Advanced Tags Help

Table of Contents

Advanced Tags Help

Help version 1.041

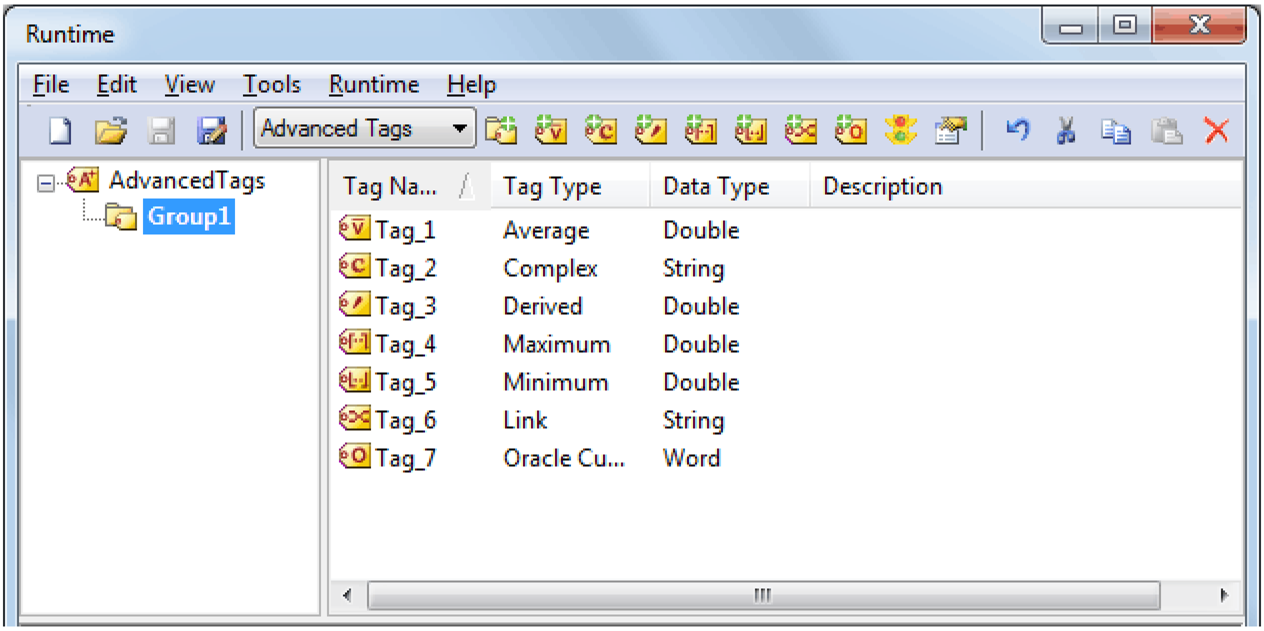
CONTENTS

Overview

Advanced Tags are used to perform simple analytics such as math, logic and evaluation functions on other tags. There are currently seven types of Advanced Tags: Average, Complex, Derived, Link, Minimum, Maximum and Oracle Cumulative.

Plug-In Interface

The Advanced Tags Plug-In interface consists of a Project View and Detail View, located on the left and right panes of the user interface respectively. The toolbar is located above both views, and contains all of the options available to the Advanced Tags objects. Users may also access these options by clicking **Edit** | **Advanced Tags**.



Tag Na...	Tag Type	Data Type	Description
Tag_1	Average	Double	
Tag_2	Complex	String	
Tag_3	Derived	Double	
Tag_4	Maximum	Double	
Tag_5	Minimum	Double	
Tag_6	Link	String	
Tag_7	Oracle Cu...	Word	

Note: Drag and drop can be used to move objects. To copy objects, press and hold the Control key during drag and drop.

Project View

The Project View displays the Advanced Tags hierarchy, which consists of a root _AdvancedTags node and Advanced Tag groups. _AdvancedTags is the root node for the Advanced Tags hierarchy, to which Advanced Tags and/or groups can be added. Advanced Tag groups are used for organization, and may contain Advanced Tags and/or other groups.

Note: In the image above, the Advanced Tag Group is labeled "Group1."

Detail View

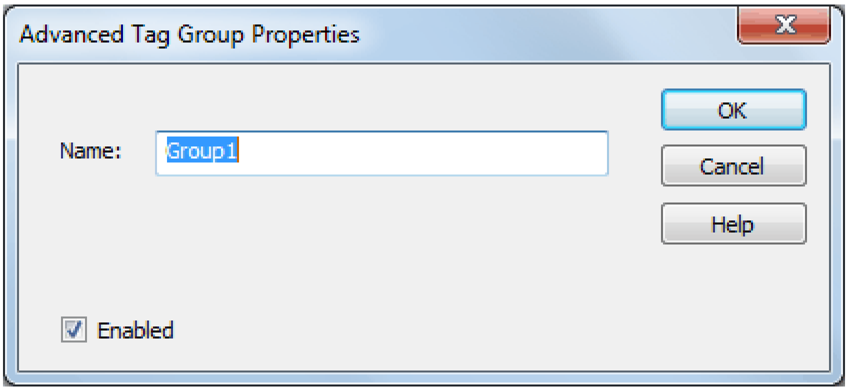
The Detail View displays the Advanced Tags under the node currently selected in the Project View. Users can create, edit, toggle, copy, cut, paste, delete and drag and drop tags to nodes in the Project View.

Toolbar

The Toolbar provides convenient access to all of the Advanced Tag functions (such as adding objects, editing objects, toggling objects, cutting, copying, pasting and undoing events).

Advanced Tag Groups

Advanced Tag groups are used to organize Advanced Tags and other groups.



Name: Group1

☑ Enabled

Descriptions of the parameters are as follows:

- **Name:** This parameter specifies the name of the tag group. The name cannot start with an underscore and must be unique among sibling tag groups. It may be up to 512 characters in length.
- **Tags assigned to this group:** This parameter specifies the number of tags assigned to the current tag group.
- **Tags assigned to this branch:** This parameter specifies the number of tags assigned to the current tag group and all descendant tag groups.
- **Enabled:** This parameter specifies the state of the tag group. If a tag group is disabled, all descendant tags and tag groups will also be disabled.

Advanced Tag CSV Import/Export

The Advanced Tags Plug-In supports the import and export of tag data in a Comma Separated Variable (CSV) file. When using CSV import and export, tags are created quickly in the desired application. CSV functions are only available when an Advanced Tag Group is selected. Furthermore, parameters that are not defined in the CSV file will be assigned appropriate default values.

Note: For information on specifying which character to use as the variable (comma or semicolon), refer to "Options - General" in the server help file.

To jump to a specific section, select a link from the list below.

Exporting an Advanced Tag Group List
Importing a CSV Tag List into the Server
Using Other Characters as the Delimiter

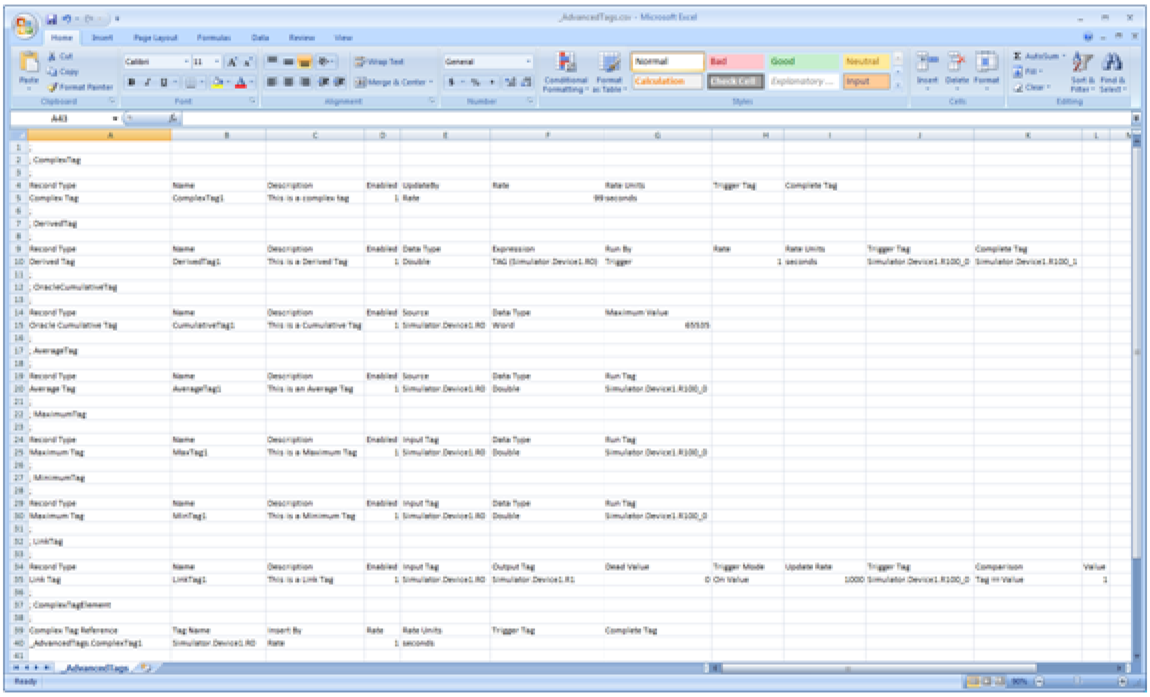
Creating a Template

The easiest way to create an import CSV file is to create a template. For more information, refer to the instructions below.

1. To start, create an Advanced Tag Group. Then, define an Advanced Tag of each type that will be used in the project.
2. Next, export the tag group as a CSV file.
3. Use this template in a spreadsheet application that supports CSV files, and then modify the file as desired.

   Note: Users can save the CSV file to disk, and then re-import it into the server beneath the same tag group or a new tag group.

Note: Microsoft Excel is an excellent tool for editing large groups of tags outside the server. Once a template CSV file has been exported, it can be loaded directly into Excel for editing. A CSV file load in Excel would appear as shown below.



	A	B	C	D	E	F	G	H	I	J	K	L
1	;											
2	; ComplexTag											
3	;											
4	Record Type	Name	Description	Enabled	UpdateBy	Rate	Rate Units	Trigger Tag	Complete Tag			
5	Complex Tag	ComplexTag1	This is a complex tag	1	Rate	99	seconds					
6	;											
7	; DerivedTag											
8	;											
9	Record Type	Name	Description	Enabled	Data Type	Expression	Run By	Rate	Rate Units	Trigger Tag	Complete Tag	
10	Derived Tag	DerivedTag1	This is a Derived Tag	1	Double	TAG (Simulator Device1.R0)	Trigger	1	seconds	Simulator Device1.R100_0	Simulator Device1.R100_1	
11	;											
12	; OracleCumulativeTag											
13	;											
14	Record Type	Name	Description	Enabled	Source	Data Type	Maximum Value					
15	Oracle Cumulative Tag	CumulativeTag1	This is a Cumulative Tag	1	Simulator Device1.R0	Word	65535					
16	;											
17	; AverageTag											
18	;											
19	Record Type	Name	Description	Enabled	Source	Data Type	Run Tag					
20	Average Tag	AverageTag1	This is an Average Tag	1	Simulator Device1.R0	Double	Simulator Device1.R100_0					
21	;											
22	; MaximumTag											
23	;											
24	Record Type	Name	Description	Enabled	Input Tag	Data Type	Run Tag					
25	Maximum Tag	MaxTag1	This is a Maximum Tag	1	Simulator Device1.R0	Double	Simulator Device1.R100_0					
26	;											
27	; MinimumTag											
28	;											
29	Record Type	Name	Description	Enabled	Input Tag	Data Type	Run Tag					
30	Maximum Tag	MinTag1	This is a Minimum Tag	1	Simulator Device1.R0	Double	Simulator Device1.R100_0					
31	;											
32	; LinkTag											
33	;											
34	Record Type	Name	Description	Enabled	Input Tag	Output Tag	Dead Value	Trigger Mode	Update Rate	Trigger Tag	Comparison	Value
35	Link Tag	LinkTag1	This is a Link Tag	1	Simulator Device1.R0	Simulator Device1.R1	0	On Value	1000	Simulator Device1.R100_0	Tag == Value	1
36	;											
37	; ComplexTagElement											
38	;											
39	Complex Tag Reference	Tag Name	Insert By	Rate	Rate Units	Trigger Tag	Complete Tag					
40	_AdvancedTags.ComplexTag1	Simulator Device1.R0	Rate	1	seconds							
41												

Exporting an Advanced Tag Group List

Exporting an Advanced Tag Group list will generate a .CSV text file that contains a section for each Advanced Tag type, in addition to a section for Complex Tag elements. Each section contains a heading record followed by a record for each tag defined under the selected tag group. The heading records contain the following column names: AverageTag, OracleCumulativeTag, ComplexTag, DerivedTag, LinkTag, MaximumTag, MinimumTag, and ComplexTagElement. Column names must be exactly the same as those listed; however, columns may be in any order and all columns (except for "Name," "Tag Name," and "Complex Tag Reference") are optional.

Important: The Advanced Tag type sections may be in any order, with the exception of the Complex Tag Elements section. That section must be placed after the Complex Tags section. A Complex Tag Element must reference the Complex Tag to which it belongs. If the parent Complex Tag is not added during import (due to omission from the CSV file or another error), the Complex Tag Element will not be added.

AverageTag Fields

Column Name	Values
Record Type	"Average Tag"
Name	The tag name (with path relative to the tag group being imported/exported).
Description	The tag description (which may be up to 64 characters).
Enabled	1 to enable, and 0 to disable.
Source	The tag from the server for which a running average will be calculated.
Data Type	The Average Tag's data type.
Run Tag	The tag that will be monitored to determine if it is appropriate to calculate the average value of the Average Tag.

OracleCumulativeTag Fields

Column Name	Values
Record Type	"Oracle Cumulative Tag"
Name	The tag name (with path relative to the tag group being imported/exported).
Description	The tag description (which may be up to 64 characters).
Enabled	1 to enable, and 0 to disable.
Source	The tag from the server for which the cumulative value will be calculated.
Data Type	The Oracle Cumulative Tag's data type.
Maximum Value	The maximum value of the tag being used.

ComplexTag Fields

Column Name	Values

Record Type	"Complex Tag"
Name	The tag name (with path relative to the tag group being imported/exported).
Description	The tag description (which may be up to 64 characters).
Enabled	1 to enable, and 0 to disable.
Update By	Options include Rate or Trigger.
Rate	The value for the update rate.
Rate Units	Options include milliseconds, seconds, minutes, hours, or days.
Trigger Tag	The tag from the server that triggers the Complex Tag update.
Complete Tag	The tag from the server that is set once a Complex Tag update has been sent to all subscribing clients.

DerivedTag Fields

Column Name	Values
Record Type	"Derived Tag"
Name	The tag name (with path relative to the tag group being imported/exported).
Description	The tag description (which may be up to 64 characters).
Enabled	1 to enable, and 0 to disable.
Data Type	The Derived Tag's data type.
Expression	The expression that will be executed by the tag.
Run By	Options include Rate or Trigger.
Rate	The value for the update rate.
Rate Units	Options include milliseconds, seconds, minutes, hours, or days.
Trigger Tag	The tag from the server that triggers the expression to be evaluated/run.
Complete Tag	The tag from the server that is set once the expression has been evaluated/run.

LinkTag Fields

Column Name	Values
Record Type	"Link Tag"
Name	The tag name (with path relative to the tag group being imported/exported).
Description	The tag description (which may be up to 64 characters).
Enabled	1 to enable, and 0 to disable.
Input Tag	The server tag used as the source of the link.
Output Tag	The server tag used as the destination of the link.
Dead Value	The value written to the Output if the Input is of bad quality.
Trigger Mode	Options include On Input Value Change, On Timer, and On Value.
Update Rate	The update rate is in milliseconds.
Trigger Tag	The tag value compared to the Value parameter using the Comparison parameter.
Comparison	Options include the following: Tag == Value Tag != Value Tag > Value Tag >= Value Tag < Value Tag <= Value
Value	The Value compared to the Trigger Tag.

MaximumTag Fields

Column Name	Values
Record Type	"Maximum Tag"
Name	The tag name (with path relative to the tag group being imported/exported).
Description	The tag description (which may be up to 64 characters).
Enabled	1 to enable, and 0 to disable.
Input Tag	The tag from the server for which a maximum value will be calculated.
Data Type	The Maximum Tag's data type.
Run Tag	The tag that will be monitored to determine if it is appropriate to calculate the

	maximum value of the Maximum Tag.

MinimumTag Fields

Column Name	Values
Record Type	"Minimum Tag"
Name	The tag name (with path relative to the tag group being imported/exported).
Description	The tag description (which may be up to 64 characters).
Enabled	1 to enable, and 0 to disable.
Input Tag	The tag from the server for which a minimum value will be calculated.
Data Type	The Minimum Tag's data type.
Run Tag	The tag that will be monitored to determine if it is appropriate to calculate the minimum value of the Minimum Tag.

ComplexTagElement Fields

Column Name	Values
Complex Tag Reference	The Complex Tag to which the element belongs.
Tag Name	The server tag that is inserted into the Complex Tag.
Insert By	Options include Rate or Trigger.
Rate	The value for the update rate.
Rate Units	Options include milliseconds, seconds, minutes, hours, or days.
Trigger Tag	The tag from the server that triggers the element insertion.
Complete Tag	The tag from the server that is set once the element insertion is complete.

See Also: Advanced Tag Types

Importing a CSV Tag List into the Server

Once the tag list has been edited, it can be re-imported into the server by clicking **File** | **Import CSV**.

Note: This option is only available when an Advanced Tag Group is selected.

Using Other Characters as the Delimiter

When utilizing a CSV file that does not use a comma or semi-colon delimiter, do one of the following:

1. Save the project in XML. Then, perform mass configuration on the XML file instead of using CSV.
2. Perform a search-and-replace on the delimiter in the CSV file and then replace the delimiter with a comma or semicolon. The delimiter being used by the server (either comma or semi-colon) must be set to the replacement character.

Note: For more information, refer to "Options - General" in the server help file.

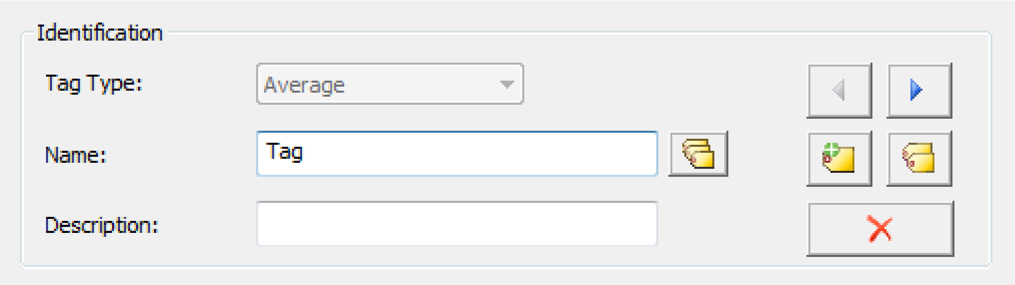
Advanced Tag Types

Select a link from the following list to obtain specific information for the Advanced Tag of interest.

Average Tags
Complex Tags
Derived Tags
Link Tags
Maximum Tags
Minimum Tags
Oracle Cumulative Tags

Identification

All Advanced Tags' property dialogs include an Identification section that contains parameters available to all Advanced Tags. Users may create multiple Advanced Tags by utilizing the options displayed in this section.



Identification
Tag Type: Average
Name: Tag
Description:

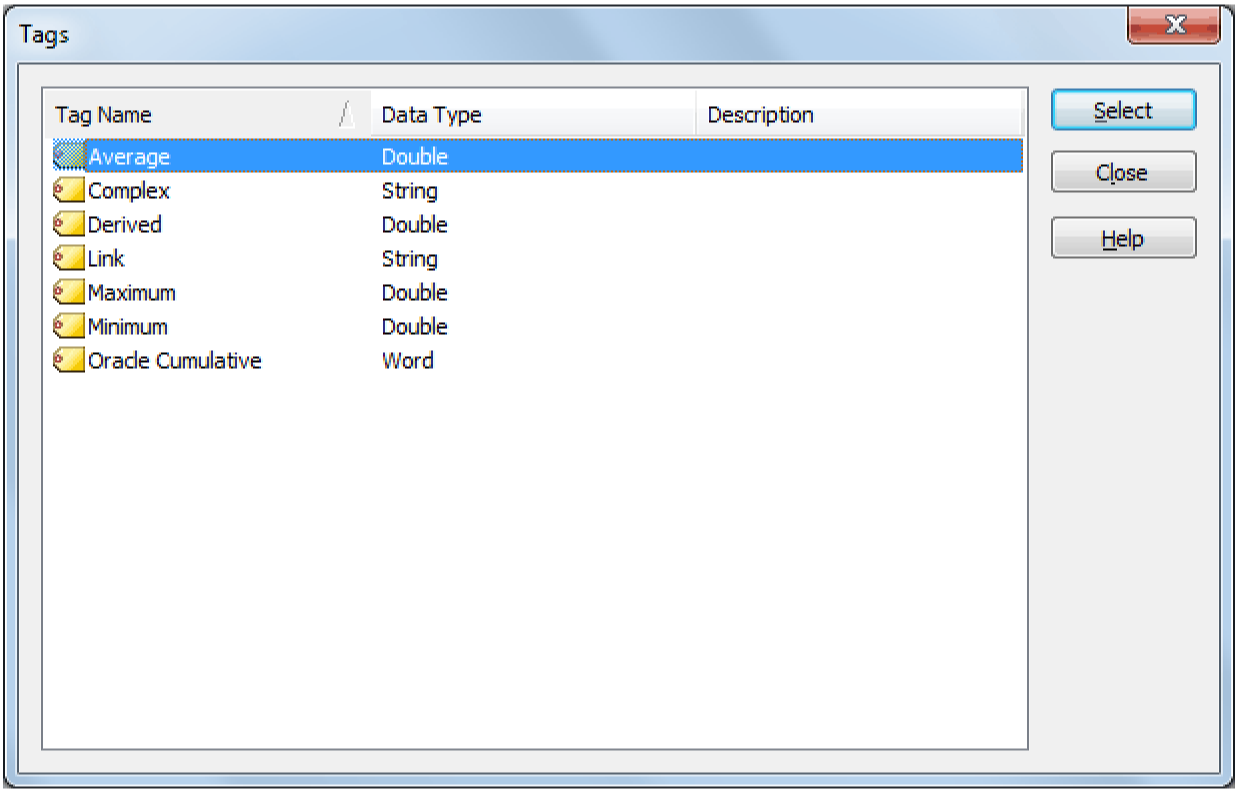
Descriptions of the parameters are as follows:

- **Tag Type:** This parameter specifies the tag type. When creating a new tag, users set this value to the desired tag type through the drop-down menu. When editing an existing tag, this parameter will display the tag type of the tag being edited and cannot be changed.
- **Name:** This parameter specifies the name of the tag. The name can be up to 256 characters in length and cannot contain periods, double quotes or start with an underscore.
- **Show Tag List:** When clicked, this button displays the **Tags** dialog used to select a tag under the currently selected node in the Project View. If the selected node has no tags, this parameter will be disabled. For more information, refer to **Tag List**.

  Note: To open and close the tag list, press **Alt+L**.
- **Description:** This parameter attaches a comment to a tag. The description may be a string of up to 64 characters.
- **Previous:** When clicked, this button selects the previous tag in the tag list. It will be disabled if a new tag is being created (or if there is no previous tag).
- **Next:** When clicked, this button selects the next tag in the tag list. It will be disabled if a new tag is being created (or if there is no next tag).
- **New:** When clicked, this button creates a new tag. This will cancel any edits made to the current tag.
- **Duplicate:** When clicked, this button duplicates the current tag. It will apply any edits to the current tag and then create a new tag that is a copy of the current tag. The tag name will be unique among sibling tags.
- **Delete:** When clicked, this button deletes the current tag. Afterward, the dialog will move to the next tag in the list. If no tag follows, the dialog will move to the previous tag in the list. If there are neither following nor preceding tags, the dialog will default to creating a new tag.
- **Enabled:** This parameter specifies the state of the tag. Tags that are enabled have references to other server tags and do computational work. Tags that are disabled have no references to server tags and do no work.

Tag List

The Tags list displays all the tags under the currently selected tag group and is used to select a tag for editing.

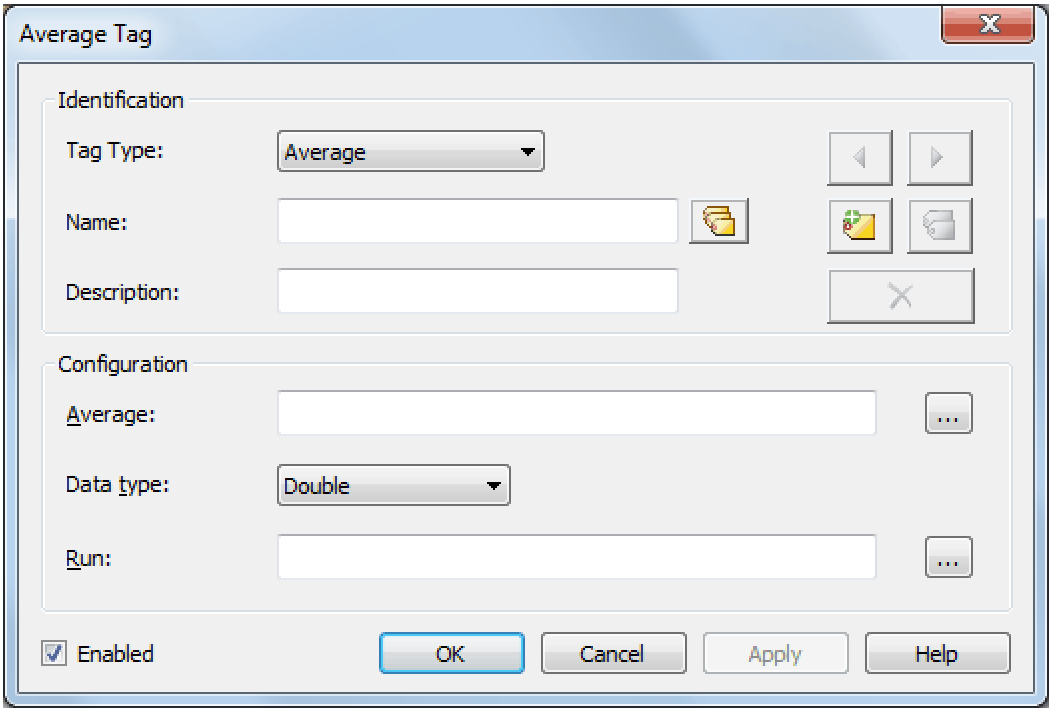


Tag Name	Data Type	Description
Average	Double	
Complex	String	
Derived	Double	
Link	String	
Maximum	Double	
Minimum	Double	
Oracle Cumulative	Word	

Average Tags

When enabled, Average Tags calculate a running average of a tag's value. Averaging is initiated by the value of a Run Tag transitioning from zero to non-zero. The rate that the tag is updated from the server with a new value comes from the update/scan rate on the device tag in the server. The default scan rate is 100 milliseconds. When disabled, Average Tags do no averaging and have a value of 0.

Note: Array tags are not supported.



Tag Type: Average

Name:

Description:

Average:

Data type: Double

Run:

☑ Enabled

Descriptions of the Configuration parameters are as follows:

- **Average:** This parameter specifies the tag from the server for which a running average will be calculated.
- **Data Type:** This parameter specifies the data type of the Average Tag.

  Note: Although Double is the only data type available in the Tag Properties, users can request a different data type in the OPC Client. The server will take ownership for coercing the data.
- **Run:** This parameter specifies the tag that will be monitored to determine if it is appropriate to calculate the average value of the Average Tag. Whenever the Run Tag's value transitions from a zero to non-zero value, a new running average of the Average Tag will be calculated. The calculation is based on the scan rate of the Average Tag, so intermediate average values will be reported to client applications. When the Run Tag's value transitions back to zero, the last average value will be maintained and reported to the client applications.

Example

A tag that measures the temperature at a location (with a scan rate of 60000 milliseconds) would update the Average Tag with the following values every minute:

72.3 @ 10:00
72.5 @ 10:01
72.9 @ 10:02
73.4 @ 10:03
72.4 @ 10:04
71.6 @ 10:05
71.2 @ 10:06
70.5 @ 10:07

If the Trigger Tag had successfully transitioned from zero to one, the Average Tag would show the following values:

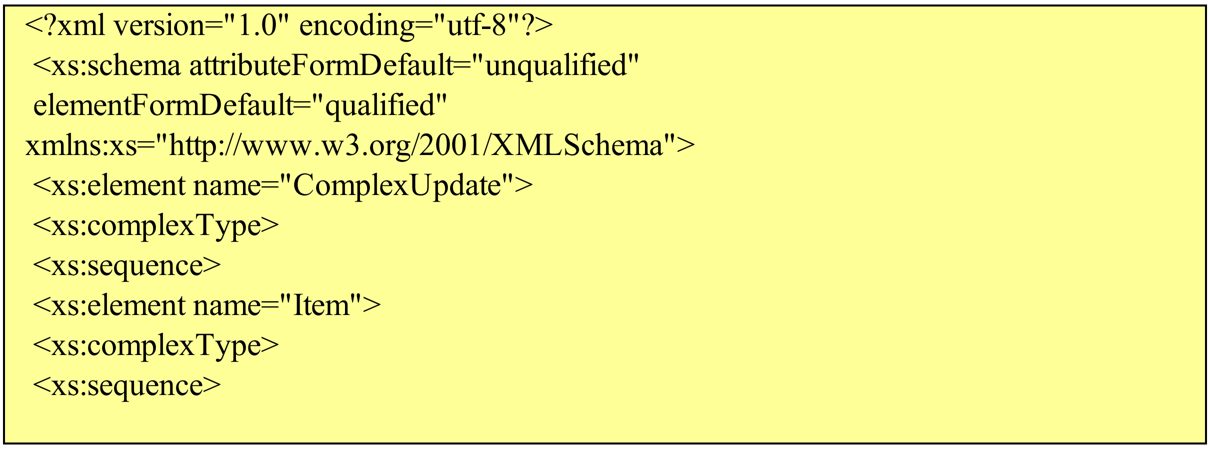
72.3 @ 10:00
72.4 @ 10:01
72.6 @ 10:02
72.8 @ 10:03
72.7 @ 10:04
72.6 @ 10:05
72.3 @ 10:06
72.1 @ 10:07

Note: If the quality of the Average Tag or the Trigger Tag is bad, the quality of the Average Tag will also be bad. Furthermore, if the quality of the Average Tag or the Trigger Tag is bad, averaging will stop. In order for it to restart, both tags must return to good quality and the Trigger Tag value must successfully transition from zero to non-zero.

Complex Tags

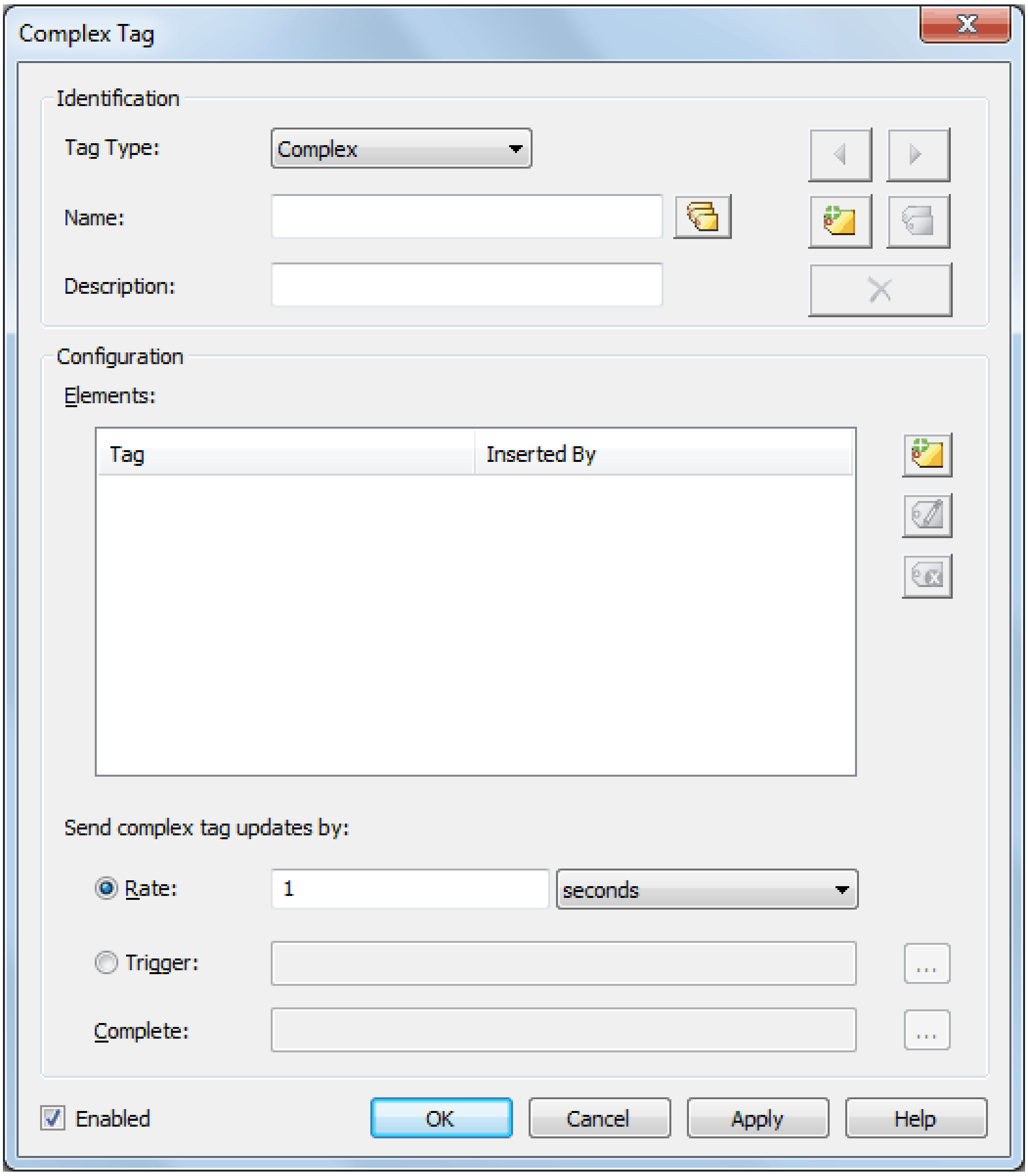
When enabled, Complex Tags group multiple tags of varying data types as a single complex item or structure. When disabled, they do no work and have a string value of *<ComplexUpdate></ComplexUpdate>*. Although they are primarily used with Oracle MOC connections, any client can read the tags. The tags return data as an XML string defined by the schema displayed in the table below.

Note: Array tags are not supported.



```
<?xml version="1.0" encoding="utf-8"?>
 <xs:schema attributeFormDefault="unqualified"
 elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="ComplexUpdate">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="Item">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="Name" type="xs:string" />
 <xs:element name="Value">
 <xs:complexType>
 <xs:simpleContent>
 <xs:extension base="xs:string">
 <xs:attribute name="DataType" type="xs:unsignedByte" use="required" />
 </xs:extension>
 </xs:simpleContent>
 </xs:complexType>
 </xs:element>
 <xs:element name="Quality" type="xs:int" />
 <xs:element name="TimeStamp" type="xs:string" />
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 </xs:sequence>
 </xs:complexType>
 </xs:element>
</xs:schema>
```

Note: The updates grouped in a complex update are not atomic, but are aggregated over a time interval. As a result, clients cannot assume that all updates in a complex update occurred at the same time. To get the time of the update, use the item's time stamp element.



Complex Tag

Identification

Tag Type: Complex

Name:

Description:

Configuration

Elements:

Tag	Inserted By

Send complex tag updates by:

● Rate: 1 seconds

○ Trigger:

Complete:

☑ Enabled

Descriptions of the Configuration parameters are as follows:

- **Elements:** This parameter lists tags that will be part of the Complex Tag.

  Note: Array tags cannot be added to the Element List.
- **Add Element:** When clicked, this button launches the Complex Element Property, which is then used to select tags for the Element list.
- **Modify Element:** When clicked, this button launches the Complex Element Property, which is then used to Modify elements that are highlighted in the Element list.
- **Delete Element:** When clicked, this button deletes elements that are highlighted in the Element list.
- **Send complex tag updates by:** This parameter specifies whether complex tag updates are sent by Rate or Trigger. The default setting is Rate. Descriptions are as follows:
  - **Rate:** When selected, this parameter dictates the rate at which the Complex Tag updates will be published. The default setting is 1 second.

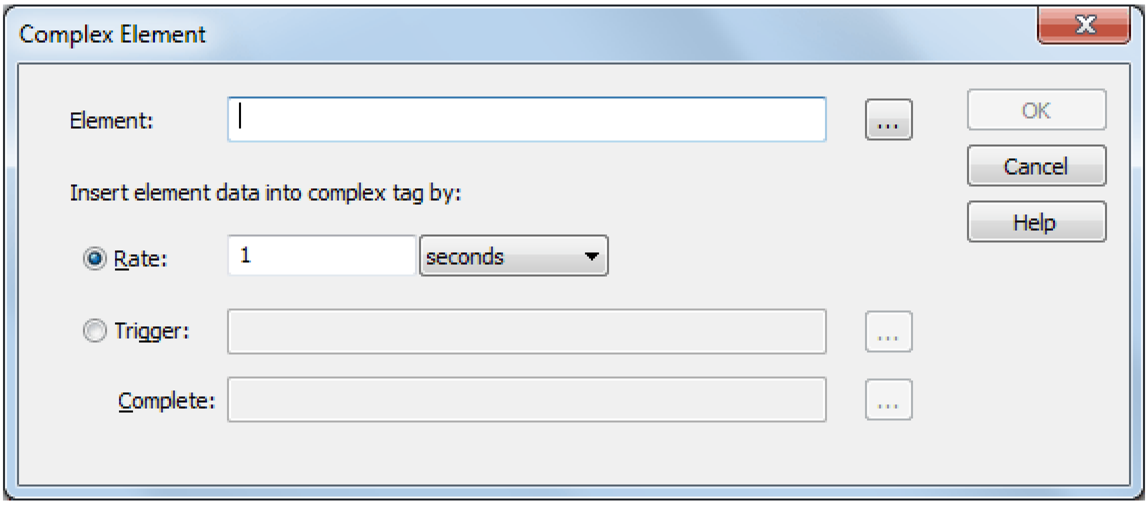
    Note: The range depends on the unit. The range for milliseconds is 10 to 999,999,999. The range for seconds is 1 to 4294967. The range for minutes is 1 to 71582. The range for hours is 1 to 1193. The range for days is 1 to 49. The default unit is seconds.

- **Trigger:** When selected, this parameter dictates the condition at which Complex Tag updates will be published. There are two types: **Trigger** and **Complete**. The Trigger Tag's value will be monitored for a zero to non-zero transition. When this transition occurs, it results in a Complex Tag update. No additional updates will occur until the tag's value is reset to zero and another zero to non-zero transition occurs. When a Complete Tag is provided, the server will write a value of 1 to the Complete Tag (in order to notify that the update has been completed) once a Complex Tag update has been sent to all subscribing clients. Users can tie this to logic in a controller or other subsystem to indicate that the next complex value is ready to be formulated.

  Note: Array tags cannot be used as Trigger or Complete tags. Complete tags must have writeable access.

Complex Element

The Complex Element dialog is used to browse for an element and then insert it into complex tag. The element data may be inserted by Rate or Trigger. For descriptions of these options, refer to the parameter descriptions above.



Complex Element

Element:

Insert element data into complex tag by:

● Rate: 1 seconds

○ Trigger:

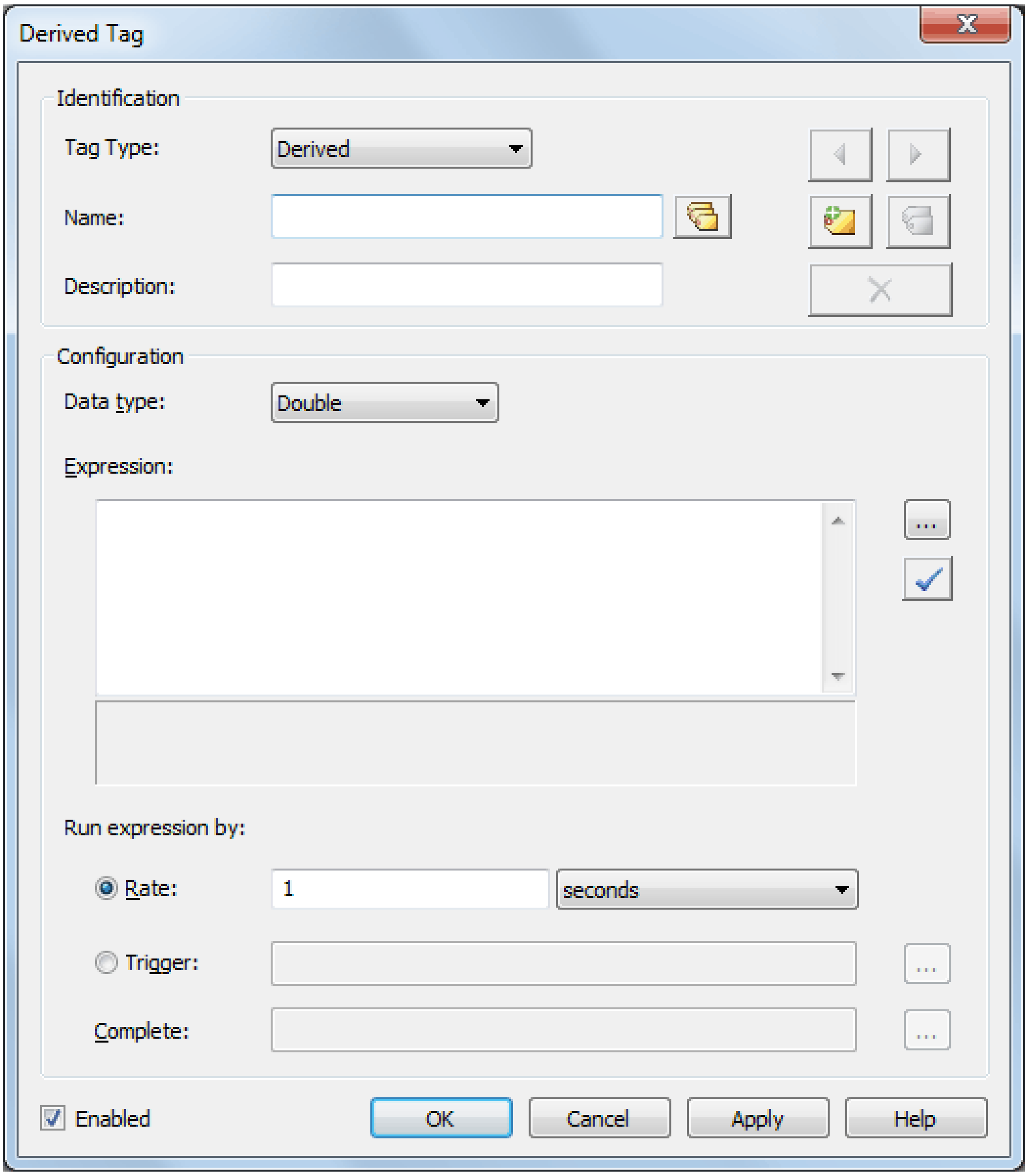
Complete:

Derived Tags

Derived Tags are used to provide math and/or logic operations on one or more tag values. These functions can be combined to perform complex math routines and evaluations. When enabled, the expression is evaluated. When disabled, the expression is not evaluated and the tag value is 0.

Note 1: The execution of the string follows standard arithmetic rules. For more information, refer to **Expression Strings**.

Note 2: Array tags are not supported.



Tag Type: Derived

Data type: Double

● Rate: 1 seconds

○ Trigger:

☑ Enabled

Descriptions of the Configuration parameters are as follows:

- **Data Type:** This parameter specifies the data type of the result of the expression execution.

  Note: If a Derived Tag has a data type of String, the maximum length of the tag's String value is 4096. This limit includes characters in an expression (such as "abc"). Any characters that exceed this limit will be truncated.
- **Expression:** This parameter specifies the expression that will be executed by the tag.
- **Tag Browser:** When clicked, this button launches the Tag Browser (from which users may locate expressions).
- **Check Expressions:** When clicked, this button checks the validity of the selected expression.
- **Run expression by:** This parameter specifies whether the expression is run by Rate or Trigger. The default setting is Rate. Descriptions are as follows:

  - **Rate:** When selected, this parameter dictates the rate at which the expression is evaluated/run. The default setting is 1 second.

    Note: The range depends on the unit. The range for milliseconds is 10 to 999,999,999. The range for seconds is 1 to 4294967. The range for minutes is 1 to 71582. The range for hours is 1 to 1193. The range for days is 1 to 49. The default unit is seconds.

- **Trigger:** When selected, this parameter dictates the condition at which the expression is evaluated/run. There are two types: **Trigger** and **Complete**. The tag defined in the Trigger field will be monitored for a zero to non-zero transition. When this transition occurs, the expression is evaluated/run. No additional updates will occur until the tag's value is reset to zero and another zero to non-zero transition occurs. When a Complete Tag is provided, the server will write a value of 1 to the tag defined in the Complete field when the expression is evaluated/run. Users can tie this to logic in a controller or other subsystem to indicate that the evaluation of the expression is complete.

Expression Strings

Expression strings are used to specify the math formula that is being defined. Expressions are validated for correct syntax and valid format. Array tags cannot be used in an expression.

Note: When referencing tags in the Expression string, the fully qualified item name will be preceded by the Keyword Tag and enclosed in parentheses.

Example 1:
TAG(Channel1.Device1.Tag1).

Note: Comments are also supported in Expression strings. Anything after a pound sign (#) will be ignored.

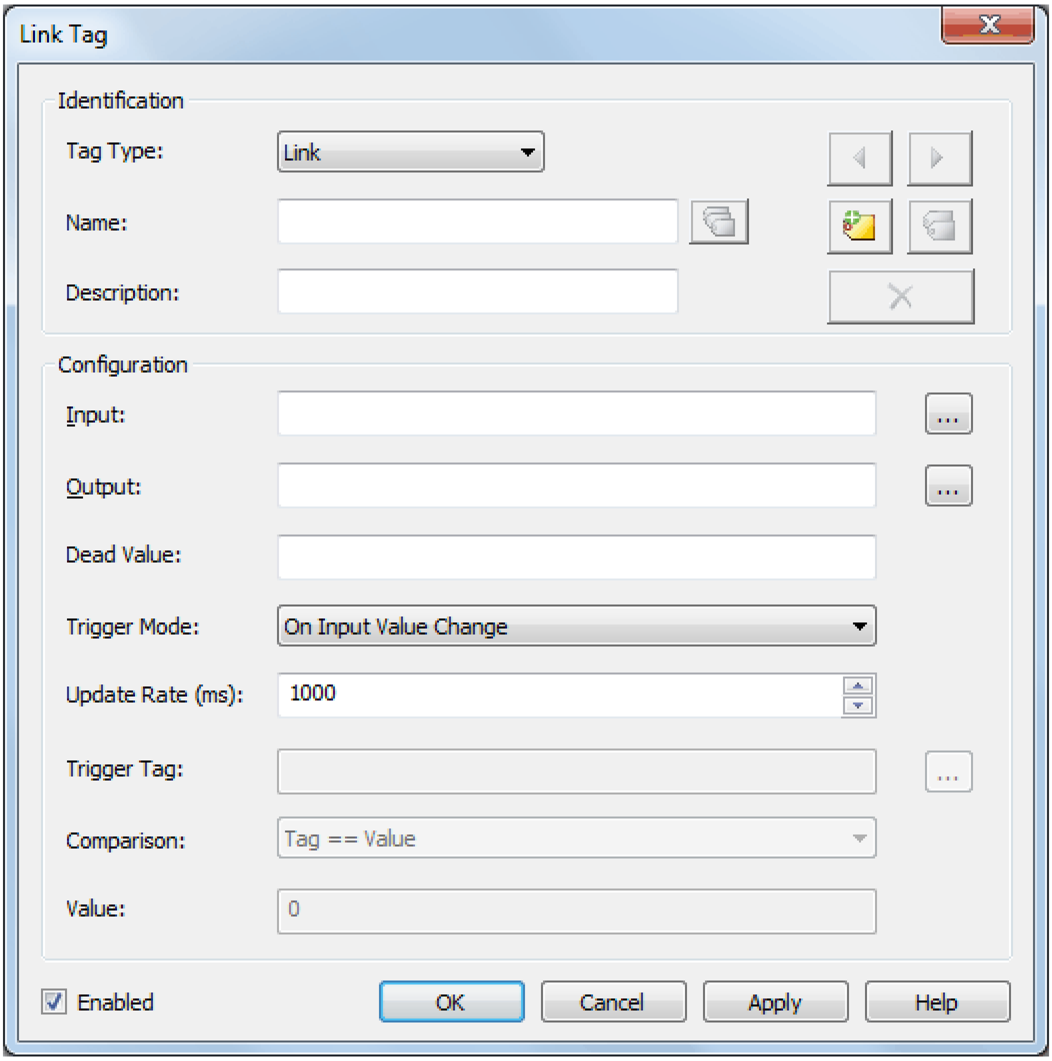
Example 2:
TAG(x) + TAG(y) # Add TAG x and Tag y

Note: For more information specific to Derived Tags, refer to **Arithmetic Operators**, **Functions** and **Logic**.

Link Tags

Link Tags are used to link two server tags. For example, this allows Tag A from Device A to be linked to Tag B from Device B without requiring a Third-Party client connection. When enabled, a Link Tag writes the value of an Input Tag to an Output Tag, thus effectively linking the two tags. It also has a string value of *<input tag> == <output tag>*. When disabled, a Link Tag does no linking and its value is *<input tag> != <output tag>*.

See Also: **Array Tags**



Link Tag

Identification

Tag Type: Link

Name:

Description:

Configuration

Input:

Output:

Dead Value:

Trigger Mode: On Input Value Change

Update Rate (ms): 1000

Trigger Tag:

Comparison: Tag == Value

Value: 0

☑ Enabled

Descriptions of the Configuration parameters are as follows:

- **Input:** This parameter specifies the Input Tag, which must be a valid server tag. It is used as the source of the link, and its value is written to the Output Tag.

  Note: The Input and Output fields cannot be the same.
- **Output:** This parameter specifies the Output Tag, which must be a valid server tag. It is used as the destination of the link and is written to using the value of the Input Tag. The Output Tag must be writable.

  Note: The Input and Output fields cannot be the same.
- **Dead Value:** This parameter specifies the value written to the Output if the Input is of bad quality. This value's data type is the data type of the Output. Users must choose dead values that are within the range of the Output tag's data type in order to ensure that no data will be lost. If the field is left empty, the dead value will be disabled. If disabled, the Output's value and quality will not change when the Input is bad quality.

  Note: If the Output is an array tag, each array element is set equal to the Dead Value.
- **Trigger Mode:** This parameter specifies the type of trigger used to initiate the link. Descriptions of the options are as follows:

  - **On Input Value Change:** The input is linked to the output anytime that the input is good quality and its value changes.
  - **On Input Value Change (Ignore Initial Update):** The input is linked to the output anytime that the input value changes, except for the initial update (which is ignored and does not count as an input value change). The Link Tag will have the String value <*input tag*> != <*output tag*>

until after the first update successfully completes. At that time, subsequent updates will have the String value <*input tag*> == <*output tag*>.

Note: An initial update of bad quality is not considered an initial update.

- **On Timer:** The input is linked to the output every Update Rate (milliseconds).
- **On Value:** The input is linked to the output each time the Trigger Tag satisfies the logic determined by the Value and Comparison fields.

  Note: It is possible to miss trigger events if the Link Tag's update rate is slower than the Trigger Tag's rate of change. For example, consider a situation in which a Link Tag's update rate is 10000 milliseconds and the Trigger Tag's rate of change is 1000 milliseconds. The trigger condition is Trigger Tag == 1.

  0 @ 09:59
  1 @ 10:00 - Link executes.
  5 @ 10:01
  1 @ 10:02

  ...

  1 @ 10:10 - No link.

  No link is made at 10:10 because the value is still 1. As far as the Link Tag is concerned, the Trigger Tag's value never changed. To get around this issue, the update rate should be at least twice the rate of change of the trigger tag.

- **Update Rate (ms):** This parameter depends on the chosen Trigger Mode. Descriptions of the options are as follows:

  - **On Input Value Change:** The scan rate of the Input Tag. This is the fastest rate at which a change in the Input value may be detected.
  - **On Timer:** The link rate.
  - **On Value:** The scan rate of the Trigger Tag. This is the fastest rate at which a change in the Trigger Tag value may be detected.

    Note: A link requires a read of the Input Tag and a write to the Output Tag. These operations are given priority. If the linking is occurring at a fast rate (such as 50 milliseconds) it is possible that other clients operating in exception based mode could miss data updates. Users should make sure that linking is occurring no faster than what is required.
- **Trigger Tag:** This parameter specifies the tag value compared to the Value parameter using the Comparison parameter (such as *Trigger Tag == Value*). The Trigger Tag cannot be an array tag.

  Note: The Trigger and Output tags cannot be the same.
- **Comparison:** This parameter specifies the logical comparison between the Trigger Tag value and the Value parameter. Descriptions of the options are as follows:

  - **Tag > Value:** The trigger tag value is greater than the value.
  - **Tag >= Value:** The trigger tag value is greater than or equal to the value.
  - **Tag < Value:** The trigger tag value is less than the value.
  - **Tag <= Value:** The trigger tag value is less than or equal to the value.
  - **Tag == Value:** The trigger tag value is equal to the value.
  - **Tag != Value:** The trigger tag value is not equal to the value.
- **Value:** This parameter specifies the Value compared to the Trigger Tag. The data type of this value is equal to the data type of the Trigger Tag.

Array Tags

Link Tags can be used to link array tags, provided the following:

- The arrays are the same size.
- The arrays are the same data type. Otherwise, the values must be moved from one type to another.

It is best to use arrays with the same dimensions. For more information, refer to the table of examples below.

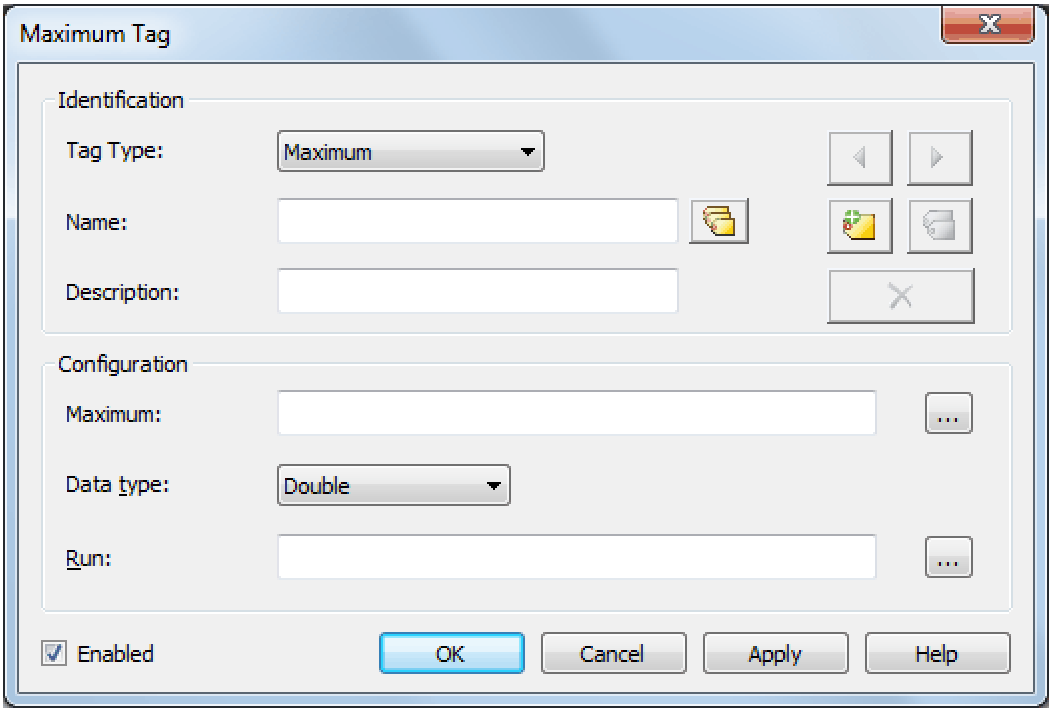
Input Array Tag/Type	Output Array Tag/Type	Succeeds w/o Error
K001[6]	K0100[6]	Yes
K001[6]	K0100[2][3]	Yes
K001[2][3]	K0100[6]	Yes
K001[6]	K0100[2][2]	No
K001[6] / Word	K0100[6] / DWord	Yes
K001[6] / Float	K0100[6] / DWord	No

Note: The examples above use address syntax from the Simulator Driver and do not include the channel and device names.

Maximum Tags

When enabled, Maximum Tags calculate the maximum value of a tag's value. Determining the maximum value is initiated by the value of a Run Tag transitioning from zero to non-zero. The rate that the tag is updated from the server with a new value comes from the update/scan rate on the device tag in the server. The default scan rate is 100 milliseconds. When disabled, Maximum Tags do no work and have a value of 0.

Note: Array tags are not supported.



Tag Type: Maximum

Name:

Description:

Maximum:

Data type: Double

Run:

☑ Enabled

Descriptions of the Configuration parameters are as follows:

- **Maximum:** This parameter specifies the tag from the server for which a maximum value is calculated.
- **Data Type:** This parameter specifies the data type of the Maximum Tag.

  Note: Although Double is the only data type available in the Tag Properties, users can request a different data type in the OPC Client. The server will take ownership for coercing the data.
- **Run:** The Run Tag will be monitored to determine if it is appropriate to calculate the maximum value of the Maximum Tag. Whenever the Run Tag's value transitions from a zero to a non-zero value, a new maximum value will be calculated for the Maximum Tag. The calculation determines whether or not the currently scanned maximum value is higher than the reported maximum value. If it is, it will replace the reported value with the scanned value. When the Run Tag's value transitions back to zero, the last maximum value will be maintained and reported to the client application.

Examples

A tag that measures the temperature at a location (with a scan rate of 60000 milliseconds) would update the Maximum Tag with the following values every minute:

72.3 @ 10:00
72.5 @ 10:01
72.9 @ 10:02
73.4 @ 10:03
72.4 @ 10:04
71.6 @ 10:05
71.2 @ 10:06
70.5 @ 10:07

If the Trigger Tag had successfully transitioned from 0 to 1, the Maximum tag would show the following values:

72.3 @ 10:00
72.5 @ 10:01
72.9 @ 10:02
73.4 @ 10:03
73.4 @ 10:04
73.4 @ 10:05
73.4 @ 10:06
73.4 @ 10:07

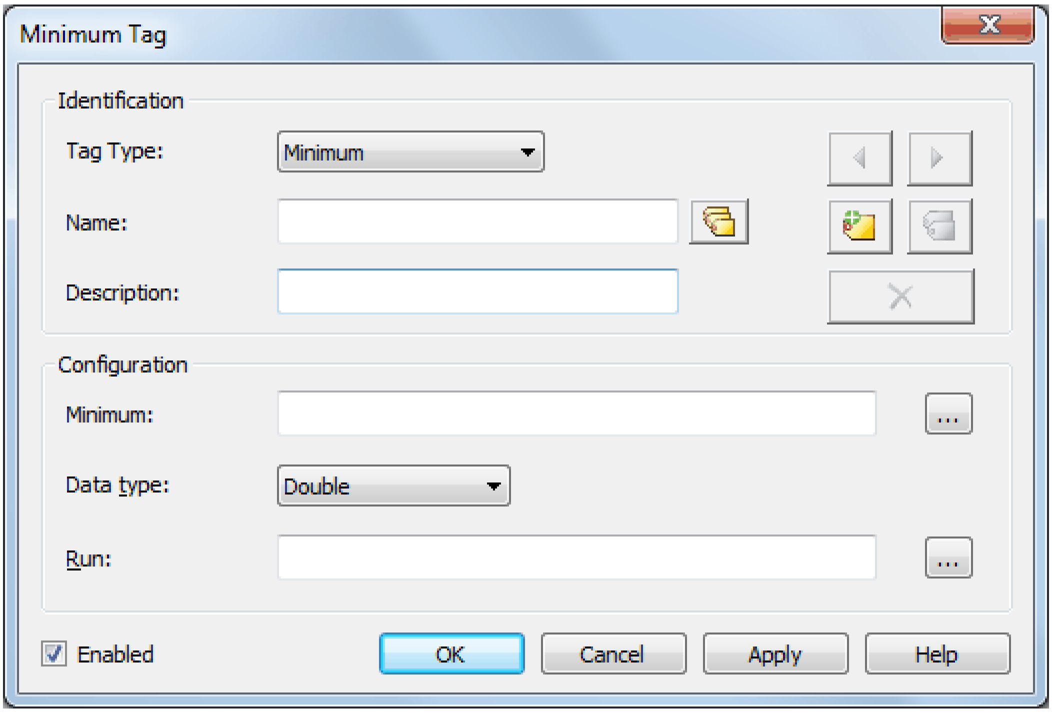
Note 1: If the quality of the Maximum Tag or the Trigger Tag is bad, the quality of the Maximum Tag will also be bad.

Note 2: If the quality of the Minimum Tag or the Trigger Tag is bad, calculation of the minimum value will stop. In order for calculation of the minimum value to restart, both tags must return to good quality and the Trigger Tag must successfully transition from zero to non-zero.

Minimum Tags

When enabled, Minimum Tags calculate the minimum value of a tag's value. Determining the Minimum value is initiated by the value of a Run tag transitioning from zero to non-zero. The rate that the tag is updated from the server with a new value comes from the update/scan rate on the device tag in the server. The default scan rate is 100 milliseconds. When disabled, Minimum Tags do no work and have a value of 0.

Note: Array tags are not supported.



Descriptions of the Configuration parameters are as follows:

- **Minimum:** This parameter specifies the tag from the server for which a minimum value will be calculated.
- **Data Type:** This parameter specifies the data type of the Minimum Tag.

  Note: Although Double is the only data type available in the Tag Properties, users can request a different data type in the OPC Client. The server will take ownership for coercing the data.
- **Run:** The Run Tag will be monitored to determine if it is appropriate to calculate the minimum value of the Minimum Tag. Whenever the Run Tag's value transitions from a zero to a non-zero value, a new minimum value will be calculated for the Minimum Tag. The calculation determines whether or not the currently scanned minimum value is lower than the reported minimum value. If it is, it will replace the reported value with the scanned value. When the Run Tag's value transitions back to zero, the last minimum value will be maintained and reported to the client application.

Examples

A tag that measures the temperature at a location (with a scan rate of 60000 milliseconds) would update the Minimum Tag with the following values every minute:

72.3 @ 10:00
72.5 @ 10:01
72.9 @ 10:02
73.4 @ 10:03
72.4 @ 10:04
71.6 @ 10:05
71.2 @ 10:06
70.5 @ 10:07

If the Trigger Tag had successfully transitioned from 0 to 1, the Minimum Tag would show the following values:

72.3 @ 10:00
72.3 @ 10:01
72.3 @ 10:02
72.3 @ 10:03
72.3 @ 10:04
71.6 @ 10:05
71.2 @ 10:06
70.5 @ 10:07

Note 1: If the quality of the Minimum Tag or the Trigger Tag is bad, the quality of the Minimum Tag will also be bad.

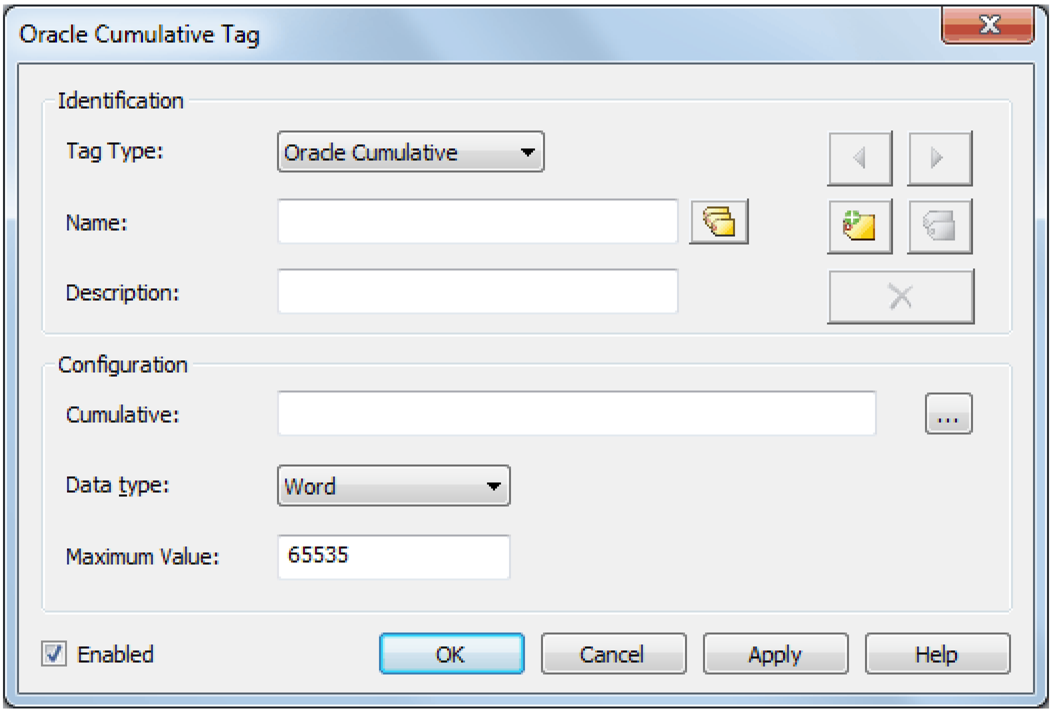
Note 2: If the quality of the Minimum Tag or the Trigger Tag is bad, calculation of the minimum value will stop. In order for calculation of the minimum to restart, both tags must return to good quality and the Trigger Tag must successfully transition from zero to non-zero.

Oracle Cumulative

Oracle Cumulative Tags are specifically designed for Oracle requirements. When enabled, they return the delta between the last value polled and the current value polled from a specific tag. When disabled, they do no work and have a value of 0.

On the first update, the raw value will be sent to the client. Each successive value will take the cumulative value and send it to the client. Values that cannot be read and that are considered bad quality will be skipped and the next good value will be used. To account for rollover, enter the maximum raw value that can be read in order to calculate the cumulative value appropriately.

Note: Array tags are not supported.


Oracle Cumulative Tag

Identification

Tag Type: Oracle Cumulative

Name:

Description:

Configuration

Cumulative:

Data type: Word

Maximum Value: 65535

☑ Enabled

Descriptions of the Configuration parameters are as follows:

- **Cumulative:** This parameter specifies the tag from the server for which the cumulative value will be calculated.
- **Data Type:** This parameter specifies the data type of the result, which will be the value of this tag.
- **Maximum Value:** The Maximum Value should be set to equal the maximum value of the tag being used. For example, if the tag is a counter with a max value of 1000, the Cumulative Tag Maximum Value property should be set to 1000.

Examples

Assume that in calculating the cumulative value for a counter that ranges from 0-100 (set the maximum value to 100), the following samples were read:

5 @ 10:00
45 @ 10:01
95@ 10:02
4 @ 10:03

The server would report the following:

5 @ 10:00 (The first Read is the raw value)
40 @ 10:01 (45-5)
50 @ 10:02 (95-45)
9 @ 10:03 (100-95+4)

Arithmetic Operators

Arithmetic Operators perform a math process. The result of that process is returned as the value of the Derived tag.

Important: Arithmetic Operators are specific to Derived Tags.

Tag Name	Formula	Result
Add*	*TAG(<Tagname>)+TAG(<Tagname>)*	For Numeric Tags, the result of this operation returns the sum of the tag values in the expression. For String Tags, the result of this operation concatenates all of the strings together.
Subtract	*TAG(<Tagname>)-Tag(<Tagname>)*	The result of this operation returns the difference of the tag values in the expression.

Multiply	*TAG(<Tagname>)*Tag(<Tagname>)*	The result of this operation returns the product of the tag values in the expression.
Divide	*TAG(<Tagname>)/Tag(<Tagname>)*	The result of this operation returns the divisor of the tag values in the expression.
Modulo	*TAG(<Tagname>)%Tag(<Tagname>)*	The result of this operation returns the remainder after division.

*At this time, Numeric and String Tag data types cannot be mixed in the same operation.

Examples

Tag Name	Formula	Result
Add	*TAG(Sim.Device.Add1)+TAG (Sim.Device.Add2)*	If Add1=6 and Add2=71, the result is 77.
Subtract	*TAG(Sim.Device.Subtract1)-TAG (Sim.Device.Subtract2)*	If Subtract1 is 5 and Subtract2 is 10, the result is -5.
Multiply	*TAG(Sim.Device.Multiply1)*TAG (Sim.Device.Multiply2)*	If Multiply1=3 and Multiply2=4, the result is 12.
Divide	*TAG(Sim.Device.Divide1)/TAG (Sim.Device.Divide2)*	If Divide1=5 and Divide2=2, the result is 2.5.
Modulo	*TAG(Sim.Device.Mod1)%TAG (Sim.Device.Mod2)*	If Mod1=5 and Mod2=2, the result is 1.

Additional Arithmetic Expression Examples

Complex Math Polynomial

(TAG(Sim.Device.Polynomial-Coefficient1)(TAG(Sim.Device.Polynomial-X)*TAG (Sim.Device.Polynomial-X)))*

+

*(TAG(Sim.Device.Polynomial-Coefficient2)*TAG(Sim.Device.Polynomial-X))*

-

TAG(Sim.Device.Polynomial-Coefficient3)

For this example, assume the following:

Polynomial - Coefficient1 = 6
Polynomial - Coefficient2 = 10
Polynomial - Coefficient3 = 4
Polynomial - CoefficientX = 2

Plugging those values into the equation yields a result of 40.

(6*(2*2)) + (10*2) - 4 = 40

Functions

Functions are used to perform a specific calculation on a tag values or expressions. The result of that process is returned as the value of the Derived tag.

Important: All Functions are specific to Derived Tags.

Tag Name	Formula	Result
Absolute Value	*ABS(TAG(<Tagname>))*	This function returns the absolute value of the tag value or expression within the expression.
Arc Cosine	*ACOS(TAG(<Tagname>))*	This function is the inverse of the Cosine function and returns the result as a number with a value range of 0 to 180. Input value range is -1.0 to

		+1.0.
Arc Sine	*ASIN(TAG(<Tagname>))*	This function is the inverse of the Sine function and returns the result as a number with a value range of -90.0 to 90.0. Input value range is -1.0 to 1.0.
Arc Tangent	*ATAN(TAG(<Tagname>))*	This function is the inverse Tangent function and returns the result as a number with the range of -90.0 to 90.0. There is no input range limit.
Cosine	*COS(TAG(<Tagname>))*	This function calculates the Cosine of the tag value. The result is a number with the range of -1.0 to 1.0. Although there is no true input limit, the Sine input is an angle. The assumed range is 0.0 to 360.0 or -180.0 to 180.0.

Power	*POW(TAG(<Tagname>),TAG(<Tagname>))*
	POW(Base,Exponent)
		This function calculates the result as the base raised to the exponent.

Sine	*SIN(TAG(<Tagname>))*	This function calculates the Sine of tag value. The result is a number with the range of -1.0 to 1.0. Although there is no true input limit, the Sine input is an angle. The assumed range is 0.0 to 360.0 or -180.0 to 180.0.
Square Root	*SQRT(TAG(<Tagname>))*	This function returns the Square Root of the tag value as the result.
Tangent	*TAN(TAG(<Tagname>))*	This function calculates the Tangent of the tag value and outputs a number ranging for negative to positive infinity. The input range is any derivative of -90.0 to 90.0.

Examples

Tag Name	Formula	Result

Absolute Value	*ABS(TAG(Sim.Device.Abs1))*
	ABS(TAG(Sim.Device.Subtract1)-TAG (Sim.Device.Subtract2))
		If Abs1=-57, the result is 57.
		If Subtract1 is 5 and Subtract2 is 10, the result of the math is -5 and the result of the Absolute value function is 5.

Arc Cosine	*ACOS(TAG(Sim.Device.ACos1))*	If ACos1=-0.5, the result is 120.
Arc Sine	*ASIN(TAG(Sim.Device.ASin1))*	If Asin1=.707107, the result is 45. If it is -.707107, the result is -45.
Arc Tangent	*ATAN(TAG(Sim.Device.ATan1))*	If ATan1=-1.5, the result is -56.3099.

Cosine	*COS(TAG(Sim.Device.Cos1))*	If Cos1=180, the result is -1.
Power	*POW(TAG(Sim.Device.Pow1),TAG(Sim.Device.Pow2))*	If Pow1 is 9 and POW2 is 3, the result is 729.
Sine	*SIN(TAG(Sim.Device.Sin1))*	If Sin1=270 the result is -1.0, if it is 90 the result is 1.0.
Square Root	*SQRT(TAG(Sim.Device.Sqrt1))*	If Sqrt1=225, the result is 15.
Tangent	*TAN(TAG(Sim.Device.Tan1))*	If Tan1 is 22.5, the result is 0.414214.

Logic Tags

Logic tags are used to evaluate an expression to be either True or False and return that result as the value of the tag. All Logic Derived tags will be a Boolean data type. A Boolean True can be a 1 or -1 depending upon the client application; a Boolean False will be 0.

Important: All Logic Tags are specific to Derived Tags.

Tag Name	Formula	Result
And	*TAG(<Tagname>)AND TAG(<Tagname>)*	This function evaluates the each tag to see if it is value is greater than 0 returning the result as True or False.
Equal To	*TAG(<Tagname>)==TAG (<Tagname>)*	This function compares the 2 values to see if they are the same.
Greater Than	*TAG(<Tagname>)>TAG (<Tagname>)*	This function tests to see of one value is greater than the other.
Less Than	*TAG(<Tagname>)<TAG (<Tagname>)*	This function tests to see if one value is less than the other.
Not	*NOT(TAG(<Tagname>))*	This function test to see if the Tag value or Expression is not True.
Or	*TAG(<Tagname>)OR TAG (<Tagname>)*	This function tests to see if any of the values in the expression are True.

Examples

Tag Name	Formula	Result
And	*TAG(Sim.Device.And1)AND TAG (Sim.Device.And2)*	If And1=0 and And2=97, the result is False. If And1=1 and And2=1, the result is True.
Equal To	*TAG(Sim.Device.EqualTo1)==TAG (Sim.Device.EqualTo2)*	If EqualTo1=Test and EqualTo2=Test, the result is True.
Greater Than	*TAG(Sim.Device.GreaterThan1)>TAG (Sim.Device.GreaterThan2)*	If GreaterThan1=16 and GreaterThan2=17, the result is False.
Less Than	*TAG(Sim.Device.LessThan1)<TAG (Sim.Device.LessThan2)*	If Lessthan1=20 and LessThan2=41 the result is True.
Not	*NOT(TAG(Sim.Device.EqualTo1)==TAG (Sim.Device.EqualTo2))*	If EqualTo1=Test and EqualTo2=Test, the result is False.
Or	*TAG(Sim.Device.Or1)OR TAG(Sim.Device.Or2)*	If Or1=0 and Or2=1 the result is True. If Or1=0 and OR2=0 the result is False.

Additional Logic Expression Examples

Tag Name	Formula	Result
Greater Than or Equal To	*TAG(Sim.Device.GreaterThanOrLessThan1) >=TAG (Sim.Device.GreaterThanOrLessThan2)*	If GreaterThanOrLessThan1=17 and GreaterThanOrLessThan2=17, the result is True. If GreaterThanOrLessThan1=16, the result is False.
IsTrue	*TAG(Sim.Device.IsTrue1)==True*	If IsTrue=0 then the result is Result is

		False.
IsFalse	*TAG(Sim.Device.IsFalse1)==FALSE*	If IsFalse=1 then the result is False.
Less Than or Equal To	*TAG(Sim.Device.LessThanOrGreaterThan1)<= TAG(Sim.Device.LessThanOrGreaterThan2)*	If LessThanOrGreaterThan1=0 and LessThanOrGreaterThan2=1000 the result is True.

Error Descriptions

The following error/warning messages may be generated. Click on the link for a description of the message.

Advanced Tag Error Messages

References to array tags are not supported '<tag name>'
The complex tag already references '<tag name>' and will not be re-added
The input and output tags cannot be the same
The tag must be writeable
The trigger and output tags cannot be the same
Unable to start complete tag '<tag name>' for element '<element>' on tag '<tag name>'
Unable to start complete tag '<tag name>' on tag '<tag name>'
Unable to start element tag '<tag name>' on tag '<tag name>'
Unable to start required tag reference '<reference>' on tag '<tag name>'
Unable to start tag '<tag name>' [Expression: '<expression>']
Unable to start trigger tag '<tag name>' for element '<element>' on tag '<tag name>'
Unable to start trigger tag '<tag name>' on tag '<tag name>'

CSV Error Messages

Error importing CSV Complex Element record '<record index>'. Insert by '<value>' is invalid. Insert by set to 'Rate'
Error importing CSV Complex Element record '<record index>'. Update rate '<value><units>' is out of range. Update rate has been set to '<value><units>'
Error importing CSV tag '<tag name>'. Comparison '<comparison>' is invalid. Comparison set to 'Tag == Value'
Error importing CSV tag '<tag name>'. Data type '<data type>' is not valid for the tag type. Data type has been changed to '<data type>'
Error importing CSV tag '<tag name>'. Maximum value '<max value>' is out of range. Maximum value has been set to '<max value>'
Error importing CSV tag '<tag name>'. Run by '<value>' is invalid. Run by set to 'Rate'
Error importing CSV tag '<tag name>'. Run by rate '<value><units>' is out of range. Run by rate has been set to '<value><units>'
Error importing CSV tag '<tag name>'. Trigger mode '<trigger mode>' is invalid. Trigger mode set to 'On Input Value Change'
Error importing CSV tag '<tag name>'. Update by '<update rate>' is invalid. Update by set to '<update rate>'
Error importing CSV tag '<tag name>'. Update rate '<value><units>' is out of range. Update rate has been set to '<value><units>'
Error importing CSV tag data. No advanced tag records found in CSV file

Derived Expression Error Messages

", expected
(expected
) expected
Expression modified
Numeric expression expected
Please enter an expression
String tag expected
Syntax error: '<syntax string>'
Unknown tag or misspelled keyword: '<tag name>'
Unrepresentable numeric constant: '<tag name>'
Unterminated string

Advanced Tag Error Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Advanced Tag Error Messages

References to array tags are not supported '<tag name>'
The complex tag already references '<tag name>' and will not be re-added
The input and output tags cannot be the same
The tag must be writeable
The trigger and output tags cannot be the same

References to array tags are not supported '<tag name>'

Error Type:
Warning

Possible Cause:
An attempt was made to address an array tag from an Advanced Tag.

Solution:
Choose a tag is that is not an array.

The complex tag already references '<tag name>' and will not be re-added

Error Type:
Warning

Possible Cause:
An attempt was made to add a tag to the Complex Tag Element List where it was already referenced.

Solution:
Choose a tag to reference that is not already being referenced.

The input and output tags cannot be the same

Error Type:
Warning

Possible Cause:
The Input and Output tags for a Link Tag are the same.

Solution:
Do not use the same tag as both the Input and Output tag. This creates an invalid circular link.

The tag must be writeable

Error Type:
Warning

Possible Cause:
1. The Output Tag for a Link Tag is not writeable.
2. The Complete Tag for a Derived or Complex Tag is not writeable.

Solution:
Make sure that the tag is writeable. Links require writing to the Output Tag; Derived and Complex tags require writing to the Complete Tag.

The trigger and output tags cannot be the same

Error Type:
Warning

Possible Cause:
The Trigger and Output tags for a Link Tag are the same.

Solution:
Do not use the same tag as both the Trigger and Output tag. This can create an invalid self-triggering Link Tag.

Unable to start complete tag '<tag name>' for element '<element>' on tag '<tag name>'

Error Type:
Warning

Possible Cause:
The tag may reference another tag that may have been deleted or renamed.

Solution:
1. Add the tag back into the project.
2. Reference the correct tag.
3. Correct the tag name.

Unable to start complete tag '<tag name>' on tag '<tag name>'

Error Type:
Warning

Possible Cause:
The tag may reference another tag that may have been deleted or renamed.

Solution:
1. Add the tag back into the project.
2. Reference the correct tag.
3. Correct the tag name.

Unable to start element tag '<tag name>' on tag '<tag name>'

Error Type:
Warning

Possible Cause:
The tag may reference another tag that may have been deleted or renamed.

Solution:
1. Add the tag back into the project.
2. Reference the correct tag.
3. Correct the tag name.

Unable to start required tag reference '<reference>' on tag '<tag name>'

Error Type:
Warning

Possible Cause:
The tag may reference another tag that may have been deleted or renamed.

Solution:
1. Add the tag back into the project.
2. Reference the correct tag.
3. Correct the tag name.

Unable to start tag '<tag name>' [Expression: '<expression>']

Error Type:
Warning

Possible Cause:
The tag may reference another tag that may have been deleted or renamed.

Solution:
1. Add the tag back into the project.
2. Reference the correct tag.
3. Correct the tag name.

Unable to start trigger tag '<tag name>' for element '<element>' on tag '<tag name>'

Error Type:
Warning

Possible Cause:
The tag may reference another tag that may have been deleted or renamed.

Solution:
1. Add the tag back into the project.
2. Reference the correct tag.
3. Correct the tag name.

Unable to start trigger tag '<tag name>' on tag '<tag name>'

Error Type:
Warning

Possible Cause:
The tag may reference another tag that may have been deleted or renamed.

Solution:
1. Add the tag back into the project.
2. Reference the correct tag.
3. Correct the tag name.

CSV Error Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

CSV Error Messages

Error importing CSV Complex Element record '<record index>'. Insert by '<value>' is invalid. Insert by set to 'Rate'
Error importing CSV Complex Element record '<record index>'. Update rate '<value><units>' is out of range. Update rate has been set to '<value><units>'
Error importing CSV tag '<tag name>'. Comparison '<comparison>' is invalid. Comparison set to 'Tag == Value'
Error importing CSV tag '<tag name>'. Data type '<data type>' is not valid for the tag type. Data type has been changed to '<data type>'
Error importing CSV tag '<tag name>'. Maximum value '<max value>' is out of range. Maximum value has been set to '<max value>'
Error importing CSV tag '<tag name>'. Run by '<value>' is invalid. Run by set to 'Rate'
Error importing CSV tag '<tag name>'. Run by rate '<value><units>' is out of range. Run by rate has been set to '<value><units>'
Error importing CSV tag '<tag name>'. Trigger mode '<trigger mode>' is invalid. Trigger mode set to 'On Input Value Change'
Error importing CSV tag '<tag name>'. Update by '<update rate>' is invalid. Update by set to '<update rate>'
Error importing CSV tag '<tag name>'. Update rate '<value><units>' is out of range. Update rate has been set to '<value><units>'
Error importing CSV tag data. No advanced tag records found in CSV file

Error importing CSV Complex Element record '<record index>'. Insert by '<value>' is invalid. Insert by set to 'Rate'

Error Type:
Warning

Possible Cause:
The Insert by action is invalid.

Solution:
Change the Insert by action to one that is valid (such as Rate or Trigger).

Error importing CSV Complex Element record '<record index>'. Update rate '<value><units>' is out of range. Update rate has been set to '<value><units>'

Error Type:
Warning

Possible Cause:
The Complex Tag Element's update rate is out of range. The provided record index is the position of the Complex Tag Element record in the Complex Tag Elements section of the CSV file.

Solution:
Change the update rate to one within the valid range.

Note:
The valid range is 10 milliseconds to 49 days.

Error importing CSV tag '<tag name>'. Comparison '<comparison>' is invalid. Comparison set to 'Tag == Value'

Error Type:
Warning

Possible Cause:
The comparison is invalid.

Solution:
Change the comparison to one that is valid.

Note:
Valid comparisons are as follows: Tag == Value, Tag != Value, Tag > Value, Tag >= Value, Tag < Value, and Tag <= Value.

Error importing CSV tag '<tag name>'. Data type '<data type>' is not valid for the tag type. Data type has been changed to '<data type>'

Error Type:
Warning

Possible Cause:
The tag's data type in the CSV file is not valid.

Solution:
Change the tag's data type to one that is valid.

Error importing CSV tag '<tag name>'. Maximum value '<max value>' is out of range. Maximum value has been set to '<max value>'

Error Type:
Warning

Possible Cause:
The tag's maximum value is out of range.

Solution:
Change the tag's maximum value to one within the valid range.

Note:
The valid range is the range that can be represented by the tag's data type.

Error importing CSV tag '<tag name>'. Run by '<value>' is invalid. Run by set to 'Rate'

Error Type:
Warning

Possible Cause:
The Run by action is invalid.

Solution:
Change the Run by action to one that is valid (such as Rate or Trigger).

Error importing CSV tag '<tag name>'. Run by rate '<value><units>' is out of range. Run by rate has been set to '<value><units>'

Error Type:
Warning

Possible Cause:
The tag's Run by rate is out of range.

Solution:
Change the tag's Run by rate to one within the valid range.

Note:
The valid range is 10 milliseconds to 49 days.

Error importing CSV tag '<tag name>'. Trigger mode '<trigger mode>' is invalid. Trigger mode set to 'On Input Value Change'

Error Type:
Warning

Possible Cause:
The trigger mode is invalid.

Solution:
Change the trigger mode to one that is valid.

Note:
The valid modes are On Input Value Change, On Timer, and On Value.

Error importing CSV tag '<tag name>'. Update by '<update rate>' is invalid. Update by set to '<update rate>'

Error Type:
Warning

Possible Cause:
The Update by rate is out of range.

Solution:
Change the Update by rate to one within the valid range.

Note:
The valid range is 50 to 1800000000.

Error importing CSV tag '<tag name>'. Update rate '<value><units>' is out of range. Update rate has been set to '<value><units>'

Error Type:
Warning

Possible Cause:

The tag's update rate is out of range.

Solution:
Change to the tag's update rate to one within the valid range.

Note:
The valid range is 10 milliseconds to 49 days.

Error importing CSV tag data. No advanced tag records found in CSV file

Error Type:
Warning

Possible Cause:
The CSV file being imported does not contain any valid advanced tags records.

Solution:
Verify the CSV file's format and contents.

See Also:
Advanced Tags CSV Import and Export

Derived Expression Error Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Derived Expression Error Messages

", expected

Error Type:
Warning

Possible Cause:
A string was entered in an expression that is missing a double quote.

Solution:
1. Place a double quote at the begining.
2. Terminate the string.

(expected

Error Type:
Warning

Possible Cause:
1. A compound expression has been created that is missing an Open parenthesis (to pair the Close parenthesis).
2. An Open and Close parenthesis are next to each other without an operator between.

Solution:
1. Verify the expression and place the parentheses where they belong.
2. Place an operator between the parentheses.

) expected

Error Type:
Warning

Possible Cause:
1. A compound expression has been created that is missing a Close parenthesis (to pair the Open parenthesis).
2. An Open and Close parenthesis are next to each other without an operator between.

Solution:
1. Verify the expression and place the parentheses where they belong.
2. Place an operator between the parentheses.

Expression modified

Error Type:
Information

Possible Cause:
The saved expression was modified.

Solution:
Accept the modification so that it will keep.

Numeric expression expected

Error Type:
Warning

Possible Cause:
A string expression was created for a numeric Derived Tag.

Solution:
Create a numeric expression.

Please enter an expression

Error Type:
Warning

Possible Cause:
The expression string field was left blank.

Solution:
Enter an expression.

String tag expected

Error Type:
Warning

Possible Cause:
1. An incorrect Math or Logic operator was used.
2. An attempt was made to create a string tag, and a non-string tag was referenced as part of the expression.

Solution:
1. Use the correct operator.
2. Use a string tag reference rather than a numeric tag reference.

Syntax error: '<syntax string>'

Error Type:
Warning

Possible Cause:

The expression failed a syntax check.

Solution:
Verify the syntax, including the placement of all parentheses.

Unknown tag or misspelled keyword: '<tag name>'

Error Type:
Warning

Possible Cause:
1. A tag was referenced that does not exist in the server project.
2. A tag was referenced that has been deleted from the project.
3. A keyword was misspelled.
4. An invalid keyword was entered.

Solution:
1. Create the referenced tag.
2. Reference a tag that exists.
3. Enter the correctly spelled keyword.

Note:
For the proper syntax, refer to the server's help documentation.

Unrepresentable numeric constant: '<tag name>'

Error Type:
Warning

Possible Cause:
A constant was used in the expression that is either not numeric or cannot be converted to a numerical value.

Solution:
Change or redefine the constant so that it can be used.

Unterminated string

Error Type:
Warning

Possible Cause:
1. A string was entered with no termination
2. A string tag was referenced with no termination point.

Solution:
1. Terminate string constants. To do so, enclose any strings in the expression with double quotes.
2. Use only terminated string tags.

Index