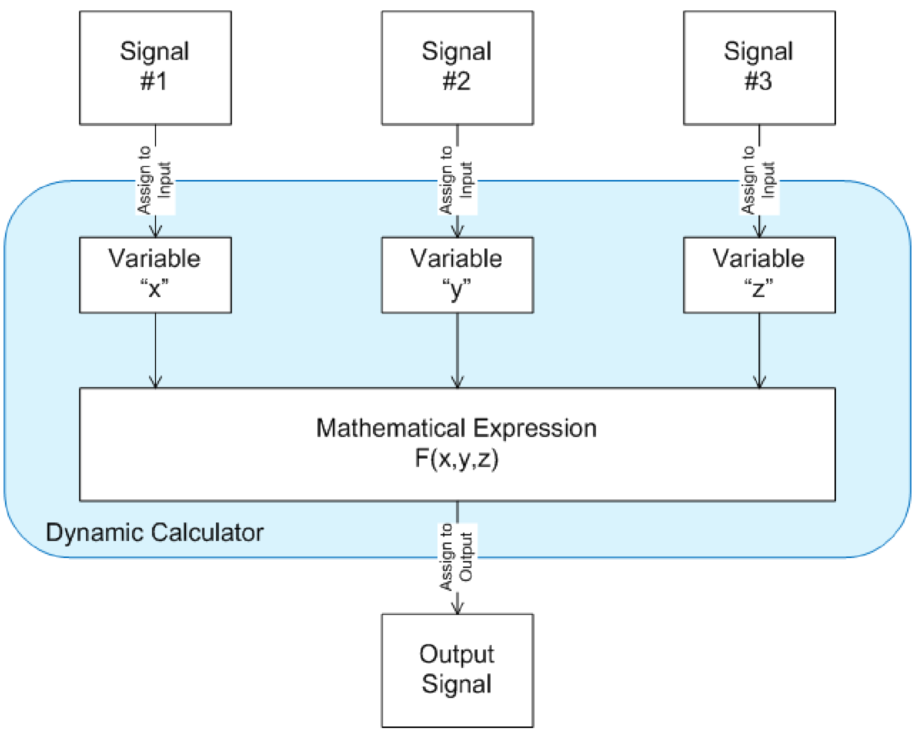
August 2012

Dynamic Calculator in openPDC and openPG

The Dynamic Calculator is an action adapter that can be used to create a calculated signal value. This calculated value is computed at the rate which is configurable and typically the rate of the incoming signals. It's not recommend that this adapter not be used for signals that span large difference in periodicity (e.g., a 4-second down-sampled value with 60 sample-per-second phasor data.)

The purpose of this action adapter is to compute a value that can be used in near-real time processes prior to data archival or other phasor data processing. The figure below shows how the dynamic calculator works for an example of three input values, or signals.



Dynamic Calculator Conceptual Data Flow

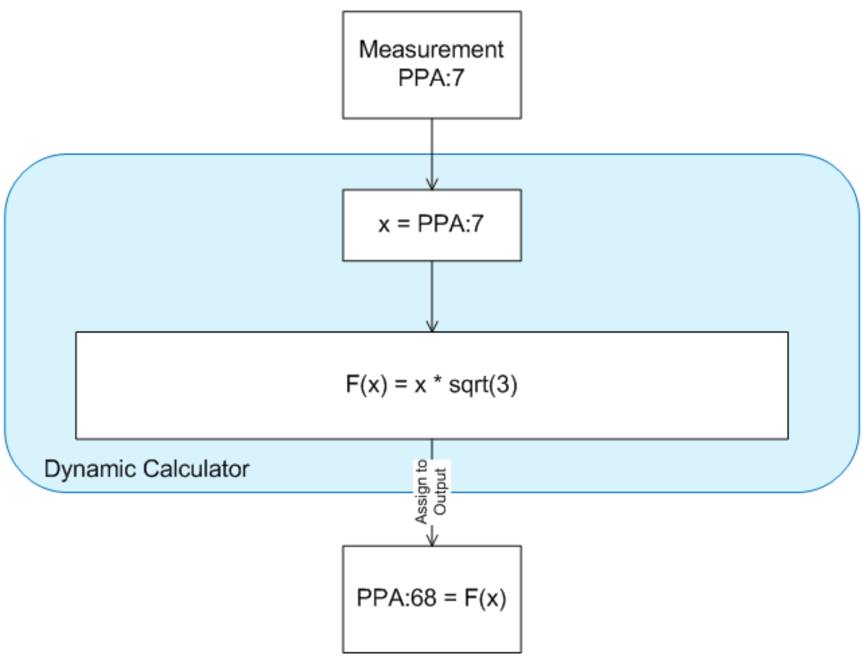
The Dynamic Calculator is configured through the "Manage Action Adapters" configuration screen where the "mathematical expression" is entered. This expression can include trigonometric and other functions as shown on page 16.

A description of how to use the Dynamic Calculator is provided through two examples, (1) The simple example of multiplying a signal by a scalar and (2) a more involved example of adding two phasors (in polar form) and returning a polar result.

Example 1 – Multiply a Voltage by √3

This is a simple mathematical function to demonstrate in detail how to set up a Dynamic Calculator as show by the data flow diagram below.

The input signal is shown below as referenced by the "Measurement Key" which is the preferred manner to reference a signal for this adapter. However, point tag or point GUID can both be used as well.



Example 1 – Data Flow

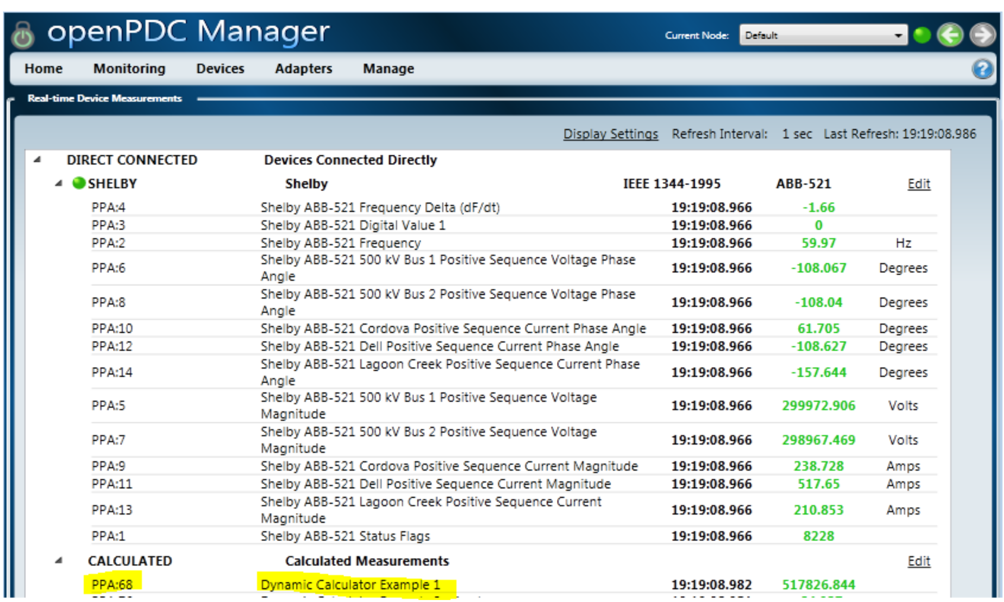
In general, the steps for configuration of the Dynamic Calculator are:

1. Create the output signal that will receive the results of this calculation (e.g., PPA:68)
2. Find the designed input signals (e.g., PPA:7)

3. Create the Action Adapter and build the mathematical expression in the connection string for this adapter. This expression can consist of standard operators, parenthesis, and functions as listed on Page 16.

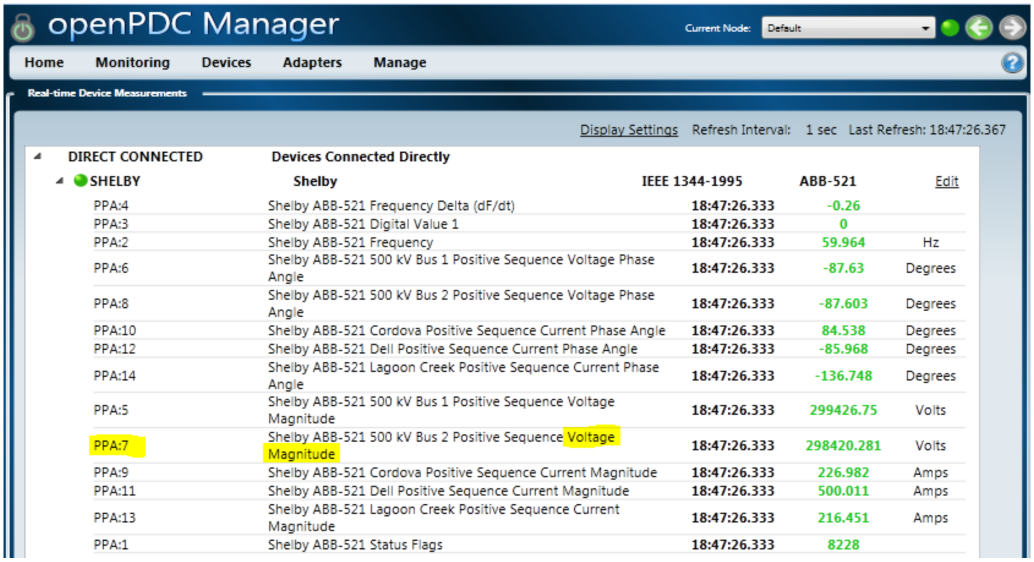
Step 1 – Using the Manage – Measurements screen, create a new signal which will receive the results of the calculation. The Point Tag should be descriptive of this signal. Note that Internal and Enabled are both checked.

The “Measurement Key” for this new output signal is created by the openPDC. Go to menu Monitoring – Device Measurement to find the Measurement Key – e.g., PPA-68.

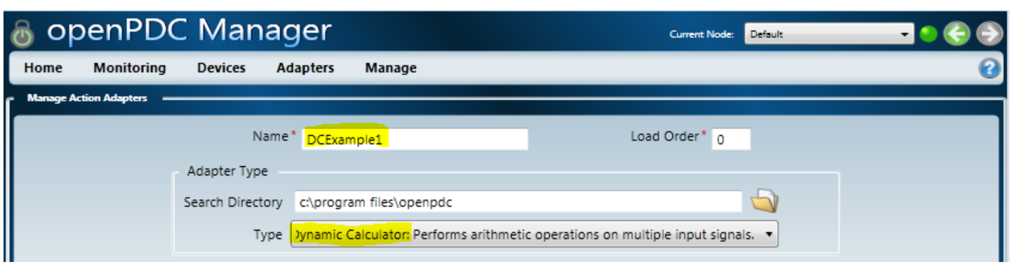


Grid Protection Alliance

Step 2 – Find the Measurement Key or Point Tag of the input signal to be used in the calculation by browsing through PDC measurements. e.g., The Measurement Key is PPA:7 as shown below.



Step 3 – Create Dynamic Calculator in menu Adapters - Action Adapters by dropping down to "Dynamic Calculation" under "Type" as shown below and entering in a unique name for this action adapter, e.g., "DCExample1".

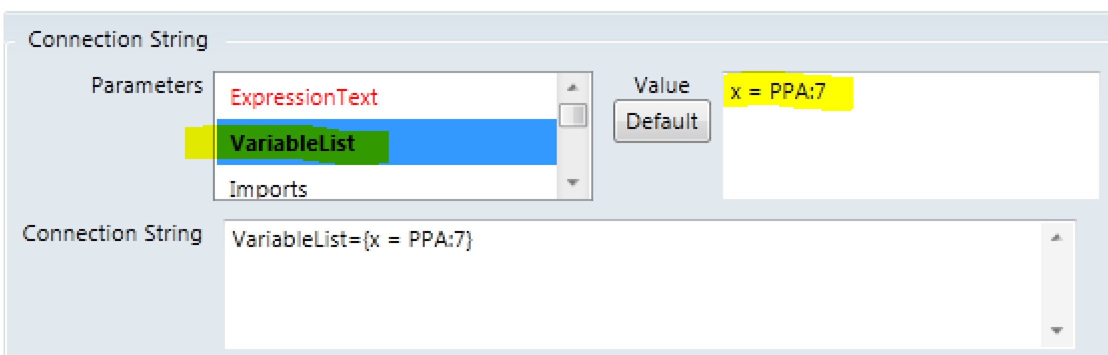


Step 4 – Enter the required the connection string parameters, including the mathematical expression.

All the required parameters for the connection string are shown in red in the parameters list dialog box as shown in the figure below. These required parameters are:

- VariableList
- ExpressionText
- OutputMeasurements
- FramesPerSecond
- LagTime
- LeadTime

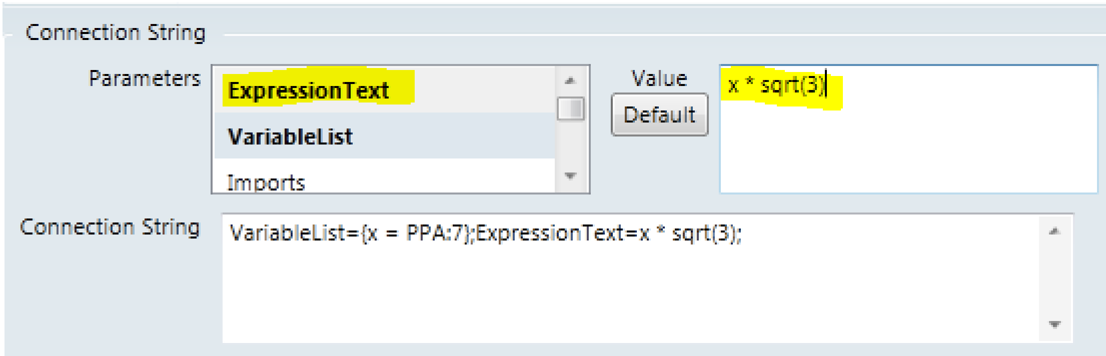
VariableList: First assign the input signal (PPA:7) to an arbitrary variable name, e.g., “x”. (The parameters in red in the parameters drop down are required and missing. The ones in bold have been set to values other than the default value.)



Note: This step is required when using the Point Tag or GUID to assign a value to the variable. As seen below, Measurement Keys can be used directly in the calculation expression without an assignment to a variable.

ExpressionTest: Enter the f(x), in this case “x” times the square root of three. While value could have been entered as a constant (1.732), the SQRT function was applied.

This expression could also have been entered as “PPA:7*SQRT(3)” without the definition of “x”.



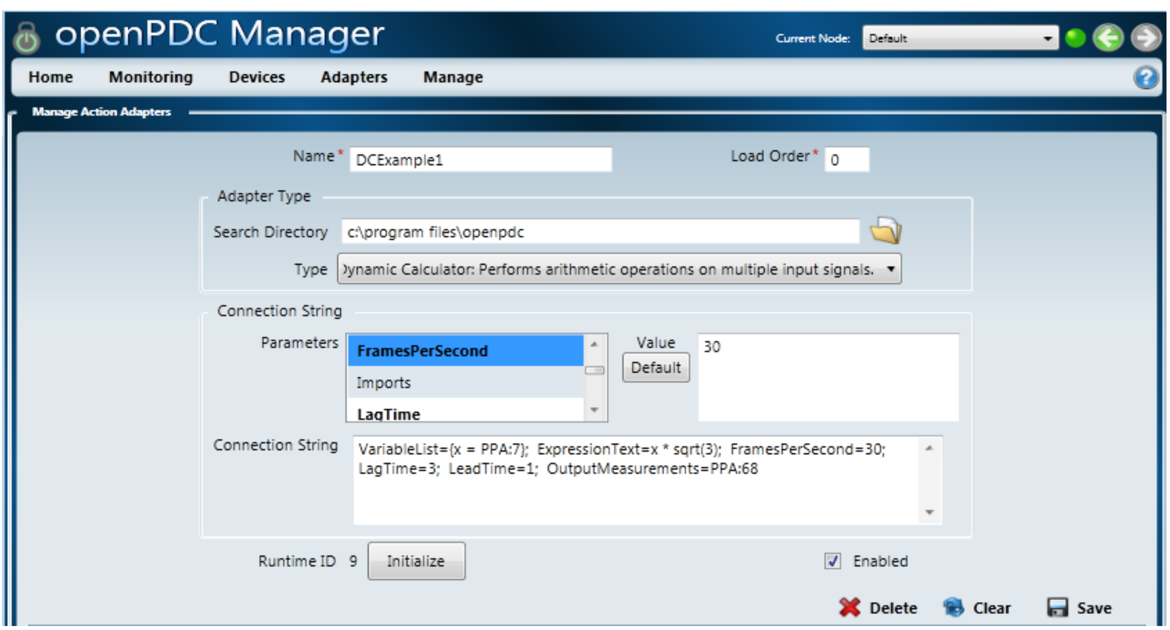
OutputMeasurements: Assign the created signal with Measurement Key = PPA:68 to the output.

FramesPerSecond: 30 – The calculation periodicity.

LagTime: 3.0 seconds (Wait for up to 3 seconds for all input variables to be initialized by a signal. The lag time for this adapter should be set to match other lag times in the system.)

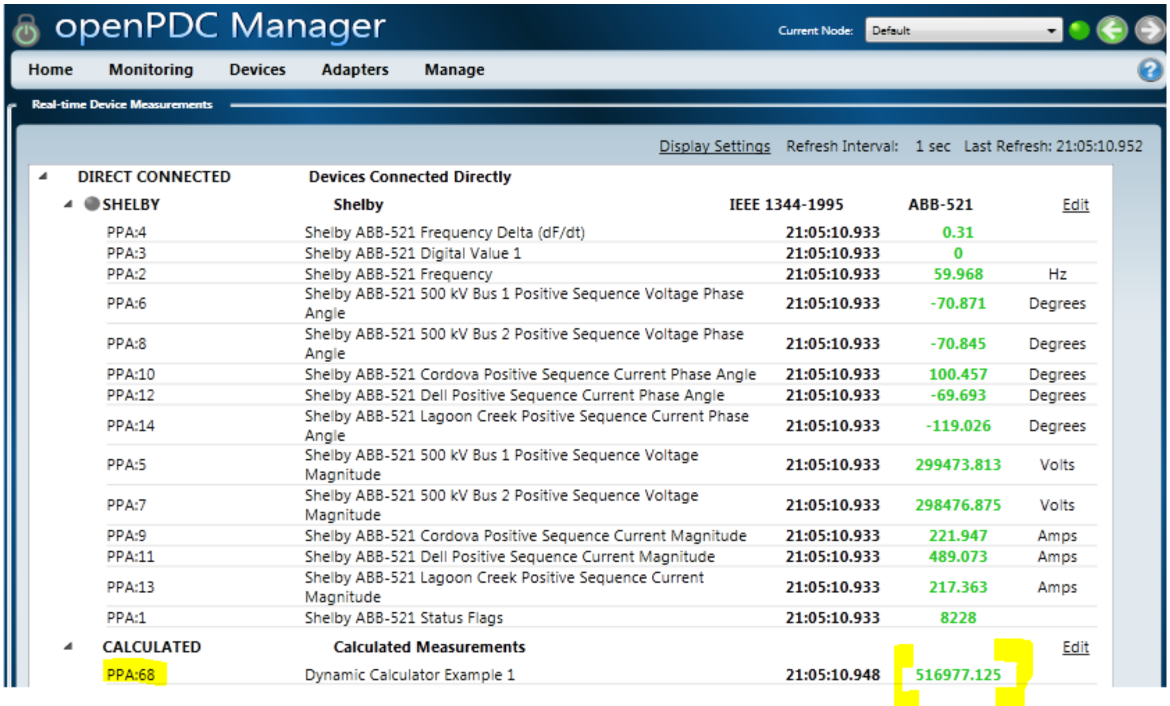
LeadTime: 1.0 seconds (Since the calculation is always performed on signals with matching timestamps, setting the lead time to a high value allows for large amounts of local clock drift prior to discarding the calculation based on a bad "future" time. A lead time of 1 second is shown in this example. It could have been set to 10 seconds.)

This is the completed Dynamic Calculator for Example 1:



Click the Initialize button to start the Calculations.

Confirm operation at menu Monitoring – Device Measurement:



Example 2 – Addition of Two Phasor Values

This example is more involved. Using two phasor values in polar coordinates it computes a resulting phasor, also in polar coordinates.

For this example, Measurement Keys are used to refer to signals. This reference can also be the Point Tag or Point GUID.

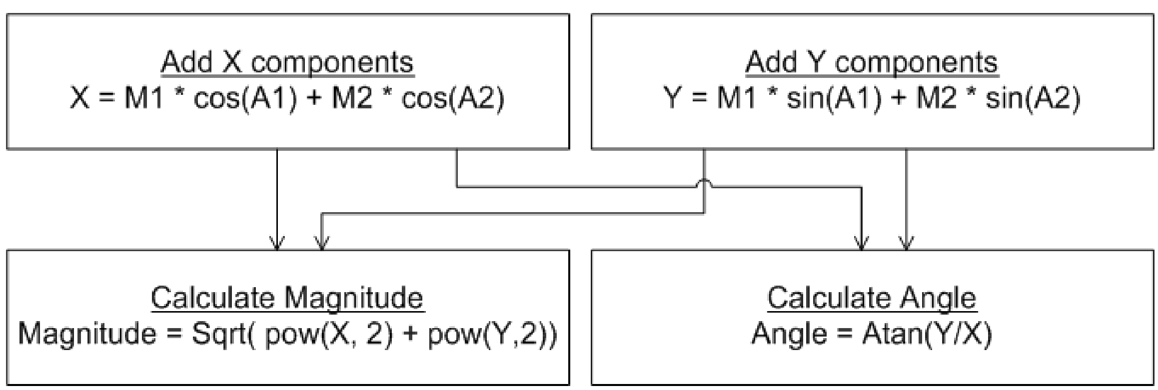
The inputs represent four (4) signals:

1. Magnitude and Angle (M1 and A1)
2. Magnitude and Angle (M2 and A2)

To perform this calculation, each phasor is converted to rectangular notation, added, then the result is converted to polar notation as shown below.

With the limitations of the Dynamic Calculator this requires four instances of the Dynamic Calculator adapter as represented by each box below.

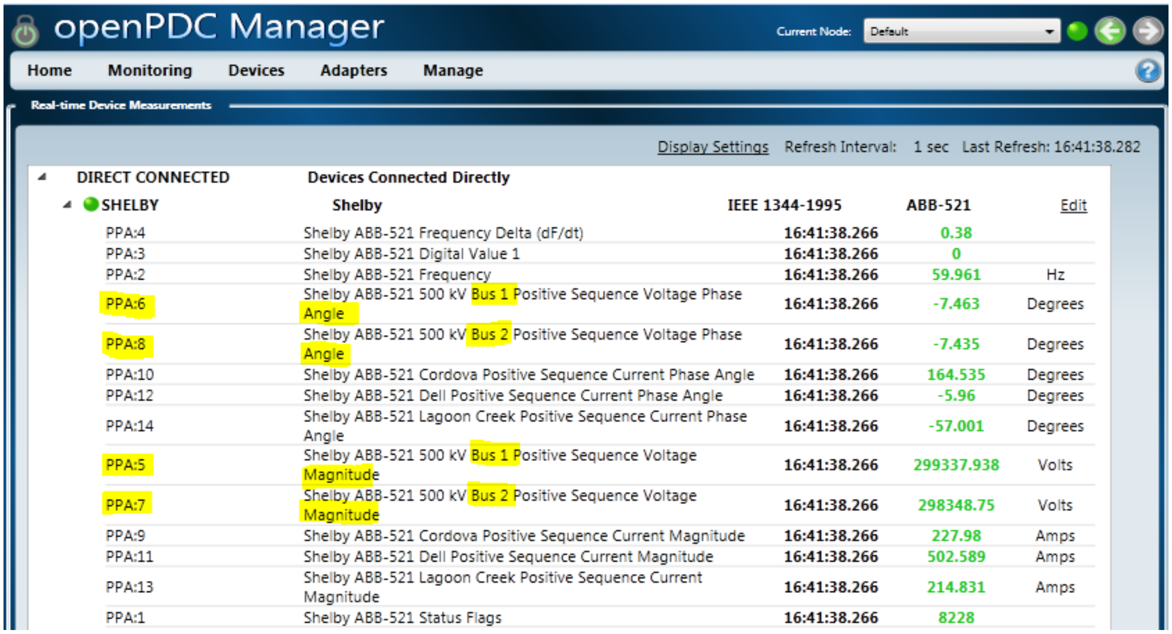
1. Add X components
2. Add Y components
3. Compute magnitude of result
4. Compute angle of result



This requires

- Identifying four input adapters with Magnitudes and Phase Angles.
- Create two Dynamic Calculator Action Adapters for the X and Y components.
- Create two Dynamic Calculator Action Adapters for the resulting Magnitude and Phase Angle.

Step 1 - Identify Input Adapters



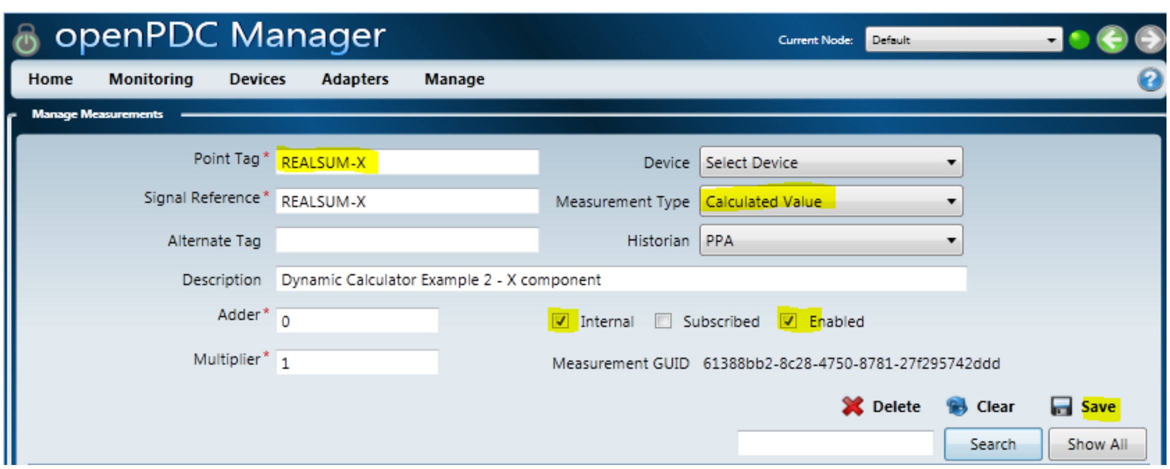
PPA:5 – Bus 1 Magnitude

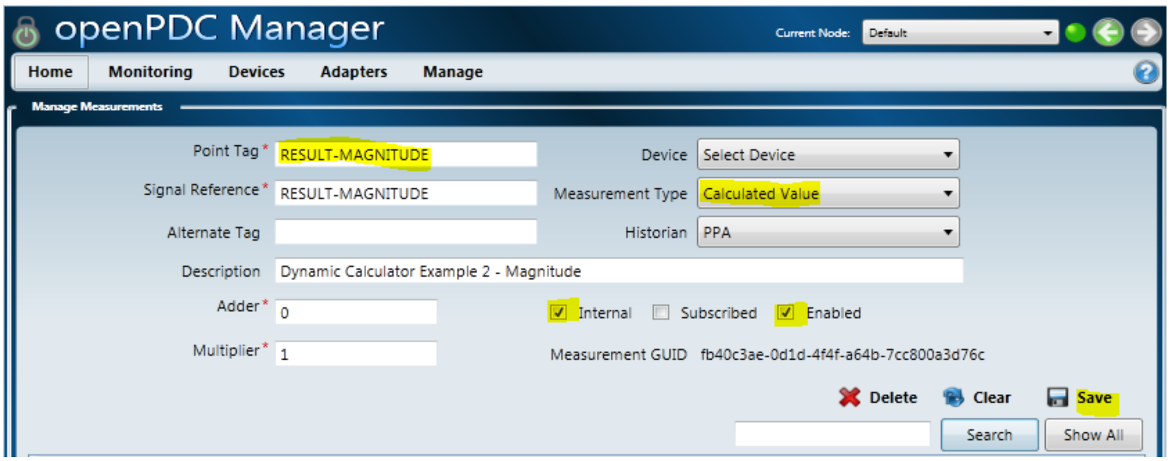
PPA:6 – Bus 1 Angle

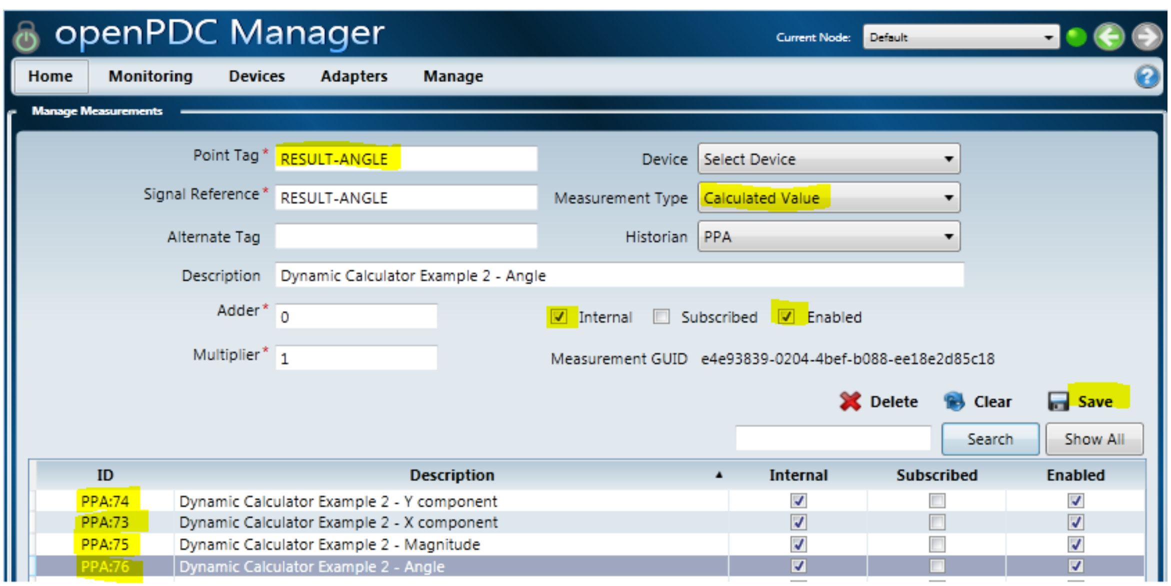
PPA:7 – Bus 2 Magnitude

PPA:8 – Bus 2 Angle

Step 2 – Create an Output for the Dynamic Calculator results in menu Manage - Measurements:

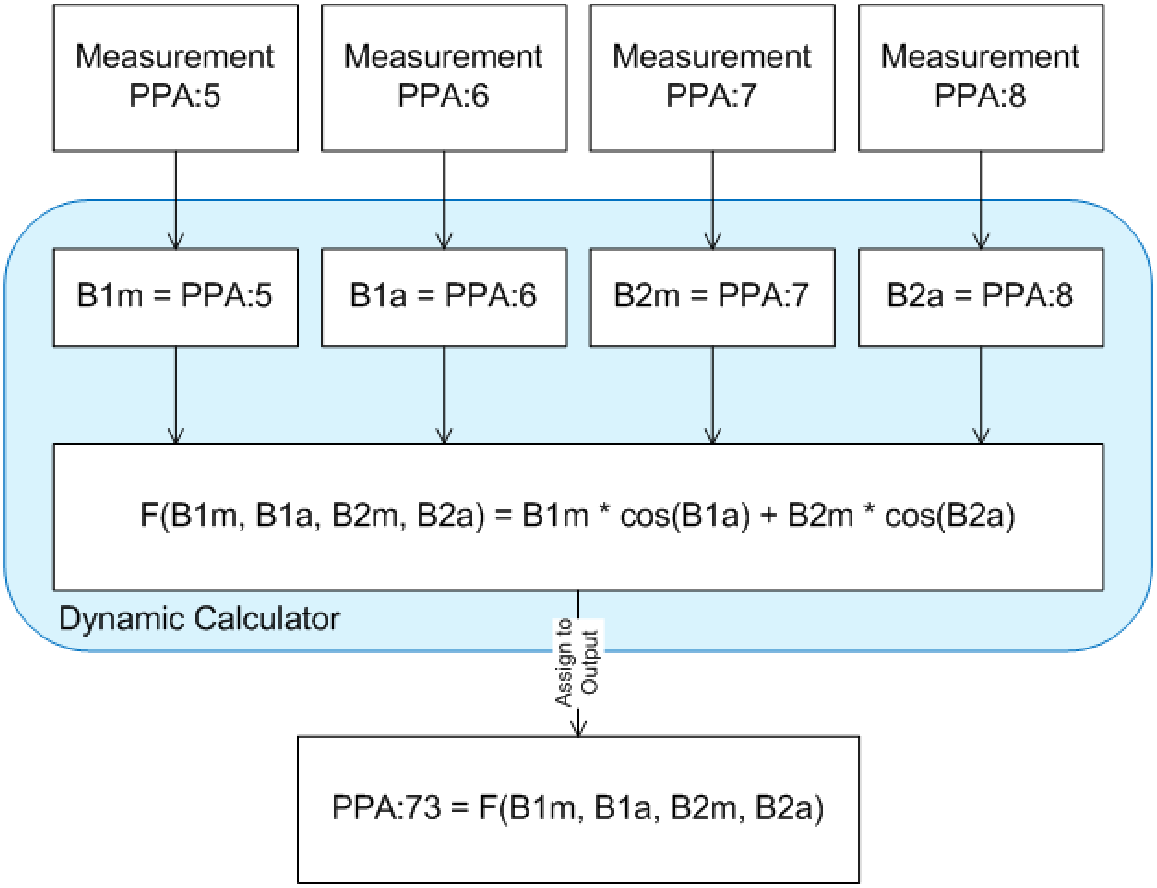




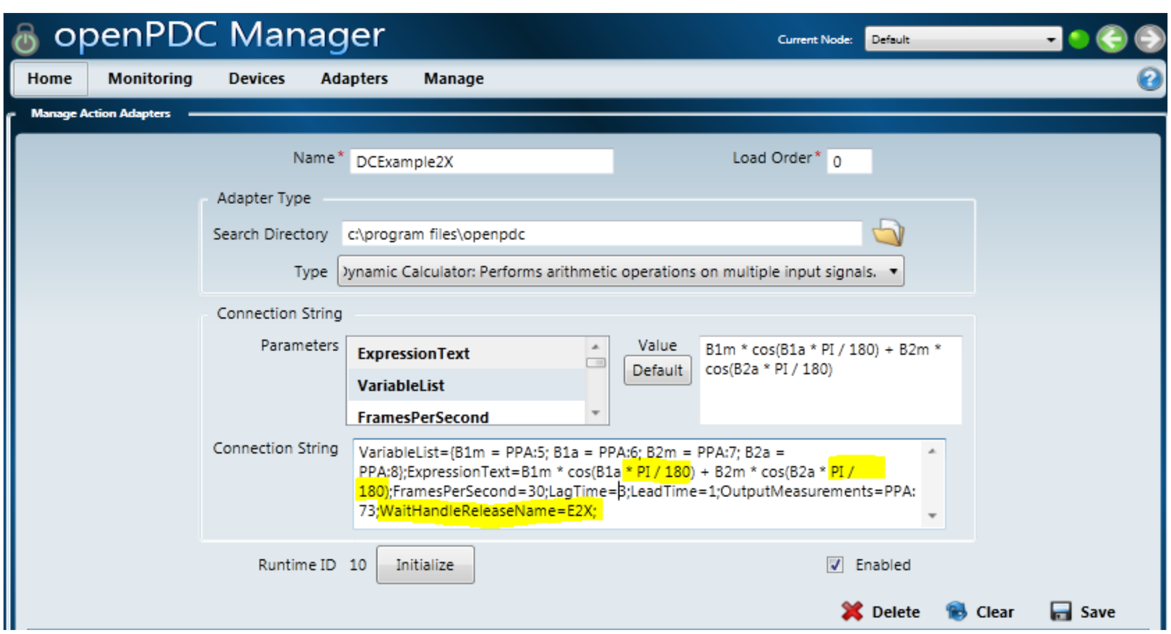


Notice the list of ID's for each Output. This will be the IDs used to define where the calculated values will be saved.

Step 3 – Create Dynamic Calculator for X Components



This is the completed Dynamic Calculator for the X components:

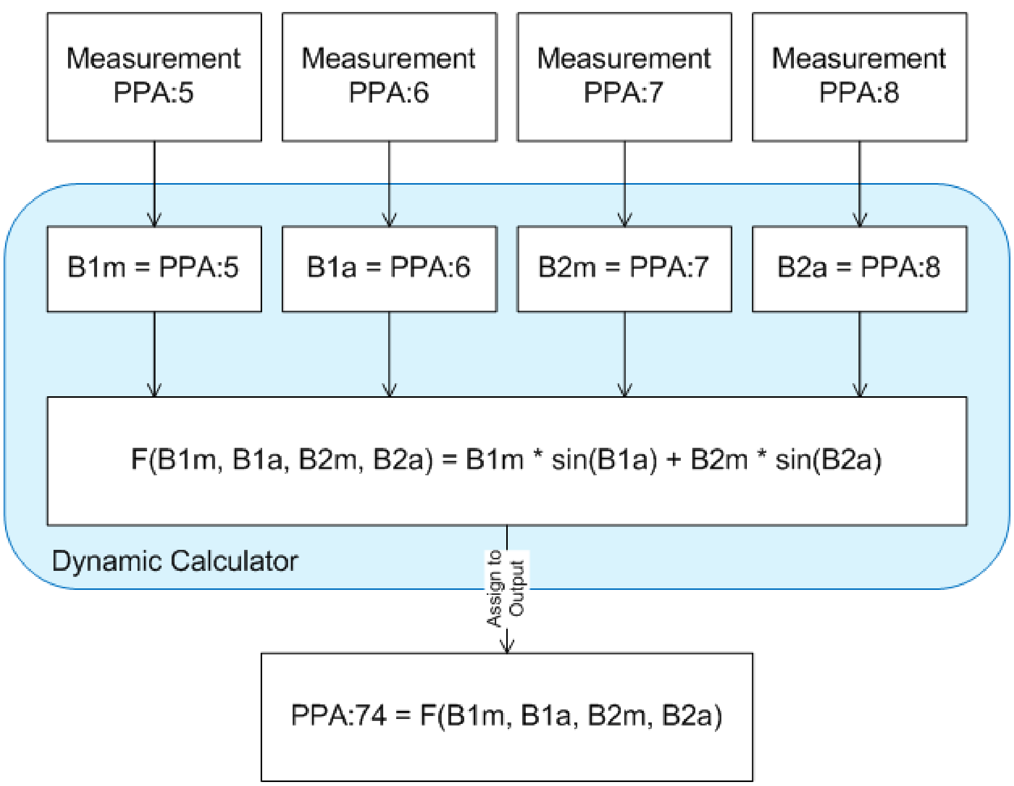


There are several things to note:

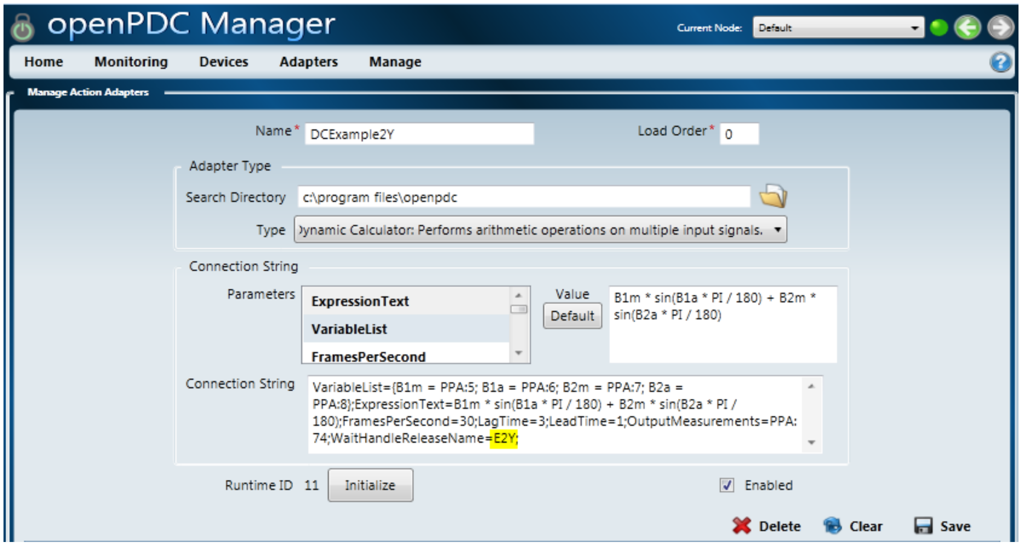
- The VariableList has four definitions and is contained in { } brackets in the Connection Stream. Between each definition is a semi-colon. This is the required syntax.
- The cosine (and sine) function expects the angle to be in radians. This requires the PI/180 conversion.
- There is a new parameter on the list – WaitHandleReleaseName. The Magnitude and Angle calculations are dependent on this calculation as well as the Y component calculation. In order to keep the dependent calculation in sync, the Wait Handle was created. When this calculation is completed, the dependent calculations are release to perform their calculations. 'WaitHandleReleaseName' is a parameter that is given a name – any non-standard or un-used name – to notify the dependent calculations.

Click the Initialize button to start the Calculations.

Step 4 – Create Dynamic Calculator for Y Components

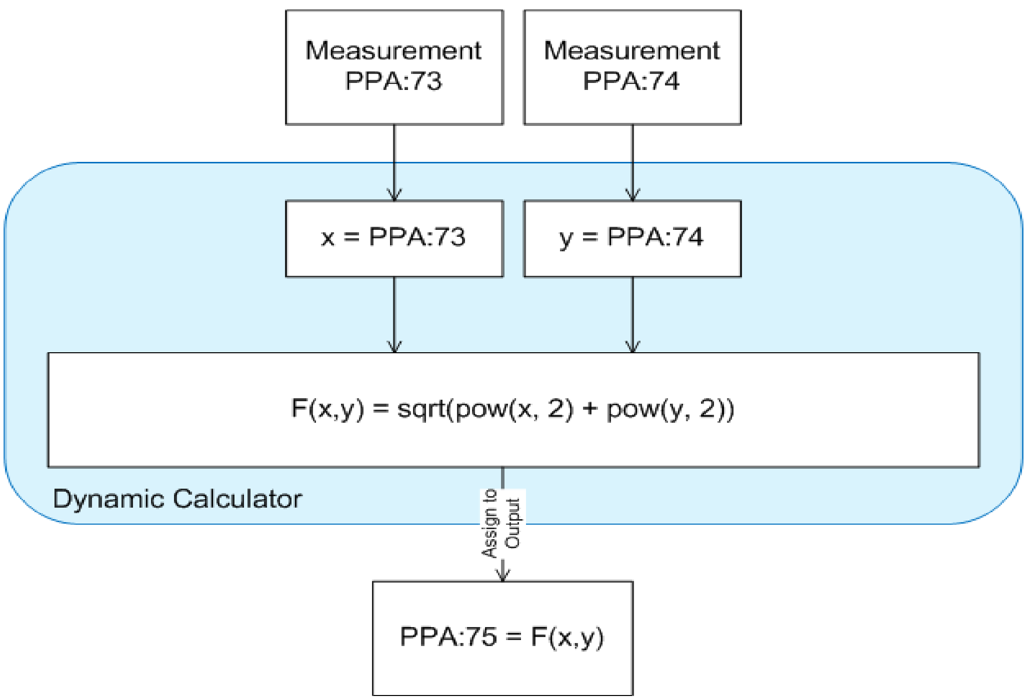


This is the completed Dynamic Calculator for the Y components:

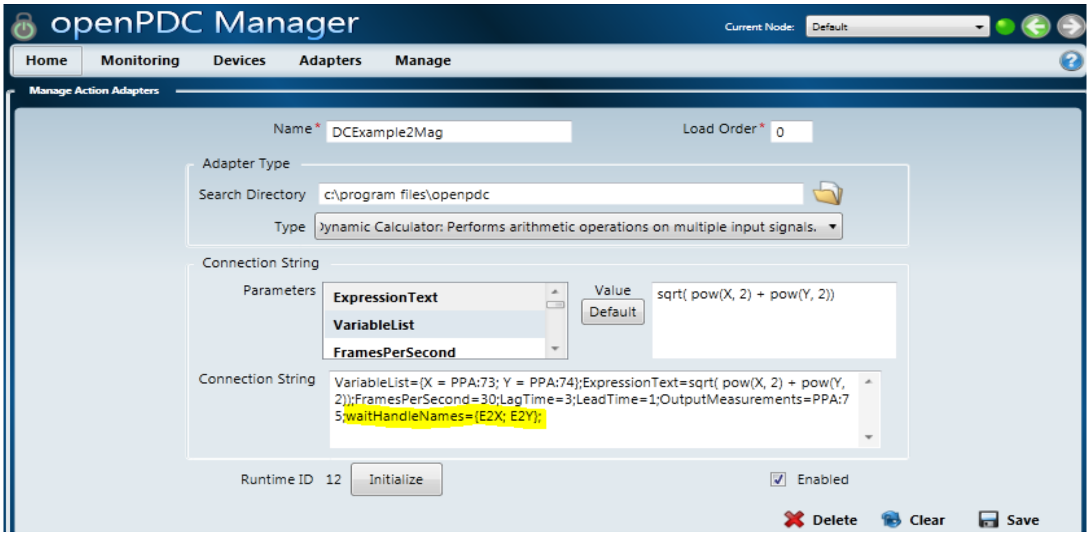


Click the Initialize button to start the Calculations.

Step 5 - Create Dynamic Calculator for Magnitude Result



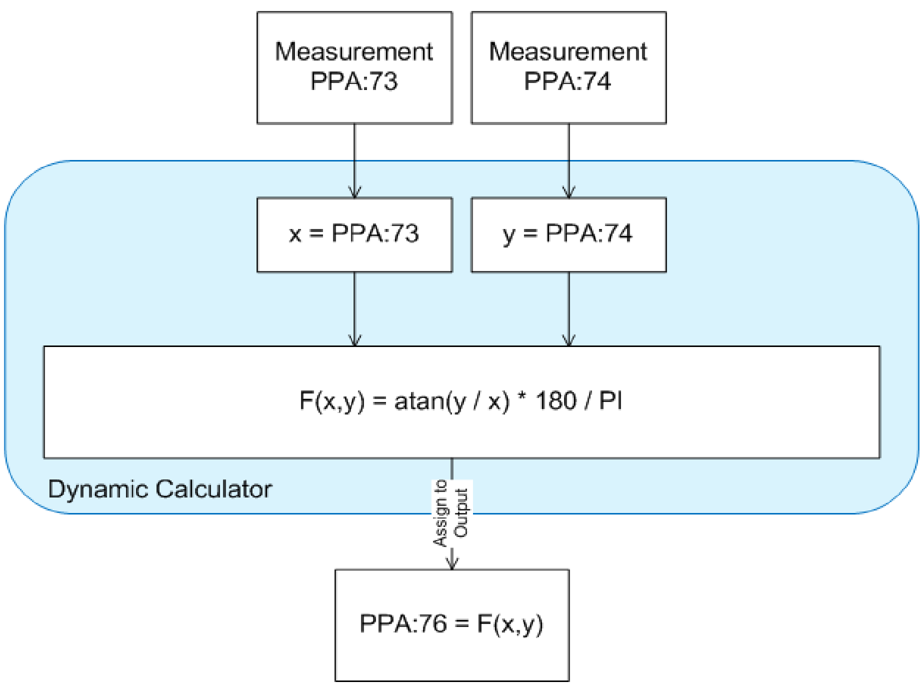
This is the completed Dynamic Calculator for the Magnitude results:

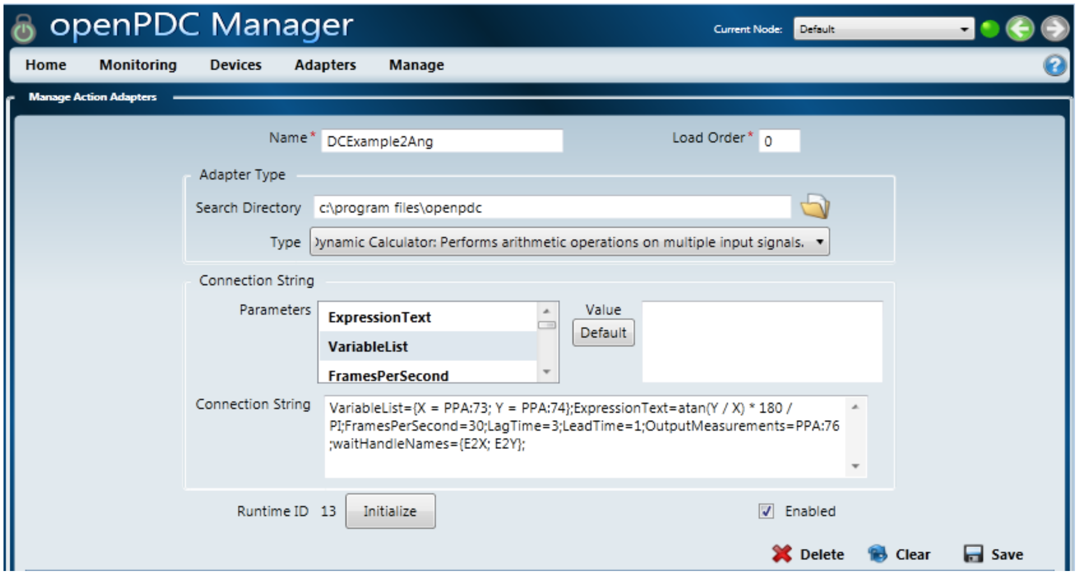


Note: Notice the 'waitHandleNames' parameter. This is how the 'WaitHandleReleaseName' is referred to for the required calculations. In usual standard method the required calculation names are separated by a semicolon and enclosed in { } brackets.

Click the Initialize button to start the Calculations.

Step 6 - Create Dynamic Calculator for Angle Result



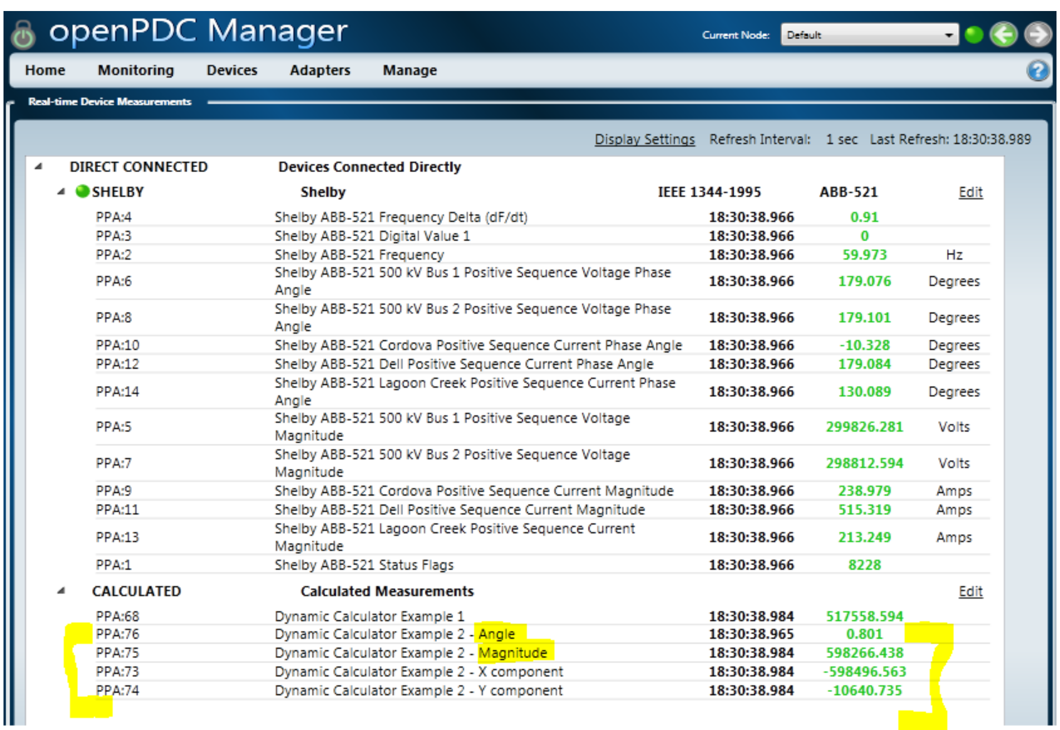


NOTE: Just as Sin and Cos require radian degrees to properly return values, the aTan returns in radian and must be converted in order to display degrees.

Click the Initialize button to start the Calculations.

Final Results:

Confirm operation at menu Monitoring – Device Measurement:



Available Functions

This is a listing of the functions available. This comes directly from the .NETv4 Math Class. Many functions are capable of accepting several number types (overloaded).

- Abs(x) - Returns the absolute value of the specified number, x.
- Acos(x) - Returns the angle whose cosine is x.
- Asin(y) - Returns the angle whose sine is y.
- Atan(z) - Returns the angle whose tangent is z
- Atan2(y, x) - Returns the angle whose tangent is the quotient of y / x
- BigMul(x, y) - Produces the full product of two numbers, x and y.
- Ceiling(x) - Returns the smallest integer value that is greater than or equal to the specified number, x.
- Cos(a) - Returns the cosine of the specified angle, a.
- Cosh(a) - Returns the hyperbolic cosine of the specified angle, a.
- DivRem(x, y, r) - Calculates the quotient of two specified number, x and y, and also returns the remainder in an output parameter, r.
- E – Returns the constant **e**
- Exp(x) - Returns **e** raised to the specified power, x.
- Floor(x) - Returns the largest integer less than or equal to the specified number, x.
- IEEERemainder(x, y) - Returns the remainder resulting from the division of a specified number, x, by another specified number, y.
- Log(x) - Returns the natural (base **e**) logarithm of a specified number, x.
- Log(x, b) - Returns the logarithm of a specified number, x, in a specified base, b.
- Max(x, y) - Returns the larger of two numbers.
- Min(x, y) - Returns the smaller of two numbers.
- PI – Returns the constant **π**
- Pow(x, p) - Returns a specified number, x, raised to the specified power, p.
- Round(x) - Rounds a specified number, x, to the nearest integer value.
- Round(x, d) - Rounds a specified number, x, to a specified number of fractional digits, d.
- Sign(x) - Returns a value indicating the sign of x. [-1, 0, 1]
- Sin(a) - Returns the sine of the specified angle, a.
- Sinh(a) - Returns the hyperbolic sine of the specified angle, a.
- Sqrt(x) - Returns the square root of a specified number, x.
- Tan(a) - Returns the tangent of the specified angle, a.
- Tanh(a) - Returns the hyperbolic tangent of the specified angle, a.
- Truncate(x) - Returns the integral part of a specified number, x.

ABOUT THE AUTHOR

Shawn Williams is a project manager at GPA with extensive experience within the process control industry.