Flexible Reports with Data Placement

Overview

Report content is driven by Data Connections. When defining a data connection for a template, two things need to be considered:

- What data source will provide content.
- Where will the content be written in the report?

This document focuses on where the content will be written and all the options that are offered by **XLReporter**.

Report as you Run

To understand why data placement is important, it is first important to understand "Report as you Run" technology. This is unique to **XLReporter.**

With **XLReporter**, a report is not a static entity that is generated in one pass. Rather, a report is dynamic and can be built up over time incrementally from many different data sources.

Let's take a simple example of a plant that only has a PLC and a PC, and needs a report with hourly values over the day from the PLC. There is no historian or database to hold this data; it needs to come straight from the PLC into the report every hour over the day.

Most reporting packages would require that data to be collected into some data repository and then, at the end of the day, generate the report with the hourly values collected. With **XLReporter**, that data can be retrieved from the PLC every hour and placed onto the report in the row designated for that hour throughout the entire day. At any time during the day the report can be viewed to see the data up through the previous hour.

Data Retrieval

Before analyzing data placement, let's first identify how data is retrieved for reports when configured as data connections.
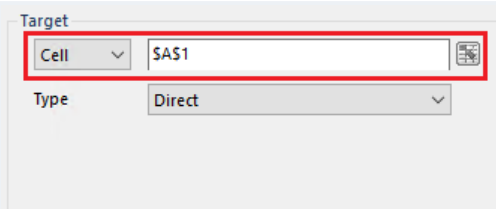
There are single value connection types like **Variables**, **Expressions** and **Real Time** tags. The data from these are always a single value which is written to a single cell.

Then there are table connections types like **Data Groups** from **Real Time**, **Historical** and **Database** sources. These connections return a table of data that can be both multiple columns and multiple rows.

Data Placement

For a data connection, data placement is defined in the **Target** section for the connection.

Cell
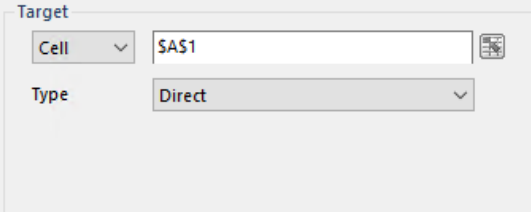

When configuring target for a data connection, the **Cell** must be defined as a single cell on the worksheet. The simplest definition of **Cell** is that it is where the data is written to on the worksheet. For table connection types, **Cell** is the upper-left corner where data is written.

For example, if the table data connection returns 4 columns and 8 rows and **Cell** is set to *$A$1*, data is written to cell range *$A$1:$D$8*.
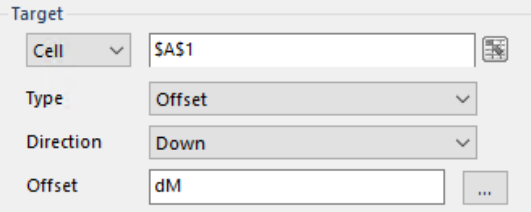
Target Placement Types

Direct

A data connection configured as *Direct* is always written to the **Cell** configured. This is done regardless if the cell already contains data.
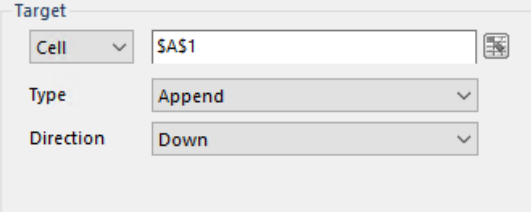
Offset

A data connection configured as *Offset* is written to the sheet offset from the **Cell** configured based on the **Direction** and **Offset** specified.

As a simple example, consider a single value connection with **Cell** set to *$A$1*, **Direction** set as *Down* and **Offset** set as 2. When updated, the value for the connection is written to cell *$A$3*; two rows down from cell *$A$1*.

In practice, **Offset** is not a static number but rather an **XLReporter** Name Type which is a keyword for an interval of time like *hD* for the hour of the day (0-23) or *dM* for the zero based day of the month (0-30).
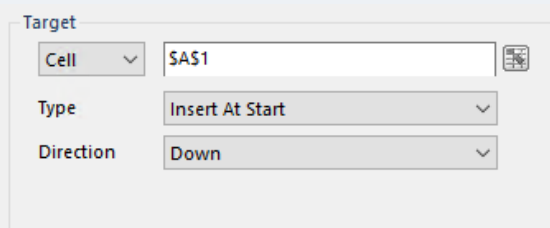
Append

A data connection configured as *Append* is written to the sheet to the next empty cell starting at **Cell** configured based on the **Direction** specified.

For example, a single value connection configured to **Cell** *$A$1* with **Direction** as *Down*. Whenever this is updated, the value is written to the next empty cell starting at $A$1 going down.

This gets a bit more complex with a table connection. Let's take a table connection with 4 columns and 8 rows with **Cell** as *$A$1*. If **Direction** is *Down*, the values are written to the first row where all 4 columns (A-D) are empty. If **Direction** is *Across*, the values are written to the first column where all 8 rows (1-8) are empty.

Insert At Start/Insert At Start (Full)

A data connection configured as *Insert At Start* or *Insert At Start (Full)* is written to the sheet by first having the appropriate amount of rows (**Direction** = *Down*) or columns (**Direction** = *Across*) inserted to the worksheet starting at the **Cell** specified and then the data is written into the inserted range.

For example, consider a table data connection that returns 4 columns and 8 rows with **Cell** set to *$A$1*. If **Type** is *Insert At Start* and **Direction** is *Down*, when updated, if *$A$1:$D$1* is empty, a range 7 rows by 4 columns is inserted at cell *$A$2*.
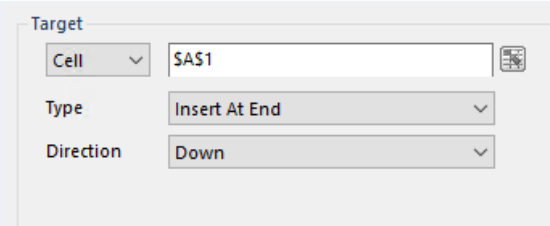
If *$A$1:$D$1* is not empty, a range of 8 rows by 4 columns is inserted at cell *$A$1*. Anything beneath this range in columns *A:D* are pushed downwards. Then, the data is written into the range *$A$1:$D$8*.

If the above example was configured with **Type** as *Insert At Start (Full)*, the same thing would happen, except instead of inserting a range 4 columns wide, 7 or 8 entire rows would be inserted before the data is written.

Insert At End/Insert At End (Full)

A data connection configured as *Insert At End* or *Insert At End (Full)* is written to the sheet by first finding the next empty row (**Direction** = *Down*) or column (**Direction** = *Across*), then the appropriate amount of rows (**Direction** = *Down*) or columns (**Direction** = *Across*) is inserted to the worksheet at the empty row or column and finally the data is written into the inserted range.

For example, consider a table data connection that returns 4 columns and 8 rows with **Cell** set to *$A$1*. If **Type** is *Insert At End* and **Direction** is *Down*, when updated, if *$A$1:$D$1* is empty, a range 7 rows by 4 columns is inserted at cell *$A$2*.

If *$A$1:$D$1* is not empty, the next empty row between *A:D* is found and a range 8 rows by 4 columns is inserted at the *A* column of that row. Anything beneath this range in columns *A:D* are pushed downwards. Then, the data is written into the inserted range. If the above example was configured with **Type** as *Insert At End (Full)*, the same thing would happen, except instead of inserting a range 4 columns wide, 7 or 8 entire rows would be inserted before the data is written.

Excel and Inserted Rows, Columns and Ranges

When ranges, rows or columns are inserted into a worksheet, it influences other items within the worksheet. This can be especially useful when the amount of data returned from a data connection varies.

Formatting

When rows or columns are inserted into a worksheet, the formatting of row above or column to the left of the range (depending on which way the cells are shifted) being inserted is applied to the inserted range.

This includes number formatting, borders, fonts, background colors and even extends to any conditional formatting that may be applied.

Charts

If the range for any series of a chart spans over a range in the worksheet that is inserted into, the series range is automatically resized with the number or rows or columns inserted.

For example, if the series of a chart is set up initially as $A$1:$A$2 and 8 rows are inserted into $A$2, the series becomes $A$1:$A$10.

Formulas

Formulas behave the same way as chart series in that if the cell reference in the formula spans a range where rows or columns are inserted, the cell reference range is automatically resized with the number of rows or columns inserted.

For example, if a cell contains the initial formula =SUM(A1:A2) and 8 rows are inserted into $A$2, the formula becomes =SUM(A1:A10).

Scenarios

To understand the flexibility available for data placements, let's consider a series of reporting scenarios and what settings work.

Scheduled Report: Daily Report with Real Time Values at the End of the Day

In this scenario, real time data is written to the daily report one time at the end of the day.

Therefore, the **Target** of the connection should be *Direct*.

Scheduled Report: Daily Report with Real Time Values Every 15 Minutes over the Day

In this scenario, real time data is written to the daily report every 15 minutes throughout the course of the day, down the column starting at midnight. The net result is a report with 96 rows of data.

There are a few options for **Target** in this scenario. The *Append* or *Insert At End* settings would work. However, if the machine is switched off during the middle of the day and then restarted later, there would be no gap in the report to account for these missing updates.

Time	MIXER_ ZONE1_ TEMP	MIXER_ ZONE2_ TEMP	MIXER_ SPEED	MIXER_ RAMPRE SSURE	BATCH_ TANKTE MP	BATCH_ MIXPRE SSURE	BATCH_ ZONE1	BATCH_ ZONE2
12:00 AM	54.28	75.59	32.50	68.53	57.28	58.82	46.67	35.00
12:15 AM	76.73	76.57	32.60	70.79	40.43	64.46	53.33	40.00
12:30 AM	76.96	77.38	32.76	73.14	53.17	70.34	46.67	35.00
12:45 AM	77.35	77.68	36.62	75.53	57.40	76.33	53.33	40.00
1:00 AM	77.59	77.95	36.45	77.91	57.52	82.28	46.67	35.00
1:15 AM	77.76	78.33	35.83	80.23	57.75	88.07	53.33	40.00
1:30 AM	73.96	78.36	33.76	82.43	57.63	93.57	46.67	35.00
1:45 AM	54.15	80.16	31.02	84.46	37.12	98.65	53.33	40.00
2:00 AM	54.10	81.78	31.93	86.28	57.95	103.20	46.67	35.00
2:15 AM	58.33	82.10	35.01	87.84	58.33	107.11	53.33	40.00
2:30 AM	76.95	82.41	36.63	89.12	58.50	110.31	46.67	35.00
4:30 AM	55.13	79.57	37.33	87.10	58.03	105.24	46.67	35.00
4:45 AM	76.57	79.80	39.57	85.40	58.10	101.00	53.33	40.00
5:00 AM	69.31	80.37	41.82	83.47	58.52	96.17	46.67	35.00
5:15 AM	54.68	77.03	40.86	81.35	56.72	90.87	53.33	40.00
5:30 AM	50.95	76.58	39.35	79.08	58.33	85.21	46.67	35.00
5:45 AM	64.86	77.06	36.79	76.73	58.66	79.32	53.33	40.00
6:00 AM	76.29	77.38	40.04	74.33	58.82	73.34	46.67	35.00
6:15 AM	76.54	77.73	39.15	71.96	59.03	67.39	53.33	40.00
6:30 AM	76.81	77.94	39.27	69.64	59.09	61.61	46.67	35.00
6:45 AM	77.16	78.27	42.75	67.45	36.55	56.14	53.33	40.00
7:00 AM	77.44	77.55	39.87	65.43	56.64	51.08	46.67	35.00
7:15 AM	77.74	77.31	38.23	63.63	36.69	46.57	53.33	40.00

To account for this, the **Target** should be set to *Offset* with the **Offset** set to *mD/15*. Now, if the machine is switched off and restarted later, there is a clear gap in the report, so the missing time is obvious.

Time	MIXER_ZONE1_TEMP	MIXER_ZONE2_TEMP	MIXER_SPEED	MIXER_RAMPRESSURE	BATCH_TANKTEMP	BATCH_MIXPRESSURE	BATCH_ZONE1	BATCH_ZONE2
12:00 AM	54.28	75.59	32.50	68.53	57.28	58.82	46.67	35.00
12:15 AM	76.73	76.57	32.60	70.79	40.43	64.46	53.33	40.00
12:30 AM	76.96	77.38	32.76	73.14	53.17	70.34	46.67	35.00
12:45 AM	77.35	77.68	36.62	75.53	57.40	76.33	53.33	40.00
1:00 AM	77.59	77.95	36.45	77.91	57.52	82.28	46.67	35.00
1:15 AM	77.76	78.33	35.83	80.23	57.75	88.07	53.33	40.00
1:30 AM	73.96	78.36	33.76	82.43	57.63	93.57	46.67	35.00
1:45 AM	54.15	80.16	31.02	84.46	37.12	98.65	53.33	40.00
2:00 AM	54.10	81.78	31.93	86.28	57.95	103.20	46.67	35.00
2:15 AM	58.33	82.10	35.01	87.84	58.33	107.11	53.33	40.00
2:30 AM	76.95	82.41	36.63	89.12	58.50	110.31	46.67	35.00
4:30 AM	55.13	79.57	37.33	87.10	58.03	105.24	46.67	35.00
4:45 AM	76.57	79.80	39.57	85.40	58.10	101.00	53.33	40.00
5:00 AM	69.31	80.37	41.82	83.47	58.52	96.17	46.67	35.00
5:15 AM	54.68	77.03	40.86	81.35	56.72	90.87	53.33	40.00
5:30 AM	50.95	76.58	39.35	79.08	58.33	85.21	46.67	35.00

Since the amount of data is known, any formatting, formulas or charts can be configured for the 96 rows of data in the template.

Scheduled Report: Daily Report with Real Time Values Every Time an Event Occurs

In this scenario, real time data is written to the daily report any time an event occurs throughout the course of the day, down the column.

There are a few options for **Target** in this scenario. The *Append* setting would work. However, any formatting, formulas or charts in the report would not be automatically adjusted every time data is written to the report. While there are management functions that can make these adjustments, it would be easier if this were done automatically.

The *Insert At Start* or *Insert At End* **Target** works best here because it will carry any formatting with it as well as adjust any chart and formula ranges.

If the data for the report should be listed in ascending order (oldest to newest) use *Insert At End*, whereas for descending order (newest to oldest), use *Insert At Start*.

Scheduled Report: Daily Report with History Values for Every Hour of the Day

In this scenario, history data is written to the daily report at the end of the day to bring in values for every hour of the day.

Since every hour of the day can be retrieved at one time, the **Target** of the connection should be *Direct*.

Scheduled Report: Monthly Report with History Values for Every Four Hours of the Month

In this scenario, history data is written to the monthly report to show every four hours over the day.

Just like in the previous scenario, all the history data for the month can be retrieved at one time at the end of the month to produce the report.

However, there are some customers who would like to see the report being built up during the month, up to the previous day.

To accomplish this, the **Target** can be set to *Offset* and the **Offset** set to *dM*6*. *dM* is the day of the month, zero based (0-30). The *6* multiplier compensates for the six values brought in for the day (every four hours).

So, when updated on the first of the month, the **Offset** evaluates to *0*. On the second of the month it evaluates to *6*, the third to *12* and so on. If an update is missed, the six rows for that day are left blank in the report.

On Demand Report: History Values Over Time Period and Interval Specified by the User

In this scenario, history data is written to the report based on the start time, end time and interval specified by the user and generated on demand based on these settings.

The **Target** for the connection could be set *Direct* since all the data is returned in a single update. However, since the amount of data in the report is completely unpredictable, any charts, formulas or even formatting cannot be applied in the template, it must be done at runtime.

For formatting, the argument could be made to just format the entire column for the formatting desired. While that does work, every cell that is formatted makes the workbook file larger and one column is more than 1 million cells. The larger the file, the longer it takes to open, save and close, so keeping a tight rein on the file size is a good thing to do.

The *Insert At End* **Target** works best for this scenario. Not only does it apply the formatting down to all the rows of data, any summary formulas and charts are updated automatically.

In addition, if there are row calculations required as part of the report, the formulas are filled downward with the data. For example, let's take a report that shows hourly temperatures and speeds for four different stations in a plant. For every hour, the report should also show the average temperature and average speed over all four stations.

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2												
3						Temperature				Speed		
4		Time	Station 1	Station 2	Station 3	Station 4	Average	Station 1	Station 2	Station 3	Station 4	Average
5							#DIV/0!					#DIV/0!
6												
7		Average	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
8		Maximum	0.00	0.00	0.00	0.00	#DIV/0!	0.00	0.00	0.00	0.00	#DIV/0!
9		Minimum	0.00	0.00	0.00	0.00	#DIV/0!	0.00	0.00	0.00	0.00	#DIV/0!
10												
11												
12												
13												

The average of the temperatures and speeds for every hour can be done using simple formulas.

	A	B	C	D	E	F	G	H	I	J	K	L
2							=AVERAGE(C5:F5)				=AVERAGE(H5:K5)	
3						Temperature				Speed		
4		Time	Station 1	Station 2	Station 3	Station 4	Average	Station 1	Station 2	Station 3	Station 4	Average
5							#DIV/0!					#DIV/0!
6												
7		Average	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
8		Maximum	0.00	0.00	0.00	0.00	#DIV/0!	0.00	0.00	0.00	0.00	#DIV/0!
9		Minimum	0.00	0.00	0.00	0.00	#DIV/0!	0.00	0.00	0.00	0.00	#DIV/0!
10												
11												
12												
13												

When the *Insert At End* placement is applied, any formulas within range being inserted are automatically filled down for the number of rows inserted.

In the **History Data Group**, under the **Columns** tab,

The column for the average temperatures is accounted for by leaving an empty row. The average speed, however, is outside the range of the group. To account for this, in the row beneath the speed for the forth station, the **Name** is set as *. This acts like a placeholder to extend the group out by one column to account for the average speed formula.