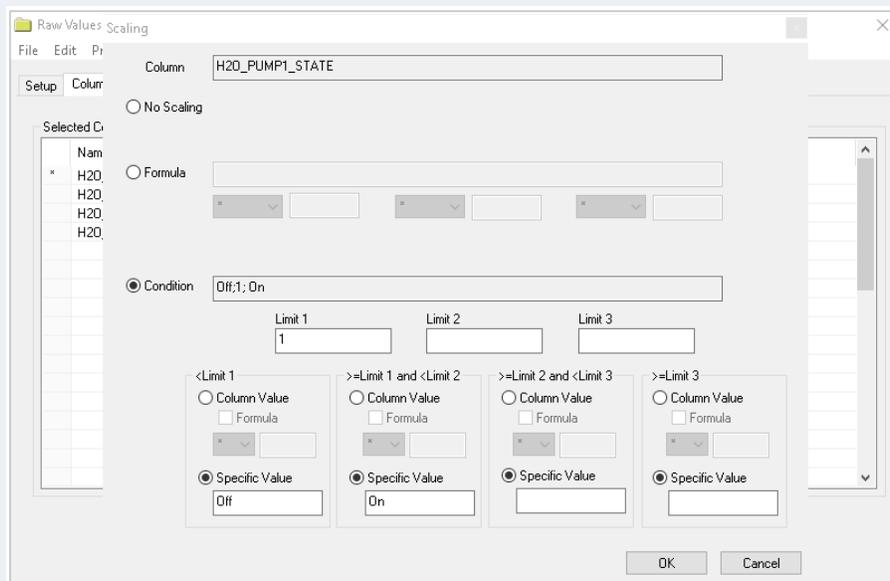
For **Specific Value**, the value is conditioned to the text entered. In the image above for example, the tag value is displayed as "Off" or "On" based on whether the value is < 1 or ≥ 1 . For numeric values, more conditions can be specified.

Example: Convert Machine Running State to Text Value

Most control systems will express the state of a machine as a Boolean or bit value for the purpose of driving ladder logic in a controller. In a report, this could be displayed as values like “On” or “Off” using **Data Group Scaling**.



Open the **Scaling** configuration for the tag and use the **Condition** option.



Set **Limit** to *1*, and assign **Specific Values** for **<Limit 1** and **>=Limit 1 and <Limit 2**. Then, assign the text which replaces the raw values.

Since the tag *H2O_PUMP1_STATE* has scaling applied, its Boolean value is replaced by the text *On* for a value of 1 as follows:

Date	H2O_PUMP1_STATE	H2O_PUMP2_STATE	H2O_PUMP3_STATE	H2O_PUMP4_STATE
3/30/2020	On	1	1	1
3/30/2020 12:01:00 AM	On	0	0	0
3/30/2020 12:02:00 AM	On	1	1	1
3/30/2020 12:03:00 AM	On	0	0	0
3/30/2020 12:04:00 AM	On	1	1	1
3/30/2020 12:05:00 AM	On	0	0	0
3/30/2020 12:06:00 AM	On	1	1	1
3/30/2020 12:07:00 AM	On	0	0	0
3/30/2020 12:08:00 AM	On	1	1	1

Feature: Workbook Formulas

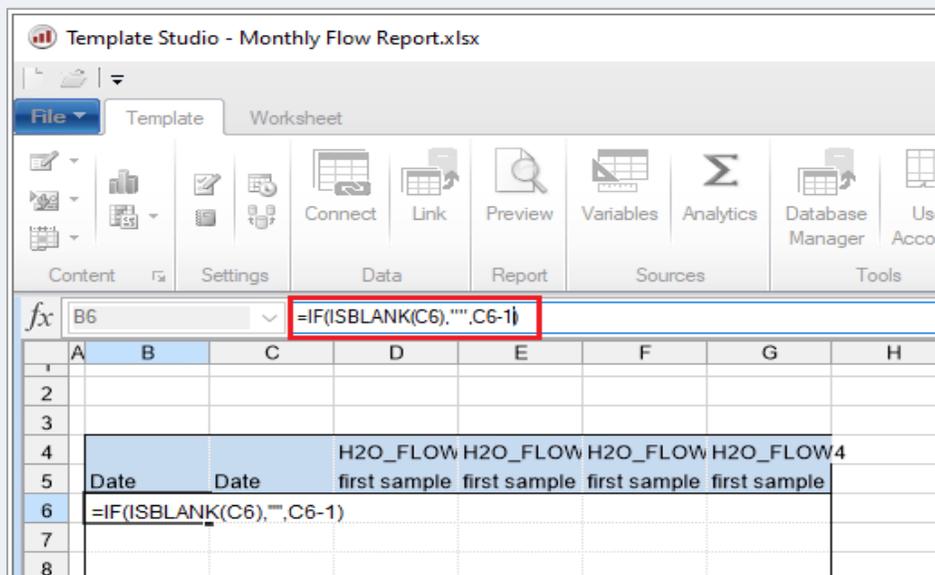
Workbook formulas are a powerful feature that provide workbook content to be adjusted using string manipulation, logical and arithmetic operations, and much more.

Example: Adjust Values with a Formula

Commonly in the water treatment industry, data is accumulated or “totalized” in the PLC and then transferred at the beginning of the next day from the current-value tag to a “previous-day” register. The register is then logged to the historian.

The problem this creates for the report is that, for any particular day, the value from the historian is not correct since it represents the previous day. With workbook formula, this can be easily rectified.

In the following, suppose the historical data is placed in columns C to G starting at row 6 where the C column receives the date.



In the B column, add the formula above for 31 rows (worst case). The formula subtracts 1 day from the C column e.g., C6-1. On completion, hide the C column.

A Closer Look

The conditional IF statement works with 3 parts as follows:

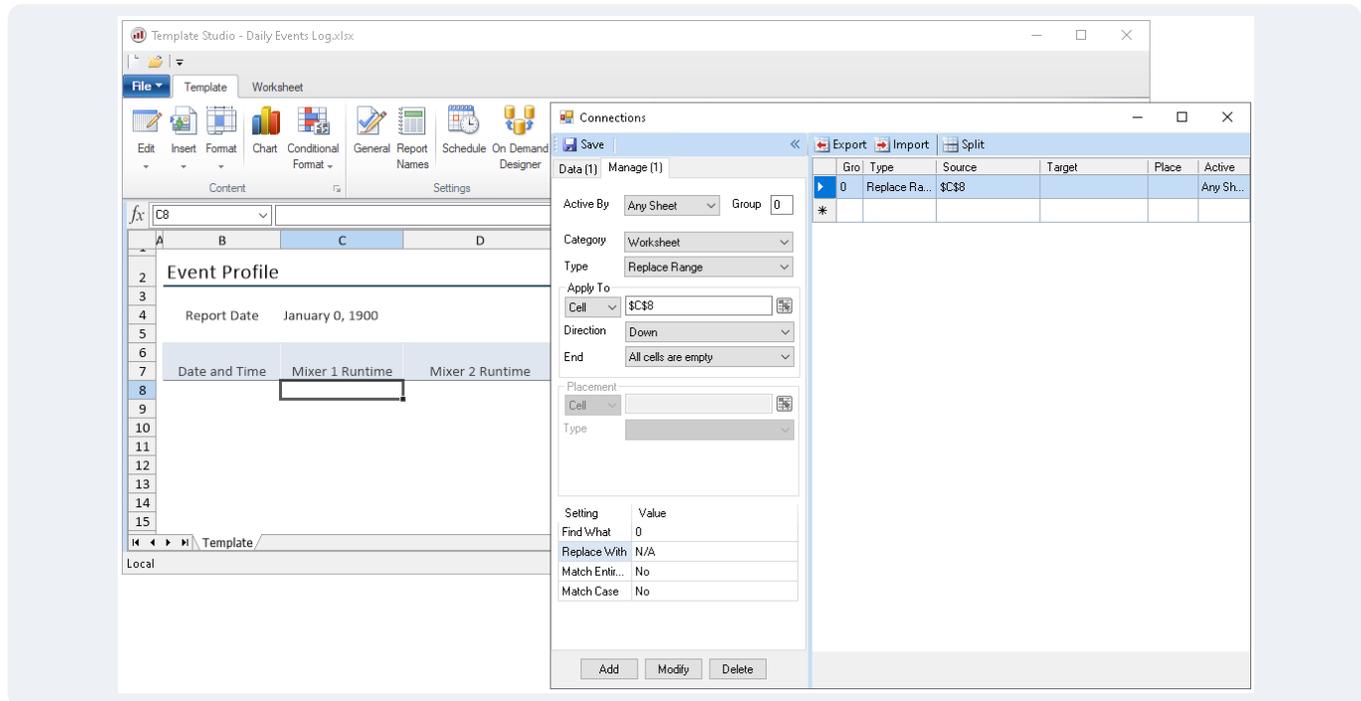
IF (condition, result for true, result for false)

In our example the condition tests if the C column is blank i.e., C6="", if it is then a blank is displayed in the report otherwise the formula result is displayed. A conditional is used so that months with less than 31 days show blank on days that are not used.

Feature: Replace Range

XLReporter provides a wealth of **Management** functions that are applied to report data. They are particularly useful because they easily handle dynamic lists.

One such function is **Replace Range**. This is an automated version of the *find and replace* feature of an Excel workbook with the difference that it is able to operate on dynamic cell ranges.



The range of cells the function will **Apply To** is determined by starting at **Cell** scanning in the **Direction** until the **End** condition. Once the range is determined, the **Find What** and **Replace With** settings indicate the value to find in the range and what to replace it with.

Example: Show No Data as Non-Applicable Data

A report is being updated with a new row of live data from various pieces of equipment each time a test cycle is complete. When a particular piece of equipment is not used, the process value is shown as 0 which is misleading and so it will be replaced by N/A (not applicable).

On each day, the number of tests differ which makes the number of rows dynamic in number. Standard workbook functions are not equipped to deal with dynamic lists so the **Replace Range** function is used because it is able to handle such ranges.

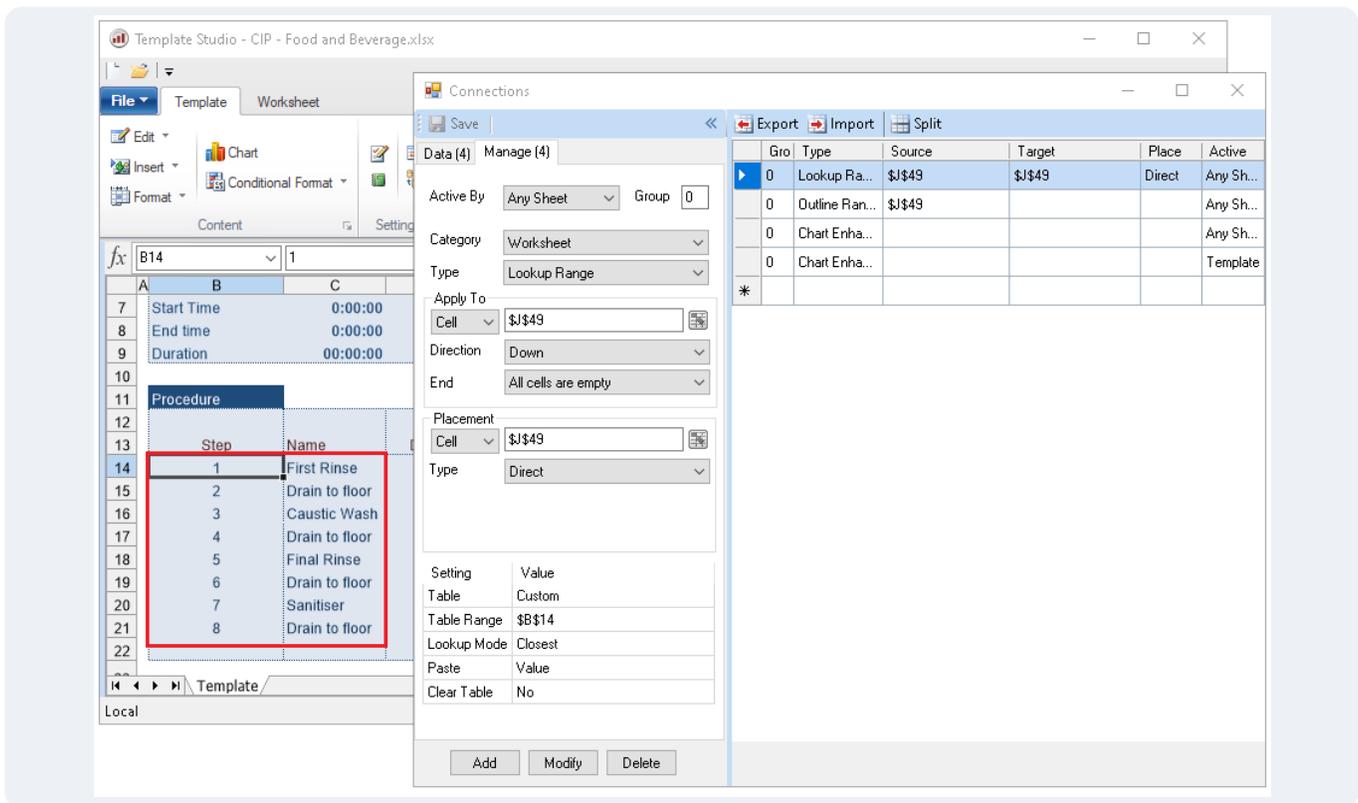
Referring to the above image:

- **Apply To cell** is \$C\$8.
- **Direction** is *Down*
- **End** is *All Cells are Empty*.
- **Find What** is 0.
- **Replace With** is *N/A*.

Feature: Lookup Range

The **Lookup Range** function takes values in the report and replaces them based on a lookup table. The lookup table contains 2 columns where the first column(source) is where the value is matched and the second column (target) is the replacement value.

This function is an automated version of the LOOKUP functions of Excel with the difference that it is able to operate on dynamic cell ranges.



The range of cells the function will **Apply To** is determined by starting at **Cell** scanning in the **Direction** until the **End** condition. The values in the range are compared to the first column of the lookup table for a match and the corresponding value in the second column is then placed in the **Placement Cell**.

The lookup table definition has several settings. **Table** indicates the type of lookup table. There are some built-in tables available like *Yes/No*, *True/False* and *On/Off*. If these do not apply, set to *Custom* to define a custom lookup table.

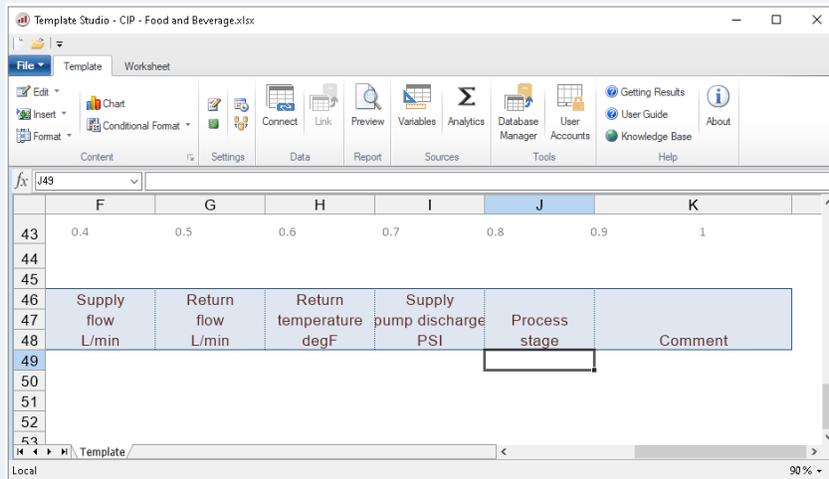
The lookup table is usually defined on the worksheet given by the **Table Range** which indicates the top-left corner of the lookup table (**Table** set to *Custom*).

The **Lookup Mode** determines the comparison method used when matching the value to the first column. There are several options such as to use the *Closest* value, the *Exact* value, the closest *Larger* value, etc.

When the lookup is completed, there may be no need to leave it on the worksheet in which case set **Clear Table** is set to Yes.

Example: CIP Phase Steps to Textual Description

In a CIP report, the process stage is read as a numerical value from the data source (live or historical). The **Lookup Range** function is used to convert the number to a textual description.



The **Lookup Range** function works by operating on a column of values and replacing the row values by using a key:/value match stored in a table in another part of the worksheet.

When the report is complete, the **Lookup** management function will take the values starting at $J49$ and change them to a textual description using the lookup table at cell $B14$.

Referring to the images above:

- **Apply To** cell is in $J49$.
- **Direction** is Down.
- **End** is All Cells are Empty.
- **Table** is Custom.
- **Table Range** is $B14$.
- **Clear** is No (because the range is part of the report).

A Closer Look

The template in this example is in the *XLR_Demo* project which can be opened in the **Template Studio**. In addition to **Lookup Range**, the **Outline Range** function and the **Chart Enhancement** are used.

The **Outline Range** function is used to clear repetitive values from the *J* column after the **Lookup Range** is processed. This makes the transition of the CIP states much clearer.

Time	Elapsed (secs)	Supply conductivity mS/cm	Return conductivity mS/cm	Supply flow L/min	Return flow L/min	Return temperature degF	Supply pump discharge PSI	Process stage
8:00:00	0		0	150		78.0		first rinse
8:01:00	60		1	150		92.0		first rinse
8:02:00	120		4	150		101.0		first rinse
8:03:00	180		3	150		106.0		first rinse
8:04:00	240		1	150		109.0		first rinse
8:05:00	300		0	150		110.0		first rinse
8:06:00	360		0	150		110.0		first rinse
8:07:00	420		0	150		109.0		first rinse
8:08:00	480		0	150		109.0		first rinse
8:09:00	540		0	150		109.0		first rinse
8:10:00	600		0	150		110.0		first rinse
8:11:00	660		0	150		110.0		first rinse
8:12:00	720		0	150		110.0		first rinse
8:13:00	780		0	150		110.0		first rinse
8:14:00	840					109.0		drain
8:15:00	900					108.0		drain
8:16:00	960					106.0		drain
8:17:00	1020					103.0		drain
8:18:00	1080		10	150		102.0		caustic wash
8:19:00	1140		10	150		105.0		caustic wash
8:20:00	1200		11	150		110.0		caustic wash

The **Chart Enhancement** function is used to enhance the chart by displaying the step transition as labels on a chart.

Procedure					
Step	Name	Duration	Return Temp deg F	Flow Rate L/min	Comment
1	First Rinse	360	110	150	From main tank
2	Drain to floor	120			
3	Caustic Wash	1200	145	150	Recirculate through equipment and back into tank
4	Drain to floor	120			
5	Final Rinse	360	70	150	From main tank
6	Drain to floor	120			
7	Sanitiser	540	70	150	Recirculate through equipment and back into tank
8	Drain to floor	120			

