Advanced Tags

© 2026 PTC Inc. All Rights Reserved.

Table of Contents

Advanced Tags ... 1
Table of Contents ... 2
Help Contents ... 4
Overview ... 4
Plug-In Interface ... 5
Advanced Tag Groups ... 6
Configuration Group ... 6
Advanced Tag Browser ... 7
Average Tags ... 8
Complex Tags ... 9
Derived Tags ... 11
Link Tags ... 13
Link Behavior Matrix ... 15
Maximum Tags ... 16
Minimum Tags ... 17
Oracle Cumulative ... 19
Arithmetic Operators ... 20
Functions ... 21
Logic Tags ... 22
Optimizing Performance ... 23
CSV Import / Export ... 24
Error Message Descriptions ... 28
Advanced Tag Messages ... 28
<Advanced tag name> is invalid. Verify the tag references valid server tags and does not contain any references to itself or to an unsupported or array data type. ... 28
<Tag name> has an invalid expression: <expression error>. ... 28
Failed to write to tag <tag name> from link tag <link tag name>. ... 29
References to array tags with an unsupported or array data type are not allowed <tag name>. ... 29
The complex tag already references <tag name> and will not be re-added. ... 29
The input and output tags cannot be the same. ... 29
The tag must be writeable. ... 30
The trigger and output tags cannot be the same. ... 30
Unable to acquire licensing for Advanced Tags - Type <advanced tag type>. ... 30
Unable to start complete tag <tag name> for element <element> on tag <advanced tag name>. ... 30
Unable to start complete tag <tag name> on tag <advanced tag name>. ... 30
Unable to start element tag <tag name> on tag <advanced tag name>. ... 31
Unable to start required tag reference <reference> on tag <tag name>. ... 31
Unable to start required tag reference <reference> in expression <expression>. ... 31
Unable to start trigger tag <tag name> for element <element> on tag <advanced tag name>. ... 31
Unable to start trigger tag <tag name> on tag <advanced tag name>. ... 32
CSV Messages ... 32

Error importing CSV complex element record <record index>. Insert by <value> is invalid. Insert by set to 'Rate'. 33
Error importing CSV complex element record <record index>. Update rate <value><units> is out of range. Update rate has been set to <value><units> 33
Error importing CSV tag <tag name>. Comparison <comparison> is invalid. Comparison set to Trigger Tag == Value. 33
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>. 33
Error importing CSV tag <tag name>. Maximum value <max value> is out of range. Maximum value has been set to <max value>. 34
Error importing CSV tag <tag name>. Run by <value> is invalid. Run by set to 'Rate'. 34
Error importing CSV tag <tag name>. Run by rate <value><units> is out of range. Run by rate has been set to <value><units>. 34
Error importing CSV tag <tag name>. Trigger <trigger mode> is invalid. Trigger mode set to On Data Change of Input Tag. 34
Error importing CSV tag <tag name>. Trigger scan rate <value> <units> is out of range. Trigger scan rate has been set to <value> <units>. 35
Error importing CSV tag <tag name>. Link mode <link mode> is invalid. Link mode set to On Data Change of Input Tag. 35
Error importing CSV tag <tag name>. Trigger type <trigger type> is invalid. Trigger type set to Always. 35
Error importing CSV tag <tag name>. Update by <update rate> is invalid. Update by set to <update rate>. 35
Error importing CSV tag <tag name>. Trigger Mode <value> <units> is out of range. Link rate has been set to <value> <units>. 36
Error importing CSV tag <tag name>. Update rate <value><units> is out of range. Update rate has been set to <value><units>. 36
Error importing CSV tag data. No advanced tag records found in CSV file. 36
Derived Expression Messages 37
", expected. 37
(expected. 37
) expected. 37
Expression modified. 38
Numeric expression expected. 38
Please enter an expression. 38
String tag expected. 38
Syntax error: <syntax string>. 38
Unknown tag or misspelled keyword: <tag name>. 39
Unrepresentable numeric constant: <tag name>. 39
Unterminated string. 39
Index 40

Help Contents

CONTENTS

Overview
What are Advanced Tags?

Advanced Tag Groups
How are Advanced Tag Groups used?

Optimizing Performance
What performance can I expect?

Error Descriptions
What error messages can Advanced Tags produce?

Help version 1.101
© 2026 PTC Inc. All Rights Reserved.

Overview

Advanced Tags enables Machine to Machine tag linking, logic, and math functions for operational communications and analysis. It can link two data tags, set a trigger based on logical states, and calculate new values from raw measures. Executing math, logic, or analysis at the connectivity platform level brings data closer to the source.

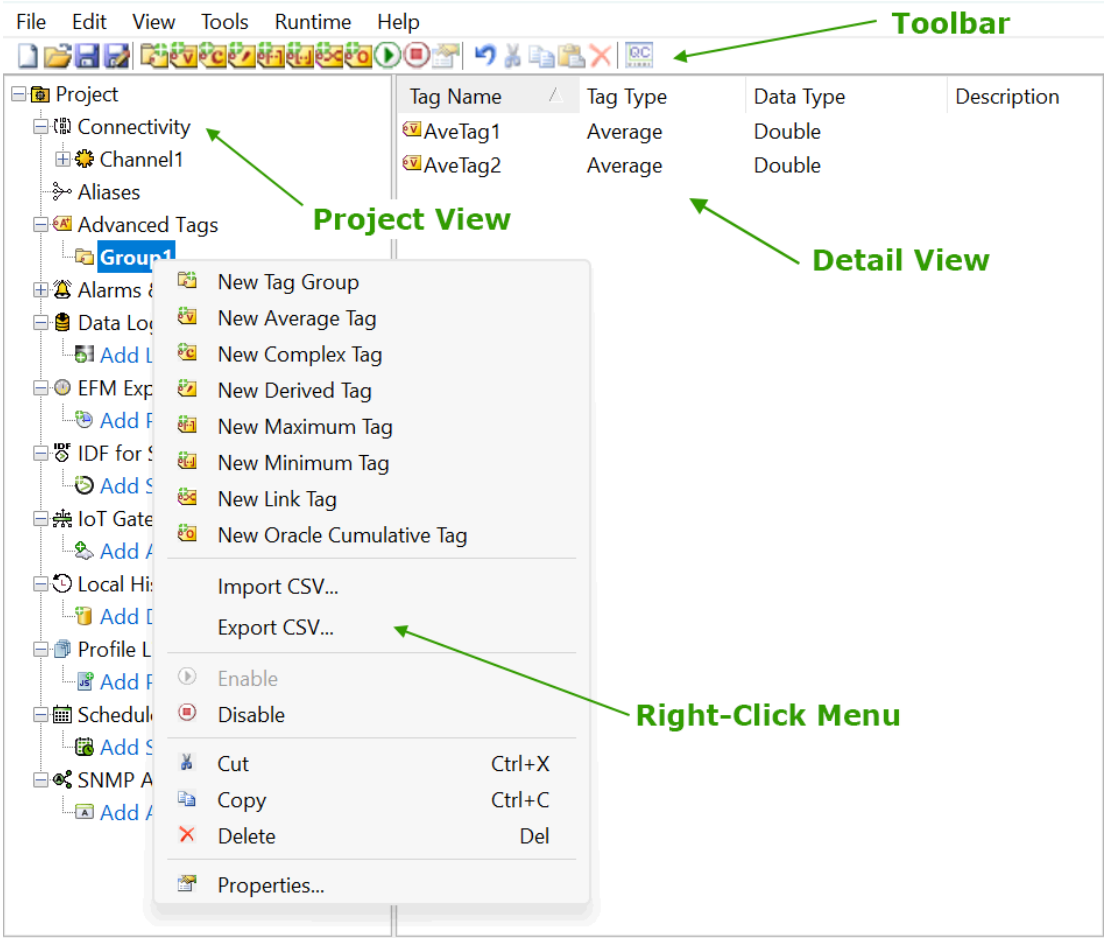
Advanced Tags provides users the tools needed for proper operational analysis, which requires that industrial control applications deliver an array of data that often needs some form of calculation to identify deficiencies and determine whether operations are functioning correctly. By offering easy-to-use, pre-defined tags to execute math, logic, or analysis at the tag level, Advanced Tags help users improve decision-making and efficiency in their industrial control applications.

The following advanced tag types are supported:
Average Tags
Complex Tags
Derived Tags
Link Tags
Maximum Tags
Minimum Tags
Oracle Cumulative Tags

See Also: Advanced Tag Groups, Optimizing Performance

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**.



Tip: 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.

Detail View

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

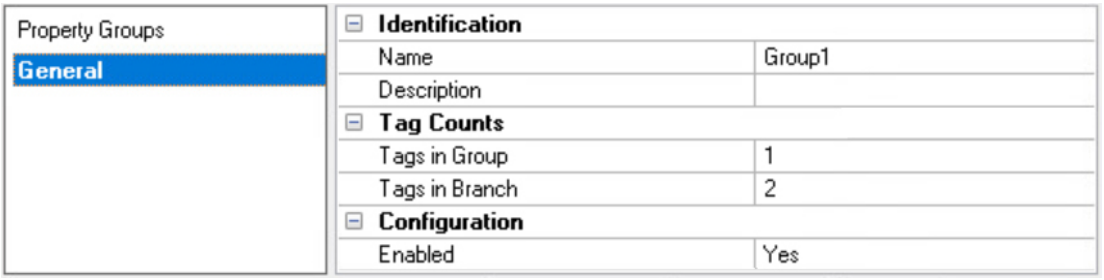
Toolbar

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

Tip: Advanced Tags are also accessible through Kepware+. *For more information, see Advanced Tags in the Kepware+ help center.*

Advanced Tag Groups

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



Identification

Name: Specify or verify the name of the tag group.
Tip: The name cannot start with an underscore and must be unique among sibling tag groups. It can be up to 256 characters in length.

Description: Specify information or an explanatory comment for the group. The description is a string of up to 255 characters.

Tag Counts

Tags in Group: Review the number of tags assigned to the current tag group.

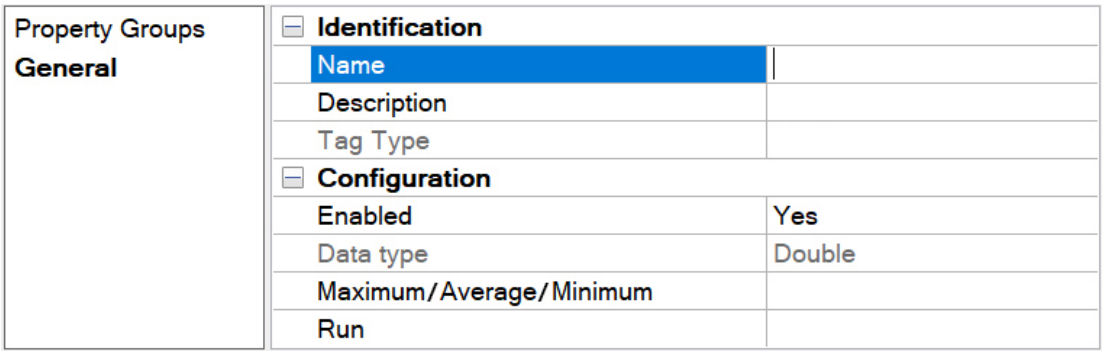
Tags in Branch: Review the number of tags assigned to the current tag group and all descendant tag groups.

Configuration

Enabled: Specifies the state of the tag group. If a tag group is disabled, all child tags and tag groups are also disabled.

Configuration Group

All tag types include the common property groups of Identification and Configuration with some common and some type-specific properties



Identification

Name: Specify (or edit) the name for the tag. The name can be up to 256 characters in length and cannot contain periods, contain double quotes, or start with an underscore.

Description: Specify information or an explanatory comment for the tag. The description is a string of up to 255 characters.

Tag Type: Verify the tag type (selected on creation of the tag; not editable).

Configuration

Enabled: Specify if this tag is currently enabled / active. Tags that are enabled have references to other server tags and do computational work. Tags that are disabled have no references to server tags.

Data Type: Verify the data type of the tag.

Average / **Maximum** / **Minimum**: Specify (or locate) the tag from the server for which this action will occur *(see specific tag types below)*.

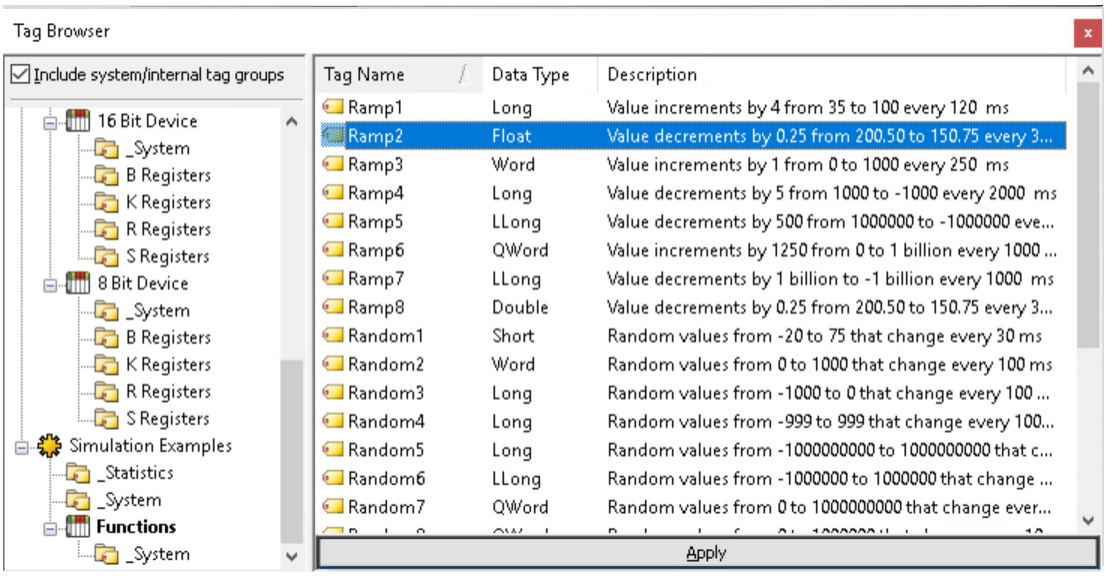
Run: Specify (or locate) the tag to monitor to determine if it is appropriate to calculate the value of the tag.

For more information, consult the tag type specific pages:

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

Advanced Tag Browser

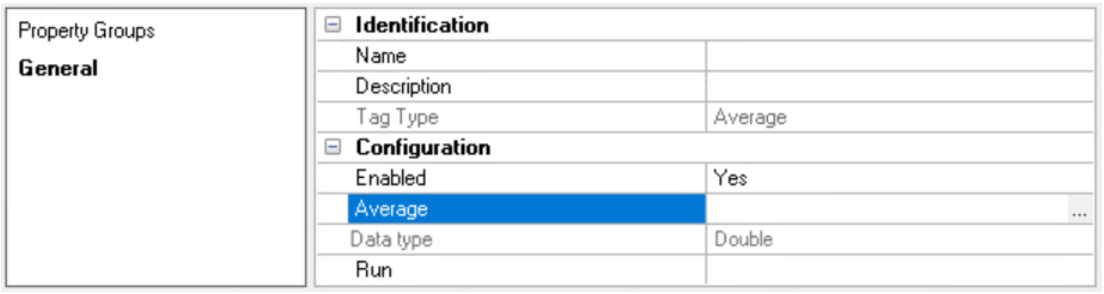
The Tag Browser can be invoked whenever a tag should be selected. It allows you to navigate the groups and heirarchies in your environment to find and select the best tag for your purpose. Use the tree in the left pane to traverse channels, features, and groups; use the right list pane to select the tag; click **Apply**.



Average Tags

When enabled, Average Tags calculate an average of a tag's value over time. Averaging is initiated by the value of a Run Tag transitioning from zero to non-zero. The rate that the Average Tag is updated from the server with a new value comes from the update / scan rate assigned to the tag to be averaged. The default scan rate is 100 milliseconds.

Note: Array tags are not supported.



Identification

Name: Specify (or edit) the name for the tag. The name can be up to 256 characters in length and cannot contain periods, contain double quotes, or start with an underscore.

Description: Specify information or an explanatory comment for the tag. The description is a string of up to 255 characters.

Tag Type: Verify the tag type (selected on creation of the tag; not editable).

Configuration

Enabled: Specify if this tag is currently enabled / active. Tags that are enabled have references to other server tags and do computational work. Tags that are disabled have no references to server tags.

Average: Specify (or locate) the tag from the server for which an average value is calculated.

Data Type: Verify the data type of the 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 takes ownership for coercing the data.

Run: Specify (or locate) the tag to monitor to determine if it is appropriate to calculate the average value of the tag. If the Run Tag's value is zero when the Average Tag is first requested by a client application, the Average Tag has no value. Whenever the Run Tag's value transitions from zero to a non-zero value, the Average Tag begins calculating a new average. The frequency of the calculation is based on the scan rate of the Average Tag, so intermediate average values are reported to client applications. When the Run Tag's value transitions back to zero, the last average value is 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 successfully transitioned from zero to one, the Average Tag would show the following values:

72.3 @ 10:00

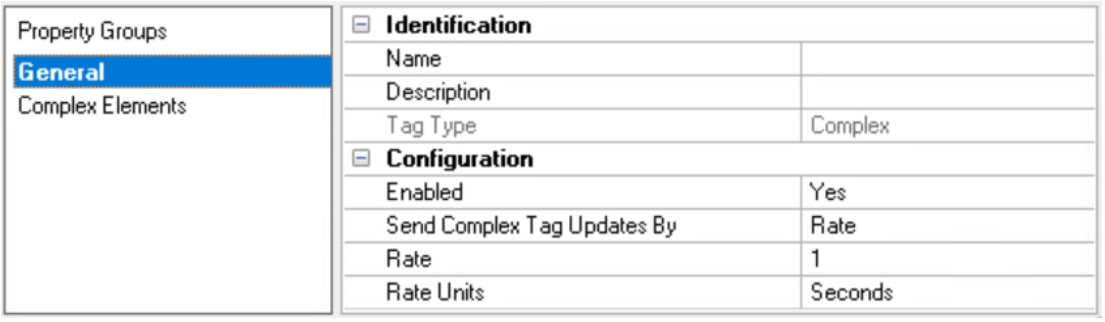
72.4 @ 10:01
72.6 @ 10:02
72.8 @ 10:03
72.7 @ 10:04
72.6 @ 10:05
72.3 @ 10:06
72.1 @ 10:07

Note: If the quality of the Run Tag or the tag being averaged is bad, the quality of the Average Tag is also bad and averaging stops. To restart, both tags must return to good quality.

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.

Note: Array tags are not supported.



Identification

Name: Specify (or edit) the name for the tag. The name can be up to 256 characters in length and cannot contain periods, contain double quotes, or start with an underscore.

Description: Specify information or an explanatory comment for the tag. The description is a string of up to 255 characters.

Tag Type: Verify the tag type (selected on creation of the tag; not editable).

Configuration

Enabled: Specify if this tag is currently enabled / active. Tags that are enabled have references to other server tags and do computational work. Tags that are disabled have no references to server tags.

Send Complex Tag Updates By: Specify whether complex tag updates are sent by Rate or Trigger. The default setting is Rate. Descriptions are as follows:

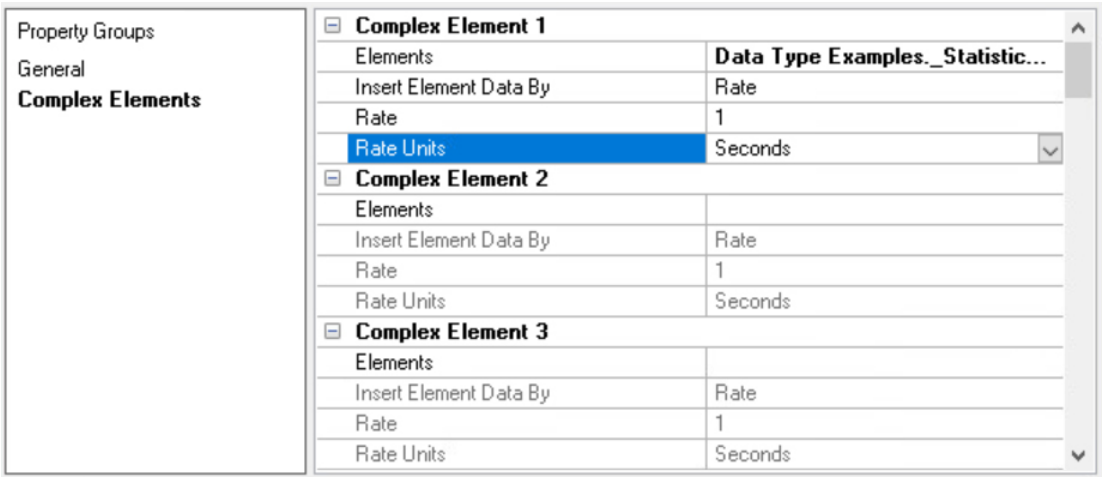
- **Rate**: Specify the frequency at which the Complex Tag updates are published. The default setting is 1 second.
  Note: The range depends on the unit. The range for milliseconds is 10 to 4,233,600,000. The range for seconds is 1 to 4233,600. The range for minutes is 1 to 70560. The range for hours is 1 to 1176. The range for days is 1 to 49. The default unit is seconds.
- **Trigger**: Specify the condition at which Complex Tag updates are published. There are two types: **Trigger** and **Complete**. The Trigger Tag's value is monitored for a zero to non-zero transition. When this transition occurs, it results in a Complex Tag update. No additional updates 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 writes a value of 1 to the Complete Tag (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 tags. If the tag has a Boolean data type, it indicates TRUE.

For other data types, the VARIANT_TRUE value is converted, resulting in -1 for signed data types or “max. value for type” for unsigned data types.

Complete: Specify the tag address where the final value should be recorded.
Note: Array tags cannot be used as Complete tags. Complete tags must have write access.

Complex Elements

The Complex Element properties are used to specify an element to insert into a complex tag. The element data may be inserted by Rate or Trigger. For descriptions of these options, refer to the descriptions above.



Complex Element *n* Specify (or locate) the tag that contains the element to use.

Insert Element Data By: Specify if the element is added by Rate or Trigger.

- **Rate**: Specify how often the expression is evaluated / run. The default setting is 1 second.
  - **Unit**: Specify the unit of time (default is seconds). The range for milliseconds is 10 to 4,233,600,000. The range for seconds is 1 to 4233,600. The range for minutes is 1 to 70560. The range for hours is 1 to 1176. The range for days is 1 to 49. The default unit is seconds.
- **Trigger**: Specify the condition at which the expression is evaluated / run. There are two types: **Trigger**and **Complete**. The tag defined in the Trigger field is monitored for a zero to non-zero transition. When this transition occurs, the expression is evaluated / run. No additional updates 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 writes a value of VARIANT TRUE 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.
  Note: If the tag has a Boolean data type, it indicates TRUE. For other data types, the VARIANT_TRUE value is converted, resulting in -1 for signed data types or “max. value for type” for unsigned data types.

Note: Up to 20 elements can be added to a single complex tag.

The tags return data as an XML string defined by the schema below.

```
<?xml version="1.0" encoding="utf-8"?>
 <xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element
 <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
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element
<xs:element
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

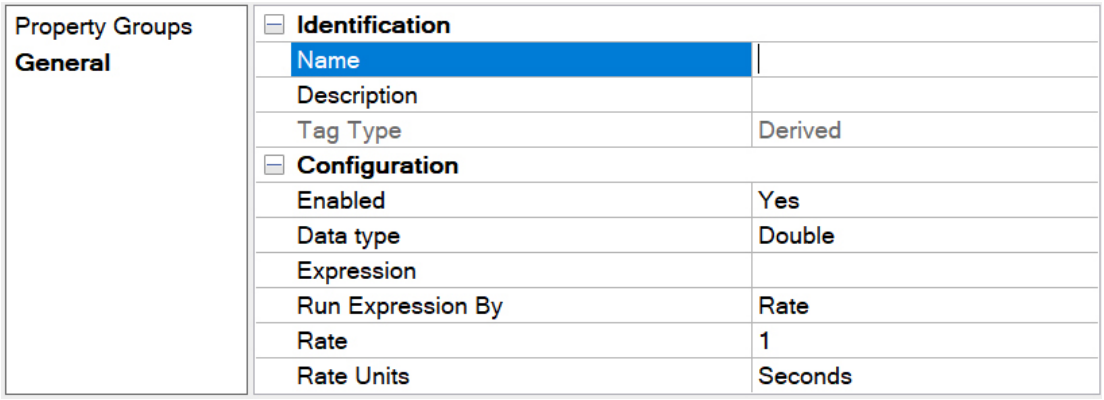
Note: The updates grouped in a complex update are not atomic, but are aggregated over a time interval. As a result, clients cannot assume that all updates in a complex update occurred at the same time. To get the time of the update, use the item's timestamp element. The timestamp format of tag elements within a Complex Tag is Windows FILETIME. *For more information, see this Microsoft article.*

Derived Tags

Derived Tags provide math and/or logic operations on the value or quality of one or more tags. 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 OPC DA quality of the tag is “Bad - Out of Service.”

Notes:

1. The execution of the string follows standard arithmetic rules.
   For more information, refer to Expression Strings.
2. Array tags are not supported.



Identification

Name: Specify (or edit) the name for the tag. The name can be up to 256 characters in length and cannot contain periods, contain double quotes, or start with an underscore.

Description: Specify information or an explanatory comment for the tag. The description is a string of up to 255 characters.

Tag Type: Verify the tag type (selected on creation of the tag; not editable).

Configuration

Enabled: Specify if this tag is currently enabled / active. Tags that are enabled have references to other server tags and do computational work. Tags that are disabled have no references to server tags.

Data Type: Specify 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 are truncated.

Expression: Specify the expression to be executed by the tag *(see below)*.

Tag Browser: Click to launch the browser (from which users may locate tags).

Run Expression By: Specify whether the expression is run by Rate or Trigger. The default setting is Rate. Descriptions are as follows:

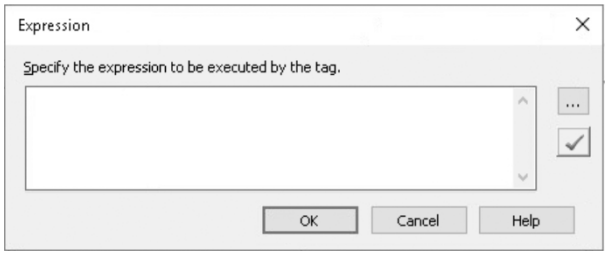
- **Rate**: Specify how often the expression is evaluated / run. The default setting is 1 second.
  - **Unit**: Specify the unit of time (default is seconds). The range for milliseconds is 10 to 4,233,600,000. The range for seconds is 1 to 4233,600. The range for minutes is 1 to 70560. The range for hours is 1 to 1176. The range for days is 1 to 49. The default unit is seconds.
- **Trigger**: Specify the condition at which the expression is evaluated / run. There are two types: **Trigger**and **Complete**. The tag defined in the Trigger field is monitored for a zero to non-zero transition. When this transition occurs, the expression is evaluated / run. No additional updates 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 writes a value of VARIANT TRUE 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.
  Note: If the tag has a Boolean data type, it indicates TRUE. For other data types, the VARIANT_TRUE value is converted, resulting in -1 for signed data types or “max. value for type” for unsigned data types.

Expression Strings

Expression strings are used to specify the math formula to be defined. The Browse button allows you to locate a tag. 
Expressions are validated for correct syntax and valid format. The Validate button allows you to verify the syntax of the expression while you're building it. 

Note:

- When referencing the value of a tag in the Expression string, the fully qualified item name is preceded by the keyword TAG and enclosed in parentheses. When referencing the quality of a tag in the Expression string, the fully qualified item name is preceded by the keyword QUALITY and enclosed in parentheses.
- Array tags cannot be used in an expression.



Example 1:

TAG(Channel1.Device1.Tag1)

Example 2:
QUALITY(Channel1.Device1.Tag1)

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

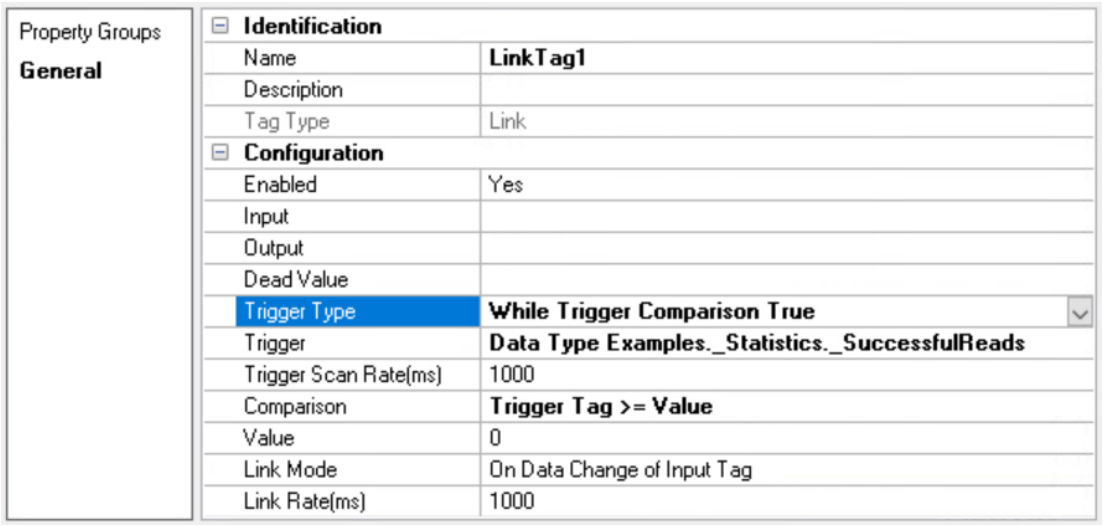
Example 3:
TAG(x) + TAG(y) # Add tag x and tag y

Example 4:
QUALITY(x) AND QUALITY(y) # Examine the quality of tag x and tag y

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

Link Tags

Link Tags are used to create a relationship between 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, effectively linking the two tags. The Link tag has the string value of *<input tag> != <output tag>* until the first link occurs, then the string becomes *<input tag> == <output tag>*. When disabled, a Link Tag does no linking and the OPC DA quality of the tag is "Bad - Out of Service."
See Also: Array Tags



Identification

Name: Specify (or edit) the name for the tag. The name can be up to 256 characters in length and cannot contain periods, contain double quotes, or start with an underscore.

Description: Specify information or an explanatory comment for the tag. The description is a string of up to 255 characters.

Tag Type: Verify the tag type (selected on creation of the tag; not editable).

Configuration

Enabled: Specify if this tag is currently enabled / active. Tags that are enabled have references to other server tags and do computational work. Tags that are disabled have no references to server tags.

Input: Specify (or locate) the tag from the server to be 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: Specify (or locate) the tag from the server to be the source of the link and its value is written to the Input Tag. The Output Tag must be writable.
Note: The Input and Output fields cannot be the same.

Dead Value: Specify 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 to ensure that no data is lost. If the field is left empty, the dead value is disabled. If disabled, the Output's value and quality does 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 Type: Specify when the configured link is active. The options are: Always, While Trigger Comparison True, and On Trigger Comparison True. The default selection is Always. Descriptions of the options are as follows:

- **Always**: The configured link is active as long as the server is running and the Link Tag is enabled.
- **While Trigger Comparison True**: The configured link is active when the server is running, the Link Tag is enabled, and the configured trigger comparison evaluates to true. While the logic defined for the trigger in the Comparison and Value fields is satisfied, linking occurs based on the configured Link Mode (i.e. On Data Change of Input Tag or On Interval). When the trigger comparison evaluates to false, no linking occurs.
- **On Trigger Comparison True**: If the server is running and the Link Tag is enabled, then the input is linked to the output when the logic defined for the trigger in the Comparison and Value fields is satisfied. At that moment, the Input Tag is read and the Output Tag is updated. No link occurs until a change in the trigger value occurs and satisfies the configured trigger logic.

Trigger Tag: Specify (or locate) a tag whose value is compared to the Value parameter using the Comparison parameter (such as Trigger Tag == Value). This parameter is only available to configure when the Trigger Type is While Trigger Comparison True or On Trigger Comparison True.
Notes:

- The specified tag must be a valid server tag and cannot be an array tag. The Trigger and Output Tags cannot be the same.
- **While Trigger Comparison True**: While the Trigger Tag value satisfies the Comparison and Value configuration, the input is linked to the output based on the Link Mode configuration. No link occurs when the trigger comparison is not met.
- **On Trigger Comparison True**: The Trigger Tag is scanned at the configured Trigger Scan Rate. Each time a change in the Trigger Tag value that satisfies the Comparison and Value configuration is detected, a read of the Input Tag occurs and the Output Tag is updated. No link occurs until a change in the trigger value occurs and satisfies the configured trigger logic.

Trigger Scan Rate: Specify how often to read the Trigger Tag. The valid range is 50 – 1800000000 milliseconds. The default setting is 1000ms.

Comparison: Specify the logical comparison between the Trigger Tag value and the Value parameter. Options are:

- **Trigger Tag > Value**: The trigger tag value is greater than the value.
- **Trigger Tag >= Value**: The trigger tag value is greater than or equal to the value.
- **Trigger Tag < Value**: The trigger tag value is less than the value.
- **Trigger Tag <= Value**: The trigger tag value is less than or equal to the value.
- **Trigger Tag == Value**: The trigger tag value is equal to the value.
- **Trigger Tag != Value**: The trigger tag value is not equal to the value.

Value: Specify the value to compare to the value of the Trigger Tag. The data type of this value is equal to the data type of the Trigger Tag.

Link Mode: Specify the required conditions to initiate the link. The default selection is On Data Change of Input Tag. Descriptions of the options are as follows:

- **On Data Change of Input Tag**: The input is linked to the output any time the input is good quality and its value changes.
  Tip: When the link mode is **On Trigger Comparison True**, a true trigger tag input change also causes a

link.

- **On Data Change of Input Tag (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 has the string value <input tag> != <output tag> until after the first update successfully completes. At that time, subsequent updates have the string value <input tag> == <output tag>.
  Note: An initial update of bad quality is not considered an initial update.
- **On Interval**: The input is linked to the output every Link Rate (milliseconds).

Link Rate (ms): Specify the frequency of update as defined in this parameter depends on the chosen Link Mode. The valid range is 50 – 1800000000 milliseconds. The default setting is 1000ms. Description of the Link Rate for each of the Link Mode options is as follows:

- **On Data Change of Input Tag**: The configured Link Rate defines the scan rate of the Input Tag. This is the fastest rate at which a change in the Input value may be detected.
- **On Interval**: The Link Rate defines the rate at which to update the output. Each time the interval elapses, the Input Tag is read and the Output Tag is updated.
  Note: A link for the On Interval Link Mode 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. Make sure that linking is occurring no faster than is required.

See Also: *Link Behavior Matrix*

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 without 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.

Link Behavior Matrix

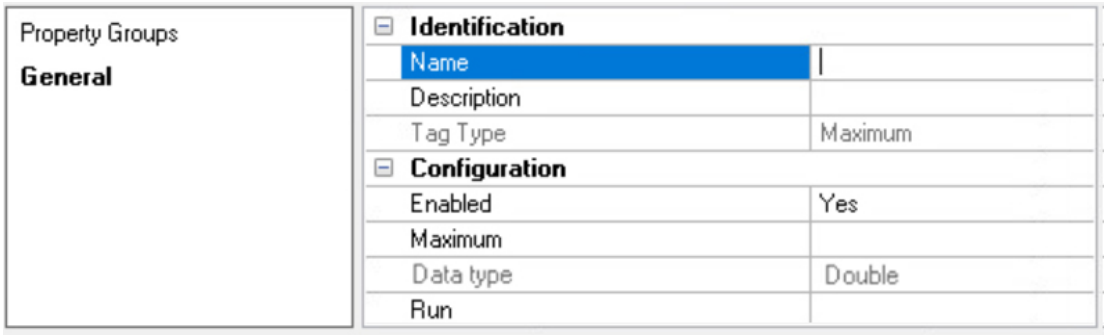
Trigger Type	Link Mode	Link Rate Behavior	Link Behavior
Always	On Data Change of Input Tag	Scan Rate of the Input Tag	Each detected change in the value of the input tag is written to the output tag.
Always	On Data Change of Input Tag (Ignore Initial Update)	Scan Rate of the Input Tag	After the input tag value has changed at least once, each detected change in the value of the input tag is written to the output tag.
Always	On Interval	Interval to Update the Output Tag	Each time the link rate elapses, a read of the input tag occurs. When that read completes, the value of the input tag is written to the output tag.

Trigger Type	Link Mode	Link Rate Behavior	Link Behavior
While Trigger Comparison True	On Data Change of Input Tag	Scan Rate of the Input Tag	If the configured trigger comparison evaluates to true, each detected change in the value of the input tag is written to the output tag. When the trigger comparison is false, no linking occurs.
While Trigger Comparison True	On Data Change of Input Tag (Ignore Initial Update)	Scan Rate of the Input Tag	If the configured trigger comparison evaluates to true and the input tag value has changed at least once; each detected change in the value of the input tag is written to the output tag. When the trigger comparison is false, no linking occurs.
While Trigger Comparison True	On Interval	Interval to Update the Output Tag	When the trigger comparison initially evaluates to true, a read of the input tag occurs. When that read completes, the value of the input tag is written to the output tag. While the trigger comparison is true, each time the link rate elapses, a read of the input tag occurs. When that read completes, the value of the input tag is written to the output tag. When the trigger comparison is false, no linking occurs.
On Trigger Comparison True	N/A	N/A	If the trigger tag value changes and the trigger comparison evaluates to true, a read of the input tag occurs. When that read completes, the value of the input tag is written to the output tag. Only a single write occurs each time the trigger is evaluated as true.

Maximum Tags

When enabled, Maximum Tags calculate the maximum of a tag's range of values over time. 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. Maximum Tags report OPC DA quality of “Bad - Out of Service” when disabled.

Note: Array tags are not supported.



Identification

Name: Specify (or edit) the name for the tag. The name can be up to 256 characters in length and cannot contain periods, contain double quotes, or start with an underscore.

Description: Specify information or an explanatory comment for the tag. The description is a string of up to 255 characters.

Tag Type: Verify the tag type (selected on creation of the tag; not editable).

Configuration

Enabled: Specify if this tag is currently enabled / active. Tags that are enabled have references to other server tags and do computational work. Tags that are disabled have no references to server tags.

Maximum: Specify (or locate) the tag from the server for which a maximum value is calculated.

Data Type: Verify the data type for the 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 takes ownership for coercing the data.

Run: Specify (or locate) the tag to monitor to determine if it is appropriate to calculate the maximum value of the tag. Whenever the Run Tag's value is non-zero, new maximum values are calculated for the Maximum Tag at the scan rate applied to the source tag. The calculation determines whether or not the currently scanned maximum value is higher than the reported maximum value. If it is, it replaces the reported value with the scanned value. When the Run Tag's value transitions back to zero, the last maximum value is maintained and reported to the client application. Upon return to non-zero, the last maintained maximum value is cleared.

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

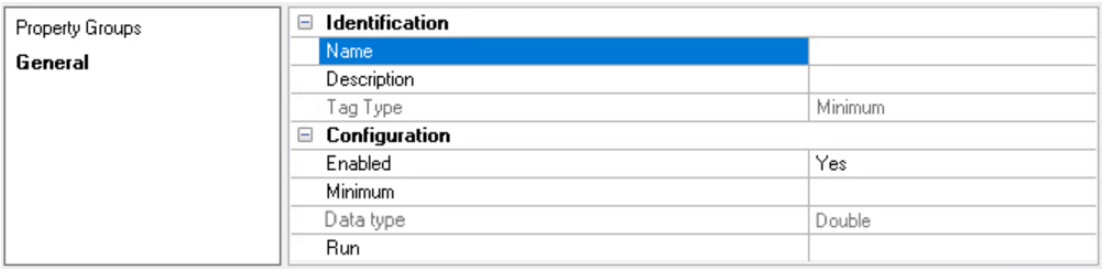
Notes:

1. If the quality of the Maximum Tag or the Trigger Tag is bad, the quality of the Maximum Tag is also bad.
2. If the quality of the Maximum Tag or the Trigger Tag is bad, calculation of the maximum value stops. For calculation of the maximum 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 of a tag's range of values over time. 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. Minimum Tags report OPC DA quality of “Bad - Out of Service” when disabled.

Note: Array tags are not supported.



Identification

Name: Specify (or edit) the name for the tag. The name can be up to 256 characters in length and cannot contain periods, contain double quotes, or start with an underscore.

Description: Specify information or an explanatory comment for the tag. The description is a string of up to 255 characters.

Tag Type: Verify the tag type (selected on creation of the tag; not editable).

Configuration

Enabled: Specify if this tag is currently enabled / active. Tags that are enabled have references to other server tags and do computational work. Tags that are disabled have no references to server tags.

Minimum: Specify (or locate) the tag from the server for which a minimum value is calculated.

Data Type: Verify the data type for the 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 takes ownership for coercing the data.

Run: Specify (or locate) the tag to monitor to determine if it is appropriate to calculate the minimum value of the tag. Whenever the Run Tag's value transitions from a zero to a non-zero value, a new minimum value is calculated for the Minimum Tag. The calculation determines whether or not the currently scanned minimum value is higher than the reported minimum value. If it is, it replaces the reported value with the scanned value.When the Run Tag's value transitions back to zero, the last minimum value is 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

Notes:

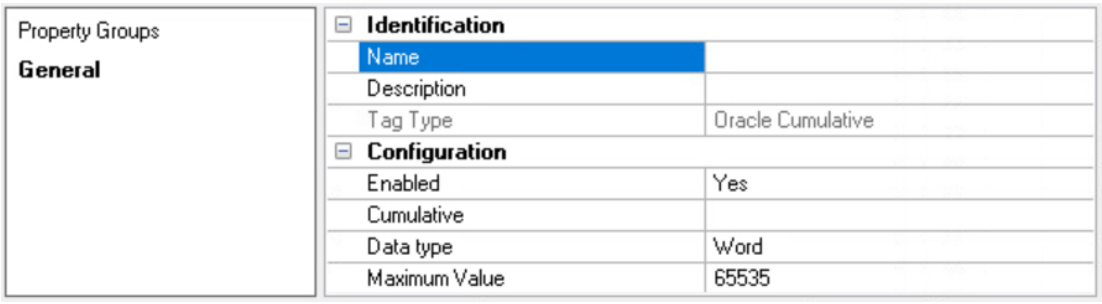
1. If the quality of the Minimum Tag or the Trigger Tag is bad, the quality of the Minimum Tag is also bad.
2. If the quality of the Minimum Tag or the Trigger Tag is bad, calculation of the minimum value stops. For calculation of the minimum to restart, both tags must return to good quality and the 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. This tag type reports an OPC DA quality of "Bad - out of service" when disabled.

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

Note: Array tags are not supported.



Identification

Name: Specify (or edit) the name for the tag. The name can be up to 256 characters in length and cannot contain periods, contain double quotes, or start with an underscore.

Description: Specify information or an explanatory comment for the tag. The description is a string of up to 255 characters.

Tag Type: Verify the tag type (selected on creation of the tag; not editable).

Configuration

Enabled: Specify if this tag is currently enabled / active. Tags that are enabled have references to other server tags and do computational work. Tags that are disabled have no references to server tags.

Cumulative Specify (or locate) the tag from the server for which a cumulative value is calculated for the tag.

Data Type Choose the data type for the tag.

Maximum Value Specify the highest number the monitored tag can reach. For example, if the tag is a counter with a maximum 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.

Arithmetic Function	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

Arithmetic Function	Formula	Result
Add Tag	*TAG(Sim.Device.Add1)+TAG (Sim.Device.Add2)*	If Add1=6 and Add2=71, the result is 77.
Add Quality	*QUALITY(Sim.Device.Add1)+QUALITY (Sim.Device.Add2)*	If both the Add1 and the Add2 tags have good quality, the result is 2. If neither tag has good quality, the result is 0. If only one of the tags has good quality, the result is 1.
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.

Function	Formula	Result
Absolute Value	*ABS(TAG(<Tagname>))*	This function returns the absolute value of the tag value or expression within the parenthesis.
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 from negative to positive infinity. The input range is any derivative of -90.0 to 90.0.

Examples

Function	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 are Boolean data type. A Boolean True can be a 1 or -1 depending on the client application; a Boolean False is 0.

Important: All Logic Tags are specific to Derived Tags.

Logic Function	Formula	Result
And	*TAG(<Tagname>)AND TAG (<Tagname>)*	This function returns True if each of the expressions evaluate to True. If any expression returns False, the function returns False.
Equal To	*TAG(<Tagname>)==TAG(<Tagname>)*	This function compares the results of each expression and returns True if they are the same, False if they are not the same.
Greater Than	*TAG(<Tagname>)>TAG(<Tagname>)*	This function compares the results of each expression and returns True if the first is greater than the other; otherwise it returns False.
Less Than	*TAG(<Tagname>)<TAG(<Tagname>)*	This function compares the results of each expression and returns True if the first is less than the other; otherwise it returns False.
Not	*NOT(TAG(<Tagname>))*	This function returns True if the expression evaluates to 0 and False if the expression result is greater than 0.
Or	*TAG(<Tagname>)OR TAG (<Tagname>)*	This function returns True if any of the expressions evaluate to greater than 0. If all expressions evaluate to 0, the function returns False.

Examples

Logic Function	Formula	Result
And Values	*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.
And Qualities	*QUALITY(Sim.Device.And1)AND QUALITY(Sim.Device.And2)*	If both the And1 and the And2 tags have good quality, the result is True. If either of the And1 or the And2 tags do not have good quality, the result is False.
Equal To	*TAG(Sim.Device.EqualTo1)-*	If EqualTo1=Test and EqualTo2=Test, the result is

Logic Function	Formula	Result
	==TAG(Sim.Device.EqualTo2)	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

Logic Function	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.

Optimizing Performance

The rate of change Advanced Tags can acquire is correlated to the number of advanced tags. As more advanced tags are added, the rate slows.

Below are some very general statistics on the performance using the most basic Link Tag type. This data on its performance is specific to the computing environment where the specific OPC server resided during testing. Performance varies depending on the computing environment and the type and number of advanced tags being processed.

Considerations

- Advanced tags are processed at a *maximum* rate of 100 milliseconds.
  Use caution configuring tags at 100 ms, especially with large projects with many tags configured. Verify the operation of the configuration prior to running the project in production.
- Tag groups are processed sequentially. Grouping should be avoided to achieve best possible performance.
- The Advanced Tags Plug-In implements one process thread per Advanced Tag type; multiple threads per type is not possible.
- Using multiple cores for the server is not likely to increase performance, though it may free processing time for other applications.
- In the test scenario in the table below, when using the Link Tags type with Link Rate (milliseconds) at least five times the number of tags, updates are provided in real time.

Link Tag Testing (10-Minute Time Frame)

Number of Tags	Link Rate (ms)	Expected Updates(1)	Actual Updates
1000	5000	120	119

Number of Tags	Link Rate (ms)	Expected Updates(1)	Actual Updates
1000	**2500**	**240**	**233**
1000	**1000**(2)	**600**	**534**
500	2500	240	238
200	1000(2)	600	597
100	500	1200	1200
100	**400**	**1500**	**1478**
100	**300**	**2000**	**1918**

Values in **bold** are the test scenarios well below the number of expected updates.

Notes:

1. Expected updates: the number of updates that occur if the Link Rate is honored with 100% accuracy.
2. As a comparison; at the same Link Rate setting of 1,000 milliseconds the number of updates is less than expected for 1,000 Link Tags; whereas with 200 Link Tags, the number of updates occur as expected.

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 are assigned appropriate default values.

Notes:

- For information on specifying which character to use as the variable (comma or semicolon), refer to "Options - General" in the server help file.
- CSV files generated with an earlier version of the Advanced Tags Plug-In may contain property names and values that differ from the current version format. Although these files import without problems, it is recommended that editing for CSV import occurs using the latest CSV export format.

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 to 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.

Example CSV

Exporting an Advanced Tag Group list generates 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 is not 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 an average is calculated
Data Type	The Average Tag's data type
Run Tag	The tag to 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 is 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)

Column Name	Values
Enabled	1 to enable, and 0 to disable
Data Type	The Derived Tag's data type
Expression	The expression to 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
Link Mode	Options include On Data Change of Input Tag, On Data Change of Input Tag (Ignore Initial Update), and On Interval
Link Rate	The link rate is in milliseconds
Trigger Tag	The tag value compared to the Value parameter using the Comparison parameter
Comparison	Options include the following: Trigger Tag == Value Trigger Tag != Value Trigger Tag > Value Trigger Tag >= Value Trigger Tag < Value Trigger Tag <= Value
Value	The Value compared to the Trigger Tag
Trigger Link Mode	Options include Always, While Trigger Comparison True, and On Trigger Comparison True
Trigger Scan Rate	The trigger scan rate in milliseconds

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 is calculated
Data Type	The Maximum Tag's data type
Run Tag	The tag to 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"

Column Name	Values
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 is calculated
Data Type	The Minimum Tag's data type
Run Tag	The tag to 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**.

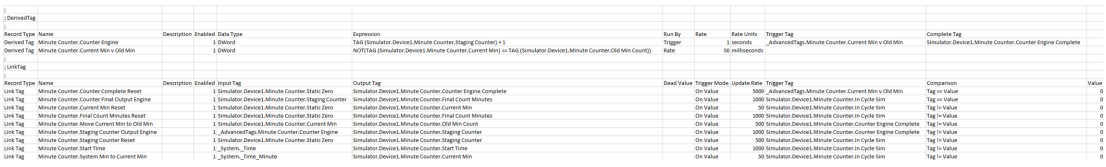
Note: This option is only available when an Advanced Tag Group is selected.

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.

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



//													
// DerivedTag													
//													
Record Type	Name	Description	Enabled	Data Type	Expression	Run By	Rate	Rate Units	Trigger Tag	Complete Tag			
Derived Tag	Minute Counter.Counter Engine		1	DWord	TAG (Simulator.Device1.Minute Counter.Staging Counter) + 1	Trigger	1	seconds	_AdvancedTags.Minute Counter.Current Min v Old Min	Simulator.Device1.Minute Counter.Counter Engine Complete			
Derived Tag	Minute Counter.Current Min v Old Min		1	DWord	NOT(TAG (Simulator.Device1.Minute Counter.Current Min) == TAG (Simulator.Device1.Minute Counter.Old Min Count))	Rate	50	milliseconds					
//													
// LinkTag													
//													
Record Type	Name	Description	Enabled	Input Tag	Output Tag	Dead Value	Trigger Mode	Update Rate	Trigger Tag	Comparison	Value		
Link Tag	Minute Counter.Counter Complete Reset		1	Simulator.Device1.Minute Counter.Static Zero	Simulator.Device1.Minute Counter.Counter Engine Complete		On Value	5000	_AdvancedTags.Minute Counter.Current Min v Old Min	Tag == Value	0		
Link Tag	Minute Counter.Counter Final Output Engine		1	Simulator.Device1.Minute Counter.Staging Counter	Simulator.Device1.Minute Counter.Final Count Minutes		On Value	1000	Simulator.Device1.Minute Counter.In Cycle Sim	Tag == Value	0		
Link Tag	Minute Counter.Current Min Reset		1	Simulator.Device1.Minute Counter.Static Zero	Simulator.Device1.Minute Counter.Current Min		On Value	50	Simulator.Device1.Minute Counter.In Cycle Sim	Tag != Value	0		
Link Tag	Minute Counter.Final Count Minutes Reset		1	Simulator.Device1.Minute Counter.Static Zero	Simulator.Device1.Minute Counter.Final Count Minutes		On Value	1000	Simulator.Device1.Minute Counter.In Cycle Sim	Tag != Value	0		
Link Tag	Minute Counter.Move Current Min to Old Min		1	Simulator.Device1.Minute Counter.Current Min	Simulator.Device1.Minute Counter.Old Min Count		On Value	500	Simulator.Device1.Minute Counter.Counter Engine Complete	Tag != Value	0		
Link Tag	Minute Counter.Staging Counter Output Engine		1	_AdvancedTags.Minute Counter.Counter Engine	Simulator.Device1.Minute Counter.Staging Counter		On Value	1000	Simulator.Device1.Minute Counter.Counter Engine Complete	Tag != Value	0		
Link Tag	Minute Counter.Staging Counter Reset		1	Simulator.Device1.Minute Counter.Static Zero	Simulator.Device1.Minute Counter.Staging Counter		On Value	500	Simulator.Device1.Minute Counter.In Cycle Sim	Tag != Value	0		
Link Tag	Minute Counter.Start Time		1	_System._Time	Simulator.Device1.Minute Counter.Start Time		On Value	1000	Simulator.Device1.Minute Counter.In Cycle Sim	Tag != Value	0		
Link Tag	Minute Counter.System Min to Current Min		1	_System._Time_Minute	Simulator.Device1.Minute Counter.Current Min		On Value	50	Simulator.Device1.Minute Counter.In Cycle Sim	Tag != Value	0		

Error Message Descriptions

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

Advanced Tag Messages
CSV Messages
Derived Expression Messages

Advanced Tag Messages

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

<Advanced tag name> is invalid. Verify the tag references valid server tags and does not contain any references to itself or to an unsupported or array data type.
<Tag name> has an invalid expression: <expression error>.
Failed to write to tag <tag name> from link tag <link tag name>.
References to array tags with an unsupported or array data type are not allowed <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 <advanced tag name>.
Unable to start complete tag <tag name> on tag <advanced tag name>.
Unable to start element tag <tag name> on tag <advanced tag name>.
Unable to start required tag reference <reference> on tag <tag name>.
Unable to start required tag reference <reference> in expression <expression>.
Unable to start trigger tag <tag name> for element <element> on tag <advanced tag name>.
Unable to start trigger tag <tag name> on tag <advanced tag name>.

<Advanced tag name> is invalid. Verify the tag references valid server tags and does not contain any references to itself or to an unsupported or array data type.

Error Type:

Error

Possible Cause:

The tag does not exist in the project, is already referenced, or is an unsupported data type.

Solutions:

1. Add the tag to the project.
2. Reference the correct tag.
3. Choose a tag with a supported data type.

<Tag name> has an invalid expression: <expression error>.

Error Type:

Warning

Possible Cause:

The expression may contain an unknown tag, misspelled keyword, or syntax error.

Solution:

1. Verify any tags in the expression exist and are valid. Correct as necessary.
2. Check the expression for misspelled keywords and syntax errors. Correct as necessary.

Failed to write to tag <tag name> from link tag <link tag name>.

Error Type:
Warning

Possible Cause:
1. The write tag does not have a data type (set to Default) or no conversion between data types is valid.
2. The write tag is not accessible because of limited privileges or user permissions.
3. The tag is not writable.
4. Server is not in a licensed state.

Solution:
1. The write tag should have proper data type.
2. Fix user specific privileges for the write tag.
3. Change the write tag to be writable.
4. License the server and / or drivers.

References to array tags with an unsupported or array data type are not allowed <tag name>.

Error Type:
Warning

Possible Cause:
An attempt was made to reference a tag with an unsupported or array data type from an advanced tag.

Solution:
Choose a tag is that is not an array and has a supported data type for the advanced tag type.

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 acquire licensing for Advanced Tags - Type <advanced tag type>.

Error Type:
Error

Possible Cause:
The configured advanced tag type may not have the required license.

Solution:
Verify and include the required license for the Advanced Tag Type.

Unable to start complete tag <tag name> for element <element> on tag <advanced 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 <advanced 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 <advanced 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 required tag reference <reference> in expression <expression>.

Error Type:
Warning

Possible Cause:
The tag reference in the expression may not exist, has an invalid address, or has an unsupported data type.

Solution:
Verify the tag reference in the expression exists, has a valid address, and has a supported data type. Correct as necessary.

Unable to start trigger tag <tag name> for element <element> on tag <advanced 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 <advanced 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 Messages
The following messages may be generated. Click on the link for a description of the message.

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 Trigger 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>. Link mode <link mode> is invalid. Link mode set to On Data Change of Input Tag.
Error importing CSV tag <tag name>. Trigger type <trigger type> is invalid. Trigger type set to Always.
Error importing CSV tag <tag name>. Trigger scan rate <value> <units> is out of range. Trigger scan rate has been set to <value> <units>.
Error importing CSV tag <tag name>. Link rate <value> <units> is out of range. Link rate has been set to <value> <units>.
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 Trigger Tag == Value.

Error Type:
Warning

Possible Cause:
The comparison is invalid.

Solutions:

1. In the user interface, change the comparison to one that is valid.
2. Export the tag to a CSV file and change the comparison to one that is valid, then re-import the CSV file.

Note:
Valid comparisons are as follows: Trigger Tag == value, Trigger Tag != value, Trigger Tag > value, Trigger Tag >= value, Trigger Tag < value, and Trigger 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 <trigger mode> is invalid. Trigger mode set to On Data Change of Input Tag.

Error Type:
Warning

Possible Cause:
The trigger mode is invalid.

Solutions:

1. In the user interface, change the tag's trigger mode to a valid mode.
2. Export the tag to a CSV file and change the trigger mode to a valid mode, then re-import the CSV file.

Note:
The valid modes are On Data Change of Input Tag, On Data Change of Input Tag (Ignore Initial Update), and On Interval.

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

Error Type:
Warning

Possible Cause:
The tag's trigger scan rate is out of range.

Solutions:

1. In the user interface, change the tag's trigger scan rate to a value within the valid range.
2. Export the tag to a CSV file and change the trigger scan rate to a valid value, then re-import the CSV file.

Note:
The valid range is 50 milliseconds to 1800000 milliseconds.

Error importing CSV tag <tag name>. Link mode <link mode> is invalid. Link mode set to On Data Change of Input Tag.

Error Type:
Warning

Possible Cause:
The link mode is invalid.

Solutions:

1. In the user interface, change the tag's link mode to a valid mode.
2. Export the tag to a CSV file and change the link mode to a valid mode, then re-import the CSV file.

Note:
The valid modes are On Data Change of Input Tag, On Data Change of Input Tag (Ignore Initial Update), and On Interval.

Error importing CSV tag <tag name>. Trigger type <trigger type> is invalid. Trigger type set to Always.

Error Type:
Warning

Possible Cause:
The trigger type is invalid.

Solutions:

1. In the user interface, change the tag's trigger type to a valid type .
2. Export the tag to a CSV file and change the trigger type to a valid type , then re-import the CSV file.

Note:
The valid type are Always, While Trigger Comparison True, and On Trigger Comparison True.

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>. Trigger Mode <value> <units> is out of range. Link rate has been set to <value> <units>.

Error Type:
Warning

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

Solutions:
1. In the user interface, change the tag's link rate to a value within the valid range.
2. Export the tag to a CSV file and change the link rate to a valid value, then re-import the CSV file.

Note:
The valid range is 50 milliseconds to 1800000 milliseconds.

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 Messages

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

", 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.

", 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 beginning.
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 is saved.

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

A

Advanced Tag Browser 7
Advanced Tag Error Messages 28
Advanced Tag Groups 6
Advanced tag is invalid. Verify the tag references valid server tags and does not contain any references to itself or to an unsupported or array data type. 28
Arithmetic Operators 20
Average Tags 8

C

common property 6
Complex Tags 9
Configuration 6
Configuration Group 6
CSV Error Messages 32
CSV Import/Export 24

D

Derived Expression Error Messages 37
Derived Tags 11

E

Error Descriptions 28
Error importing CSV complex element record <record index>. Insert by <value> is invalid. Insert by set to 'Rate'. 33
Error importing CSV complex element record <record index>. Update rate <value><units> is out of range. Update rate has been set to <value><units> 33
Error importing CSV tag <tag name>. Comparison <comparison> is invalid. Comparison set to Trigger Tag == Value. 33
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>. 33
Error importing CSV tag <tag name>. Link mode <link mode> is invalid. Link mode set to On Data Change of Input Tag. 34-35
Error importing CSV tag <tag name>. Link rate <value> <units> is out of range. Link rate has been set to <value> <units>. 36
Error importing CSV tag <tag name>. Maximum value <max value> is out of range. Maximum value has been set to <max value>. 34
Error importing CSV tag <tag name>. Run by <value> is invalid. Run by set to 'Rate'. 34

Error importing CSV tag <tag name>. Run by rate <value><units> is out of range. Run by rate has been set to <value><units>. 34
Error importing CSV tag <tag name>. Trigger link mode <link mode> is invalid. Trigger link mode set to Always. 35
Error importing CSV tag <tag name>. Trigger scan rate <value> <units> is out of range. Trigger scan rate has been set to <value> <units>. 35
Error importing CSV tag <tag name>. Update by <update rate> is invalid. Update by set to <update rate>. 35
Error importing CSV tag <tag name>. Update rate <value><units> is out of range. Update rate has been set to <value><units>. 36
Error importing CSV tag data. No advanced tag records found in CSV file. 36
Expected (. 37
Expected) . 37
Expected comma. 37
Expression modified. 38

F

Failed to write to tag <tag name> from link tag <link tag name>. 29
FILETIME 11
Functions 21

H

Help Contents 4

I

Identification 6

L

Link Tags 13
Logic Tags 22

M

Maximum Tags 16
Minimum Tags 17

N

Numeric expression expected. 38

O

Optimizing Performance 23
Oracle Cumulative Tags 19
Oracle MOC 10
Overview 4

P

Please enter an expression. 38
Plug-In Interface 5

R

Rate 9
References to array tags with an unsupported or array data type are not allowed <tag name>. 29

S

Schema 10
String tag expected. 38
Syntax error: <syntax string>. 38

T

Tag has an invalid expression <expression error>. 28
The complex tag already references <tag name> and will not be re-added. 29
The input and output tags cannot be the same. 29
The tag must be writeable. 30
The trigger and output tags cannot be the same. 30
Timestamp 11
Timestamp format 11
Trigger 9

U

Unable to acquire licensing for Advanced Tags - Type <advanced tag type>. 30
Unable to start complete tag <tag name> for element <element> on tag <advanced tag name>. 30
Unable to start complete tag <tag name> on tag <advanced tag name>. 30
Unable to start element tag <tag name> on tag <advanced tag name>. 31
Unable to start required tag reference <reference> in expression <expression>. 31
Unable to start required tag reference <reference> on tag <tag name>. 31

Unable to start trigger tag <tag name> for element <element> on tag <advanced tag name>. 31
Unable to start trigger tag <tag name> on tag <advanced tag name>. 32
Unknown tag or misspelled keyword: <tag name>. 39
Unrepresentable numeric constant: <tag name>. 39
Unterminated string. 39

X

XML string 10