

Advanced Tags Help

© 2009 Kepware Technologies

Table of Contents

1	Getting Started.....	3
	Help Contents.....	3
	Overview	3
2	Advanced Tag Creation.....	3
	Advanced Tags Creation.....	3
	Tag Browser	4
3	Advanced Tag Types.....	5
	Advanced Tag Types.....	5
	Average Value Tags.....	5
	Complex Tags.....	7
	Maximum Value Tags.....	10
	Minimum Value Tags.....	13
	Custom Advanced Tags.....	15
	Derived Value Tags.....	17
	Expression Strings.....	20
	Arithmetic Operators.....	20
	Functions.....	21
	Logic Tags.....	22
4	Error Descriptions.....	23
	Error Descriptions.....	23
	Advanced Tag Error Messages.....	24
	Advanced Tag Error Messages.....	24
	The advanced tag configuration cannot be modified since there is at least one client item referencing this tag	25
	The complex tag already references '%s' and will not be re-added.....	25
	Complex tags cannot contain references to other complex tags.....	25
	Tag '%s' does not exist or you are trying to add a dynamic or advanced tag reference which is not allowed... ..	25
	References to advanced tags are not supported (%s).....	25
	References to array tags are not supported ('%s').....	26
	References to complex tags are not supported ('%s').....	26
	Unable to start tag '%s' [Expression: '%s'].....	26
	Unable to start trigger tag '%s' on tag '%s'.....	26
	Unable to start complete tag '%s' on tag '%s'.....	27
	Unable to start element tag '%s' on tag '%s'.....	27
	Unable to start trigger tag '%s' for element '%s' on tag '%s'.....	27
	Unable to start complete tag '%s' for element '%s' on tag '%s'.....	27
	Unable to start required tag reference '%s' on tag '%s'.....	27
	Attempt to add Oracle Connectivity Suite client item '%s' failed.....	28
	Unable to start tag '%s' since it contains invalid tag references.....	28
	Derived Expression Error Messages.....	28
	Derived Expression Error Messages.....	28
	Expression invalid: '%s'.....	28
	Expression modified	29
	Unrepresentable numeric constant: '%s'.....	29
	Unterminated string.....	29
	Unknown tag or misspelled keyword: '%s'.....	29
	Syntax error: '%s'.....	30
	", expected.....	30

(expected.....	30
) expected.....	30
Numeric expression expected.....	30
Please enter an expression.....	31
String tag expected.....	31

Index

0

Advanced Tags Help

Help version 1.006

CONTENTS

[Overview](#)

What are Advanced Tags?

[Advanced Tag Creation](#)

How do I create Advanced Tags?

[Advanced Tag Types](#)

What are the different kinds of Advanced Tags?

[Error Descriptions](#)

What error messages do Advanced Tags produce?

Overview

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

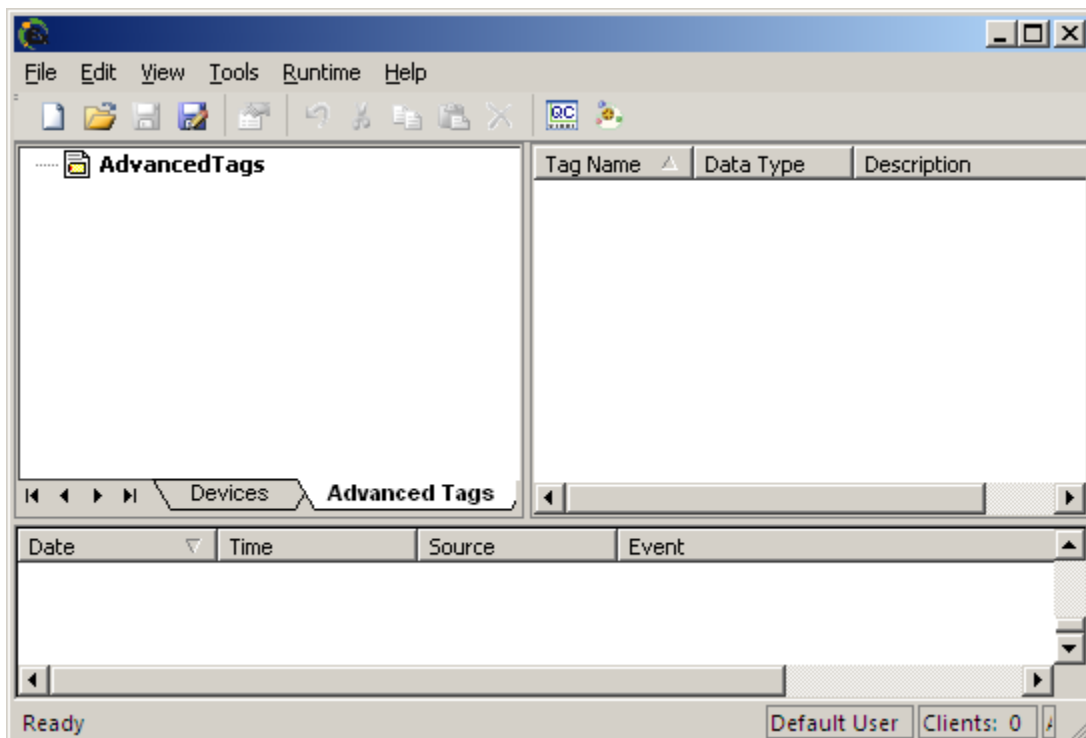
There are two methods of syntax for using one of these tags in a client application.

1. <_AdvancedTags>.<TagName>
2. <_AdvancedTags>.<TagGroup>.<TagName>.

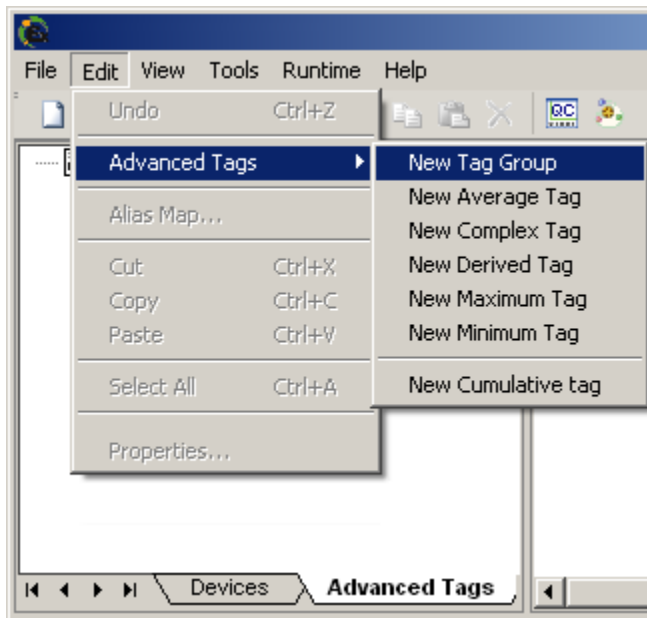
Advanced Tag Creation

Follow the instructions below for information on how to create an Advanced Tag.

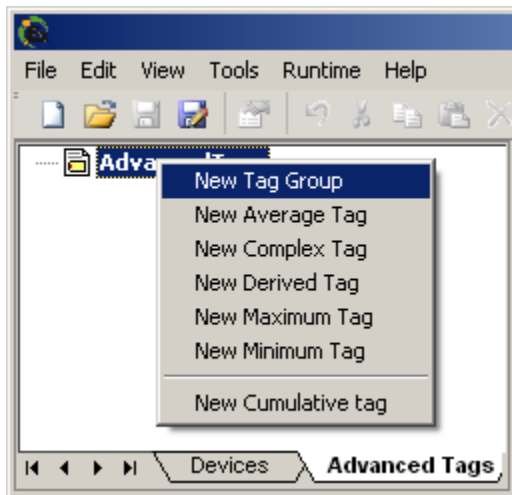
1. In the **Project** tree pane, select the **Advanced Tag Group** tab.
2. Click **Advanced Tags**.



3. Click **Edit | Advanced** and then select **<type of tag>**.



Note: Alternatively, right-click in the **Tag** pane and then select the type of tag from the **Context Menu**.

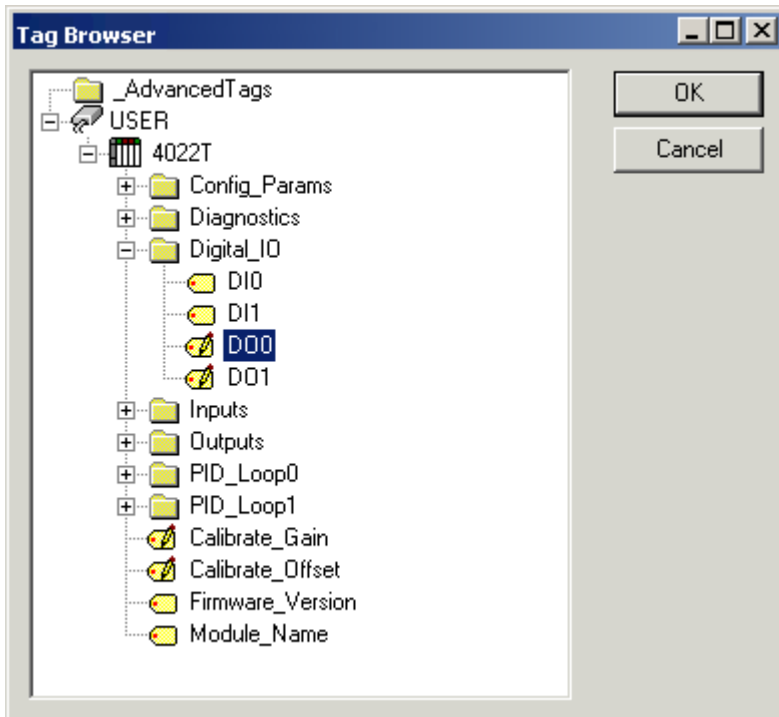


4. Next, edit and set the tag parameters as desired.

Note: Average, Derived, Maximum and Minimum Value Tags must specify the data type so that the result of the operation can be reported in an appropriate format. For example, if the result of an operation is a Floating point number and the data type is set to Word, then the decimal portion of the result will be stripped. Only the whole number portion of the result will be reported.

Tag Browser

The Tag Browser opens the server tag database, thus allowing tags that will be used as Advanced Tags to be selected directly from the server project.



Advanced Tag Types

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

[Average Value Tags](#)

[Complex Tags](#)

[Maximum Value Tags](#)

[Minimum Value Tags](#)

[Custom Advanced Tags](#)

[Derived Value Tags](#)

Average Value Tags

Average Value 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 MSEC.

Note: Array, Complex and Dynamic tags are not supported.

Example

A tag that measures the temperature at a location (with a scan rate of 60000 MSEC) 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 1: If the quality of the Average Tag or the Trigger Tag is bad, the quality of the Average Value 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.

Identification

Field	Description
Name	The Tag Name parameter can be used to enter the string that will represent the data available from the tag. The tag name can be up to 256 characters in length. While using long descriptive names is generally a good idea, keep in

	mind that some OPC client applications may have a limited display window when browsing the tag space of an OPC server. Note: Names can not contain periods, double quotes or start with an underscore.
Description	The Description parameter can be used to attach a comment to the tag. A string of up to 64 characters can be entered for the description.

Data Properties

Field	Description
Average	The tag from the server for which a running average will be calculated.
Data Type	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	The Run Tag 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.

Complex Tags

Complex Tags group multiple tags of varying data types as a single complex item or structure. They are currently supported for the Oracle MOC connection only.

Complex Tag Properties

General

Identification

Name:

Description:

Data properties

Elements:

Tag	Inserted By

Send complex tag updates by:

Rate: seconds

Trigger: ...

Complete: ...

Note: Only Oracle Connectivity Suite - Manufacturing_OperationsCenter Connections can utilize complex data at this time.

OK Cancel Apply Help

Identification

Identification

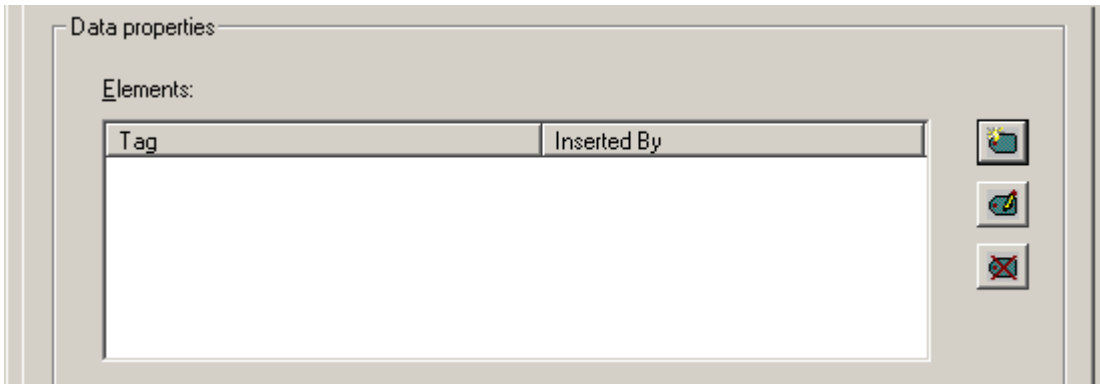
Name:




Description:

Field	Description
Name	The Tag Name parameter can be used to enter the string that will represent the data available from the tag. The tag name can be up to 256 characters in length. While using long descriptive names is generally a good idea, keep in mind that some OPC client applications may have a limited

	display window when browsing the tag space of an OPC server. Note: Names can not contain periods, double quotes or start with an underscore.
Description	The Description parameter can be used to attach a comment to the tag. A string of up to 64 characters can be entered for the description.

Data Properties



Field	Description
Elements	Element lists tags that will be part of the Complex tag. Note: Dynamic, Array and Complex tags can not be added to the Element List.
Browse Tags 	Browse Tags launch the Complex Element Property, which is then used to select tags for the Element list.
Modify Tags 	Modify Tags launch the Complex Element Property, which is then used to Modify elements that are highlighted in the Element list.
Delete Tags 	Delete Tags delete elements that are highlighted in the Element list.

Send Update Parameters



Field	Description
Rate	When selected, this parameter dictates the rate at which the Complex Tag updates will be published.
Trigger	When selected, this parameter dictates the condition at which Complex Tag updates will be published. Trigger: 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

Complete: When a Complex Tag update has been sent to all subscribing clients, the server will Write a value of 1 to the Complete Tag in order to notify that the update has been completed. 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: Dynamic, Complex and Array tags can not be used as Trigger or Complete tags. Complete tags must have writable access.

Complex Element

Field	Description
Element	The element is the tag that should be included as part the Complex Tag. Note: Dynamic, Complex and Array tags are not supported.
Rate	When selected, this parameter dictates the rate at which the Element will insert its value into the Complex Tag.
Trigger	When selected, this parameter dictates the condition at which Complex Tag updates will be published. Trigger: 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. Complete: When a Complex Tag update has been sent to all subscribing clients, the server will Write a value of 1 to the Complete Tag to notify that the update has been completed. 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: Dynamic, Complex and Array Tags can not be used as Trigger or Complete Tags. Complete Tags must have writable access.

Maximum Value Tags

Maximum Value 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 MSEC.

Note: Array, Complex and Dynamic tags are not supported.

Example

A tag that measures the temperature at a location (with a scan rate of 60000 MSEC) 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 Value 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.

Maximum Tag Properties

General

Identification

Name:

Description:

Data properties

Maximum: ...

Data type: Double ▾

Run: ...

OK Cancel Apply Help

Identification

Identification

Name:

Description:

Field	Description
Name	The Tag Name parameter can be used to enter the string that will represent the data available from the tag. The tag name can be up to 256 characters in length. While using long descriptive names is generally a good idea, keep in mind that some OPC client applications may have a limited display window when browsing the tag space of an OPC server. Note: Names can not contain periods, double quotes or start with an underscore.
Description	The Description parameter can be used to attach a comment to the tag. A string of up to 64 characters can be entered for the description.

Data Properties

Field	Description
Maximum	The tag from the server for which a maximum value is calculated.
Data Type	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 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.

Minimum Value Tags

Minimum Value 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 MSEC.

Note: Array, Complex and Dynamic tags are not supported.

Example

A tag that measures the temperature at a location (with a scan rate of 60000 MSEC) 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 value tag will also be bad.

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

The screenshot shows the 'Minimum Tag Properties' dialog box with the 'General' tab selected. The dialog is divided into two main sections: 'Identification' and 'Data properties'. The 'Identification' section contains two text input fields: 'Name' and 'Description'. The 'Data properties' section contains three fields: 'Minimum' (with a browse button), 'Data type' (a dropdown menu currently set to 'Double'), and 'Run' (with a browse button). At the bottom of the dialog are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

Identification

This close-up view shows the 'Identification' section of the dialog. It features two text input fields: 'Name' and 'Description', each with a small icon to its left. The fields are empty.

Field	Description
Name	<p>The Tag Name parameter can be used to enter the string that will represent the data available from the tag. The tag name can be up to 256 characters in length. While using long descriptive names is generally a good idea, keep in mind that some OPC client applications may have a limited display window when browsing the tag space of an OPC server.</p> <p>Note: Names can not contain periods, double quotes or start</p>

	with an underscore.
Description	The Description parameter can be used to attach a comment to the tag. A string of up to 64 characters can be entered for the description.

Data Properties

Field	Description
Minimum	The tag from the server for which a minimum value will be calculated.
Data Type	The data type of the Minimum Value 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 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.

Custom Advanced Tags

Custom Advanced Tags are created for specific needs by partner companies and/or customers.

Oracle Cumulative Tags

Oracle Cumulative Tags are specifically designed for Oracle requirements. They work by returning the delta between the last value polled and the current value polled from a specific tag.

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 can not 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.

Example

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)

Oracle Cumulative Tag Properties

Identification

Field	Description
Name	The Tag Name parameter can be used to enter the string that will represent the data available from the tag. The tag name can be up to 256 characters in length. While using long descriptive names is generally a good idea, keep in mind that some OPC client applications may have a limited display window when browsing the tag space of an OPC server. Note: Names can not contain periods, double quotes or start with an underscore.
Description	The Description parameter can be used to attach a comment to the tag. A string of up to 64 characters can be entered for

the description.

Data Properties

The screenshot shows a 'Data Properties' dialog box with the following fields:

- Cumulative:** A text input field that is currently empty, followed by a button with three dots.
- Data type:** A dropdown menu currently set to 'Word'.
- Maximum Value:** A text input field containing the number '500'.

Field	Description
Cumulative	The tag from the server for which the cumulative value will be calculated.
Data Type	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.

Derived Value Tags

Derived Value 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. The execution of the string follows standard arithmetic rules. For examples, please refer to the individual examples of each function.

Note: Array, Average, Complex, Custom Advanced, Dynamic, Maximum and Minimum tags are not supported.

Derived Tag Properties [X]

General

Identification

Name:

Description:

Data properties

Data type: Double

Expression:

Run expression by:

Rate: seconds

Trigger:

Complete:

OK Cancel Apply Help

Identification

Identification

Name:

Description:

Field	Description
Tag Name	<p>Tag Name is used to enter the string that will represent the tag's available data. It can be up to 256 characters in length. Although it is usually good to use descriptive names, some OPC client applications may have a limited display window when browsing the tag space of an OPC server.</p> <p>Note: Names can not contain periods, double quotes or start with an</p>

	underscore.
Description	Description is used to attach a comment to the tag. A string of up to 64 characters can be entered.

Data Properties

The screenshot shows a dialog box titled 'Data properties'. It contains a 'Data type' dropdown menu with 'Double' selected. Below it is an 'Expression' text area with a vertical scrollbar. To the right of the text area are two buttons: one with three dots and one with a green checkmark.

Field	Description
Data Type	The data type of the result of the expression execution.
Expression	The expression that will be executed by the tag.

Run Trigger

The screenshot shows a dialog box titled 'Run expression by:'. It has three radio button options: 'Rate', 'Trigger', and 'Complete'. The 'Rate' option is selected. The 'Rate' field contains the value '1' and a dropdown menu set to 'seconds'. The 'Trigger' and 'Complete' fields are empty. Each field has a help button (three dots) to its right.

Field	Description
Rate	When selected, this parameter dictates the rate at which the expression is evaluated/run.
Trigger	When selected, this parameter dictates the condition at which Complex Tag updates will be published. Trigger: The Trigger Tag's value will 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. Complete: When a Complex Tag update has been sent to all subscribing clients, the server will write a value of 1 to the Complete Tag to notify that the update has been completed. Users can tie this to logic in a controller or other subsystem to indicate that the next complex value is ready to be formulated.

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, Average, Complex, Custom Advanced, Dynamic, Minimum and Maximum 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 parenthesis.

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](#).

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	$TAG(<Tagname>)+TAG(<Tagname>)$	The result of this operation returns the sum of the tag values in the expression.
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.

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	<i>TAG(<Tagname>)<TAG(<Tagname>)</i>	This function tests to see if one value is less than the other.
Not	<i>NOT(TAG(<Tagname>))</i>	This function test to see if the Tag value or Expression is not True.
Or	<i>TAG(<Tagname>)OR TAG(<Tagname>)</i>	This function tests to see if any of the values in the expression are True.

Examples

Tag Name	Formula	Result
And	<i>TAG(Sim.Device.And1)AND TAG(Sim.Device.And2)</i>	If And1=0 and And2=97, the result is False. If And1=1 and And2=1, the result is True.
Equal To	<i>TAG(Sim.Device.EqualTo1)==TAG(Sim.Device.EqualTo2)</i>	If EqualTo1=Test and EqualTo2=Test, the result is True.
Greater Than	<i>TAG(Sim.Device.GreaterThan1)>TAG(Sim.Device.GreaterThan2)</i>	If GreaterThan1=16 and GreaterThan2=17, the result is False.
Less Than	<i>TAG(Sim.Device.LessThan1)<TAG(Sim.Device.LessThan2)</i>	If LessThan1=20 and LessThan2=41 the result is True.
Not	<i>NOT(TAG(Sim.Device.EqualTo1)==TAG(Sim.Device.EqualTo2))</i>	If EqualTo1=Test and EqualTo2=Test, the result is False.
Or	<i>TAG(Sim.Device.Or1)OR TAG(Sim.Device.Or2)</i>	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	<i>TAG(Sim.Device.GreaterThanOrLessThan1)>=TAG(Sim.Device.GreaterThanOrLessThan2)</i>	If GreaterThanOrLessThan1=17 and GreaterThanOrLessThan2=17, the result is True. If GreaterThanOrLessThan1=16, the result is False.
IsTrue	<i>TAG(Sim.Device.IsTrue1)==True</i>	If IsTrue=0 then the result is Result is False.
IsFalse	<i>TAG(Sim.Device.IsFalse1)==FALSE</i>	If IsFalse=1 then the result is False.
Less Than or Equal To	<i>TAG(Sim.Device.LessThanOrGreaterThan1)<=TAG(Sim.Device.LessThanOrGreaterThan2)</i>	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

[The advanced tag configuration cannot be modified since there is at least one client item referencing this tag](#)

[The complex tag already references '%s' and will not be re-added](#)

[Complex tags cannot contain references to other complex tags](#)

[Tag '%s' does not exist or you are trying to add a dynamic or advanced tag reference which is not allowed](#)

[References to advanced tags are not supported \('%s'\)](#)

[References to array tags are not supported \('%s'\)](#)

[References to complex tags are not supported \('%s'\)](#)

[Unable to start tag '%s' \[Expression: '%s'\]](#)

[Unable to start trigger tag '%s' on tag '%s'](#)

[Unable to start complete tag '%s' on tag '%s'](#)

[Unable to start element tag '%s' on tag '%s'](#)

[Unable to start trigger tag '%s' for element '%s' on tag '%s'](#)

[Unable to start complete tag '%s' for element '%s' on tag '%s'](#)

[Unable to start required tag reference '%s' on tag '%s'](#)

[Attempt to add Oracle Connectivity Suite client item '%s' failed](#)

[Unable to start tag '%s' since it contains invalid tag references](#)

Derived Expression Error Messages

[Expression invalid: '%s'](#)

[Expression modified](#)

[Unrepresentable numeric constant: '%s'](#)

[Unterminated string](#)

[Unknown tag or misspelled keyword: '%s'](#)

[Syntax error: '%s'](#)

[", expected](#)

[\(expected](#)

[\) expected](#)

[Numeric expression expected](#)

[Please enter an expression](#)

[String tag expected](#)

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

[The advanced tag configuration cannot be modified since there is at least one client item referencing this tag](#)

[The complex tag already references '%s' and will not be re-added](#)

[Complex tags cannot contain references to other complex tags](#)

[Tag '%s' does not exist or you are trying to add a dynamic or advanced tag reference which is not allowed](#)

[References to advanced tags are not supported \('%s'\)](#)

[References to array tags are not supported \('%s'\)](#)

[References to complex tags are not supported \('%s'\)](#)

[Unable to start tag '%s' \[Expression: '%s'\]](#)

[Unable to start trigger tag '%s' on tag '%s'](#)

[Unable to start complete tag '%s' on tag '%s'](#)

[Unable to start element tag '%s' on tag '%s'](#)

[Unable to start trigger tag '%s' for element '%s' on tag '%s'](#)

[Unable to start complete tag '%s' for element '%s' on tag '%s'](#)

[Unable to start required tag reference '%s' on tag '%s'](#)

[Attempt to add Oracle Connectivity Suite client item '%s' failed](#)

[Unable to start tag '%s' since it contains invalid tag references](#)

The advanced tag configuration cannot be modified since there is at least one client item referencing this tag

Error Type:

Warning

Possible Cause:

1. A client is already using the tag.
2. Another advanced tag has been defined that is currently being referenced by a client. This, in turn, references the tag.

Solution:

1. Disconnect the client application from the server.
2. Have the client application remove the item so that it is no longer being used.

The complex tag already references '%s' 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.

Complex tags cannot contain references to other complex tags

Error Type:

Warning

Possible Cause:

An attempt was made to reference a Complex Tag from the Complex Tag Element List.

Solution:

Choose another tag to add to the element list.

Tag '%s' does not exist or you are trying to add a dynamic or advanced tag reference which is not allowed

Error Type:

Warning

Possible Cause:

1. A tag reference entered manually is incorrect.
2. A device address was referenced directly (i.e., dynamic tags).
3. While creating or editing the server project in XML format, a tag that was being referenced was removed or renamed.

Solution:

1. Reference only Static tags in the server.
2. When manually entering tag names or editing the project file, use the correct tag names.

References to advanced tags are not supported (%s)

Error Type:

Warning

Possible Cause:

An attempt was made to reference an Advanced Tag.

Solution:

Choose a Device Tag.

References to array tags are not supported ('%s')

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.

References to complex tags are not supported ('%s')

Error Type:

Warning

Possible Cause:

An attempt was made to reference a Complex Tag in an Advanced Tag.

Solution:

Reference a different type of tag.

Unable to start tag '%s' [Expression: '%s']

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 '%s' on tag '%s'

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 '%s' on tag '%s'

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 '%s' on tag '%s'

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 '%s' for element '%s' on tag '%s'

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 '%s' for element '%s' on tag '%s'

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 '%s' on tag '%s'

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.

Attempt to add Oracle Connectivity Suite client item '%s' failed**Error Type:**

Warning

Possible Cause:

1. An item referenced in the connector no longer exists in the server project.
2. While saving and editing the project in the XML format, an item referenced in the connector has had its name changed.

Solution:

1. Rename or re-add the item being referenced from the connector.
2. Remove the item reference from the connector.

Unable to start tag '%s' since it contains invalid tag references.**Error Type:**

Warning

Possible Cause:

1. The referenced tag may be missing or renamed.
2. There may be a circular reference where TagA references TagB which references TagA.

Solution:

1. Add the tag back into the project.
2. Reference the correct tag.
3. Correct the tag name.
4. Remove the circular reference.

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

[Expression invalid: '%s'](#)

[Expression modified](#)

[Unrepresentable numeric constant: '%s'](#)

[Unterminated string](#)

[Unknown tag or misspelled keyword: '%s'](#)

[Syntax error: '%s'](#)

[", expected](#)

[\(expected](#)

[\) expected](#)

[Numeric expression expected](#)

[Please enter an expression](#)

[String tag expected](#)

Expression invalid: '%s'**Error Type:**

Information

Possible Cause:

The expression failed validation.

Solution:

Modify the expression so that it conforms to proper syntax and references tags which exist in the Configuration.

Expression modified

Error Type:

Information

Possible Cause:

The saved expression was modified.

Solution:

Accept the modification so that it will keep.

Unrepresentable numeric constant: '%s'

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.

Unknown tag or misspelled keyword: '%s'

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 Help File.

Syntax error: '%s'

Error Type:

Warning

Possible Cause:

The expression failed a syntax check.

Solution:

Verify the syntax, including the placement of all parentheses.

", 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 parenthesis where they belong.
2. Place an operator between the parentheses.

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.