

Worksheet to XML

What is XML

Extensible Markup Language (**XML**) defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

The following shows a sample XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- A sample XML document -->
<nutrition>
<daily-values total=2000>
  <total-fat units="g">65</total-fat>
  <saturated-fat units="g">20</saturated-fat>
  <cholesterol units="mg">300</cholesterol>
  <sodium units="mg">2400</sodium>
  <carb units="g">300</carb>
  <fiber units="g">25</fiber>
  <protein units="g">50</protein>
</daily-values>
<food>
  <name>Avocado Dip</name>
  <mfr>Sunnydale</mfr>
  <serving units="g">29</serving>
  <calories total="110" fat="100">
  <total-fat>11</total-fat>
  <saturated-fat>3</saturated-fat>
  <cholesterol>5</cholesterol>
</food>
<food>
  <name>Truffles, Dark Chocolate</name>
  <mfr>Lyndon's</mfr>
  <serving units="g">39</serving>
  <calories total="220" fat="170"/>
  <total-fat>19</total-fat>
  <saturated-fat>14</saturated-fat>
  <cholesterol>25</cholesterol>
</food>
</nutrition>
```

This document consists of the `<nutrition>` tag followed by the `<daily-values>` and `<food>` elements. The elements contain other elements which can appear multiple times.

In general, an element contains **attribute** and **text** (value) content. For example:

```
<book category=fiction>Harry Potter</book>
```

Here, *fiction* is the **attribute** and *Harry Potter* is the **text** content.

In order to take the content of a worksheet and produce an XML document, a relationship needs to be formed between the cell values on the worksheet and the XML elements of the document. This relationship needs to consider that elements can contain a set of sub elements.

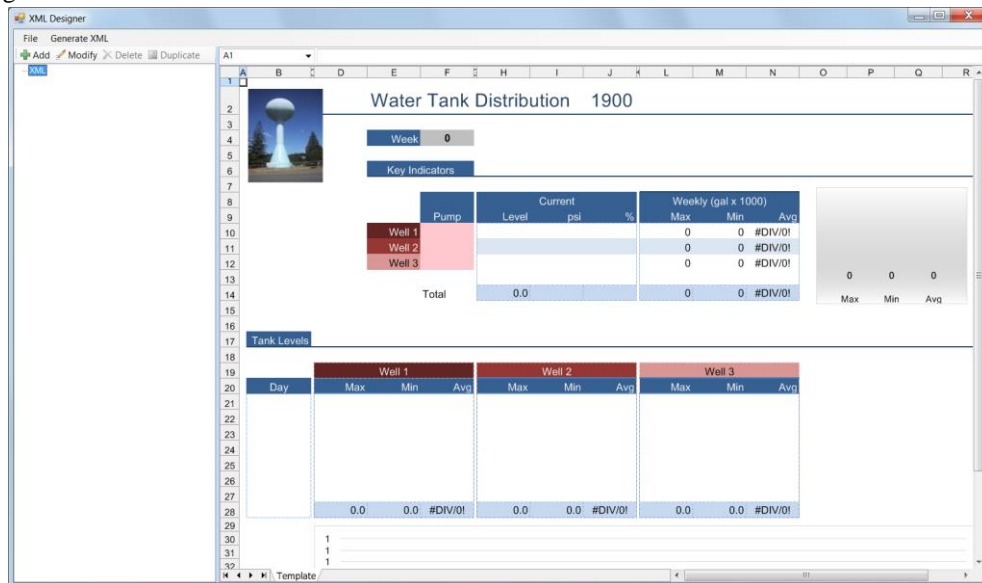
XML Designer

The XML Designer is used to specify the schema (structure) of an XML document and to establish the relationship between the elements of the XML document and the cell values. For our purposes, we will create a report template in the Design Studio and then use the XML designer to configure an XML structure to define how to export the template content.

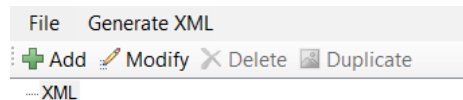
From the **Project Explorer**, open the *XLR-Demo* project which is provided in the installation.

Design

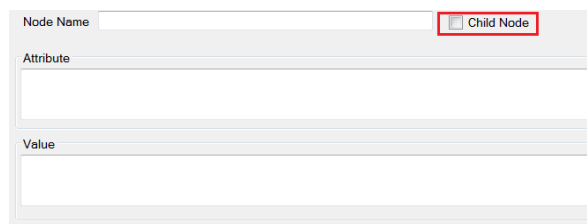
The Designer is opened from the **Tools** ribbon of the **Project Explorer** by selecting **XML Designer**. From the menu, select **File, Open Template...** to open a template in the project to define XML configuration.



The **Designer** consists of a menu bar, a left pane showing the XML structure and a right pane where the worksheet is loaded. For a workbook that has no associated XML document, the root element **XML** appears in the left.

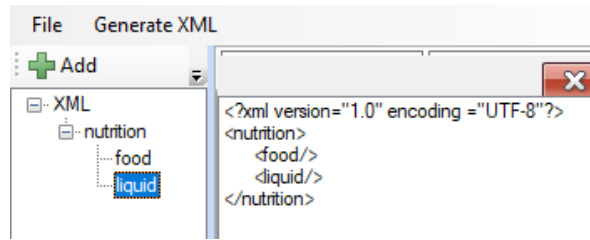


A selected element can be **Modified** or **Deleted**. If an element is **Added** it is placed as a child or a sibling relative to the selected element.



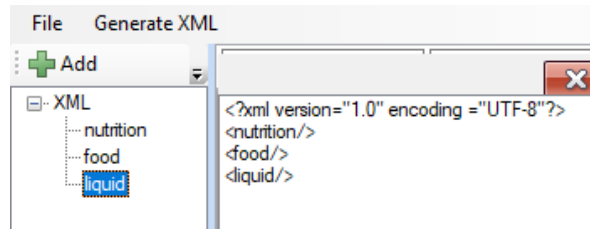
As a sibling (**Child Node** not checked), the new element will be added at the same level in the tree as the selected element whereas as a child element (**Child Node** checked), the new element will be added at the next level in the tree of the selected element. Being a parent, sibling or child will affect the XML output as follows:

Child



In the left pane, *food* and *liquid* are configured as child elements to *nutrition*. When the XML is generated (right pane), the XML shows the *nutrition* element containing the other elements.

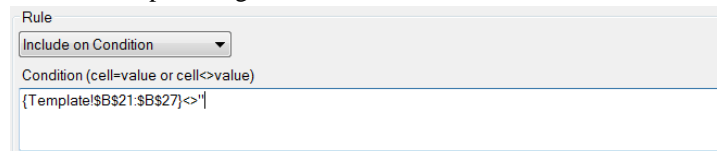
Sibling



In the left pane, *food* and *liquid* are configured as sibling elements to *nutrition*. When the XML is generated (right pane), the XML shows *all* elements as siblings with the root element as their parent.

Parent

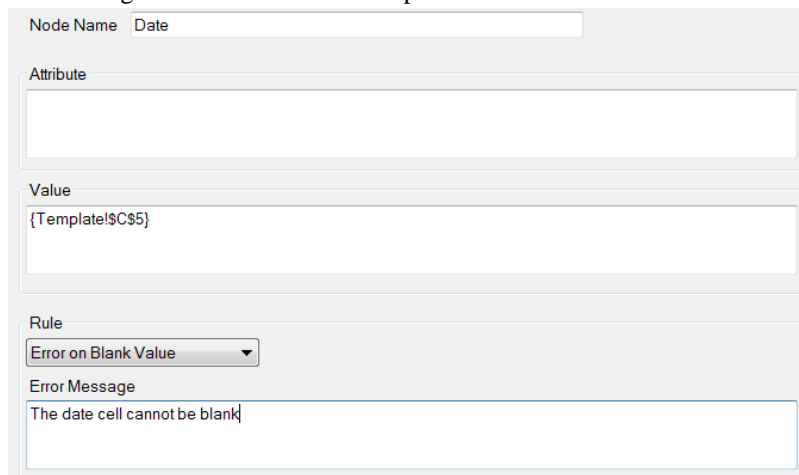
A parent node that has a cell range in the **Include on Condition** rule (see below) has special meaning. For each cell in the cell range that satisfies the condition, all the children of the parent will be iteratively sent to the XML output using the cell row.



Suppose that a parent has the settings in the above example. Every cell in the range *\$B\$21:\$B\$27* that is not blank will cause all of the child nodes to be sent to output e.g., if *\$B\$23* is not blank, all the child nodes are sent to the output, adjusted to row 23.

Settings

The XML element settings determine the XML output.



The **Name** is the name of the XML element which has the **Attribute**, **Value**, and a **Rule**. In most cases, each can contain cell references which are entered by clicking the location on the dialog and then clicking on the cell in the template.

For the root element, the **Name** is always **XML** and the **Attribute** field is used to enter a comment.

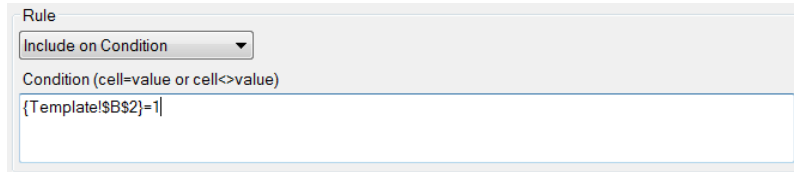
The **Rule** influences the XML output. The **Setting** is labelled according to the **Rule** selected.

- **None**

No rule is applied.

- **Include on Condition**

A specified condition determines if the element (and children) will appear in the output.



Rule
Include on Condition
Condition (cell=value or cell<>value)
{Template!\$B\$2}=1

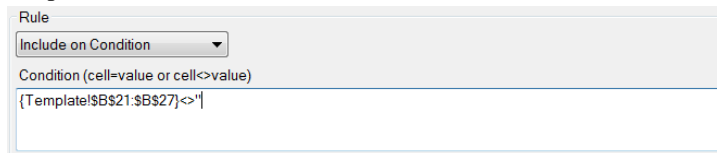
The **Condition** is applied to a range of cells and a specified value and entered in the following format:

Cell Range=Value
Cell Range<>Value

Value is a number or text. For a blank, enter ‘ ’ (consecutive single quotes).

If the **Condition** evaluates to true, then the element is included in the output. Note that this will influence all the children of the element.

If a range of cells is entered in the condition, each cell in the range is compared with the *Value*. For example:

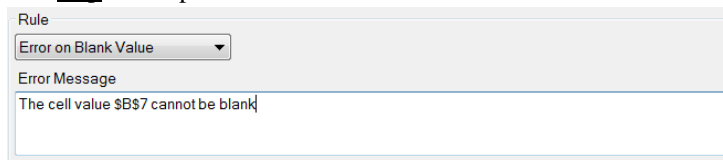


Rule
Include on Condition
Condition (cell=value or cell<>value)
{Template!\$B\$21:\$B\$27}<>'

In the above, the element (and its children) will output provided every cell in the cell range *\$B\$21* to *\$B\$27* is not blank.

- **Error on Blank Value**

If the **Value** is enumerated to blank, which can happen if a cell reference is used, then the XML output will stop at this point.

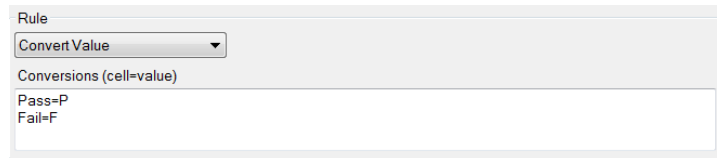


Rule
Error on Blank Value
Error Message
The cell value \$B\$7 cannot be blank

The **Error Message** is written to the XML output.

- **Convert Value**

The **Value** is converted in the XML output according to the **Conversions** setting.

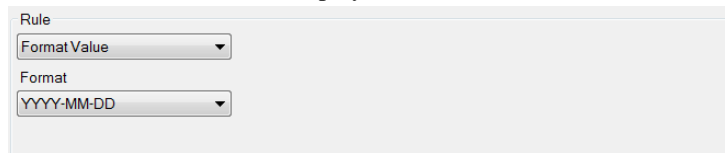


Rule
Convert Value
Conversions (cell=value)
Pass=P
Fail=F

In the above example the XML output is *P* or *F* whenever the **Value** is *Pass* or *Fail*.

- **Format Value**

The **Value** in the XML output is determined by the display value of the cell, in other words the output adheres to the cell formatting. When the cell represents a date/time, the custom format rule can be used to override the display value.



The image shows a configuration window with two dropdown menus. The first dropdown is labeled 'Rule' and is set to 'Format Value'. The second dropdown is labeled 'Format' and is set to 'YYYY-MM-DD'.

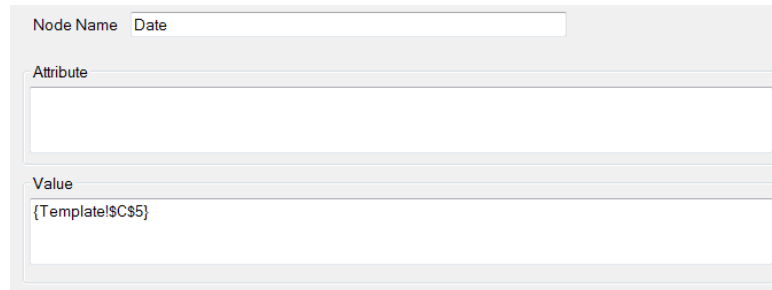
In the above example if the cell shows 1/2/2021 the XML output is 2021-01-02.

Connect

XML elements retrieve values from a workbook when they are generated.

Single Value

An element is configured by adding a cell reference to the **Value** field of the element as follows:



The image shows a configuration window for an XML element. The 'Node Name' field contains 'Date'. The 'Attribute' field is empty. The 'Value' field contains '{Template!\$C\$5}'.

In the above, the *Date* element will get its value from the value displayed in *Template!\$C\$5*.

Keywords

The keyword *{date}*, when used, will evaluate to the universal date in the form yyyy-mm-dd.

Special Characters

XML has the following special characters.

- **Quotation Mark “**
Use *"*;
- **Ampersand &**
Use *&*;
- **Apostrophe ‘**
Use *'*;
- **Less Than <**
Use *<*;
- **Greater Than >**
Use *>*;

If **Value** references cells containing a special character, the replacement is performed automatically.

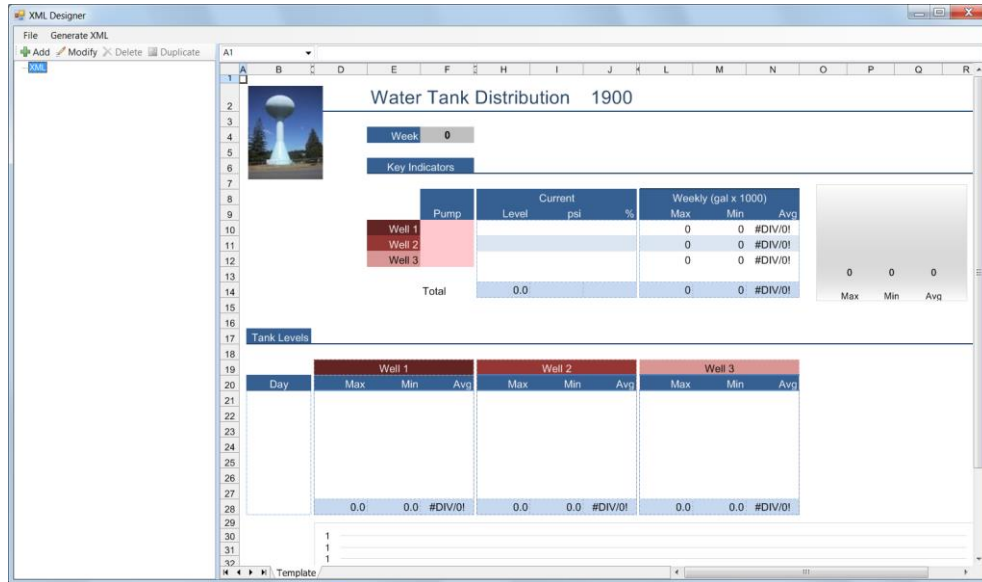
Deploy

With an XML structure defined above, an XML file can be produced from a workbook report.

Example

Design

From the **Tools** ribbon of the **Project Explorer**, select **XML designer**. From the menu, select **File, Open Template...** and open the workbook *Water Distribution.xlsx* located in the **Input** folder of the project.



The XML structure is usually designed with an understanding of how the worksheet content will appear in XML. In our example, we will use the singular values such as the week number in FS4$ and the tabular *Tank Level* values for each well starting in row 21.

In the left pane, highlight the root element *XML* and click **Modify**. The root node supports comments.

- For **Comment** enter *Water Distribution XML*
- Click **OK**

With the root element highlighted click **Add**

- Set the **Name** to *Distribution*
- Click **OK**

This will create a child element of the root.

Select the *Distribution* element and click **Add**

- Set the **Name** to *Distribution Date*
- Check **Child Node**
- Click **OK**

Select *Distribution Date* element, click **Add**

- Set the **Name** to *Date*
- Check **Child Node**
- Set the **Value** to *{Template!\$J\$2}*. This can be done by placing the cursor in the **Value** field and selecting the cell on the worksheet.
- Click **OK**

Select *Date* element, click **Add**

- Set the **Name** to *Week*
- Set the **Value** to *{Template!\$F\$4}*
- Click **OK**

At this point select **File, Save** from the menu to see the structure of the XML file with the cell references selected.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Water Distribution XML -->
<Distribution>
  <Distribution Date>
    <Date>{Template!$J$2}</Date>
    <Week>{Template!$F$4}</Week>
  </Distribution Date>
</Distribution>
```

When the XML is produced, the cell references will be replaced by the values of the report.

- Click the red X to close the XML and view the template again.

We now turn our attention to the tank levels. We will arrange this information in the XML by creating an element for each well and setting child elements to represent the data.

Highlight *Distribution Date* in the left pane, click **Add**

- Set the **Name** to *Well 1*
- Set the **Rule** to *Include on Condition*
- Set the **Setting** to *{Template!\$B\$21:\$B\$27}<>'*

- Click **OK**

The rule is configured to a range of cells which are tested to be non-blank. For a non-blank cell, all of its children are sent to the XML output adjusted by the row of the non-blank cell.

Configure the 4 **child elements Day, Max, Min, and Avg**

For the **Value** use *{Template!\$B\$21}*, *{Template!\$D\$21}*, *{Template!\$E\$21}* and *{Template!\$F\$21}*.

We will repeat for the configuration of *Well 2* and *Well 3* elements in a similar fashion to *Well 1*.

Highlight **Well 1**

- Click **Duplicate** and rename to *Well 2*
- Update the **Max, Min** and **Avg** to *{Template!\$H\$21}*, *{Template!\$I\$21}* and *{Template!\$J\$21}*.
- Repeat for *Well 3* with *\$L\$21*, *\$M\$21* and *\$N\$21*.
- Click **File, Save**

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Water Distribution XML -->
<Distribution>
  <Distribution Date>
    <Date>{Template!$J$2}</Date>
    <Week>{Template!$F$4}</Week>
  </Distribution Date>
  <Well 1>~3~{Template!$B$21:$B$27}<'
    <Day>{Template!$B$21}</Day>
    <Max>{Template!$D$21}</Max>
    <Min>{Template!$E$21}</Min>
    <Avg>{Template!$F$21}</Avg>
  </Well 1>
  <Well 2>~3~{Template!$B$21:$B$27}<'
    <Day>{Template!$B$21}</Day>
    <Max>{Template!$H$21}</Max>
    <Min>{Template!$I$21}</Min>
    <Avg>{Template!$J$21}</Avg>
  </Well 2>
  <Well 3>~3~{Template!$B$21:$B$27}<'
    <Day>{Template!$B$21}</Day>
    <Max>{Template!$L$21}</Max>
    <Min>{Template!$M$21}</Min>
    <Avg>{Template!$N$21}</Avg>
  </Well 3>
</Distribution>
```

Validate

The XML file can be generated manually from the designer.

Select the menu option **Generate XML**. This opens a dialog for a report name.

- Click the browse [...]
- In the Reports folder, select the file *2022\Oct\Water Distribution - Week 41*
- Click **OK**

This will display the XML file generated. The XML file will be in the same folder as the report.

Information in this document is subject to change without notice. SmartSights, LLC assumes no responsibility for any errors or omissions that may be in this document. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the prior written permission of SmartSights, LLC.

Copyright 2000 - 2024, SmartSights, LLC. All rights reserved.

XLReporter® is a registered trademark of SmartSights, LLC.

Microsoft® and Microsoft Excel® are registered trademarks of Microsoft, Inc.
All registered names are the property of their respective owners.