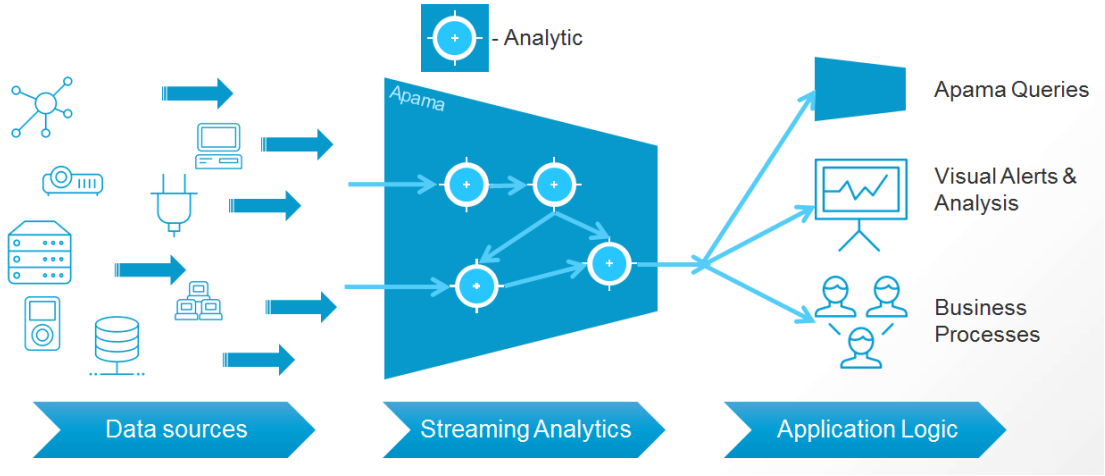
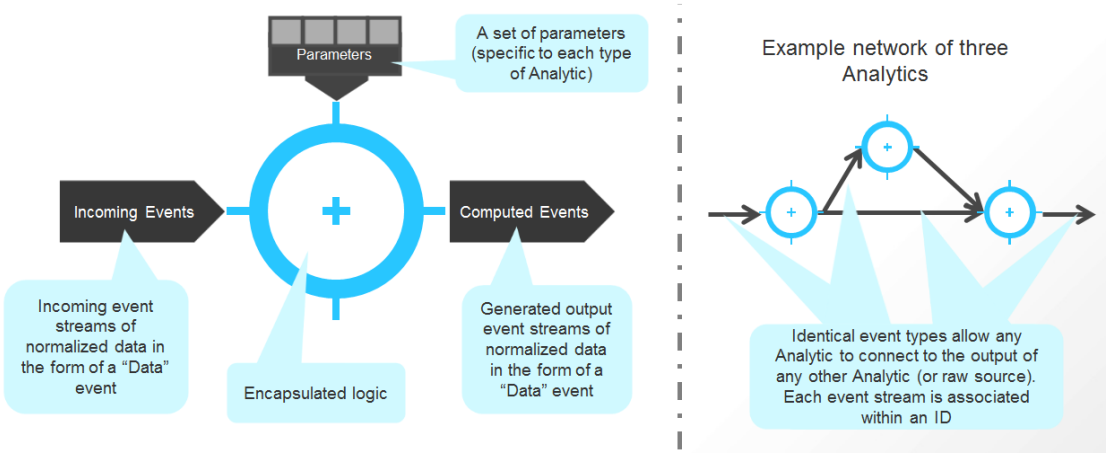
Industry Analytics Kit

Overview

The Industry Analytics Toolkit consists of a set of re-usable and scalable “micro services” that perform a range of streaming analytics, transformations, and detections over event data streams. The scope of the analytics within the Toolkit is that they consume and generate defined “Data” events. The elements within the Industry Analytics Kit are referred to here as “Analytics”.



Analytics have to have the ability to listen for Data event so that they can use incoming raw data, and the ability to send Data events to other Analytics or send them externally so that Apama Queries and other component can be used if required. In this way streaming networks of analytics can be defined.



Most Analytics are implemented such that they automatically multiplex the incoming data streams; this means that all data values for all data sources on a single stream fed into a single Analytic, e.g. Average, will incur separate (moving average) calculations for each individual source ID, and each

output from that Analytic will be the updated moving average for that specific data source. So, a single instantiated Analytic can perform the required calculation for any number of individual data sources on a single stream. Should data source values need to be combined, then an Aggregator Analytic may be used in front of the required calculation Analytic (s).

Apama Event Structures

A "Data" event is a normalized event structure that will be used within the toolkit for passing around values (both from raw data sources and Analytics). Events coming in from external sources will need to be converted into this event format prior to use:

```
event Data {
   string    streamName;          The name of the data stream
   string    type;                Enumerated type(Measure Types) describing
                                  this event as ANOMALY, COMPUTED or RAW
   string    sourceId;            The id of the source of the data
   decimal   timestamp;           The timestamp of the measurement
   decimal   dValue;              The value of the measurement as a decimal
   string    sValue;              The value of the measurement as a string
   float     xValue;              Three floats intended for supplementing data
                                  in dValue or sValue (such as a location)
   float     yValue;              //
   float     zValue;              //
   dictionary <string, string>    To hold any other data associated with the
                  extraParams;    measurement
}
```

The intent is to have a set of Analytics that can be chained together as desired. They are created though the generation of a single event type. The structure of this "Analytic" event type is:

```
event Analytic {
   string name;                                  The name of this new node
   sequence <string> inputMeasureNames;          The stream(s) to receive
   sequence <string> outputMeasureNames;         Names of the stream(s) to
                                                 generate
   dictionary <string, string> params;           A list of parameters to customize
                                                 this instance
}
```

For example:

```
Analytic("Average", ["Input"] , ["1MinMA"], {"timeWindow":"60.0d"})
```

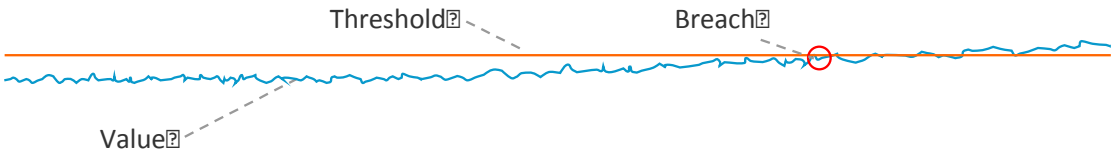
This calculates a 60 second moving Average of "Input" feed and generates a "1MinMA" feed, also:

```
Analytic("Volatility", ["1MinMA"], ["Vols"] , {"timeWindow":"3600.0d"})
```

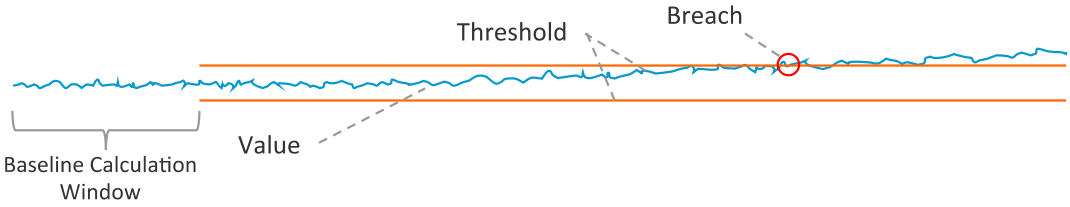
Will calculate a 1 hour Volatility (variance) calculation over the earlier calculated moving average feed (from 1MinMA), and distribute this on the "Vols" feed for further consumption.

Example Analytics

The Threshold Analytic will generate alerts when a data value from the incoming data stream exceeds a given value (in a direction you specify):



An extension of this is the Drift Analytic, which will automatically compute the threshold value from the initial set of readings from the data source rather this being required to be manually entered. It then looks for deviations from this value. The assumption is that in the initial conditions, the data source is behaving normally. This can be used to ensure that data source values don't slowly drift from initial conditions:



The Spike Analytic will continually use the moving average and continuous volatility calculations to define an acceptable range for a data source value that will be dynamically and continuously computed. Only when the value exceeds one of these thresholds will an alert be generated. This Analytics can usefully be used to automatically define a "nominal" operating range for a data source without prior knowledge of what that range is and where the value itself can correctly slowly move around a wide value range.

Technical Notes

- Timestamp Values are in Apama standard format: Float value of **seconds** since 1970 [Java standard is milliseconds since 1970]
- Analytic creation event is intentionally made uniform to allow simple creation of Analytic networks through scripts or a graphical editor

Included Sample

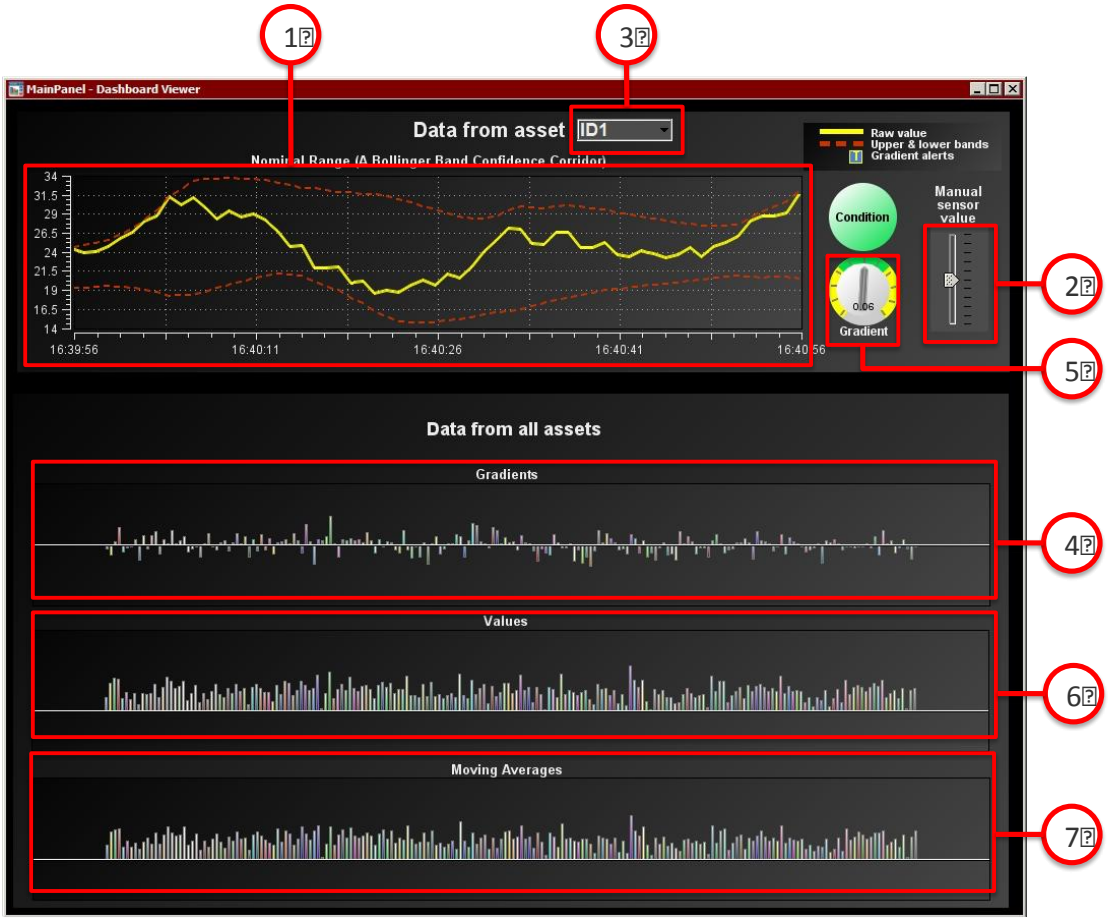
The sample within the Industry Analytics Kit gives a simple end-to-end demonstration of Analytics in use and includes a simple simulator to generate data feeds and an Apama Dashboard to visualize the results.

The Sample:

- Includes a Spike Analytic to identify when each individual data sources value is within "normal" operating bounds **(1)**. Also allows the value of one data source to be manually altered **(2)** to see the impact of large changes using the slide on the upper right hand side. By moving this you can modify the simulated value for that data source manually (e.g. to simulate a dramatic change)
    - Use the drop-down box **(3)** at the top of the user interface to browse through the details (ie: the data calculated by the Spike Analytic) for each individual data source.
- Uses the Gradient **(4)** Analytic to calculate the gradient of recent values for each data source, so we can tell how much a data sources value is changing.

- The gradient of the selected Data sourceID is also shown on a dial **(5)**.

- Exposes a chart of the current values of all data sources **(6)**
- Calculates the moving average (using the Average Analytic) of each data source in the chart at the bottom **(7)**.



Migration from the IoT Analytics Kit

The Industry Analytics Kit was originally termed the "IoT Analytics Kit", and while the analytics themselves still provide the same (and extended) use cases for the Internet of Things (IoT), they are also used in other industries. So, we have taken the opportunity to genericise some of the keywords used in the names of the components and the events used to define them.

There are no behavioural changes, just name changes. Refactoring or search and replace would be the simplest. Replacing the IoT Analytics Kit bundle with the new Industry Analytics Kit bundle will highlight any errors within Software AG Designer tool.

Below is the list of changes between the IoT Analytics Kit and the current Industry Analytics Kit.

- `com.industry.iot.Measure` event name is now `com.industry.analytics.Data`
- Two parameter names of the `com.industry.analytics.Data` event have also been modified:

Old parameter name	New parameter name
`name`	`streamName`
`assetId`	`sourceId`

- `com.industry.iot.Transform` event name is now `com.industry.analytics.Analytic`
- Some of the names of the Analytics have now been changed:

Old Transformer name	New Analytics name
`SortedMerge`	`Sorter`
`Quasher`	`Suppressor`
`MovingAverage`	`Average`
`Variance`	`Volatility`
`ThresholdBreach`	`Threshold`
`BaselineThreshold`	`Drift`
`NominalRange`	`Spike`
`PMML`	`Prediction`
`Simulator`	`DataSimulator`

- Some of the parameters of the other Analytics that are provided may also have changed. Please refer to the Technical Documentation provided for further details.