


指标看板-文本组件字体颜色告警效果使用说明

 在V10.5.8 版本中，告警已转用“条件格式”功能实现，已禁用相关宏代码。

- [问题描述](#)
- [解决方法](#)
- [注意事项](#)

问题描述

在【指标看板】组件模板资源中，有以下几个文本组件模板实现了字体颜色告警的功能。**若需要改变告警条件中的告警值，应该如何设置呢？**



资源名称	组件模板	告警效果	默认告警条件
指标看板	图标在右单指标告警		大于等于0，字体红色告警 小于0，字体绿色告警
指标看板	无图标多指标告警指标看板		大于等于0，字体红色告警 小于0，字体绿色告警





解决方法



1、首先，编辑这些文本组件时，会发现上述告警的这些数据是通过公式得到的，并且有一对大括号将这个公式给括起来了。这对大括号就是告警功能的具体标识，宏代码中根据大括号找到需要进行告警设置的数据，然后进行告警处理。注意该大括号不要随意去除。




编辑文本[图标在右单指标告警（一）]

← → 微软雅黑 12px


A ∠ B I U  

[GetCellValue(self,1,1)]

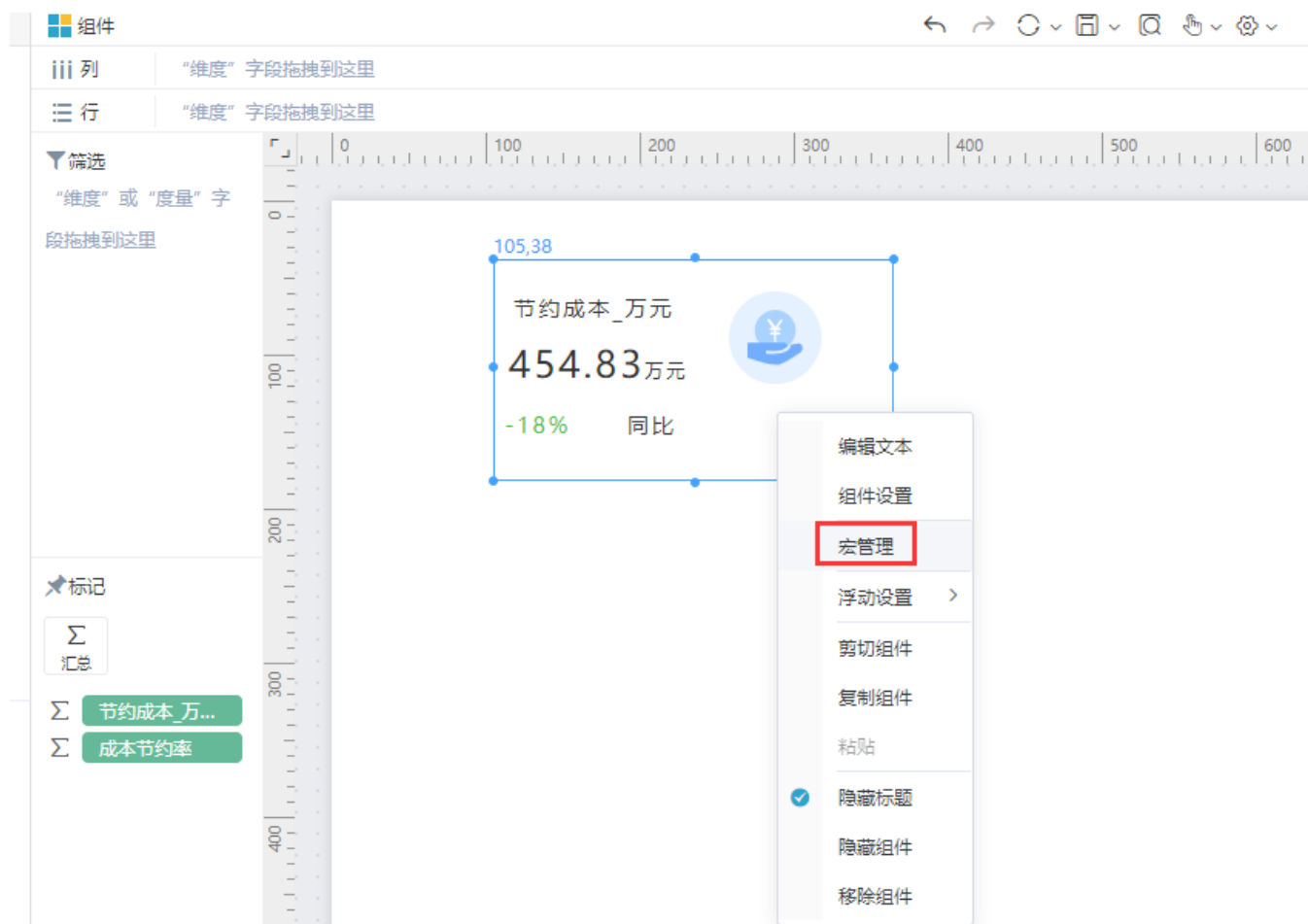
[GetCellValue(self,2,1)]万元 

[[GetCellValue(self,2,2)]] 同比

使用 {} 括起来作为告警的标识的

2、将组件模板复制过来，切换成自己的数据模型、配置好字段后。如果有需要修改告警的条件，如设置告警条件为 “大于20%，字体红色告警；小于10%，字体绿色告警”。可按如下步骤设置

(1) 选中该告警的文本组件，右键→宏管理



(2) 在宏管理界面上，点击一下刷新按钮，这样可以刷新宏代码菜单。然后双击对应的组件的宏菜单，打开宏代码编辑界面

III 导航 | 新建大屏可视化 | 宏管理[新建仪表盘] x

模块

自助仪表盘

图标在右单指标告警 (一)

告警-富文本改颜色

单击刷新

双击打开

名称: 告警-富文本改颜色 类型: ClientSide 对象: 图标在右单指标告警 (一)

```
1 /*****宏代码说明*****/
2 * 类型: ClientSide
3 * 事件: onAfterRender
4 * 效果描述: 根据富文本中的某个数值的正负, 修改该数值的字体颜色
5 * 简述实现步骤:
6 * (1) 在富文本中将需要设置告警的数值或者公式用 {} 符号包裹起来, 举例: "{28}"
7 * (2) 在宏中可修改一些配置项
8 * 注意事项:
9 * (1) 该宏判断数值的依据是基于有无正负号来判断的, 如有其它需求, 需要修改
10 * (2) {} 符号包裹的内容, 需要移除它们的字体颜色样式 (选中该内容, 富文本的设置项中可以移除字体颜色)
11 *****/
12 /**
13 * 告警配置类型
14 * color: 告警的字体颜色
15 * alarmValue: 告警的界定值
16 */
17 type AlarmConfig = {
18   color: string,
19   alarmValue: number
20 }
21 function main(page: IPage, portlet: IStaticTextPortlet) {
22   let config = {
23     type: ['STATIC_TEXT']
24   }
25   let type = portlet.getType();
26   if (config.type.indexOf(type) < 0) return;
27
28   /*****开始配置*****/
29   //正指标告警配置
30   let positiveAlarmConfig: AlarmConfig = {
31     color: "#43645"
```

(3) 根据需要修改代码中如下红框中的值。

名称: 告警-富文本改颜色

类型: ClientSide

对象: 图标在右单指

```
18     color: string,
19     alarmValue: number
20 }
21 function main(page: IPage, portlet: IStaticTextPortlet) {
22     let config = {
23         type: ['STATIC_TEXT']
24     }
25     let type = portlet.getType();
26     if (config.type.indexOf(type) < 0) return;
27
28     /*****开始配置*****/
29     //正指标告警配置
30     let positiveAlarmConfig: AlarmConfig = {
31         color: "#c43645",
32         alarmValue: 0 // 大于此数值正告警
33     }
34
35     //负指标告警配置
36     let negativeAlarmConfig: AlarmConfig = {
37         color: "#38c238",
38         alarmValue: 0 // 小于此数值负告警
39     }
40     /*****结束配置*****/
41
42     var plist: any = portlet.getHtmlContent().getElementsByTagName("p");
43     var spanList: any = portlet.getHtmlContent().getElementsByTagName("span");
44     var targetDomList: any = [];
45     targetDomList.push(...plist);
46     targetDomList.push(...spanList);
47     if (targetDomList.length == 0) {
48         return;
49     }
50     //从后往前找, 使其避免先从大的dom中找, 因为大的dom中可能包含小的dom, 而小的dom可能才是我们要设置样式的目标
```

三段式告警。

指标数值 > 上面的告警值, 红色预警

指标数值 < 下面的告警值, 绿色告警

指标数值位于两个告警值中间, 则不作告警

“大于20%, 字体红色告警; 小于10%, 字体绿色告警” 修改后代码如下:

名称: 告警-富文本改颜色

类型: ClientSide

```
12 /**
13  * 告警配置类型
14  * color: 告警的字体颜色
15  * alarmValue: 告警的界定值
16  */
17 type AlarmConfig = {
18   color: string,
19   alarmValue: number
20 }
21 function main(page: IPage, portlet: IStaticTextPortlet) {
22   let config = {
23     type: ['STATIC_TEXT']
24   }
25   let type = portlet.getType();
26   if (config.type.indexOf(type) < 0) return;
27
28   /*****开始配置*****/
29   //正指标告警配置
30   let positiveAlarmConfig: AlarmConfig = {
31     color: "#c43645",
32     alarmValue: 20 // 大于此数值正告警
33   }
34
35   //负指标告警配置
36   let negativeAlarmConfig: AlarmConfig = {
37     color: "#38c238",
38     alarmValue: 10 // 小于此数值负告警
39   }
40   /*****结束配置*****/
41 }
```

(4) 若需要修改边界条件, 比如修改 红色告警的条件为 “指标数值” 大于等于 上面的告警值。可以找到如下的代码。

名称: 告警-富文本改颜色

类型: ClientSide

对象: 图标在右单指

```
51     for (let i = targetDomList.length - 1; i >= 0; i--) {
52         if (targetDomList[i].innerText.match(getRegular())) {
53             setColor(targetDomList[i], positiveAlarmConfig, negativeAlarmConfig);
54         }
55     }
56 }
57
58 /**
59  * 给目标dom元素设置告警颜色
60  */
61 function setColor(targetDom: any, positiveAlarmConfig: AlarmConfig, negativeAlarmConfig: AlarmConfig) {
62     //在同一个p标签匹配到告警图标的字符串以及要判断的告警数值
63     let value = targetDom.innerText.match(getRegular());
64     if (value) {
65         value = value[0];
66         //去除{}和%
67         value = value.replace(/{\}|%/g, '');
68         //将字符串转为数值
69         value = Number(value);
70         //通过正负来判断是否需要告警
71         if (value > positiveAlarmConfig.alarmValue) {
72             targetDom.innerHTML = getAlarm(targetDom, positiveAlarmConfig.color);
73         } else if (value < negativeAlarmConfig.alarmValue) {
74             targetDom.innerHTML = getAlarm(targetDom, negativeAlarmConfig.color);
75         } else {
```

在 > 后面加上等于号=, 如下

名称: 告警-富文本改颜色

类型: ClientSide

```
59  * 给目标dom元素设置告警颜色
60  */
61 function setColor(targetDom: any, positiveAlarmConfig: AlarmConfig, negativeAlarmConfig: /
62     //在同一个p标签匹配到告警图标的字符串以及要判断的告警数值
63     let value = targetDom.innerText.match(getRegular());
64     if (value) {
65         value = value[0];
66         //去除{}和%
67         value = value.replace(/{\}|%/g, '');
68         //将字符串转为数值
69         value = Number(value);
70         //通过正负来判断是否需要告警
71         if (value >= positiveAlarmConfig.alarmValue) {
72             targetDom.innerHTML = getAlarm(targetDom, positiveAlarmConfig.color);
73         } else if (value < negativeAlarmConfig.alarmValue) {
74             targetDom.innerHTML = getAlarm(targetDom, negativeAlarmConfig.color);
75         } else {
```

(5) 小于的告警条件修改也同理。只要在

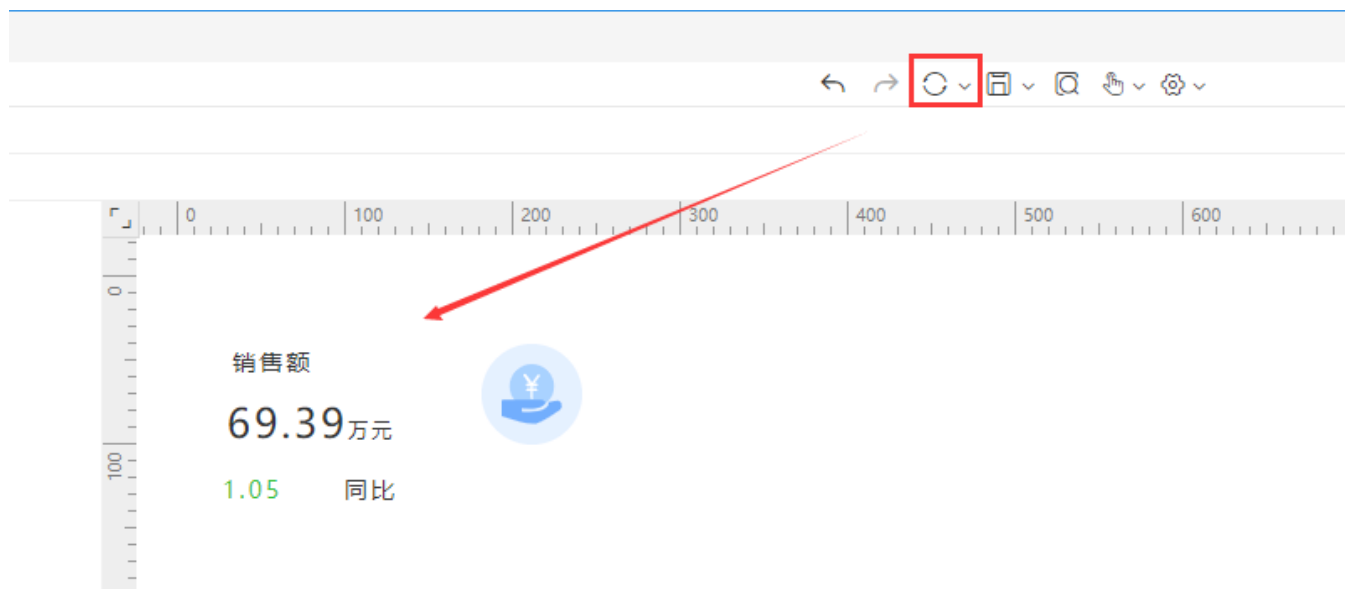
if (value < negativeAlarmConfig.alarmValue) { 中, 小于号的后面加上等于号即可。

名称: 告警-富文本改颜色

类型: ClientSide

```
59  * 给目标dom元素设置告警颜色
60  */
61  function setColor(targetDom: any, positiveAlarmConfig: AlarmConfig, negativeAlarmConfig:
62      //在同一个p标签匹配到告警图标的字符串以及要判断的告警数值
63      let value = targetDom.innerText.match(getRegular());
64      if (value) {
65          value = value[0];
66          //去除{}和%
67          value = value.replace(/\{|}%/g, '');
68          //将字符串转为数值
69          value = Number(value);
70          //通过正负来判断是否需要告警
71          if (value >= positiveAlarmConfig.alarmValue) {
72              targetDom.innerHTML = getAlarm(targetDom, positiveAlarmConfig.color);
73          } else if (value <= negativeAlarmConfig.alarmValue) {
74              targetDom.innerHTML = getAlarm(targetDom, negativeAlarmConfig.color);
75          } else {
```

(6) 点击“宏管理”界面 右下角的“保存”按钮，回到组件编辑界面，点击工具栏上的“刷新”按钮，即可看到效果。



注意事项

1、如果产品为2022年6月8日前的V10.5版本，且告警数值为百分比格式（如12.3%），那么大于30%的告警条件，对应的alarmValue的值也是 30（而不是0.03 或者3%）。如下：

名称: 告警-富文本改颜色

类型: ClientSide

对象: 图标在右单

```
4  * 效果描述: 根据富文本中的某个数值的正负, 修改该数值的字体颜色
5  * 简述实现步骤:
6  * (1) 在富文本中将需要设置告警的数值或者公式用 {} 符号包裹起来, 举例: "{28}"
7  * (2) 在宏中可修改一些配置项
8  * 注意事项:
9  * (1) 该宏判断数值的依据是基于有无正负号来判断的, 如有其它需求, 需要修改
10 * (2) {} 符号包裹的内容, 需要移除它们的字体颜色样式 (选中该内容, 富文本的设置项中可以移除字体颜色)
11 *****/
12 /**
13 * 告警配置类型
14 * color: 告警的字体颜色
15 * alarmValue: 告警的界定值
16 */
17 type AlarmConfig = {
18     color: string,
19     alarmValue: number
20 }
21 function main(page: IPage, portlet: IStaticTextPortlet) {
22     let config = {
23         type: ['STATIC_TEXT']
24     }
25     let type = portlet.getType();
26     if (config.type.indexOf(type) < 0) return;
27
28     /*******开始配置******/
29     /**正指标告警配置
30     let positiveAlarmConfig: AlarmConfig = {
31         color: "#c43645",
32         alarmValue: 3 // 大于此数值正告警
33     }
34
35     /**负指标告警配置
36     let negativeAlarmConfig: AlarmConfig = {
37         color: "#38c238",
38         alarmValue: 0 // 小于此数值负告警
39     }
40     /*******结束配置******/
41 }
```

2、本组件模板中的告警代码, 只适用于以下三种类型的数据格式

- (1) 纯数字。如123、123.23
- (2) 纯数字带千分位符。如345,334、2,345,345.33
- (3) 百分比数字。如89.3%、3,45.35%

若有其他带前缀、后缀等数据格式, 可根据用户实际需要定制服务方式实现。

3、通过文本组件实现告警的组件模板, 具体的宏代码逻辑只支持文本组件中有一个数值进行告警设置。不支持多个数值进行告警。

若有此场景, 可通过多个告警文本组件叠加摆放进行实现。或者以定制服务方式根据需求实现。