EIGEN INGENUITY

eigen+

# Product Information Sheet

## Data Transformation Engine

*(Virtual/Synthetic Timeseries and The Calculation Historian)*

## What is the Data Transformation Engine

The Data Transformation Engine applies "in-line" transforms to any timeseries data source connected to Ingenuity. Anything from simple mathematical operators such as Add, Subtract, Multiply and Divide, through more complex Totalisers and statistical functions like Average and Standard Deviation, logical If..Then functions and complex transforms such as Timeshifting.

Transforms can be nested, and multiple transforms can be applied sequentially.

The output of a transform is a Virtual or Synthetic timeseries that can be used in the Ingenuity in exactly the same way other Timeseries sources.

## What are Virtual Timeseries?

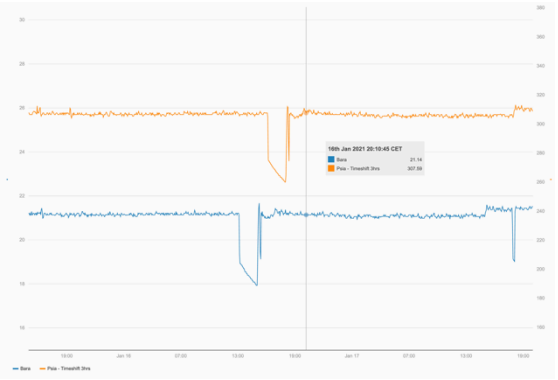
Virtual timeseries behave exactly like 'normal' (i.e. real or raw) signals stored as tags in process data historians but they are evaluated only when they are called. They can be used in exactly the same way as normal tags in Ingenuity.

For example, you can set limits & alerts or use as the basis for another virtual timeseries, but they are evaluated at runtime and do not need the values to be written back to a database.

The advantage of this is that they can be immediately plotted for any time period for which the source data exists, but they do not fill up storage space.

### Example – unit conversion

If we are measuring a pressure in 'bara' but want to plot it in 'psia' then it must be transformed by multiplying it by 14.5. This lets us experiment and create quick calculations for analysis without worrying about filling up memory or waiting until historical values are recalculated.



*The orange series is a virtual timeseries created by multiplying the original tag (blue) in 'Bara' by 14.5 to get 'Psia' and then timeshifting the results by 3 hours forwards.*

Virtual timeseries can be nested and multiple different functions combined in a single timeseries to produce complex calculations.

## Editing the Data Transformation Equations

Ingenuity 7's virtual calculation graphical editor is an easy-to-use drag-and-drop user interface in which any Ingenuity user can quickly configure complex calculations while minimizing human error.

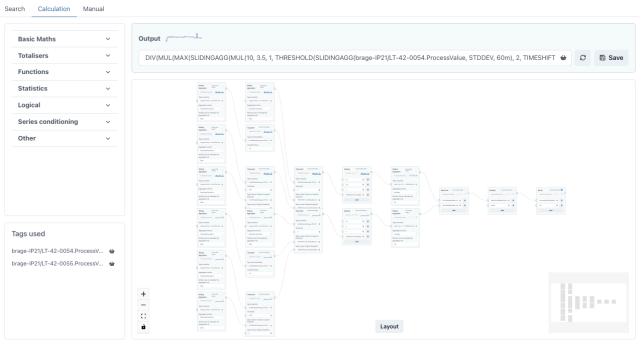
Transform function blocks are dragged and dropped into a canvas, after which inputs and outputs can be connected to compose any complex transformation.

## Transform Functions

Choose from the following 36 transform functions to create simple or complex calculations and synthetic timeseries.

| | |
|---|---|
| Basic Maths | Add |
| | Subtract |
| | Multiply |
| | Divide |
| | Percent Deviation |
| Totalisers | Totalise |
| | Totalise Raw |
| Function | Exponential |
| | Natural Log |
| | Log |
| | Square Root |

| | |
|---|---|
| | Power |
| Sliding Aggregates (normal and raw) | Average |
| | Count |
| | Number of Bad Points |
| | Number of Good Points |
| | Standard Deviation |
| | Variance |
| | Minimum |
| | Maximum |
| | Sum |
| Statistics | Maximum |
| | Minimum |
| | Mean |
| | Median |
| | Standard Deviation |
| | Variance |
| | If Tag Exists |
| Logical | If Equals |
| | Threshold |
| Series Conditioning | Stepped |
| | Stepped Raw |
| | No BAD |
| | Timeshift |
| | Timeshift Dynamic |
| Other | Point in Time |



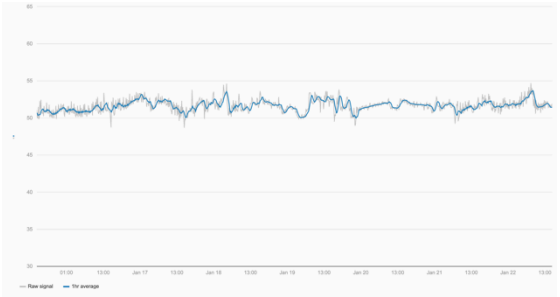
*Canvas of a complex chemical dosing transformation function.*

*Detail of the complex transformation function shown above.*

# Example use cases

## Smoothing noisy signals

Averages can be applied to smooth out noisy signals



*By applying an averaging function with a 1-hour window the underlying trends in the data can be seen more easily.*

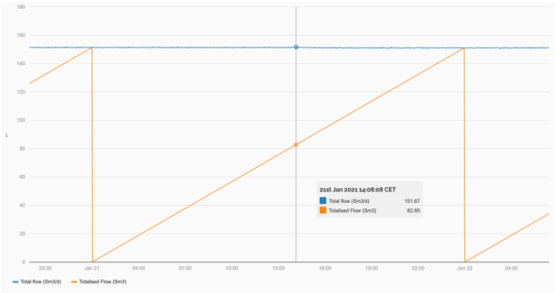
## Logical expressions

The green series in the trend below uses the Threshold function to create a virtual timeseries of the flow only when Pump B is running.

*By using a Threshold function to only display the flowrate if the running signal is >0.5, a virtual timeseries is created that could drive a virtual flowmeter graphic.*

## Totaliser

Using the Totaliser function allows a value to be totalised over any window from 1 minute to years.



*A flowrate in Sm3/d is totalised over 24hrs starting at midnight, giving the cumulative flow so far in any given day (orange line).*

## Combining two functions

Combining a Totaliser function with a Threshold function we can combine the two examples above to see the total amount of fluid pumped by Pump B in a year.

*The virtual timeseries of the flow through Pump B is totalised in a second virtual timeseries over a window of a year starting at midnight on the 1st January.*

## Taking the Maximum of several tags

The Max function lets users create a virtual timeseries that only shows the maximum value at any time from two or more tags.


*The upper trend shows the readings from two different transmitters. Using the Max function, we are able to create a virtual timeseries (lower trend – black line) that always shows the maximum reading from either of these two transmitters.*