



---

# Serfboard Instruments Guide and Reference Version 1

CML00024-01

---

Code Magus Limited (England reg. no. 4024745)  
Number 6, 69 Woodstock Road  
Oxford, OX2 6EY, United Kingdom  
[www.codemagus.com](http://www.codemagus.com)  
Copyright © 2014 by Code Magus Limited  
All rights reserved



August 16, 2016

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	<i>Serfboard</i> overview . . . . .	4
1.2	About this Manual . . . . .	4
<b>2</b>	<b>Instruments Overview</b>	<b>6</b>
2.1	Configuration files . . . . .	6
2.2	Input Metrics . . . . .	6
<b>3</b>	<b>Analog Dial</b>	<b>8</b>
3.1	Description . . . . .	8
3.2	Panel Options . . . . .	8
3.2.1	<i>AverageWindow</i> . . . . .	8
3.2.2	<i>Duration</i> . . . . .	9
3.2.3	<i>Layout</i> . . . . .	9
3.2.4	<i>Scaling</i> . . . . .	10
3.3	Dial Options . . . . .	10
3.3.1	<i>Colour</i> . . . . .	10
3.3.2	<i>Max</i> . . . . .	11
3.3.3	<i>ScaleFactor</i> . . . . .	11
3.3.4	<i>Type</i> . . . . .	11
3.3.5	<i>Title</i> . . . . .	12
3.3.6	<i>Units</i> . . . . .	12
3.4	<i>Example</i> . . . . .	12
<b>4</b>	<b>Rate and Response Analog Dial</b>	<b>14</b>
4.1	Description . . . . .	14
4.2	Panel Options . . . . .	14
4.3	Dial Options . . . . .	14
4.4	Example . . . . .	14
<b>5</b>	<b>Real Time Graph</b>	<b>16</b>
5.1	Description . . . . .	16
5.2	Graph Graduation . . . . .	16
5.3	Panel Options . . . . .	16
5.3.1	<i>AverageWindow</i> . . . . .	17
5.3.2	<i>Duration</i> . . . . .	17
5.3.3	<i>GraphType</i> . . . . .	18
5.3.4	<i>LegendColumns</i> . . . . .	18
5.3.5	<i>Max</i> . . . . .	19
5.3.6	<i>ScaleFactor</i> . . . . .	19
5.3.7	<i>Scaling</i> . . . . .	19
5.3.8	<i>Xticks</i> . . . . .	20
5.3.9	<i>YUnits</i> . . . . .	20
5.3.10	<i>Yticks</i> . . . . .	20
5.4	Graph Options . . . . .	20

5.4.1	<i>Colour</i>	21
5.5	Example	21
<b>6</b>	<b>Digital display</b>	<b>23</b>
6.1	Description	23
6.2	Panel Options	23
6.2.1	AlertBackground	23
6.2.2	<i>Duration</i>	24
6.3	Display Options	24
6.3.1	Alert	24
6.3.2	<i>Action</i>	25
6.3.3	<i>Background</i>	26
6.3.4	<i>DisplayValue</i>	27
6.3.5	<i>Title</i>	27
6.3.6	<i>Units</i>	28
6.4	<i>Expression Variables</i>	28
6.5	Examples	31
<b>7</b>	<b>Distribution of Values</b>	<b>34</b>
7.1	Description	34
7.2	Panel Options	34
7.2.1	LegendColumns	34
7.2.2	ShowYaxis	34
7.2.3	Ymax	35
7.3	Example	35
<b>8</b>	<b>System Status Warning Light</b>	<b>37</b>
8.1	Description	37
8.2	Options	37
8.2.1	TimeOut	37
8.3	Example	37
<b>9</b>	<b>Events log</b>	<b>39</b>
9.1	Description	39
9.2	Panel Options	39
9.2.1	EventGroup	39
9.2.2	Expire	39
9.2.3	MaxEvents	40
9.2.4	Truncate	40
9.3	Example	40
<b>10</b>	<b>Data Feed Console Link</b>	<b>42</b>
10.1	Description	42
10.2	Panel Options	42

---

10.3 Example . . . . . 42

# 1 Introduction

## 1.1 *Serfboard* overview

The *Serfboard* system enables a user to monitor, and possibly interact with, a running system or systems via a web based graphical dashboard. Dashboards of this nature that show a unified view of key performance indicators of different systems are extremely effective in the management of networked systems testing or systems stress testing. In stress testing, being interactive, the user can change various thresholds and key values in order to watch the effect of different loads on the system under stress.

The *Serfboard* system comprises the following parts:

- The core server (*serfboard*) which is a daemon process (executes in the background) that receives metric data, stores it and supplies it to the web application on request.
- The display instrument plug-in programs and web application which are responsible for rendering the final dashboard as viewed by the user. The web application continuously requests updates of data from the server.
- Metric feed probes and feed library. The metric feed probes are responsible for querying a specific system that supplies metrics (for example SNMP), extracting the relevant data, converting it to the *Serfboard* format and forwarding the metric data on the the server. There is also a C programming library interface available for development of user written metric feeder programs. See the following for more information about supplied probes:
  - *cmlxaixp*: AIX Performance Metric Probe [7]
  - *cmlxsolp*: Solaris Performance Metric Probe [11]
  - *cmlxsnmp*: SNMP Metric Probe [6]
  - *cmlxwinp*: Windows Performance Metric Probe [8]
  - *cmlxwasp*: Websphere Application Server Performance Metric Probe [9]

## 1.2 About this Manual

This manual catalogues and describes the display instruments, and their options, that may be used when constructing a web based graphical dash board within *Serfboard*. Even though the display instruments are specified within a panel configuration, they are documented separately in this manual as each instrument may have many options that may be different to other instruments.

The graphical dash board is made up of rectangular panels each of which is described within a panel configuration. A display instrument is a program that processes input metric data and generates the web graphics data required for rendering the information within a panel. Its name and options are specified within the panel configuration.

Metrics can be rendered by the display instruments on a choice of analog dials; digital displays; graphs; and bar charts or be used to turn warning lights on.

Web browser compatibility is also discussed.

For additional technical information and how to configure *Serfboard*, refer to Serfboard Configuration Guide and Reference Version 1 [2] and for a full description on the user interface to *Serfboard*, refer to Serfboard User Guide Version 1 [4]. This manual assumes that the *Serfboard* system has been correctly installed as per Serfboard Installation Guide and Reference Version 1 [3].

## 2 Instruments Overview

### 2.1 Configuration files

The display instrument that will render the graphic on the web page and its options are specified in the panel configuration file. Refer to *Serfboard Configuration Guide and Reference Version 1* [2] for information on the syntax for defining a panel; the options, which are “keyword=value” statements, are explained in this manual, as they are specific to a display instrument.

### 2.2 Input Metrics

Most of the display instruments accept metrics from an external data source. For example this can be an *orkhestra* instance or a metric feeder agent.

Each instrument has a finite maximum number of channels it can process input metrics through. Where more than the maximum number is required additional panels must be configured.

The instruments use one of a number of types of metric. When designing a dashboard and choosing the display instruments it is imperative to know which type of metric will need to be rendered. These are explained below.

- Value list

This is the standard type of metric and the following data fields are key in rendering information:

- `time`. This is a time stamp of when the metric was produced.
- Data value 1: `count` or `value`.

If this is a `count` then it is an ever incrementing value. The `time` and `count` delta of the current and previous metric are used to calculate a rate value. An example of this type of metric is one produced by a system that monitors transactions. The count is incremented by the number of transactions that have occurred since the last metric was produced.

If this is a `value` then it is an arbitrary value where no discernible pattern can be assumed between successive metrics and no rate can be calculated from it. An example of this type of metric is one produced by a system that reports the current memory usage of a computer. This value may fluctuate in an apparent random manner as the load on the machine changes.

- Data value 2: `value`.

This is a value associated with data value 1. For example it could be:

- \* the sum of all response times since the first metric was produced.  
If the first value is a `count`, the deltas of the current and previous metric can be used to calculate the response time.
- \* the sum of all message lengths since the first metric was produced.  
If the first value is a `count`, the deltas of the current and previous metric can be used to calculate the average message length.

– Data value 3: sum of the square of value 2.

It is the sum of the squares of all values 2 since the first metric was produced. Its main function is for calculating the standard deviation of the set of metrics.

- Labelled list

This metric contains a list of labels with values. The main use of this type of metric is in rendering a bar graph. Each label and value corresponds to a bar in the graph.

For full documentation of CML metrics, refer to `metric: Metric Library Reference` [1].

## 3 Analog Dial

### 3.1 Description

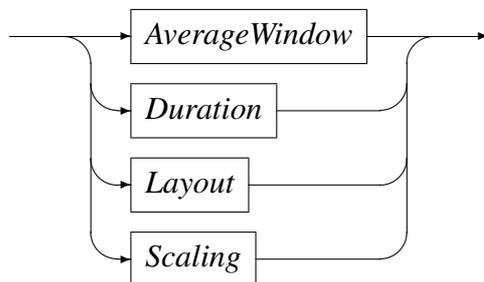
**Name:** Dials

This instrument displays a rate or response time value on up to four analog dials. Each dial in the panel can display either rate or response time. The current value is indicated by the needle on the dial. A digital windowed average value can be displayed inside the dial.

This instrument requires one metric for each dial as input. The metric must be a values list type and should contain count, sum of response times in seconds and the sum of the square of each response time.

### 3.2 Panel Options

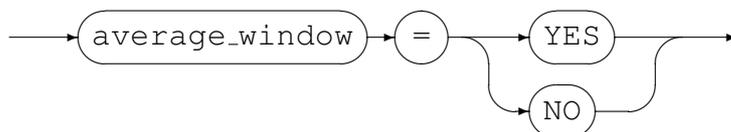
This instrument has the following options:



These options are described in the following sub sections.

#### 3.2.1 *Average Window*

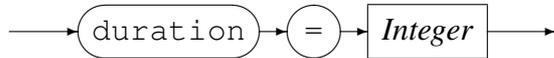
*AverageWindow*



The moving average value is displayed as a number in the centre of the dial if this option is set to YES. If it is set to NO or omitted then the moving average is not displayed. The window for calculating the moving average is configured in the metric configuration, refer to Serfboard Configuration Guide and Reference Version 1 [2] for documentation.

### 3.2.2 Duration

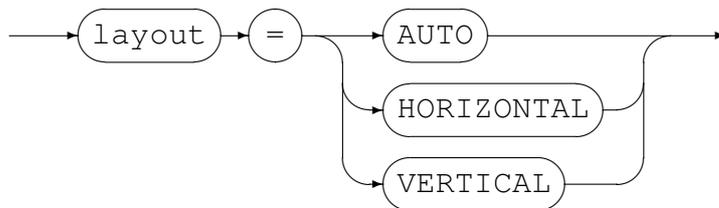
#### Duration



This option specifies, in seconds, how the dial reacts when no metrics have arrived within this time. The default is zero and means that the dial should always display the last value. If specified and *Integer* is greater than zero then the dial will revert to displaying a zero value if no further metrics are received after *Integer* seconds.

### 3.2.3 Layout

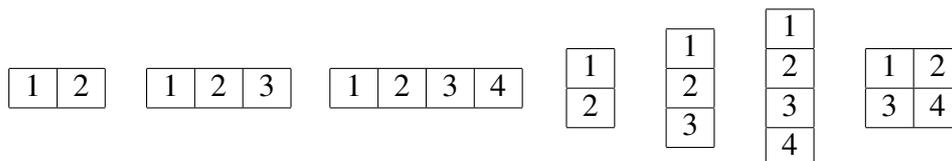
#### Layout



This option specifies how the dials are arranged within the panel. The default value is “AUTO”. The size of the dials is automatically determined by the width and height of the panel. Depending on what is specified, the dials are aligned as follows:

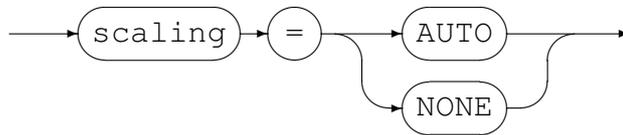
- “HORIZONTAL”. Forces horizontal alignment of the dials.
- “VERTICAL”. Forces vertical alignment of the dials.
- “AUTO”. The positions of the dials are automatically determined by the number of dials and the width and height of the panel. Depending on the number of dials specified in the configuration the dials are arranged as follows:
  - One dial. Centred in the panel.
  - Two or three dials. If the height of the panel is greater than its width the alignment is vertically, otherwise horizontally.
  - Four dials. The dials are arranged in a 2 by 2 square format.

The numbering of the dials in the various possible configurations are shown below:



### 3.2.4 *Scaling*

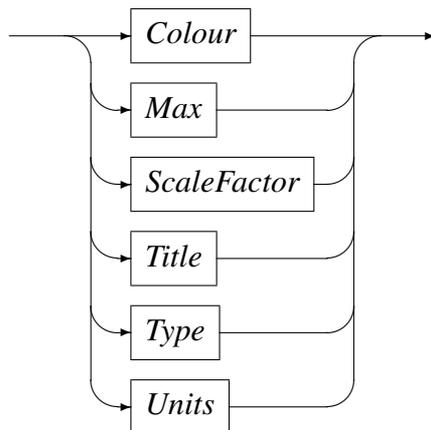
#### *Scaling*



This option either sets the dials maximum value or allows it to be re-calculated based on the current value being displayed. If `AUTO` is specified the maximum value will be re-calculated each time a new value for the dial is calculated. If `NONE` is specified the maximum will remain constant and any value above the maximum will only be displayed as the maximum. The default value is `AUTO`.

## 3.3 Dial Options

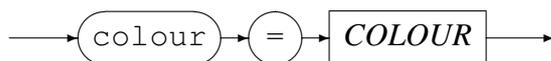
This instrument has the following options for each of the dials specified:



These options are described in the following sub sections.

### 3.3.1 *Colour*

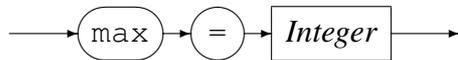
#### *Colour*



The colour of the dial legend may be set to a colour, other than the default, using a valid HTML colour name. These are usually one of the “W3C HTML and CSS standards” valid colour names [12].

### 3.3.2 Max

*Max*



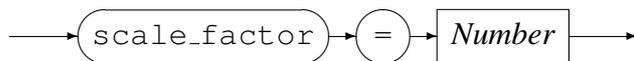
*Integer* specifies the maximum value that can be shown on the dial. If not specified, the maximum is set to one.

If *scaling* is set to *auto* then this is only the initial maximum value. If *scaling* is set to *none* then this is an absolute maximum and any values greater than this amount are shown as the maximum. This option has no effect on the calculations for the moving average.

The minimum value is always zero.

### 3.3.3 ScaleFactor

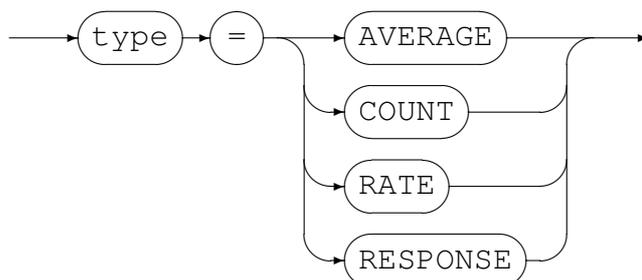
*ScaleFactor*



This option allows the values on each dial to be scaled by the given amount. To scale up *Number* must be greater than 1 and to scale down it must be less than 1.

### 3.3.4 Type

*Type*



This option specifies what is displayed in the dial. The values have the following meanings:

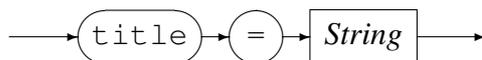
- AVERAGE displays calculated value of the average.
- COUNT displays count value direct from the metric.
- RATE displays calculated value of response time.

- RESPONSE displays calculated value of rate.

Refer to section 2.2 on page 6 for the definition of count, average, rate and response time.

### 3.3.5 Title

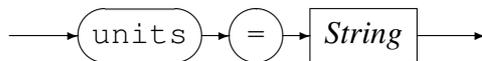
*Title*



This option overrides the title of the dial. By default the title of a dial is set to title of the metric, refer to Serfboard Configuration Guide and Reference Version 1 [2] for documentation.

### 3.3.6 Units

*Units*



String describes the numeric values on the dials and is displayed as a suffix to the number. The default values are “s” (seconds) for a response time dial and “/s” (per second) for a rate dial. For the numbering of the dials see section 3.2.3 on page 9.

## 3.4 Example

In this example four dials will each display a different metric. They will all display RATE and a moving average, and the maximum for each dial is set to 40. The moving average is calculated based on the the window size of 120 seconds as set in each of the metrics specified in the corresponding metric configuration.

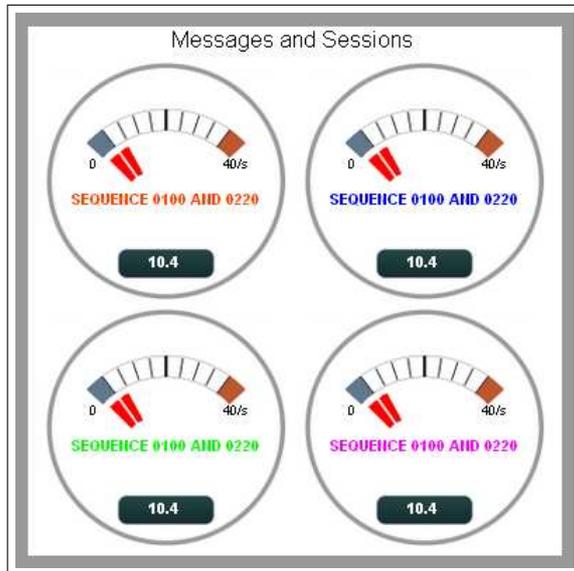
The entry, showing the options, in the panel configuration file would be as follows:

```
panel Pdials (width(400) height(400) origin(1180,580) control(Dials)
  delay(10000)
  title("Messages and Sessions")
  description("Messages and Sessions")
  options (scaling=NONE, average_window=YES)
  metric seq_100 (group(serfboard_demo02) options (type=RATE,max=40))
  metric seq_200 (group(serfboard_demo02) options (type=RATE,max=40))
  metric reversals (group(serfboard_demo02) options (type=RATE,max=40))
  metric Sessions (group(serfboard_demo02) options (type=RATE,max=40))
)
```

The metric configuration file entries for the metrics rendered in this panel are:

```
metric seq_100 (  
    average_window(120)  
    title("Sequence 0100 and 0220")  
    description("Message types 0100 and 0220"))  
metric seq_200 (  
    average_window(120)  
    title("Sequence 0200")  
    description("Message type 0200"))  
metric reversals (  
    average_window(120)  
    title("Transaction Reversals")  
    description("Transaction reversals"))  
metric good_responses (  
    title("Requests approved")  
    description("Requests approved"))
```

The generated panel image would look as follows:



## 4 Rate and Response Analog Dial

### 4.1 Description

**Name:** RateRespDials

This instrument process one metric and displays the values of the current rates and response times on two dials. The current value is indicated by the needle. A digital windowed average value can be displayed inside the dial.

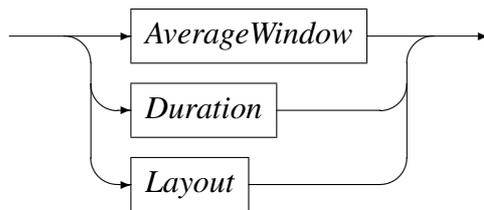
The dials are arranged either horizontally or vertically - see section 3.2.3 on page 9. The rate dial is always on the left or top and the response time dial is always on the right or bottom.

This instrument is a sub set of the Analog Dial described in the previous section - see section 3.1 on page 8. The combination of having a rate dial and a response dial from one metric is used very frequently, and this instrument simplifies the configuration there of.

This instrument requires only one metric as input. The metric must be a values list type, as described in section 2.2 on page 6.

### 4.2 Panel Options

This instrument has the following options:



Please refer to section 3.1 on page 8 for a description of these options.

### 4.3 Dial Options

This instrument has the no options for the dials:

### 4.4 Example

In this example the panel will have dials that automatically alter the maximum value depending on the values calculated from the current metric. A small digital display

inside the dial will display the average value for the last 60 seconds and the dial will drop to zero if no metrics are received within 300 seconds.

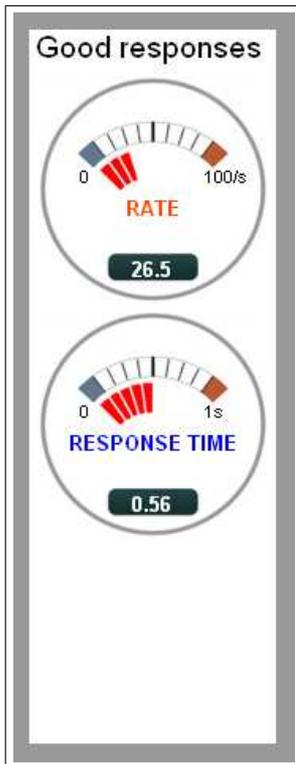
The entry, and options, in the panel configuration file would be as follows:

```
panel Pgood2 (width(150) height(400) origin(1430,170) control(RateRespDials)
  delay(10000)
  title("Good responses")
  description("Good responses")
  options(duration=300,average_window=YES)
  metric good_responses (group(serfboard_demo02))
)
```

The metric configuration file entry for this panel would be as follows:

```
metric good_responses (
  average_window(60)
  title("Requests approved")
  description("Requests approved"))
```

The generated panel image would look as follows:



## 5 Real Time Graph

### 5.1 Description

**Name:** Graph

This instrument displays values of current or average rates or response times as one or more plots on a graph. The maximum number of input metrics that it can display is eight.

This instrument requires one metric for each graph as input. Each one must be a values list type of metric as described in section [2.2](#) on page [6](#).

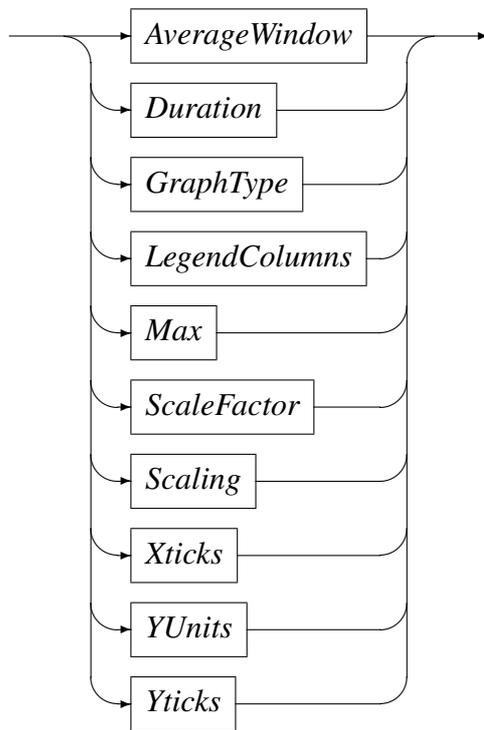
### 5.2 Graph Graduation

The X-axis shows the time scale and the Y-axis is graduated to show the value of the metric at a particular time.

Usually all the curves on the graph would be of a similar type and therefore the Y-axis, shown on the left hand side, uses a common graduation scale. If the data of the curves do not have a common graduation scale, an extra graduation scale for the Y-axis can be specified and is shown on the right hand side of the graph. If more graduation scales are required then the curves should be drawn on a separate graph.

### 5.3 Panel Options

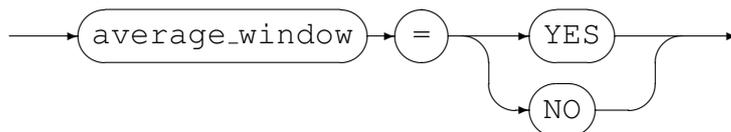
This instrument has the following options:



These options are described in the following sub sections.

### 5.3.1 *AverageWindow*

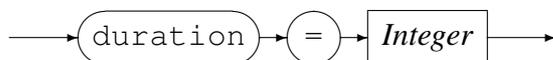
*AverageWindow*



When set to YES this causes a smoothed curve to be plotted on the graph as a dotted line in the same colour as the actual curve. If this option is not specified or set to NO the moving average curve is not drawn. The default is NO. The window for calculating the moving average is configured in the metric configuration, refer to [Serfboard Configuration Guide and Reference Version 1 \[2\]](#) for documentation.

### 5.3.2 *Duration*

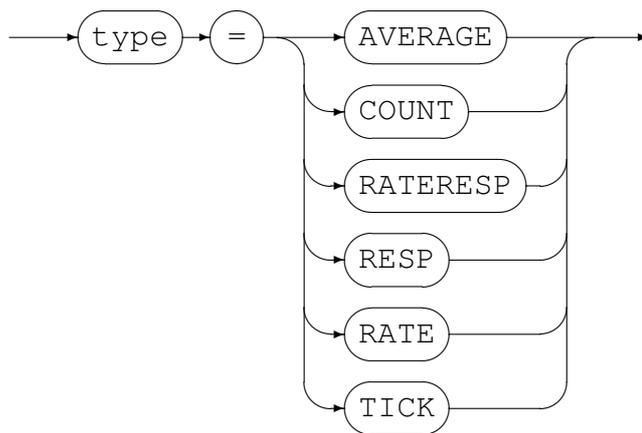
*Duration*



This option specifies the number of seconds over which the graph is plotted and is shown as the range on the X-axis. The default is 300 seconds (five minutes).

### 5.3.3 *GraphType*

*GraphType*



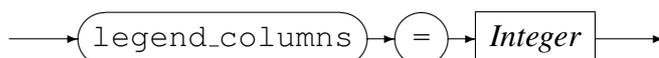
This option specifies what is plotted on the graph. The values have the the following meanings:

- AVERAGE plots calculated value of the average.
- COUNT plots count value direct from the metric.
- RATERESP plots calculated values of response time and rate.
- RESP plots calculated value of response time.
- RATE plots calculated value of rate.
- TICK plots ticks marks when the count value changes.

The maximum number of input metrics that a graph can display is eight, except for the RATERESP graph, which can only display one input metric.

### 5.3.4 *LegendColumns*

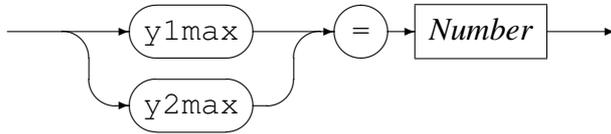
*LegendColumns*



The legends for the graphs are displayed below the graph. This option sets the number of columns in which the legends are arranged. The default value is one column.

### 5.3.5 Max

*Max*

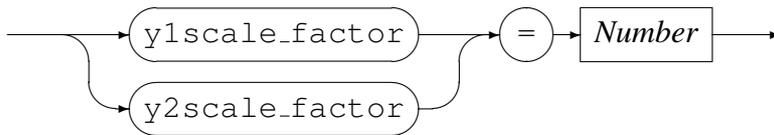


The two ranges specify the maximum values for the instrument. `y1max` is used for the left Y-axis and the `y2max` is used for the right Y-axis. If `scale` is set to `NONE` then this value is an absolute maximum and the graph may not show the current value, otherwise if it is set to `AUTO` this value is only the initial maximum value.

The Y-axis minimum is always zero.

### 5.3.6 ScaleFactor

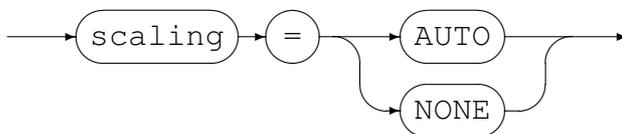
*ScaleFactor*



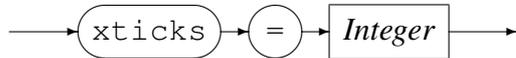
The calculated plot value is multiplied by this scale factor before being displayed. The default scale factor is 1.0. `y1scale_factor` is used for the left Y-axis and the `y2scale_factor` is used for the right Y-axis.

### 5.3.7 Scaling

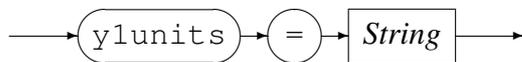
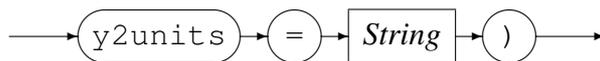
*Scaling*



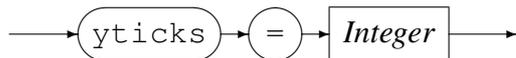
This option determines whether the graph should be scaled to fit the data or not. If `AUTO` is specified the maximum value on the Y-axis will be recalculated as required and will cause the entire graph to be redrawn in the new scale. If `NONE` is specified the maximum will remain constant and point values that exceed this value will only be shown as the maximum. The default value is `AUTO`.

**5.3.8 Xticks***Xticks*

This is the number of ticks or sub-divisions on the X-axis. The default value is three, meaning that values are displayed on the left, right and in the middle of the X-axis.

**5.3.9 YUnits***Y1Units**Y2Units*

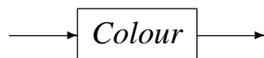
String describes the numeric values on the Y-axis and is displayed as a suffix to the number. The default values are “ms” for a response time plot and “/s” for a rate plot. `y1units` is used for the left Y-axis and the `y2units` is used for the right Y-axis.

**5.3.10 Yticks***Yticks*

This is the number of ticks or sub-divisions on the Y-axis. The default value is three, i.e. values are displayed on the top, middle and bottom of the Y-axis.

**5.4 Graph Options**

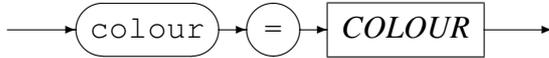
This instrument has the following options for the graphs requested:



These options are described in the following sub sections.

### 5.4.1 Colour

*Colour*



The colour of the graph plot lines may be set to a colour other than the default using a valid HTML colour name. These are usually one of the “W3C HTML and CSS standards” valid colour names [12].

## 5.5 Example

In this example, a graph will be plotted with rates and responses for the metric(s). Data for the last five minutes will be displayed. The moving average will be calculated over 60 seconds and displayed.

The panel configuration file contains the following entry:

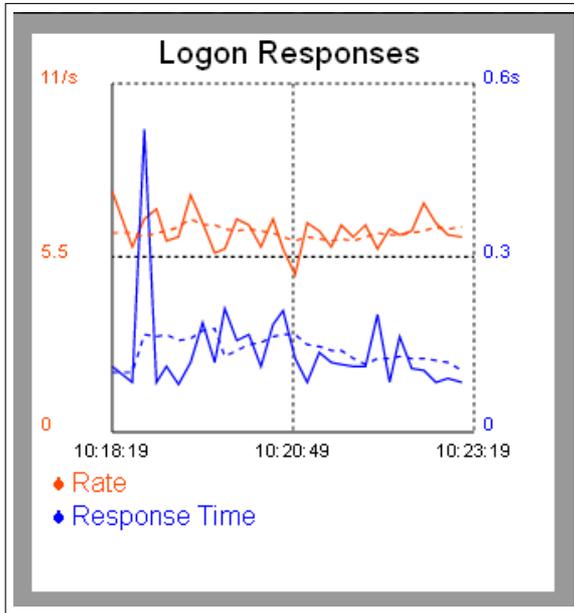
```

panel Plogonok (width(280) height(300) origin(0,480) control(Graph)
  delay(16000)
  title("Logon Responses")
  description("Logon rate and response")
  options(type=RATERESP,scaling=AUTO,duration=300,average_window=YES)
  metric logon_OKAY (group(serfboard_demo01))
)
  
```

The metric configuration file entry for this panel would be as follows:

```

metric logon_OKAY (
  title("Logon Responses")
  description("POS device logons successful"))
  
```



## 6 Digital display

### 6.1 Description

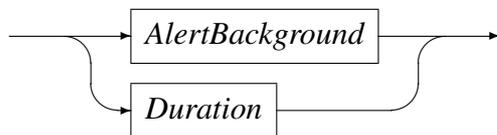
**Name:** display

This instrument displays a value on up to eight digital display counters. Each counter displays the name of the metric and the calculated value. This value is calculated by the expressions given via the display option `display_value`; see section 6.3.4 on page 27. Optionally the background colour can be set by expressions given via the display option `background`; see section 6.3.3 on page 26.

This instrument requires one metric for each display as input. The metric must be a values list type; see section 2.2 on page 6.

Each display can optionally act as a alert light: if the value of the display changes the background colour is changed. The warning light may be reset by clicking on it. See section 6.3.1 on page 24.

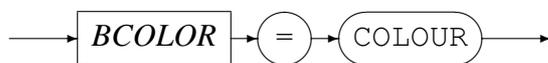
### 6.2 Panel Options



These options are described in the following sub sections.

#### 6.2.1 AlertBackground

*AlertBackground*



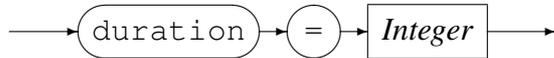
This option changes the background colour of the warning light and must be specified as a valid HTML colour. These are usually one of the “W3C HTML and CSS standards” valid colour names [12].

The background colour is used when a display is configured to be a warning light, see section 6.3.1 on page 24.

The default value for the background colour is `red`.

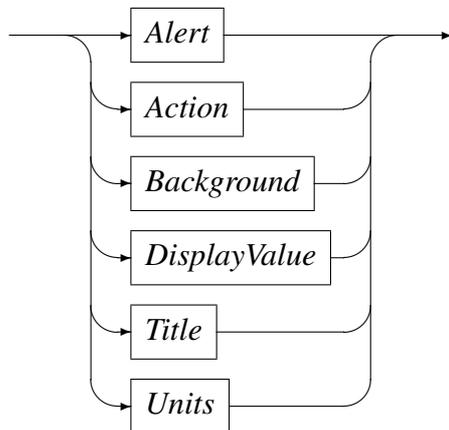
### 6.2.2 Duration

*Duration*



This option specifies, in seconds, how the display reacts when no metrics have arrived within this time. The default is zero and means that the dial should always display the last value. If specified and *Integer* is greater than zero then the display will revert to displaying a zero value if no further metrics are received after *Integer* seconds.

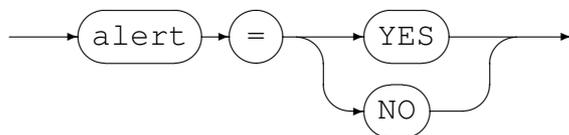
## 6.3 Display Options



These options are described in the following sub sections.

### 6.3.1 Alert

*Alert*

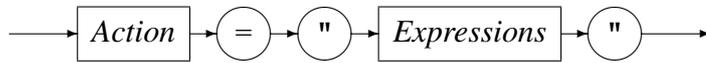


When alert is set to YES, the background colour of a display will change when its value changes. The warning light may be reset by clicking on it.

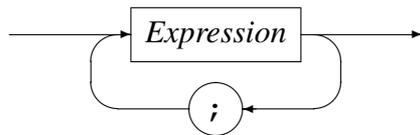
The default background colour of the warning light is red, it can be over written by using the panel option `alert_background`, see section [6.2.1](#) on page [23](#).

### 6.3.2 Action

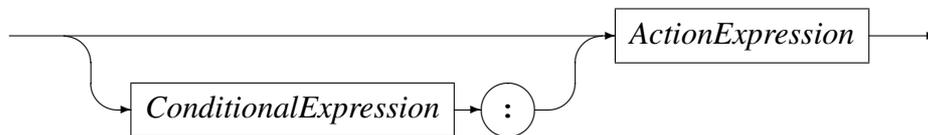
#### *DisplayValue*



#### *Expressions*



#### *Expression*



This option specifies actions to be taken. Currently only one action is available in the form of a function to execute a system command:

#### *SystemCommand*



#### *Command*



#### *FuncningParameter*



Please refer to `expeval: Expression Evaluation API Reference [10]` for documentation on expressions and to section 6.4 on page 28 for the variables that can be used in the expressions.

The action taken is the action expression associated with the first condition that evaluates to true,

In this example the script `'cspalert.sh Host response exceeds 20 seconds'` are executed if the Host was active, the response exceeds 20 seconds and the script was not executed in the last half hour (`lastcall>1800`).

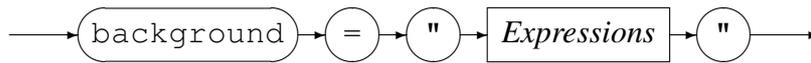
#### **Example**

```
action="(prod_posipcon_00.hstactv.delta_sum>0) and"
        "(prod_posipcon_00.hstactv.delta_count>0) and"
        "((prod_posipcon_00.hstactv.delta_sum/delta_count) >= 20000) :"
```

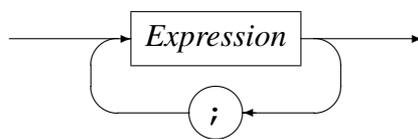
```
"(lastcall>1800) and (System('cspalert.sh "
"Host response exceeds 20 seconds',0))"
```

### 6.3.3 Background

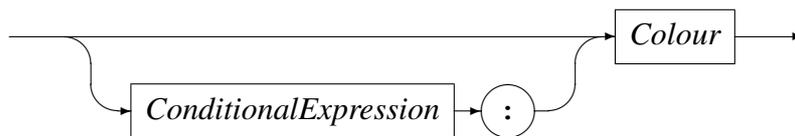
#### DisplayValue



#### Expressions



#### Expression



#### Colour



This option specifies how the background colour of the displayed value is set. The colour specified must be a valid HTML colour name. These are usually one of the “W3C HTML and CSS standards” valid colour names [12].

Please refer to `expeval: Expression Evaluation API Reference` [10] for documentation on expressions and to section 6.4 on page 28 for the variables that can be used in the expressions.

The colour is set to the colour specified on the first condition that evaluates to true, if no match, the background will be set to the default background colour, which is `white`.

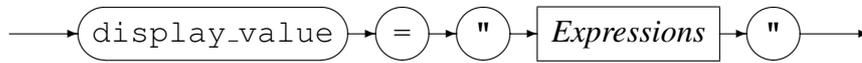
#### Example

This example set the background to `red` when `current_value` is zero, to `green` when equals to one and to `pink` for any other value.

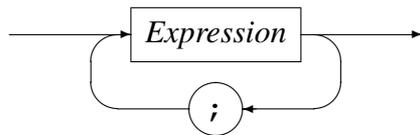
```
background = "prod_posipcon_00.hostcon.current_value = 0 : red;"
              "prod_posipcon_00.hostcon.current_value = 1 : green"
```

### 6.3.4 *DisplayValue*

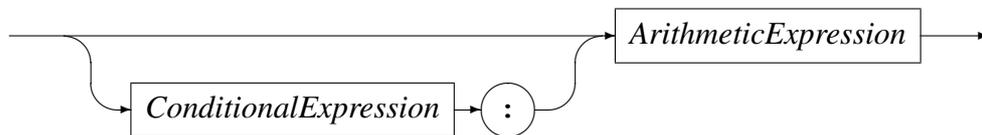
*DisplayValue*



*Expressions*



*Expression*



This option specifies how the display value of the counter is calculated.

Please refer to `expeval: Expression Evaluation API Reference` [10] for documentation on expressions and to section 6.4 on page 28 for the variables that can be used in the expressions.

The calculation for the display value is the arithmetic expression associated with the first condition that evaluates to true, if no match, the value will set to zero.

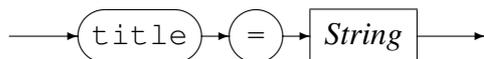
This value to be displayed in this example is rate. If `delta_seconds` and `delta_count` are both greater than zero the rate is calculated, else zero is displayed.

#### Example

```
display_value="(prod_posipcon_00.hstactv.delta_seconds>0) and"
              "(prod_posipcon_00.hstactv.delta_count>0) : "
              "prod_posipcon_00.hstactv.delta_count/"
              "prod_posipcon_00.hstactv.delta_seconds;"
              "0"
```

### 6.3.5 *Title*

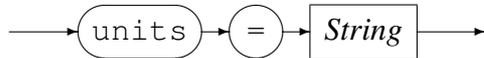
*Title*



This option specifies the title of a display. If omitted the metric title from the metric configuration will be used.

### 6.3.6 Units

*Units*

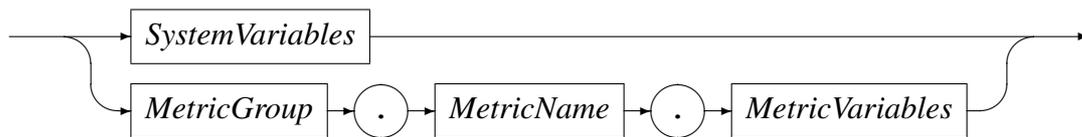


*String* describes the numeric values on the display and is displayed as a suffix to the number. The default values are “s” for a response time display, “/s” for a rate display and none for a count display.

## 6.4 Expression Variables

This section describe the variables that can be used in expressions.

*Variable*



*MetricGroup*



This is the group that the metric belongs to.

*MetricName*



This is the name of the metric.

*SystemVariables*

- `lastcall`  
The time in seconds when the last system call was made for an action, see section [6.3.2](#) on page 25. This variable is associated with the action.
- `uptime`  
The time in seconds elapsed since the start up of the `serfboard` instance.

*MetricVariables*

- `current_count`  
Data value 1 of the current metric.
- `current_sum`  
Data value 2 of the current metric.

- `current_sum_sq`  
Data value 3 of the current metric.
- `previous_count`  
Data value 1 of the previous metric.
- `previous_sum`  
Data value 2 of the previous metric.
- `previous_sum_sq`  
Data value 3 of the previous metric.
- `delta_seconds`  
Time difference in seconds between the current and previous metric.
- `delta_count`  
Delta of Data value 1 of the current and previous metric.
- `delta_sum`  
Delta of Data value 2 of the current and previous metric.
- `delta_sum_sq`  
Delta of Data value 3 of the current and previous metric.
- `last_5minutes_seconds`  
Time in seconds for the duration of the the last five minutes interval.
- `last_5minutes_count`  
Data value 1 of the last five minutes.
- `last_5minutes_sum`  
Data value 2 of the last five minutes.
- `last_5minutes_sum_sq`  
Data value 3 of the last five minutes.
- `last_hour_seconds`  
Time in seconds for the duration of the the last hour interval.
- `last_hour_count`  
Data value 1 of the last hour.
- `last_hour_sum`  
Data value 2 of the last hour.
- `last_hour_sum_sq`  
Data value 3 of the last hour.
- `start_of_day_count`  
Data value 1 as at the start of the day.

- `start_of_day_sum`  
Data value 2 as at the start of the day.
- `start_of_day_sum_sq`  
Data value 3 as at the start of the day.
- `this_day_count`  
Data value 1 of the day.
- `this_day_sum`  
Data value 2 of the day.
- `this_day_sum_sq`  
Data value 3 of the day.
- `previous_day_count`  
Data value 1 of the previous day.
- `previous_day_sum`  
Data value 2 of the previous day.
- `previous_day_sum_sq`  
Data value 3 of the previous day.
- `start_of_hour_count`  
Data value 1 as at the start of the hour.
- `start_of_hour_sum`  
Data value 2 as at the start of the hour.
- `start_of_hour_sum_sq`  
Data value 3 as at the start of the hour.
- `this_hour_count`  
Data value 1 of the hour.
- `this_hour_sum`  
Data value 2 of the hour.
- `this_hour_sum_sq`  
Data value 3 of the hour.
- `previous_hour_count`  
Data value 1 of the previous hour.
- `previous_hour_sum`  
Data value 2 of the previous hour.
- `previous_hour_sum_sq`  
Data value 3 of the previous hour.

- `start_of_minute_count`  
Data value 1 as at the start of the minute.
- `start_of_minute_sum`  
Data value 2 as at the start of the minute.
- `start_of_minute_sum_sq`  
Data value 3 as at the start of the minute.
- `this_minute_count`  
Data value 1 of the minute.
- `this_minute_sum`  
Data value 2 of the minute.
- `this_minute_sum_sq`  
Data value 3 of the minute.
- `previous_minute_count`  
Data value 1 of the previous minute.
- `previous_minute_sum`  
Data value 2 of the previous minute.
- `previous_minute_sum_sq`  
Data value 3 of the previous minute.

## 6.5 Examples

### Example 1

This example implements alert lights for all the metrics in the panel. The background colour of the counters changes to red when their value changes.

The panel configuration file contains the following entry:

```
panel Perror (width(420) height(100) origin(430,480) control(display)
  delay(10000)
  title("Errors")
  description("The indicators in this panel lights up when an error occurs.")
  metric Timeouts (group(serfboard_demo01)
    options(display_value="current_count", alert=YES))
  metric Error_disconnect (group(serfboard_demo01)
    options(display_value="current_count", alert=YES))
  metric hsm_errors (group(serfboard_demo01)
    options(display_value="current_count", alert=YES))
  metric DB_auth_DENY_INV_PIN (group(serfboard_demo01)
    options(display_value="current_count", alert=YES))
  metric logon_BAD (group(serfboard_demo01)
    options(display_value="current_count", alert=YES))
  metric logon_DENY (group(serfboard_demo01)
```

```

    options(display_value="current_count", alert=YES))
)

```

The metric configuration file entries for this panel would be as follows:

```

metric Timeouts (
    title("Timeouts")
    description("Timeout waiting for a response to an request sent.))
metric Error_disconnect (
    title("Disconnects")
    description("Unexpected disconnects"))
metric hsm_errors (
    title("HSM Errors")
    description("HSM Errors - serious problem"))
metric DB_auth_DENY_INV_PIN (
    title("Pin Errors")
    description("Debit card authorisations denied - Invalid PIN"))
metric logon_BAD (
    title("Bad Logon")
    description("Short logon response received. Very serious error, that needs to be f
metric logon_DENY (
    title("Logon Deny")
    description("Logon denied. Very serious error, that needs to be fixed, before cont

```

The generated panel image would look as follows:

Errors					
Timeouts	Disconnects	HSM Errors	Pin Errors	Bad Logon	Logon Deny
0	4	0	0	0	0

### Example 2

This example will display the values of the counters.

The entry in the panel configuration file contains the following:

```

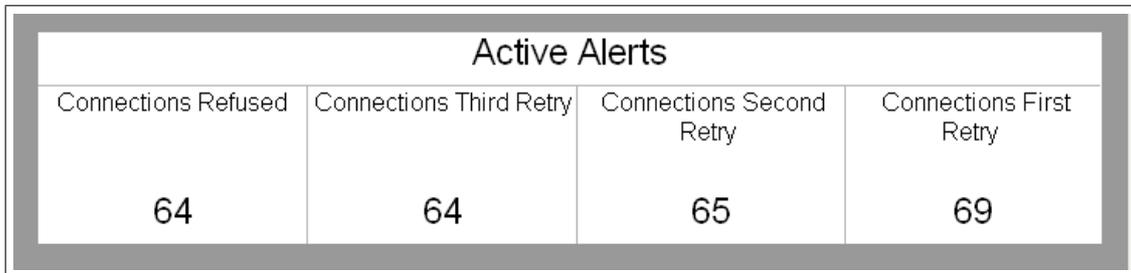
panel Pwarn (width(420) height(100) origin(860,480) control(display)
    delay(10000)
    title("Active Alerts")
    description("Active alerts and warnings")
    metric Connection_Refused (group(serfboard_demo01)
        options(display_value="current_count"))
    metric Connection_Retry_3 (group(serfboard_demo01)
        options(display_value="current_count"))
    metric Connection_Retry_2 (group(serfboard_demo01)
        options(display_value="current_count"))
    metric Connection_Retry_1 (group(serfboard_demo01)
        options(display_value="current_count"))
)

```

The metric configuration file entries for this panel would be as follows:

```
metric Connection_Retry_1 (  
    title("Connections First Retry")  
    description("Connection retry - first attempt"))  
metric Connection_Retry_2 (  
    title("Connections Second Retry ")  
    description("Connection retry - second attempt"))  
metric Connection_Retry_3 (  
    title("Connections Third Retry ")  
    description("Connection retry - third attempt"))  
metric Connection_Refused (  
    title("Connections Refused")  
    description("Unable to establish an Network connection"))
```

The generated panel image would look as follows:



Active Alerts			
Connections Refused	Connections Third Retry	Connections Second Retry	Connections First Retry
64	64	65	69

## 7 Distribution of Values

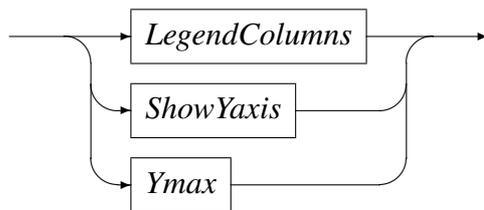
### 7.1 Description

**Name:** BarChart

This instrument displays distribution data plotted as a bar chart. The input must be of the metric type labelled list.

This instrument uses only one input metric. Each bar in the graph is associated with one of the labelled list values in the metric.

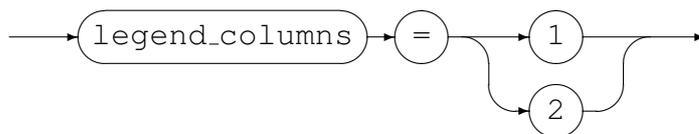
### 7.2 Panel Options



These options are described in the following sub sections.

#### 7.2.1 LegendColumns

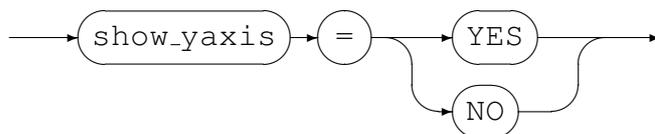
*LegendColumns*



The legends for the bars are displayed below the graph. This option sets the number of columns in which the legends are arranged. The default value is one column.

#### 7.2.2 ShowYaxis

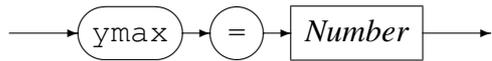
*ShowYaxis*



This controls the display of Y-axis values on the left of the graph. The default value is “no.” The value of this option is not case sensitive.

### 7.2.3 Ymax

*Ymax*



This specifies the maximum value for the bar chart on the Y-axis, the default is 100.

## 7.3 Example

In this example, a bar chart will be plotted with Y-axis values and the legends will be displayed in two columns. The panel configuration file contains the following entry:

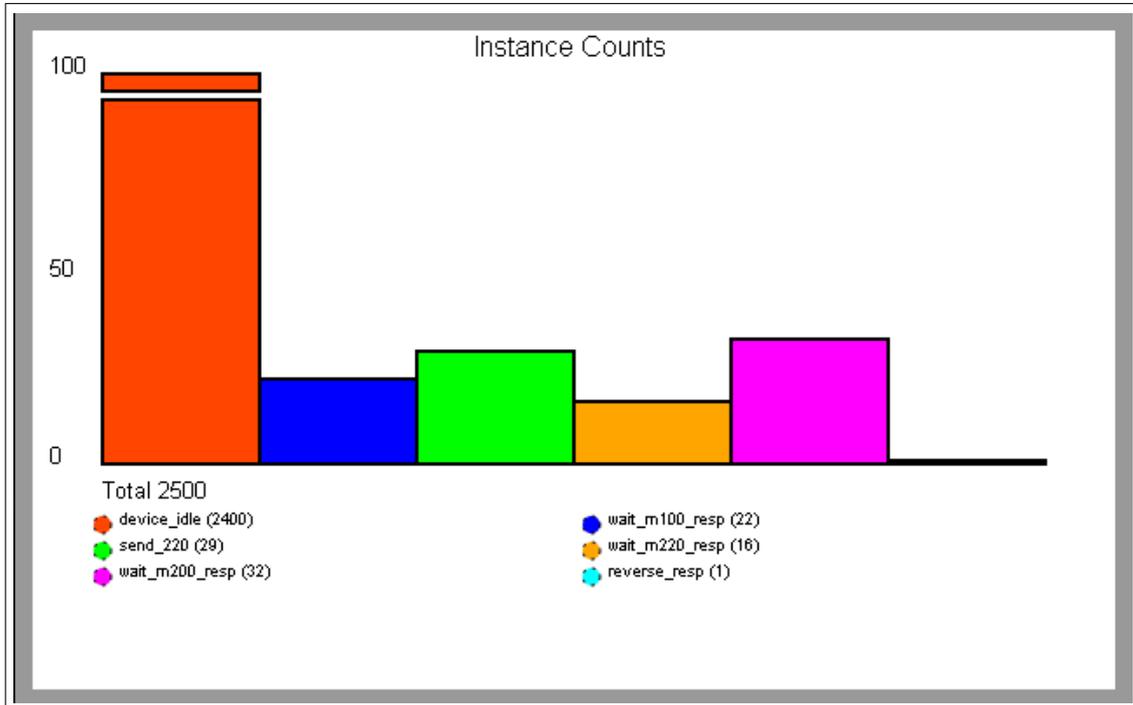
```

panel Pinst (width(640) height(400) origin(530,580) control(BarChart)
  delay(10000)
  title("Instance Counts")
  description("Instances count in state machine states.")
  options(show_yaxis=YES,legend_columns=2)
  metric State_instances (group(serfboard_demo02))
)
  
```

The metric configuration file entry for this panel would be as follows:

```

metric State_instances (
  title("Instances of states")
  description("Snap shot of all the states in the state machine with "
    "instances waiting to transition out of the state.")
)
  
```



## 8 System Status Warning Light

### 8.1 Description

**Name:** systemStatus

This instrument displays the status of the *Serfboard* instance and sets the background to reflect the current status. By clicking on the panel a separate browser window is opened describing the status.

The following conditions are defined, and the colour of the warning light is set to:

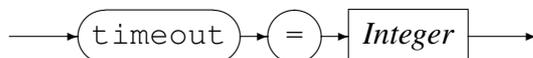
- Green.  
All the *Serfboard* metric feeders are connected and metrics was received from all the feeders within the `timeout` period.
- Orange.  
For some of the *Serfboard* metric feeders no metrics was received within the `timeout` period.
- Red.  
Either the WEB browser have no contact to the *Serfboard* instance or no metrics was received from any of the metric feeders within the `timeout` period.

### 8.2 Options



#### 8.2.1 TimeOut

*TimeOut*



This options sets the number of seconds to wait before the warning light may be set. The default value is 60 seconds.

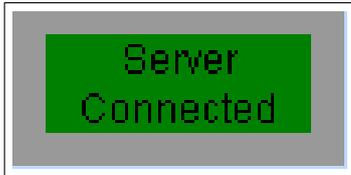
### 8.3 Example

In this example the system will display a warning with a `timeout` period of one minute.

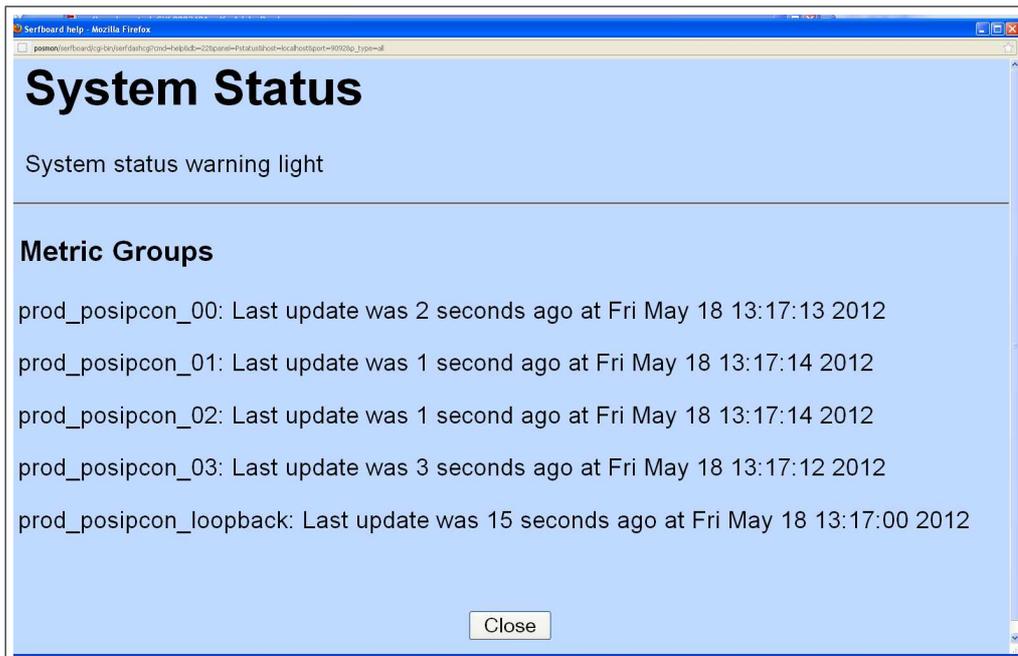
The entry in the panel configuration file contains the following information:

```
panel Pstatus (width(100) height(50) origin(1400,110) control(systemStatus)
  delay(15000)
  title("System Status")
  description("System Status warning light")
  options(timeout=60)
)
```

The generated panel image would look as follows when everything is OK.



By clicking on the panel the status window opened would look as follows:



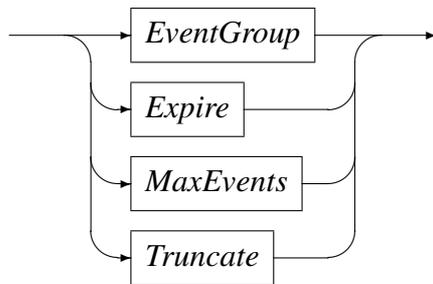
## 9 Events log

### 9.1 Description

**Name:** eventLog

This instrument displays event messages sent to and/or generated by the server. A warning light is turned on when a new event occurs. To see the events click inside the instrument to display them in a separate browser window; This also resets the warning light if it was set.

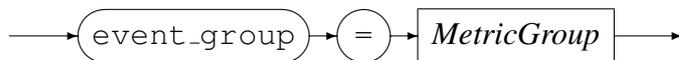
### 9.2 Panel Options



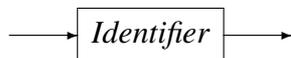
These options are described in the following sub sections.

#### 9.2.1 EventGroup

*EventGroup*



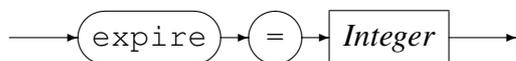
*MetricGroup*



Display events for this metric group, if this option is omitted, all events will be displayed.

#### 9.2.2 Expire

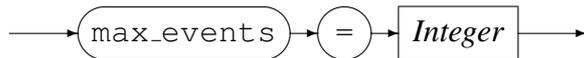
*Expire*



The period of time, in seconds, for which events are stored. The default is zero and means that all events are stored.

### 9.2.3 MaxEvents

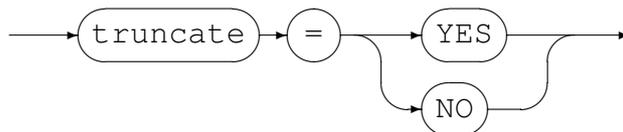
*MaxEvents*



The maximum number of events displayed at a time.

### 9.2.4 Truncate

*Truncate*



This option may be used to shorten the event message. If set to YES, only the first line will be displayed.

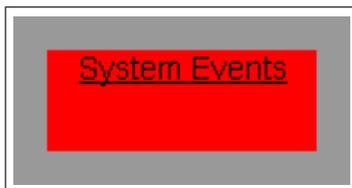
## 9.3 Example

This example implements an event light panel. Under normal circumstances the display is blank. The background colour changes to red when a new event occurs.

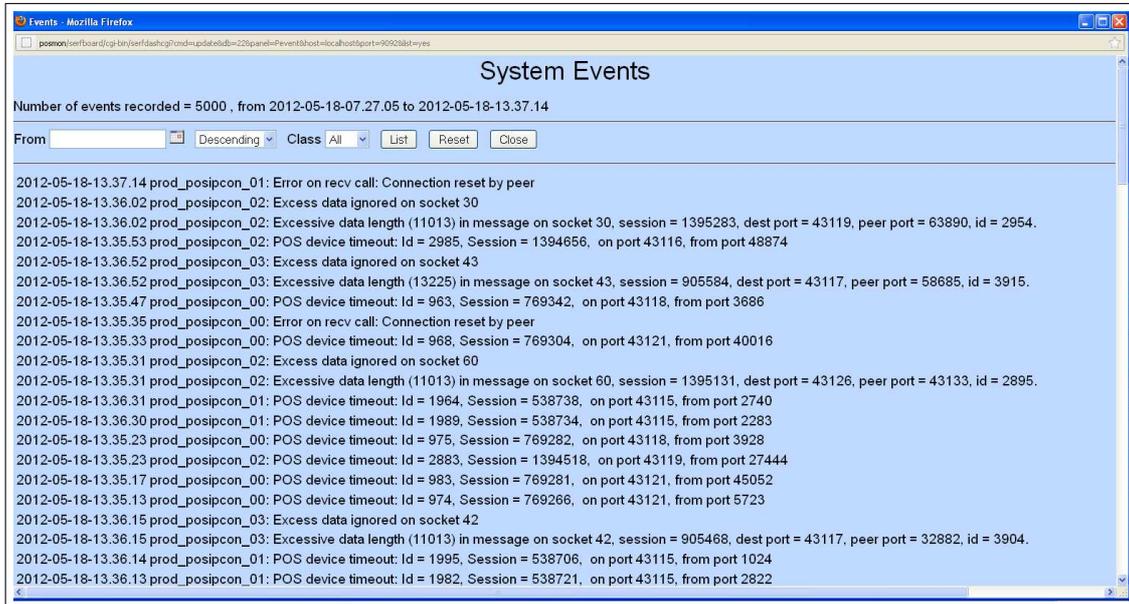
The entry in the panel configuration file contains the following:

```
panel Pevent (width(100) height(50) origin(1290,110) control(eventLog)
  delay(15000)
  title("System Events")
  description("System events log")
  options(max_events=100,expire=0,truncate=YES)
)
```

The generated panel image would look as follows when there are events to view:



Clicking inside the event panel will display the event list in a separate window and would look as follows:



## 10 Data Feed Console Link

**Name:** ConsoleLink

### 10.1 Description

This instrument displays an icon with a hyperlink that opens a window for a console to *serfboard* and data feeders. The new window takes the form of an edit box allowing the user to enter a command in order to enquire about or change the configuration of Serfboard or the data feeders. An example of a data feed is *orkhestra*. Refer to *orkhestra: Configuration and User Reference Version 1* [5] for information on the command syntax and response data.

This panel does not read input metrics.

### 10.2 Panel Options

This instrument does not have any options.

### 10.3 Example

This example implements a console link on a dashboard.

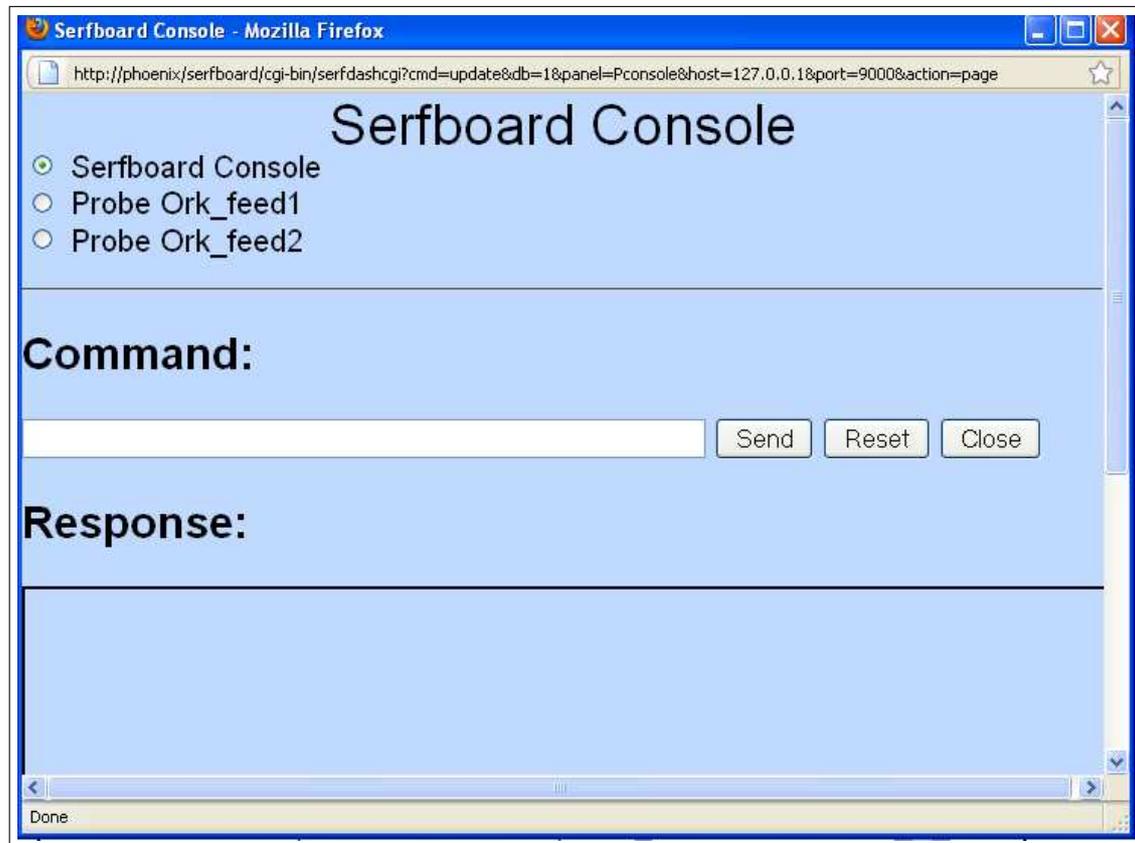
The entry in the dashboard configuration file contains the following:

```
panel Pconsole (width(70) height(50) origin(1510,10) control(ConsoleLink)
    title("Serfboard console")
    description("Feeder console for entering commands"))
```

The generated panel image would look as follows:



The console that opens would look as follows:



The check buttons are for selecting to which command interface to interface with. In this example the top button titled ('SerfBoard Console') is for interfacing to *Serfboard*. The next two buttons are to the command interfaces of two metric feed probes sending metrics to this particular *Serfboard*. The serfboard commands that defined this two command interfaces are as follows:

```
serfboard>open interface probe Ork_feed1(host (codemagus.it.nednet.co.za)
port (60050) )
Opened probe interface
serfboard>open interface probe Ork_feed2(host (codemagus.it.nednet.co.za)
port (60052) )
Opened probe interface
```

## References

- [1] metric: Metric Library Reference. CML Document CML00006-01, Code Magus Limited, September 2010. [PDF](#).
- [2] Serfboard Configuration Guide and Reference Version 1. CML Document CML00023-01, Code Magus Limited, July 2008. [PDF](#).
- [3] Serfboard Installation Guide and Reference Version 1. CML Document CML00025-01, Code Magus Limited, July 2008. [PDF](#).
- [4] Serfboard User Guide Version 1. CML Document CML00027-01, Code Magus Limited, July 2008. [PDF](#).
- [5] orkhestra: Configuration and User Reference Version 1. CML Document CML00041-01, Code Magus Limited, June 2011. [PDF](#).
- [6] cmlxsnmp: SNMP Metric Probe. CML Document CML00044-01, Code Magus Limited, June 2009. [PDF](#).
- [7] cmlxaixp: AIX Performance Metric Probe. CML Document CML00045-01, Code Magus Limited, June 2009. [PDF](#).
- [8] cmlxwinp: Windows Performance Metric Probe. CML Document CML00048-01, Code Magus Limited, June 2009. [PDF](#).
- [9] cmlxwasp: Websphere Application Server Performance Metric Probe. CML Document CML00049-01, Code Magus Limited, June 2009. [PDF](#).
- [10] expeval: Expression Evaluation API Reference. CML Document CML00052-01, Code Magus Limited, November 2009. [PDF](#).
- [11] cmlxsolp: Solaris Performance Metric Probe. CML Document CML00065-01, Code Magus Limited, June 2009. [PDF](#).
- [12] W3C. Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS). Recommendation, World Wide Web Consortium, 2006.  
<http://www.w3.org>.