Supervisor - Thresholds for Table Columns and Wallboard Tiles

FROM RELEASE 3.34

It is possible to setup thresholds for values in table columns in the supervisor.

The mechanism to perform this involves a combination of setting the threshold values - i.e. setting up values to go into "buckets", and then configuring CSS styles to color these elements.

Difficulty Level

🛇

You should have html and CSS experience before attempting these settings.

Example

To make the following documentation easier to understand, we will take the **Availability** column as our example.

We want to colour this green when the availability is 80% or above, yellow when it is 60% or above and red otherwise.

Buckets

The first step is to define buckets into which values are placed. Each bucket will ultimately be used to define the CSS used (e.g. for foreground and background colors) with which the value will be rendered.

Tip

ⓘ

CSS experts: you can add whatever you want including animations and effects.

The number of buckets you configure is not restricted, however, to keep the CSS simple, we will consider three buckets in the following examples.

The CSS definition which will be used by each bucket in our example is shown in the table below. (We will see how to apply this later).

Bucket	Colouring	CSS Definition
0	Green	`background-color: #169A13; color: white;`
1	Yellow	`background-color: #FCC10F; color: black;`
2	Red	`background-color: #BB0F18; color: white;`

Bucket Definition

First of all, we will consider a three bucket definition using the buckets 0, 1 and 2 above (green, yellow, red - like a traffic light).

A bucket definition is a comma separated list of integers. The first value in the list is bucket 0, and so on.

When the system renders a value in a column, in order to decide what CSS rules to apply, it compares the value with the threshold all of the buckets defined using the following algorithm:

Bucket Algorithm

ⓘ

- A value can be put into a bucket if it is greater than or equal to the threshold value of the bucket.
- The value is placed in the bucket **with the highest threshold possible**.
- All of the defined buckets are processed from left to right - hence the value actually goes into the last bucket whose threshold is greater or equal to any previous threshold.

For our example, the bucket definition **80,60,0** can be used - defining three buckets with thresholds of 80, 60 and 0 respectively.

- If the value is >= 80 it will be placed in bucket 0. Whilst it could fit into all buckets, the bucket with the highest threshold is bucket 0, and this one will "win".
- If the value is >= 60 but < 80, it will be placed in bucket 1. If could fit into buckets 1 or 2, but bucket 1 has the highest threshold and will "win".
- If the value is < 60 it can only fit into the third bucket and so will be placed there.

Note: in our example if the value was negative, it would not fit into any bucket and no CSS rules would be applied.

ⓘ

Since a service level should never be less than 0, this will not matter.

However, the principle of not having a bucket which fits a value can be used safely - if a value is not placed in a bucket, no CSS rules will be applied and the value will appear as usual (black text on the grey zebra striped background).

Columns

The columns which are available to be highlighted are shown in the supervisor with a grey bar underneath them.

Here the Availability column is shown.

Availability
100.00 %
0.00 %
0.00 %
100.00 %

When you **mouse over** the grey bar in the column header, you are shown the relevant settings:

Setting Thresholds.AcdSupervisor.Statistics.ServiceLevelAvailability
Column Stats_ServiceLevelAvailability
ID AcdGroupsID

Setting

This is the name of the setting required - see the previous section on how to create the settings.

Column

This is the name of the column used in the CSS definition. You will use this further below.

ID

This is the associated ID for the CSS setting.

The purpose of the ID is to enable different coloring for different rows of data.

Each row of data is rendered with a particular ID, in our example, the AcdGroupsID.

If you want to have different coloring for different ACD Groups, you can use the ID value later in the CSS too.

Attaching Buckets to Columns

Buckets are attached to columns, by defining:

- A User Setting (these can be defined by each user individually in **User Master Data**)
- An ACD Group Parameter (only used for values associated with ACD Groups - in the **ACD Groups ... Parameters Tab** for each ACD Group individually).
- A Client Parameter setting (these can be defined for the client account as a whole in **Client Master Data**)
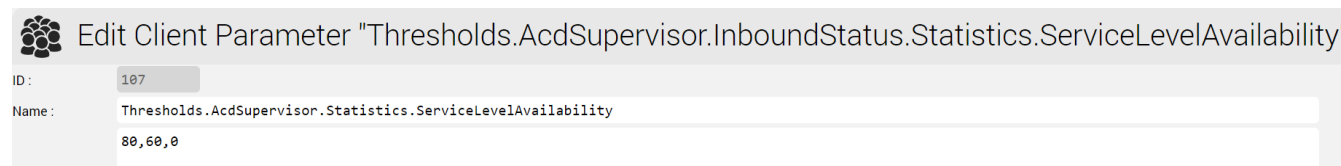
The order of processing is as follows:

- If the user has defined a setting, use this setting.
- **Otherwise:** If the value is associated with an ACD group, check the ACD group parameters and use the setting if found.
- **Otherwise:** Check the Client Parameters and use this setting if found.
- **Otherwise:** no setting is configured, no processing will occur.

The actual setting required is shown in the mouse-over help for the column - and is value to the right of **Setting,** as shown above.

For our example, we need to configure the setting **Thresholds.AcdSupervisor.Statistics.ServiceLevelAvailability**

Here it is shown in Client Master Data ... Parameters:

CSS and Rendering

All elements displayed in the table are child **div** elements of the corresponding **td** (table data) element in the DOM. The div element may contain other elements - such as a span.

For example:

```
<td class="iceDatTblCol1 fullWidthTableCol1" style=";text-align:right;">
        <div class="icePnlGrp Global_Bucket_1 Stats_ServiceLevelAvailability_Bucket_1 Stats_ServiceLevelAvailability_235_Bucket_1" id="M:
elementTodaysGroupData_dataTableTodaysGroupData:0:j_idt3875">
                <span class="iceOutTxt alignRight" id="M:elementTodaysGroupData_dataTableTodaysGroupData:0:j_idt3877">100.00 %</span>
        </div>
</td>
```

The div element rendered is with three special CSS classes to allow for coloring:

CSS Classes

Global CSS Class - Global_Bucket_X

In the example above: **Global_Bucket_1**

This is a global style available which can be used by any table column. X is be replaced by the bucket number in which the value is placed. In the example above, the bucket used is bucket 1.

Column Specific CSS Class - Column_Bucket_X

In the example above: **Stats_ServiceLevelAvailability_Bucket_1**

This is a column specific style applied to the column. The name for the start of the CSS class name is shown in the mouse-over help as "Column". After the name comes _Bucket_ followed by the number of the bucket. In our example, the column is Stats_ServiceLevelAvailability, and the bucket is 1.

Column and ID Specific CSS Class - Column_ID_Bucket_X

In the example above: Stats_ServiceLevelAvailability_235_Bucket_1

This is a column and ID specific style applied to a particular column, and an associated ID for a particular row of data. In our example, the CSS style is associated with the Column = Stats_ServiceLevelAvailability, for AcdGroupsID = 235, and the Bucket used is 1.

Defining the CSS

The last step is to define the CSS. The CSS you define here will be used for the whole supervisor page, and so it must be capable of matching all of the elements you want to match in the DOM.

The CSS is a setting which is either defined in:

- A User Setting (these can be defined by each user individually in **User Master Data**)
- An ACD Group Parameter (only used for values associated with ACD Groups - in the **ACD Groups ... Parameters Tab** for each ACD Group individually).
- A Client Parameter setting (these can be defined for the client account as a whole in **Client Master Data**)

The name of the setting is **CSS.AcdSupervisor**

Note

This is not a tutorial on how CSS or CSS selectors work - complete familiarity is assumed!

First we write the CSS we need. In this example, we are selecting all **td** elements in the DOM which contain a direct child **div** styled with the appropriate ServiceLevelAvailability bucket.

```
td:has(> div.Stats_ServiceLevelAvailability_Bucket_0) {
        background-color: #169A13;
        color: white;
}

td:has(> div.Stats_ServiceLevelAvailability_Bucket_1) {
    background-color: #FCC10F;
        color: white;
}

td:has(> div.Stats_ServiceLevelAvailability_Bucket_2) {
    background-color: #BB0F18;
        color: black;
}
```

Now we setup the parameter. Here you can see the parameter associated with Client Master Data:

Edit Client Parameter "CSS.AcdSupervisor"

ID :	109
Name :	CSS.AcdSupervisor
| Value : | ```
td:has(> div.Stats_ServiceLevelAvailability_Bucket_0) {
        background-color: #169A13;
        color: white;
}

td:has(> div.Stats_ServiceLevelAvailability_Bucket_1) {
    background-color: #FCC10F;
        color: white;
}

td:has(> div.Stats_ServiceLevelAvailability_Bucket_2) {
    background-color: #BB0F18;
        color: black;
}
``` |

Note

ⓘ

Note the CSS is loaded from both the User Master Data Settings and the Client Master Data Parameters. This gives the most flexibility, since individual user settings can be made or even individual colours used.

Result

If you have set all of this up correctly, you should see a result similar to the following:

Availability
100.00 %
0.00 %
0.00 %
100.00 %