宝信基础自动化软件iPlat-BA用户手册

当前版本V6

目录

1. 安装指南 4
1.1 Windows下安装与启动 4
1.2 Linux下安装与启动 9
1.3 高可用版本安装与启动 13
1.4 iPlat-BA达梦数据库版本安装 19
1.5 安装许可证 19
1.6 系统登录 20
1.7 如何执行卸载 20
2. 快速入门 27
2.1 新建SCADA节点 27
2.2 驱动设备连接 28
2.3 新建变量 32
2.4 配置部署 36
2.5 组态画面搭建 39
2.6 组态画面运行 44
3. SCADA创建和部署 45
3.1 SCADA节点创建 45
3.2 SCADA节点配置 47
3.3 SCADA节点部署 49
3.4 SCADA节点配置导入导出 52
3.5 变量类型 231
3.6 变量属性 232
3.7 变量命名规则 235
3.8 配置AI变量基本属性 236
3.9 配置DI变量基本属性 245
3.10 模拟量输出AO 251
3.11 数字量输出DO 256
3.12 二进制大对象量BLOB 260
3.14 文本量TXT 263
3.15 模拟计算量CA 266
3.16 数字计算量CD 272
3.17 分级报警量LA 278
3.18 多数字量输入MDI 285
3.19 创建类模板及自定义属性 291
3.20 添加子系统 296
3.21 创建对象 298
3.22 对象变量查询 299

3.23 对象点使用示例 300
3.24 配置MidVarDriver变量 305
3.25 为本地变量写值 307
3.26 绑定本地变量 307
3.27 全局变量的配置 308
3.28 系统变量的分类 310
4. 组态与画面 314
5.1 设计器基本功能 315
5.2 2D画面搭建与运行 320
5.3 元素功能属性 328
5.4 元素属性总表 346
5.5 元素属性绑点 350
5.6 元素坐标与定位 356
5.7 元素闪烁配置 360
5.8 闪烁速度：图元闪烁的速度 368
5.9 画面图层使用 368
5.10 画面跳转与弹窗 372
5.11 画面签入签出 390
5.12 画面预览 394
5.13 元素的变量替换 396
5.14 画面的变量替换 397
5.15 画面变量导出和导入 401
5.16 画面查找与定位 402
5.17 画面导出导入 404
5.18 画面历史版本 407
5.19 画面快捷方式 415
5.20 自定义字体导入 416
5.21 画面页面配置 417
5.22 画笔绘制 419
5.23 图元使用 427
5.24 系统图元 429
5.25 图元自定义 464
5.26 图元分组 472
5.27 图元引用列表 485
5.28 实时报警控件 492
5.29 历史报警控件 497
5.30 实时趋势控件 500
5.31 历史趋势控件 503
5.32 文本趋势 521

5.33 图元趋势配置 522
5.34 图元趋势 582
5.35 组态回放控件 586
5.36 表格控件 591
5.37 多层级下拉框图元 613
5.38 高性能趋势 631
5. 数据发布 644
5.1 转储数据到ODBC数据源 644
5.2 modbus发布 658
5.3 opcua发布 672
5.4 opc发布 681
6. 联动介绍 681
7.1 触发源介绍 682
7.2 触发源分类配置 682
7.3 触发源配置 683
7.4 事件触发源 685
7.5 时间触发源 686
7.6 自定义触发源 687
7.7 预案介绍 688
7.8 预案分类配置 688
7.9 动作配置 690
7.10 变量控制 694
7.11 联动运行 701
7.12 变量表达式一览 701
7. 权限与安全 706
7.13 用户权限配置 706
7.14 管理端页面授权 708
7.15 配置过程 717

1. 安装指南

硬件要求

•处理器：≥InteI Xeon(TM)8 CPU*3.00GHz
•内存配置：≥16G
•硬盘配置：≥500G
•网卡：100M/1000M

操作系统要求

•操作系统：
。Windows 10 version 1607以上版本
。Windows Server 2016以上版本
。Redhat 7.8版本
。银河麒麟V10服务器版
。统信V20服务器版

注意

宝信基础自动化软件Windows只有64位版本。
宝信基础自动化软件Linux只有64位版本。

前端画面展示浏览器要求

•**3.0版本**：仅支持Chrome浏览器94版本及以上
•**4.0版本**：支持Chrome 116及以上版本

1.1 Windows下安装与启动

安装步骤

注意

- 安装前检查当前机器，如果安装了Tomcat、MySQL、Redis软件，应要卸载后再安装
- 安装开始前，请先查阅安装环境要求。
- 如果安装过程中报错缺少系统动态库等，可以上网下载directx修复工具进行修复系统
- 关闭Windows防火墙和杀毒软件避免某些安装部分脚本被杀掉

务必请用管理员权限运行`install-ba.exe`。

注意：如果是初次安装或重新覆盖安装，请选择"全新安装"。重新覆盖安装时，安装目录**不可变更**。对于已安装过，但需要变更安装目录的情况，请先选择"卸载"，卸载以前安装的iPlatBa（**卸载时将自动保留原有安装目录下的`data`目录，以便备份**），然后进行全新安装。

点击“下一步”，在安装路径设置页面中，点击"浏览"选择要安装的目录。

点击“下一步”，进入“选择安装”版本页面，根据需要选择**单机版**或者**分布式版本**。**注意：**单机版包含SCADA管理和权限管理，分布式版本则没有。

点击“安装”，等待安装完成。

安装完成后如何启动服务

SCADA启动

Windows下安装后，会在桌面生成一个**服务管理器**快捷方式。右击该快捷方式，选择**"以管理员身份运行"**。

在打开的服务管理器可视化界面中，点击**进程管理**可以查看所有服务是否正常运行。

MySQL,Redis,Tomcat服务自启动说明

安装程序会将mysql-scada,redis-scada,tomcat-cview注册为系统服务，并设置为自动启动。tomcat-cview服务会确保在MySQL以及Redis服务启动之后才启动。

注意： tomcat-cview服务无法通过右击服务列表正常关闭，需要在任务管理器详细信息中（见图4）右击进程并选择"结束任务"。

图1：Windows服务列表（示例）

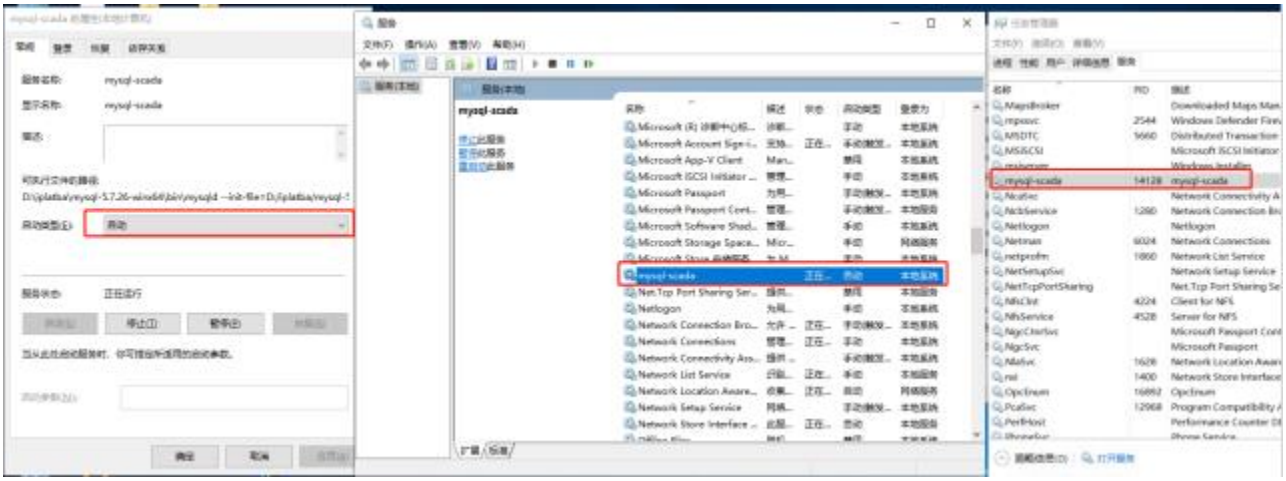


图2：Windows服务列表（示例）

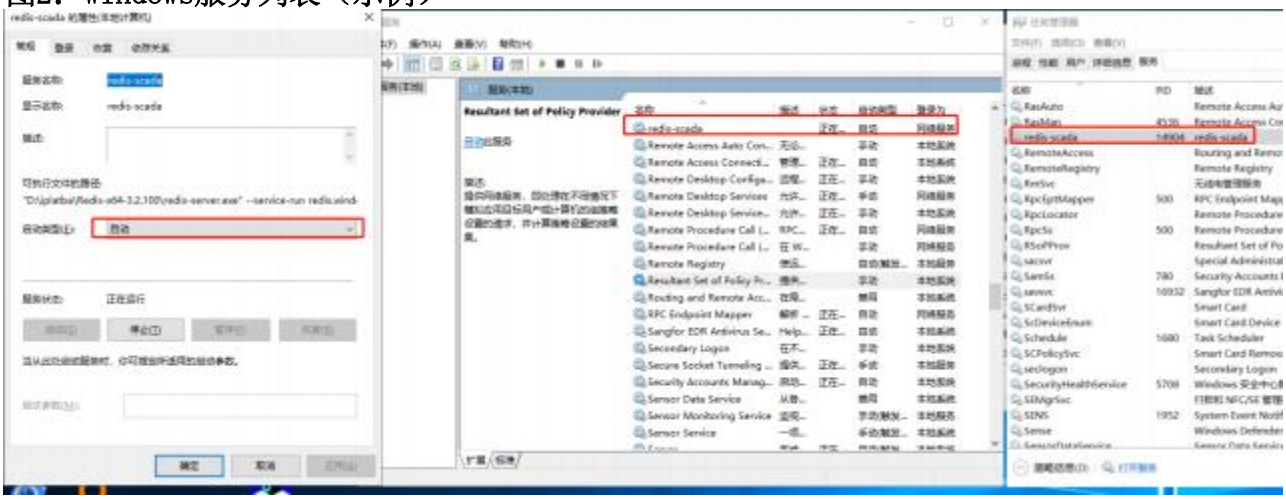


图3：Windows服务列表（示例）

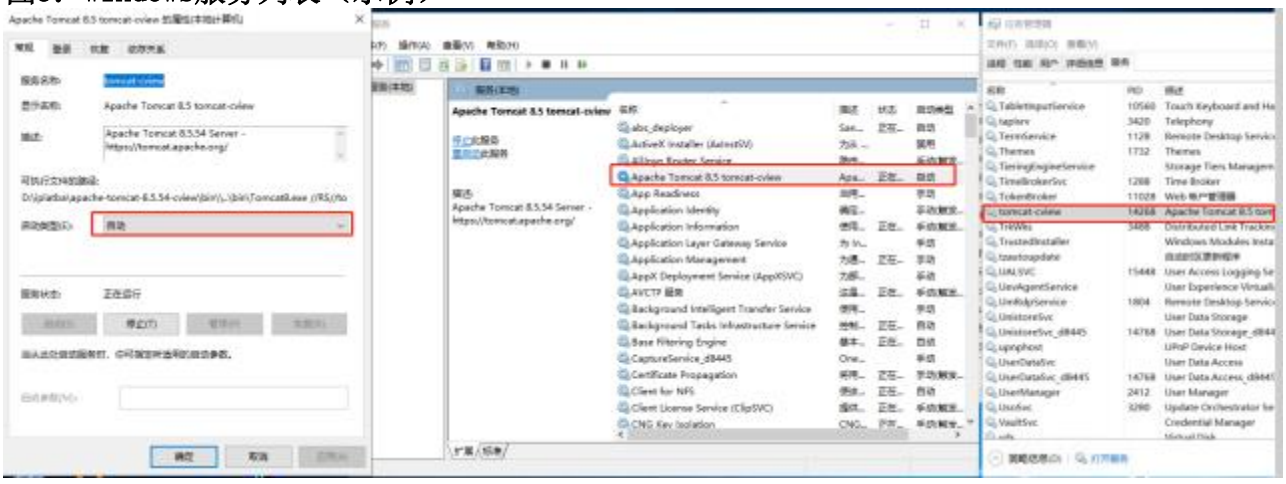
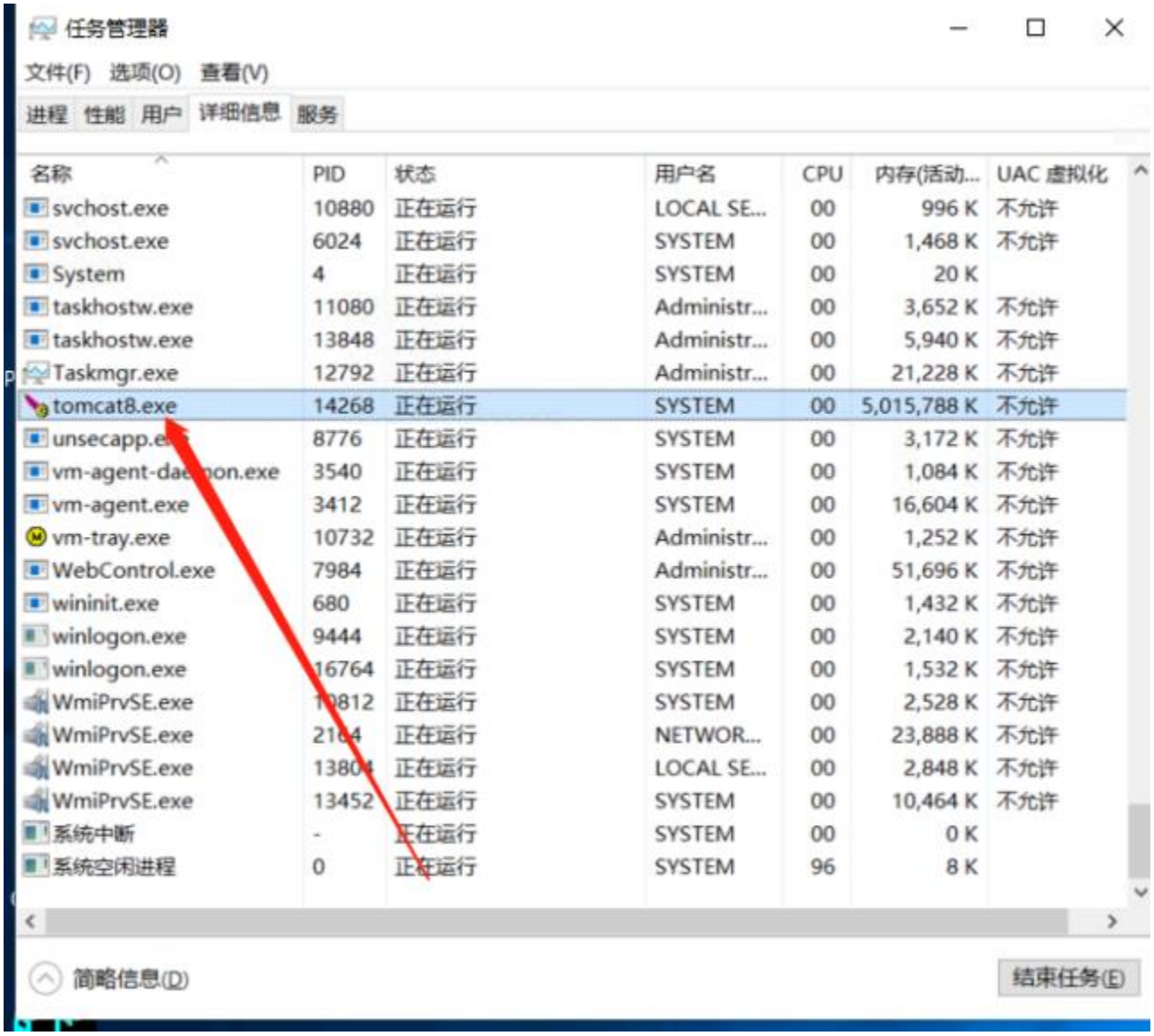


图4：通过任务管理器结束tomcat-cview进程



访问地址

宝信基础自动化软件安装完毕后的访问地址为：
http:/IP:8080/iplat/login.jsp
请将IP替换为实际服务器的IP地址（例如本地访问可使用：
`http:/127.0.0.1:8080/iplat/login.jsp`）。

1.2 Linux下安装与启动

安装步骤

注意

- 安装开始前，请先查阅安装环境要求。保证安装环境没有安装 `mysql`, `redis`, `tomcat` 等软件。

使用相应权限的用户上传安装文件到相应权限的文件夹（推荐使用root用户），注意安装文件不能放在宝信基础自动化软件安装目录下，例如想要在/data目录安装，安装文件不要放在/data目录：

```
[root@iplatba-slave iplat-ba]# ll
total 801648
-rw-r--r--. 1 root root      1016 Jul 30 13:03 ba-setup.ini
-rwxrwxrwx. 1 root root      3650 Aug  3 13:56 install.sh
-rw-r--r--. 1 root root 820877684 Aug  6 10:55 iplatba-20210806.tar.gz
```

按实际情况修改ba-setup.ini文件配置

- 设置BAHA=0
- 设置LOCALIP为本机IP
- 设置MainIP为本机IP
- 设置BAINSTMODE=1•设置LINUXINSTDIR（如果想安装在根目录下，直接设置路径为/）

```
;iPlat-BA Configuration File
[BAHAPARAM]
;iPlat-BA is High-Available? 0 is No, 1 is Yes.
;BAHA=0
BAHA=0
;Local IP.
;LOCALIP=168.2.8.104
LOCALIP=168.2.8.108
;Main Host IP.
;MAINIP=168.2.8.104
MAINIP=168.2.8.108

;If you set BAHA=1, You must Add BAKIP and VIP.
;Back Host IP.
;BAKIP=168.2.8.105
BAKIP=168.2.8.106
;Virtual IP for iPlat-BA.
;VIP=168.2.8.106
VIP=168.2.8.105
;Network Interface Name
NETINTANME=ens33

[InstallMode]
;You can select iPlat-BA Install Mode. 0 is Lin+Win, 1 is Lin.
;Linux+Win: iPlat-DA/Tomcat/JRE install in Windows and ba-mysql/ba-redis/ba-keepalived/ba-nginx install in Linux.
;Lin: All Components intall in Linux.
BAINSTMODE=1

[Linux]
;Linux-iPlat-BA or Related Components Install Path. Please set Full Path.
LINUXINSTDIR=/

[Windows]
;If You select BAINSTMODE Lin+Win, You must set Win Param.
WINTOMCATINSTDIR=C:\iplatba\apache-tomcat-8.5.54
WINDAINSTDIR=C:\iplatba\iplatda
;The location of ZTFiles.
WINDATAINSTDIR=C:\iplatba\data
;Windows's IP
WINIP1=168.2.8.103
WINIP2=168.2.8.29
```

对install.sh授权，使之有执行权限

```
chmod+x install.sh
```

```
[root@iplatba-slave iplat-ba]# chmod +x install.sh
[root@iplatba-slave iplat-ba]#
```

执行安装命令，开始安装宝信基础自动化软件Linux

```
./install.sh
```

等待

安装完成。（注：安装速度视服务器性能而定）

```
mysql: [Warning] Using a password on the command line interface can be insecure.
INFO: Modify tedcc03 variables......
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
INFO: Import SQL SCRIPT Finished!
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
INFO: You select Linux Mode.
setenforce: SELinux is disabled
INFO: Start to Install ba-tomcat/ba-iplatda...
INFO: UnZip Installation Files...
INFO: Create Tomcat's User: tomcat Passwd: tomcat.
Changing password for user tomcat.
New password: BAD PASSWORD: The password is shorter than 8 characters
Retype new password: passwd: all authentication tokens updated successfully.
INFO: Check&Config JRE...
INFO: JAVA Enviroment already exists. Installation Will Skip.
INFO: Config ba-tomcat...
INFO: Config ba-iplatda...
INFO: Config Environment...
```

注："mysql：[Warning]"信息是因为MySQL安全机制不允许直接在命令行中输入密码而打印的提示，该提示不影响安装过程。

至此，宝信基础自动化软件Linux安装完成。

5.如何启动服务

（1）查看服务状态

```
service ba-iplatba status
```

```
[root@iplatba-master config]# service ba-iplatba status
------------iPlat-BA Proc Status------------
ba-mysql          Running.  USER: mysql.  PID: 12548
ba-redis          Running.  USER: root.   PID: 12829
ba-tomcat         Running.  USER: root.   PID: 26392
ba-iplatda        Running.  USER: root.

------------iplatda Proc Status------------
rmservice         Running.  USER: root.   PID: 8263
rmdatasync        Running.  USER: root.   PID: 8264
rmcfgsync         Running.  USER: root.   PID: 8265
pbscanner         Running.  USER: root.   PID: 8266
rdaserver         Running.  USER: root.   PID: 8306
evtservice        Running.  USER: root.   PID: 8307
alarmserver       Running.  USER: root.   PID: 8308
crservice         Running.  USER: root.   PID: 8309
drvctrl           Running.  USER: root.   PID: 8310
hisdatasvr        Running.  USER: root.   PID: 8311
ihd3transfer      Running.  USER: root.   PID: 8312
hd3configcenter   Running.  USER: root.   PID: 8709
hdstatssch        Running.  USER: root.   PID: 8710
hd3configsyncsvc  Running.  USER: root.   PID: 8711
hdrysrv           Running.  USER: root.   PID: 8712
hdnetmgr          Running.  USER: root.   PID: 8796
hdkernel          Running.  USER: root.   PID: 8313
ifdtransfer       Running.  USER: root.   PID: 8797
opcuaserver       Running.  USER: root.   PID: 8799
sqltransfer       Running.  USER: root.   PID: 8798
smartopserver     Running.  USER: root.   PID: 8800
cv6ccagent        Running.  USER: root.   PID: 8801
hisdataquery      Running.  USER: root.   PID: 8802
```

（2）整体启动服务

```
service ba-iplatba start
```

```
[root@iplatba-master config]# service ba-iplatba start
Starting ba-redis......
ba-redis is Running
Starting ba-tomcat......
Starting ba-iplatda......
kernel.sem = 4096 512000 1600 900
net.ipv4.ip_local_port_range = 15000 60000
net.ipv4.ip_local_port_range = 15000 60000
net.ipv4.ip_local_port_range = 15000 60000
net.ipv4.ip_local_port_range = 15000 60000
net.ipv4.ip_local_port_range = 15000 60000
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv4.ip_local_port_range = 15000 60000
Starting ba-iplatda...
Start ba-iplatda successful.
iPlat-BA Started!
```

（3）整体停止服务

```
service ba-iplatba stop
```

```
[root@iplatba-master config]# service ba-iplatba stop
Stopping ba-iplatda......
Stopping ba-iplatda...
Stop ba-iplatda successful.
Stopping ba-tomcat......
Stopping ba-redis......
Stopping ...
Waiting for Redis-Scada to shutdown ...
ba-redis stopped
iPlat-BA Stopped!
[root@iplatba-master config]#
```

（4）整体重启服务

```
service ba-iplatba restart
```

```
[root@iplatba-master config]# service ba-iplatba restart
Restarting ba-redis......
ba-redis stopped.
Starting ba-redis server...
Restarting ba-tomcat......
Restarting ba-iplatda......
Restarting ba-iplatda...
kernel.sem = 4096 512000 1600 900
net.ipv4.ip_local_port_range = 15000 60000
net.ipv4.ip_local_port_range = 15000 60000
net.ipv4.ip_local_port_range = 15000 60000
net.ipv4.ip_local_port_range = 15000 60000
net.ipv4.ip_local_port_range = 15000 60000
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv4.ip_local_port_range = 15000 60000
Restart ba-iplatda successful.
iPlat-BA Restarted!
```

（5）单独操作服务与整体类似，命令格式为：

```
service ba-mysql/ba-redis/ba-tomcat/ba-iplatda status/start/stop/restart
```

（mysql服务/redis服务/tomcat服务/Scada服务的状态/启动/停止/重启）

额外补充：LINUX下系统参数补充说明

用户可以自己选择在sysctl.conf中添加参数来修改，重启后生效。

名称	默认值	建议值	
tcp_retries2	15	5	当遇到类似于拔网线，断网卡之类的操作时，网络切换较慢（默认

1.3 高可用版本安装与启动

安装方式

安装开始前，请先查阅安装环境要求。

- 保证安装环境没有安装mysql,redis,tomcat,keepalived,nginx软件。
- 保证安装的所有机器时间同步
  - 两台Linux
  - 两台Windows（需要在每个Windows内虚拟出一个Linux虚拟机）
  - 四台Linux（两台tomcat，两台安装中间件）

推荐方式a安装高可用版本。

方式a两台Linux安装步骤

1. **使用相应权限的用户上传安装文件到相应权限的文件夹（推荐使用root用户），注意安装文件不能放在宝信基础自动化软件安装目录下，例如想要在/data目录安装，安装文件不要放在/data目录：**

```
[root@iplatba-slave iplat-ba]# ll
total 801648
-rw-r--r--. 1 root root      1016 Jul 30 13:03 ba-setup.ini
-rwxrwxrwx. 1 root root      3650 Aug  3 13:56 install.sh
-rw-r--r--. 1 root root 820877684 Aug  6 10:55 iplatba-20210806.tar.gz
```

2. **主机修改配置文件**
   - 编辑ba-setup.ini
   - 设置BAHA=1
   - 设置LOCALIP为本地IP
   - 设置MAINIP为主机IP
   - 设置BAKIP为备机IP
   - 设置VIP(VIP确定原则：通过向网络管理员申请,局域网内可用的空闲IP,保证VIP可以与主备机正常通信（主备机VIP设置参数应相同）
   - 设置NETINTNAME,使用ifconfig或者ip addr命令查看主机设置的网卡接口名称

```
[root@iplatba-master ~]# ifconfig
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 168.2.4.182  netmask 255.255.255.0  broadcast 168.2.4.255
        ether 00:50:56:10:7a  txqueuelen 1000  (Ethernet)
        RX packets 21278098  bytes 6737041342 (6.2 GiB)
        RX errors 0  dropped 10  overruns 0  frame 0
        TX packets 11370256  bytes 1350400652 (1.2 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 15620436  bytes 1049250354 (1000.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15620436  bytes 1049250354 (1000.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

NETINTANME

- 设置BAINSTMODE=1
- 配置示例如下

```
;iPlat-BA Configuration File
[BAHAPARAM]
;iPlat-BA is High-Available? 0 is No, 1 is Yes.
;BAHA=0
BAHA=1
;Local IP.
;LOCALIP=168.2.8.104
LOCALIP=168.2.4.182
;Main Host IP.
;MAINIP=168.2.8.104
MAINIP=168.2.4.181

;If you set BAHA=1, You must Add BAKIP and VIP.
;Back Host IP.
;BAKIP=168.2.8.105
BAKIP=168.2.4.182
;Virtual IP for iPlat-BA.
;VIP=168.2.8.106
VIP=168.2.4.101
;Network Interface Name
NETINTANME=ens192

[InstallMode]
;You can select iPlat-BA Install Mode. 0 is Lin+Win, 1 is Lin.
;Linux+Win: iPlat-DA/Tomcat/JRE install in Windows and ba-mysql/ba-redis/ba-keepalived/ba-nginx install in Linux.
;Lin: All Components intall in Linux.
BAINSTMODE=1

[Linux]
;Linux-iPlat-BA or Related Components Install Path. Please set Full Path.
LINUXINSTDIR=/data
```

本地IP

主机IP

备机IP

VIP

3. 主机执行安装脚本
   - 在安装包目录,执行chmod+x install.sh,然后执行./install.sh执行安装。(安装包不能放在宝信基础自动化软件安装目录下)

```
[root@iplatba-slave iplat-ba]# chmod +x install.sh
[root@iplatba-slave iplat-ba]# ./install.sh
INFO: Install Disk Free Space: 35 GB
INFO: iPlat-Scada Will be Installed at /data
kernel.sem = 4096 512000 1600 900
net.ipv6.conf.all.disable_ipv6 = 1
INFO: UnZip Installation File...
INFO: Start to Install MySQL/Redis...
INFO: UnZip Installation Files...
80.6MiB 0:00:01 [45.0MiB/s] [==========================================================================================>] 100%
4.08KiB 0:00:00 [7.71KiB/s] [==========================================================================================>] 100%
1.32MiB 0:00:00 [53.1MiB/s] [==========================================================================================>] 100%
1.51KiB 0:00:00 [77.6MiB/s] [==========================================================================================>] 100%
INFO: Modify Configuration Files & Config ba-redis...
Restarting network (via systemctl):                        [  OK  ]
INFO: Config ba-mysql...
INFO: Add MySQL's User.
```

4. 备机修改配置文件
   - 设置LOCALIP为本地IP
   - 设置NETINTNAME,使用ifconfig或者ip addr命令查看备机设置的网卡接口名称
   - 其余各项配置同主机配置文件
5. 备机执行安装脚本

   同步骤2。

注意：安装后手动启动tomcat,redis,keepalive，这些安装后不自启

6. 服务启动/停止命令一览（安装完后以后的启动停止命令全部推荐使用service执行！）

①ba-iplatba启动停止：

```
service ba-iplatba start
service ba-iplatba stop
```

②ba-tomcat启动停止：只启动停止tomcat

```
service ba-tomcat start
service ba-tomcat stop
```

③ba-tomcat运行状态监测：

```
service ba-iplatba status
```

④ba-nginx启动停止：

```
systemctl start ba-nginx
systemctl stop ba-nginx
```

⑤ba-keepalived启动停止：

```
systemctl start ba-keepalived
systemctl stop ba-keepalived
```

方式b两台Windows安装步骤

1. **Windows主机安装**

安装开始前，请先查阅安装环境要求。

①. 运行install-ba-ha.exe,如果是初次安装或重新覆盖安装，请选择“全新安装”，重新覆盖安装的话，安装目录不可变更。对于已安装过，但需要变更安装目录，请先选择“卸载”，卸载

以前安装的iplatba，【卸载时将自动保留原有安装目录下的data目录，以便备份】然后进行全新安装。

②. 点击"下一步"，在安装路径设置页面中，点击"浏览"选择要安装的目录。

③. 设置MySQL服务器地址,服务器地址为Linux虚拟机VIP。

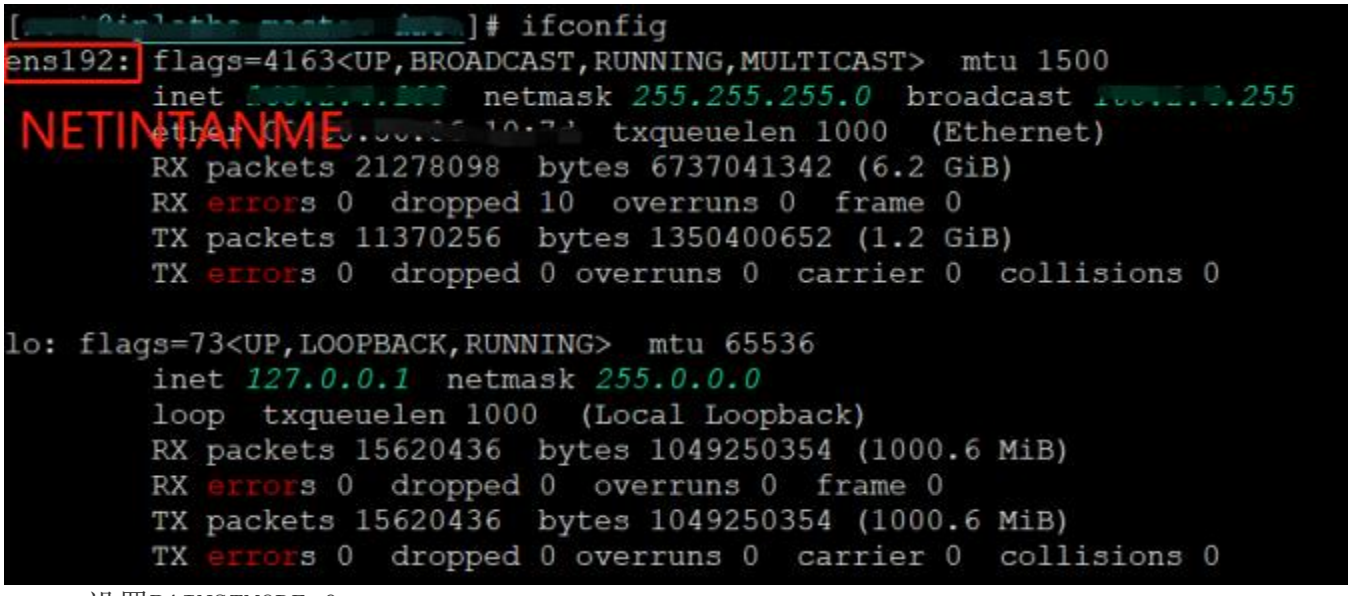
④. 点击"安装"，等待安装完成。

2. **Windows备机安装**

同Windows主机安装。

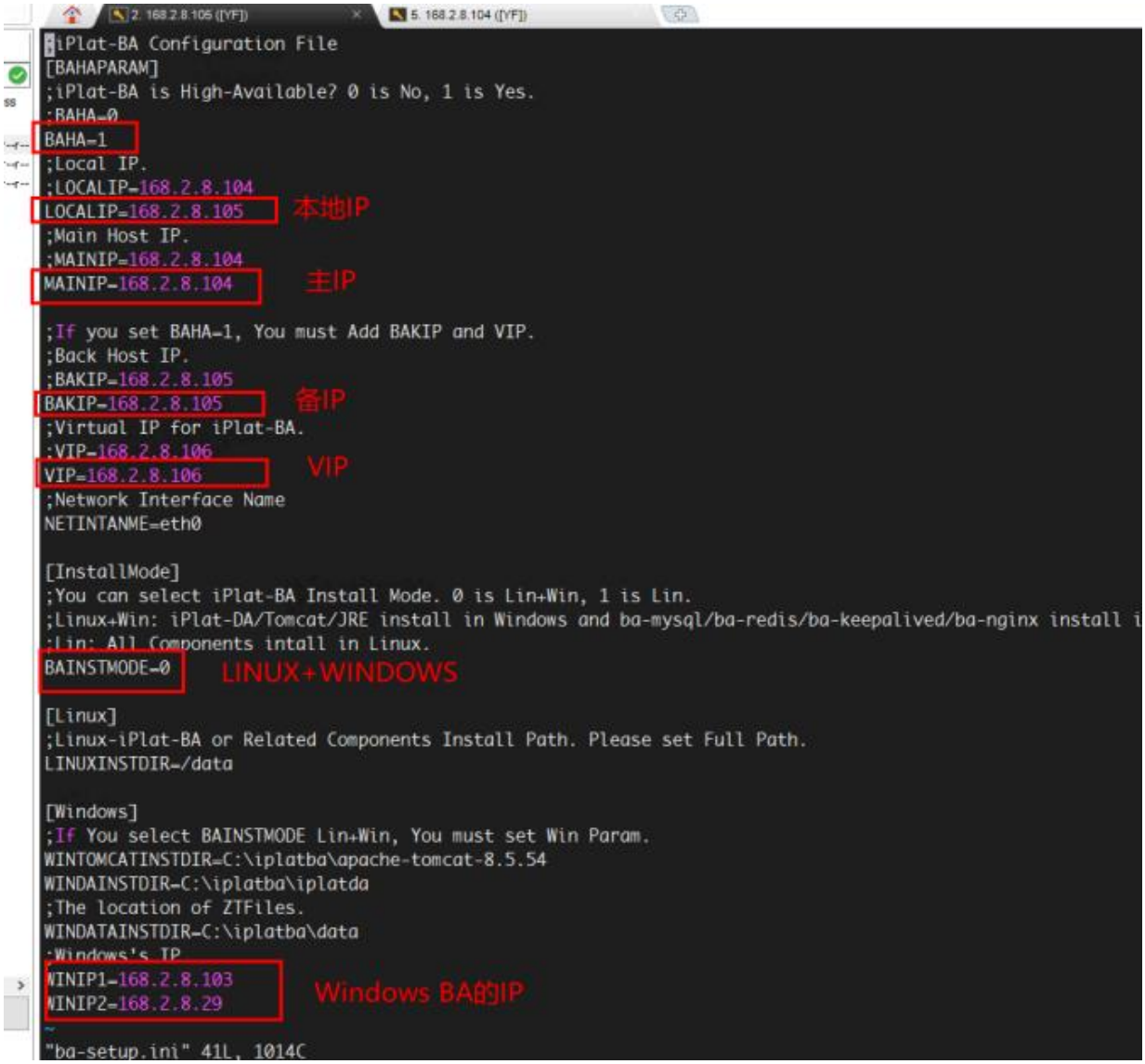
3. **Linux虚拟机主机安装**

- 编辑ba-setup.ini
- 设置BAHA=1
- 设置LOCALIP为本地IP
- 设置MAINIP为主机IP
- 设置BAKIP为备机IP
- 设置VIP（VIP确定原则：通过向网络管理员申请,局域网内可用的空闲IP,保证VIP可以与主备机正常通信（主备机VIP设置参数应相同，如主机VIP参数设置为168.2.8.106，则备用机VIP参数也为168.2.8.106））
- 设置NETINTNAME,使用ifconfig或者ip addr命令查看主机设置的网卡接口名称



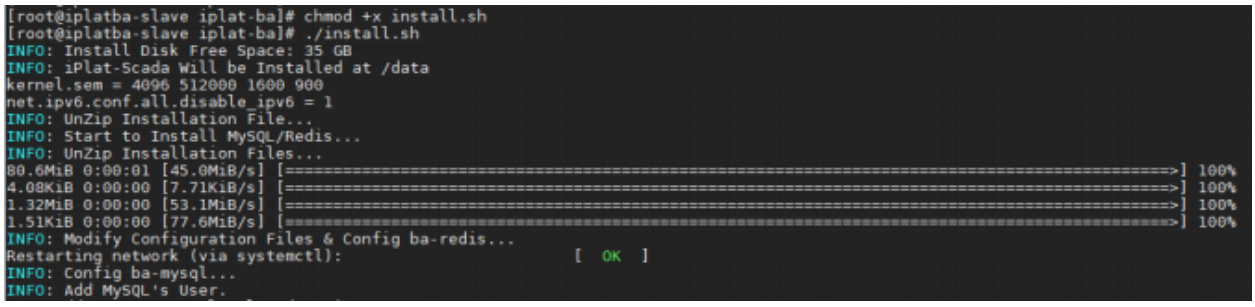
- 设置BAINSTMODE=0

•配置示例如下



•执行安装脚本：

•在安装包目录，执行chmod+xinstall.sh，然后执行./install.sh执行安装.（安装包不能放在宝信基础自动化软件安装目录下）



4. Linux虚拟机备机安装

①修改配置文件

•设置LOCALIP为本地IP

•设置NETINTNAME，使用ifconfig或者ip addr命令查看备机设置的网卡接口名称

•其余各项配置同Linux虚拟机主机配置文件

②. 执行安装脚本

•同Linux虚拟机主机执行安装脚本

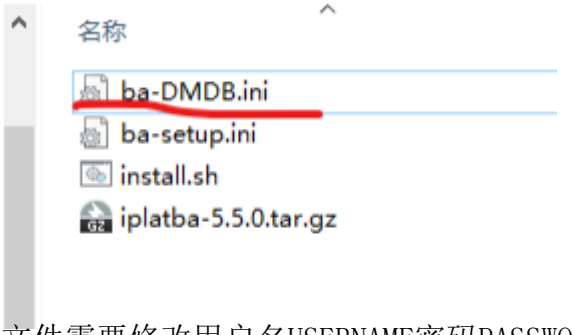
注意事项

- 当高可用环境中其中一台故障失去响应时，所有负荷均集中到一台，会导致请求响应速度变慢，请尽快解决故障恢复高可用环境
- 平台运维组件可以观察到高可用环境中两台服务器节点的状态，进入“节点状态”，点击“详情”如果有post返回报错，则节点Tomcat访问异常

1.4 iPlat-BA达梦数据库版本安装

安装步骤

1. 修改达梦数据库连接信息文件ba-DMDB.ini



文件需要修改用户名USERNAME密码PASSWORD ip地址和端口IP_PORT

```
[DMDB]
USERNAME=SYSDBA
PASSWORD=SYSDBA001
IP_PORT=127.0.0.1:5236
```

注意： 卸载后如果要重新安装IplatBA达梦数据库版本，需要删除达梦数据库中的`iplat` 、 `iplatdb` 、 `eplat` 这三个schema。

2. 后续安装步骤

请参考Linux下安装与启动

注意： 忽略【Linux下安装与启动】中关于mysql的部分。

1.5 安装许可证

默认安装包里没有宝信基础自动化软件正式许可证，默认安装之后每次启动有两小时的试用时间。BA的正式许可证需要申请，有需求的工程部门或者项目经理请联系：

联系人	李琳
公司	上海宝信软件股份有限公司
地址	上海浦东新区郭守敬路515号
电话	13764598787
邮箱	lilin@baosight.com

许可证存放路径
Windows：安装目录下iplatba\iplatda\executable\config
linux：安装目录下/iplatba/ba-iplatda/executable/config/

1.6 系统登录

系统初始用户名和密码为：admin/admin

> **重要提示：** 第一次登录系统时，用户必须修改admin用户的密码。

1.7 如何执行卸载

Windows系统下卸载宝信基础自动化软件

到宝信基础自动化软件安装目录下面，管理员权限运行目录下的uninstall.exe，卸载时将自动保留安装目录下的data目录，以便备份。

Linux系统下卸载宝信基础自动化软件

1. 进入到宝信基础自动化软件Linux安装目录，执行./uninstall【卸载时将自动保留原有安装目录下的data目录，以便备份】
2. 等待卸载完成。

```
[root@iplatba-master iplatba]# ll
total 12
drwxr-xr-x. 6 root  root    66 Aug  6 01:24 ba-iplatda
drwxr-xr-x. 4 root  root    75 Mar 19 05:16 ba-jre
drwxr-xr-x. 6 mysql mysql  141 Aug  6 01:24 ba-mysql
-rwxr-xr-x. 1 root  root  2123 Aug  6 01:24 ba-proc
drwxrwxrwx. 2 root  root   142 Aug  9 01:25 ba-redis
drwxrwxrwx. 9 root  root  4096 Aug  6 01:25 ba-tomcat
drwxr-xr-x. 5 root  root    48 Aug  5 17:01 data
-rwxr-xr-x. 1 root  root  1259 Aug  6 01:24 uninstall
[root@iplatba-master iplatba]# ./uninstall
Failed to stop ba-keepalived.service: Unit ba-keepalived.service not loaded.
Shutting down MySQL.... SUCCESS!
Stopping ...
Waiting for Redis-Scada to shutdown ...
redis stopped
Stopping ba-iplatda...
./uninstall: line 8:  7898 Killed                  service ba-iplatda stop
Failed to execute operation: No such file or directory
Removed symlink /etc/systemd/system/multi-user.target.wants/ba-tomcat.service.
Failed to execute operation: No such file or directory
[root@iplatba-master iplatba]#
```

方式a两台Linux卸载

1. 进入到宝信基础自动化软件Linux安装目录,执行./uninstall
2. 等待卸载完成。

```
[root@iplatba-slave iplatba]# ll
total 8
drwxr-xr-x. 6 root  root    66 Aug  9 14:27 ba-iplatda
drwxr-xr-x. 4 root  root    75 Mar 19 17:16 ba-jre
drwxr-xr-x. 5 root  root    86 Aug  9 14:27 ba-keepalived
drwxr-xr-x. 6 mysql mysql  141 Aug  9 14:27 ba-mysql
drwxrwxrwx. 3 root  root    69 Aug  9 14:27 ba-nginx
-rwxr-xr-x. 1 root  root  2123 Aug  9 14:27 ba-proc
drwxrwxrwx. 2 root  root    77 Aug  9 14:27 ba-redis
drwxrwxrwx. 9 root  root   263 Aug  6 10:46 ba-tomcat
drwxr-xr-x. 5 root  root    48 Aug  6 05:01 data
-rwxr-xr-x. 1 root  root  1259 Aug  9 14:27 uninstall
[root@iplatba-slave iplatba]# ./uninstall
Shutting down MySQL.. SUCCESS!
redis stopped.
Stopping ba-iplatda...
./uninstall: line 8:  2266 Killed                  service ba-iplatda stop
Removed symlink /etc/systemd/system/multi-user.target.wants/ba-nginx.service.
Removed symlink /etc/systemd/system/multi-user.target.wants/ba-tomcat.service.
Removed symlink /etc/systemd/system/multi-user.target.wants/ba-keepalived.service.
[root@iplatba-slave iplatba]#
```

3.卸载完成。

方式b两台Windows卸载

1. Windows主机卸载,双击宝信基础自动化软件安装目录下的uninstall.exe，卸载时将自动保留安装目录下的data目录，以便备份。

2.Windows备机卸载,同Windows主机卸载。

3.Linux虚拟机主机卸载,进入到安装目录,执行./uninstall,等待卸载完成。

4.Linux虚拟机备机卸载,同Linux虚拟机主机卸载。

如何升级版本

Windows升级的方法

Step1:将patch包文件和install-ba.exe文件放在iplatba的根目录

Step2:执行install-ba.exe，显示要升级的版本

注意：升级时会选择比当前宝信基础自动化软件版本高的升级包来执行。例如：当前宝信基础自动化软件的版本是3.0.2，升级程序会选择3.0.3版本的升级包进行升级。

宝信基础自动化软件以及其中各个组件的版本可以通过iplatba_version.xml文件查看：

iplatba_version.xml | 升级程序的会议纪要.txt | 周报.txt | proto_googleprotobuff_命令.txt | 更新的lib.txt

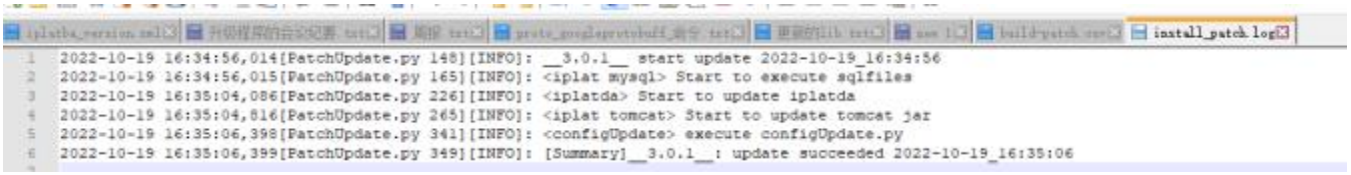
```
<?xml version="1.0" encoding="UTF-8"?>
<iplatba bit="64" installtime="2022-10-19 16:35" language="zh_cn" version="3.0.1">
  <iplatda installtime="2022-10-19 16:35" version="2.21.0">
    <history installtime="2022-09-16 13:32" version="2.20.0"/>
    <history installtime="2022-10-19 16:35" version="2.21.0"/>
  </iplatda>
  <iplat-base installtime="2022-09-16 13:32" version="2.8.7">
    <history installtime="2022-09-16 13:32" version="2.8.7"/>
  </iplat-base>
  <iplat-cm installtime="2022-09-16 13:32" version="2.34.0">
    <history installtime="2022-09-16 13:32" version="2.34.0"/>
  </iplat-cm>
  <iplat-dm installtime="2022-10-19 16:35" version="2.20.1">
    <history installtime="2022-09-16 13:32" version="2.19.0"/>
    <history installtime="2022-10-19 16:35" version="2.20.1"/>
  </iplat-dm>
  <iplat-sm installtime="2022-10-19 16:35" version="2.21.1">
    <history installtime="2022-09-16 13:32" version="2.20.0"/>
    <history installtime="2022-10-19 16:35" version="2.21.1"/>
  </iplat-sm>
  <iplat-res installtime="2022-09-16 13:32" version="1.13.0">
    <history installtime="2022-09-16 13:32" version="1.13.0"/>
  </iplat-res>
  <iplat-face installtime="2022-09-16 13:32" version="1.7.1">
    <history installtime="2022-09-16 13:32" version="1.7.1"/>
  </iplat-face>
  <iplat-event installtime="2022-09-16 13:32" version="2.8.0">
    <history installtime="2022-09-16 13:32" version="2.8.0"/>
  </iplat-event>
  <iplat-op installtime="2022-09-16 13:32" version="2.17.0">
    <history installtime="2022-09-16 13:32" version="2.17.0"/>
  </iplat-op>
  <iplat-opagent installtime="2022-09-16 13:32" version="2.5.0">
    <history installtime="2022-09-16 13:32" version="2.5.0"/>
  </iplat-opagent>
  <iplat-au installtime="2022-09-16 13:32" version="2.0.5">
    <history installtime="2022-09-16 13:32" version="2.0.5"/>
  </iplat-au>
  <iplat-is installtime="2022-09-16 13:32" version="1.0.0-SNAPSHOT">
    <history installtime="2022-09-16 13:32" version="1.0.0-SNAPSHOT"/>
  </iplat-is>
</iplatba>
```

install-ba.exe运行时显示的将要升级的版本，升级程序会自动选择当前目录中包含的所有可升级的补丁包：
（没有勾选的代表系统已经安装了该补丁包，如果只需要升级到某个版本，那需要手动将高于该版本的补丁包在目录中删除）

Step3:查看生成的install_patch.log文件，确认升级结果

升级完成后，界面会提示升级成功或失败，也可以通过查看install_patch.log获得更多详细信息。

成功的log：



失败的log：

```
2022-10-19 16:50:33,132[PatchUpdate.py 148][INFO]: __3.0.1__ start update 2022-10-19_16:50:33
2022-10-19 16:50:33,132[PatchUpdate.py 165][INFO]: <iplat mysql> Start to execute sqlfiles
2022-10-19 16:50:33,494[PatchUpdate.py 205][ERROR]: Execute sql failed: iplat-dm/V2_20_0__up.sql
2022-10-19 16:50:33,494[PatchUpdate.py 206][ERROR]:
(1)ERROR 1146 (42S02) at line 3: Table 'iplat.tedpi10' doesn't exist
--------------
DELETE FROM `TEDPI10` WHERE TREE_ENAME='DM' AND NODE_ENAME='BIDX04'
--------------


2022-10-19 16:50:33,526[PatchUpdate.py 205][ERROR]: Execute sql failed: iplat-dm/V2_20_1__up.sql
2022-10-19 16:50:33,526[PatchUpdate.py 206][ERROR]:
(1)ERROR 1146 (42S02) at line 2: Table 'iplat.tedcc03' doesn't exist
--------------
INSERT INTO TEDCC03 (STATUS, CONFIG_TYPE, CONFIG_CUE, REC_CREATOR, REC_CREATE_TIME, REC_REVISOR, REC_REVISE_TIME, CONFIG_ENV_ID, FKEY, FVALUE, TENANT
(1,'2',' ','admin','20190709162850','admin','20190819162231','f327c3b5-8d92-49d9-bc0a-94afbf38531d','iplat.bi.dx.long.rdaRequestTimeout','15000','BDA
--------------


2022-10-19 16:50:33,552[PatchUpdate.py 205][ERROR]: Execute sql failed: iplat-sm/V2_21_0__up.sql
2022-10-19 16:50:33,552[PatchUpdate.py 206][ERROR]:
(1)ERROR 1146 (42S02) at line 2: Table 'iplat.tedcc03' doesn't exist
--------------
INSERT INTO `tedcc03` VALUES (1, '2', ' ', 'admin', '20220817140409', 'admin', '20220817140409', '1', 'iplat.bi.sm.otherTable', '', 'BDAS', '0', '整
--------------


2022-10-19 16:50:33,557[PatchUpdate.py 226][INFO]: <iplatda> Start to update iplatda
2022-10-19 16:50:33,591[PatchUpdate.py 265][INFO]: <iplat tomcat> Start to update tomcat jar
2022-10-19 16:50:33,625[PatchUpdate.py 341][INFO]: <configUpdate> execute configUpdate.py
2022-10-19 16:50:33,625[PatchUpdate.py 351][ERROR]: __3.0.1__: update Error
2022-10-19 16:50:33,625[PatchUpdate.py 354][ERROR]: [Summary]Execute sql failed: iplat-dm/V2_20_0__up.sql

2022-10-19 16:50:33,625[PatchUpdate.py 354][ERROR]: [Summary]Execute sql failed: iplat-dm/V2_20_1__up.sql

2022-10-19 16:50:33,625[PatchUpdate.py 354][ERROR]: [Summary]Execute sql failed: iplat-sm/V2_21_0__up.sql
```

Step4:升级成功后，确认、重新启动宝信基础自动化软件，Tomcat，Mysql检查升级效果

Linux升级的方法

Step1:将patch包文件和install.sh文件放在宝信基础自动化软件的根目录，如/data/iplatba

Linux放置patch包文件的地址：

```
[root@iplatba-master iplatba]# pwd
/data/iplatba
[root@iplatba-master iplatba]# ll
total 28
drwxr-xr-x  6 root  root    66 Sep  1 15:01 ba-iplatda
drwxr-xr-x  8 10143 10143  273 Dec  9  2020 ba-jre
drwxr-xr-x 11 mysql mysql  211 Sep  7 13:44 ba-mysql
-rwxr-xr-x  1 root  root  2216 Sep  1 15:00 ba-proc
drwxrwxrwx  2 root  root   142 Oct 10 15:37 ba-redis
drwxrwxrwx 12 root  root   313 Oct 10 10:32 ba-tomcat
drwxr-xr-x  6 root  root    63 Aug 26 05:00 data
-rw-r--r--  1 root  root  1861 Oct 10 10:31 install_patch.log
-rwxr-xr-x  1 root  root  4485 Oct 10 10:15 install.sh
-rwxr-xr-x  1 root  root  1662 Oct 10 10:31 iplatba_version.xml
drwxr-xr-x  3 root  root    67 Oct 10 10:31 patch
-rwxr-xr-x  1 root  root   144 Oct 10 14:11 test.sh
-rwxr-xr-x  1 root  root  1274 Sep  1 15:00 uninstall
[root@iplatba-master iplatba]#
```

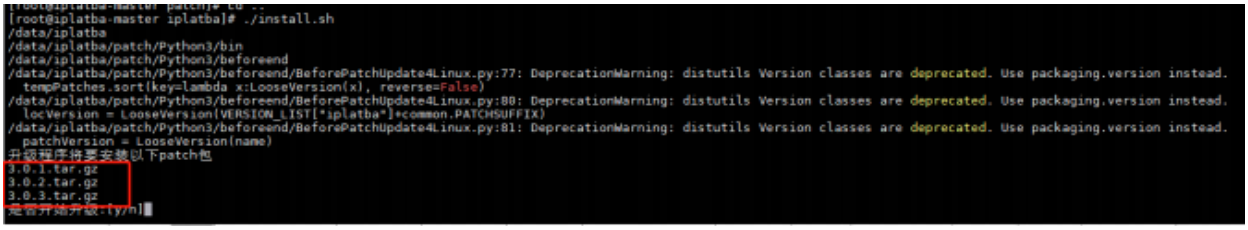
注意：如果install.sh没有可执行权限，需要给install.sh赋予可执行权限：

chmod+x install.sh

Step2 执行install.sh

> **注意：**升级时会选择比当前宝信基础自动化软件版本高的升级包来执行。例如：当前宝信基础自动化软件的版本是3.0.2，升级程序会选择3.0.3版本的升级包进行升级。

install.sh运行时显示的将要升级的版本（输入y继续升级，输入n结束）：



Step3:查看生成的install_patch.log文件，确认升级结果

升级完成后，界面会提示升级成功或失败，也可以通过查看`install_patch.log`获得更多详细信息。

Step4:升级成功后，确认、重新启动iplatda、tomcat、mysql，检查升级效果

可以通过以下命令，查看程序运行状态：

```
service      ba-tomcat

status   service   ba-

iplatda          status

service       ba-mysql

status
```

注意事项

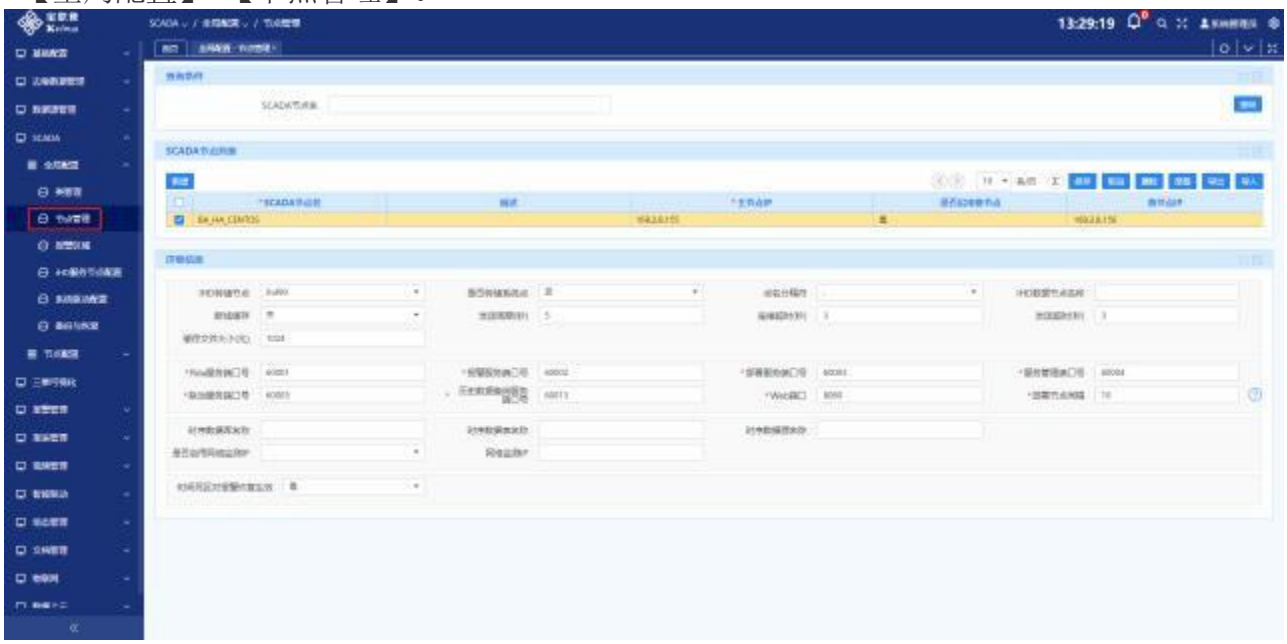
在升级Linux的高可用的宝信基础自动化软件时，需要按照主机完成升级后，再执行备机的升级，不能同时对两台机器进行升级操作。同时执行升级操作会导致升级的sql脚本执行错误。

2. 快速入门

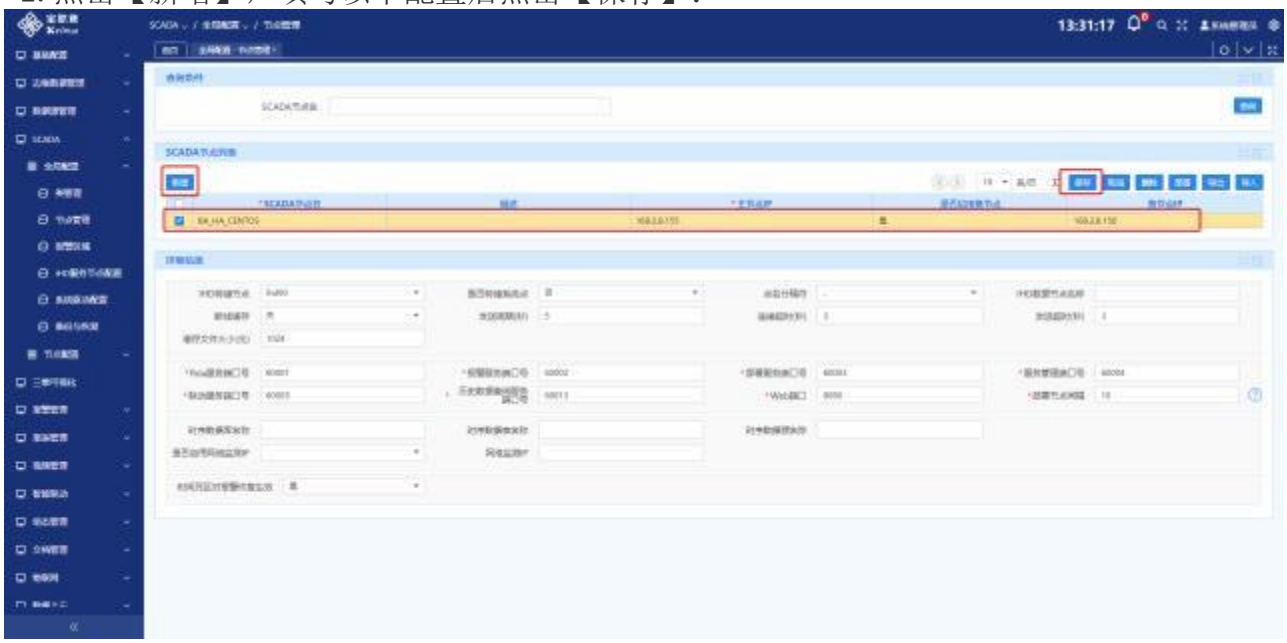
2.1新建SCADA节点

步骤演示

1. 打开谷歌浏览器，输入搭建环境设备的IP（默认端口8080），登录iPlat，进入【SCADA】-【全局配置】-【节点管理】。



2. 点击【新增】，填写以下配置后点击【保存】：



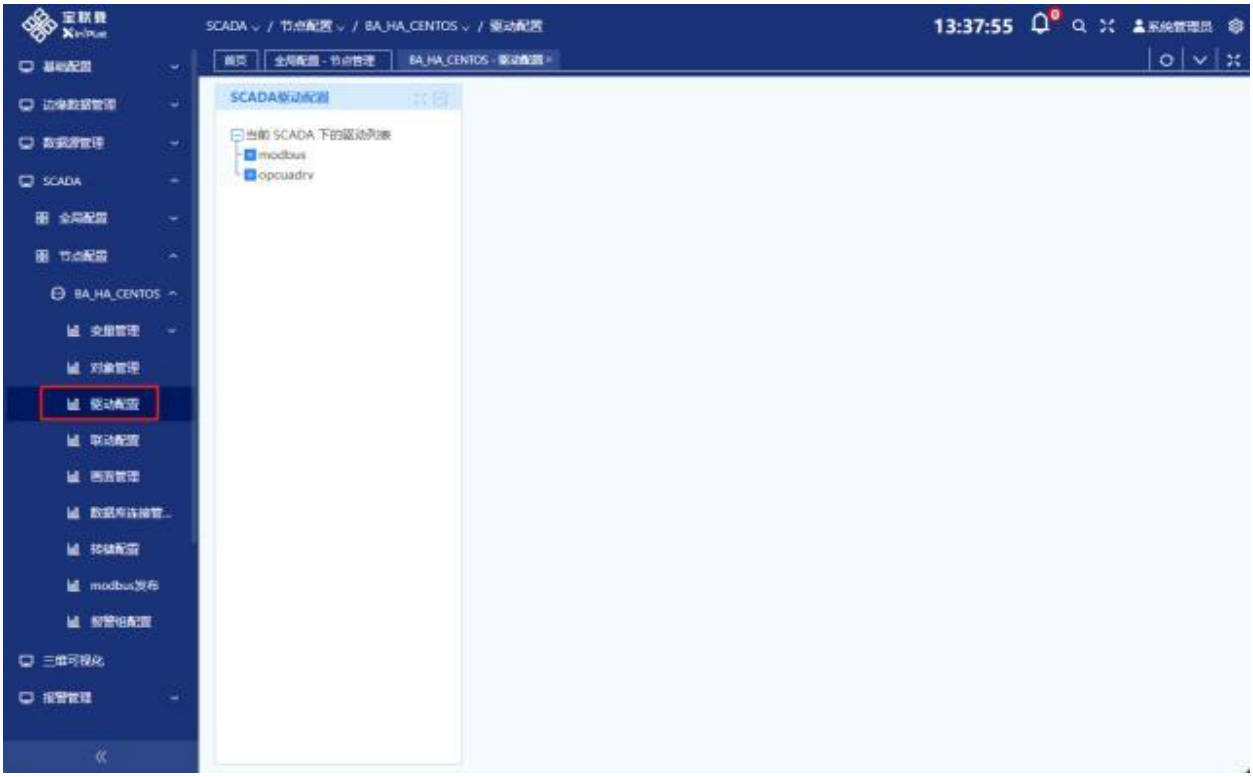
- SCADA节点名：必填，由字母、数字、下划线组成，必须以字母开头，不可使用特殊关键字（如E、PI、true、false等）；
- 描述：非必填，用于说明节点用途；
- 主节点IP：必填，若无冗余节点可填127.0.0.1；
- 是否启用备节点：非必填，若启用需填写备节点IP；

备节点IP：非必填，默认端口由系统自动填写。

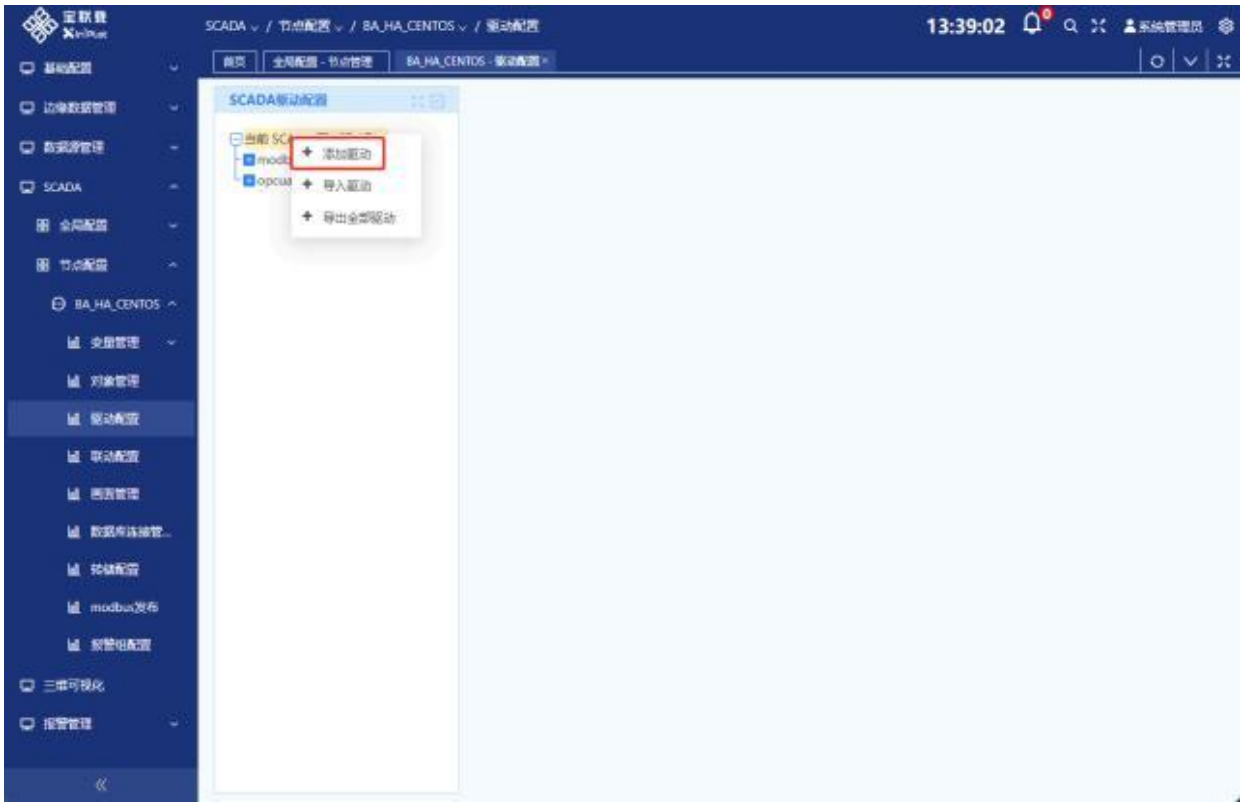
2.2 驱动设备连接

步骤演示

1.进入【SCADA】-【节点配置】-【驱动配置】。



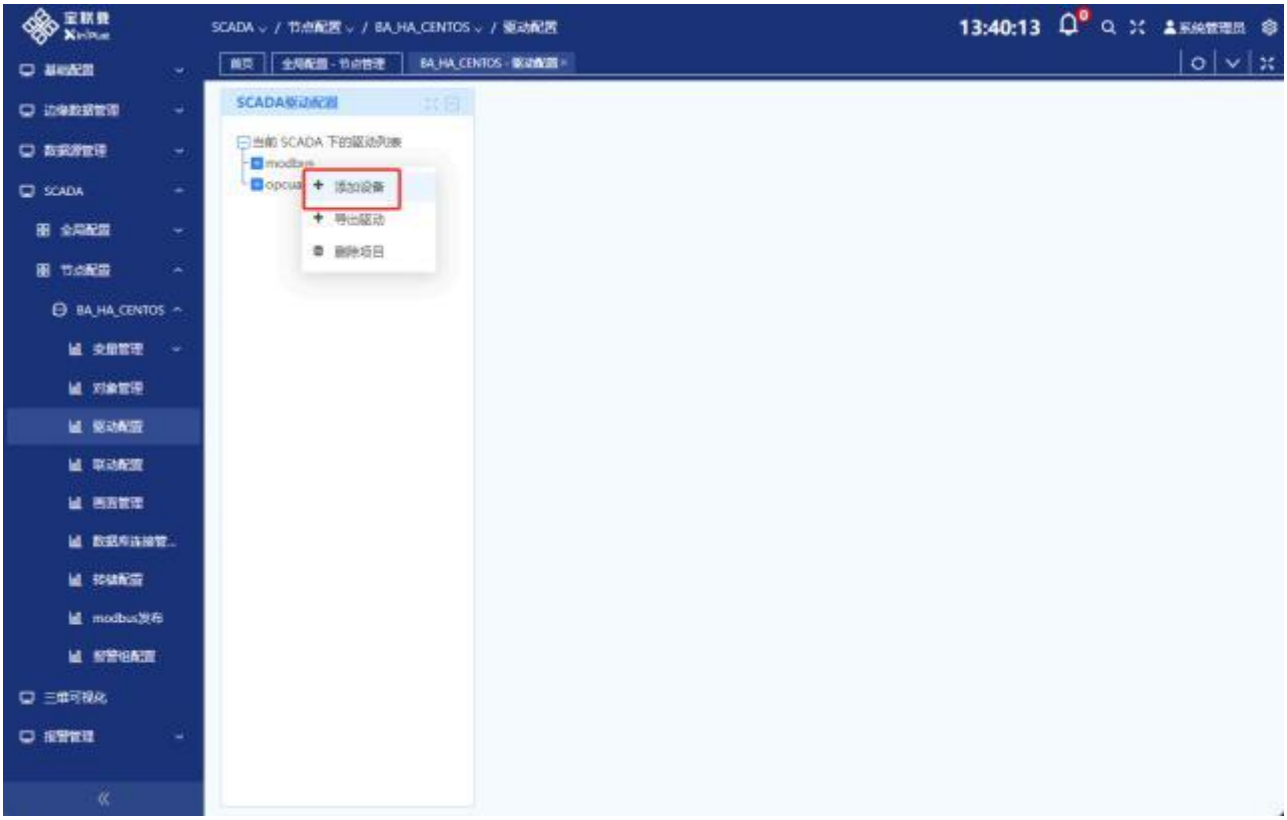
2.右键点击“当前SCADA下的驱动列表”→【添加驱动】。



3.选择驱动类型（如Modbus），点击【保存】。

4. 驱动添加成功后显示在左侧栏中。

5. 右键点击驱动名称→【添加设备】。



6. 填写设备信息后点击【保存】。

- **设备名称**：不能为空，不能以数字开头；
- **IP地址**：目标设备IP；
- **端口号**：默认为502；
- **连接类型**：选择TCPClient；
- **响应超时**：连接超时时间；
- **协议类型**：如Modbus/TCP；

- **备机地址/端口**：备用设备信息；
- **是否支持多连接**：是否允许多连接；
- **任务号**：同一任务号设备在同一进程中处理；
- **扩展参数**：用户自定义；
- **多寄存器命令**：是否启用多寄存器写入。

支持的驱动列表

驱动名	支持的设备	备注	操作系统
OPCDRV	OPC协议		Windows
OPCUA	OPCUA协议		Windows
			CentOS
SiemensS7Drv	西门子S7系列PLC	目前更推荐使用snap7驱动	Windows CentOS
IEC104Driver	104协议	支持所有标准104协议的设备	Windows CentOS
CIPDriver	AB PLC	AB PLC 1756	Windows CentOS
CIPDriverX	AB PLC	支持结构数组的地址格式	Windows CentOS
ICGDriver	ICG网关		Windows
modbus	modbus协议	支持modbus TCP和RTU，施耐德65150，贝加莱	Windows CentOS
snmp2	snmp2协议		Windows CentOS
snap7drv	西门子S7系列PLC	支持S7 200/300/400(H)/1200/1500	Windows CentOS
tdcdrv	tdc产品	支持百毫秒级别高速读	Windows CentOS
tdcwritedrv	tdc产品		Windows CentOS

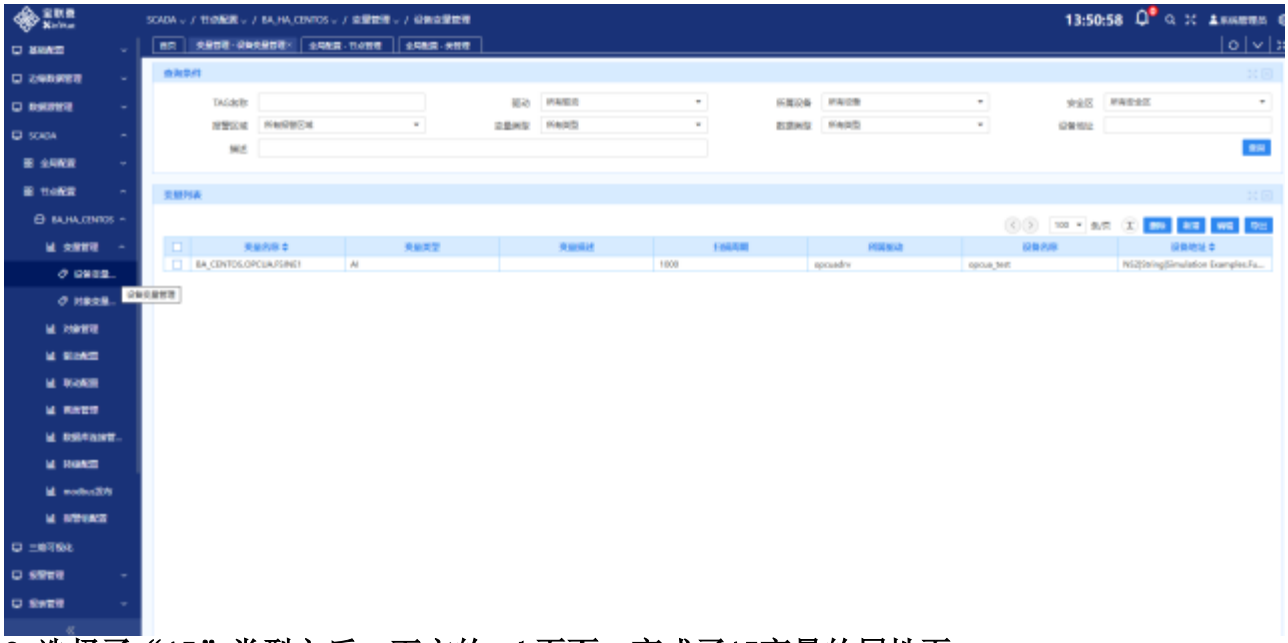
cvrdadriver	与其他iPlat SCADA数据交互		Windows CentOS
midvardriver	中间变量驱动		Windows CentOS
iplaturedrv	iplature数据采集/控制下发		Windows CentOS
dbdriver	关系数据库	SQLite、MySQL、DB2、PostgreSQL、Oracle等	Windows CentOS
melsecdrv	三菱PLC	三菱PLC（Q/L和iQ-R）支持设备冗余	Windows CentOS
hitachidrv	日立PLC驱动	支持MICA	Windows CentOS
codesysdrv	宝信PLC驱动	支持mpoc vpoc	Windows CentOS
iplatdadriver	iPlat模拟驱动	可以模拟其它驱动生成数据，测试验证等	Windows CentOS
abdrv	罗克韦尔PLC		Windows
fileblockdrv	文件块驱动		Windows CentOS
filetagdrv	文件点驱动		Windows CentOS

2.3 新建变量

单个变量配置

宝信基础自动化软件定义了丰富的变量类型，有模拟量输入AI、模拟量输出AO、数字量输入DI、数字量输出DO、文本变量TXT、二进制大对象BLOB、多数字量输入MDI、模拟计算量
CA、数字计算量CD、分级报警变量LA快速入门章节我们对AI，DI变量进行初步介绍，令读者有个初步印象，需要详细学习各类变量的配置请查看变量使用。

1.在左侧栏中—>SCADA—>SCADA名称下—>变量管理—>设备变量管理界面中的“新增”



2.选择了“AI”类型之后，下方的tab页面，变成了AI变量的属性页。

- AI：模拟量输入，提供对模拟量数据的读写访问，并允许设置报警；
- AO：模拟量输出，当操作员、程序块、脚本提供了一个数值的时候，向一个I/O驱动发送模拟量数据；
- BLOB：二进制大对象（binary large object）通常是一个大型的文件，例如一个图片或声音文件；
- CA：模拟计算量，使用最多七个其它的常数或数据块的值，执行数学计算。根据配置的表达式和系统中其他变量的当前值，计算出结果；
- DI：数字量输入，提供对数字量数据的读写访问，并允许设置报警；
- DO：数字量输出，当操作员、程序块、脚本提供了一个数值的时候，向一个I/O驱动发送数字量数据；
- LA：分级报警，可以表示最多16个类型的区间报警，每个报警可以设置单独的报警描述，支持报警时间死区和报警值死区特性；
- MDI：多态数字量输入，为来自一个I/O驱动的最多三个输入重组数字量数据，将输入组成一个原始数值，并允许你设置报警限。相对于DI变量只有0和1两个值，MDI的值可以为0到7，共8种值；
- TXT：文本，允许您对设备的文本信息进行读写操作。最大支持256个字符；
- CD：数字计算量，根据配置的表达式和系统中的其他变量的当前值计算出结果，其结果为DI量，当输出值小于等于e的-24次方时，输出为0，当输出值大于e的-24次方时，输出为1。

3.依次填写变量名称、描述、数据类型等，最后＂应用＂确认。

- **变量名称：**变量的名字，用于标识Tag点。因为变量可以用于iPlat系统中脚本的编写，因此，命名变量名称需根据一定的规则如下

◦设备点名的命名规则：

-名字不能为空，长度小于不等于64

-可包含中文、数字、英文字母、中文标点符号（只含-：。——()【】+=%@#）、英文标点符号(只含-:._()[]+=%@#)

-点名最后一个.后不能包含关键字F_或A_开头的字符串

◦设备点名的命名规则：

-可以包含中文、数字、英文字母或下划线“ ”

-名字不能为空，长度不超过64（含结尾“()”）

*-不能以数字或‘’*开头，不可全数字组成

-名称不能等于表达式里面的关键字
（
"in","max","min","sin","cos","tan","abs","asin","acos","atan","log","ceil","pow","sq rt","eq","neq","conret","strconcat","todecimal","int","exp","fmod"等）
-类的自定义属性不能以（F_或A_）开头
-类的自定义属性不能以数字或“ ”*开头，不可全数字组成*
-变量名称的第二部分不能以（F_或A）开头
-变量名称不能包含'.'

•**描述：**对变量的补充说明；

•**数据类型：**配置的变量所用的类型，DI变量不用选择，AI是模拟量，有浮点型、整型等各种类型需要选择；

•**所属驱动：**选择所新增的这个变量所要采集的数据，来自哪种驱动；

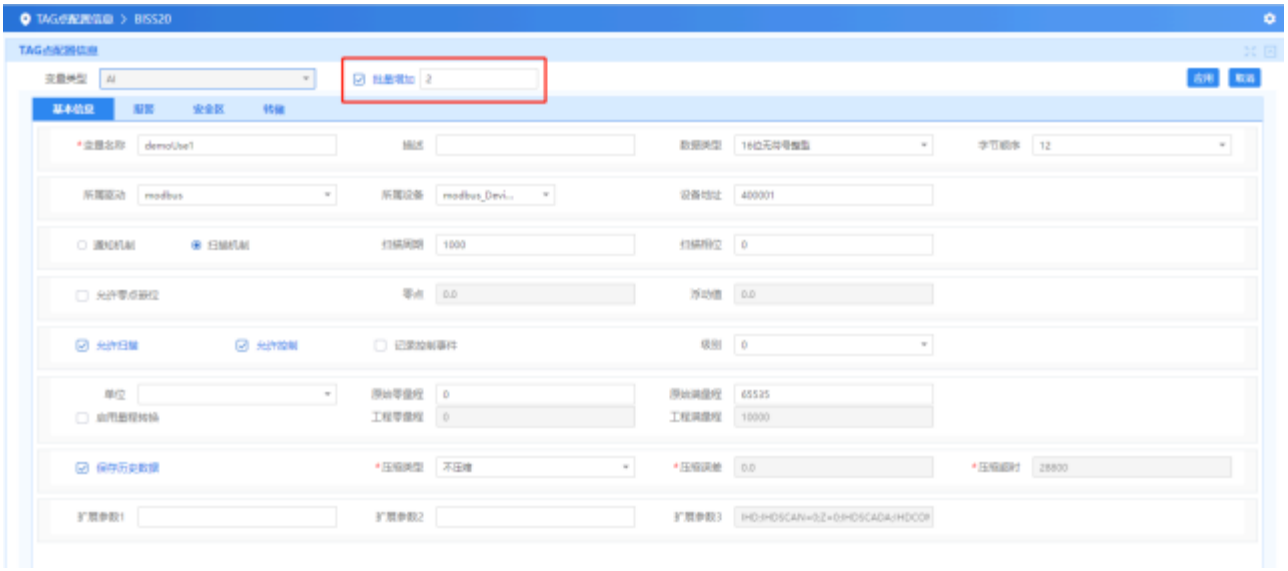
•**设备名称：**选择所新增的这个变量所要采集的驱动下对应的设备；

•**设备地址：**填写新增的这个变量的寄存器具体地址。如果选择了是驱动协议是免块方式的配置，请直接填写寄存器地址，如果选择的是驱动中配置数据块的方式，请按照“数据块：寄存器地址”的格式填写；

•**启动选项中的“允许控制”**：在启动选项这一栏中，可以选择是否允许控制这个变量，它的意思是：新增的变量，是否会将赋值给它的控制值下发下去，最后是否能成功控制设备中的值,还取决于控制的这个值，是单通道的还是双通道的，即，设备的值本身是否允许控制。

批量新增变量

1. 新增变量时可以通过配置基本属性页上面的批量新增信息进行所有变量的批量新增，在AI变量的配置页面上，勾选＂批量新增＂复选框，增加个数文本框变为可用状态。
2. 确定按＂应用＂。



模拟量的报警属性配置

1. 在上面新增变量的基础上，点击“基本信息”页右边的“报警”tab页，**打开报警属性的配置页面。勾选“启用报警处理”，即表示，这个变量会根据接下来的配置的报警条件，来产生报警。

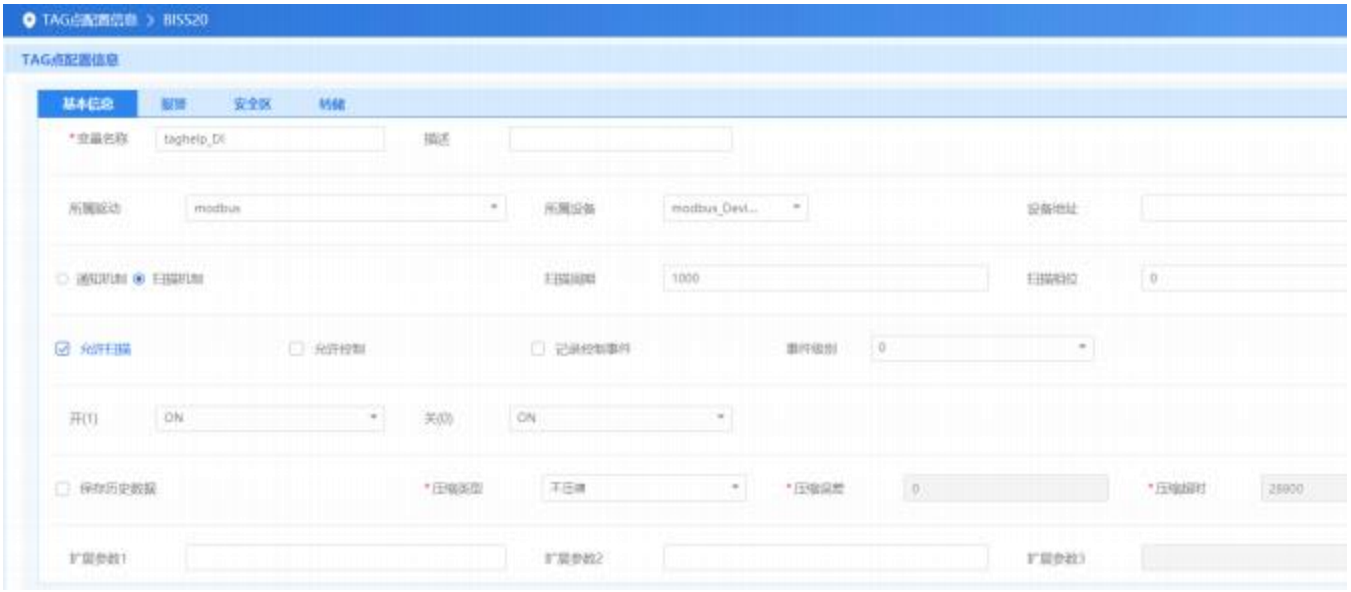
2. 依次设置变量的低低、低、高、高高报警值，在设置时，先勾选了报警级别，相应的报警值和报警级别就可以变为可编辑状态。

新增数字量的变量

1. 点击右上角的“新增”按钮，在“变量类型”的下拉框中，选择数字量DI。

2. 当点击了新增按钮之后，弹出的TAG点配置信息为白色的可编辑状态。可以配置的白色属性，在基本信息中，填写或勾选“变量名称”、“描述”、“数据类型”、“字节顺序”、“所属驱

动”、“所属设备”、“设备地址”等。详解同上述AI点属性。



数字量的报警属性配置

1.在上面新增了名为“taghelp_DI”变量的基础上，点击“基本”tab页右边的“报警”tab页，打开报警属性的配置页面。

2.选择**状态0报警和打开报警。而数字量的报警，分为四种，状态0报警、状态1报警、打开报警和关闭报警，用户可以根据自己的报警需要，进行选择。**

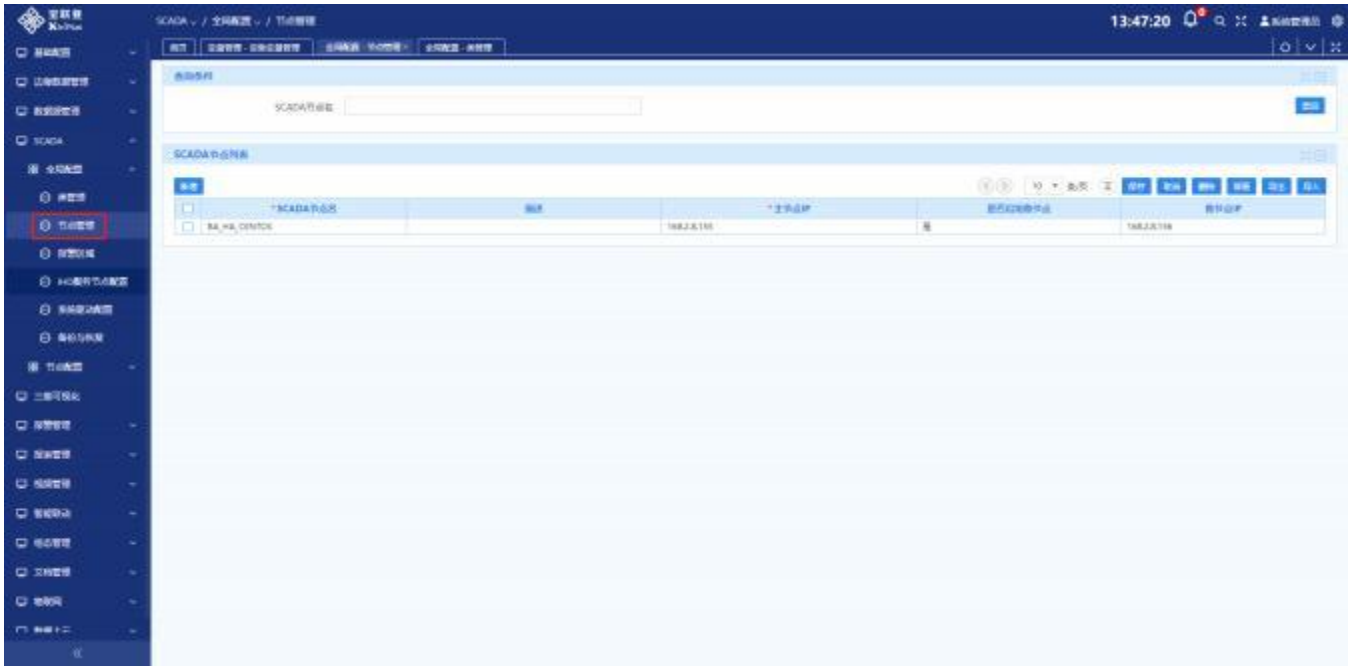
- **报警区域的选择**：双击报警区域设置的位置，打开报警区域选择框，选择系统中的已经配置好的报警区域
- **(报警、事件)级别**：0-255，共256个级别，数值越低，优先级越高
- **报警类型**：
  - **报警状态：状态报警分3种**：
    - **禁用状态报警**:选择此项，将不产生状态报警
    - **状态0报警**:当tag的数据为0的时候，产生“状态0报警”
    - **状态1报警**:当tag的数据为1的时候，产生“状态1报警”
  - **打开(0-1)报警**：当tag点的数据从0变为1的时候，产生报警
  - **关闭(1-0)报警**：当tag点的数据从1变为0的时候，产生报警

2.4 配置部署

完成了SCADA节点，驱动，变量的参数设置之后，我们要把所有的配置参数部署下发到SCADA节点中，后面才可以获得采集数据。

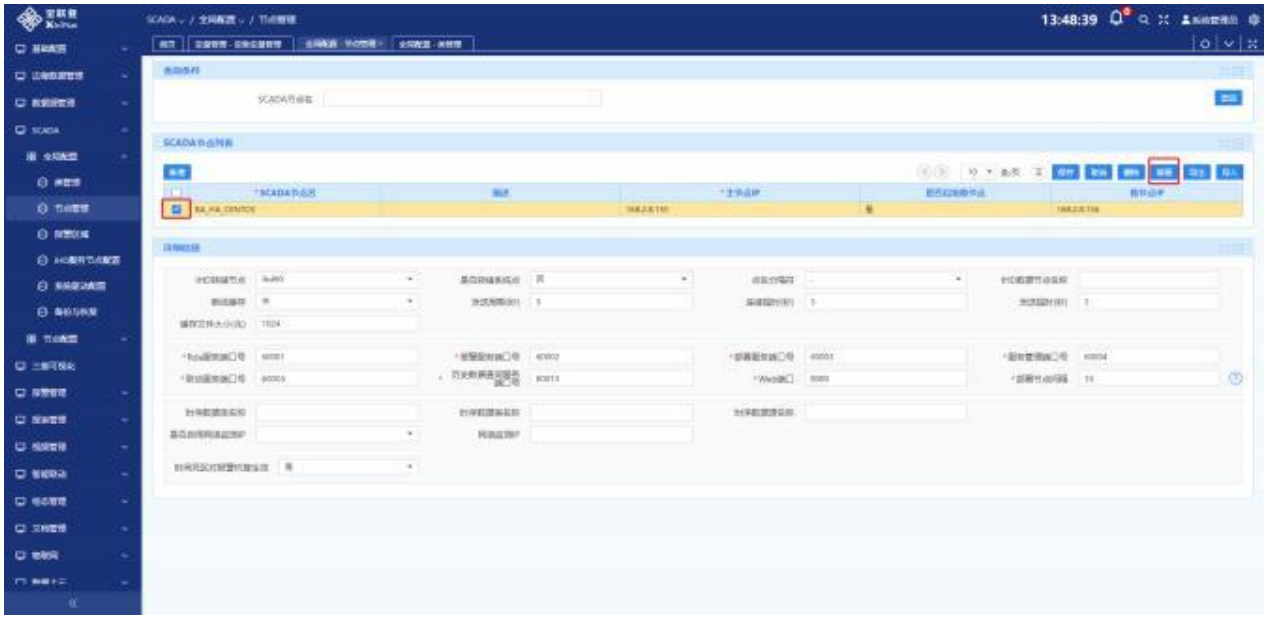
选择需要部署的节点

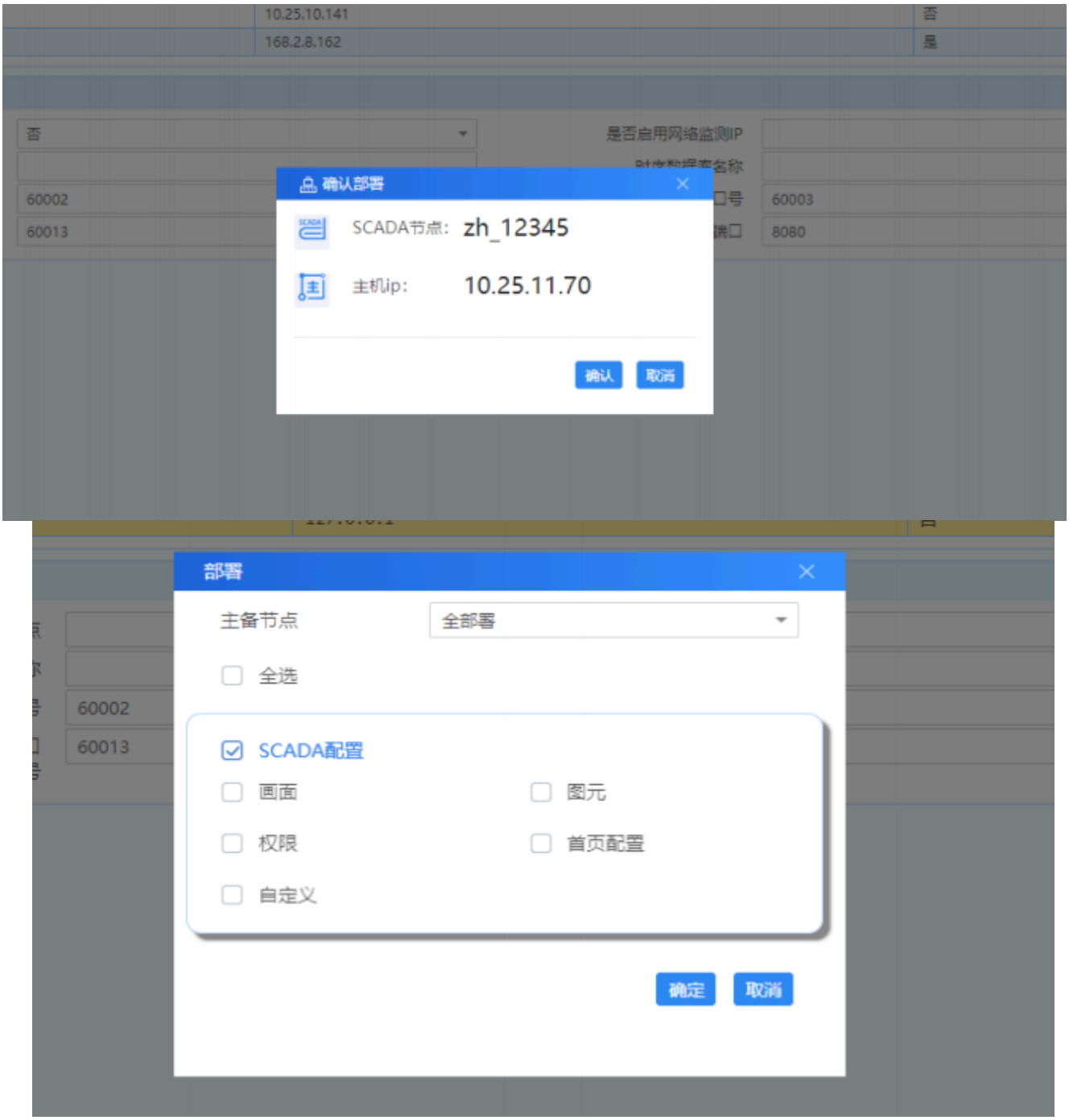
点击左侧主菜单中SCADA-全局配置，点击【节点管理】，进入节点管理界面。



执行部署

1. 部署的含义：用户通过iPlat浏览器界面中对节点、变量进行的配置信息其实是存储在数据库中，并没有发送到iPlat后台服务进程中，服务进程还无法获取具体的参数来提供服务；节点部署就是将前端配置好的配置信息发送到后台服务程序的过程。
2. 信息确定：如果是BC+BA的模式，需要确定当前部署BA节点是我们的目标节点。
3. 具体操作：点击选中需要部署的节点，点击右侧部署将会弹出部署画面选择框，勾选SCADA配置，点击确定。




10.25.10.141
否
168.2.8.162
是
否
是否启用网络监测IP
60002
60003
60013
8080
确认部署
SCADA节点：
zh_12345
主机ip：
10.25.11.70
确认
取消
部署
主备节点
全部署
全选
SCADA配置
画面
图元
权限
首页配置
自定义
确定
取消

4. **部署完成**。

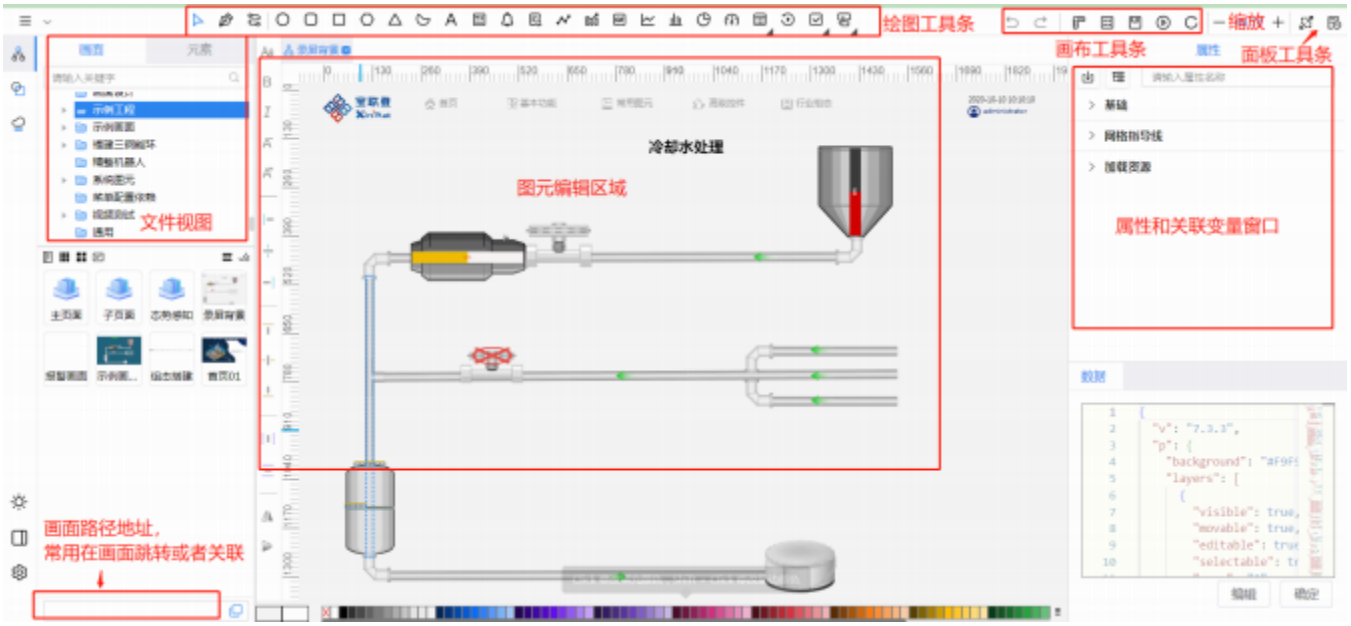
注意:

如果是BC+BA模式下部署，一定要保证BC和BA同步的数据库表结构相同，否则会有同步表错误导致部署失败

例如：在BC上修改了某个表的字段长度以满足配置需要，而BA对应的表没有修改，此时部署就会发生表同步错误

2.5 组态画面搭建

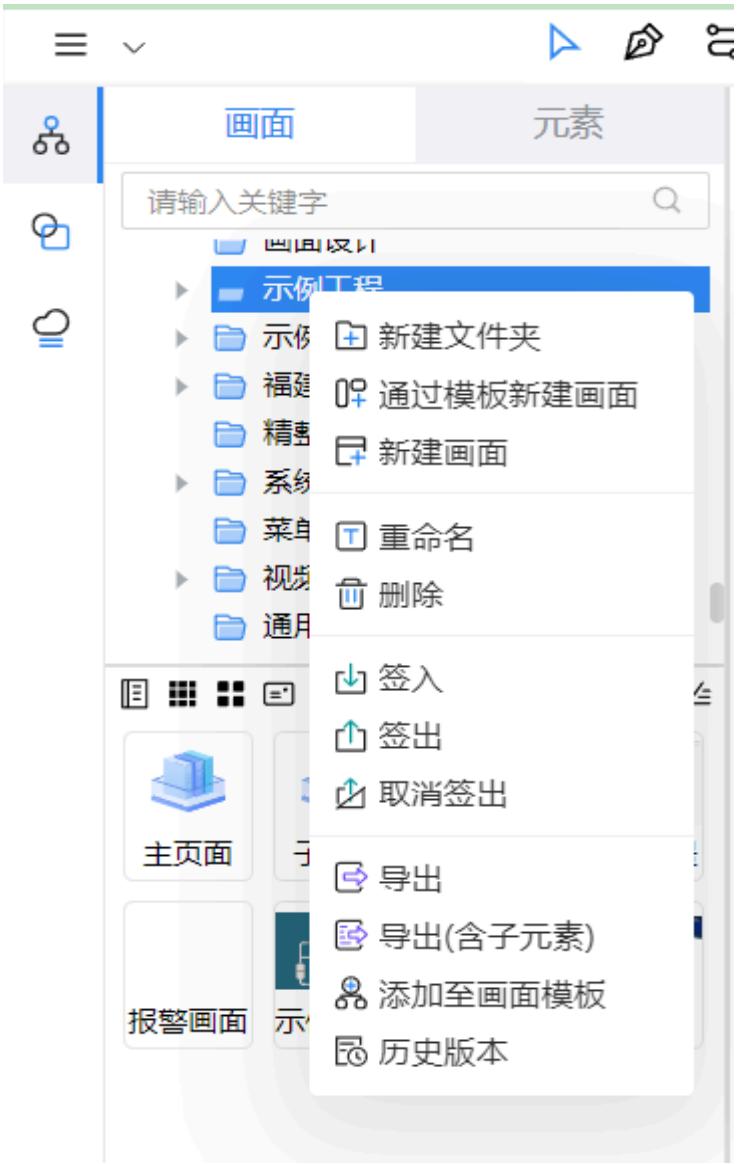
画面布局



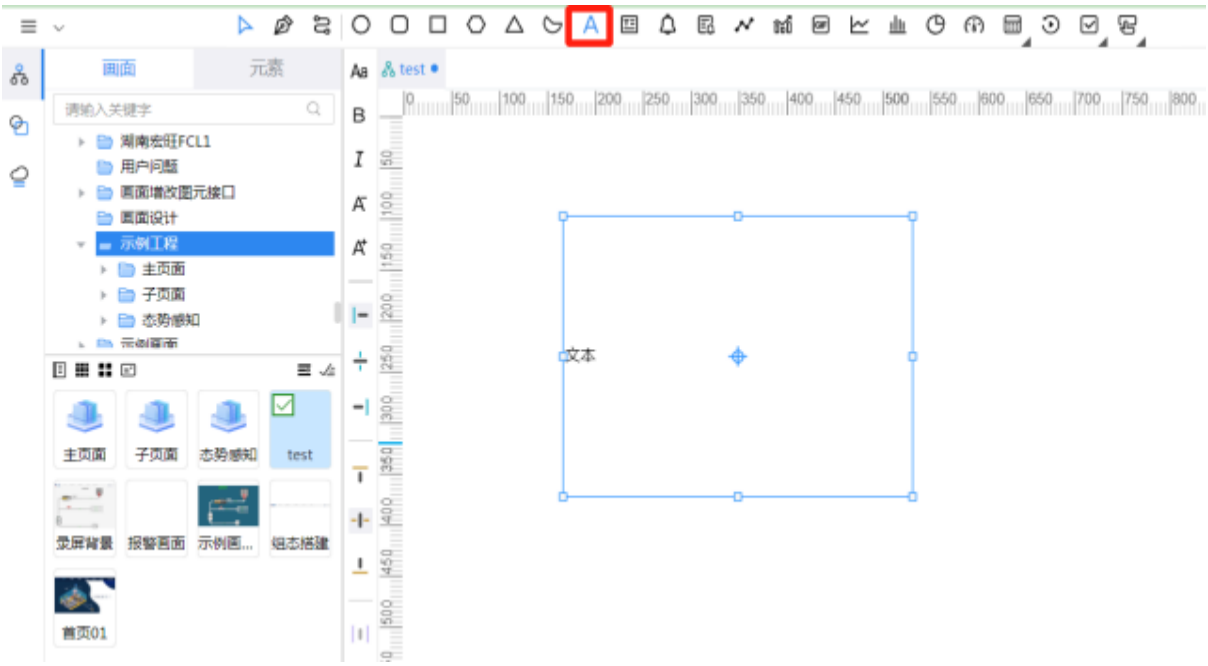
新建画面

1. 画面相关信息都是通过组态管理功能实现的，点击组态管理下的2D设计器。

2. 进入2D设计器界面，选择画面选项卡，然后选择一个目录，右键新建画面，也可以根据需要新建目录。



3. 从上侧导航框中选中文本放到画面编辑区域。



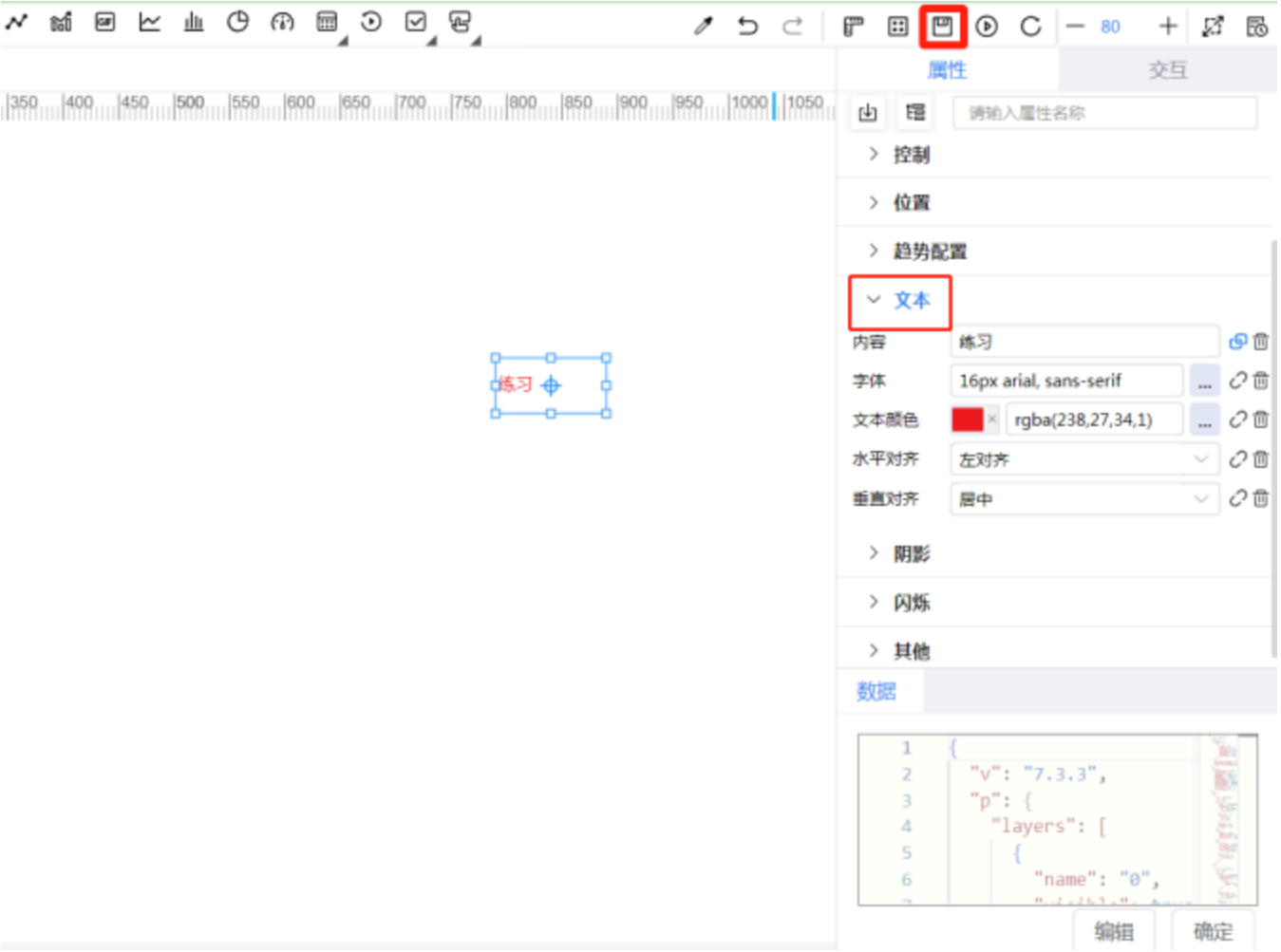
4. 在右侧属性框中将文本颜色设置为红色，内容改为练习，字体大小设置为16。

5. 绑点。

属性
交互
请输入属性名称
控制
位置
趋势配置
文本
内容
练习
字体
16px arial, sans-serif
文本颜色
rgba(238,27,34,1)
水平对齐
左对齐
垂直对齐
居中
阴影
闪烁
其他
数据
练习
数据绑定 属性 : text
SCADA
全局变量
本地变量
实时数据库
IOT
数据源选择
iplat_demo
所有变量
子系统
名称
设备名称
设备地址
demoUse1
modbus_Device0
400001
demoUse11
demoUse12
400011
demoUse13
400012
demoUse14
400013
demoUse15
400014
demoUse16
400015
demouse17
400016
demouse18
400017
demouse19
400018
当前显示1-20条，总共46条,总共3页
输入
iplat_demo.demoUse1
函数
确认
取消

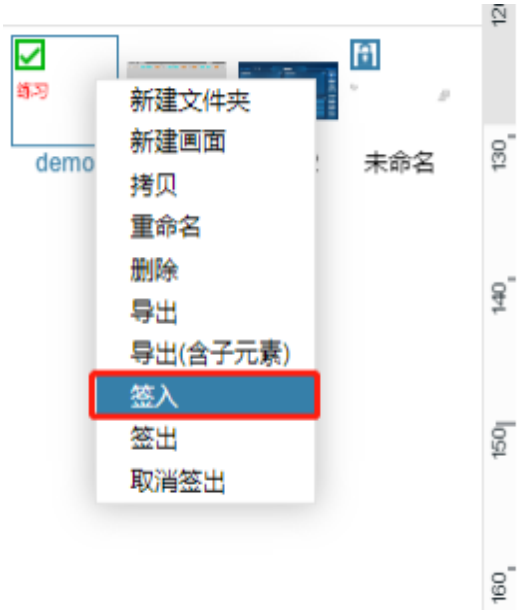
保存画面

1. 使用Ctrl+S或点击右上角保存按钮。



签入画面

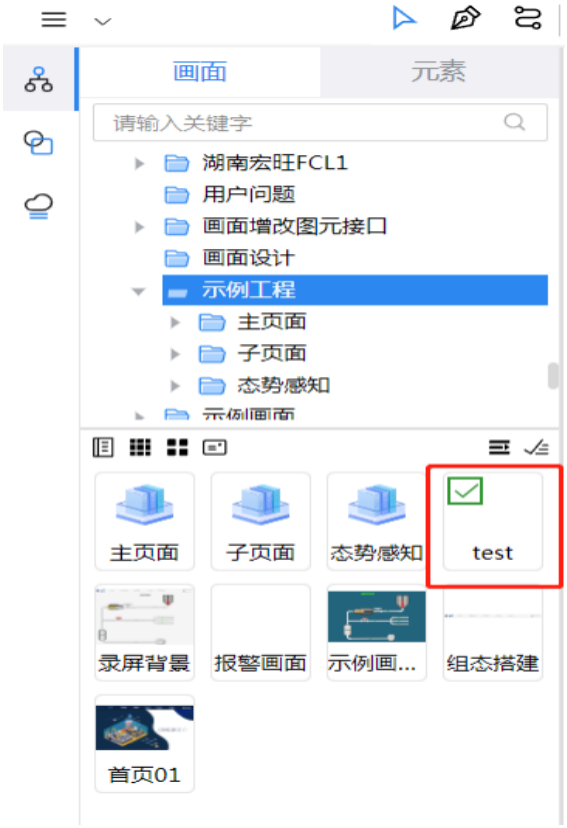
右键画面→【签入】。



2.6 组态画面运行

运行画面

1. 找到设计好的画面，先保存画面。



2. 绑点。

3.运行结果显示。

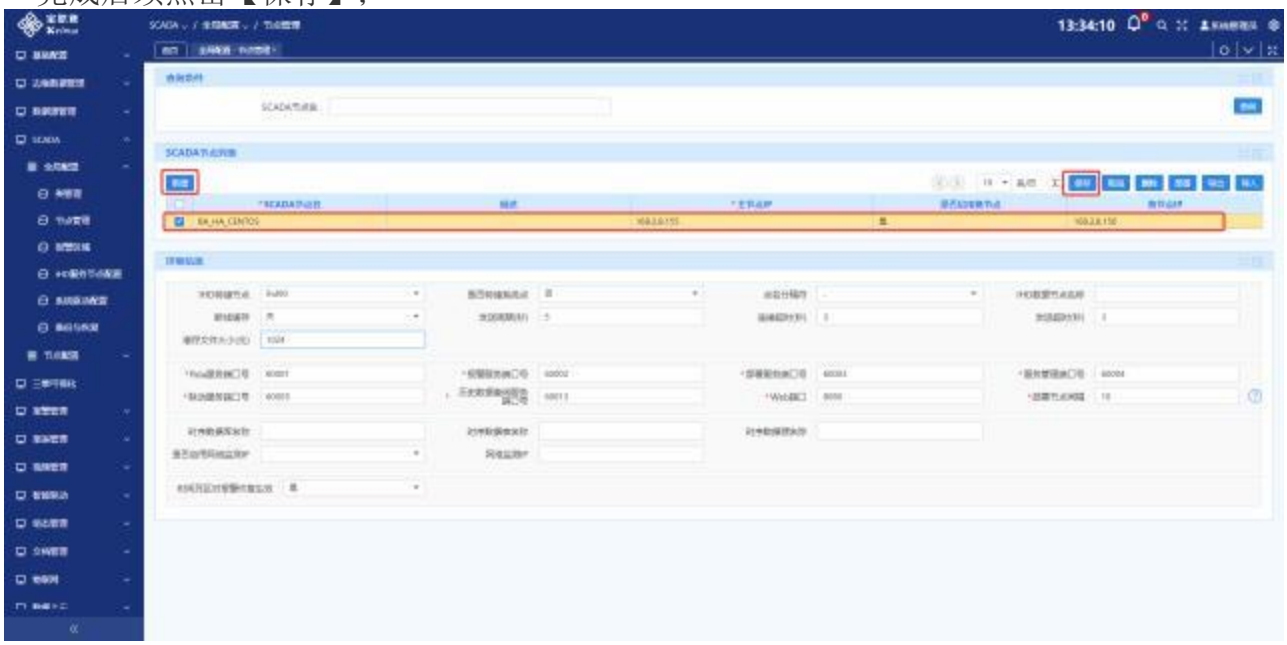
3. SCADA创建和部署

3.1 SCADA节点创建

步骤演示

1.打开谷歌浏览器，输入iPlat IP地址【IP:端口/iplat】端口号默认为8080，登录iPlat，找到页面左侧【SCADA】菜单，鼠标左键单击【全局配置】的【节点管理】，如下图

2. 鼠标左键单击边缘节点列表左侧的【新增】按钮，对如下图所示的配置项进行填写，填写完成后须点击【保存】；



•边缘节点名：必填项，由字母、数字、下划线组成，必须以字母开头，不可以E、PI、true、false等具有特殊含义的标识开头，不可随意修改。节点名变化会导致到组态画面及联动等功能中绑定的tag点失效

•描述：非必填项，大多用于节点业务用途描述

•主节点IP：必填项若无冗余节点，可直接填写节点的实际IP地址；若存在冗余节点，主节点IP与冗余节点IP必须填写服务器实际地址

•是否启用备用节点：若有冗余节点则选"是"

•备节点IP：非必填项，若存在冗余节点，则选择启用冗余节点，且备机的该边缘节点所有信息须与主机完全一致

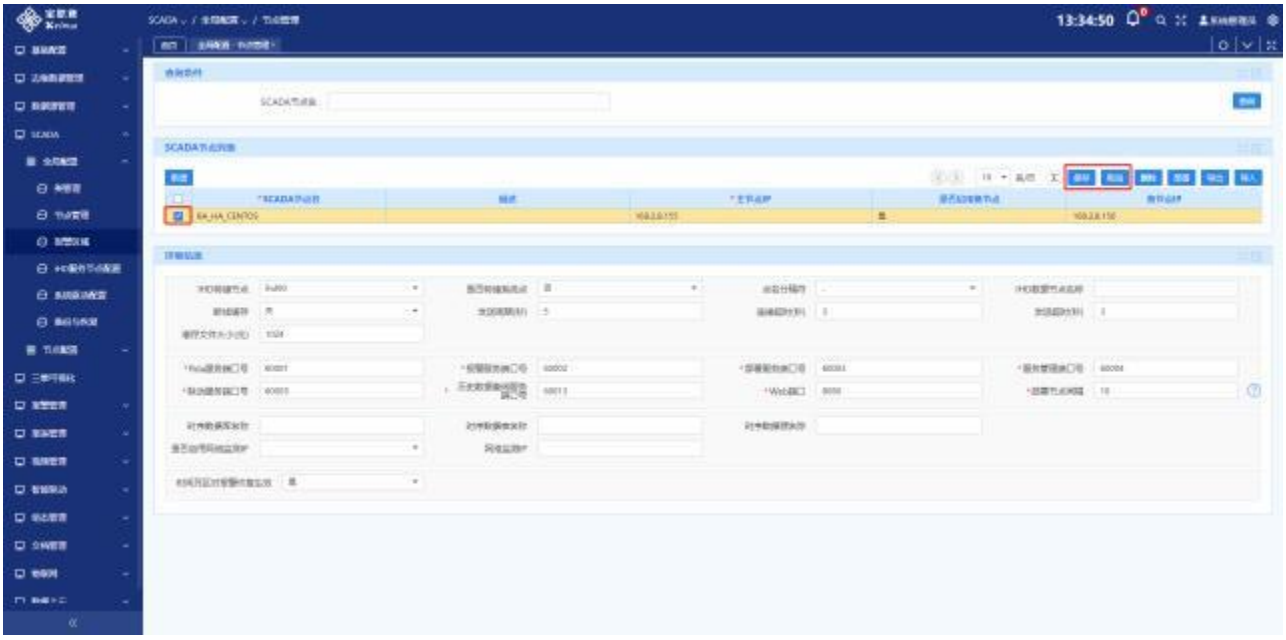
•详细信息：一般默认，可按照实际情况进行填写、修改

◦IHD转储节点：当【SCADA管理】=>【全局配置】=>【iHD服务节点配置】中配置了多个用于转储数据的iHD服务器节点时，可在此指定当前节点的目的节点

◦服务端口号：60001~60013为SCADA服务和iPlat的web服务用于通讯的端口号，若项目中因网络安全问题需要关闭端口，须保证iPlat-BC服务器与宝信基础自动化软件服务服务器的以上端口要双向开通，否则部署和数据推送会受影响

◦部署节点间隔：当SDADA节点是冗余架构时，点击部署，主节点先部署，完成后，等待该选项设定的时间（默认60秒），这段时间的SCADA功能由备节点支撑，主节点会在这个时间段内更新配置并完成重启，等待时间结束后，SCADA功能切换回主节点支撑，备节点再开始部署

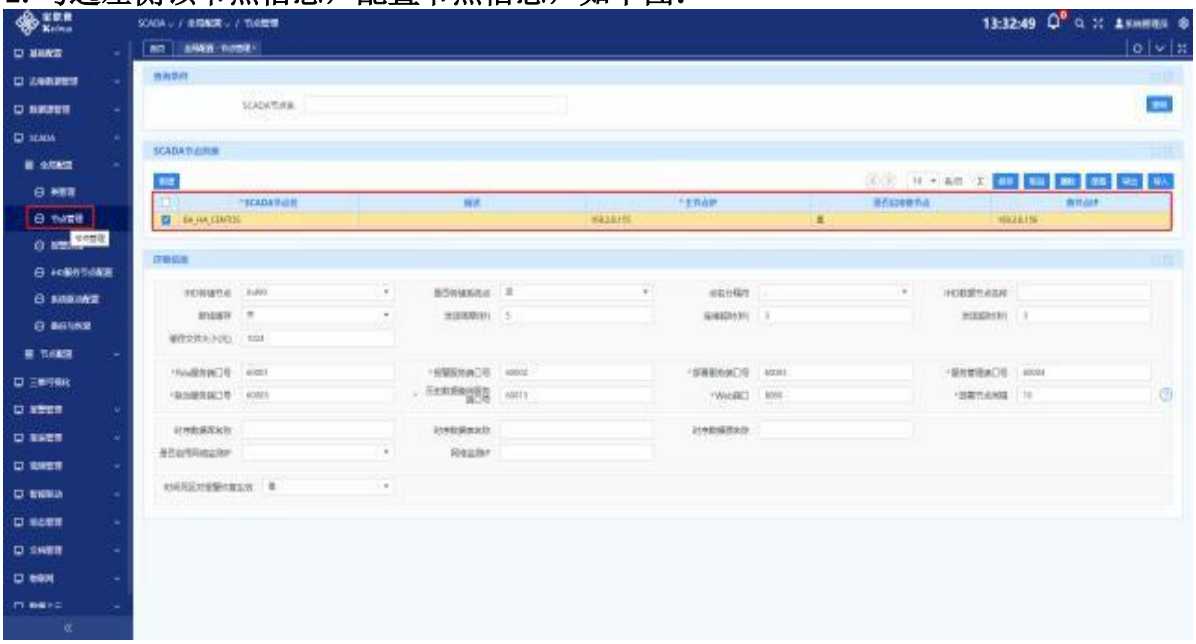
3.若修改节点，须先在左侧勾选该节点信息，修改完成后点击【保存】，如点击【取消】，则取消未保存的更改，如下图：



3.2SCADA节点配置

步骤演示

1.通过【SCADA】→【全局配置】→【节点管理】进入边缘节点配置界面。
2.勾选左侧该节点信息，配置节点信息，如下图：



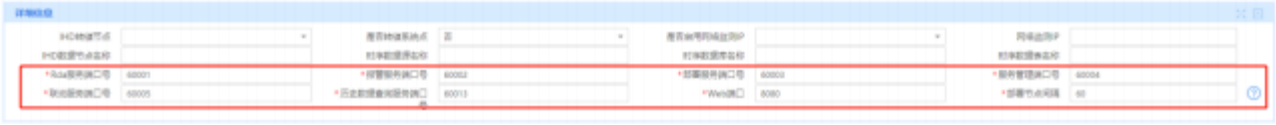
•BC模式下主节点IP和备节点IP不允许配置127.0.0.1或者本机实际IP

- SCADA节点名：必填项，由字母、数字、下划线组成，必须以字母开头，不可以E、PI、true、false等具有特殊含义的标识开头，不可随意修改。节点名变化会导致到组态画面及联动等功能中绑定的tag点失效
- 描述：非必填项，大多用于节点业务用途描述
- 主节点IP：必填项若无冗余节点，可直接填写127.0.0.1；若存在冗余节点，主节点IP与冗余节点IP必须填写服务器实际地址
- 是否启用备节点：非必填项，若存在冗余节点，则选择启用冗余节点，且备机的该边缘节点所有信息须与主机完全一致
- 备节点IP：默认端口号由系统自动填写，如非必要请勿更改

3. 配置节点详细信息。

- IHD转储节点：当【SCADA管理】=>【全局配置】=>【iHD服务节点配置】中配置了多个用于转储数据的iHD服务器节点时，可在此指定当前SCADA要转储的目的iHD
- IHD数据节点名称：当IHD转储节点是个多数据节点的分布式系统时，配置此项目可以将当前SCADA的数据都指定转储到某一个IHD的数据节点，数据节点名称为IHD的数据节点名称，如果不配置，则按照IHD默认的分配规则进行分配。对于单节点的IHD无需配置该项。
- 断线缓存：开启断线缓存，iHD服务关闭后，BA转储的数据暂存在BA服务器，待iHD服务恢复后，转储继续（注意：过程中不可关闭BA服务，BA服务关闭后则转储中断）
- 是否转储系统点：选择是否将SCADA的系统点转储到IHD节点中
- 是否启用网络监测IP：非必填项，可选择是或者否,当选择是，需填写网络监测IP
- 网络监测IP：此处IP为与节点IP同网段的第三方地址，用于检测当前节点IP服务器是否在线。对于不在线的服务器，经第三方仲裁，可将封降级为非活动节点
- 时序数据源名称：可在【数据源管理】>【时序型数据管理】查看可用数据源名称，选择数据需要转储到的时序型数据源，非必填项
- 时序数据库名称：填写需要转储到的数据源下的数据库名称，非必填项
- 时序数据表名称：填写需要转储到的数据库下的数据表名称，非必填项

- 时间死区对报警恢复生效：配置全局报警时间死区开关



- 服务端口号：60001~60013为SCADA服务和iPlat的web服务用于通讯的端口号，若项目中因网络安全问题需要关闭端口，须保证iPlat-BC服务器与宝信基础自动化软件服务服务器的以上端口要双向开通，否则部署和数据推送会受影响
- 部署节点间隔：当SDADA节点是冗余架构时，点击部署，主节点先部署，完成后，等待该选项设定的时间（默认60秒），这段时间的SCADA功能由备节点支撑，主节点会在这个时间段内更新配置并完成重启，等待时间结束后，SCADA功能切换回主节点支撑，备节点再开始部署

以上红框内的信息均为必填项，可默认填写

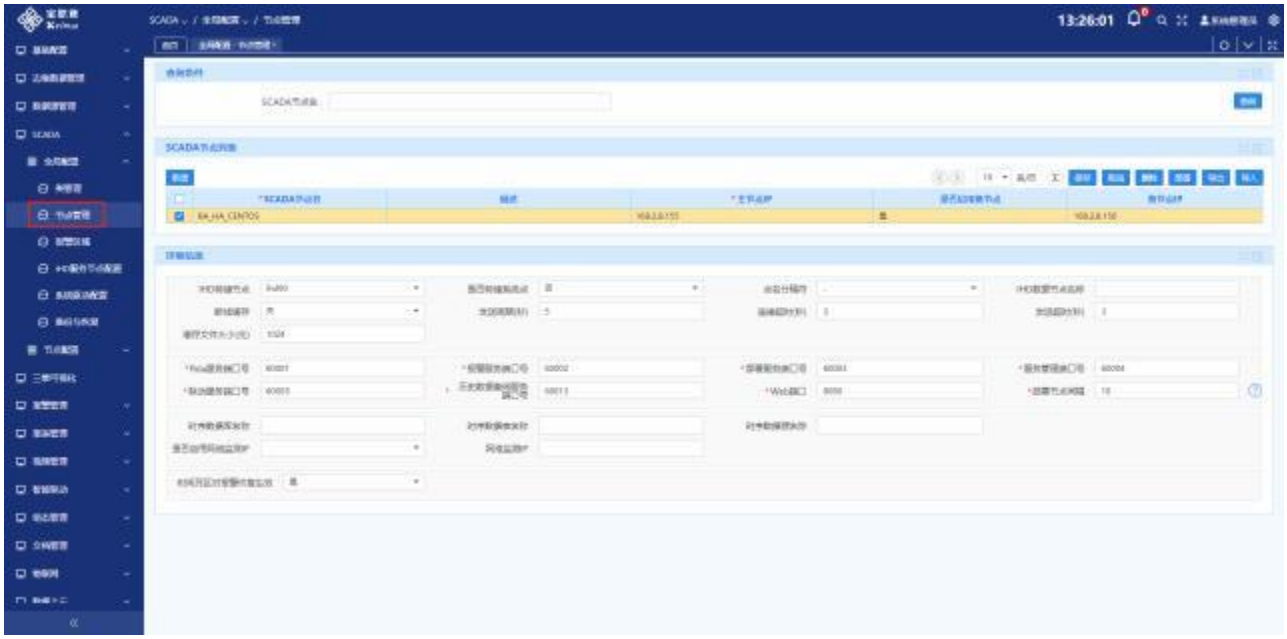
注意

SCADA管理中，对节点配置和全局配置的任何修改，必须部署后才能生效

3.3 SCADA节点部署

选择需要部署的节点

1.点击左侧主菜单中节点管理，进入节点管理界面。



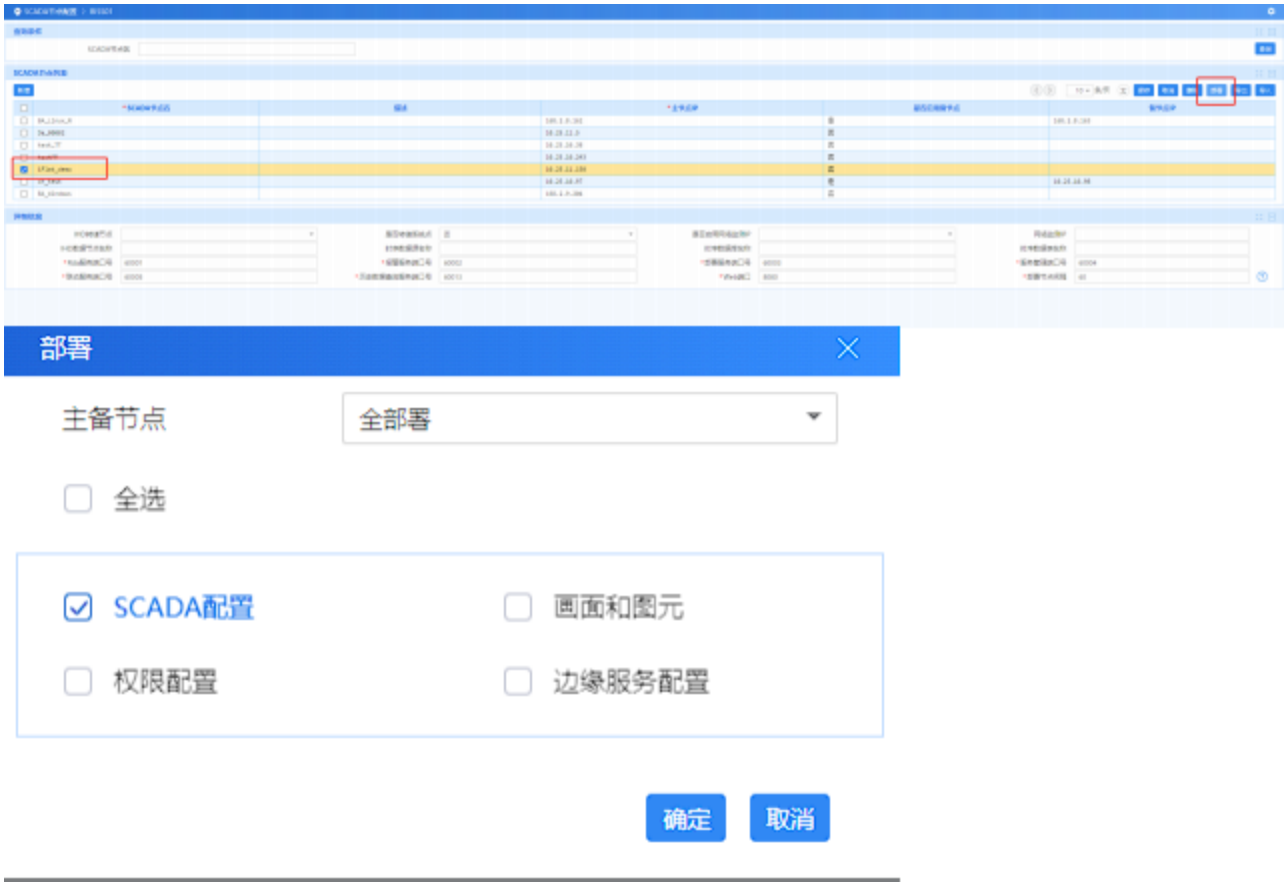
执行部署

1.部署的含义：

•用户通过iPlat浏览器界面中对一个节点下的变量、驱动、转储等功能进行的配置，仅是变更了iPlat数据库的配置，还没有同步到SCADA服务进程中，服务进程无法获取最新的、变更后的参数来提供服务

•节点部署就是将前端配置好的配置信息更新同步到后台SCADA服务的过程

2. 具体操作：点击选中需要部署的节点，点击右侧部署将会弹出部署画面选择框，勾选"SCADA配置"，点击部署。



-**SCADA配置：** 只部署主节点和冗余节点

- **画面和图元：** 部署了所选择的画面和图元，会同步画面资源所包括的表
- **权限配置：** 部署权限以及首页，把BC里选择的部署到BA中

> 注意
>
> BA只有部署SCADA配置，其他项部署都是 BC中的部署

3. **部署完成。**

- **部署结果判断逻辑：**
  - SCADA配置：如果生成表或者部署表，xml中有一个失败，就认为SCADA配置部署失败，后续SCADA配置的部署下发操作即停止执行，否则成功；
  - 画面和图元：中间有失败的画面、图元或者报警音频文件，显示部署完成，并显示成功X个，失败Y个；如果画面相关的权限表部署失败，则部署失败；

◦权限配置：14张表有一张失败，就认为失败，否则成功

BC+BA部署模式下部署说明

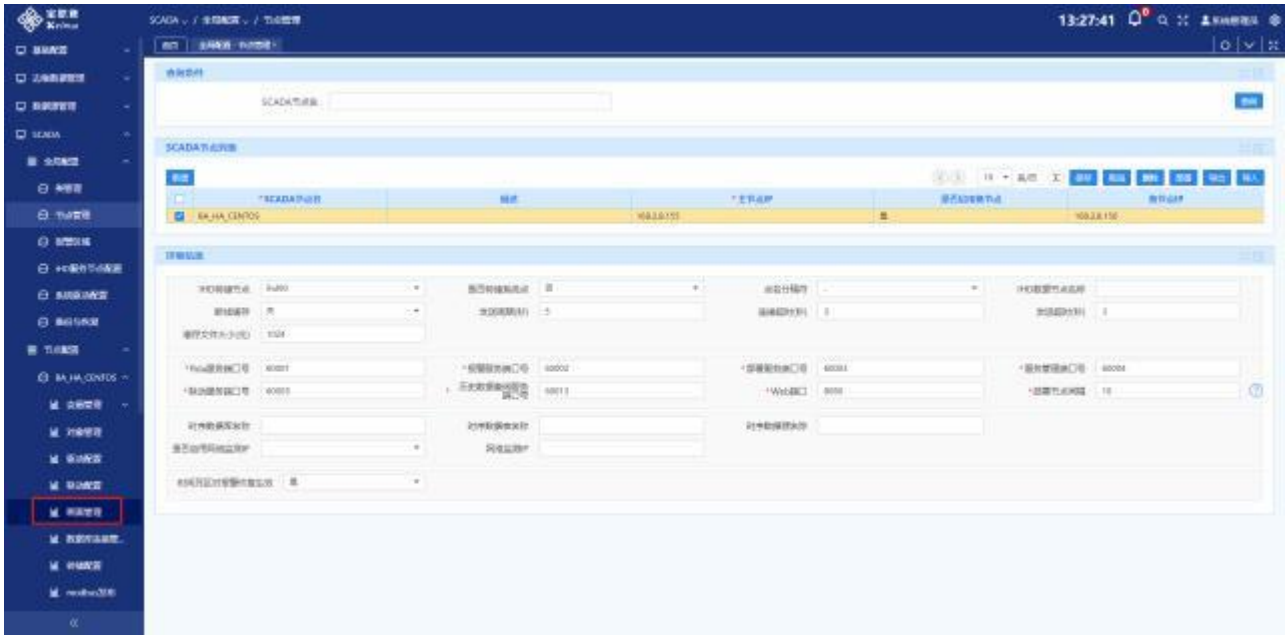
•高可用版本在进行节点配置时需要注意：

◦在高可用架构中，仅在iPlat-BC的组态管理中进行组态画面编辑，通过部署的方式更新到两台宝信基础自动化软件中，不要在宝信基础自动化软件中做任何组态画面修改

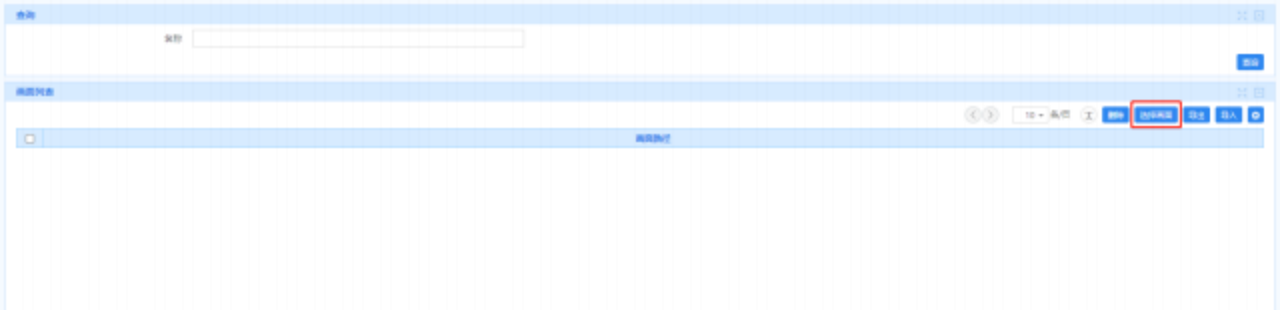
◦iPlat-BC部署前须保证EDCC03的项目配置中iplat.access.url配置项为iPlat-BC的地址（格式为http://IP:port/iplat更改后须重启web服务）

◦然后部署时需要首先在节点画面管理中选中需要部署的画面，最后在部署下发选择界面（如上图）进行部署

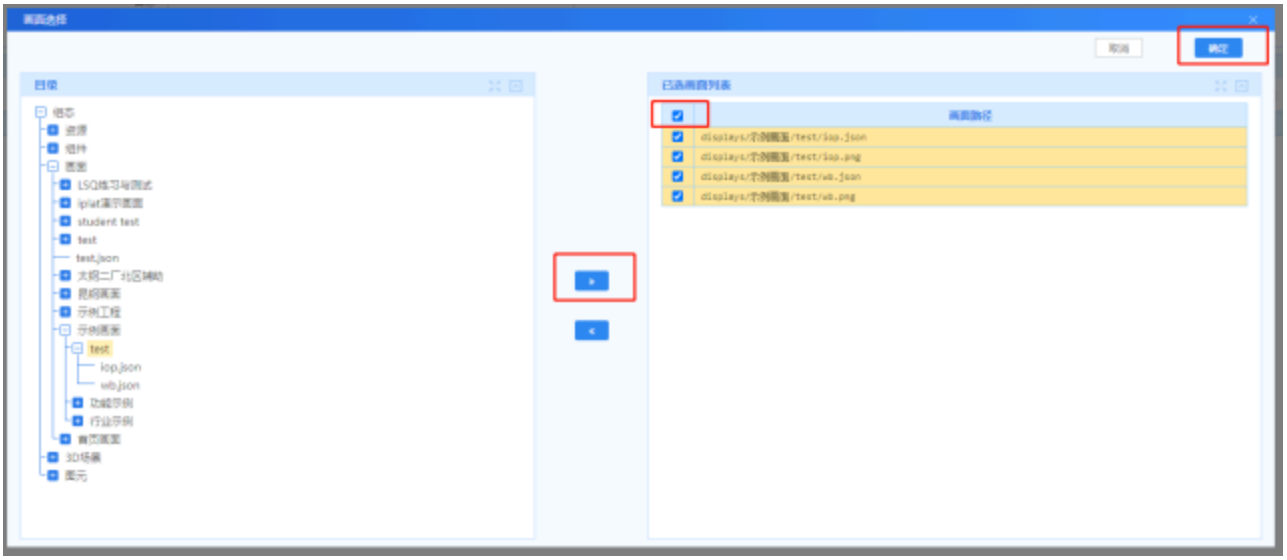
•节点画面管理简介：点击节点下拉箭头可以看到如下画面管理菜单。



•点击选择画面按钮



•弹出画面选择界面，在左侧选择需要的画面，画面会加入到右侧，然后在右侧框勾选需要下发的画面，点击确定



•部署成功之后，接下来可以运行画面了

3.4 SCADA节点配置导入导出

导入变量

1. 当节点中用到的采集、控制点数目较多时一般通过导入变量功能进行批量导入。
2. 具体操作，首先准备好导入的点表（后面介绍）选择需要导入变量的节点，点击右侧导入变量，平台会弹出导入模式选择框：
   - •增量导入：只导入新增部分，重复部分不导入
   - •全量导入：删除原有点表，导入新的点表
   - •模块增量导入：该模式下只更新excel中存在列的数据，不进行点的新增；必须要有固定列头"变量来源"、"变量名称前缀/对象名"、"变量名称后缀/自定义属性"；Bit按位报警不支持缺列导入。
   - •循环导入条数：程序单次处理的条数，可选100或者1000，按照条数导入，导入过程出错，可以根据导入的条数更快定位
   - •启用报警：勾选启用报警后，所有配置按位报警的点的报警都开启；

3. 选择导入方式后，在上传文件弹窗点击【选择】按钮，在弹出的文件资源管理器中选择后缀

为.xlsx的tag点表，导入的点表变量信息以Excel形式存放，导入进度到100%后，关闭弹窗，可前往该节点下的变量管理中查询验证导入的tag点信息。

4. **导入完成后需要重新部署，否则不会生效，部署部分详见SCADA节点部署。**

注意

导入时，若需要空白的Excel表格模板进行批量导入Tag点的操作，可先行配置一个Tag点，将其导出的Excel作为模板和示例

导出变量

选中需要导出变量的节点，然后点击右侧的导出变量，将会弹出导出变量选择框。

。选择变量配置则会导出变量，导出一个包含变量配置的excel文件

.完整配置，会将全部变量的列导出

.模块配置：下面的四个模块可以勾选，勾选任一个或者多个，按照勾选的导出tag属性到excel中

.如果只勾选了历史数据模块，则导出的tag属性有，变量来源，变量名称前缀/对象名，变量名称后缀/自定义属性，是否保存历史数据，压缩类型，压缩误差，压缩超时；

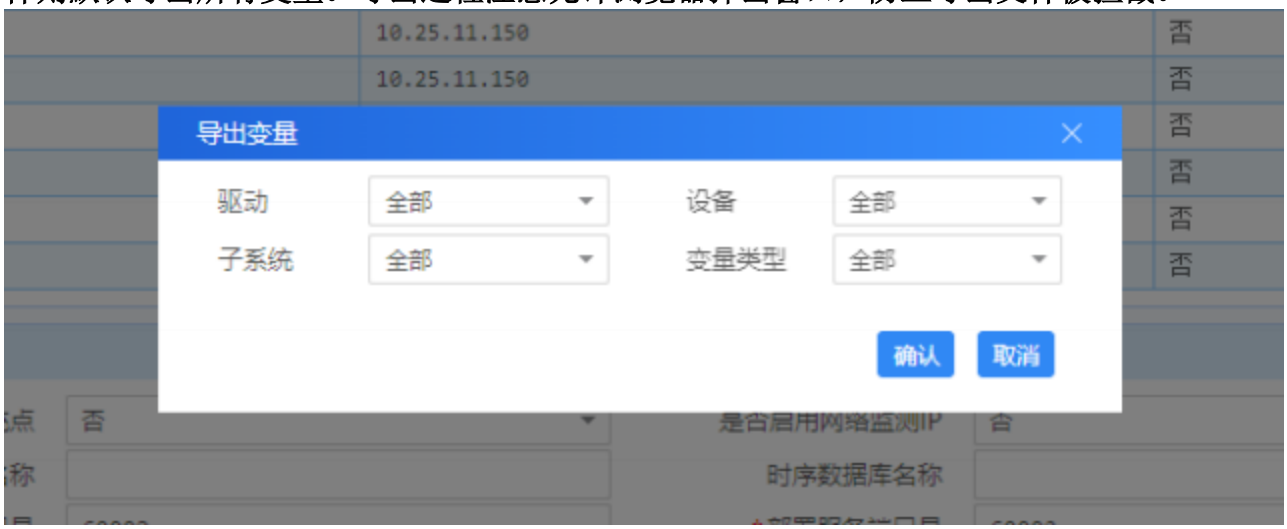
.如果勾选了基础报警，根据不同的tag点类型，报警页面的所有字段都会导出，不包括AI点的按位报警；

.如果勾选了转储，转储页面的所有配置信息导出，在excel中是一个字段；

.如果勾选了按位报警，按位报警信息导出在独立的sheet也中；

。勾选了其他配置，则会导出包含该scada其他配置(驱动、联动、画面、数据库连接、转储和全局配置)的zip包文

•通过该选择框，可以根据驱动、设备、子系统、变量类型对变量进行筛选。如果不选择筛选条件则默认导出所有变量。导出过程注意允许浏览器弹出窗口，防止导出文件被拦截。



。**驱动：**可以根据驱动类型过滤需要导出的变量点表

。**设备：**可以根据驱动下的设备进行过滤，导出所需要的变量点表

。**子系统：**当配置了对象点时，可选择对应的对象子系统，将对应的对象子系统下的点全部导出

。**变量类型：**可根据变量类型，如只导出AI或DI类型数据，就可以通过变量类型过滤，进行变量点表导出

节点状态页

点击左侧菜单进入"节点状态"页面。节点状态页面具备如下功能

（1）显示所有节点的列表

BC模式下包括：本机（BC），BA，iHD。BA模式下包括：本机（BA），一个SCADA。

（2）活动状态

包括活动，非活动，离线3中状态

（3）节点状态告警

在告警配置页面可配置CPU，内存，磁盘的超限告警。超限之后表格背景标红。

（4）详情列按钮

点击可产科异常信息提示。

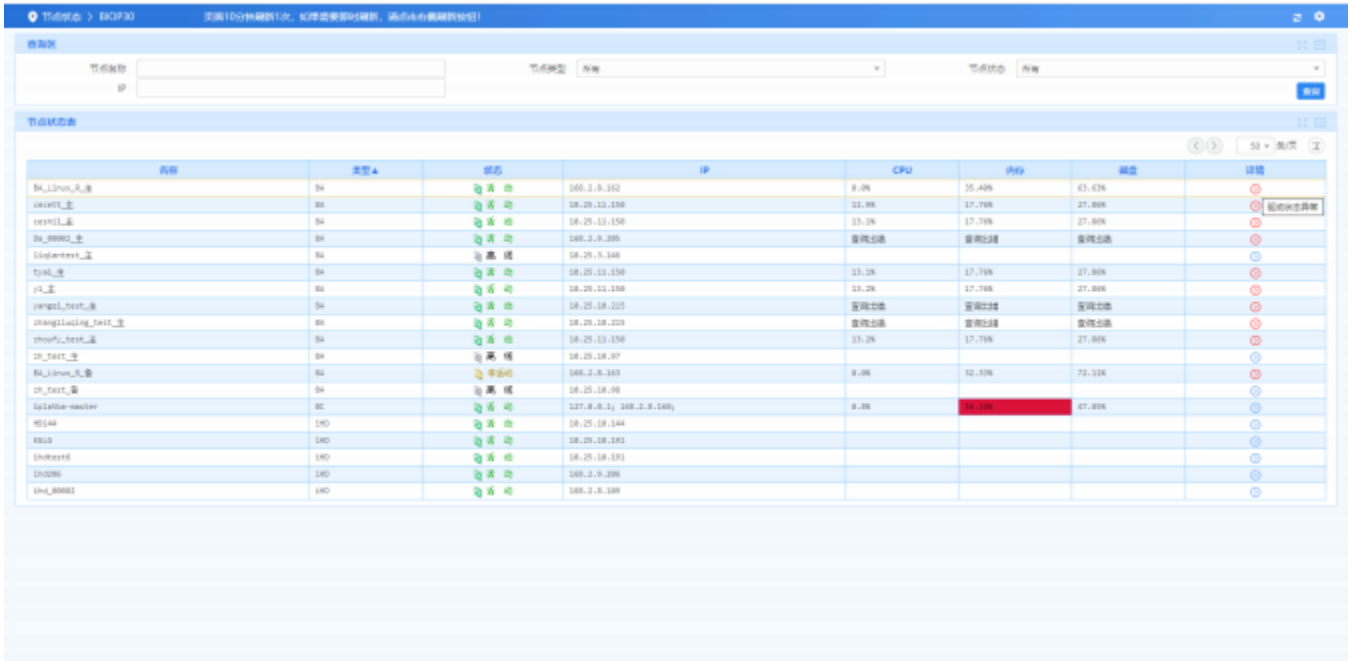
异常检测包含：da进程停止，驱动停止，设备停止；

异常显示：详情按钮为蓝色表示无异常，为红色表示有异常。鼠标悬浮在按钮上会有tooltip弹出提示异常项。异常只显示一个，优先级顺序是：（1）进程（2）驱动（3）设备

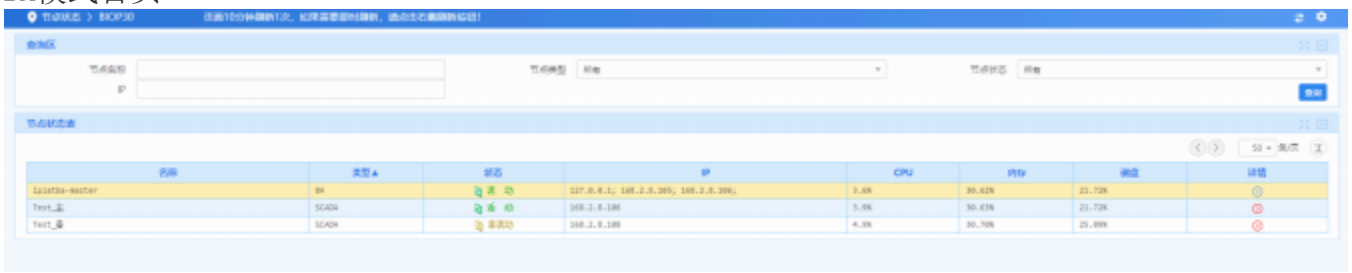
（5）定时刷新

保持页面打开，页面会定时10分钟刷新一次。

BC模式首页



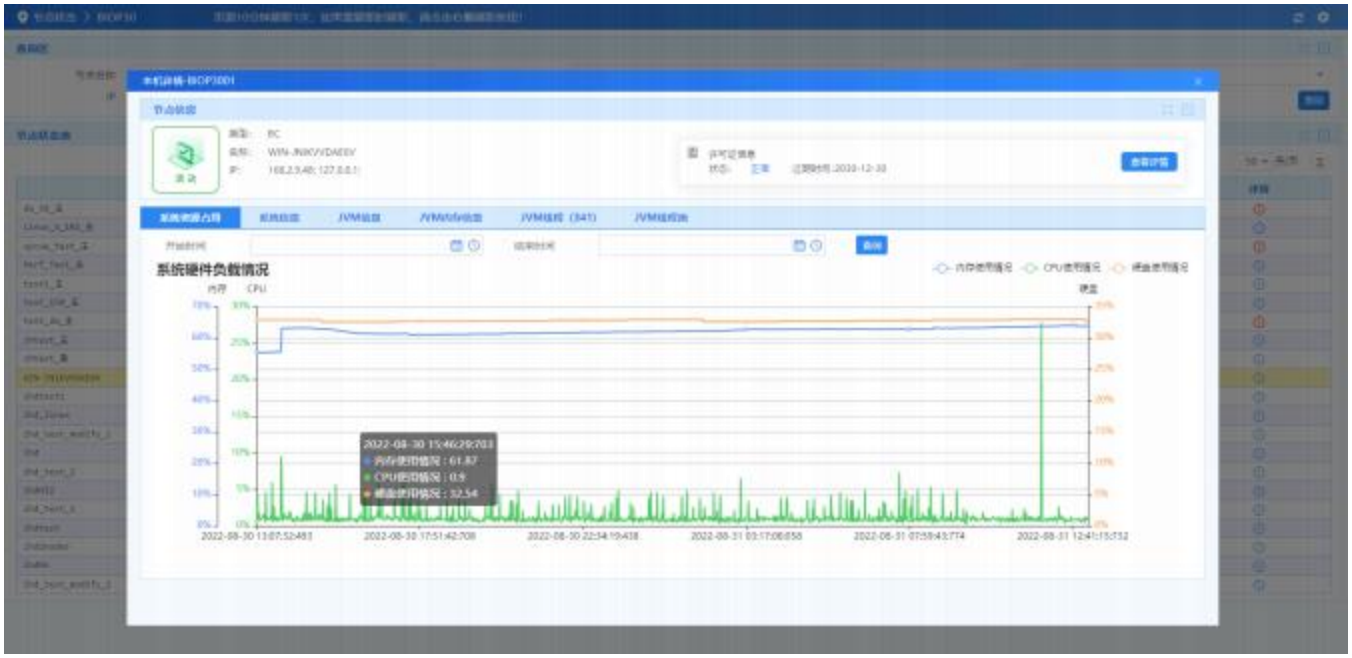
BA模式首页



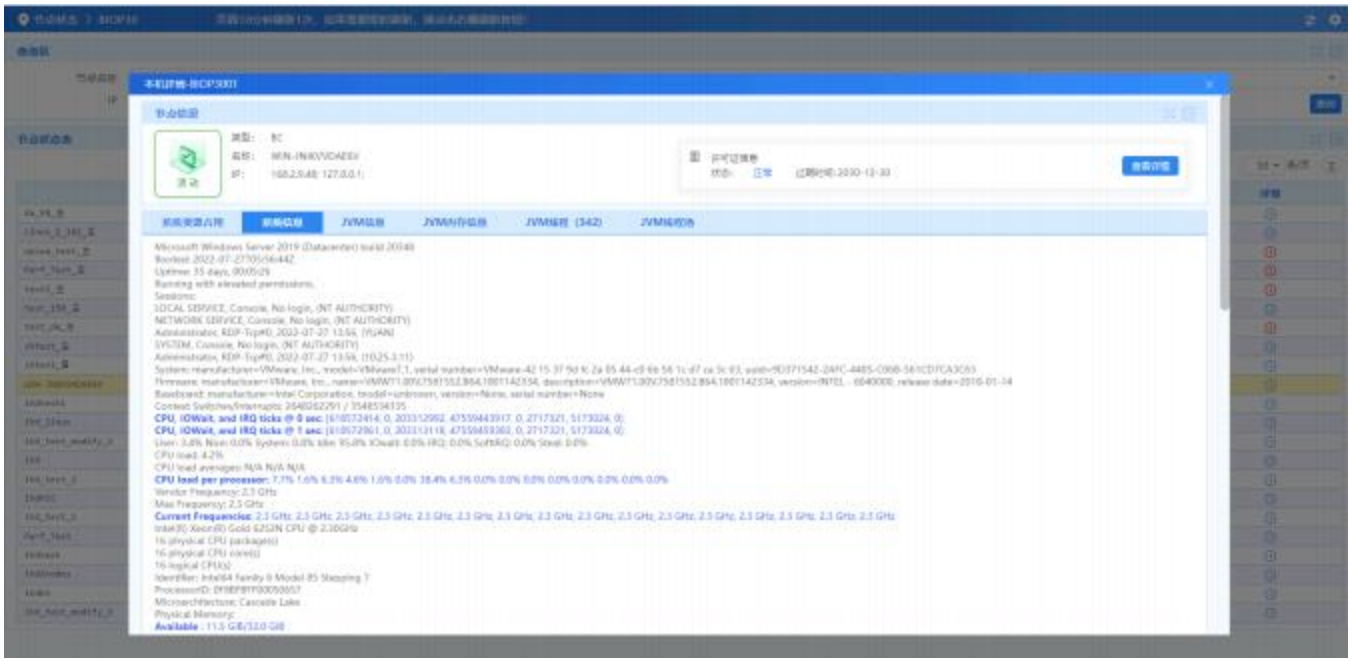
本机详情页

bc模式下查看本机详情页

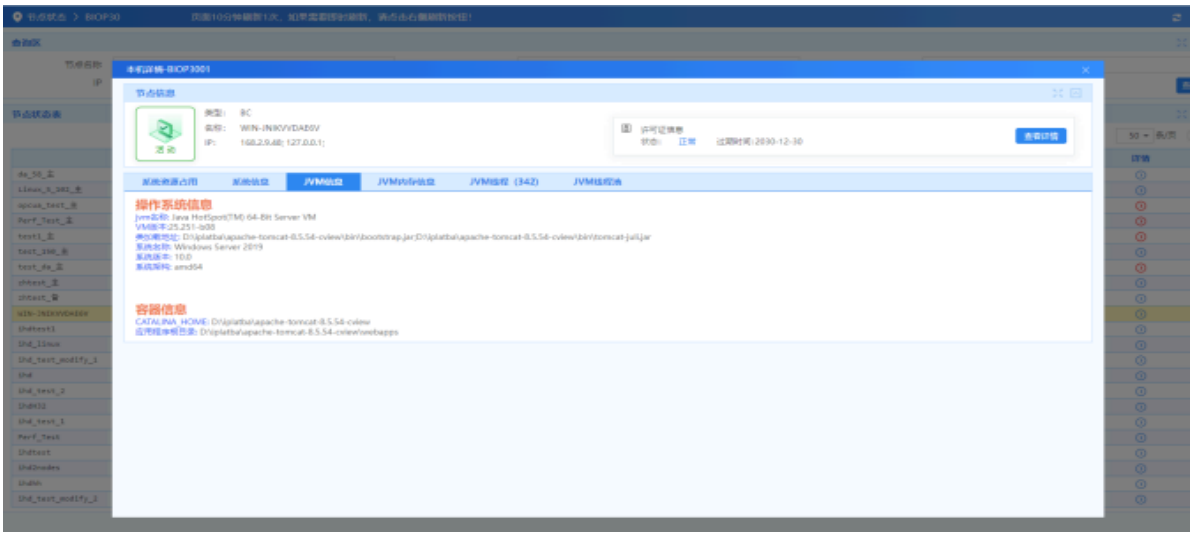
点击节点状态的详情按钮进入本机详情页，系统资源占用默认获取最新1天的数据绘制曲线显示



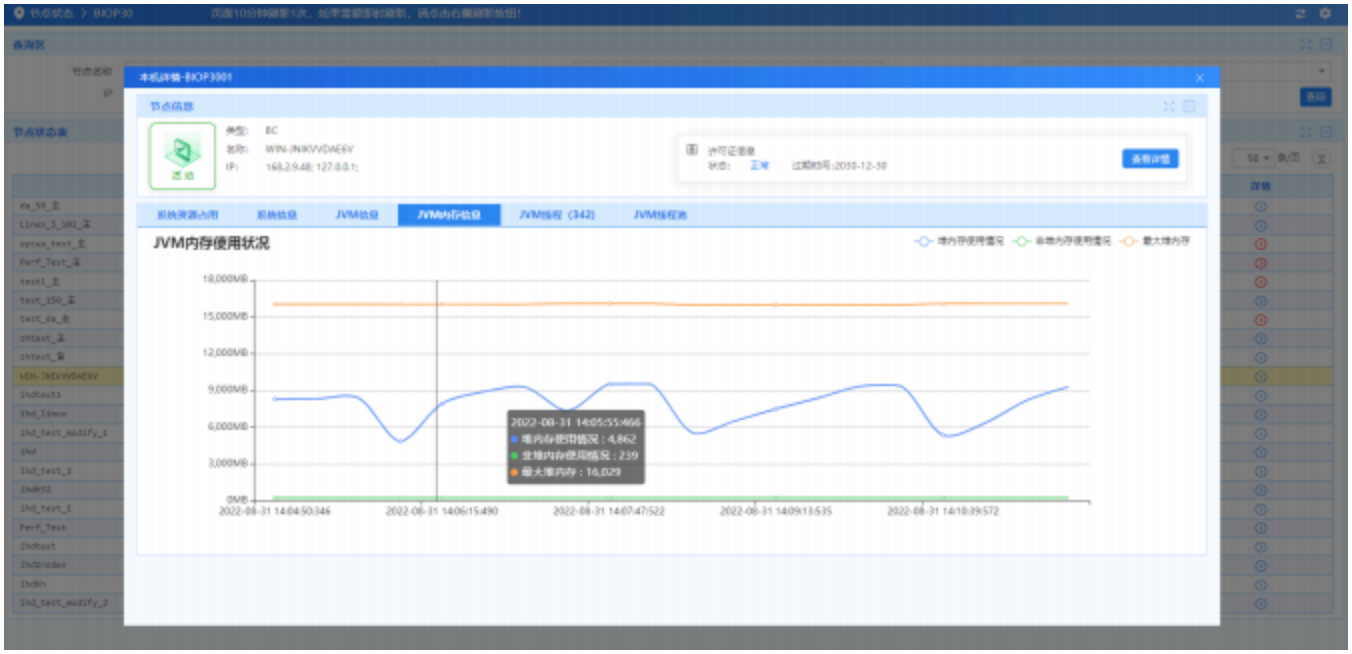
系统信息选项



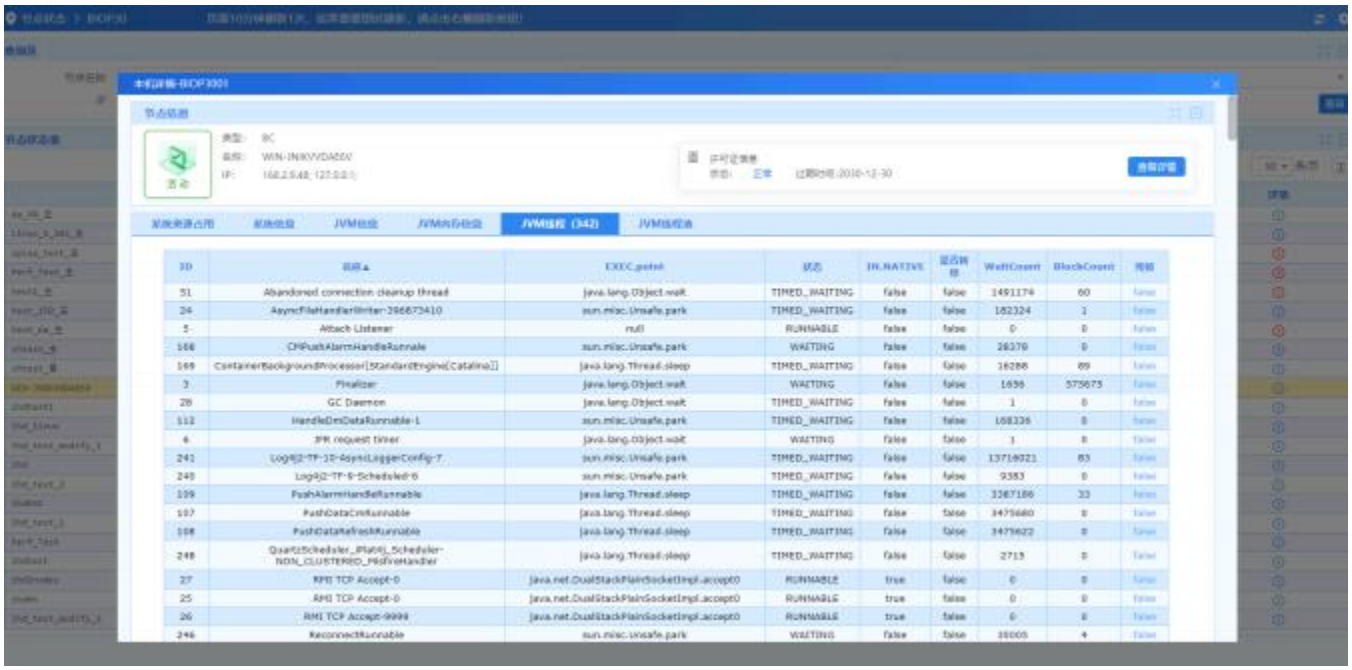
JVM信息



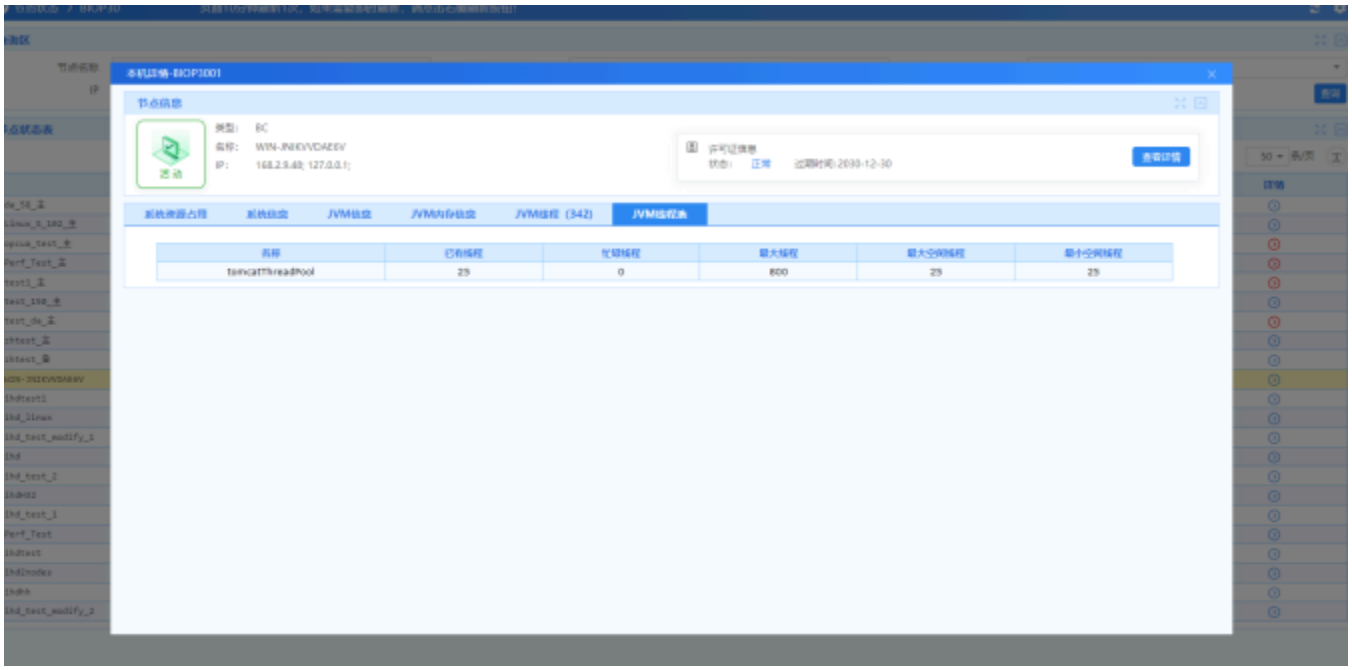
JVM内存信息



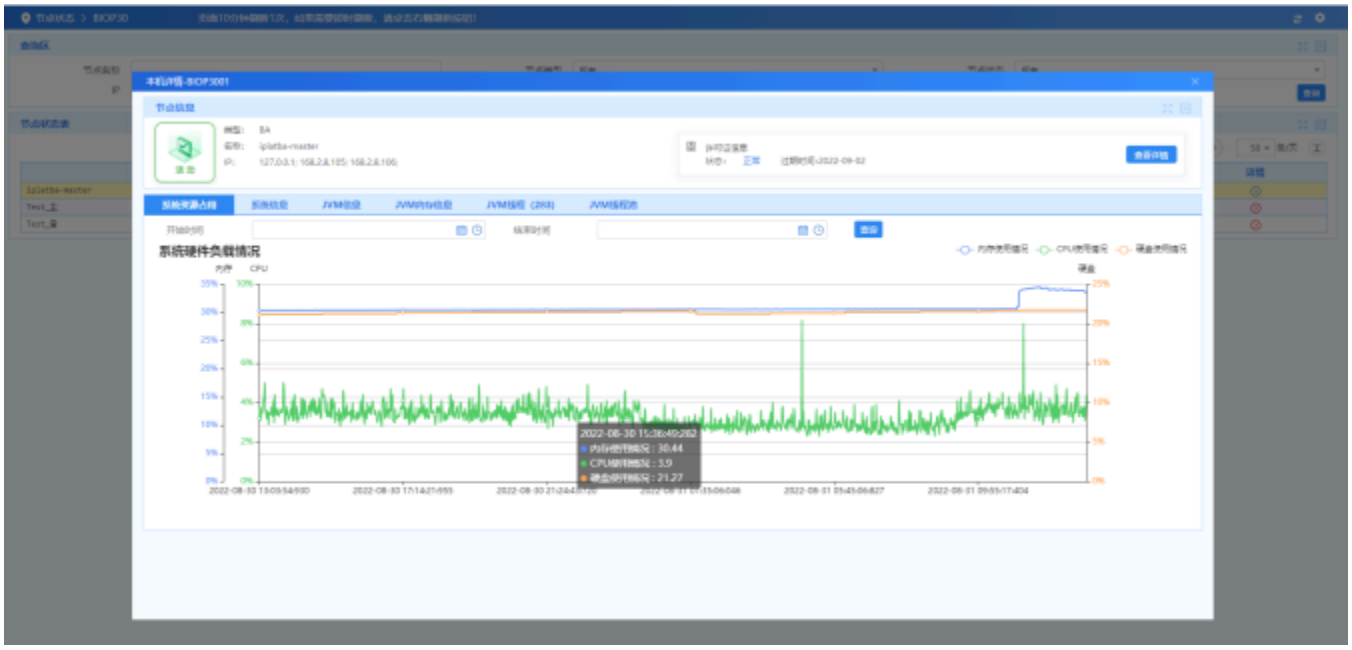
JVM线程，括号中显示的是线程个数



JVM线程



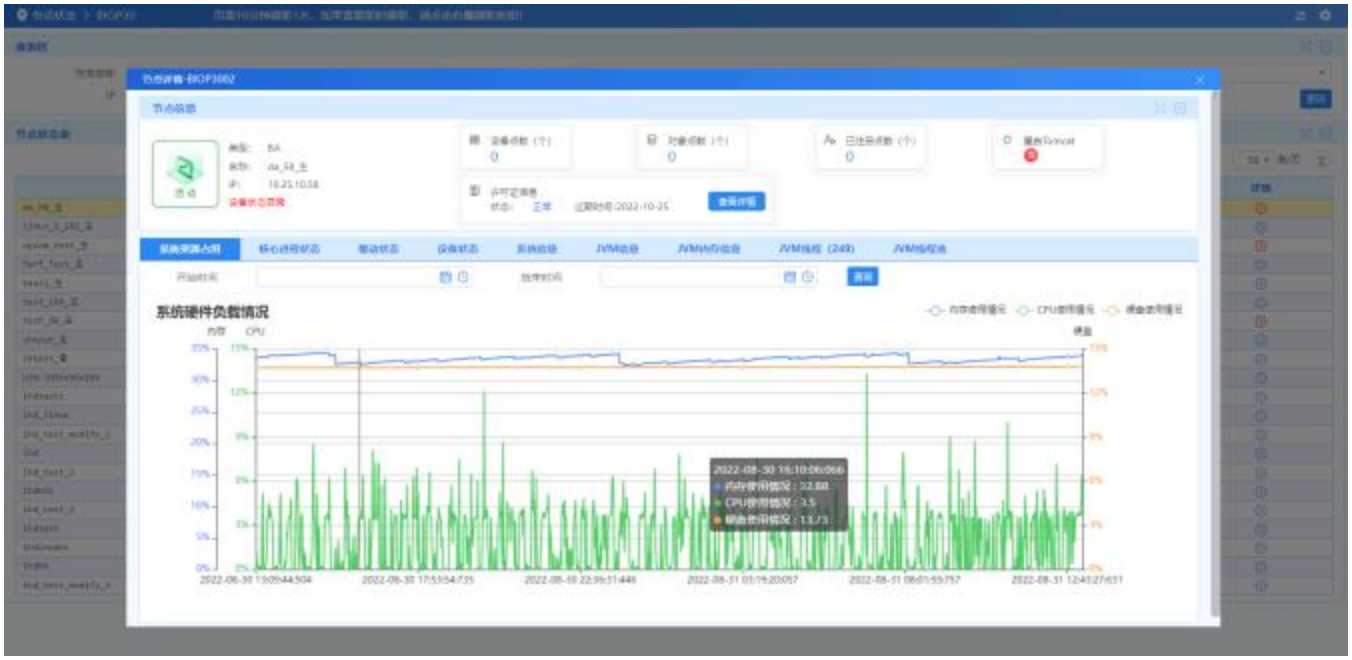
ba模式下查看本机详情



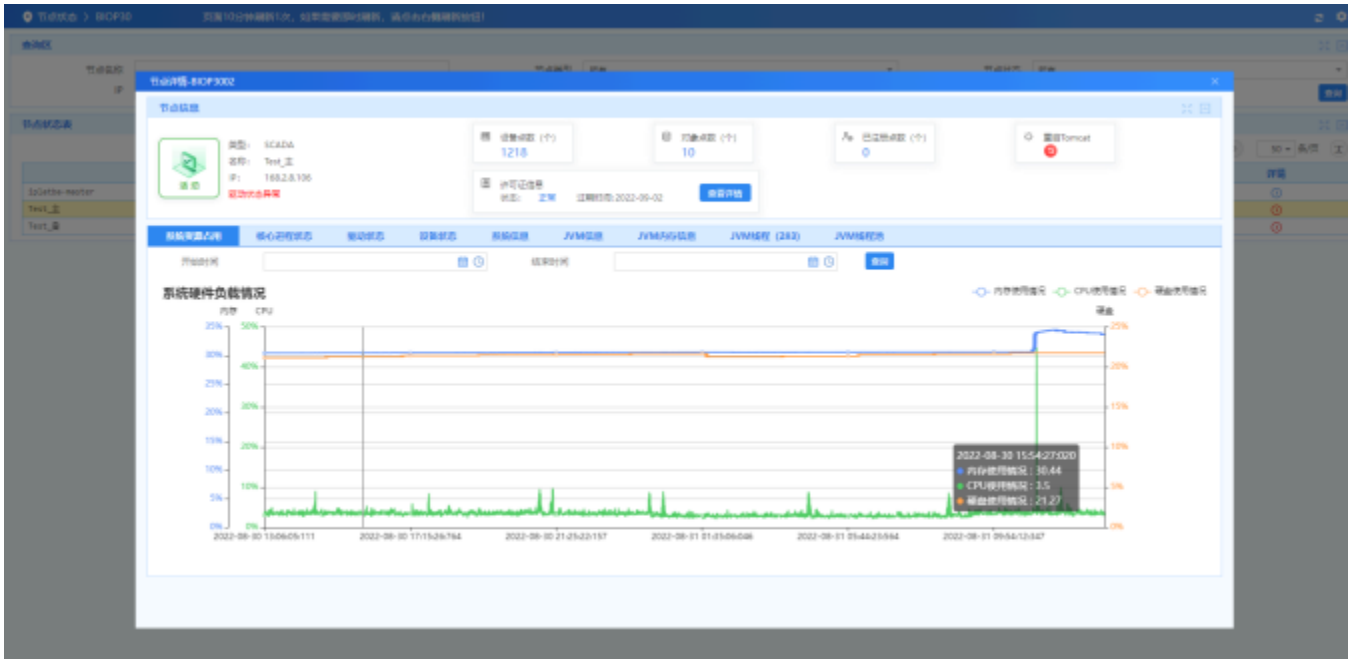
BA详情

bc模式下查看BA节点的详情

重启tomcat功能仅支持linux环境，在windows环境下不支持。



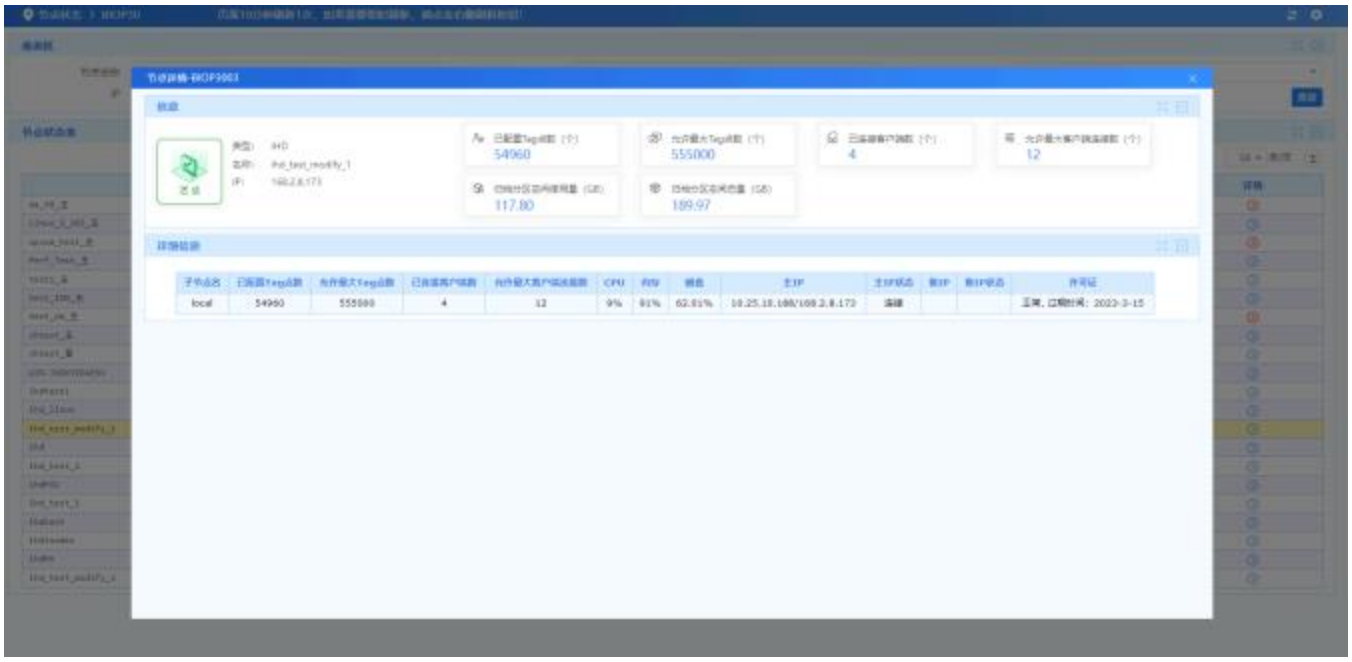
BA模式下查看SCADA的详情



iHD详情

bc模式看ihd

dm通过pubserver与ihd建立连接，因此节点状态页配置的IP是pubserver的ip，真正的ihd节点是在详情页面。因此ihd的参数都是在详情页面来显示。

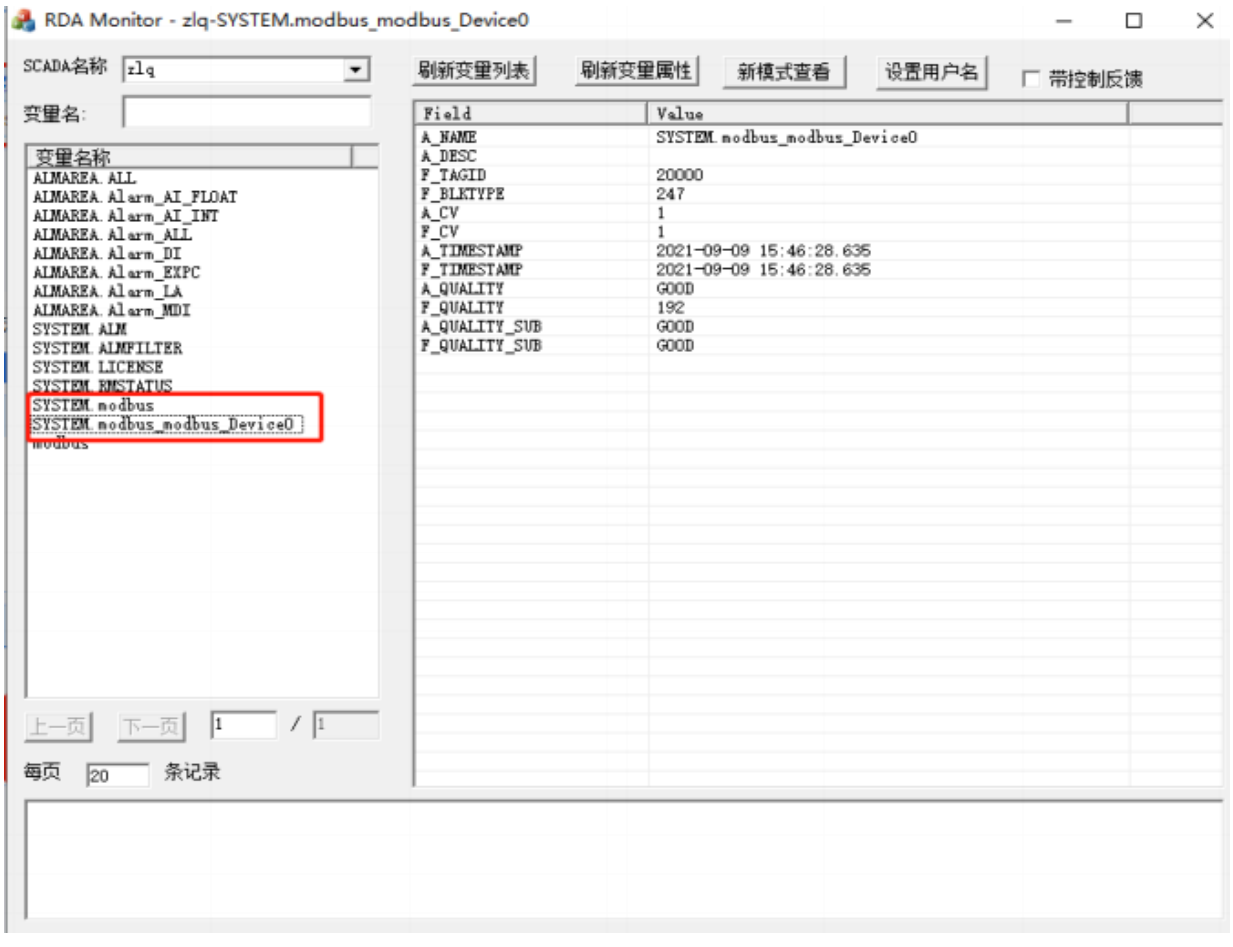


扩展：通过rdamonitor查看状态

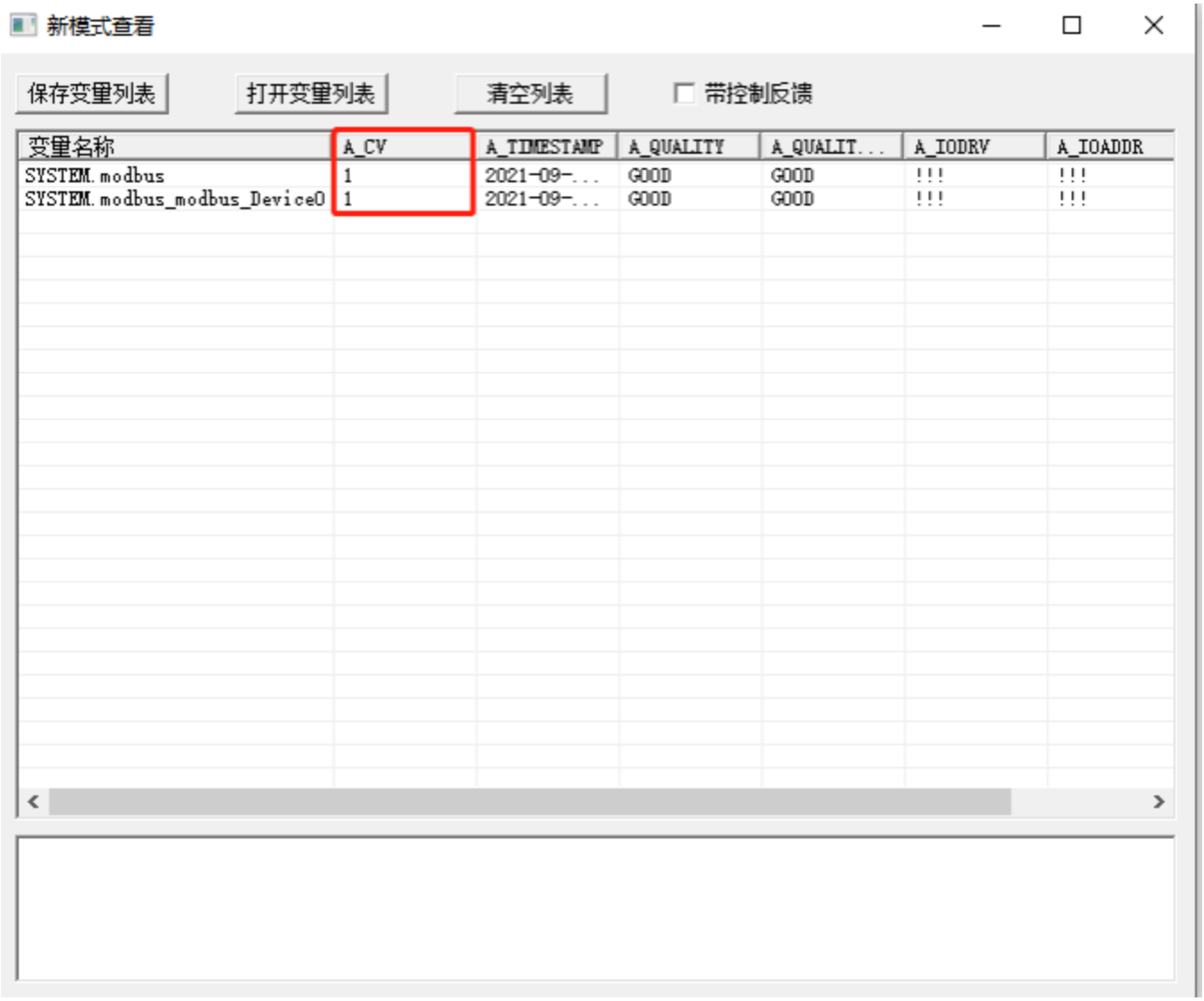
Windows环境下，如果需要排除其他组件干扰，直接查看平台运行数据采集第一步是否正常，则可以通过rdamonitor查看，具体方法如下：

按照安装目录iplatba\iplatda\executable\路径打开iplatba\iplatda\executable\rdamonitor.exe，查看SYSTEM开头的系统点，驱动状态的点的格式为：SYSTEM.驱动名，设备状态的点的格式

为：STYSTEM.驱动名_设备名，如下图



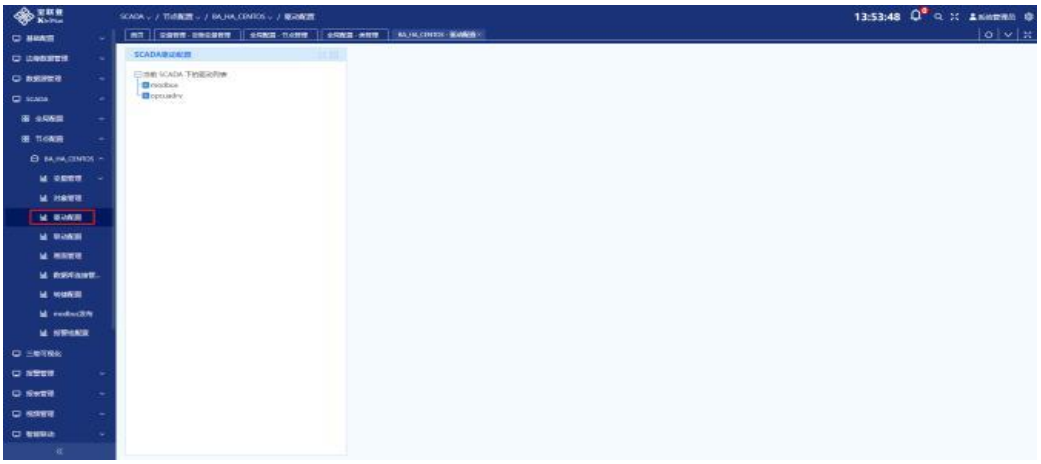
状态点一共有三种，分别为0，1，???。0和???均表示未连接，1表示已连接



gedrv支持GE Rx3i/Rx7i系列PLC数据读写。

打开节点驱动配置

添加scada节点后，点击配置好的scada节点，在下拉列表中点击驱动配置。



添加GE PLC驱动

在scada驱动配置中，右键驱动列表，点击添加驱动

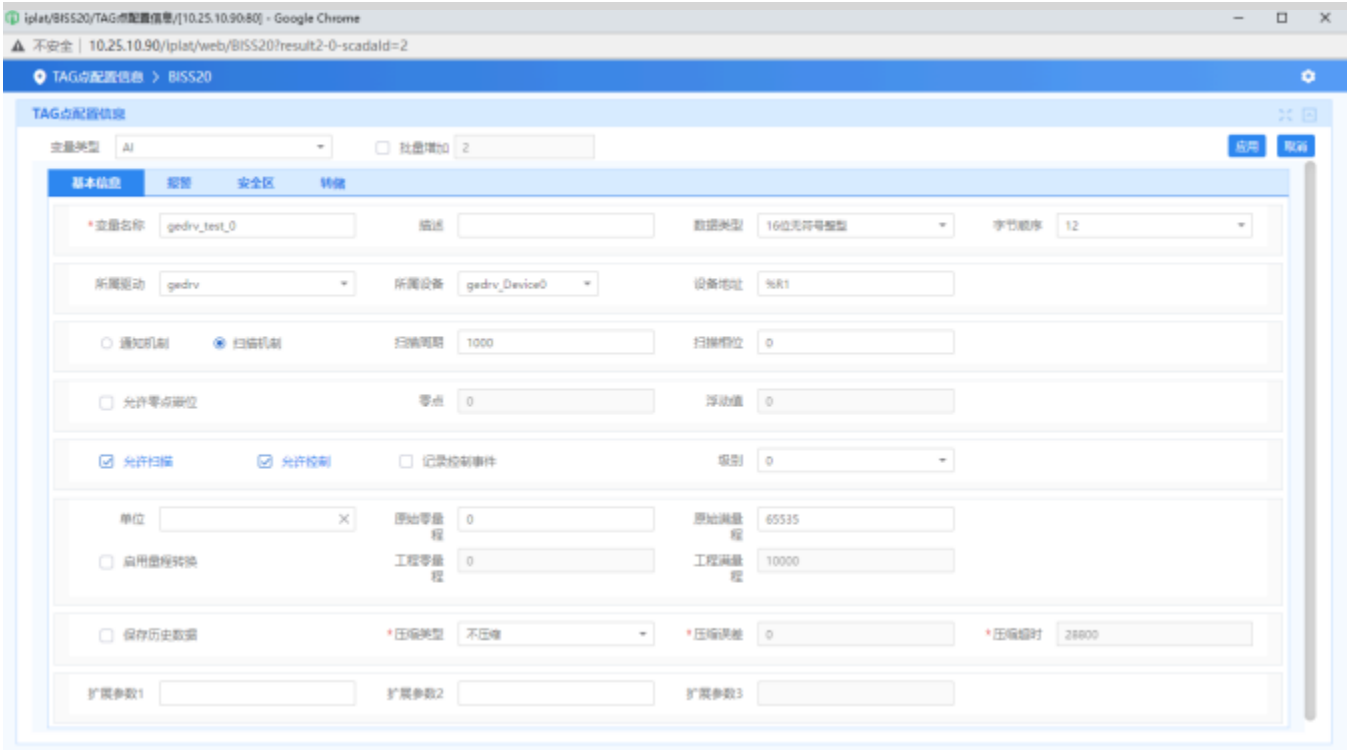
进入系统驱动配置界面，按照红框中的设置添加gedrv驱动

添加设备配置：
在左侧界面驱动名上右击，添加设备

1. 设备名称:设备名,可自定义
2. 描述:设备描述,可选填
3. 任务号:设备的线程号，每个设备配置一个唯一的任务号
4. 响应超时:设备的接收超时，单位毫秒，默认1000ms
5. IP地址（主）：主机plc地址
6. 端口（主）：主机plc端口号，默认为18245
7. 槽号（主）：主机plc槽号，默认为1，可不填
8. IP地址（备）：备机plc地址

9. 端口（备）：备机plc端口
10. 槽号（备）：备机plc槽号，默认为1，可不填
11. 支持多连接：选是允许活动及非活动机同时采集，选否则仅活动机采集。

添加变量配置：



变量地址格式为：%R1，%M2，%M1:0:7，%R12#10，%Q100，%SA1等。

变量地址首位必须是百分号，第二位开始必须是驱动支持的寄存器类型（详见下表），偏移地址最小从1开始。

变量字节顺序为顺序排列。

变量地址支持12种不同类型寄存器。

注意：

1. 寄存器I,Q,M,T,SA,SB,SC,S,G类型的变量不按照DI点配置时，其寄存器地址格式有特殊要求，必须是8的整数倍+1.例如：%Q1,%Q9,%Q17,%T33,%M65.
2. 寄存器I,Q,M,T,SA,SB,SC,S,G类型的变量按照DI点配置时，可按照第18位配置成%T18这种方式配置。

gedrv驱动采用免配块配置方式生成自组块，驱动通过解析变量地址，自动生成自组块。在..\..\executable\drivers\gedrv\目录下的gedrvcfg.txt查看生成的数据块信息。例如：

ge plc共有12中寄存器，表格如下：

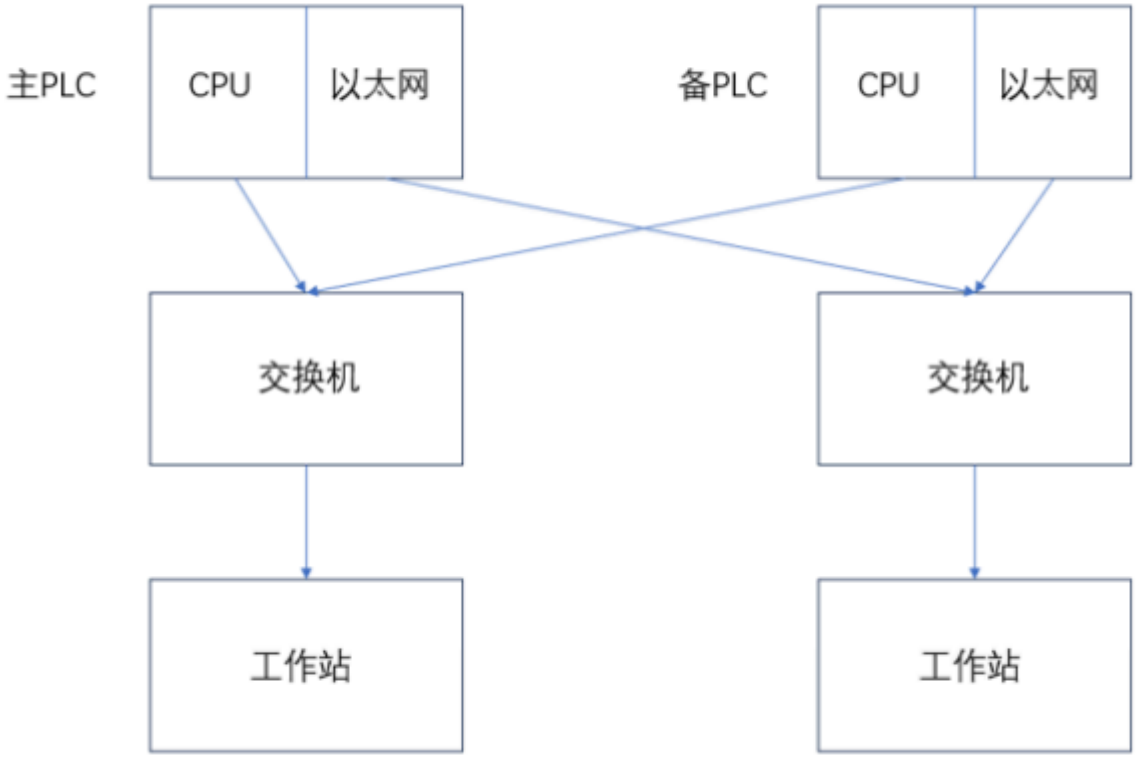
寄存器类型	按位读/写	按字节读/写	按字读/写	是否支持批量读	
I	是/否	是/否		是	用于存储输入信号，如传
Q	是/否	是/否		是	用于存储输出信号，如控
M	是/是	是/是		是	用于存储中间计算结果，
T	是/是	是/是		是	用于存储计时器的当前值，
SA	是/否	是/否		是	用于存储系统的状态信息，
SB	是/否	是/否		是	用于存储控制系统的状态
SC	是/否	是/否		是	用于存储计数器的当前值，
S	是/否	是/否		是	用于存储控制系统的状态
G	是/是	是/是		是	用于存储全局变量和系统
AI			是/是	是	用于存储模拟量输入信号，
AQ			是/是	是	用于存储模拟量输出信号，
R			是/是	是	用于存储数据块，可用于
P	暂不支持	暂不支持	暂不支持	暂不支持	程序寄存器，用于存储程
L	暂不支持	暂不支持	暂不支持	暂不支持	锁存器寄存器，用于存储
W	暂不支持	暂不支持	暂不支持	暂不支持	工作寄存器，用于存储临

GEPLC冗余相关说明

GEPLC的内部冗余设置需要在PME软件中进行修改，通常来说会生成一个冗余IP对外提供访问，同时也可以直接访问主备设备IP，但是非Active设备无法进行写值操作。其中，冗余IP永远只会指向Active设备，若此设备网络连接中断，并不会指向另一台非Active设备。

本驱动进行冗余配置时同时支持配置单个冗余IP或者配置主备设备IP。配置冗余IP时，驱动会以采集单机设备的方式进行数据读写，不做冗余判断。配置主备设备IP时，本驱动只会向Active设备进行读写。

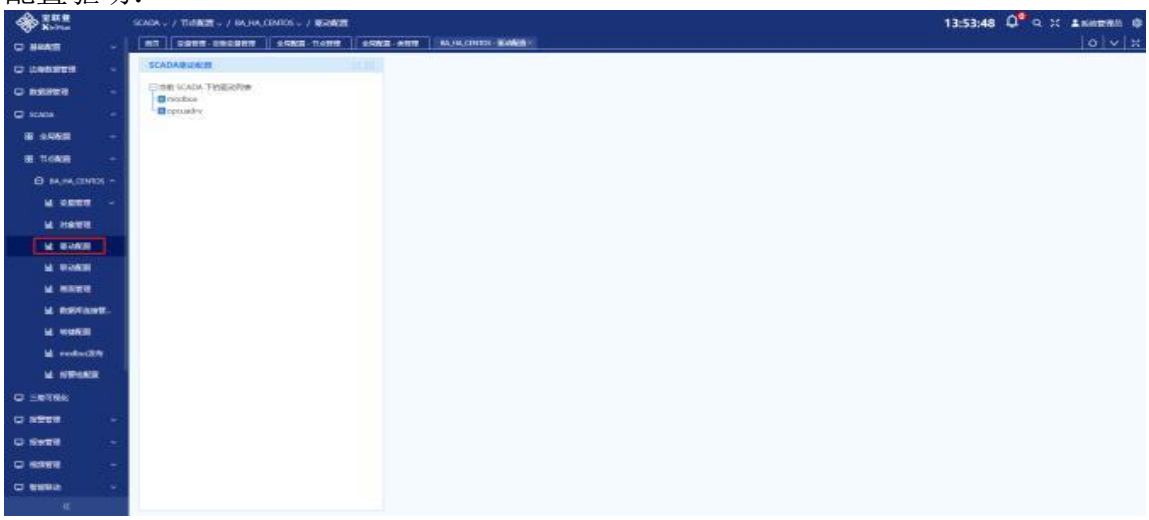
本驱动同样支持双环网配置中的双冗余IP设置，双环网设置大致如下图所示：



在此环境中，冗余IP1是指向的主备的CPU网口，冗余IP2是指向主备以太网网口，如若图中两个交换机网络互相能连通，GE驱动可以同时配置冗余IP1和冗余IP2实现双环网GEPLC的冗余。

说明：AB PLC驱动支持AB公司PLC，目前支持ControlLogix，SLC，micrologix等系列，使用流程和其他驱动类似，具体如下

配置驱动：



添加驱动，可参考SIEMENS snap7drv驱动使用设备配置

- 设备名称：按需求填写
- 连接类型：**固定选Other**
- 响应超时：连接设备的超时时间。
- 轮询周期：设备轮询数据的周期时间，同时会影响到在一次轮询中等待这批次数据的最大时间，目前为轮询周期的一半，若达到该时间有数据未准备好，对应的TAG会被更新为BAD。
- 任务号：按需求填写，同一任务号的设备会放在同一线程进行读取。
- 连接参数：设备IP（注：5.4.6之后的版本AB驱动支持了单连接，格式为"ip=127.0.0.1;multiLink=1;",multilink为1时是支持多连接，原先格式同样支持，默认为单连接）
- 扩展参数1：槽号，默认值为0
- 扩展参数2：超时次数，当设备下的某一个块连续读取失败次数大于该数值时才会将块上的点更新为BAD，可不填
- 扩展参数3：设备类型，填写格式为cpu=XXX，XXX为设备类型，目前支持ControlLogix，compactlogix,SLC，micrologix等系列，此处不填则默认为ControlLogix，填写时注意大小写。

ControlLogix系列通常指AB5000型号。

micrologix系列包含微控制器型号，例如micro1400,micro1500,但是不包含micro800。

SLC系列包含SLC500（AB500）。

配置方式一：配置数据块，然后配置点（目前推荐使用配置方式二）

数据块配置：

- 数据块名称：自行定义，不能为空，不能以数字开头；

•轮询周期：数据块的刷新周期；
•相位：轮询周期中的相位偏移量，用于通讯性能调整；
•类型：可不填；
•地址：数据块所在PLC地址；
•任务号：为所添加数据块分配任务号（线程号）；
•元素个数：要读取的总字节长度；
•元素大小：固定填1；
•描述：自定义的描述信息；
•可选参数1：要读取的数组元素个数，读取非数组此处填1，读取数组时按照实际情况填写
•可选参数2，3不填；

点配置：

基本和原先一致，地址目前的格式是：数据块名称:字节偏移:位偏移

例外的：

1.TXT类型

如果是TXT类型（string），则是：数据块名称:偏移#长度，需要注意的是string由于PLC中包含了开头的4位显示实际长度，这边偏移填写时需要加上该4位，长度不能超过字符串实际最大长度，显示时我们会根据读取到的长度和配置长度进行比较自动截取。

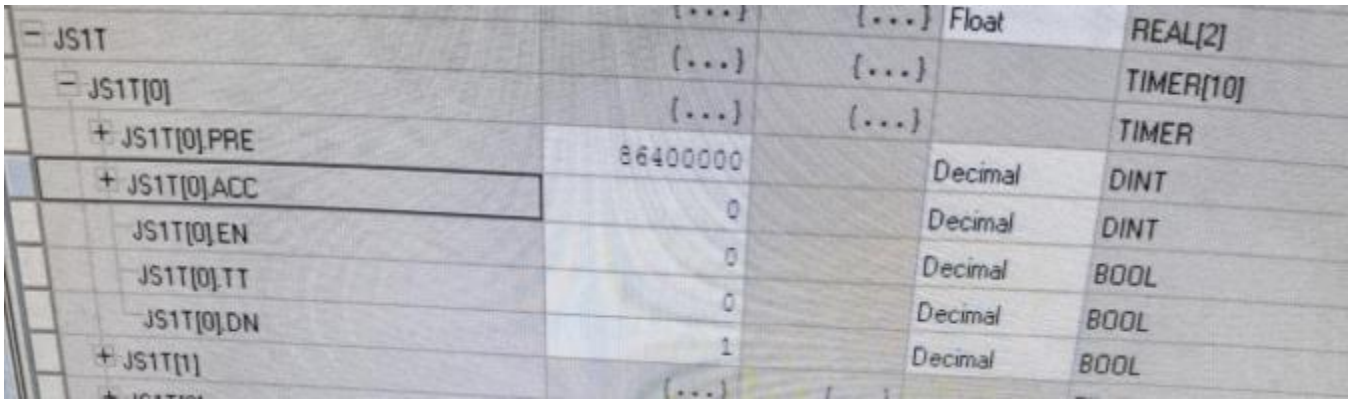
2.BOOL类型

BOOL类型比较特殊，配置时按照如下进行配置：

如果使用自主块方式去取单个BOOL点，需要将点的类型配置成DI，地址中字节偏移和位偏移都是0即可，需要注意的是，由于默认的BOOL的实际请求长度为1，如果需要取得一个4个字节长度中的某一位时，会导致后三个字节的值取不到，这种需求目前必须使用配块方式进行读取。

如果使用配置块的方式进行读取时，需要区分结构体和纯BOOL数组两种情况，在结构体中，每32个bool点会占用4个字节，不足32个会自动补齐成4个字节，用户需要注意的是这些BOOL点实际是如何存储在PLC地址里的（可能会与PLC编程程序中展示的不一致）.

特殊情况如下图Timer结构体所示：



其中JS1T[0].EN，JS1T[0].TT，JS1T[0].DN在JS1T[0]这12个字节的内存中分别对应的是第32位，第31位，第30位（即偏移位31,30,29），JS1T[0].PRE的偏移为4个字节，JS1T[0].ACC为8个。用户需要根据实际情况进行配置。

如果使用配置块方式进行纯BOOL数组读取，需要注意元素个数为数组内的BOOL数量除以32再向上取整再乘以4，因为纯BOOL数组也是每32个BOOL点组成4个字节，可选参数1处填写数组内的BOOL数量除以32再向上取整。

3.String结构体组成的数组

如果需要读取String结构体组成的数组时，需要注意该结构体第二部分由SINT组成的内容在读取时会自动补齐成四个字节大小的倍数，比如当PLC程序里显示SINT长度为6时，实际读取长度为8（总STRING的长度还需要加上前面表示长度的四个字节），在实际配置中，TXT点的长度仍然填写6，但是第二个及以后元素时，填写的偏移需要计算上补齐的字节数，比如此处要读取第二个string的内容时配置应为:DB:16#6，偏移16的计算方式为前一个string的总长度12+第二个string的前4个字节偏移。

4.DI类型

配置DI点时，需要注意该点在PLC中所在的块占多少字节，程序中默认读取长度为一个字节

配置方式二：免配块配置方式(推荐)：

地址格式：与原设备tag地址格式一致，如MotorArray[3].Coils[0].length;

现在免配块方式同时支持离散点、结构体和数组等数据结构，只需要将实际地址填入配置界面中的设备地址即可。

需要注意的是，TXT变量的地址格式为XXXX.DATA#长度，例如：test[0].DATA#12。此处长度为DATA字符串的最大长度。

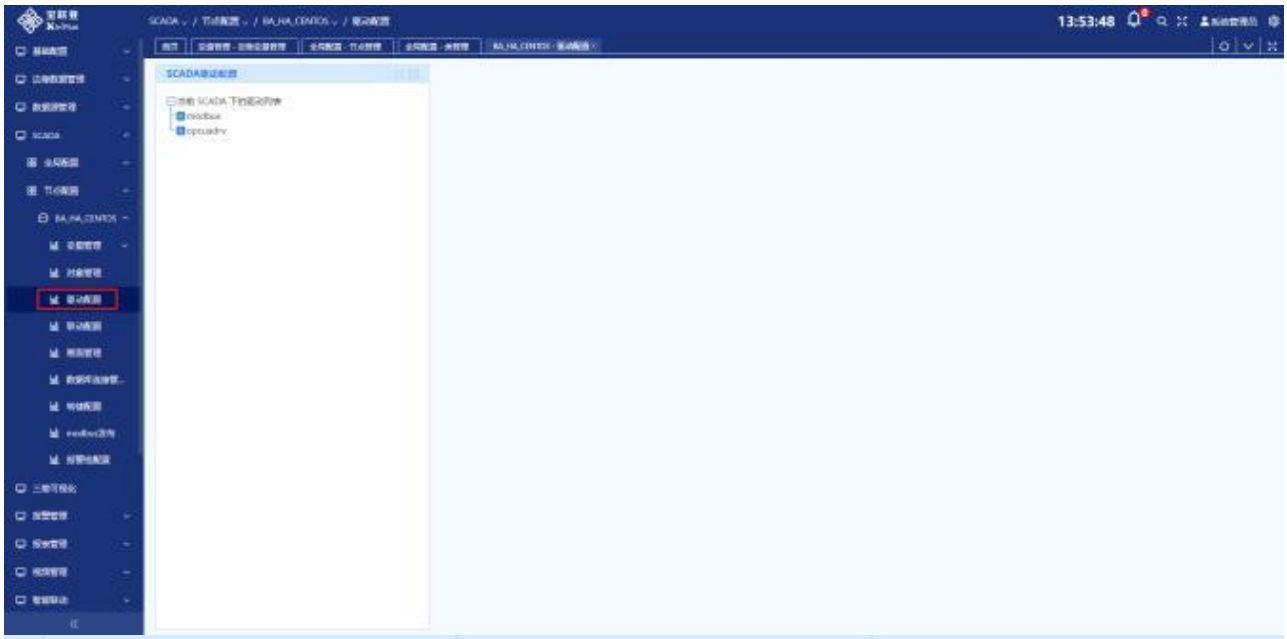
注意事项：

1. 如果需要取一个变量中的某一位，例如AAAA.1，目前仅支持通过配块方式进行读取，具体配置方式参见配置方式一
2. 用户自定义数据类型UDT个数不能超过100万。
3. 若DI类型点，要从其他长度中读取一位，则必须新建一个设备添加数据块，再通过偏移的方式进行读取（即上述配块方式，如下）。

三菱PLC驱动，支持三菱PLC（Q/L和iQ-R）系列PLC

添加melsecdrv驱动

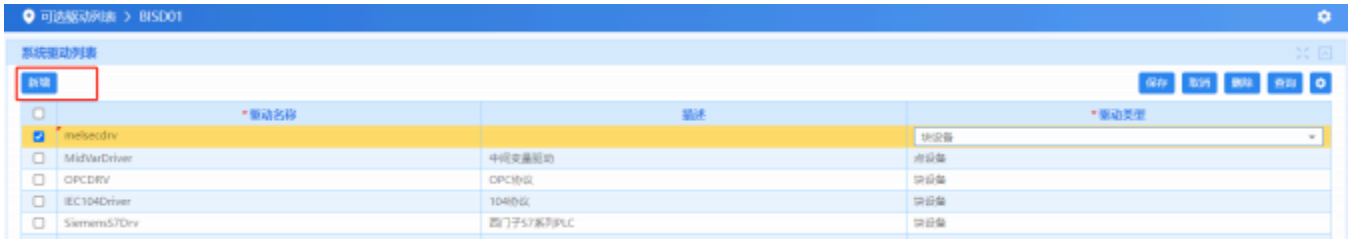
1. 依次点击菜单进入系统驱动配置界面，按照红框中的设置添加三菱驱动。



*驱动名称	描述	*驱动类型
cvrdadriver	IPLAT SCADA数据交互	块设备
CIPDriverX	AB PLC	块设备
SNMP2	SNMP2协议	块设备
tdcdrv	iXcomPCS产品	块设备
tdcwritedrv	iXcomPCS产品	点设备
snap7drv	西门子S7系列PLC	块设备
opcuadrv	OPCUA协议	点设备
dbdrv	关系数据库	块设备
filetagdrv	File Tag Driver	点设备
fileblockdrv	File Block Driver	块设备
abdrv	abdrv 块设备	块设备
codesysdrv	codesys 点设备	点设备
modbusbs	modbus协议	块设备
iplaturedrv	iplature数据采集/控制下发	块设备
geniitekdrv	捷杰驱动	块设备
melsecdrv	三菱驱动	块设备
bscidriver	宝钢联锁驱动	点设备
gedrv	GE PLC	块设备
redisdrv	redis驱动	点设备

注意：

若驱动列表中没有melsecdrv驱动，可在SCADA管理->系统驱动配置中手动添加。



配置melsecdrv设备

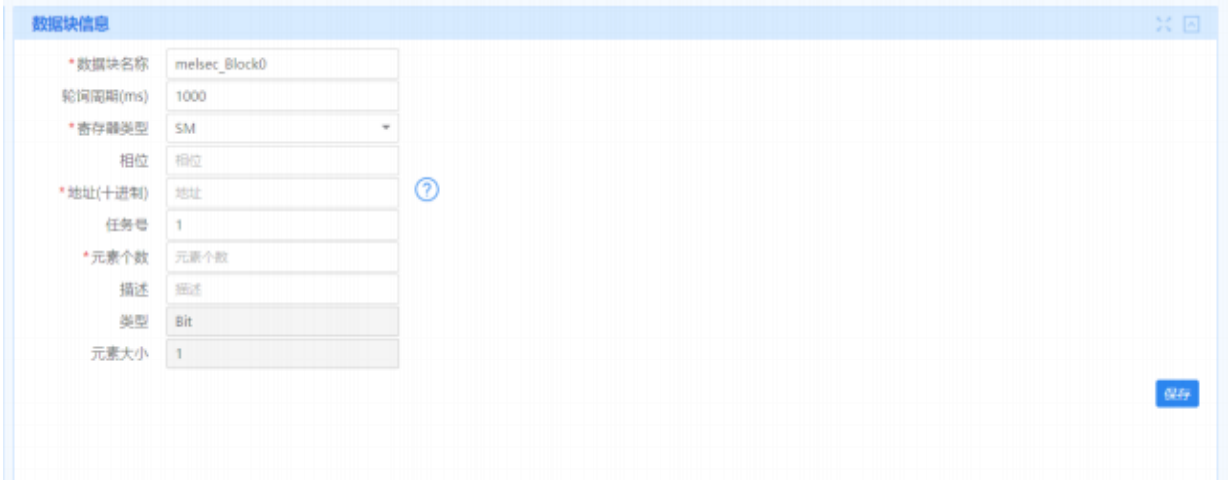
1. 在scada驱动配置中添加melsec驱动，右键选择添加设备，设备配置界面如下图：

- 设备名称：不能为空，不能以数字开头。新增是名称后面的数字会自增；
- 响应超时：连接设备和发送数据的响应超时时间。默认之1000；
- 连接类型：固定选择TCPClient；
- IP地址：欲连接设备的IP，默认值127.0.0.1；
- 端口号：欲连接设备使用的端口号。（注意三菱plc端口号的配置是16进制的，而此处端口号输入的是十进制的，所以配置好设备的端口号记得转换），默认值2001；
- 描述：设备的描述信息；
- 多连接：该设备是否支持多连接，1表示支持多连接。默认值是；
- 任务号：新增时自增，只能填写大于0小于101的正整数；
- 设备类型：目前支持两种类型的设备，Q/L和iQ-R；
- 网络编号:默认值00；
- PIC编号：默认值255；
- 请求目标模块站编号：默认值00；
- 扩展参数1：此处由三个参数组成，网络编号，PIC编号，请求目标模块站编号，一般情况下这三个参数分别为00,255,00，三个参数之间需要用冒号隔开。
- 扩展参数2：设备类型，目前支持两种类型的设备，Q/L和iQ-R。

配置melsecdrv数据块

注意：melsec驱动支持免配块方式，若该设备下没有配置任何块，相应的变量配置参见免配块方式下的变量配置。

1. 选中已经配置的设备，点击右键，选择添加数据块
2. 数据块配置界面，如下图：



- 数据块名称：新增时，名称后面的数字会自增；
- 轮询周期：默认值1000；
- 寄存器类型：下拉框选择寄存器；前支持的类型如下：SD，X，Y，M，L，F，V，B,D，W，TS，TC，TN，STS，STC，STN，CS，CC，CN，SB，SW，DX，DY，Z，R，ZR。
- 地址：寄存器起始地址，需要注意的是，部分寄存器类型的地址需要在填写时转化成十六进制，参考后面的软元件代码表格，**凡是地址范围写十进制的都需要转换成十六进制填写**。问好按钮可以点击后，有地址范围显示
- 任务号：不同任务号的块将使用不同的线程进行采集。默认值为1；
- 元素个数：从起始地址开始取的元素个数，word最大是960，bit最大是7168。需要注意的是，Bit的元素个数至少为2
- 元素大小：BIT写1，Word写16。
- 类型：Word或者Bit，请对照后面的软元件代码表格填写对应的类型。
- 可选参数1：此处需要填写一个特定报文码，根据软元件代码表格对照寄存器类型填写。
- 可选参数2：此处选择寄存器类型，目前支持的类型如下：SD，X，Y，M，L，F，V，B,D，W，TS，TC，TN，STS，STC，STN，CS，CC，CN，SB，SW，DX，DY，Z，R，ZR。

不同的寄存器类型需要使用不同进制的地址，具体参考如下：

软元件代码	地址范围	类型	特定报文码
SM(位单位)	000000---002047（十进制）	Bit	91H
SD(字单位)	000000---002047（十进制）	Word	A9H
X(位单位)	000000--001FFF（十六进制）	Bit	9CH
Y(位单位)	000000--001FFF（十六进制）	Bit	9DH
M(位单位)	000000--008191（十进制）	Bit	90H
L(位单位)	000000--008191（十进制）	Bit	92H
F(位单位)	000000--002047（十进制）	Bit	93H
V(位单位)	000000--002047（十进制）	Bit	94H
B(位单位)	000000--001FFF（十六进制）	Bit	A0H
D(字单位)	000000--012287（十进制）	Word	A8H
W(字单位)	000000--001FFF（十六进制）	Word	B4H
TS(位单位)	000000--002047（十进制）	Bit	C1H
TC(位单位)	000000--002047（十进制）	Bit	C0H
TN(字单位)	000000--002047（十进制）	Word	C2H
SS(位单位)	000000--002047（十进制）	不支持	不支持

SC(位单位)	000000--002047（十进制）	不支持	不支持
SN(字单位)	000000--002047（十进制）	不支持	不支持
CS(位单位)	000000--001023（十进制）	Bit	C4H
CC(位单位)	000000--001023（十进制）	Bit	C3H
CN(字单位)	000000--001023（十进制）	Word	C5H
SB(位单位)	000000--0007FF（十六进制）	Bit	A1H
SW(字单位)	000000--0007FF（十六进制）	Word	B5H
S(位单位)	000000--008191（十进制）	不支持	不支持
DX(位单位)	000000--001FFF（十六进制）	Bit	A2H
DY(位单位)	000000--001FFF（十六进制）	Bit	A3H
Z(字单位)	000000--000015（十进制）	Word	CCH
R(字单位)	000000--032767（十进制）	Word	AFH
ZR(字单位)	000000--0FE7FF（十六进制）	Word	B0H

备注：根据具体硬件设备不同，某些软元件不可读写。
配置melsecdrv变量
免配块的Tag点设备地址的配置方式：
•DI/DO：X100,X0F1,D100.0（注意，如果是十六进制的，第一个地址如果为A-F，前面请加0。）前面字母大小写不敏感。注意，例如D100.0中"."之后的数字为十进制，如果是C之类的十六进制显示需转换为十进制配置
•AI/AO：D100。
•TEXT/BLOB：起始地址#数据长度（字节）
例如：D102#10
配块方式变量配置：

- •AI变量/AO变量地址：
  设备地址格式为：数据块名:B偏移量。（B代表以字节为单位进行偏移）
  例如：block1:B(W,DW,FW)2
- •DI变量/DO变量地址：
  设备地址格式为：数据块名:b偏移量。（b代表以位为单位进行偏移）
  例如：block1:B2：1
- •TEXT、BLOB变量地址：
  设备地址格式为：数据块名:偏移字节#字节长度
  例：block1:2#50

表示从数据块Block1的第2字节开始读取50字节的长度。
扩展：监控发送写消息命令
对于用户输入的写消息命令，根据配置不同，分别按位、字节进行说明。

1. 按位写入

输入此功能的监控命令时，用户根据配置文件，输入需要写入的数据，这里只能输入一个Bit位，0或1。数据类型为数字量。

2. 按字写入

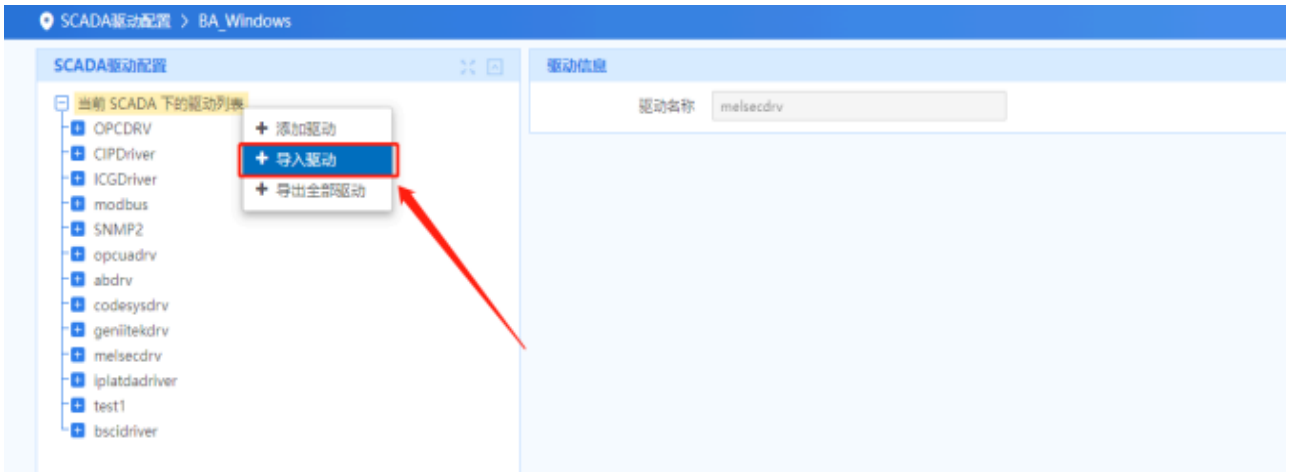
输入此功能的监控命令时，用户根据配置文件以及指定软元件地址和软元件点数输入需要写入的数据。

扩展：驱动配置文件说明

三菱驱动配置对应文件名为：melsecdrv.xml，驱动配置文件所在目录为iplat-da\projects\defaultproject\config\drivers

文件格式为，具体内容可以参考前面几节的说明：

```
<?xml version="1.0"encoding="GB2312"?>
<driver name="melsecdrv"type="block"desc="">
                                                
```



选中SCADA根目录，右键点击，选择“导入驱动”，在弹出的窗口选择导入文件，点击打开按钮。导入成功，系统出现提示信息。

3. Tag点表导出

点击上图所示图标，出现如下页面

在此画面中先选择相对应SCADA节点，再点击右上“导出”键，弹出如下窗口

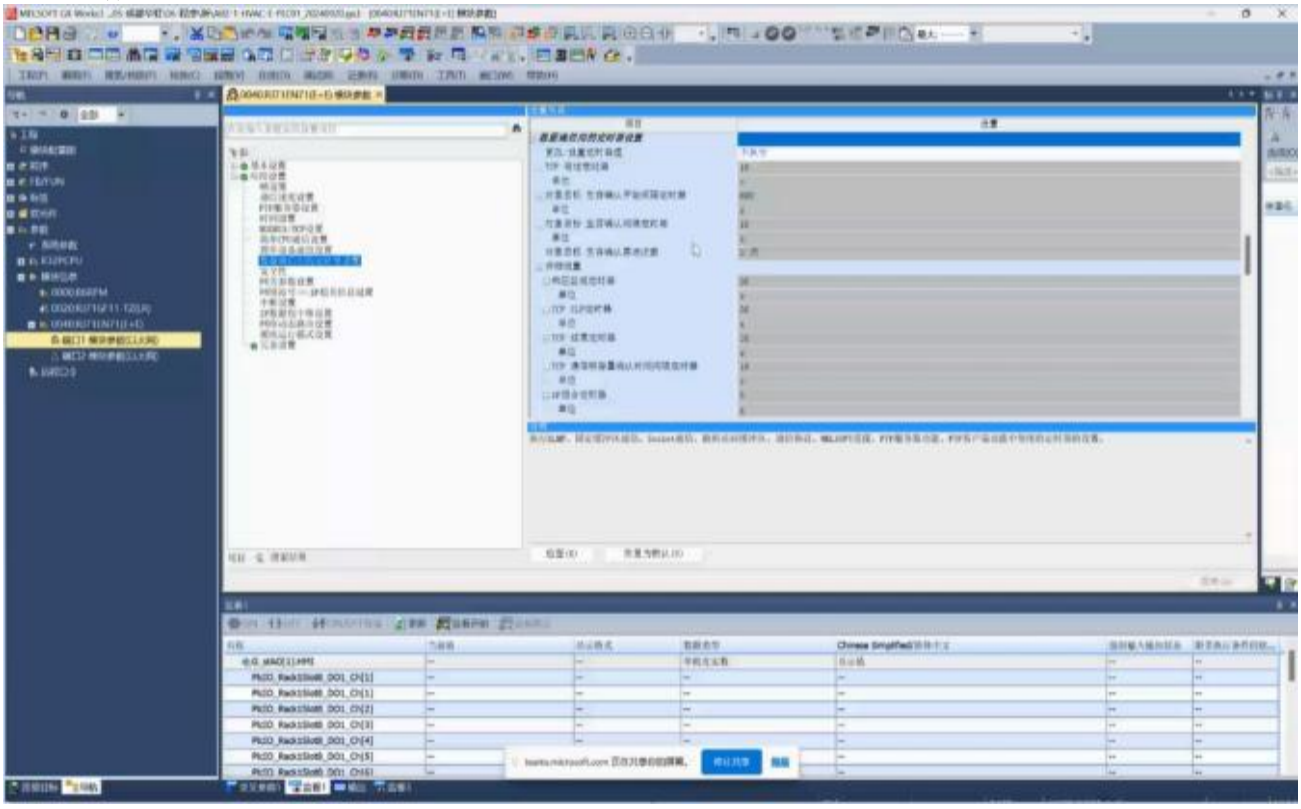
在此窗口中选择变量配置，然后点击“确定”按钮，弹出如下窗口

先选择对应驱动，再根据业务所需选择导出设备,子系统,变量类型等信息后，点击“确认”按钮，然后在网页左下角可查看导出的相对应的表格

PLC设备相关配置参数说明

注意：下述的设备参数修改下发到设备会使得设备短暂中断工作，任何三菱设备侧的相关参数修改请在三菱工程师的指导下进行

当在使用三菱驱动过程中遇到网络中断后恢复，但数据需要较长时间才能重新正常采集的情况，可能是由于三菱PLC侧相关参数对象目标生存确认开始间隔定时器或对象目标生存确认间隔定时器的设置导致的，相关配置界面如下图：



OPC协议简介

OPC协议（Object Linking and Embedding(OLE)for Process Control）是微软公司的对象连接和嵌入技术在过程控制方面的应用。OPC规范从OLE/COM/DCOM的技术基础上发展而来，并以C/S模式为面向对象的工业自动化软件的开发建立了统一标准，该标准中定义了在基于PC的客户机之间进行自动化数据实时交换的方法。

注意：OPC协议只支持Windows系统

OPC驱动配置说明

OPC驱动配置分两步：•在使用OPC驱动之前，需要先在系统中配置DCOM；

•在宝信基础自动化软件中添加OPC驱动及变量配置。

DCOM配置

⚠无论客户端服务器是否均在本地都请先安装OPC Core Components 2.00 SDK 2.20.msi和OPC Core Components Redistributable(x64)3.00.107.msi。

OPCServer服务端

系统组件服务

1、开始菜单：

Win7（server2008）：开始菜单-附件-系统工具-管理工具-组件服务

Win8以上（server2012以上）：开始菜单-管理工具-组件服务

2、控制面板：

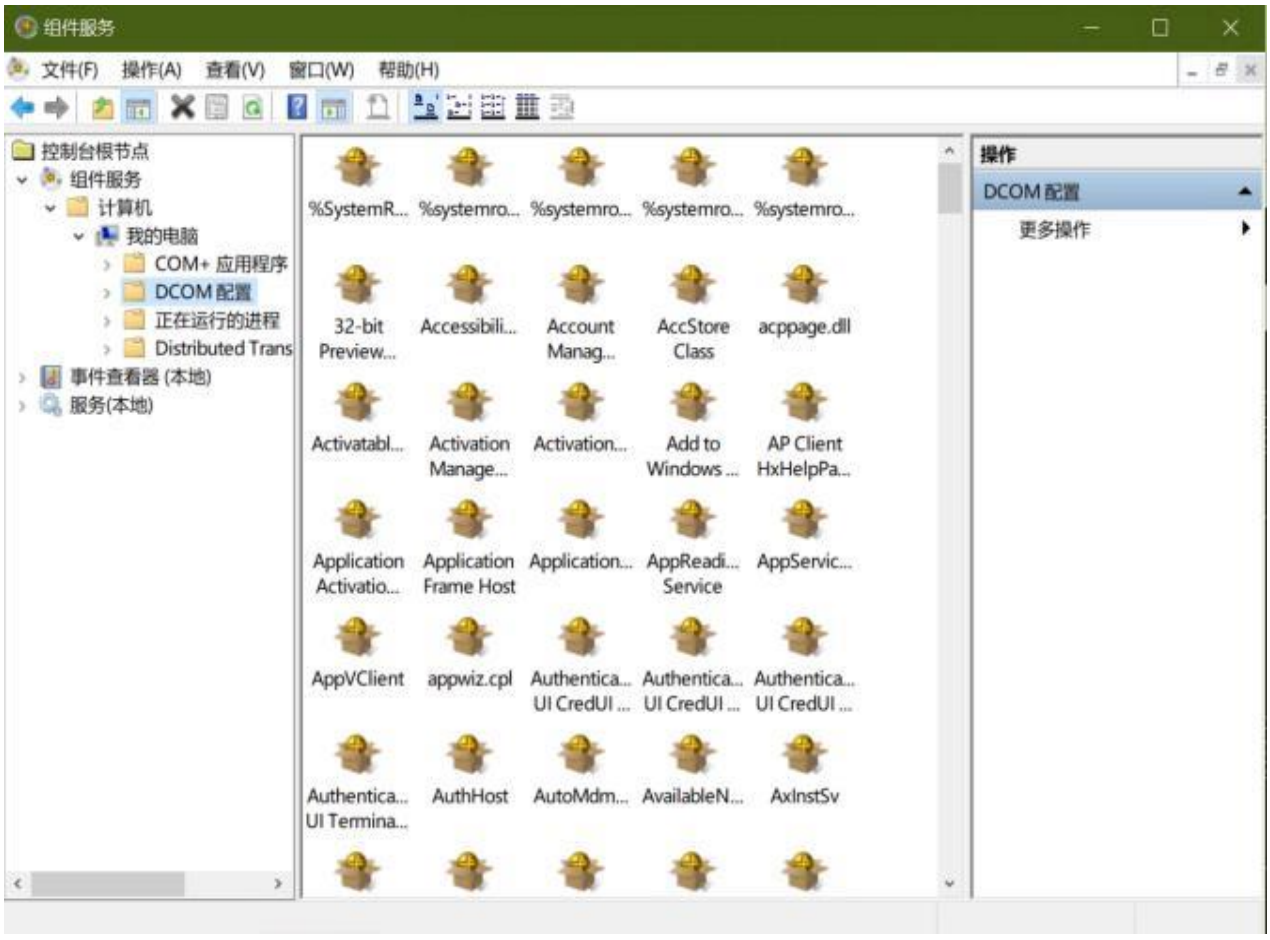
Win7（server2008）：开始菜单-控制面板（大图标）-管理工具-组件服务

Win8以上（server2012以上）：右击开始菜单-控制面板（大图标）-管理工具-组件服务

3、快捷命令：

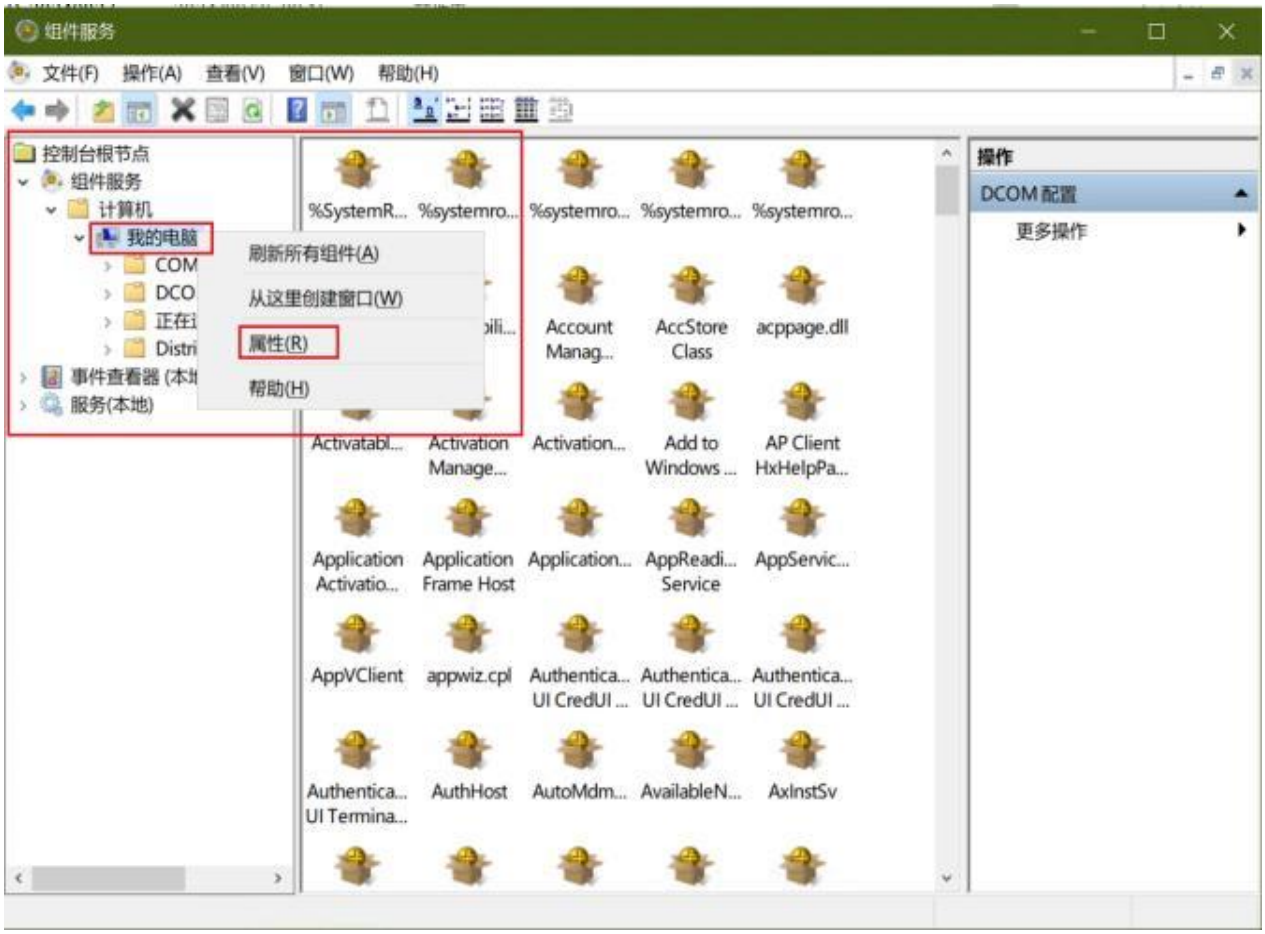
Win+R：输入

dcomcnfg

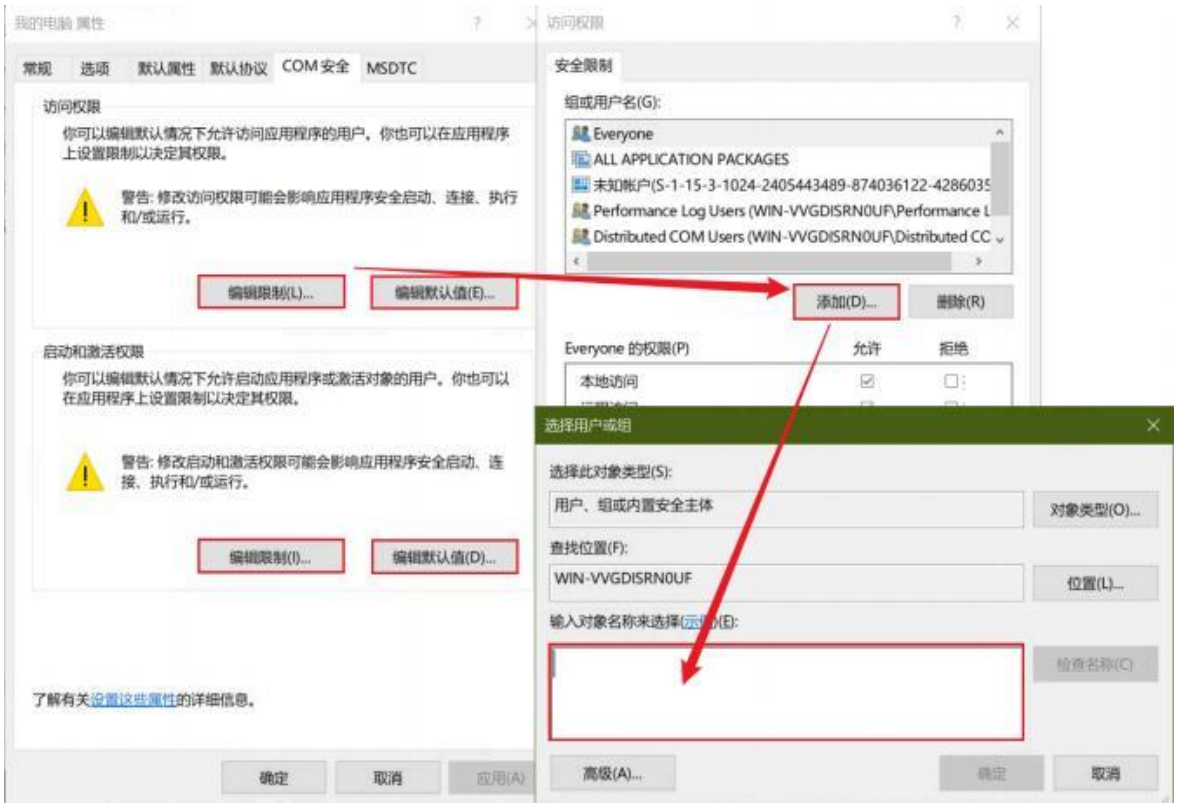


⚠添加用户（重要）
⚠如果Server端和Client端的用户名和密码一致，可以跳过此步。
⚠如果Server端和Client端的用户名一样，但是密码不一样，请修改密码。
⚠如果server端和client端启动的用户不一样，需要先添加用户，用户名和密码需要一致，并授予该用户管理员权限，建议用该用户登入系统进行配置。如果因条件限制，不能使用该用户登入配置，具体见配置OPCServer属性。

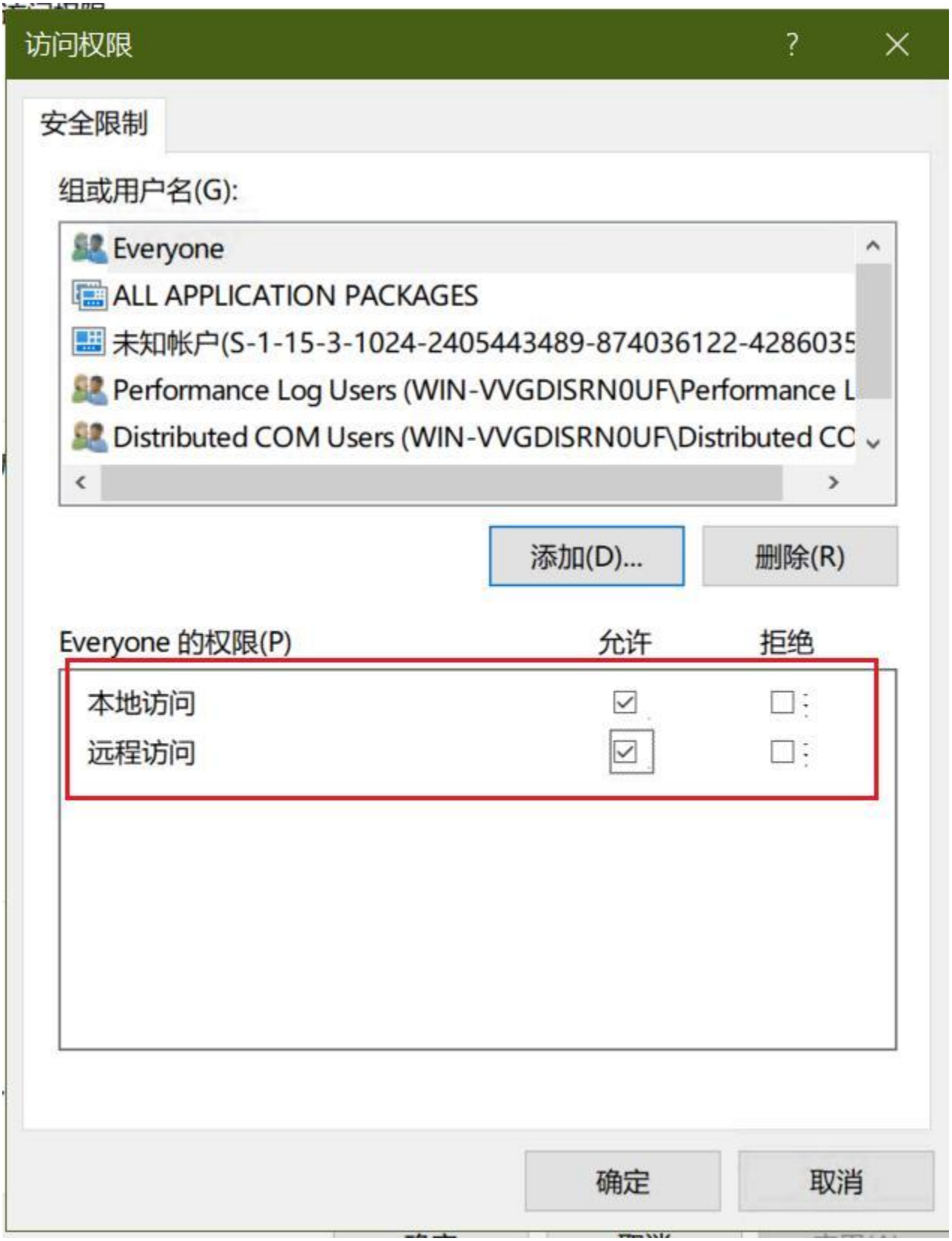
配置计算机属性
右击组件服务-计算机-我的电脑-属性



点击图中的4个红框按钮，编辑安全限制：


我的电脑 属性
常规
选项
默认属性
默认协议
COM 安全
MSDTC
访问权限
你可以编辑默认情况下允许访问应用程序的用户。你也可以在应用程序上设置限制以决定其权限。
警告: 修改访问权限可能会影响应用程序安全启动、连接、执行和/或运行。
编辑限制(L)...
编辑默认值(E)...
启动和激活权限
你可以编辑默认情况下允许启动应用程序或激活对象的用户。你也可以在应用程序上设置限制以决定其权限。
警告: 修改启动和激活权限可能会影响应用程序安全启动、连接、执行和/或运行。
编辑限制(I)...
编辑默认值(D)...
了解有关设置这些属性的详细信息。
确定
取消
应用(A)
访问权限
安全限制
组或用户名(G):
Everyone
ALL APPLICATION PACKAGES
未知帐户(S-1-15-3-1024-2405443489-874036122-4286035
Performance Log Users (WIN-VVGDISRN0UF\Performance L
Distributed COM Users (WIN-VVGDISRN0UF\Distributed CC
添加(D)...
删除(R)
Everyone 的权限(P)
允许
拒绝
本地访问
选择用户或组
选择此对象类型(S):
用户、组或内置安全主体
对象类型(O)...
查找位置(F):
WIN-VVGDISRN0UF
位置(L)...
输入对象名称来选择(示例)(E):
检查名称(C)
高级(A)...
确定
取消

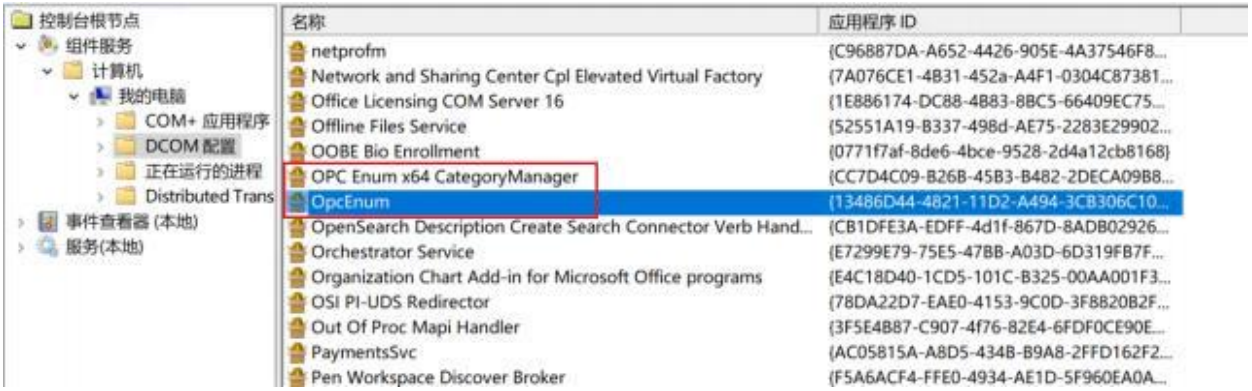
依次添加Everyone、ANONYMOUS LOGON、SYSTEM、Administrators、INTERACTIVE组或用户名，对所有组或用户名中列出的用户授予全部的本地和远程权限。



配置完成后点击确定。

配置OPCServer属性

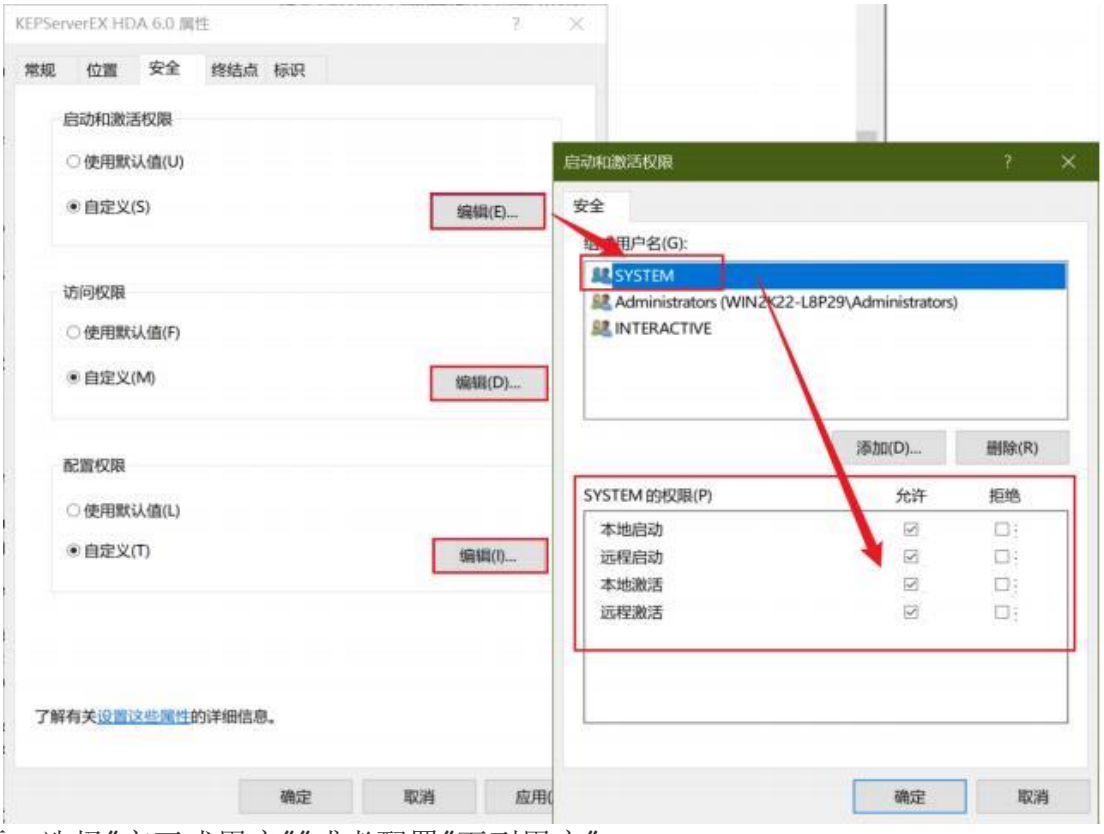
•点击我的电脑-DCOM配置，在右侧选择已安装的OPCServer服务以及OpcEnum服务。注：这里是以KepWare的OPCSERVER为例，其他OPC数据源请以实际为准。



对已安装的OPCServer服务配置安全策略：方法同第3步

右击OPCServer服务以及OpcEnum→属性，点击安全→红框中的3个编辑：

添加Everyone、ANONYMOUS LOGON、SYSTEM、Administrators、INTERACTIVE用户，并授予全部的本地和远程权限。



在标识页，选择"交互式用户""或者配置"下列用户"。

配置本地安全策略

打开“控制面板→管理工具→本地安全策略”，点击“本地策略→安全选项”，在右侧选择“网络访

问：本地账户的共享和安全模型”，右击→属性或者双击打开打开属性页，选择“经典-对本地用户进行身份验证，不改变其本来身份”。

所有配置完成后，务必重启机器。

OPCClient客户端

Client端配置同服务端的配置第1-3步，其他无需操作，配置完成后重启机器。

如果已经可以查找到OPC Server，但是在编辑节点的时候提示“不能浏览该OPCserver”或者提示“连接OPC Server失败，错误码：1”，请检查一下Server端和Client端的用户名密码是否一致。

驱动及变量使用

使用OPCClient配置OPC驱动

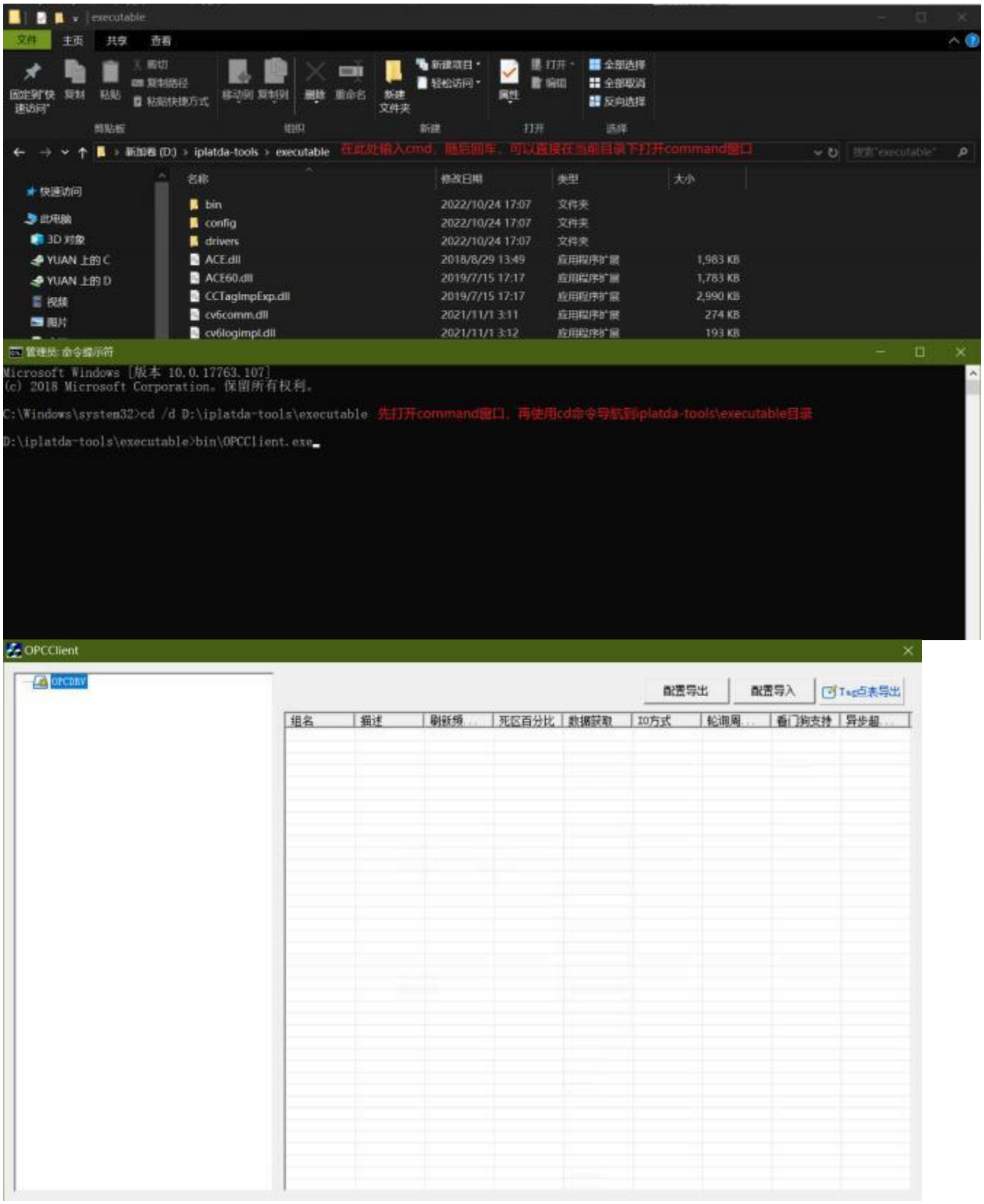
配置OPC驱动需要OPCClient小工具。OPCClient小工具是宝信基础自动化软件的一个驱动辅助工具。后期已经合成iplatda-tools小工具集，可以向支持人员获取。

启动工具集中的OPC-Client

在command窗口中导航到iplatda-tools/executable目录下，或者先用文件管理器打开iplatda-tools/executable目录，随后在地址栏中输入“cmd”打开command窗口，随后执行bin\OPCClient.exe，打开OPCClient驱动配置页面。

若弹出“导出初始化失败”的窗口，点击“确定”即可。

请务必是在executable目录下执行bin\OPCClient.exe命令，否则会出现找不到dll的报错！

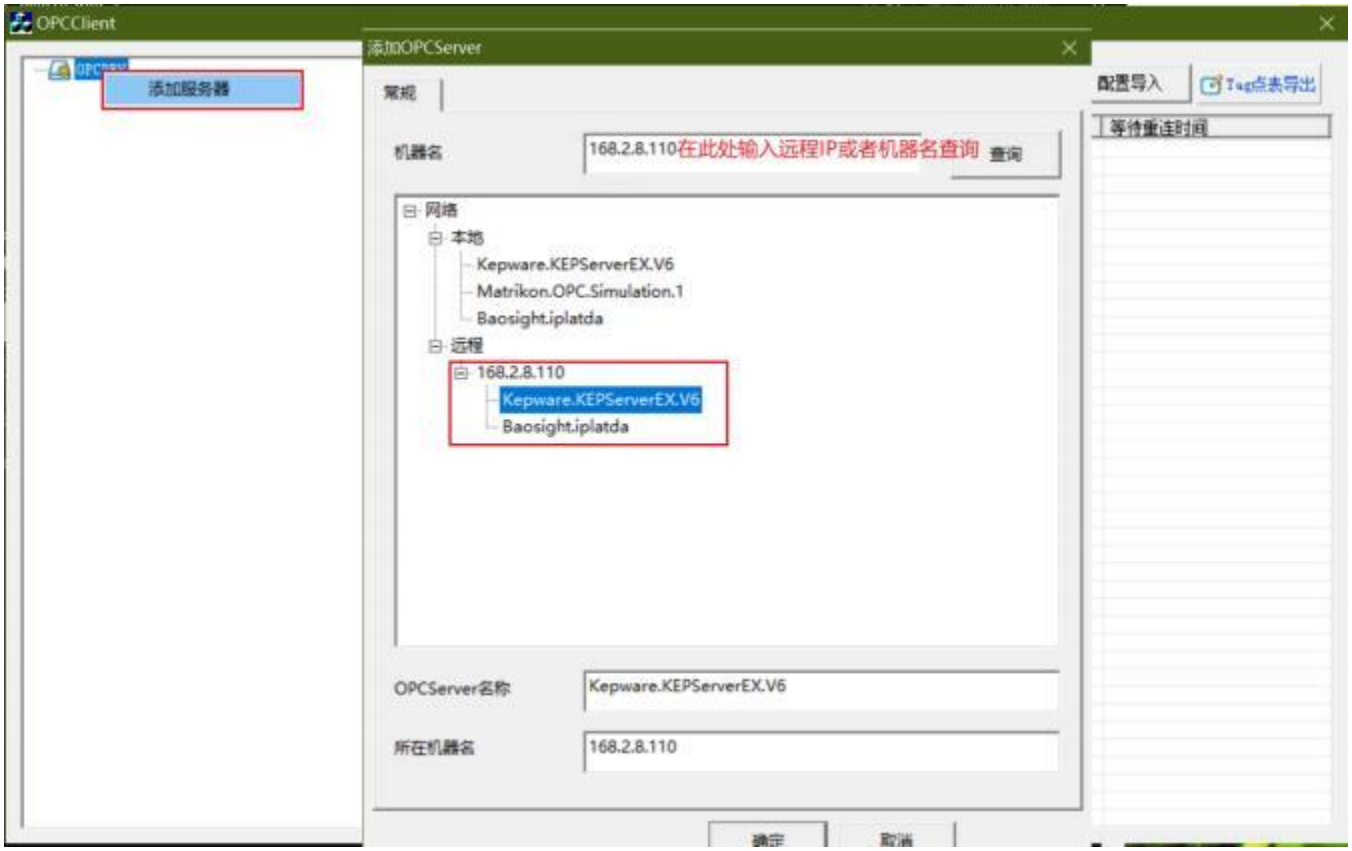


添加服务器

在OPCClient界面上，右键点击树列表中的驱动名称节点，弹出右键菜单，选择“添加服务器”菜单。
在弹出的添加服务器界面中点击树的根节点“网络”，会展开“本地”和“远程”两个子节点。

•点击“本地”节点，会列出本地机器上的所有OPC-Server名称；
•在上方的搜索框中填入远程IP或者远程机器名，点击搜索，如果能搜索到相应的OPC-Server，则会在下方的“远程”节点下面显示相应的OPC-Server；

•点击某一个OPC-Server，下方编辑框中会显示相应的OPC-Server所在机器的机器名或者所在机器的IP。

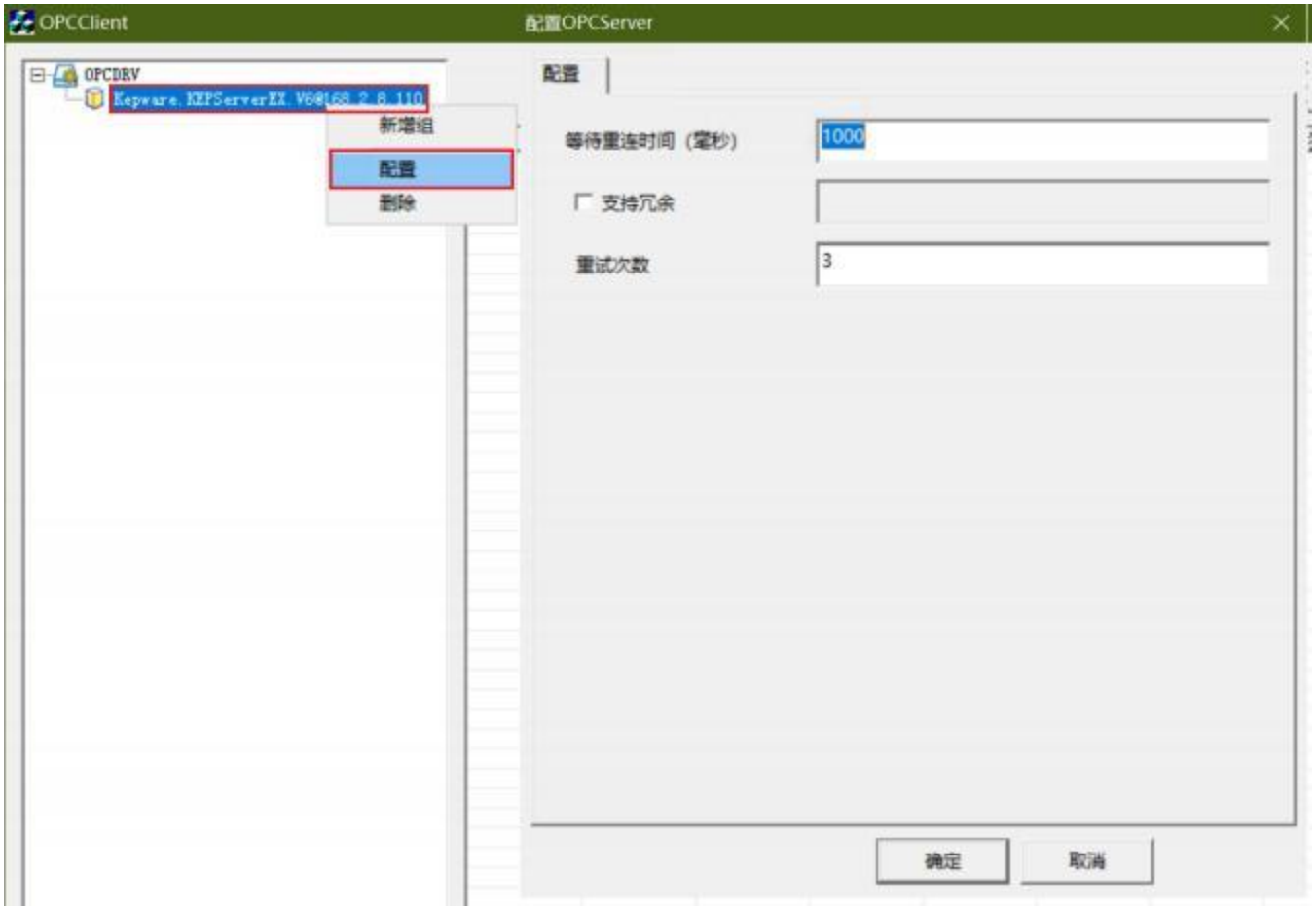


⚠OPC客户端连接远程服务端必须配置DCOM端口，详见DCOM配置中的远程连接OPC服务器配置。

在所有查询到的OPC-Server中选中需要添加的OPC-Server数据源，点击“确定”，系统会在根节点下增加子节点，名称为“OPC-Server名称@机器名/IP”格式。

⚠OPC-Server名称不能为空，“OPC-Server名称@机器名”不能重复。

在驱动配置界面上，右键点击树列表中的OPC-Server名称节点，弹出右键菜单，点击配置。



- 等待重连时间：当出现与设备断开连接的时候，自动重新连接opc设备，此处默认为1000ms
- 支持冗余：勾选后可填写OPCServer的冗余机器ip地址。
- 重试次数：当OPCServer的主机发生连接不上的时候，重连一定次数后，会去连接配置的冗余机器。如果没有OPCServer冗余，则会一直重连下去。
- 使用本地时间戳：勾选的话，采集到的数据使用的是机器的本地时间戳；不勾选的话，数据采集的时间戳是OPCServer的时间戳。

组配置

在驱动配置界面上，右键点击树列表中的OPC-Server名称节点，选择“新增组”，打开“组配置”窗口。

- 名称：不能为空且不能重复。在1-31个字符
- 描述：在0-63个字符
- 刷新频率：OPC服务端到设备读取数据的周期。在1ms-24H之间，必须是100的正整数倍。默认值：1秒
  - OPC客户端配置的刷新频率会被客户端传送给服务端，具体实现看服务端。
  - 为了避免丢点现象，如果用户知道设备的数据变化周期，建议将驱动的刷新周期配置为设备的数据变化周期的一半，点的扫描周期配置为数据变化周期的4分之一，如果使用到转储功能，建议将转储数据属性中的扫描周期配置为数据变化周期的8分之一，如设备的数据变化周期为4000毫秒，那么驱动刷新周期为2000毫秒，点的扫描周期为1000毫秒，转储数据的扫描周期为500毫秒。
- 死区百分比：OPC服务在接收请求时进行判断：和上一次获取的数据值相比，波动范围在死去百分比之内变化时，仍然发送上次的数据值，如第一次获取的数据值为100，死区百分比设为2%，第二次获取的数据值为98~102，那么仍然发送100。
- 数据获取：OPC数据获取有缓存和设备两个选项，用户可以根据自己的需求进行选择，区别如下：
  - 缓存：从缓存上获取数据比设备的读取速度更加的快；
  - 设备：从设备上获取的数据比缓存中获取到的新；
- IO方式：包含异步和同步两个选项。
  - 轮询周期：只在I/O方式为同步读取时可配，OPC客户端去服务器读数据的周期。在1ms-24H之间，必须是100的正整数倍。默认值：在同步时，值为1秒；异步时为50秒。
- 支持看门狗：只在I/O方式为异步读取时可配，作用是在一定周期（异步超时）内OPC服务未返回数据，则将变量的质量置为未确定（Uncertain）。
- 异步超时：只在I/O方式为异步读取时可配，在1ms-24H之间，必须是100的正整数倍。值必须大于轮询周期。默认值:1000秒。

配置完成后，点击确定按钮，系统增加该组。

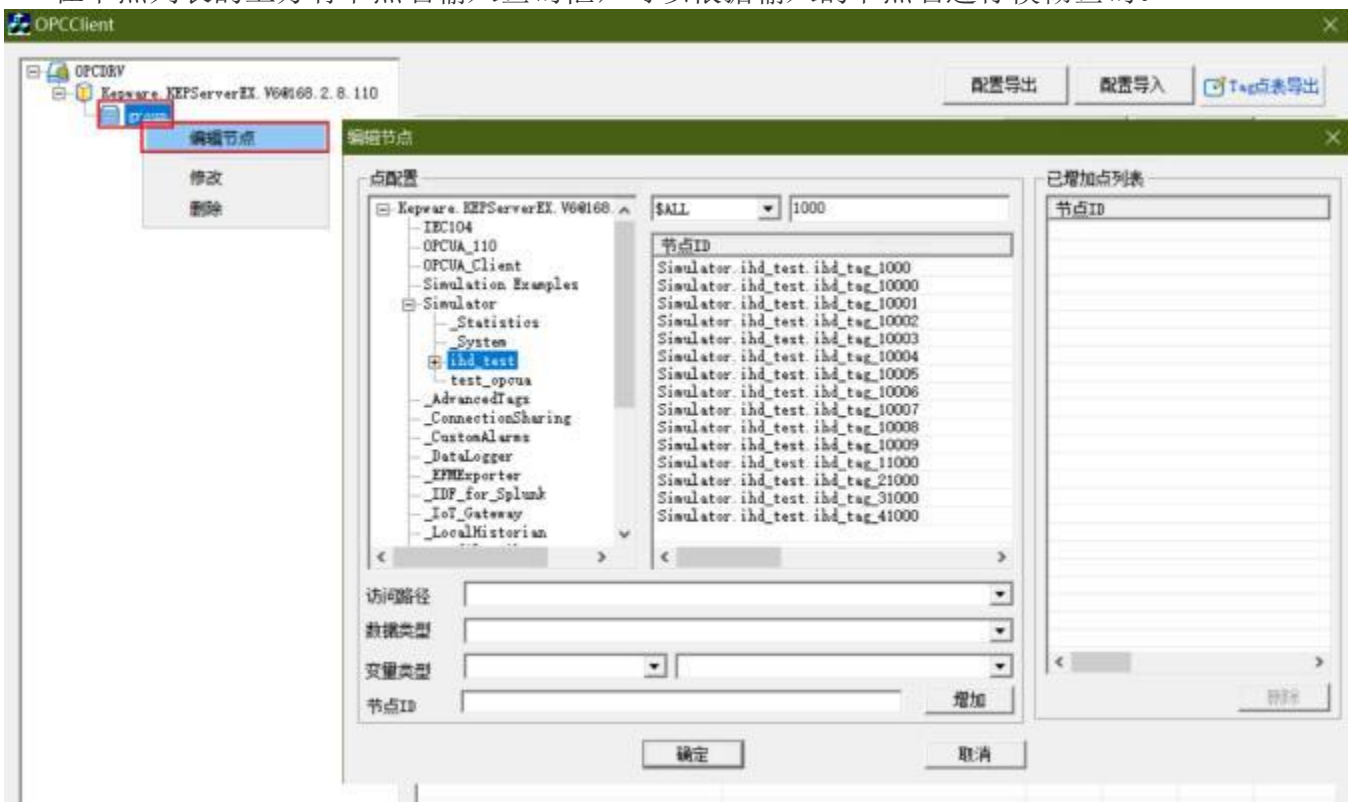
节点配置

新增节点

在驱动配置界面上，右击树列表中的新增加的组名称节点，选择“编辑节点”选项，弹出“编辑节点”窗口。

- 界面左上方的显示OPC-Server信息，点击根节点前的+号，展开子节点，点击每一层子节点前的+号，系统会展开该节点下的子节点。选择任意节点,该节点下配置的节点信息会显示在界面中间的节点列表中。

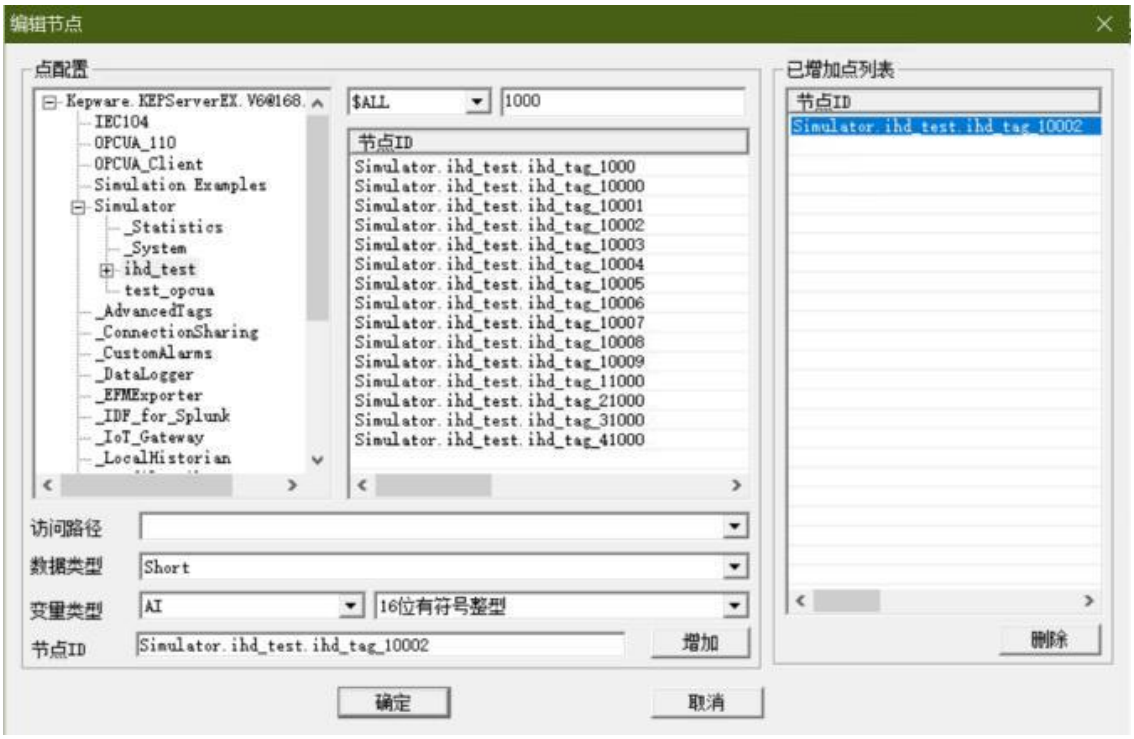
- 在节点列表的上方有节点名输入查询框，可以根据输入的节点名进行模糊查询。



点击中间的节点ID列表，选择一个节点，下方的编辑框中会显示该节点的访问路径（访问路径是由用户在OPC服务中设置的。如果使用OPC客户端连接其他支持访问路径的OPC服务器时，选中节点列表中的节点后，如果该节点有访问路径，会自动填充到下拉列表中，进行选择。）、ID等信息、用户选择要配置的变量类型信息，如果选择的不是AI/AO，则变量类型后的combobox不可用。

点击“节点ID”右边的“增加”按钮，系统会将该节点添加到界面右侧的列表中。

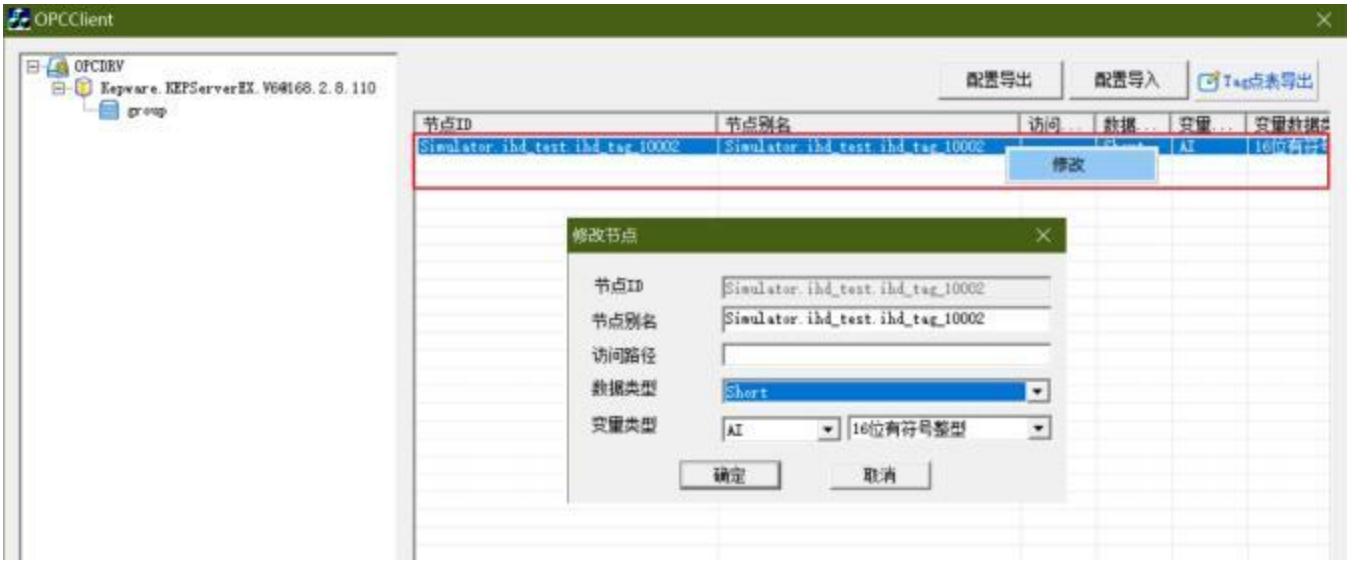
- 节点名不能为空，且不能超过63。
- 对同类型的节点，可以使用ctrl或者shift键选择多个点进行批量添加。
- 在右侧列表中，选择一条或多条记录，下方删除按钮可用，点击删除确定删除对话框，选择确定从列表中删除。



点击确定按钮，将界面右侧列表中的记录保存，并显示在主界面右侧列表中。

修改节点

选中OPCClient中右侧的节点列表的节点，右键点击节点，选择“修改”，弹出修改单个节点窗口。



- 节点别名：节点别名的功能等同于节点ID，如果配置了节点别名，节点别名将会替代节点ID，节点ID将无效。如果节点ID的长度超过64个字节，会自动显示为红色，帮助用户识别修改。
- 访问路径：由设备提供，默认为空。如果修改访问路径，则用户必须保证路径的正确性。

点击确定，保存用户修改的信息。

导出OPC点表

上述配置完成之后，可以点击右上角的“Tag点表导出”按钮，导出点表。

⚠该点表只能作为配置Tag点的辅助表格，切不可直接导入宝信基础自动化软件。

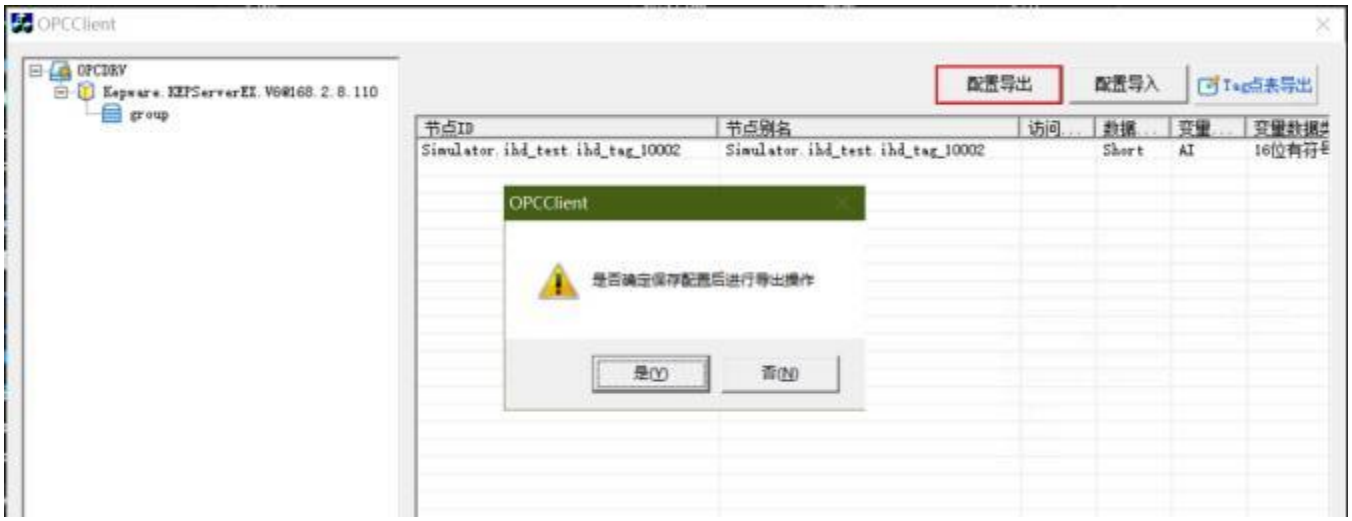
- ◇选择文件：选择文件保存位置和文件名
- 执行：执行导出点表的操作。
- 取消：取消执行导出点表操作。
- 退出：退出点表导出页面。

导入/导出配置文件（xls）

上述配置完成之后，点击“配置导出”，弹出确认对话框，点击“是”弹出文件选择及保存位置对话框，选择好保存位置和文件名之后点击确定，将OPCDRV配置导出到本地保存。

默认导出的文件名为：

opcdrv.xml



打开OPCClient.exe，点击界面右上角的“配置导入”，弹出选择文件的对话框，选择好之前保存好的OPCDRV配置xls文件，点击“确定”之后，开始导入OPCDRV配置到OPCClient工具。如果文件格式没有问题，则导入成功。如果文件错误，会有相应的提示。

在宝信基础自动化软件中配置OPC驱动

OPC驱动可以通过手动配置或是导入配置的方式来做配置。

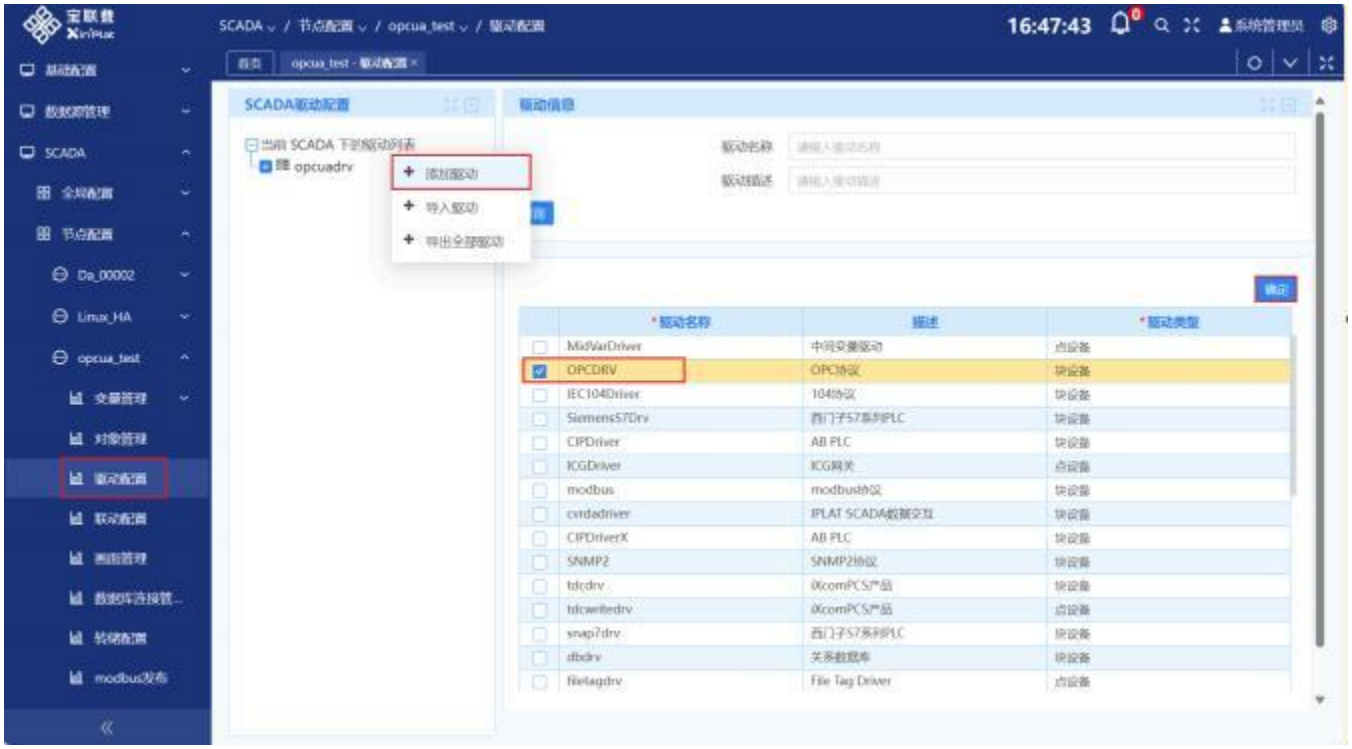
手动配置

手动配置时，可以借助使用OPCClient配置OPC驱动章节中导出的opcdrv.xls中的配置信息来做opc驱动配置。

添加OPCDRV

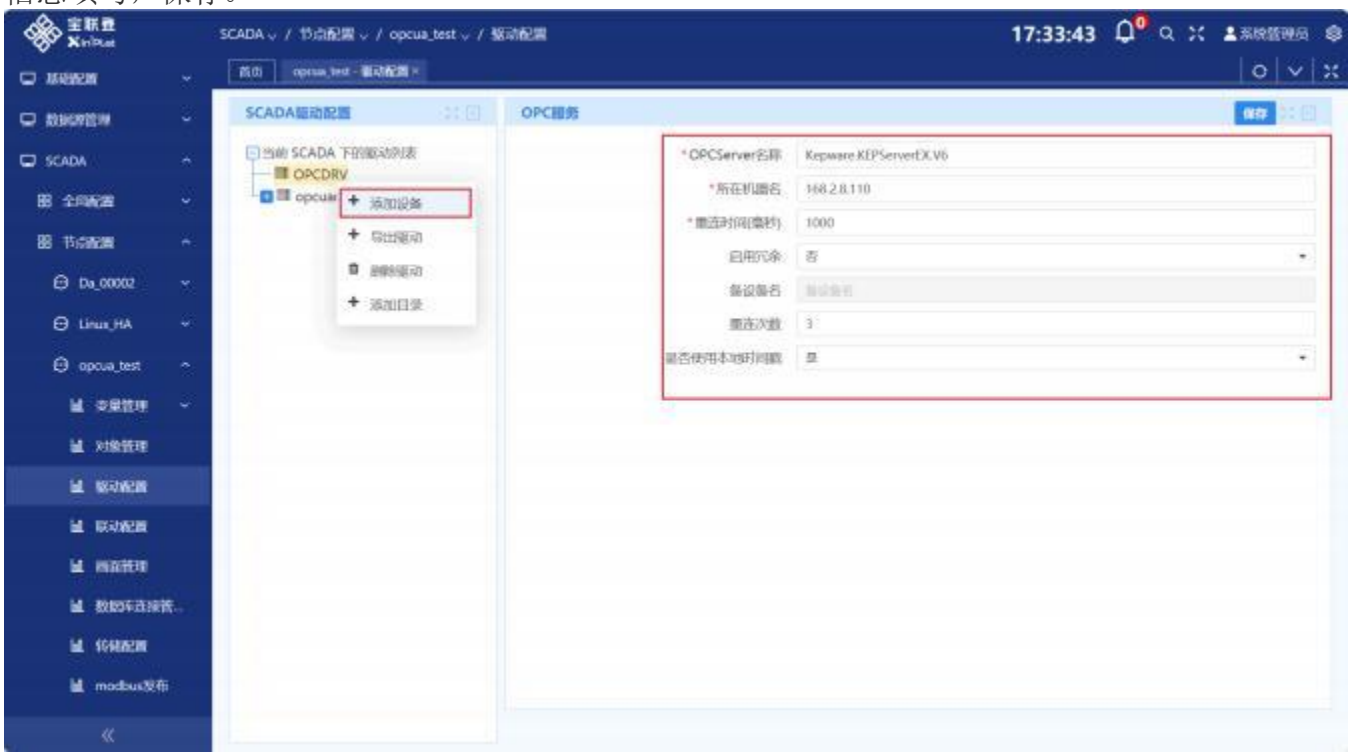
配置好相应的SCADA节点，点击SCADA节点下的驱动配置，进入驱动配置页面，在页面的左上方，右击当前SCADA下的驱动列表，可以看到“添加驱动”的按钮，点击添加驱动，在可用驱动列表

中，选择OPCUA驱动所对应的“OPCDRV”，然后单击右上方确定。



添加OPCDRV设备

在左侧的驱动列表中右击刚添加的OPCDRV驱动，选择“添加设备”，根据实际的的OPCServer信息填写，保存。

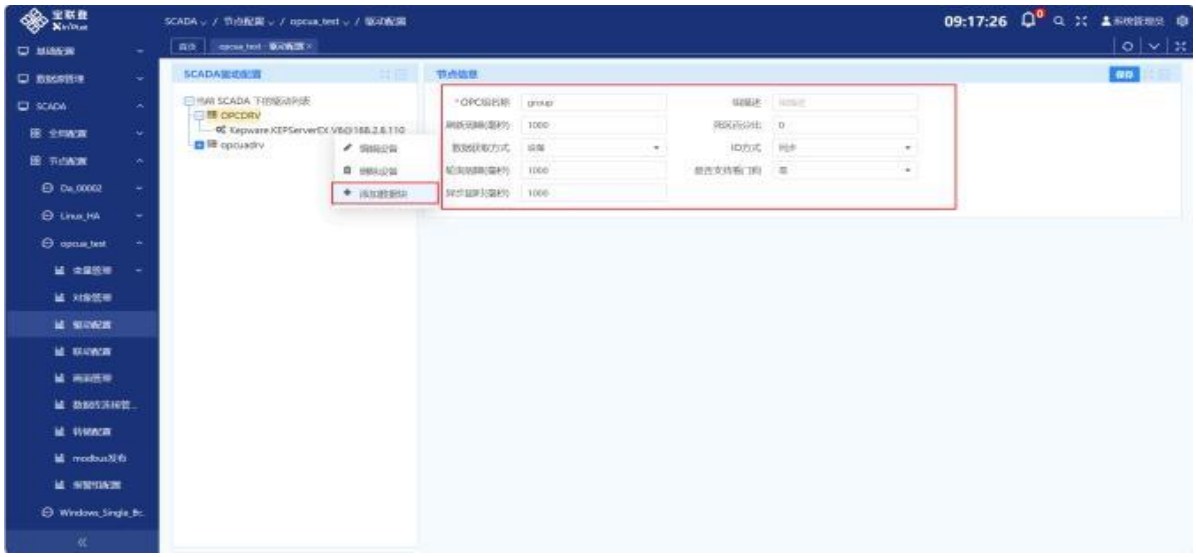


- OPCServer名称：不能为空，在1-31个字符
- 所在机器名：不能为空
- 重连时间：默认值1000
- 支持冗余：默认值否
- 备设备名：备OPCServer名称
- 重连次数：默认值3

•是否使用本地时间戳：默认值是

添加OPCDRV数据块（组）

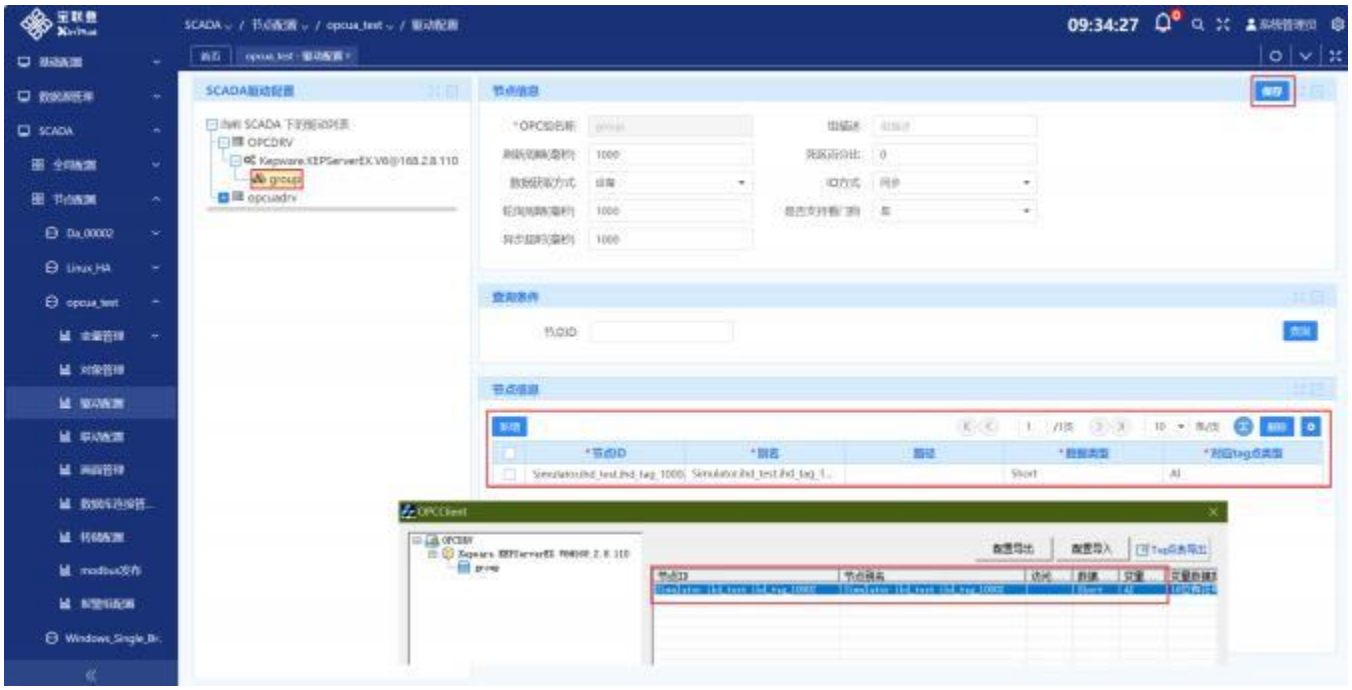
在左侧OPCDRV下右击已添加的OPCServer，选择添加数据块，修改OPC组名称，其他的按照实际情况填写，保存。



- OPC组名称：不能为空且不能重复。在1-31个字符。
- 组描述：在0-63个字符。
- 刷新周期（毫秒）：OPC服务端到设备读取数据的周期，在1ms-24H之间，必须是100的正整数倍。默认值：1000毫秒。
  - OPC客户端配置的刷新频率会被客户端传送给服务端，具体实现看服务端。
  - 为了避免丢点现象，如果用户知道设备的数据变化周期，建议将驱动的刷新周期配置为设备的数据变化周期的一半，点的扫描周期配置为数据变化周期的4分之一，如果使用到转储功能，建议将转储数据属性中的扫描周期配置为数据变化周期的8分之一，如设备的数据变化周期为4000毫秒，那么驱动刷新周期为2000毫秒，点的扫描周期为1000毫秒，转储数据的扫描周期为500毫秒。
- 死区百分比：OPC服务在接收请求时进行判断：和上一次获取的数据值相比，波动范围在死去百分比之内变化时，仍然发送上次的数据值，如第一次获取的数据值为100，死区百分比设为2%，第二次获取的数据值为98~102，那么仍然发送100。
- 数据获取方式：OPC数据获取有缓存和设备两个选项，用户可以根据自己的需求进行选择，区别如下：
  - 缓存：从缓存上获取数据比设备的读取速度更加的快。
  - 设备：从设备上获取的数据比缓存中获取到的新。
- IO方式：包含异步和同步两个选项。
- 轮询周期（毫秒）：只在I/O方式为同步读取时可配，OPC客户端去服务器读数据的周期。在1ms-24H之间，必须是100的正整数倍。默认值：1000毫秒。
- 是否支持看门狗：只在I/O方式为异步读取时可配，作用是在一定周期（异步超时）内OPC服务未返回数据，则将变量的质量置为未确定（Uncertain）。
- 异步超时（毫秒）：只在I/O方式为异步读取时可配，在1ms-24H之间，必须是100的正整数倍。值必须大于轮询周期。默认值:1000毫秒。

添加OPCDRV节点

点击已添加的OPCDRV数据块（组），在右侧下方的“节点信息”区域“新增”节点信息，新增完成后，点击右上角的“保存”按钮。（可以参照OPCClient新增节点章节最终的配置来填写）。



- 节点ID：OPCServer中实际的节点地址。
- 别名：节点别名的功能等同于节点ID，如果配置了节点别名，节点别名将会替代节点ID，节点ID将无效。如果节点ID的长度超过64个字节，会自动显示为红色，帮助用户识别修改。
- 路径：由设备提供，默认为空。如果修改访问路径，则用户必须保证路径的正确性。
- 数据类型：Word、Short、DWord、Long、Byte、Char、Float、Double、Boolean、String。
- 对应tag点类型：AI、AO、BLOB、CA、DI、DO、LA、MDI、TXT

导入配置

将OPCClient导出的opcdrv.xls导入到宝信基础自动化软件中。
在导入opcdrv.xls前，需要对xls文件做简单的修改。

⚠务必注意，所添加的列跟已有列之间不得有空列

Sheet页	列名	数值	备注
OPC服务	是否使用本地时间	0,1	0：不使用本地时间戳，1：使用本地时间戳
OPC服务	驱动ID	6	
OPC组	驱动ID	6	
节点	驱动ID	6	

在驱动配置页面，右击“当前SCADA下的驱动列表”选择导入驱动，选择opcdrv.xml。

在宝信基础自动化软件中配置OPC变量

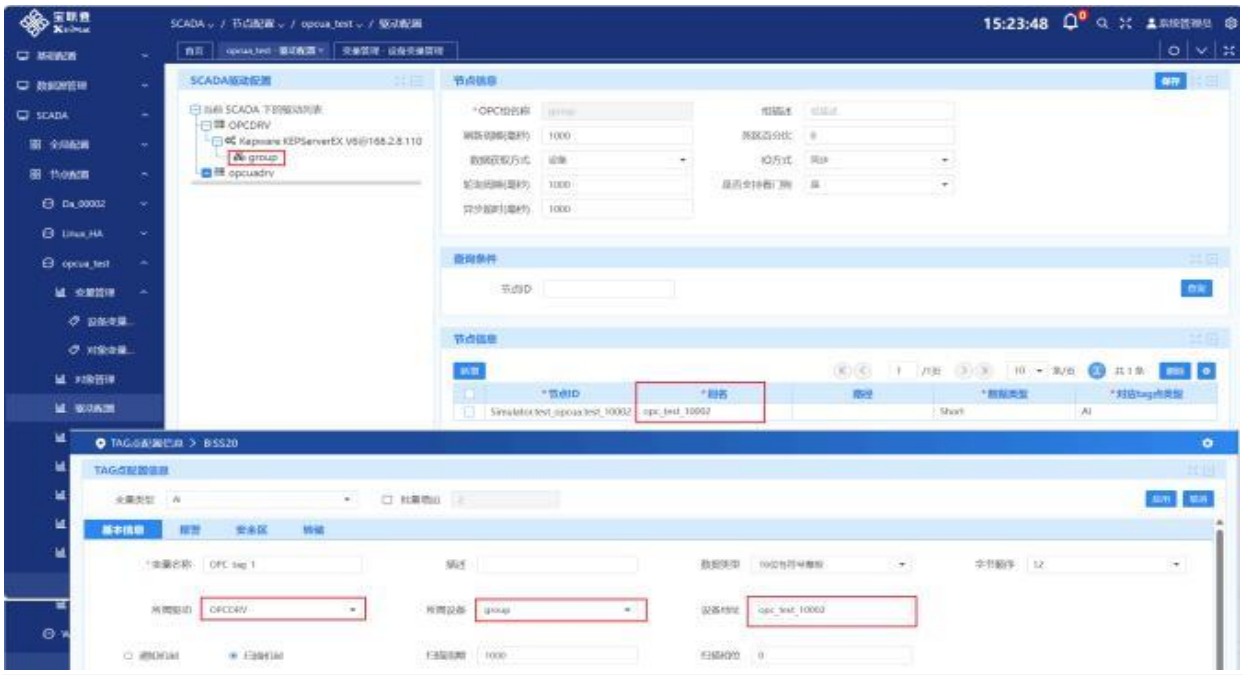
导入OPC变量

- 通过导入的方式配置OPC变量，需要从宝信基础自动化软件中先配置一个点，然后导出一个最新的Excel表格模板；
- 将OPCClient导出的旧点表（以下简称“旧点表”）中的数据按照列和Sheet工作簿一一对应的填充到最新的Excel表格模板（以下简称“新点表”）中，复制完成后，按照先前配置的
  点，填充其他列，确保所有必须的列都已填充相应的数据，保存新点表；

•参照Excel导入变量，选择导入方式后导入OPC变量。

手动配置OPC变量

在宝信基础自动化软件中创建OPC变量，相对与其他设备变量，需要注意的是“所属驱动”、“所属设备”和“设备地址”。



•所属设备：OPC驱动里面的数据块（组），详情查看添加opcdrv数据块组。

•设备地址：OPC节点信息中节点别名。

⚠⚠⚠导入OPC变量、手动创建OPC变量完成后需要部署到scada节点，才能使得配置生效。

驱动配置文件

```
<?xml version="1.0"encoding="GB2312"?>
<DriverCfg name="OPCDRV"devGroupCount="1">
  <DevGrpCfg
    name="Kepware.KEPServerEX.V6"mainsvr="168.2.8.110"reconnecttime="1000"isbacku
    p="0"<Description/>
    <DeviceCfg
      name="group"refreshrate="1000"percentdb="0"pollrate="1000"astimeout="1000"w
      atc<Description></Description>
      <ItemCfg
      name="Simulator.test_opcua.test_10002"aliasname="opc_test_10002"datatype="S
      hort"
        <Description/>
      </ItemCfg>
    </DeviceCfg>
  </DevGrpCfg>
</DriverCfg>
```

⚠⚠⚠使用OPCDRV驱动进行采集时，同一个OPCServer下采集的每个组点数建议不要超过2000个，否则可能导致性能下降和OPCServer的不稳定。

- 添加modbus驱动
- 配置modbus设备
- 变量管理-设备地址格式

modbus驱动配置主要包含以下步骤：

添加modbus驱动

右键scada驱动列表中选择添加驱动，在可选驱动列表中勾选modbus驱动，点击确定按钮。

配置modbus设备

1. 右键modbus，选择添加设备，如下图

3. 在右侧设备信息中填写设备相关参数：

❖以TCP方式采集的设备按照以下方式配置

参数说明：

- 设备名称：不能为空，不能以数字开头；
- IP地址：欲连接设备的IP；
- 端口号：欲连接设备使用的端口号，默认为502；
- 连接类型：选择TCPClient；
- 响应超时：连接设备和发送数据的响应超时时间；

- 协议类型：选择Modbus/TCP；
- 备机地址：备设备的IP；
- 备机端口号：欲连接备设备使用的端口号；
- 描述：自定义的设备描述信息；
- 是否支持多连接：该设备是否支持多连接；
- 任务号：同一任务号下的设备消息放在同一个进程中；
- 扩展参数：用户自定义；
- 多寄存器命令：用写多寄存器（或多线圈）命令去实现写单寄存器（或单线圈），默认0为不勾选状态，选1则为勾选成功状态。

配置modbus数据块

注意：modbus驱动支持免配块方式。若选择免配块方式，则无需进行本章操作，相应的变量配置参见免配块方式下的变量设备地址配置。若采用配数据块方式，具体操作参见本章说明，其变量配置参见配块方式下的变量设备地址配置。

1. 选中已经配置的设备，右键设备，选择"添加数据块"，弹出数据块信息配置界面，如下图：

- 数据块名称：自定义的块名。
- 地址：设备地址。
  - DO（线圈）:000000-065535
  - DI（触点）:100000-165535
  - AI（输入寄存器）:300000-365535
  - AO（输出寄存器）:400000-465535

  通过地址来区分信号类型。（注意这里的地址比常见的地址多加了一个0，例如AO地址为40001，需要配置地址400001）

  注：这里的AI,AO,DI,DO并非指IPLATBA中的点类型，而是指modbus中的类型，配置时需要区分清楚。
- 元素个数：该数据块的长度,即，用户要在数据块中保存的查询记录的最大条数。

  对于AI、AO而言，为寄存器个数，最大为125，推荐最大使用120；（注:某些设备一次请求能处理的寄存器个数会小于125，如果按照最大寄存器个数配置，设备可能不回复数据）注：元素个数填写，不用最大值的意义在于，当如FLOAT类型4个字节跨数据块时，可使用预留的部分进行两个数据块间的调整。数据块长度限制，是由于ModBus协议的通讯包限制为256bit。

  对于DI、DO而言，为线圈个数，最大为1000；
- 相位：轮询周期中的相位。
- 任务号：不同任务号的块将使用不同的线程进行采集。

•轮询周期：扫描数据块的周期。
•元素大小[字节]：用户查询的每条记录的元素大小。
对于AI、AO而言，为寄存器大小，输入2即可；（注:AI和AO类型的元素大小不能小于2，如果填入数据小于2，程序将会按照寄存器大小等于2来处理）
对于DI、DO而言，输入1即可；
a.数据块的长度=元素个数*元素大小。
b.配置数据块长度的时候需注意，Modbus的数据块长度是指个数，数据块总大小不得超过250个字节。如配置浮点型变量，每个浮点型变量需占用4个字节，所属驱动是Modbus驱动，那么数据块长度最大能配62个*4字节=248字节，第63个的数据块的长度为63个*4字节=252字节，超过了250最大限制，则无效。因此在工程实施时，到块的边界时，如果有变量占用的字节超过了一个块，可能会导致该变量读取数据不对，这个需要工程实施人员注意，在驱动的数据块配置时，对照实际点表，合理设定数据块的长度，同样在配置变量的时候，我们应该选择具有4个字节的单精度浮点型。
c.blob类型长度最大250。
•设备站号：modbus设备的站号。
注意：以ASCII方式向modbus驱动的BLOB类型变量写入时要大于2个字节。
•描述：自定义的描述信息。

2.配置完毕后，点击"保存"按钮，完成数据块配置。

变量管理-设备地址格式

•注意：由于MODBUS协议是以寄存器为单位写的，因而双字寄存器拆分使用的变量在一个扫描周期内连续对同一寄存器进行写入会存在遗漏风险。因此保守起见，此种情况下，建议Tag点要配置成只读。以下列举了此情况下的Tag点配置：
◦DI/DO变量使用如400001:0、300001:15、DataBlock0:b161；
◦AI/AO变量可使用8位无符号/有符号整型等，具体根据modbus设备侧信号表数据类型而定；
◦TEXT/BLOB变量起始地址和长度跨字。

免配块方式变量设备地址配置

•DI/DO:
格式：寄存器地址:位地址
例如：
300001:15表示读取Modbus寄存器地址300001的第15个bit；
400001:0表示读取Modbus寄存器地址400001的第0个bit；
000001表示读取Modbus寄存器地址000001的第0个bit；
•AI/AO：
◦8位无符号整型、8位有符合整型：
例如：400001:0、400001:8
400001:0代表取400001的第个一字节，400001:8代表取400001的第二个字节；
◦16位无符号整型、16位有符合整型：
例如：400001，读取一个完整400001寄存器；
◦32位无符号整型、32位有符合整型、单精度浮点型：
例如：400001，表示读取400001和400002两个寄存器；
◦双精度浮点型：
例如：400001，表示读取400001～400004四个寄存器。
（注:驱动设置免配块时，AI、AO点自动组块的最大寄存器个数为125。这是按照modbus协议规定的标准设定的，但某些设备实现modbusserver时一次请求能处理的寄存器个数会小于125，如果按照最大寄存器个数配置，设备可能不回复数据。
现场使用免配块时，出现免配块配置大于125个AI点数据无法读取的情况，请确认是否是设备不支持一次报文读取125个寄存器的原因造成的）

- TEXT/BLOB:

  格式：起始地址#长度（字节），如果不配置长度，默认长度为250。
  例如：400001#16，表示从400001开始读取16个字节。
  变量配点的时候可以支持"站号@地址"的形式，例如1@400001，表示站号为1的地址为400001的变量；
  没有配站号的话，默认为站1，用户想配站2的400001地址的话，格式为2@400001。
  注意：
  - 选择免配块方式，每个数据块最大250bytes。
  - 选择免配块方式，对于某些modbus设备，如果某个地址的数据读取失败，会导致该地址后的数据按块读取时失败，这种情况建议采用配块方式，规避这些读取失败的地址。

  例如：某些设备（数据段是固定的），寄存器地址1～125，中间的90～100寄存器是空的。请不要采用免配块方式，请配置两个数据块分别对应1～89和101～125。

配块方式变量设备地址配置

- DI变量/DO变量的配置：

  设备地址格式为：数据块名:b偏移量。（b代表以位为单位进行偏移）
  例如：Datablock1:b61，表示读取数据块Datablock0中的第62位。
- AI变量/AO变量配置：

  设备地址格式为：数据块名:B（W）偏移量。（B代表以字节为单位进行偏移,W表示以字为单位进行偏移）

  例如：Datablock0:B23，表示从数据块Datablock0中的第24字节开始读取，读取的长度根据AI变量的数据类型自动选取，8位整形读取1个字节，16位整形读取2个字节，32位整形和单精度浮点型读取4个字节，双精度浮点型读取8个字节。
  Datablock0:W23，表示从数据块Datablock0中的第24字开始读取，读取的长度根据AI变量的数据类型自动选取，由于W单位是字，一个字是2个字节，所以配置的数据类型不能是8位的整形，从16位整形开始读取1个字，32位整形和单精度浮点型读取2个字，双精度浮点型读取4个字。
- TEXT、BLOB：

  设备地址格式为：数据块名:偏移字节#字节长度
  例：DataBlock0:2#50
  表示从数据块DataBlock0的第3字节开始读取50字节的长度。
  注：变量配点的时，也支持W，以16位为一个字，也支持按字去采集，一个字是两个字节。两个字节一个字，在采集string或者blob的时候会用到。

modbus.xml

```
<?xml version="1.0"encoding="GB2312"?>
<driver name="modbus"type="block"desc="">
<device
name="ACSX_Device0"conntype="TCPClient"connparam="ip=168.2.9.229;port=502;port2=502;
<device
name="BAS_Device0"conntype="TCPClient"connparam="ip=10.3.17.21;port=502;multiLink=1;
</driver>
```

- 添加IEC104Drv驱动
- IEC104Drv设备配置
- IEC104Drv数据块配置
- IEC104Drv变量配置
- 102指令定值召唤功能

IEC104驱动是宝信基础自动化软件与遵循IEC104规约的设备进行通讯的驱动程序，功能包括读取104规约设备遥信、遥测、电度量等数据，下发控制给遥控、遥调点，从而实现在iPlat平台上对IEC104规约设备数据的上传及下发功能。

使用IEC104驱动主要包含以下步骤：

添加IEC104Drv驱动

配置好相应的节点。

点击节点下的驱动配置，进入驱动配置页面，右击页面的左上方的"当前SCADA下的驱动列表"，点击添加驱动，在右下方的"可选驱动列表-CC10"中勾选IEC104Drv，点击"保存"以完成添加"IEC104Drv"驱动。

IEC104Drv设备配置

1. 在驱动配置界面上，右键点击驱动列表中的，选择"添加设备"菜单，在树控件列表的右侧显示新添加设备的配置界面，设备的名称自动生成，其他属性为默认值。如下图：

参数配置的意义如下：

- 设备名：自定义的设备名称。
- 所属设备组：当一个工程中，需要连接不同工艺组的众多设备时，可以用"设备组"来进行分组管理。如果所连设备不多，也可归为一个设备组。**IEC104驱动会将一个设备组下的设备放在一个线程中执行，若服务器性能足够，推荐一个设备组下只配置一个设备以获得最高性能，推荐是设备组个数不超过CPU核数的2倍**

- 设备描述：自定义的设备描述信息。
- 主IP地址：IEC104主设备的IP地址。
- 启用备机：下拉框选择1后可以对IEC104备设备的设备IP地址进行配置。
- 公共地址：实际设备的公共地址，缺省1。
- 发送最大数目：发送方未被确认的I格式帧的最大数目。
- 接收最大数目：接收方未被确认的I格式帧的最大数目。
- 端口号：IEC104设备连接的端口号。
- 响应超时（秒）：与设备的连接和发送时的响应超时时间。
- 重连间隔时间（秒）：重连设备间隔时间设定，单位秒。
- 时钟同步周期（秒）：IEC104驱动与实际设备进行时钟同步的周期。
- 设备总召唤周期（秒）：定期召唤电度量的定时周期；总召唤过程中，设备不上传电度量。
- 电度量召唤周期（秒）：定期召唤电度量的定时周期。
- 发送或测试电文超时时间（秒）：驱动向设备发送测试电文等待反馈所需超时时间。
- 无数据报文确认时间（秒）：当驱动在此时间内没有收到设备发送的数据时，发送测试帧。
- 空闲状态发送测试帧时间（秒）：空闲状态时发送测试帧时间。
- 首次总召唤延迟时间（秒）：在与设备连接后，驱动等待一段延时后，再发送总召唤。
- 从站计数有效：是否对从站发送的计数进行校验，校验不成功会自动断掉TCP连接。
- 电度量召唤冻结：需要用户按照实际使用的设备需要，选择是否勾选该选项。（某些厂商的设备在读取电度量时需要先发送冻结指令，不然读取到的值不变，当遇到此类问题且又无法向厂商确认时可以勾选该参数进行尝试）
- SOE数据更新到变量：默认勾选，表示允许SOE数据更新到宝信基础自动化软件的变量，不勾表示不允许。
- 多连接：指设备是否支持多连接，勾选后主备机将同时对设备进行采集。
- 遥测符号转换：上传的遥测数据是否要进行符号转换。
- 重试次数：发送超时后重送的次数。
- 连接超时时间：建立TCP/IP连接的超时时间。

IEC104Drv数据块配置

1. 在设备信息配置完成后，点击保存。在左侧驱动列表中会自动出现设备名，右键点击设备，选择添加数据块，如下图：

- **数据块名称**：自定义的数据块名称。
- **数据块描述**：自定义的数据库描述信息。
- **数据块类型**：该数据块对应到设备上的数据类型。包含以下数据类型：单点遥控量、双点遥控量、整形遥测量、浮点型遥测量、电度量、分接头位置、遥调量、遥信量、归一化遥测、双点遥信量。
- **数据块起始地址**：该数据块的起始地址。
- **数据块长度**：该数据块的长度（字节）。

IEC104Drv变量配置

添加变量参见变量管理快速新建变量。

注意：配置完成后需要部署到边缘节点，使得配置生效。

设备地址配置

设备地址配置为对应的104设备点的实际地址。

注意：该地址必须属于已经配置的块。比如，有一个块，起始地址为16300，长度为40，则16300~16339均为可使用的地址。

在IPLATBA5.1.2版本后支持SOE点，只需要在设备地址开头加上SOE（不区分大小写），该点仅会更新驱动收到的soe报文中的内容，方便用户展示SOE报警和保存SOE历史数据。

变量类型选择

不同类型的数据块需要配置所对应的数据类型，否则可能读取到不正常的数据，数据类型对应关系：

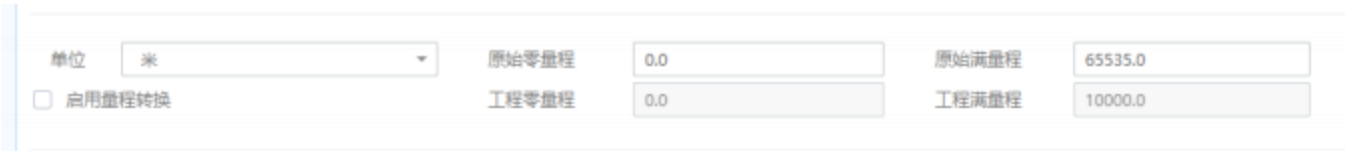
- 单点遥控量、双点遥控量、遥调量：配置成8位无符号整型；

注意：对于这三种变量，在进行控制时，需要分成2个步骤执行

1. 选择要执行的操作。
2. 稍等一会后执行操作

具体的操作如下：

- **单点遥控和双点遥控量（下行）**，此变量下方指令执行后，设备会将成功与否的返回值写入此变量8位中的最高位，成功为1，不成功为0。
  - 执行"分/开"操作，先写2，再写0（写2表示选择了"分/开"操作，写0表示执行"分/开"操作）
  - 执行"合/关"操作，先写3，在写1（写3表示选择了"合/关"操作，写1表示执行"合/关"操作）
  - 撤销"分/开"选择操作，写6
  - 撤销"合/关"选择操作，写7
- **遥调量（下行）**
  - 执行"档位向下降"操作，先写2，再写0（写2表示选择了"档位向下降"操作，写0表示执行"档位向下降"操作）
  - 执行"档位向上升"操作，先写3，在写1（写3表示选择了"档位向上升"操作，写1表示执行"档位向上升"操作）
  - 撤销"档位向下降"选择操作，写6
  - 撤销"档位向上升"选择操作，写7
- 分接头位置、遥信量（上行）：配置成8位无符号整型。
- 浮点型遥测量（上行）：配置成单精度浮点型。
- 电度量、整型遥测量（上行）：配置成32位整型。
- 归一化遥测（上行）：配置成单精度浮点型。启用量程转换，原始量-1~1，工程量-32767～32767。如下图：



- **对标度化值、归一化值以及短浮点遥测的设定（写值）操作步骤**

1. 配置一个TXT的变量用来下发控制，驱动配置为104，设备地址配置为要设定的遥测点的地址，不要忘记勾选允许控制；

2. 向该变量下发控制指令：格式为"类型标识:动作:值"
   - 类型标识为整形数字，遵从104协议中的类型标识，十进制数字48、49、50，分别代表归一化、标度化以及短浮点
   - 动作有以下几种：执行为0，选择为1，撤销选择为2
   - 值的取值范围分别为-1到1的浮点数、无符号short整形、单精度浮点型。比如向地址为1793的归一化值发送值为0.5的选择指令，向该TXT变量写入"48:1:0.5"；成功后，再向该地址发送执行指令，写入"48:0:0.5"，或者在选择成功后要撤销选择，写入"48:2:0.5"。

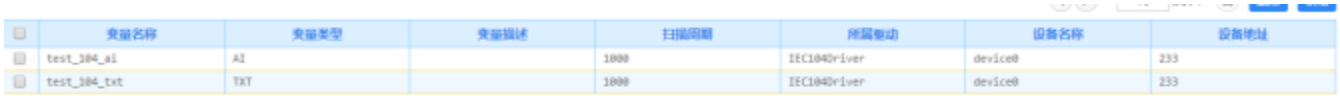
102指令定值召唤功能

可以通过102指令定值召唤方式立即获取实时值，适用于小批量、临时性数据读取。大批量数据建议使用总召方式。

具体配置方法如下：

1. 驱动中为相应的地址配置一个整型遥测量的数据块；

2. 变量配置配置两个，一个为txt一个为AI 16位无符号；



3. 需要发送定值召唤指令时，向txt变量写入"102"即可，驱动会相应地址发送定值请求指令，只有一次下发指令，没有先选择后执行的过程；

设备收到指令会，会上传该地址相应值，类型为11；

4. 下面是修改驱动后，与PMA模拟发送和接收的电文主站发送

68 0d 02 00 00 00 66 01 05 00 01 00 e9 00 00

起始字节=68数据单元长度(APDU)=13 I格式帧发送序号(NS)=1接收序号(NR)=0 TI=102
VSQ=01 SQ=0 INFONUM=1 COT=05 T=0 PN=0 CAUSE=5 COA=1 C_RD_NA_1

读数据命令肯定认可请求或被请求点号=233从站发送

68 10 00 00 06 00 0b 01 05 00 01 00 e9 00 00 23 01 00

起始字节=68数据单元长度(APDU)=16 I格式帧发送序号(NS)=0接收序号(NR)=3 TI=11
VSQ=01 SQ=0 INFONUM=1 COT=05 T=0 PN=0 CAUSE=5 COA=1 M_ME_NB_1

带品质描述的测量值,标度化值肯定认可请求或被请求

变化有效IV=0当前值NT=-44171264未被取代SB=0未被封锁BL=0点号=233 OV=0未溢出值=291

收到设备上传的电文后，test变量值为291；

IEC104Driver.xml

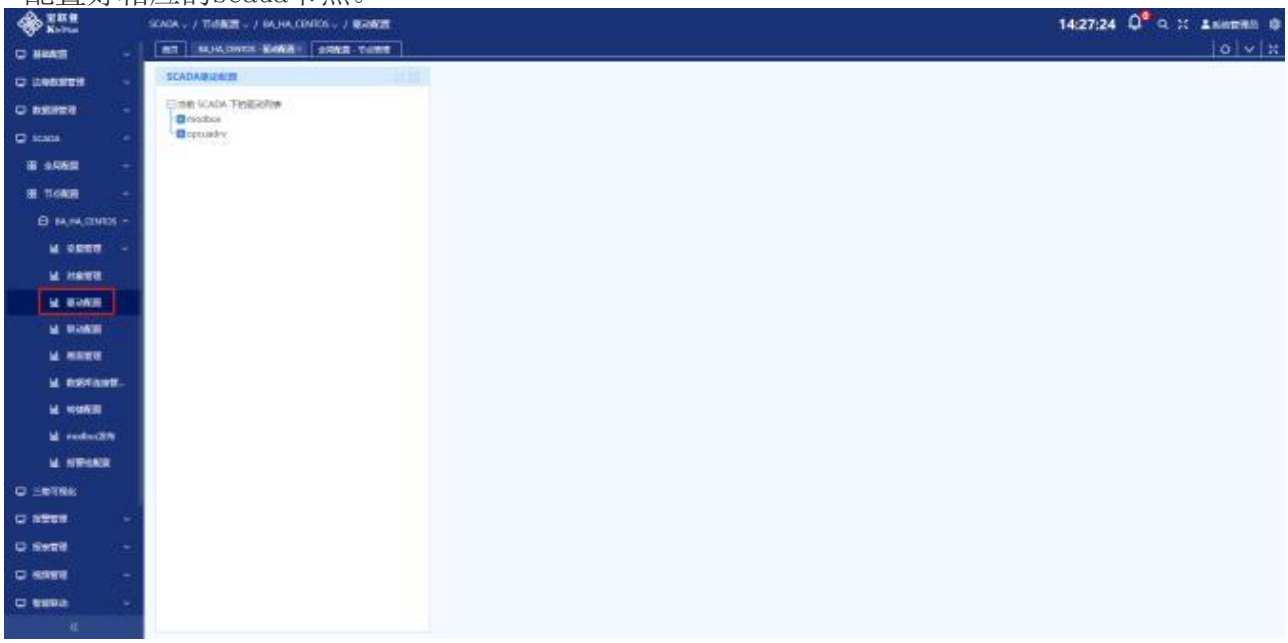
```
<?xml version="1.0"encoding="GB2312"?>
<DriverCfg name="IEC104Driver"devGrpCount="1">
    <DevGrpCfg name="trans104"devCount="1"description="">
                                                                <DeviceCfg
      name="trans104"multiLink="0"soeupdate="1"dataBlockCount="10"ip="168.2.238.2
<DataBlockCfg name="dandianyaokong"blockType="0"startaddr="0"length="1"description
<DataBlockCfg name="shuangdianyaokong1"blockType="1"startaddr="10"length="1"descri
<DataBlockCfg name="zhengxingyaoce"blockType="2"startaddr="20"length="4"descriptio
<DataBlockCfg name="fudianxingyaoce"blockType="3"startaddr="30"length="4"descripti

<DataBlockCfg name="dianduliang"blockType="4"startaddr="40"length="4"description="
<DataBlockCfg name="fenjietouweizhi"blockType="5"startaddr="50"length="1"descripti
<DataBlockCfg name="yaotiaoliang"blockType="6"startaddr="60"length="1"description=
<DataBlockCfg name="yaoxinliang"blockType="7"startaddr="70"length="1"description="
<DataBlockCfg name="guiyihuayaoce"blockType="8"startaddr="80"length="2"description
<DataBlockCfg name="fudianxingyaoshe"blockType="0"startaddr="90"length="4"descript
        </DeviceCfg>
    </DevGrpCfg>
</DriverCfg>
```

SNMPV2.0驱动该驱动是基于SNMPV2协议使用批量读值接口读取目标设备指定的值，
相较于SNMPDRV驱动，解决了因配置大量的点而可能出现的读值慢的情况。

使用SNMP驱动主要包含以下步骤：

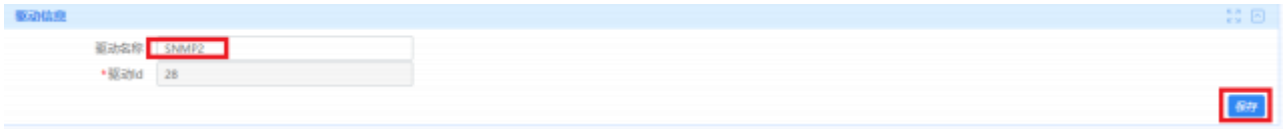
添加SNMPV2.0驱动

1. 配置好相应的scada节点。



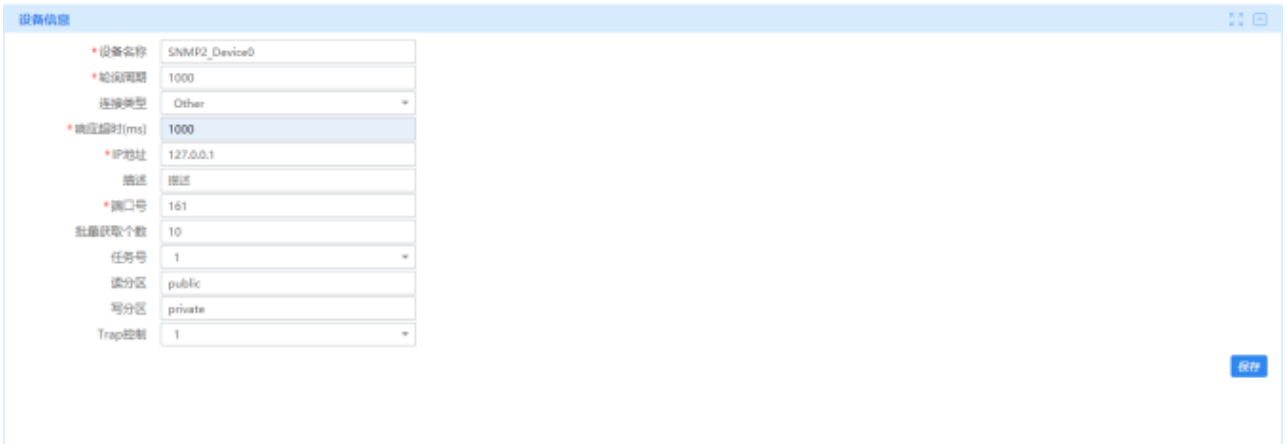
2. 点击SCADA节点下的驱动配置，进入驱动配置页面。右键点击**【当前SCADA下的驱动列表】**添加驱动弹出可用驱动的选择对话框，
在右侧显示的**【驱动信息】**页面中，在**【可选驱动列表】**中勾选"SNMP2"，点击保存，完成驱动添加。

SNMP2驱动添加成功后，点击**【保存】**按钮。



在【当前SCADA下的驱动列表】**中，右键点击**【SNMP2】，选择添加设备。

弹出SNMPV2.0设备配置界面。



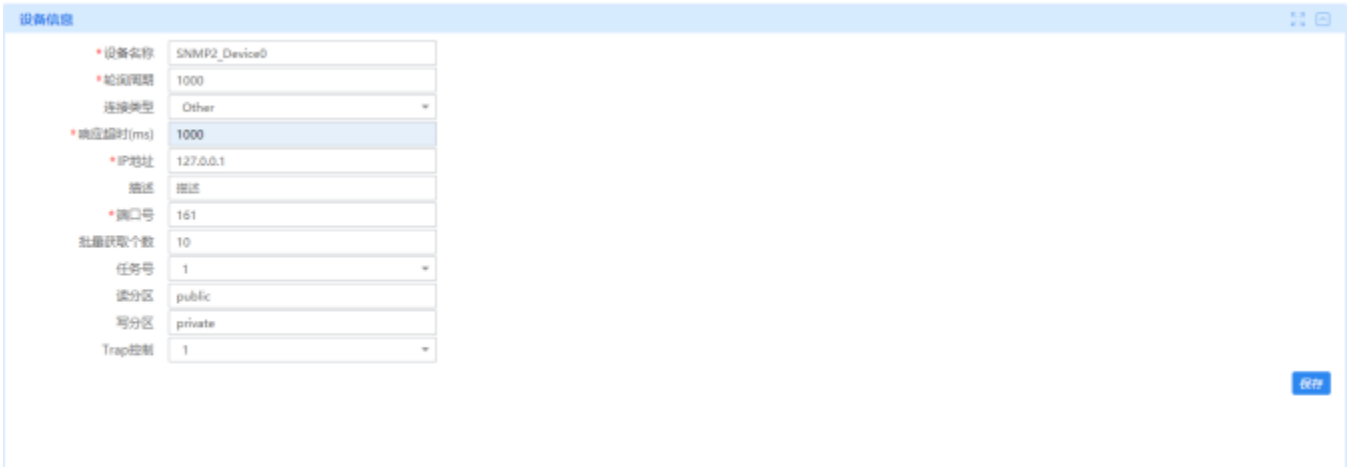
注意：

- *在启用Trap功能之前，需要确认下需要的端口是否已经被占用（Trap服务使用的默认端口为162，端口查看cmd命令：netstat-ano），若启用时该端口已经被占用，snmp2驱动不会反复尝试去监听，需要在排除占用后重启snmp2驱动。*

- *这里的Trap端口指的是服务器接收Trap信息端口，如果有任何设备开启Trap功能的话这里需要填写Trap端口，一般情况下为162，若没有填写端口，则服务默认开启162端口（即使没有任何设备开启Trap），修改完成之后需要点击右边的"应用Trap端口"按钮，不然直接签入部署该端口配置不会保存。如果配置本驱动时修改了Trap端口配置，则部署时必须要选择重启服务方式，不然修改后的端口不会在服务中生效。*
- *在启用Trap功能之前，需要确认下需要的端口是否已经被占用（端口查看cmd命令：netstat-ano），若启用时该端口已经被占用，snmp2驱动不会反复尝试去监听，需要在排除占用后重启snmp2驱动。*

配置SNMPV2.0设备

在设备列表处点击鼠标右键，选择**【添加设备】**选项，系统弹出设备配置窗口，填入相关配置项，如下图：



参数说明：

- 设备名称：不能为空，不能以数字开头
- 轮询周期：读取对应值得周期时间
- 连接类型：固定为Other
- 响应超时：连接设备和发送数据的响应超时时间。
- IP地址：欲连接设备的IP地址。支持主备设备冗余连接功能，主IP地址用"/"与备用IP分隔。写法如上图："168.2.237.100/168.128.0.20"。有主/备IP配置时，驱动会缺省连接左侧主IP地址，连接不上时，会尝试连接右侧备IP地址。
- 描述：自定义的设备描述信息。
- 端口号：欲连接设备使用的端口号，默认为161。
- 批量获取个数：每次读取，一次性读取设备内数据块的个数。优点：一个报文一次可以读取多个数据块，轮询获取数据速度快；缺点：假设此处设定值为"10"，其中1个读不到值，那么其他9个数据块也将读不到值。

注意：某些设备不支持批量读取功能，此处参数需要设为"1"。

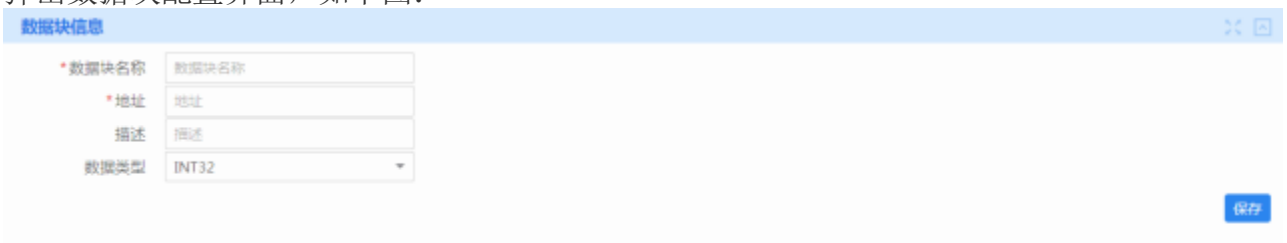
- 任务号：同一任务号下的设备消息放在同一个进程中。
- Trap控制：开启/关闭Trap机制，效果是当目标设备的值变化时会自动推送，此时轮询仍然按照配置的时间进行。默认Trap服务使用的端口为162
- 读分区：读分区名。
- 写分区：写分区名。

注意：读分区、写分区是SNMP特有的用来区分设备权限分区的，具体配置请咨询厂家。

配置SNMPV2.0数据块

1. 选中已经配置的设备，在需要配置数据块的设备上点击右键，选择【添加数据块】选项，

弹出数据块配置界面，如下图：



- 名称：自定义的块名
- 类型：取到值得类型，根据选择的类型自动分配元素个数，
  INT32,UINT32,GAUGE32,COUNT32为4，OCTET，OID为255，TIMETICKET为16
- 地址：此处填写目标OID的值（同一设备下请不要配置相同地址的数据块,有的第三方的采集工具在采集OID的时候会默认在前面加一个点，宝信基础自动化软件是默认不加点的）
- 描述：自定义的描述信息
- 任务号：不同任务号的块将使用不同的线程进行采集

2.配置完毕后，点击"保存"按钮，完成数据块配置。

配置SNMPV2.0变量

所有类型的变量配置：

变量配置分为普通配置与自主块配置。

普通配置：
配置SNMP驱动的变量时，变量地址为SNMP驱动配置的"数据块名称"。

- 模拟量：包括AI/AO变量。选择"32位无符号整型"或"32位有符号整型"，可对应到SNMP中的数据类型：int32、uint32、counter32、gauge32。
- TXT变量和BLOB变量：可对应到SNMP中的数据类型：octet、oid、timeticks。自主块配置：

自主块配置无需在驱动配置中添加数据块，变量地址填目标OID地址，其他与普通配置相同。

```
<?xml version="1.0"encoding="GB2312"?>
<driver name="SNMP2"type="block"desc=""param1="162">
<device
name="SNMP2"conntype="Other"connparam="ip=168.2.237.88;port=161;"cyclerate="1000"r
ec
```

<datablock
name="COUNTER32"type="COUNTER32"address="1.3.6.1.2.1.2.2.1.10.7"elemcount="4"el
<datablock
name="GAUGE32"type="GAUGE32"address="1.3.6.1.2.1.2.2.1.5.1"elemcount="4"elembyt
<datablock
name="INT32"type="INT32"address="1.3.6.1.2.1.1.7.0"elemcount="4"elembytes=""cy
<datablock
name="OID"type="OID"address="1.3.6.1.2.1.1.2.0"elemcount="255"elembytes=""cycl
<datablock
name="TIMETICKS"type="TIMETICKS"address="1.3.6.1.2.1.1.3.0"elemcount="16"elemby
<datablock
name="TXT"type="OCTET"address="1.3.6.1.2.1.1.1.0"elemcount="255"elembytes=""cy
<datablock
name="UINT32"type="UINT32"address="1.3.6.1.2.1.2.1.0"elemcount="4"elembytes="
"
</device>
<device
name="SNMP2_1"conntype="Other"connparam="ip=168.2.8.110;ip2=168.2.8.29;port=161;"c
ycl
<datablock
name="COUNTER32"type="COUNTER32"address="1.3.6.1.2.1.2.2.1.10.1"elemcount="4"el
<datablock
name="GAUGE32"type="GAUGE32"address="1.3.6.1.2.1.2.2.1.5.1"elemcount="4"elembyt
<datablock
name="INT32"type="INT32"address="1.3.6.1.2.1.1.7.0"elemcount="4"elembytes=""cy
<datablock name="OID-
1"type="OID"address="1.3.6.1.2.1.1.2.0"elemcount="255"elembytes=""cy
<datablock
name="TIMETICKS"type="TIMETICKS"address="1.3.6.1.2.1.1.3.0"elemcount="30"elemby
<datablock
name="TXT"type="OCTET"address="1.3.6.1.2.1.1.1.0"elemcount="255"elembytes=""cy
<datablock
name="UINT32"type="UINT32"address="1.3.6.1.2.1.2.1.0"elemcount="4"elembytes="
"
<datablock name="test_int32"type="INT32"address="1.3.6.1.2.1.4.24.2.极长的OID

地址被截断，请检

iCG驱动用于从iCentroGate设备上采集数据的驱动。注意：ICG驱动已经支持底层设备的通道冗余。

使用iCG网关驱动主要包含以下步骤：

symlinkapi版本注意事项

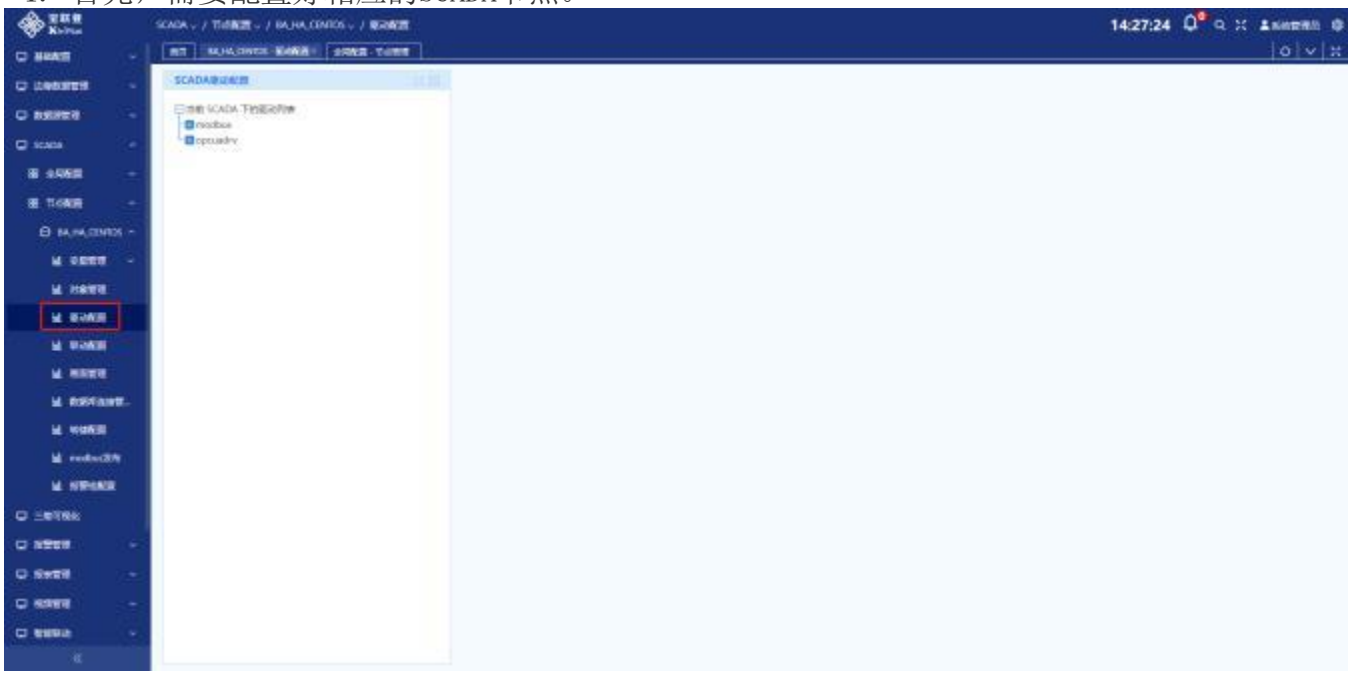
iplatba版本	symlinkapi版本	ICG网关软件版本	备注
3.0.0~5.1.0	2.0.0.1	2.10.0.7137及此版本之前的版本	
	2.3.0.1	2.10.0.7300及此版本之前的版本	

备注：

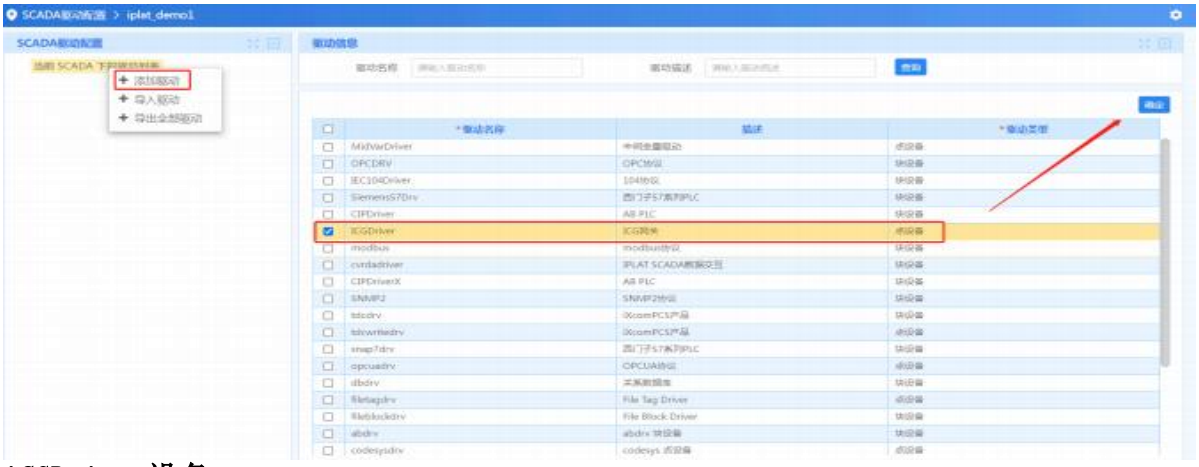
1. 新版本symlinkapi是向上兼容的，支持老版本icg网关软件
2. 使用旧版本symlinkapi不会发生连接不上的情况，但是会发生数据错乱，因此要确认好ICG网关上运行的软件版本

添加iCGDriver驱动

1. 首先，需要配置好相应的SCADA节点。

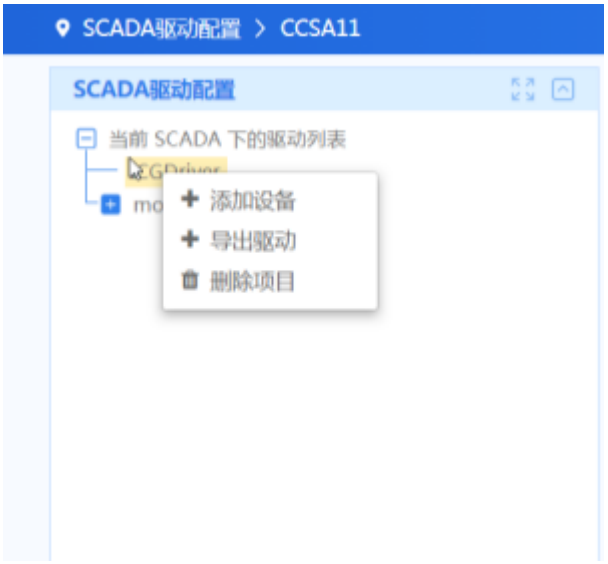


2. 单击SCADA节点下的驱动配置，进入驱动配置页面，右键单击"当前SCADA下的驱动列表",点击添加驱动，在可用驱动列表中，勾选ICGDriver驱动所对应的"ICGDriver",然后点击保存完成驱动新增。



配置iCGDriver设备

1. 选择ICGDriver点击鼠标右键，选择"添加设备"选项，如下图：



2. 系统弹出设备配置窗口，填入相关配置项，如下图：

必填参数：

❖设备名称：不能为空，不能以数字开头

❖连接参数为：配置格式如下：

连接参数：主设备主链接/主设备备链接/备设备主链接/备设备备链接

每个链接格式为：ip,端口,用户名,密码,

其中密码可以为空，但是密码与用户名之间必须加个逗号，表示存在密码但是密码为空。

主设备主链接是必须的，其他可以不填，例如"link=168.2.239.5,9230,usr1,pwd1,"

如果设备支持多连接，可以配置成

"link=168.2.237.5,9230,usr1,pwd1,;multiLink=1;"multiLink=1表示设备支持多连接；

multiLink=0表示设备不支持多连接；

不配置的话默认单连接。

如果需要使用异步写值方式（不等待设备返回），可以配置成"link=168.2.237.5,9230,usr1,pwd1,;multiLink=1;asyncwrite=1;"

asyncwrite=1表示使用异步方式，asyncwrite=0或者不写该参数表示使用同步方式icg版本2.10以及以后的版本支持同一个用户名和密码多连接。

❖任务号：同一任务号下的设备消息放在同一个进程中。

❖响应超时：等待设备返回消息的最长时间。

3.点击保存按钮，完成设备配置。

配置ICGDriver数据块

注意：ICGDriver驱动支持免配块方式，若该设备下没有配置任何块，相应的变量配置参见免配块方式下的变量配置。

对于ICG驱动而言，配置块与免配块方式最大的区别在于配置块时可以设置别名，并用别名作为设备地址进行tag点的配置。

1. 选中已经配置的设备，右键"添加数据块"，如下图：

2.选择"添加数据块"，弹出数据块配置界面，如下图：

❖地址：对应到ICG中的点地址串，如下图：

红框标记的ICG中的点的地址为：ChnOPC.D.S.G.Simulation_Items_Random_Int4

❖别名：非必填项，自定义的ICG点的别名。

❖长度：根据数据类型填写长度，int为4个字节、双精度double为8个字节、string是根据实际长度填写，不可超过255个字节，BLOB长度为255个字节。

❖描述：表示用户在数据块希望保存的变量记录数。

❖数据类型：目前ICG支持int、double、string和BLOB类型，如果数据要做控制，该项为必填项。

其他配置项使用默认项。

3.配置完毕后，点击"保存"按钮，完成数据块配置。

配置ICGDriver变量

添加变量,需要注意：

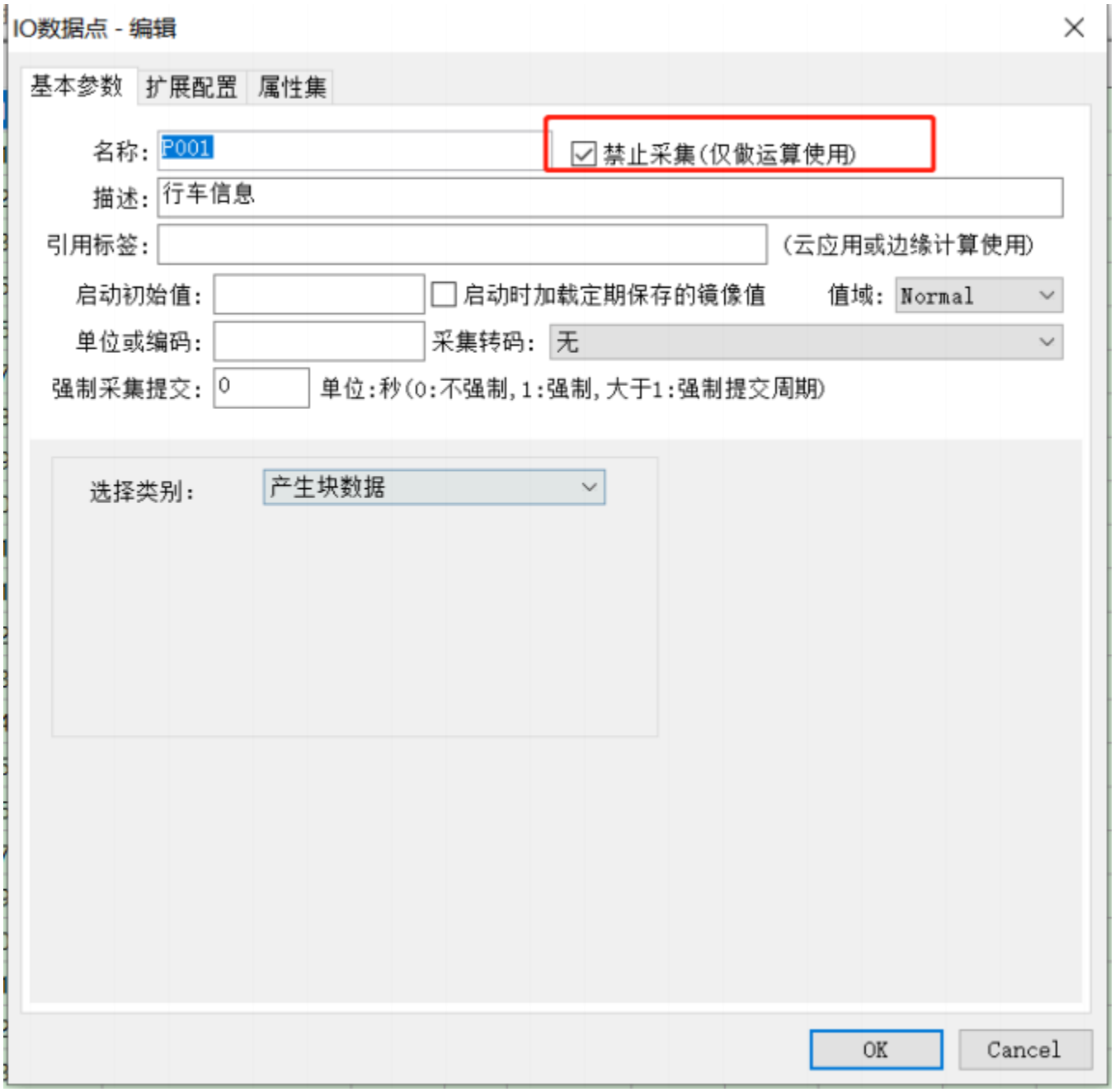
❖配置完成后需要部署到scada节点，使得配置生效。

❖如果发现变量的值刷新慢，可以检查是否有错误的ICG点地址，如果存在错误的地址，则在读取这些变量时，会等待超时，从而影响其他变量的值的读取，使得整体的数据读取性能下降。

❖在配置变量的时候，icg同一个地址只能配置一种数据类型，如果之前这个点配置的数据类型是txt,则再配置blob类型的话会导致控制命令发送不下去，只能将之前的txt变量删除才可以。

免配块方式下的变量配置

在iCGDriver不配置块时，配置变量的设备地址填入实际ICG中的点地址，如下图：



要读取红框标记的ICG中的点，则在变量中配置变量地址如下图：

配块方式下的变量配置

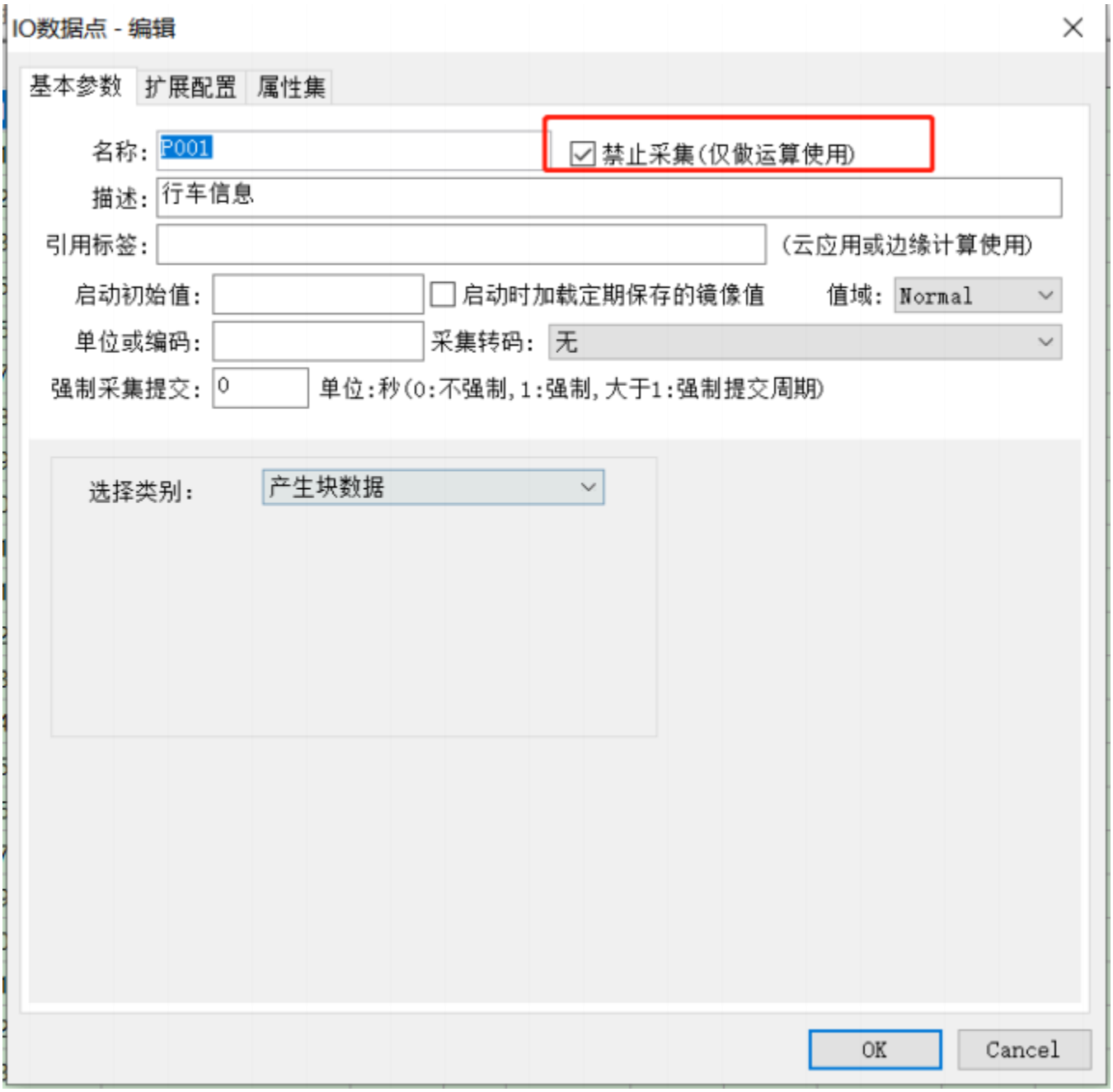
在ICGDriver配置了块时：

◇若没有设置别名，则变量中的配置同免配块方式下的变量配置。

◇若设置了别名，则变量中的设备地址输入设置的别名。

注意：对于宝信基础自动化软件变量类型的配置，不论是免块方式配置，还是配置数据块的方式配置，宝信基础自动化软件中变量的类型仅支持int32、double、TXT、BLOB这几种类型。

如果遇到配置正确但读值失败的情况，请注意ICG设备工程中该点是否勾选了禁止采集，如果勾选了需要取消。如下图所示：



如何实现ICG底层通道冗余

我们最多支持4个设备冗余，下面以四个设备为例子，配置一个通道冗余：

配置ICG驱动，添加设备，如下图：

"连接参数"中填写四个ICG设备的IP，第一个为主设备，其他以"/"为分隔；

格式："link=168.2.239.5/168.2.239.6/168.2.239.7/168.2.239.8,9230,usr1,pwd1,"

ICG侧配置：

1.要保证通道上有Iostatus和Io_Work这两个点。

ICG特殊点配置说明：

不配Io_Work点：Iostatus代表网关和通道下设备的心跳和数据通信都正常；

配置Io_Work点：Iostatus仅代表心跳正常，Io_Work代表数据通信正常。

注意：IoStatus是icg内置点，整型，无需配置，Io_Work这个点需要添加并配置，可以是int或double。

2.四个ICG设备要配置相同的通道，相同的TAG点，以保证tag名相同；

宝信基础自动化软件的ICG驱动监听Iostatus和Io_Work两个状态点的过程：

首先，读主设备主链接通道下设备的状态IoStatus。如果是0，再去读通道下设备的Io_Work，如果是0，才读取主设备主链接设备上数据，否则的话依次读取另外3个设备。如果四个icg对应通道下设备上IoStatus和Io_Work都不为0，则数据质量为bad。

至此，通道冗余配置完成。

ICG获取与宝信基础自动化软件的通讯状态

1.在ICG中创建可写点IO_heart；
2.宝信基础自动化软件定时向ICG中IO_heart点写变化的数据；
3.通过ICG端看IO_heart点是否持续变化来判断和宝信基础自动化软件的通讯是否连通。

当icg检测到宝信基础自动化软件不更新Io_Heart这个点时，icg不向底层设备采集数据。

主要错误码：

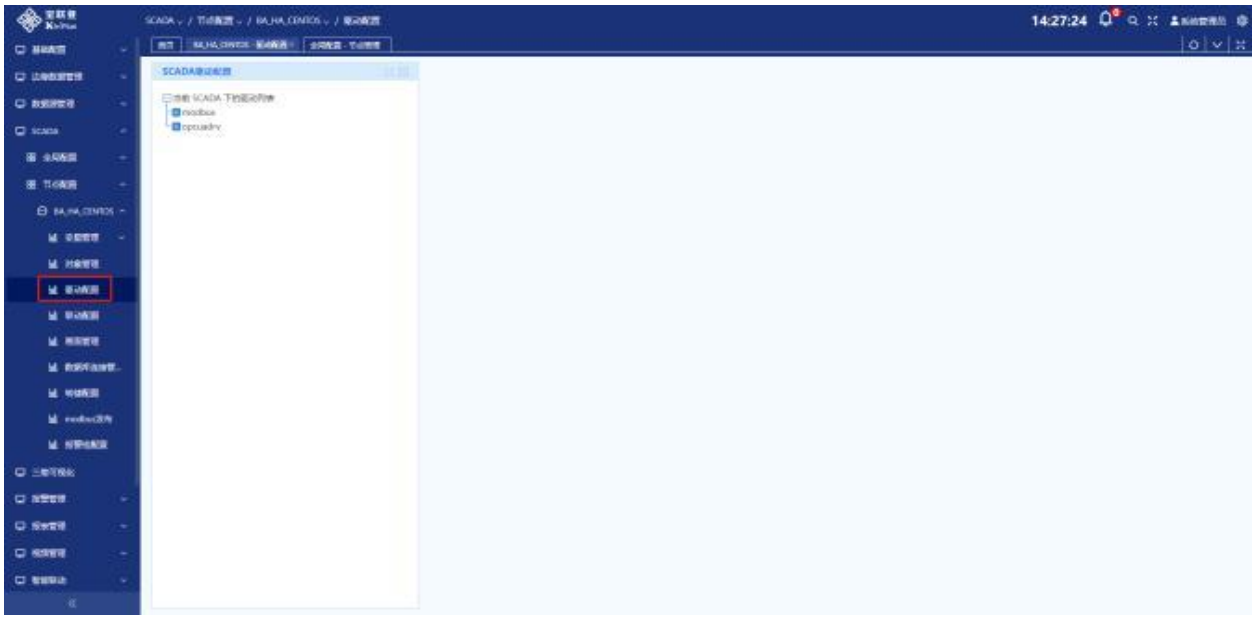
错误码	描述
-1	命令超时
-2	解压缩失败
-3	数据版本改变
-4	未连接
-5	未完成初始化
-6	数据未找到
-7	参数错误
-8	调用队列溢出
-9	调用超时

-10	调用返回错误
-11	功能被禁止

```
<?xml version="1.0"encoding="GB2312"?>
<driver name="icgdriver"type="item"desc="">
                                                                    <device
name="ICG_BLOCK"conntype="Other"connparam="link=168.2.237.222,9230,forscadaicm2,fo
                                                                        rsca
    <datablock
    address="newmodbus.modbus.blob_5"name="blob"elemcount="128"elembytes=""desc="
    "
                                                                    <datablock
   address="newmodbus.modbus.bool_10"name="bool"elemcount="1"elembytes=""desc=""p
                                                                    <datablock
  address="newmodbus.modbus.float_10"name="float"elemcount="8"elembytes=""desc=""
                                                                    <datablock
  address="newmodbus.modbus.int16_10"name="int16"elemcount="2"elembytes=""desc=""
                                                                    <datablock
  address="newmodbus.modbus.int32_10"name="int32"elemcount="4"elembytes=""desc=""
                                                                    <datablock
   address="newmodbus.modbus.int8_10"name="int8"elemcount="1"elembytes=""desc=""p
                                                                    <datablock
   address="newmodbus.modbus.string_5"name="string"elemcount="128"elembytes=""desc
                                                                    <datablock
  address="newmodbus.modbus.uint16_10"name="uint16"elemcount="2"elembytes=""desc=
                                                                    <datablock
  address="newmodbus.modbus.uint32_10"name="uint32"elemcount="4"elembytes=""desc=
                                                                    <datablock
  address="newmodbus.modbus.uint8_10"name="uint8"elemcount="1"elembytes=""desc=""
  </device>
                                                                    <device
name="icg_noblock"conntype="Other"connparam="link=168.2.237.222,9230,forscadaicm1,
                                                                        fors
</driver>
```

添加geniitekdrv驱动

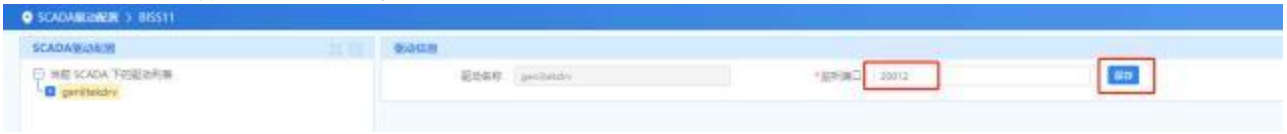
1. 在SCADA管理下的对应节点，点击驱动配置



2. 在驱动配置页面中，点击添加驱动

3. 勾选geniitekdrv驱动，再点击确定

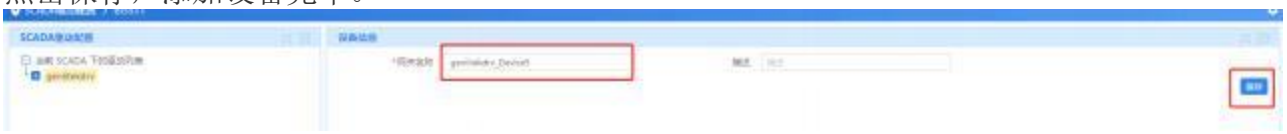
4. 填写驱动信息，geniitekdrv驱动是tcp server模式，这里需要填写统一的端口号，添加完成后再点击保存，添加驱动完成。



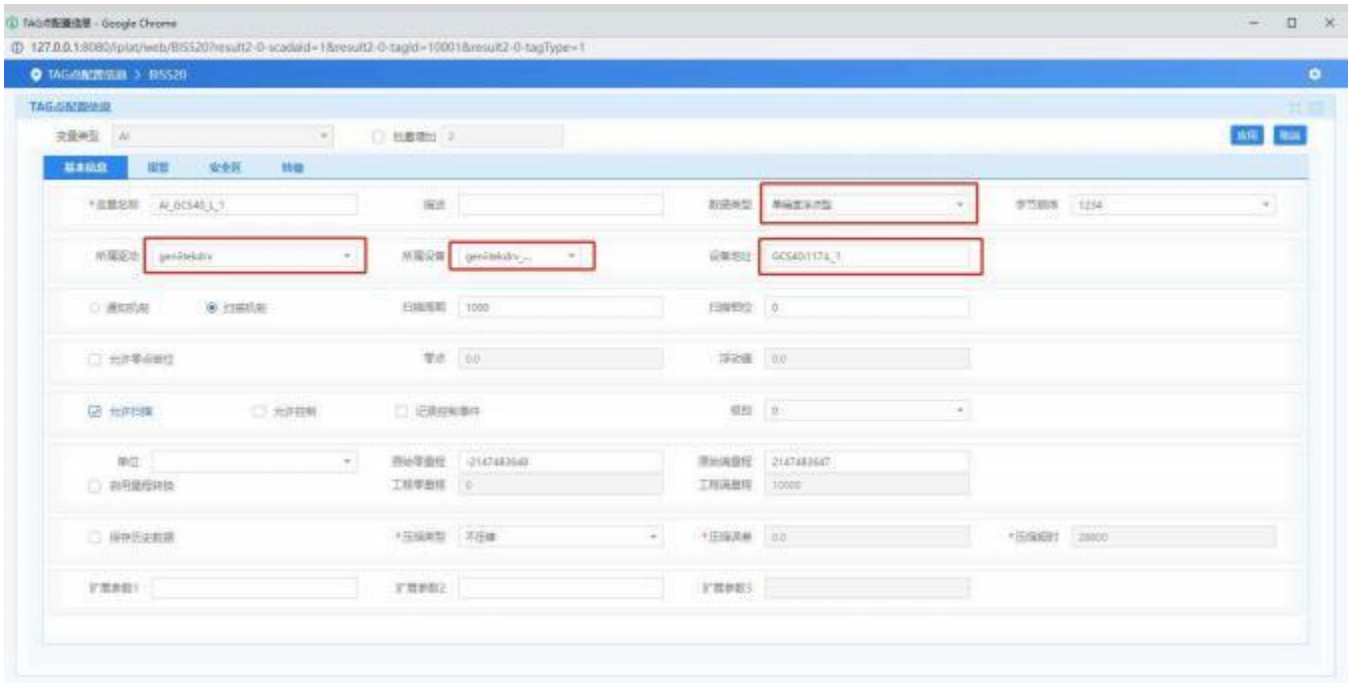
添加geniitekdrv设备

1. 右键点击添加设备

2. 在设备信息页面，填写驱动设备名称，这里的设备指的是网关，描述可以不填，填写完成后点击保存，添加设备完毕。



添加变量



添加变量步骤与一般设备相同，这里主要描述变量关键信息:

所属驱动:选择geniitekdrv;

所属设备:选择geniitekdrv对应设备;

数据类型:理论上可以选择任意数据类型，驱动会将实际接收数据转成配置的类型，但还是建议根据报文里的数据类型来配置;

设备地址:由设备类型、设备传感器ID(10进制数字)和设备标签组成(大小写不敏感)，在实际配置

时，需要将设备机身信息的设备类型和传感器ID按正确方式填写到这里。

各个设备标签的数据类型:

设备标签	**数据类型**
电流	单/双精度浮点型
温度	单/双精度浮点型
电压	单/双精度浮点型

设备标签	数据类型
油位	16/32位有符号整型
休眠时间	16/32位有符号整型
频率	单/双精度浮点型
加速度	单/双精度浮点型
速度RMS值	单/双精度浮点型
位移峰峰值	16/32位有符号整型
倾角	单/双精度浮点型

传感器类型英文缩写:

传感器类型	英文缩写
无线电流传感器(三相)	GCS40
无线温度传感器(防爆型)	GT40E
无线温度传感器	GT40
无线温振一体传感器	VB40
无线油位传感器	GLS40
无线振动传感器(防爆型)	VB40SAE
无线振动传感器	VB40SA

以下是设备地址详细示例:

设备地址	说明
VB40:4010:F_X	传感器类型:VB40，传感器ID:4010，频率x轴
VB40:4010:F_Y	传感器类型:VB40，传感器ID:4010，频率y轴
VB40:4010:F_Z	传感器类型:VB40，传感器ID:4010，频率z轴
VB40:5010:A_X	传感器类型:VB40，传感器ID:5010，加速度峰值x轴
VB40:5010:A_Y	传感器类型:VB40，传感器ID:5010，加速度峰值y轴
VB40:5010:A_Z	传感器类型:VB40，传感器ID:5010，加速度峰值z轴
VB40SA:3077:V_X	传感器类型:VB40SA，传感器ID:3077，速度RMS值x轴

设备地址	说明
VB40SA:3077:V_Y	传感器类型:VB40SA，传感器ID:3077，速度RMS值y轴
VB40SA:3077:V_Z	传感器类型:VB40SA，传感器ID:3077，速度RMS值z轴
VB40SA:5049:D_X	传感器类型:VB40SA，传感器ID:5049，位移峰峰值x轴
VB40SA:5049:D_Y	传感器类型:VB40SA，传感器ID:5049，位移峰峰值y轴
VB40SA:5049:D_Z	传感器类型:VB40SA，传感器ID:5049，位移峰峰值z轴
VB40SA:5049:R_X	传感器类型:VB40SA，传感器ID:5049，倾角x轴
VB40SA:5049:R_Y	传感器类型:VB40SA，传感器ID:5049，倾角y轴
VB40SA:5049:R_Z	传感器类型:VB40SA，传感器ID:5049，倾角z轴
GCS40:117:L_1	传感器类型:GCS40，传感器ID:117，交流电流1
GCS40:117:L_2	传感器类型:GCS40，传感器ID:117，交流电流2
GCS40:117:L_3	传感器类型:GCS40，传感器ID:117，交流电流3
GLS40:07:O	传感器类型:GLS40，传感器ID:07，油位
GT40:5016:T	传感器类型:GT40，传感器ID:5016，温度
GT40:5016:B	传感器类型:GT40，传感器ID:5016，电压
GT40:5016:S	传感器类型:GT40，传感器ID:5016，休眠时间

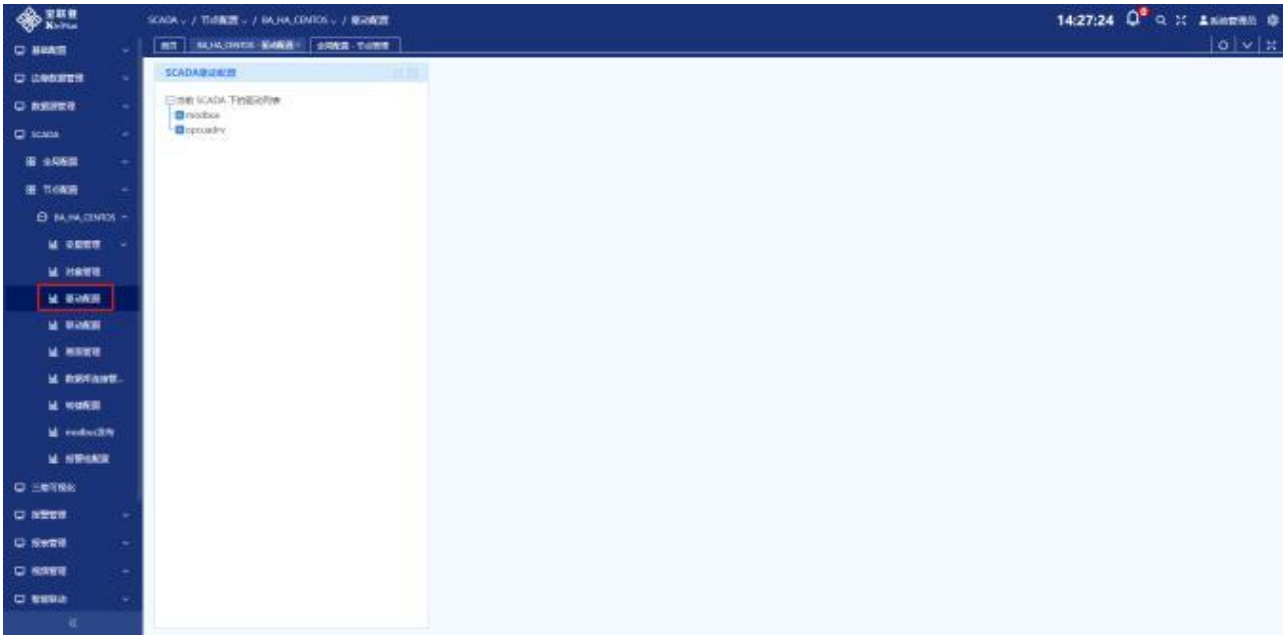
读取文件（filetag和fileblock）

注意：该驱动读取的目标文件的内容由用户自己维护，且需要在内容更新完毕后在同目录下生成一个同名后缀为.ok的文件，当该.ok文件存在时，不得再次对读取目标文件进行修改。

驱动会根据.ok文件是否存在判断目标文件是否可读，避免同时读写产生影响，且驱动会在一次读取完成之后将.ok文件删除，若读取时发现.ok文件不存在，则跳过此次读取请求。

驱动配置

1. 添加scada节点后，点击配置好的scada节点，在下拉列表中点击驱动配置。



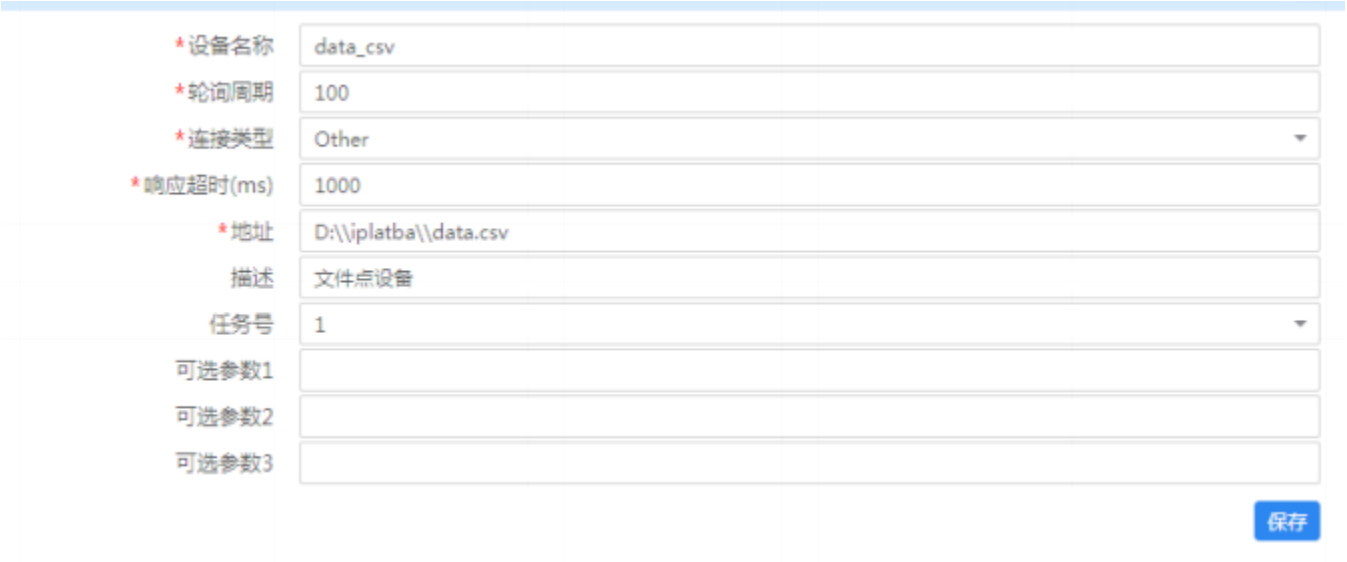
2. 在scada驱动配置中，右键驱动列表，点击添加驱动。

3. 进入系统驱动配置界面，按照红框中的设置添加filetagdrv或fileblockdrv驱动。

filetagdrv

filetagdrv为点设备，使用自主块方式进行配置，无需配置数据块

。设备配置：



轮询周期：读取文件内容的周期时间，单位毫秒

连接类型：固定为Other

响应超时：暂时保留，目前无用处，默认1000

地址：文件路径

可选参数1：默认不填或者填写e，表示只对文件读，填写d或者D表示删文件，填写t或者T表示改文件名称，带时间

TAG点配置：tag名和tag点地址需要与文件中需要读的值的标签名对应。

文件内容格式：仅两列，第一列为标签名，第二列为实际值

fileblockdrv

fileblock为块设备，必须要配置数据块

。设备配置：

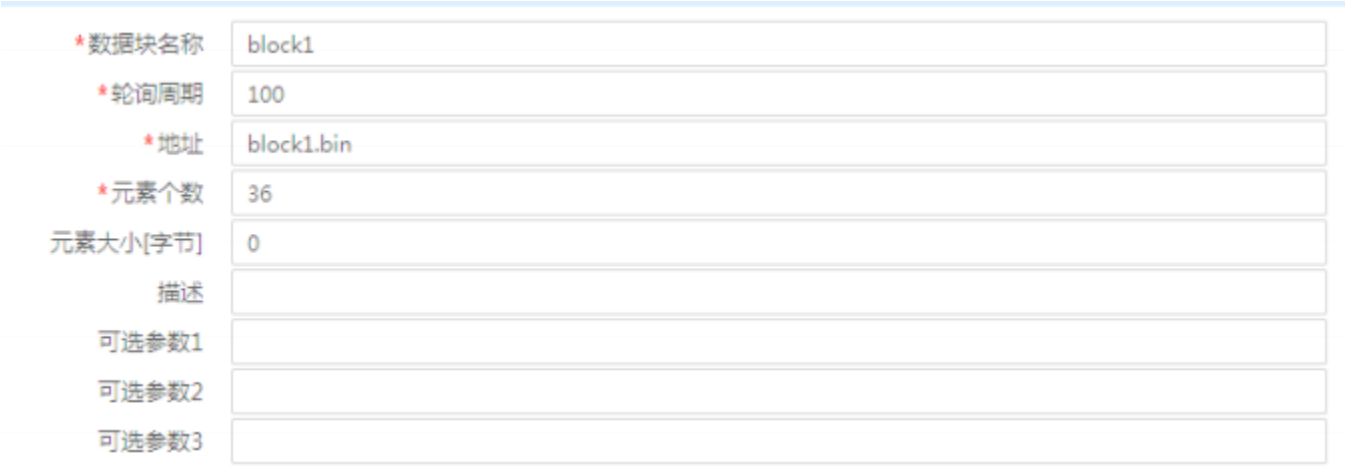
连接类型：固定为Other

响应超时：暂时保留，目前无用处，默认1000

连接参数：文件路径，填写到文件夹

可选参数1：默认不填或者填写e，表示只对文件读，填写d或者D表示删文件，填写t或者T表示改文件名称，带时间

数据块配置：



地址：文件名称

轮询周期：数据块读取周期

元素个数：为文件内容的字节数，需要严格相等，若读取出来的字节数与此处填写不一致，则会更新对应的点为BAD

元素大小：不填

TAG点配置：tag点地址填写数据块名，需要偏移则在块名后加冒号，例如Block1:2:3，该地址表示Block1的数据偏移2个字节再偏移3位，如果是TXT类型则在最后加上#长度。

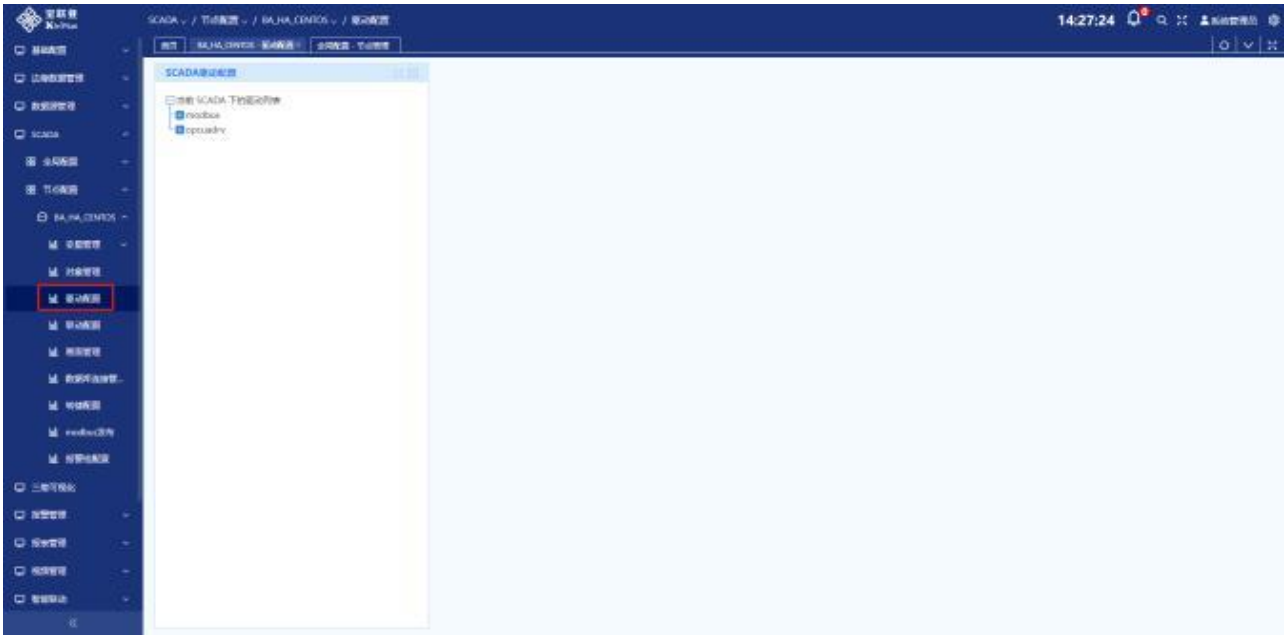
读取文件内容格式：仅数据

MidVarDriver不对应到设备，而用于配置中间变量，主要使用于一些内部值的传递。不与实际设备进行交互。

使用MidVarDriver的变量包含以下步骤：

添加MidVarDriver

1. 首先，需要配置好相应的SCADA节点。



2. 点击SCADA节点下的驱动配置，进入驱动配置页面。右键点击**【当前SCADA下的驱动列表】**添加驱动弹出可用驱动的选择对话框，在弹出的**【可选驱动列表】中，勾选MidVarDriver驱动所对应的"MidVarDriver"，点击【确定】**按钮。

配置MidVarDriver变量

MidVar驱动不需要添加设备

注意：

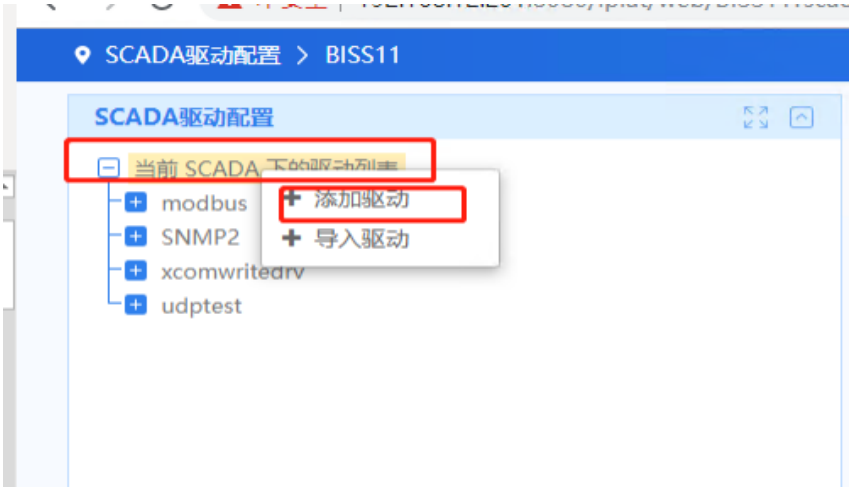
配置完成后需要**部署**到SCADA节点，使得配置生效。

选择【所属驱动】为MidVarDriver，【所属设备】【设备地址】不需要填写。数据类型可以选择任意类型，每种类型具体如何填写，请参考各类型的文档说明。

- 添加hitachidrv驱动
- 配置hitachidrv驱动

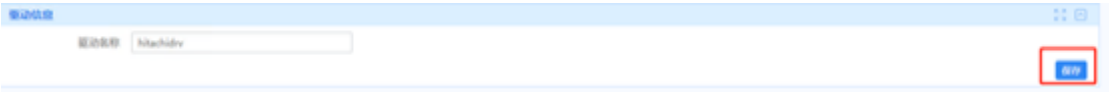
添加hitachidrv驱动

1. 首先，SCADA驱动配置中右键"SCADA下的驱动列表"，再单击"添加驱动"。



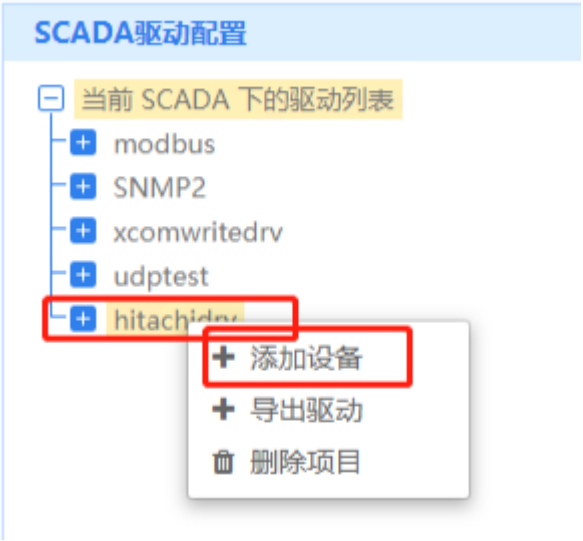
2. 点击"驱动名称"右侧"请选择驱动"对话框，弹出"可选驱动列表"，选中"驱动名称"为"hitachidrv"的行，若没有此驱动，点击左上角"新增"按钮，自行添加该驱动，输入"hitachidrv"，然后单击保存按钮。

3. 驱动添加完毕后，单击右侧保存按钮。



配置hitachidrv驱动

1. 选择"hitachidrv"右键点击"添加设备"



2. 右侧弹出设备信息，填入相关配置项，点击"保存"按钮，完成"hitachidrv"设备添加，如下图:

必填参数：

❖设备名称：自定义，不能为空，不能以数字开头

❖连接参数：localport=xxxxx;ip=xxx.xxx.xxx.xxx:port,其中localport是本地端口,ip为PLC设备的IP地址，port为PLC设备的端口

❖轮询周期：扫描数据块的周期

❖响应超时：连接设备和发送数据的响应超时时间

❖连接类型：选择UDPClient

❖任务号：不同任务号的块将使用不同的线程进行采集，如果驱动配置了多个PLC设备，建议不同的设备分配不同的任务号

3. 点击"hitachidrv_Device0"右键点击"添加数据块"

4. 右侧弹出"数据块信息"，填入相关配置项，点击"保存"按钮，完成"hitachidrv"数据块添加，如下图：

参数说明：

❖数据块名称：自定义，不能为空，不能以数字开头

❖轮询周期：扫描数据块的周期

❖响应超时：连接设备和发送数据的响应超时时间

❖类型：byte

❖地址：数据块的起始地址

❖任务号：不同任务号的块将使用不同的线程进行采集，如果驱动配置了多个PLC设备，建议不同的设备分配不同的任务号

❖元素个数：数据块的元素个数（元素个数*元素大小等于读取到的数据块的实际长度。数据块的实际长度=整体报文长度-2字节TelLen-2字节TelID）

❖元素大小【字节】：数据块中一个变量所占的字节长度，通常为"1"

❖可选参数1：为电文号TelID

数据偏移说明（tag点配置注意事项）：

❖数据类型：按照工程项目的需要填写

❖字节顺序：根据PLC的实际情况填写

❖设备地址：

◇DI变量/DO变量的配置：
设备地址格式为：数据块名:b偏移量。（b代表以位为单位进行偏移）

例如：Datablock1:b61，表示读取数据块Datablock0中的第62位。

◇AI变量/AO变量配置：

设备地址格式为：数据块名:B（W）偏移量。（B代表以字节为单位进行偏移,W表示以字为单位进行偏移）

例如：Datablock0:B23，表示从数据块Datablock0中的第24字节开始读取，读取的长度根据AI变量的数据类型自动选取，8位整形读取1个字节，16位整形读取2个字节，32位整形和单精度浮点型读取4个字节，双精度浮点型读取8个字节。

Datablock0:W23，表示从数据块Datablock0中的第24字节开始读取，读取的长度根据AI变量的数据类型自动选取，由于W单位是字，一个字是2个字节，所以配置的数据类型不能是8位的整形，从16位整形开始读取1个字，32位整形和单精度浮点型读取2个字，双精度浮点型读取4个字。

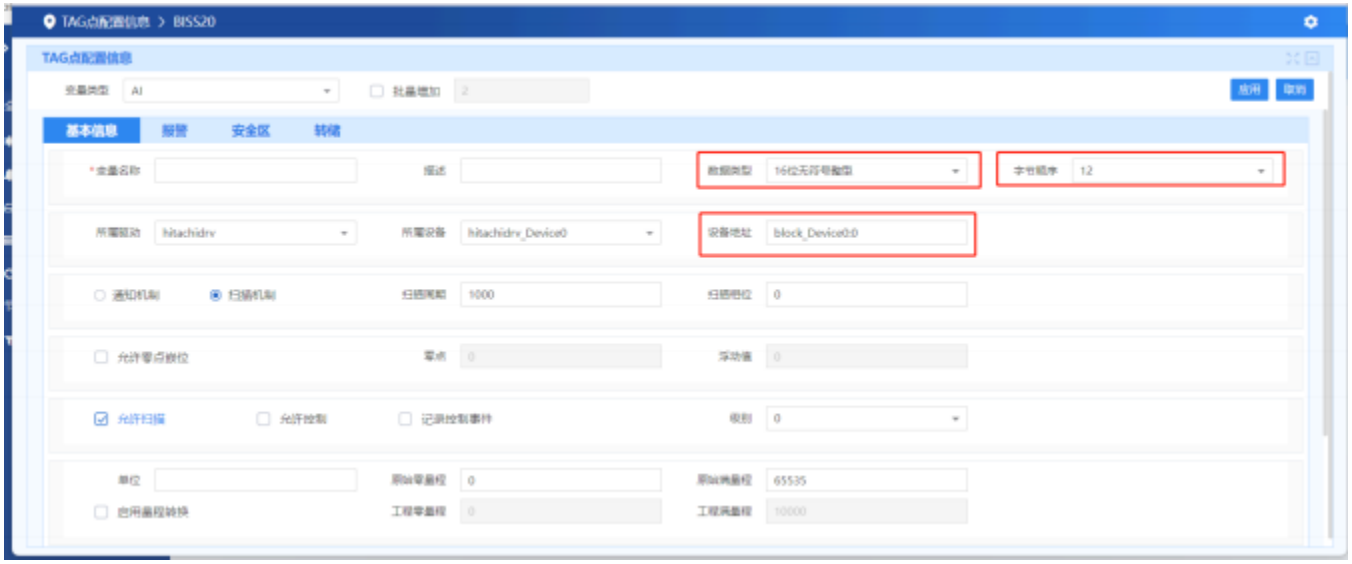
◇TEXT、BLOB：

设备地址格式为：数据块名:偏移字节#字节长度

例：DataBlock0:2#50

表示从数据块DataBlock0的第3字节开始读取50字节的长度。

注：变量配点的时，也支持W，以16位为一个字，也支持按字去采集，一个字是两个字节。两个字节一个字，在采集string或者blob的时候会用到。



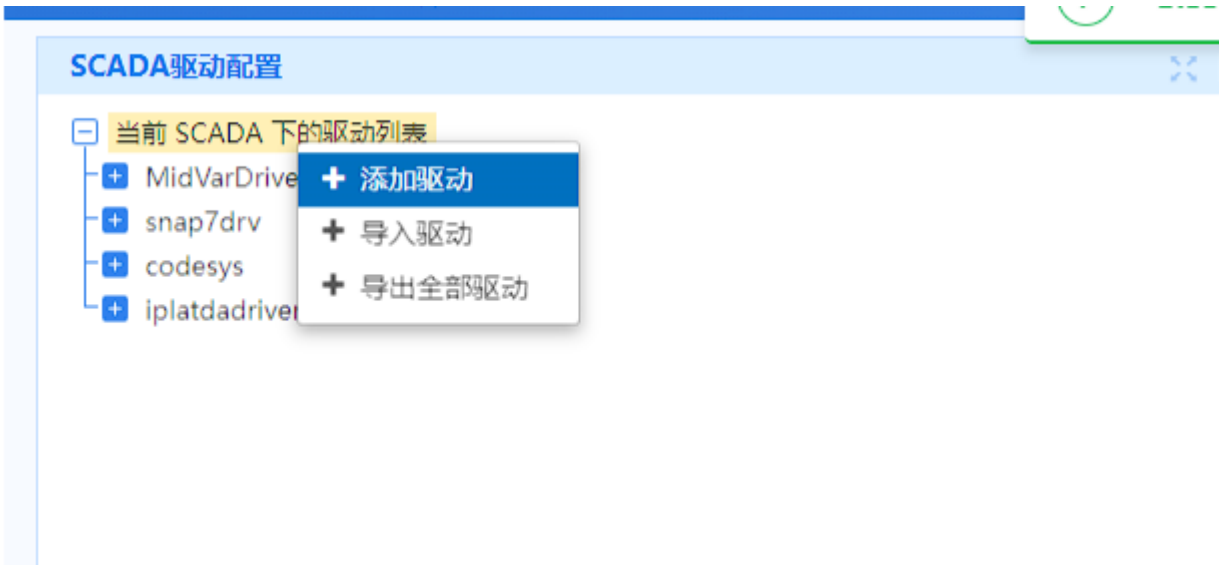
codesysdrv驱动支持宝信自主PLC vpoc、mpoc

vpoc对标西门子TDC

mpoc对标西门子s7-416+FM485,S7-1518

添加codesysdrv驱动

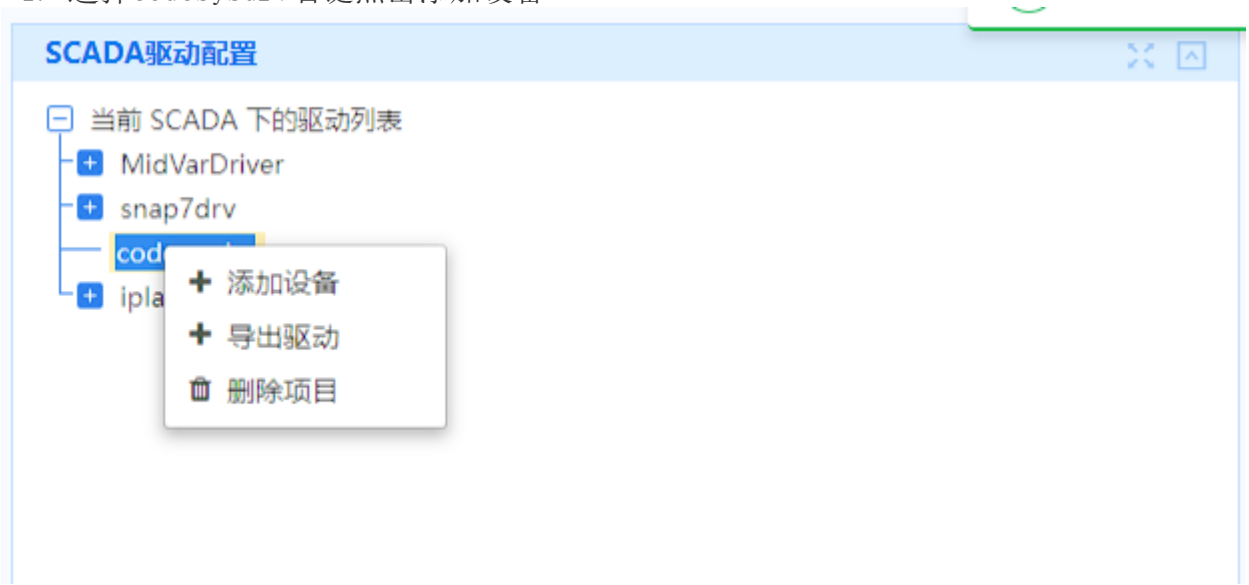
1. 首先，SCADA驱动配置中右键添加驱动。



2. 点击右侧"请选择驱动对话框"，选中codesysdrv，然后单击保存按钮。

配置codesysdrv驱动

1. 选择codesysdrv右键点击添加设备



2. 右侧弹出设备信息，填入相关配置项，如下图：

必填参数：

❖设备名称：自定义，不能为空，不能以数字开头

❖连接参数：PLC设备的IP地址，单机模式为ip=xxx.xxx.xxx.xxx;multilink=1;，其中连接参数中multilink可选0或1，0为单连接1为多连接。

❖轮询周期和响应超时：按照工程项目的需要填写

❖任务号：如果驱动配置了多个PLC设备，建议不同的设备分配不同的任务号

❖读取方式：分为循环读和同步读建议使用循环读，相比同步读，循环读性能更优。（旧版本没有读取方式配置项，是将扩展参数1配置为1）

2. 点击保存按钮，完成驱动配置。

配置codesysdrv变量

添加变量参见变量管理快速搭建手册，目前支持布尔型、整型、浮点型、字符串型变量的读写。

注意：配置完成后需要部署到scada节点，使得配置生效。

在配置变量时，变量地址的书写格式为所需连接PLC完整的变量名。

例如：Application.TEST.TEST

模拟量变量地址格式：PLC变量名，例如：Application.TEST.TEST

数字量变量地址格式：PLC变量名:位移，例如：Application.TEST.TEST:bX

（b后面跟的数字X代表偏移的位数）

字符串类型变量地址格式：PLC变量名#长度，例如：Application.TEST.TEST#20

在设备地址后面配置长度（PLC侧也需要长度一致），上例表示长度为20的字符串。地址最大长度128(指变量地址的总长度)

采集ab的plc推荐使用iPlat平台的另外一个驱动abdrv，能支持自定义结构体、结构体嵌套、基本类型数组、结构体数组，不用配数据块，配置更加方便。

CIPDriver驱动是罗克韦尔自动化公司的PLC设备的驱动,用户对象是设备管理和维护人员,需要进行读取和写入设备的操作。

设备组可以将设备进行分类，单个设备组中的设备是按照顺序以单线程的方式运行，如单个设备组中有部分设备运行故障，那么就会降低此设备组中正常设备的运行速度，所以用户在配置时，请确认各设备是否运行正常。而多个设备组之间是以多线程的并行的方式运行，设备组与设备组之间不会相互影响。用户可以根据现场情况进行分类，如运行快的设备放在一起、运行慢的设备放一起或者运行慢的和运行快的放在一起平衡速度。

注意：本驱动只支持AB-PLC中基本类型的数组（如INT,DINT,SINT,REAL,BOOL）和单个结构体中各属性的采集（如STRING类型中的string.LEN和string.DATA[0]），并对单个STRING类型的读取做过字符串处理。

但本驱动不支持结构体（包括自定义类型）数组(如string[2]中的string[1].LEN等)的采

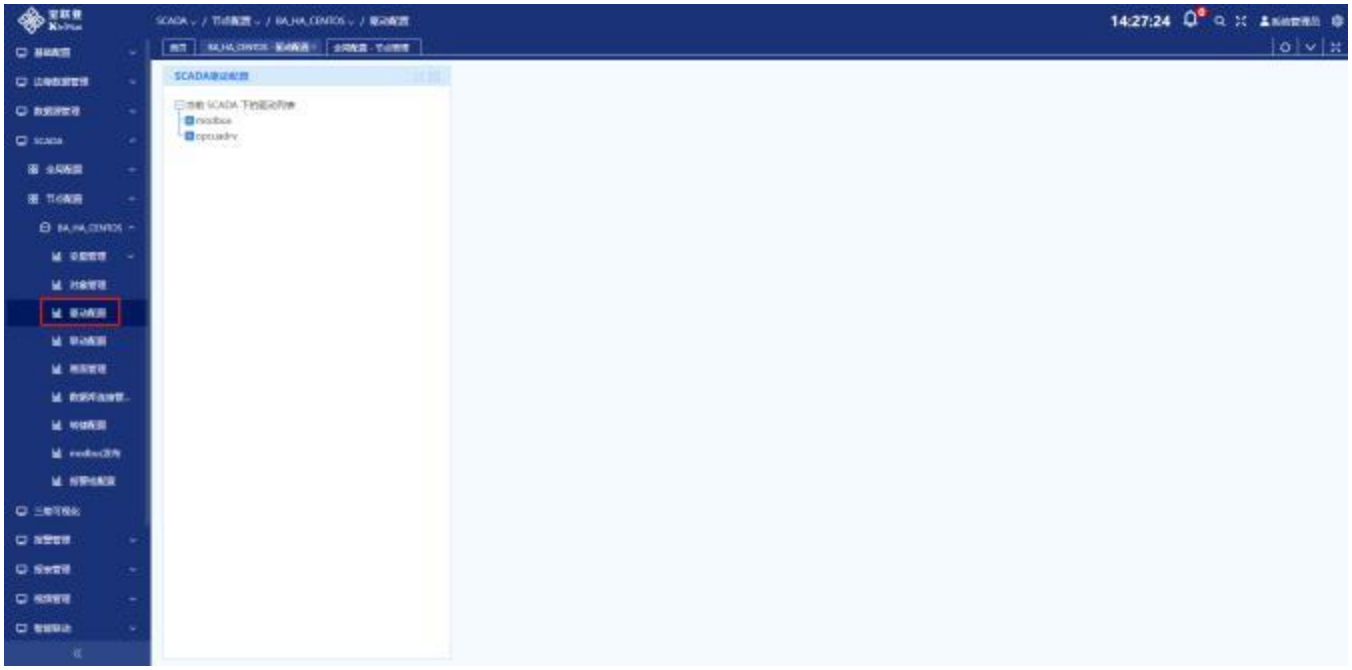
集。因此若需要对结构体数组和复杂的结构体进行采集的场景，可使用CIPDriverX驱动

。

主要包含以下步骤：

添加CIP驱动

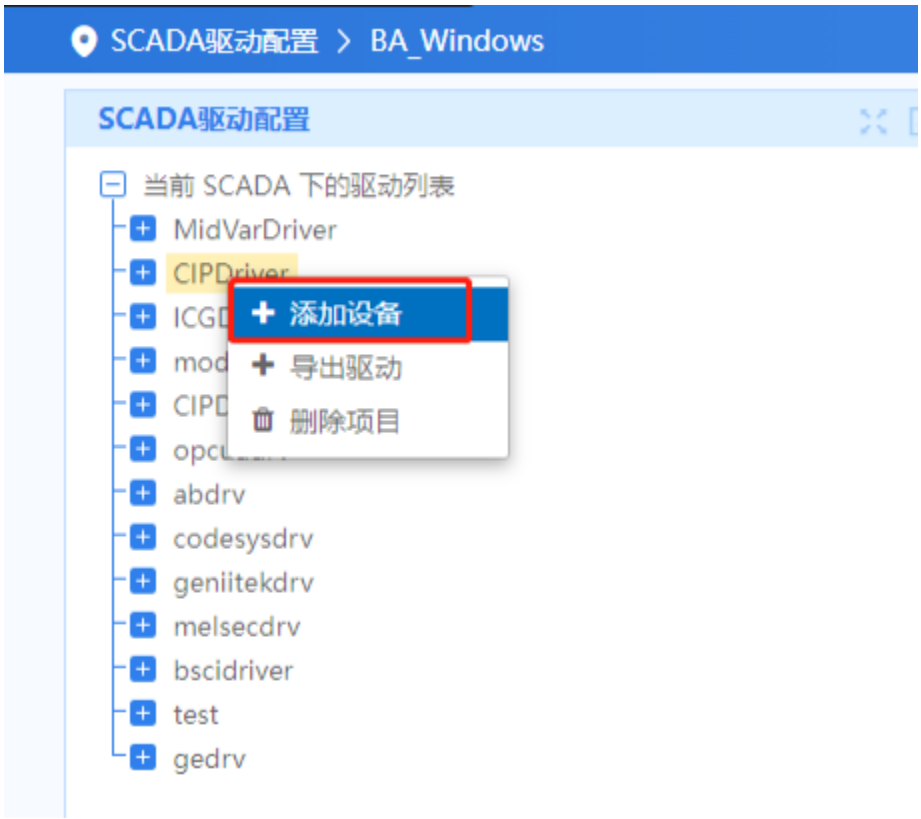
1.1、配置好相应的节点，点击节点下的驱动配置。



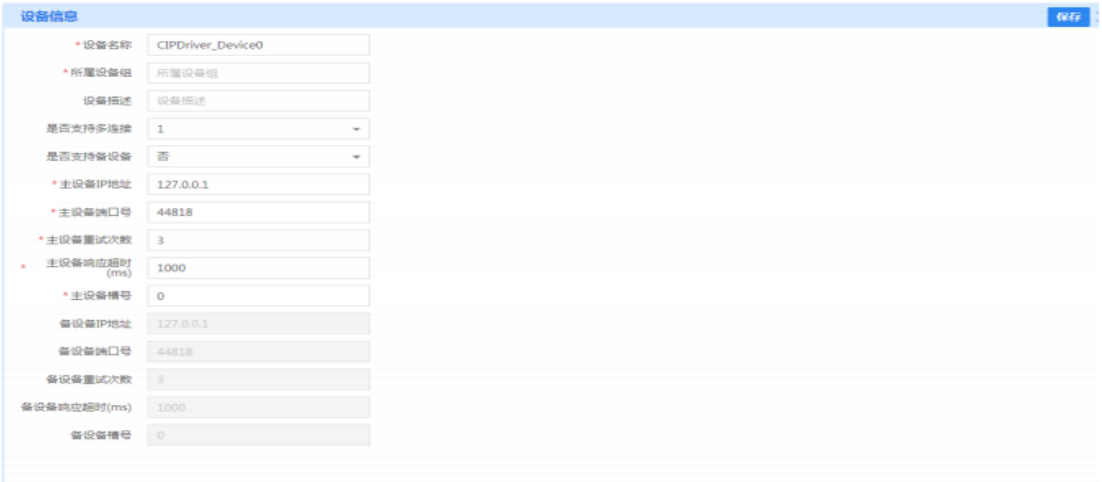
1.2、进入驱动配置页面，右击页面的左上方的"当前SCADA下的驱动列表"，点击添加驱动，，在弹出的"驱动信息"中勾选CIPDriver，点击"确定"以完成添加"CIPDriver"驱动。

添加CIP驱动设备

2.1、右击"CIPDriver"，点击"添加设备"。



2.2、点击"添加设备"后，在右侧的"设备信息"中添加设备信息。



2.3、"设备信息"表中各项参数说明。

❖*(必填项)设备名称：自定义，长度为16位字符串，仅能包含数字、英文字母和_；

❖*(必填项)所属设备组：自定义，长度为16位字符串，仅能包含数字、英文字

母和_；❖设备描述：自定义，长度为64位字符串或21个中文；

❖是否支持多连接：1为支持，0为不支持，默认为1；(支持多连接含义为设备允许多个客户端同时进行连接采集)

❖是否支持备设备："是"为支持，"否"为不支持；当选择"是"的时候，备设备信息才可以编辑；默认为"否"；

❖*(必填项)主设备IP：AB PLC主设备IP；

❖*(必填项)主设备端口号：AB PLC主设备的端口号，默认为：44818；

❖*(必填项)主设备重试次数：发送超时后重送的次数，默认为3；

❖*(必填项)主设备相应超时(ms)：驱动向设备发送需要反馈的命令所等待的超时时间；

❖*(必填项)主设备槽号：AB PLC主设备插入机架中的槽号；

❖注：备设备信息仅在"是否支持备设备"选择"是"的情况下才可编辑。

❖备设备IP：AB PLC备设备IP；

❖备设备端口号：AB PLC备设备的端口号，默认为：44818；

❖备设备重试次数：发送超时后重送的次数，默认为3；

❖备设备相应超时(ms)：驱动向设备发送需要反馈的命令所等待的超时时间；

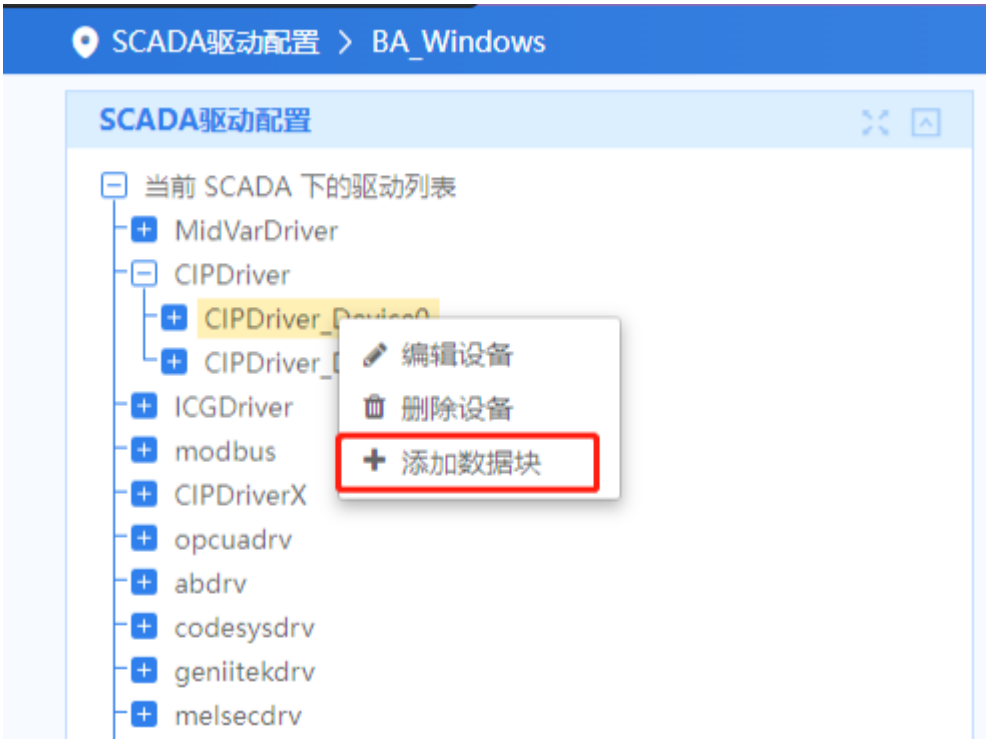
❖备设备槽号：AB PLC备设备插入机架中的槽号；

2.4、配置完成之后，点击右下角的"保存"，待系统提示"对1条记录执行新增操作成功！"，即表示设备新增加成功；

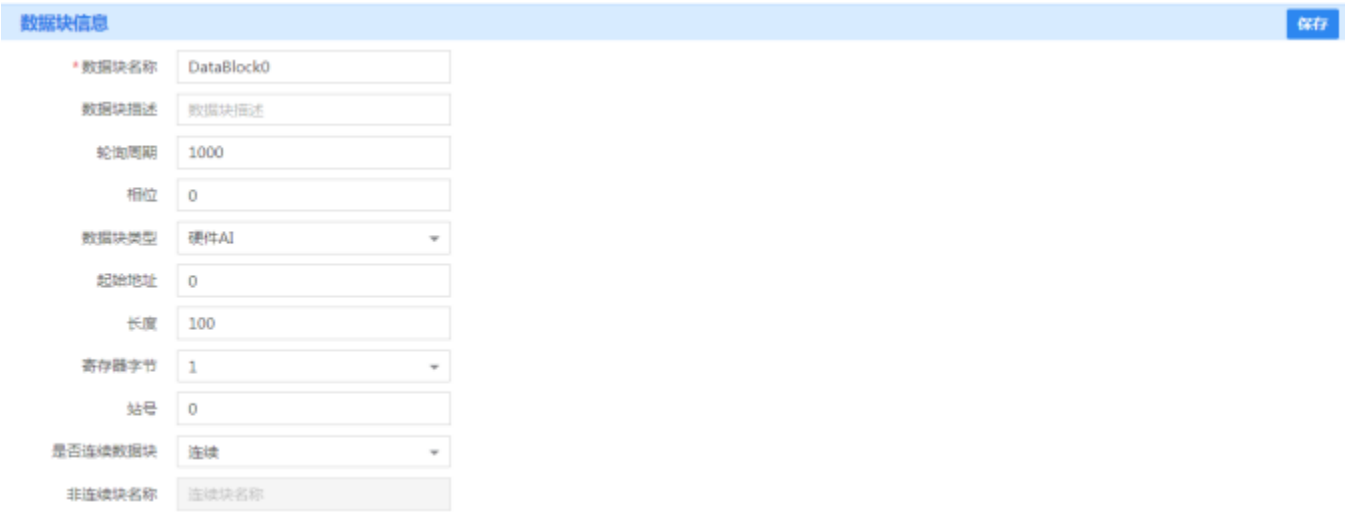
注：CIP协议支持免配块方式，如选择此方式，配到"设备"即可，可跳过配置"数据块"的过程，直接进入"变量配置"步骤，详见：添加CIP变量。

添加CIP数据块

3.1、右击"添加CIP设备"中已经添加好的CIP设备，点击"添加数据块"



3.2、点击"添加数据块"后，在右侧的"数据块信息"中添加数据块信息。



3.3、"数据块信息"表中各项参数说明。

❖*(必填项)数据块名称：如果数据块是连续块，则该名称必须是CIP上实际的数据块名；如果数据块是非连续块，则该名称为自定义，16位字符串，仅包含数字、字母和_；

❖数据块描述：自定义，128位字符串或者42位中文字符串；

❖轮询周期：扫描数据块的周期；

❖相位：轮询周期中的相位；

❖数据块类型：有"硬件AI"、"硬件AO"、"硬件DI"、"硬件DO"4种数据块类型，根据PLC中相关的数据块类型选择相应的数据块类型；

❖起始地址：数据块中元素起始位，默认0；

❖长度：元素的长度，如：数据块长度配置了4，那么包含DataBlock:0、DataBlock:1、DataBlock:2、DataBlock:3四个数据地址；

❖寄存器字节：元素的大小。

❖是否连续数据块：有"连续"和"非连续"两种，若PLC中的数据块是连续的，则选择"连续"，此时"非连续块名称"不可用；若PLC中数据块都是离散的不连续的，则选择"非连续"，如选择"非连续"，则"非连续块名称"必须添加；

❖非连续块名称：此项在"是否连续数据块"选择"非连续"时可用，格式为"数据块1;数据块2;数据块3"。

3.4、离散块的注意点

❖数据块为离散块指数据块地址是不连续的，如1001、2012。

❖为了避免丢点现象，如果用户知道设备的数据变化周期，建议将驱动的刷新周期配置为设备的数据变化周期的一半，点的扫描周期配置为数据变化周期的4分之一，如果使用到转储功能，建议将转储数据属性中的扫描周期配置为数据变化周期的8分之一，如设备的数据变化周期为4000毫秒，那么驱动刷新周期为2000毫秒，点的扫描周期为1000毫秒，转储数据的扫描周期为500毫秒。

3.5、配置完成之后，点击右下角的"保存"，待系统提示"对1条记录执行新增操作成功！"，即表示数据块新增加成功；

添加CIP变量

4.1、与驱动有关的变量配置项目主要有以下四项：

❖所属驱动：选择添加的变量来自何种驱动；

❖设备名称：选择添加的变量来自何设备，该设备名称应已配置完成；

❖设备地址：信号地址，地址格式在选择"配块"与"免配块"不同选择时，有不同的表达格式，具体请详见下面帮助内容；

❖数据类型：AI、AO类型数据有此选项，请根据PLC端传送的数据类型选择；

❖字节顺序：AI、AO类型数据有此选项，有些PLC关于模拟量数据有高低位翻转的通讯要求，可根据需要勾选并实现高低位换位；

添加变量详细说明参见新增变量。

注意：配置完成后需要部署到scada节点，使得配置生效。

CIP驱动支持免配块方式，用户可以自行选择是否配置数据块的方式，两种方式只可二选一。通常建议采用免配块方式，此种方式工作量较小。

4.2、添加CIP配块变量

❖块内只有一个变量的设备地址格式：块名称，例如：DataBlock0；

❖包含多个变量的数据块的设备地址格式：块名+":"+块内下标，即"块名:n"，例如：DataBlock:2。若还有位偏移量时，即"块名:n:bitoffset"，例如：DataBlock:2:3；

4.3、添加CIP驱动免配块变量

免块模式下，数据块的类型由变量类型决定，如下所示：

◇PLC内AI类型变量可配置为AI块

◇PLC内AO类型变量可配置为AO块

◇PLC内DI类型变量可配置为DI块

◇PLC内DO类型变量可配置为DO块

◇PLC内TXT类型变量可配置为AI块

◇PLC内BLOB类型变量可配置为AI块

◇PLC内其余Tag点类型暂不涉及

CIP驱动块内元素长度最大到64位（如64位的双精度浮点型）。

CIP驱动免配块方式不支持字符串的数组，该类型需要换配块方式。

注：由于一个设备只支持配块或免配块方式2选1，如果选择了配块方式，其他点也需要按配块方式操作。如果CIP块中存在字符串的数组，建议考虑单独为字符串的数组通讯另建一个设备连接。

由于CIP数据块名称必须是PLC上实际存在的，自组块时同一个TAG点的设备地址不可以同时配置到AO和AI变量，或者同时配置到DI和DO变量中，只能建其一。

当前AB PLC大多选用符号地址方式编程，因而CIP协议免配块方式的设备地址格式为：

◇模拟量符号地址变量的设备地址格式：组名.变量名，示例：GroupName.Tagname

◇数字量符号地址变量的设备地址格式：组名.变量名.位号，示例：GroupName.Tagname.1

（组名若PLC侧没定义，可省略。）

CIP通讯返回错误码

将驱动程序日志文件等级修改为"DEBUG"，可以查看通讯过程及详细错误信息。日志操作方法详见：日志配置

CIP驱动Invalid Status错误码：

CIP Service Errors

In general, the CIP status is returned as follows:

- good status
- general error with no extended status:

code	00

Where:	Means:
04	This general status code means that the IOI could not be deciphered. Either it was not formed correctly or the match tag does not exist
05	The particular item referenced (usually instance) could not be found
06	The amount of data requested would not fit into the response buffer. Partial data transfer has occurred.
0A	An error has occurred trying to process one of the attributes
13	Not enough command data / parameters were supplied in the command to execute the service requested
1C	An insufficient number of attributes were provided compared to the attribute count
26	The IOI word length did not match the amount of IOI which was processed
00	SUCCESS!!

- error with extended status:

code	01	extended	code

Where:			Means:
FF 01	05	21	You have tried to access beyond the end of the data object
FF 01	07	21	The abbreviated type does not match the data type of the data object.
FF 01	04	21	The beginning offset was beyond the end of the template

```
<?xml version="1.0"encoding="GB2312"?>
<DriverCfg name="CIPDriver"devGrpCount="1">
    <DevGrpCfg name="cipx_test"devCount="1">
        <Description/>
        <DeviceCfg name="device0"multiLink="1"dataBlockCount="1">
            <Description/>
            <PrimaryDevice
            ip="127.0.0.1"port="44818"retries="3"replyTimeout="1000"slot="0
            "/>
                                                        <DataBlockCfg
            name="DataBlock0"blockType="0"stationNo="0"pollrate="1000"phase=
                    <Description/>
```

```
                <DevAddress wordBytes="1"startaddr="0"length="100">
                    <ItemCfg name=""/>
                </DevAddress>
            </DataBlockCfg>
        </DeviceCfg>
    </DevGrpCfg>
</DriverCfg>
```

采集ab的plc推荐使用iPlat平台的另外一个驱动abdrv，能支持自定义结构体、结构体嵌套、基本类型数组、结构体数组，不用配数据块，配置更加方便。

设备组可以将设备进行分类，单个设备组中的设备是按照顺序以单线程的方式运行，如单个设备

组中有部分设备运行故障，那么就会降低此设备组中正常设备的运行速度，所以用户在配置时，请确认各设备是否运行正常。而多个设备组之间是以多线程的并行的方式运行，设备组与设备组之间不会相互影响。用户可以根据现场情况进行分类，如运行快的设备放在一起、运行慢的设备放一起或者运行慢的和运行快的放在一起平衡速度。

注意：本驱动支持AB-PLC中各种结构及其数组的读取，但由于解析的特性，在读取基本类型的数组（如INT,DINT,SINT,REAL,BOOL）时，采集性能低于CIPDriver，且对单个STRING类型的读取没有进行特殊处理（参见CIP驱动-字符串）。

因此，若需要进行基本类型数组和单个string类型读取操作的，可以使用CIPDriver。

本驱动做了以下测试：

1. 本驱动的采集受驱动所在设备CPU性能影响明显。
2. 在intel-i5处理器，局域网环境下，配置10个设备组（即分10个连接进行采集），每组单设备并配置1024点（即共10240点）可以达到每秒刷新

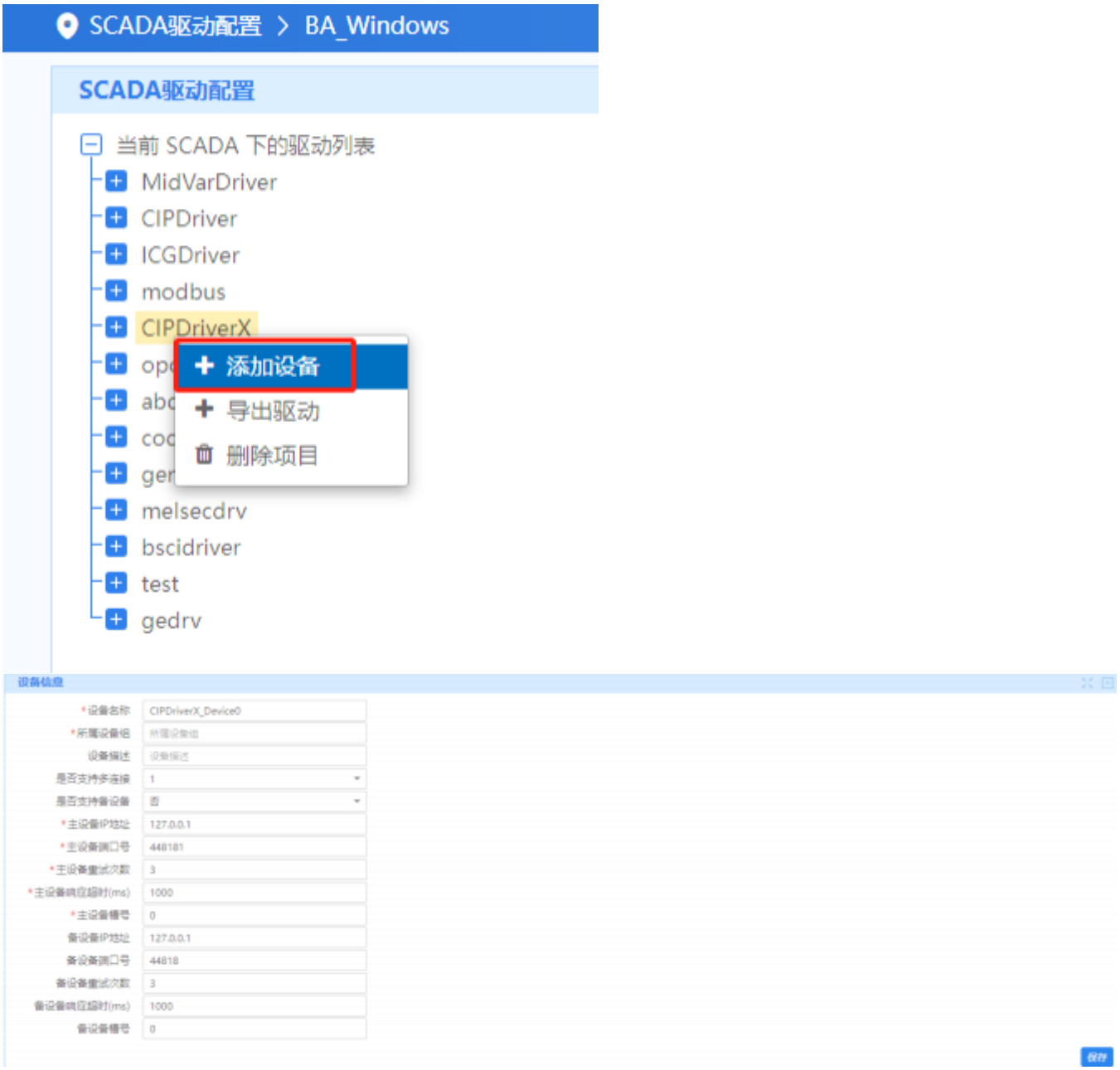
主要包含以下步骤：

添加CIPX驱动

驱动配置下右键添加驱动，在右侧驱动列表页面，选择CIPDriverX，点击"确定"，如下图：

CIPx设备配置

在左侧驱动列表中右键点击驱动，点击添加设备，右边弹出设备信息，设备的名称自动生成，其他属性为默认值。



如果现场设备存在冗余，需要选中"备设备"复选框，然后填入备机的ip地址和端口号；除此之
外，还需要配置各项参数，驱动配置工具中所列的默认配置参数为推荐的配置值，并非所有的设备均合适，如果通讯出现问题，需要调整配置参数，请咨询设备厂商。

参数配置如下：

❖设备名称：自定义设备名称。

❖所属设备组：当一个工程中，需要连接不同工艺组的众多PLC设备时，可以用"设备组"来进行分组管理。如果所连PLC设备不多，也可归为一个设备组。

❖设备描述：添加对设备的描述信息。

❖设备地址：设备的IP地址。

❖端口号：设备连接的端口号。

❖槽号：设备插入机架中的槽号。

❖响应超时：驱动向设备发送需要反馈的命令所等待的超时时间。

❖重试次数：发送超时后重送的次数。

CIPx变量配置

添加变量方法参见：变量管理快速搭建手册。变量设备地址填写格式及配置方式与CIP驱动相同，请参见：CIPDriver变量配置。

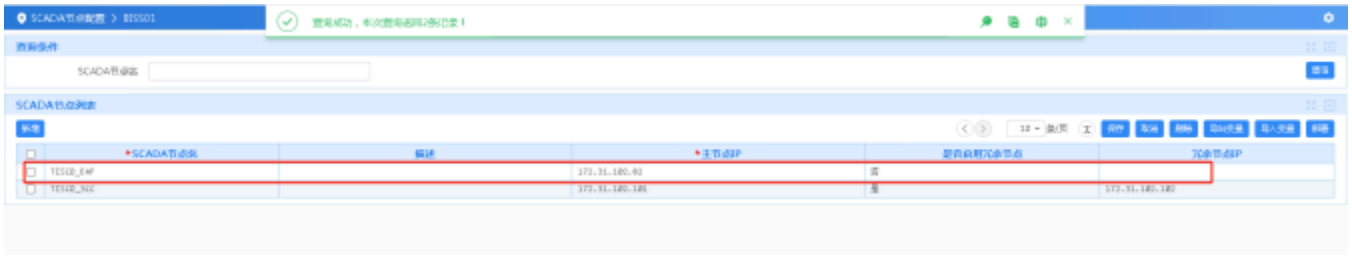
注意：配置完成后需要部署到边缘节点，使得配置生效。

CIPDriverx.xml

```
<?xml version="1.0"encoding="GB2312"?>
<DriverCfg name="cipdriverx"devGrpCount="1">
    <DevGrpCfg name="cipx"devCount="1">
        <Description></Description>
        <DeviceCfg name="cipx_d"multiLink="1"dataBlockCount="0">
            <Description></Description>
                                                    <PrimaryDevice
         ip="192.168.128.97"port="44818"retries="3"replyTimeout="1000"slot="
        </DeviceCfg>
    </DevGrpCfg>
</DriverCfg>
```

添加cvrda驱动

1. 首先，需要配置好**至少**两个宝信基础自动化软件的SCADA节点信息，如图节点test节点和TISCO_EAF节点，TISCO_EAF节点中含有正常的tag数据信息。test节点用来取
TISCO_EAF节点中的存在tag点数据。



2. 点击test节点下SCADA节点下的驱动配置，进入驱动配置页面。

右键点击**【当前SCADA下的驱动列表】**添加驱动弹出可用驱动的选择对话框，在右侧显示的**【驱动信息】**页面弹出的**【可选驱动列表】**中，**勾选cvrda驱动所对应的"cvrdadriver"，然后关闭对话框。驱动添加成功后，点击**【确定】**按钮。

配置cvrda驱动

1. 选择cvrdadriver后,点击鼠标右键，选择"添加设备"：

2. 系统弹出设备配置窗口，填入相关配置项，如下图：

❖必填参数：

◇设备名称为自定义：不能为空，不能以数字开头

◇链接类型为other

◇链接参数为：scada=scadamain;ip=192.168.128.121,192.168.128.122(IP之间以半角逗号分隔)

scada填入所需连接设备所在的scada名称(目前scada名称长度不能超过32)，ip填入所需连接设备所在的scada对应的ip。

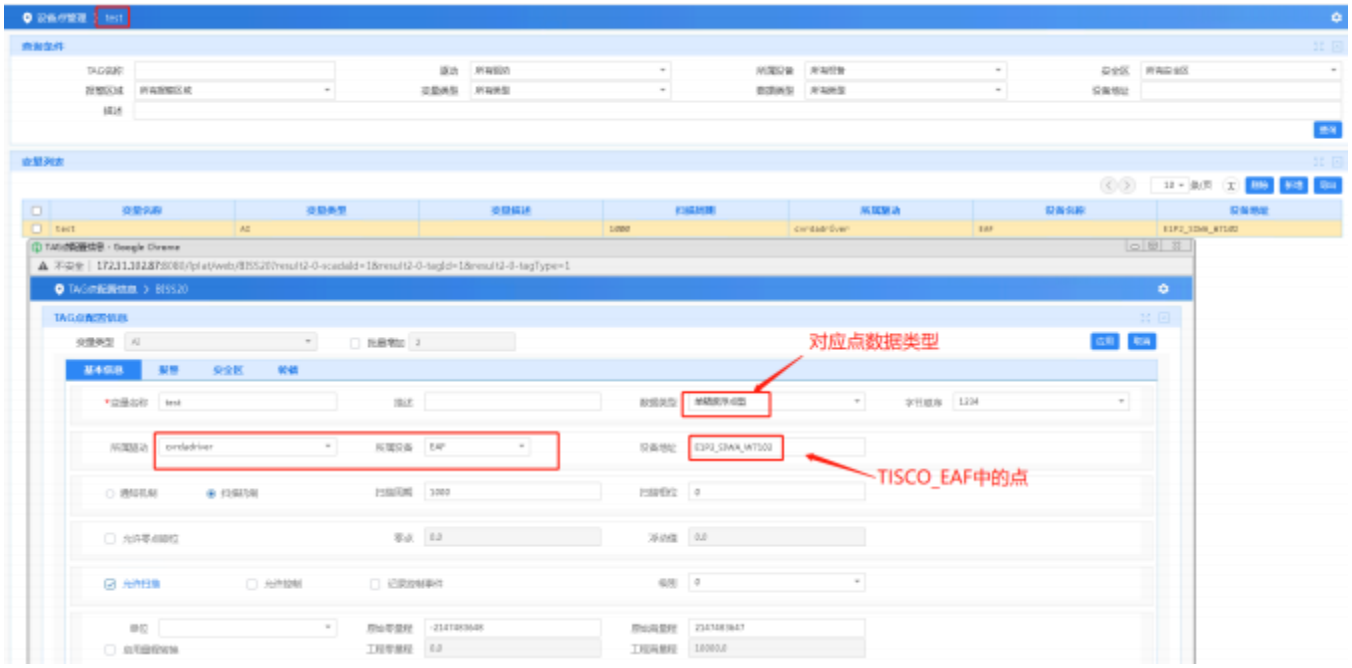
注意：cvrda驱动中配置的各设备的scada名称不可重复；在采集跨工程的scada数据时，cvrda驱动中配置的设备的scada名称可与本工程中的scada名称相同，实际以配置的ip地址为准，此时采集的是跨工程的scada的数据。

◇任务号为1-100之间的数，每次新增会生成一个不同的任务号，若不同设备设置相同任务号，则这些设备将会顺次进行采集而不是并行

3. 点击**【保存】**按钮，关闭配置页面。

配置cvrda变量

在配置变量时，变量地址为所需连接的scada上的tag名（目前地址填写的tag名长度不能超过64）。



变量支持配置长度，格式形如tag1#250；其中txt变量如果不配置长度，默认长度为254。注意：

变量类型	读长度（字节）	写长度（字节）
txt	254	254
blob	2048	2048

注意：关联到驱动的话还需要看每个驱动的协议，取驱动中读写长度和cvrdadriver长度的最小值。

配置完成后需要**部署**到scada节点，使得配置生效。

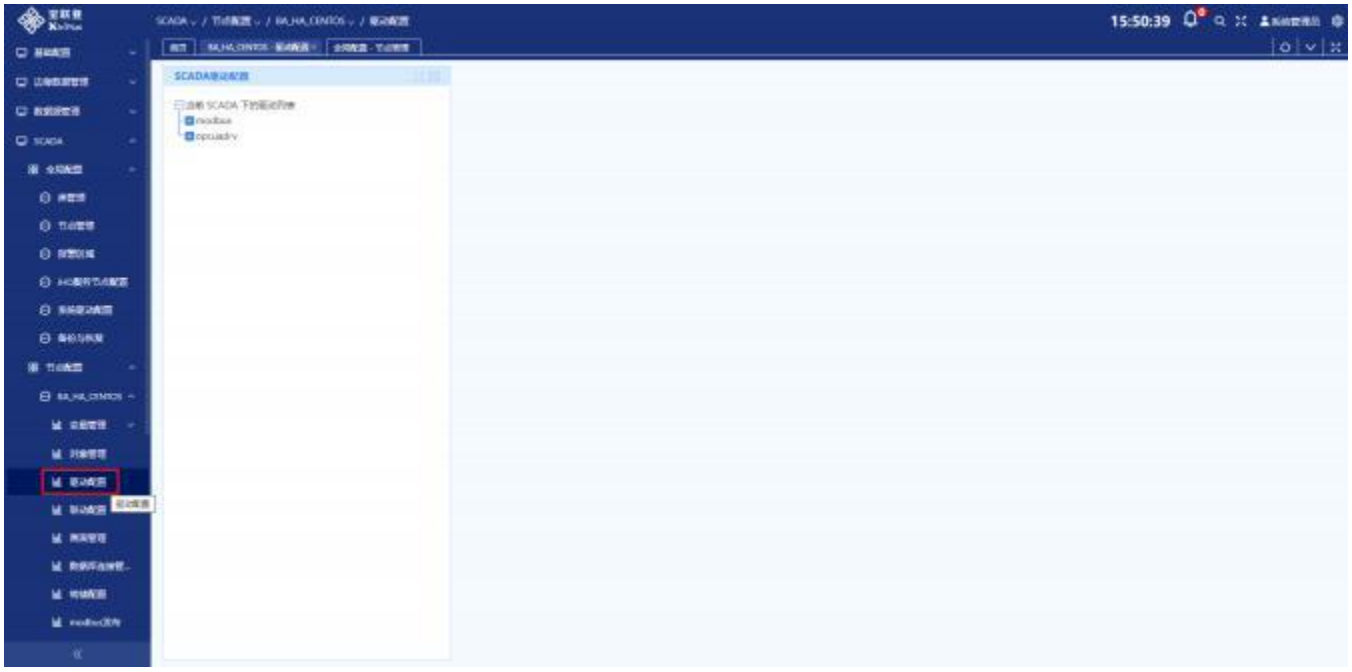
cvrdadriver.xml

```
<?xml version="1.0"encoding="GB2312"?>
<driver name="cvrdadriver"type="block"desc="">
                                                    <device
name="cvrda"conntype="Other"connparam="scada=test_5_6_2_CN;ip=168.2.8.110"cy
                                                    clerate
```

</driver>

添加bscidriver驱动

1. 在SCADA管理下的对应节点，点击驱动配置



2. 在驱动配置页面中，点击添加驱动，勾选bscidriver驱动，再点击确定

添加bscidriver设备

1. 右键点击添加设备

2. 右键bscidriver，选择添加设备，如下图

在右侧设备信息中填写设备相关参数：

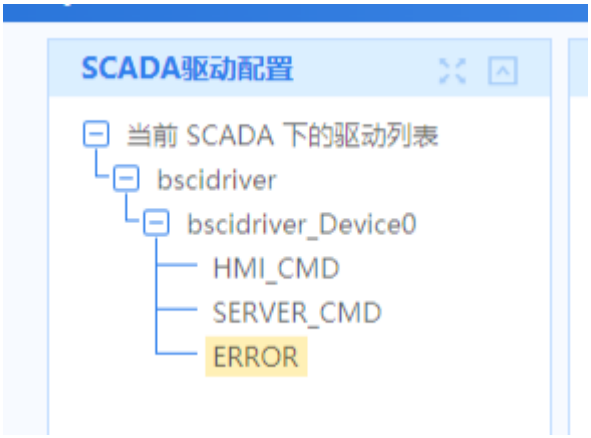
❖以TCP方式进行采集，按照以下方式配置

参数说明：

- 设备名称：不能为空，不能以数字开头；
- 描述：自定义的设备描述信息；
- 轮询周期：读取对应值得周期时间；
- 响应超时：连接设备和发送数据的响应超时时间；
- IP地址（主）：欲连接主设备的IP；
- 端口号（主）：欲连接主设备使用的端口号，添加设备时默认填入10000；
- IP地址（备）：欲连接备设备的IP；
- 端口号（备）：欲连接备设备使用的端口号，添加设备时默认填入10000；
- 任务号：同一任务号下的设备消息放在同一个进程中；
- 支持多连接：该设备是否支持多连接，1表示支持多连接；
- 扩展参数：用户自定义；

配置数据块

添加设备后会自动生成3个数据块，HMI_CMD、SERVER_CMD、ERROR



数据项信息

*地址	HMI_CMD
*元素个数	256
元素大小(字节)	1
描述	描述
可选参数1	可选参数1
可选参数2	可选参数2
可选参数3	可选参数3

保存

数据项信息

*地址	SERVER_CMD
*元素个数	256
元素大小(字节)	1
描述	描述
可选参数1	可选参数1
可选参数2	可选参数2
可选参数3	可选参数3

保存

也可以选择自行添加、删除、修改数据块。

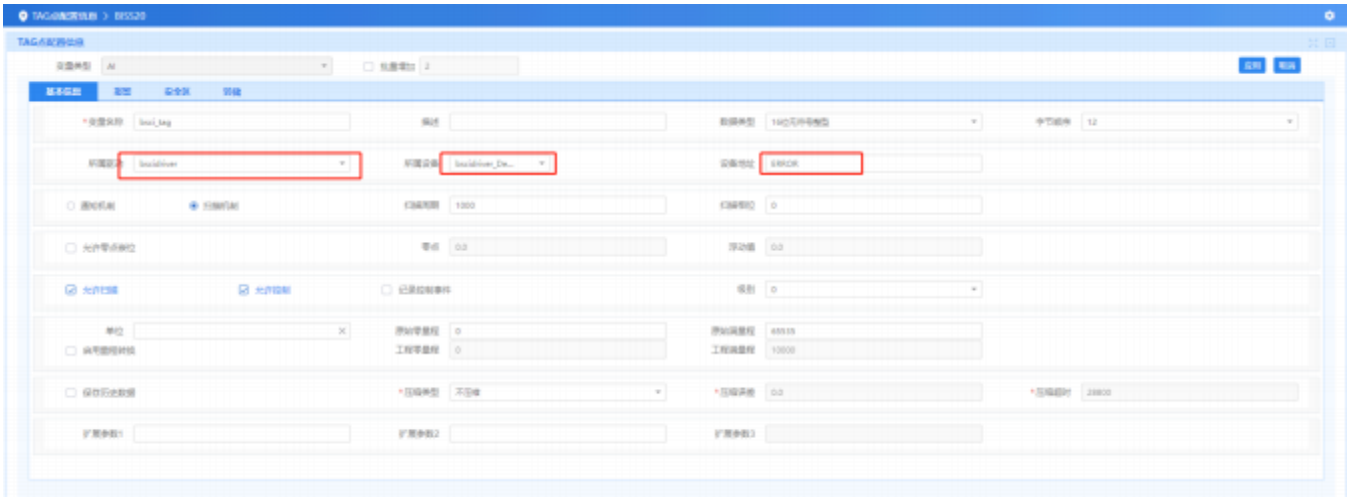
添加数据块：选中已经配置的设备，右键设备，选择"添加数据块"，弹出数据块信息配置界面，
如下图：

❖地址：设备地址。

❖元素个数：该数据块的长度,即，用户要在数据块中保存的查询记录的最大条数。本驱动默认
填入256，可以根据需求修改。

❖元素大小[字节]：用户查询的每条记录的元素大小。本驱动默认填入1，可以根据需求修改。❖描述：自定义的描述信息。
❖扩展参数：用户自定义。

添配变量



添加变量步骤与一般设备相同，这里主要描述变量关键信息:

所属驱动:选择bscidriver;

所属设备:选择bscidriver对应设备;

数据类型:理论上可以选择任意数据类型，驱动会将实际接收数据转成配置的类型，但还是建议根据报文里的数据类型来配置;

设备地址:

- DI变量/DO变量的配置:

  设备地址格式为：数据块名:b偏移量。（b代表以位为单位进行偏移）

  例如：Datablock1:b61，表示读取数据块Datablock0中的第62位。

- AI变量/AO变量配置:

  设备地址格式为：数据块名:B（W）偏移量。（B代表以字节为单位进行偏移,W表示以字为单位进行

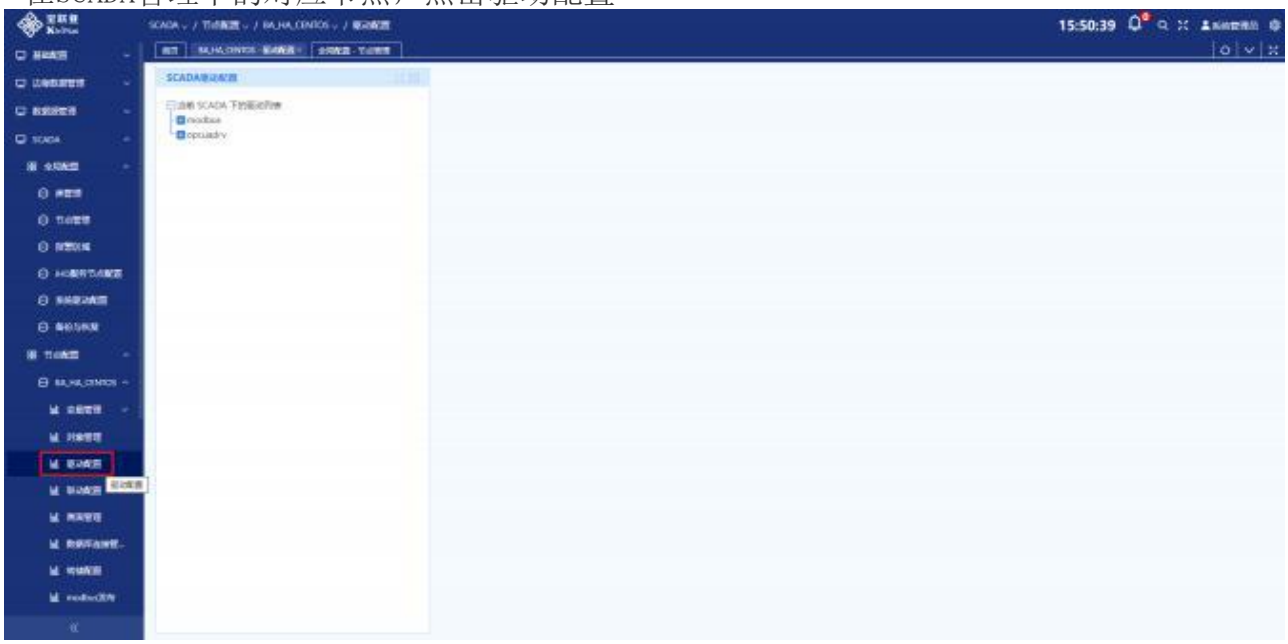
  例如：Datablock0:B23，表示从数据块Datablock0中的第24字节开始读取，读取的长度根据AI变量的数据类型自动选取，8位整形读取1个字节，16位整形读取2个字节，32位整形和单精度浮点型读取4个字节，双精度浮点型读取8个字节。
  Datablock0:W23，表示从数据块Datablock0中的第24字开始读取，读取的长度根据AI变量的数据类型自动选取，由于W单位是字，一个字是2个字节，所以配置的数据类型不能是8位的整形，从16位整形开始读取1个字，32位整形和单精度浮点型读取2个字，双精度浮点型读取4个字。

- TEXT、BLOB:
  设备地址格式为：数据块名:偏移字节#字节长度
  例：DataBlock0:2#50
  表示从数据块DataBlock0的第3字节开始读取50字节的长度。

注：变量配点的时，也支持W，以16位为一个字，也支持按字去采集，一个字是两个字
节。两个字节一个字，在采集string或者blob的时候会用到。
redisdrv采集的是redis中key-value中的value，key是点的设备地址，value是点的值，value在redis中的数据类型是string类型，在BA中配置对应数据类型点;只支持默认库。

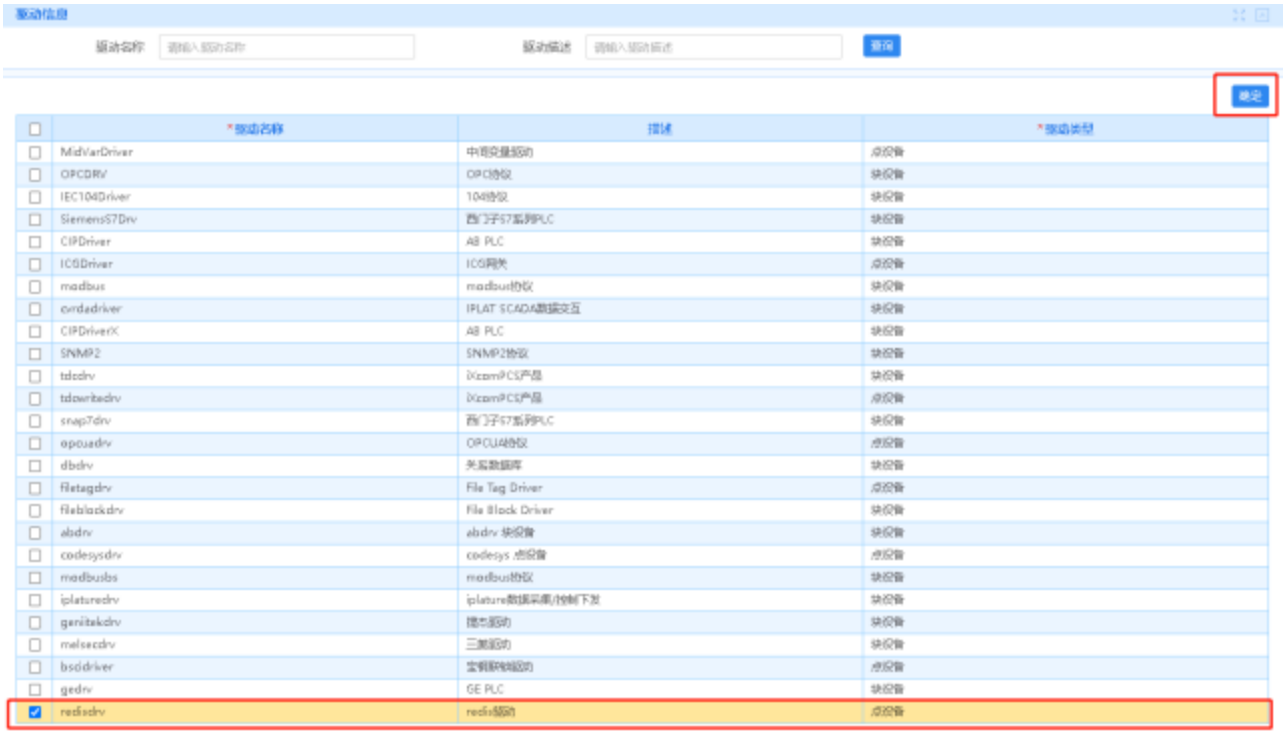
添加redisdrv驱动

1. 在SCADA管理下的对应节点，点击驱动配置



2. 在驱动配置页面中，点击添加驱动

3. 勾选redisdrv驱动，再点击确定



添加redisdrv设备

10.1 右键点击添加设备

10.2 填写所需设备信息后，点击右下角保存

设备信息参数说明：

- 设备名称：不能为空，不能以数字开头；
- 轮询周期：设备定时读取数据块周期时间(单位:毫秒)；
- 描述：用户添加对设备的描述信息；
- 响应超时：redis操作超时时间(单位:毫秒)；
- IP地址：redis所在机器的IP地址；

•端口：目标机器redis服务监听的端口号；
•连接密码：目标机器redis服务的授权密码(**目标机器redis没有设置密码时，这里就不填**)；
•心跳检测间隔：redis驱动建立连接后的心跳检测间隔时间(单位:秒)；
•任务号：同一任务号下的设备消息放在同一个线程中；
•扩展参数：用户自定义；

添加变量

添加变量步骤与一般设备相同，这里主要描述变量关键信息:

所属驱动:选择redisdrv;

所属设备:选择redisdrv对应设备;

数据类型:根据实际情况选择配置;

设备地址:存储到redis数据库的键值名字(**在键值名字后面加#和数字，可以设置redis读取的字
节长度，TEXT类型长度最大不能超过256，BLOB类型长度最大不能超过2048；
如果不加#，默认字节长度是64**)。

字符串数值读取注意事项:

•假如TEXT配了100字节长度，在redis里存储的数值长度>=100，redisdrv驱动读取之后会把第100个字符设置为'\0'结束符，所以实际只会展示99个字符数据。
•BLOB同理。

部署xml

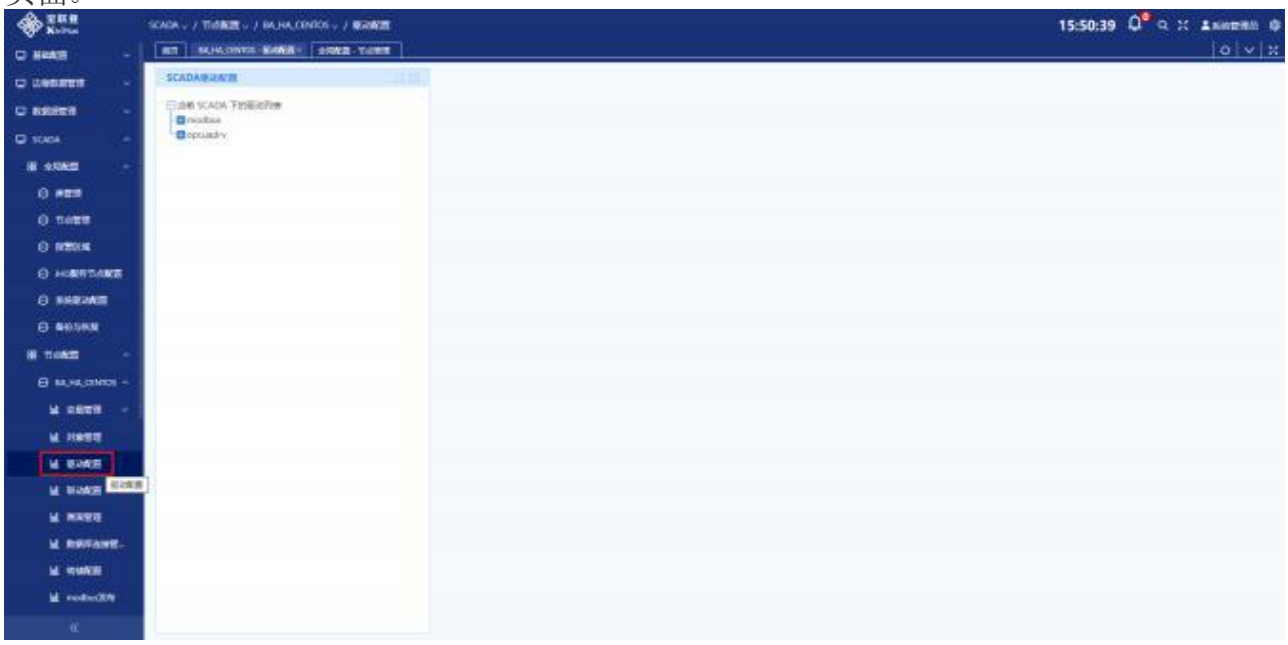
```
<driver name="redisdrv" type="item" desc="">
```

```
    <device
    name="item_Device1"conntype="Other"connparam="ip=127.0.0.1;port=6379;pas
    sword=admin;"
                                                                    <device
name="item_Device2"conntype="Other"connparam="ip=127.0.0.1;port=6379;passwor
                                                                    d=admin;in
                                                                    <device
name="item_Device3"conntype="Other"connparam="ip=10.25.3.129;port=6379;passw
                                                                    ord=admin;
  </driver>
```

tdcdrv驱动是基于iXcom读协议，负责从iXcom中读数据到BA。

添加xcomdrv

1. 首先，需要配置好相应的SCADA节点，点击SCADA节点下的驱动配置，进入驱动配置页面。



2. 右键点击【**当前SCADA下的驱动列表**】添加驱动弹出可用驱动的选择对话框，右键，【添加驱动】。

在**【当前SCADA下的驱动列表】**中，**右键点击**【tdcdrv】，**选择**【添加设备】**选项，系统弹出设备配置窗口，填入相关配置项，如下图：

- 参数说明
- 设备名称：自定义设备名，例如：AGC
- 连接类型：选择TCPServer
- IP地址：设备所在的IP地址
- 端口号：TCPServer监听的端口号
- 轮询周期、响应超时：使用默认参数或根据需求修改。
- 可选参数1：发送数据包的缓冲区长度（包含2字节TelLen+2字节TelID+最长数据块长度。即该设备下最长数据块长度+4）。

配置数据块

13.1 选中已经配置的设备，在需要配置数据块的设备上点击右键，选择**【添加数据块】**选项，弹出数据块配置界面，如下图：

参数说明

名称：自定义数据块名，例如：TCPServ

类型：填1,协议固定值

地址：数据块的起始地址

元素个数**数据块的元素个数（**元素个数元素大小**等于tdcdrv读取到的数据块的实际长度。数据块的实际长度=整体报文长度-2字节TelLen-2字节TelID）

元素大小[字节]：数据块中一个变量所占的字节长度，通常为"1"
轮询周期、响应超时：使用默认参数或根据需求修改。
相位：用于设置定时器的初始相位或偏移量,不用填

任务号：同一个任务号的数据块之间是按照顺序以单线程的方式运行；而不同任务号的数据块之间是以多线程的并行的方式运行，不会相互影响。

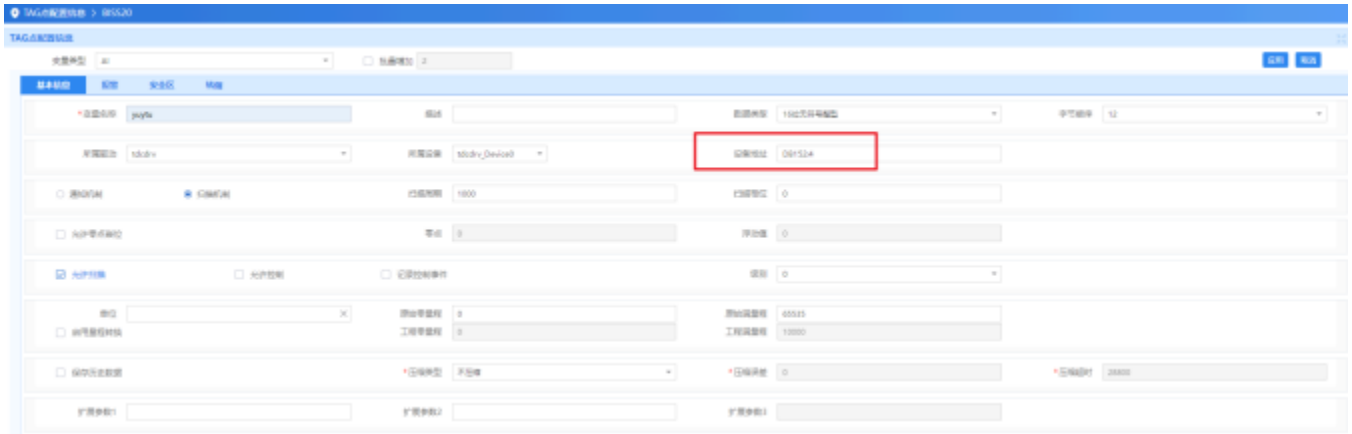
可选参数1：为电文号TelID

数据偏移说明（tag点配置注意事项）：

数据块TCPServ总长20byte，tag点配置时读取数据的有效范围TCPServ:4~TCPServ:19，

例如，下图红框中的偏移【DB152:4】，表示从有效范围
TCPServ:4~TCPServ:19的起始位置向后偏移4byte，开始读取数据。

TCPServ:0~TCPServ:为电文头(2字节TelLen-2字节TelID)



```
<driver name="tdcdrv"type="block"desc="">
                                                    <device
name="tdcdrv_Device0"conntype="TCPServer"connparam="ip=192.168.10.3;port=191
                                                    15;multiLi
</driver>
```

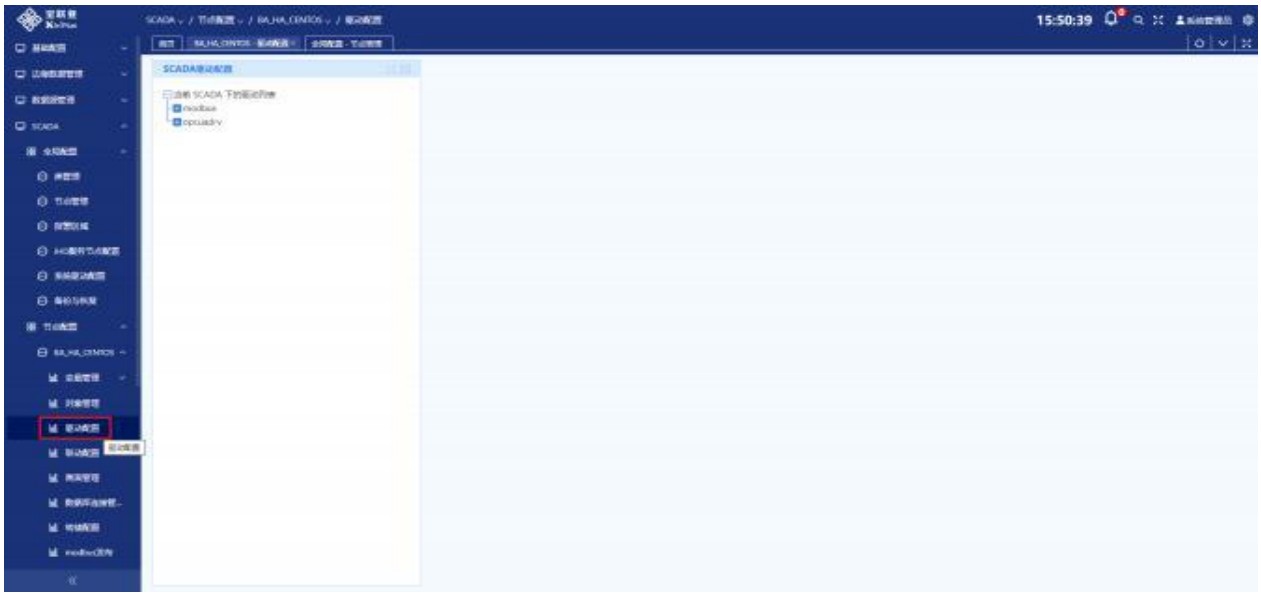
Tdcwritedrv

tdcwritedrv驱动是基于iXcom协议，负责从节点写入数据到PLC。

注：有些PLC只能连接一个Client，调试tdcwritedrv驱动时如果连接不上，需要确认是否有其他设备连接了PLC。

添加tdcwritedrv

14.1 首先，需要配置好相应的SCADA节点，点击SCADA节点下的驱动配置，进入驱动配置页面。



14.2 右键点击【**当前SCADA下的驱动列表**】添加驱动弹出可用驱动的选择对话框，在弹出的【驱动信息】，**勾选tdcwritedrv驱动所对应的"tdcwritedrv"，点击右上角**【确定】按钮。

tdcwritedrv驱动添加成功后，在**【当前SCADA下的驱动列表】**中，**右键点击**【tdcwritedrv】**，选择添加设备。

弹出tdcwritedrv设备配置界面。

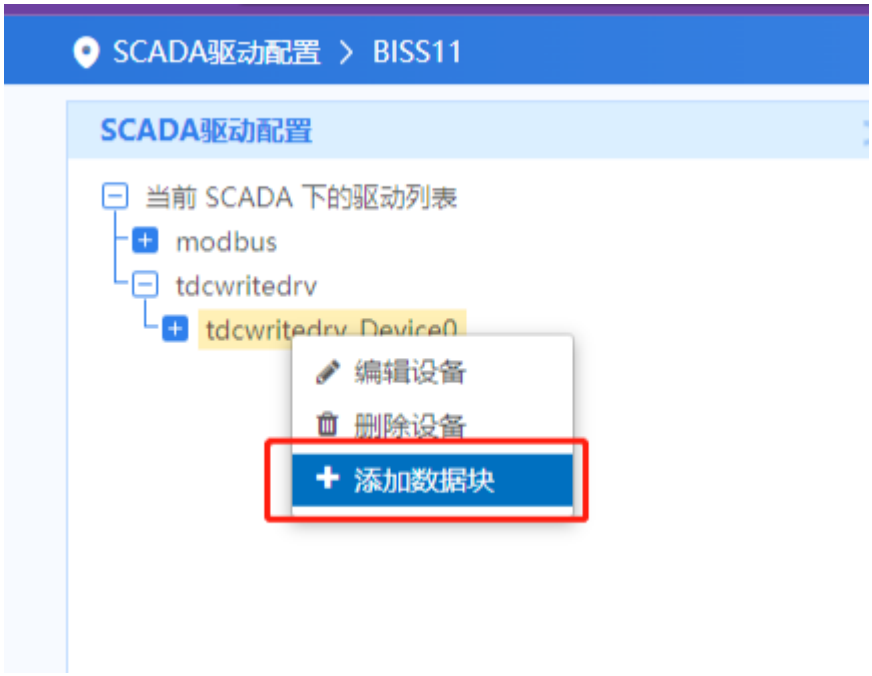
配置tdcwritedrv设备

在设备列表处点击鼠标右键，选择**【添加设备】**选项，系统弹出设备配置窗口，填入相关配置项，如下图：

- 参数说明
- 设备名称：自定义设备名，例如：tdcwritedrv_Device0
- 连接类型：选择TCPClient
- 连接参数：
- 响应（ms）：使用默认参数或根据需求修改。
- 描述：根据项目需要进行描述
- 任务号：
- 可选参数1：发送数据包的缓冲区长度（包含2字节TelLen+2字节TelID+最长数据块长度。即该设备下最长数据块长度+4）。

配置数据块

1. 选中已经配置的设备，在需要配置数据块的设备上点击右键，选择【添加数据块】选项，



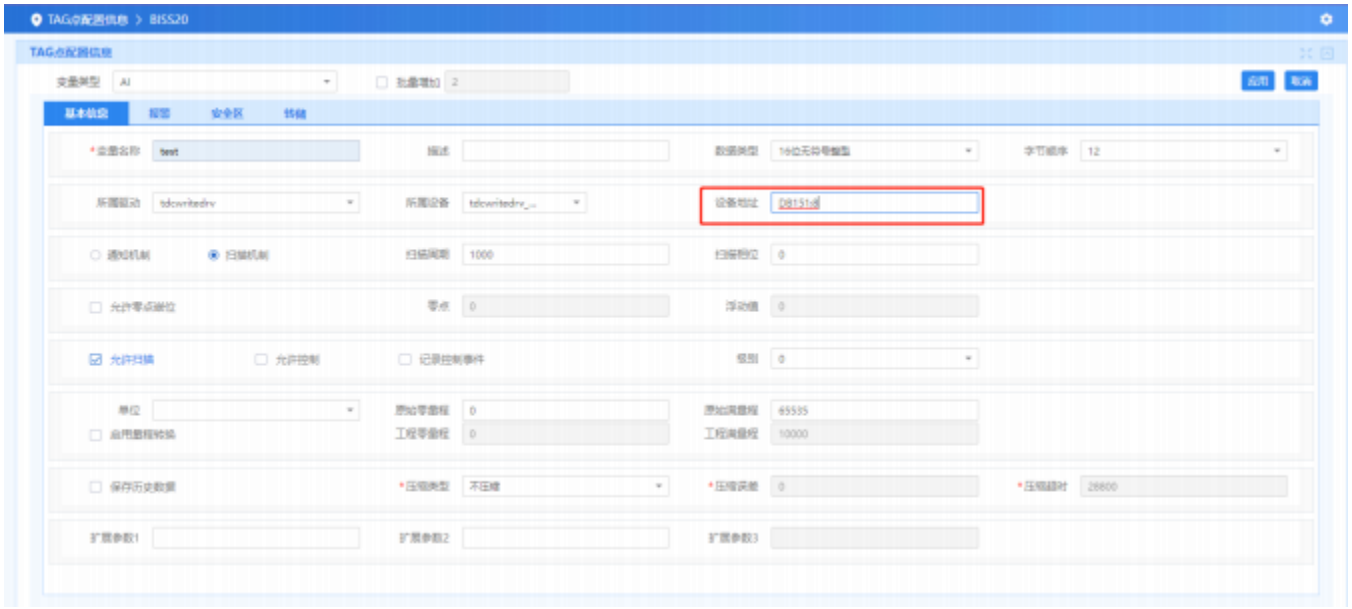
弹出数据块配置界面，如下图：

- 参数说明
- 数据块名称：自定义数据块名，例如：SendToTDC
- 轮询周期：
- 相位：
- 类型：
- 地址：设备所在的IP地址
- 长度：数据块的长度。（数据块的长度=整体报文长度-2字节TelLen-2字节TelID）

•可选参数1：为电文号TelID
•数据偏移说明（tag点配置注意事项）：
数据块TCPServ总长20byte，tag点配置时读取数据的有效范围TCPServ:4~TCPServ:19

例如，下图红框中的偏移【DB151:8】，表示从有效范围
TCPServ:4~TCPServ:19的起始位置向后偏移8byte，开始写入数据。

TCPServ:0~TCPServ:3为电文头(2字节TelLen+2字节TelID)



参考文档：

xcomwifledrv驱动配置说明.pdf

```
 <driver name="tdcwritedrv"type="item"desc="">
                                                                                <device
name="tdcwritedrv_Device0"conntype=""connparam="ip=127.0.0.1;port=502;multiL
                                                                                ink=1"cyc
 </driver>
```

注：有些PLC只能连接一个Client，调试tdcwritedrv驱动时如果连接不上，需要确认是否有其他设备连接了PLC。

•DB2数据源
•.MySQL数据源
•.ODBC数据源
•.Oracle数据源
•.Postgresql数据源
•.SQLServer数据源
•.StringN类型案例
•.数据库连接
•.配置db驱动
•.配置对应变量
•连接oracle数据库

《db驱动》用于从关系数据库采集数据的驱动，我们这里以dbdriver连接db2数据库为例。配置主要包含以下步骤：

1.数据库连接

2. 配置db驱动
3. 配置对应变量
4. StringN类型变量案例

各种数据库配置：

- ODBC数据源
- Oracle数据源
- DB2数据源
- MySQL数据源
- SQLServer数据源
- Postgresql数据源

典型案例：

- 连接oracle数据库

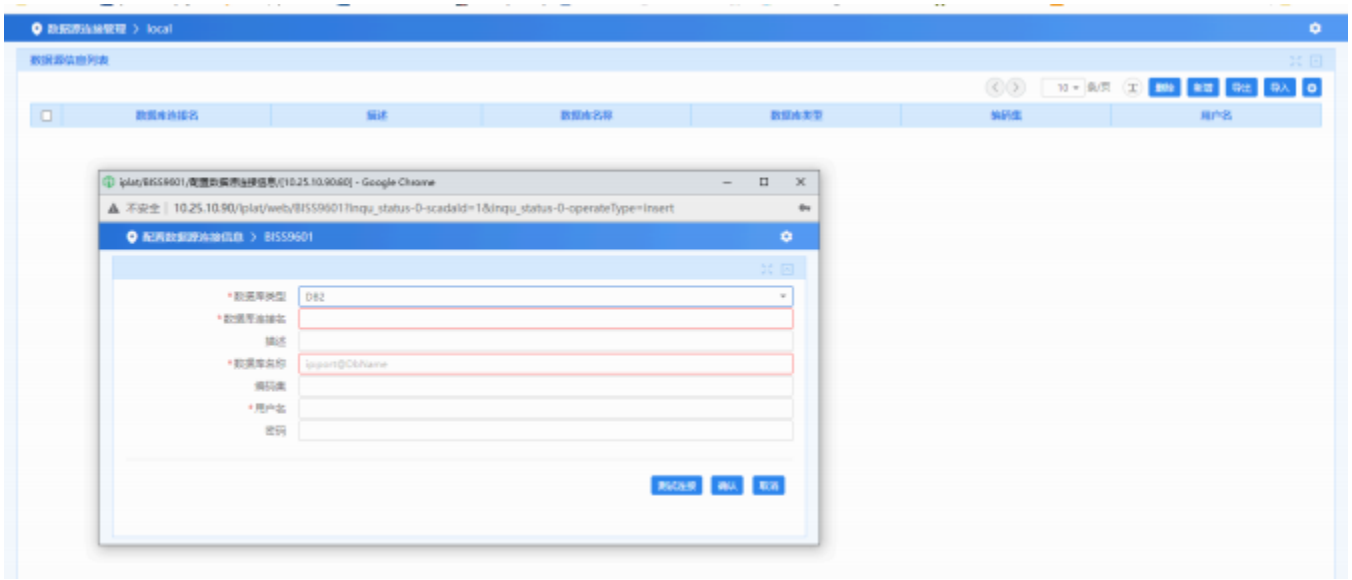
DB2数据源

本章主要介绍两部分内容：

- 如何在数据源中配置连接的DB2数据源
- 如何在系统中配置DB2数据源

如何在数据源中配置连接的DB2数据源

转储数据源配置界面如下：



❖数据库连接：数据库连接的名称，必填，自定义字符串。

❖描述：数据源的描述，选填，自定义字符串。

❖数据源类型：选择DB2。

❖数据源名称：填入在本地DB2客户端中配置的别名。参见如何在系统中配置DB2数据源。

❖用户名：登陆该数据库使用的用户名。

❖密码：登陆该数据库使用的用户密码。

❖编码集：该数据库适用的编码集，选填

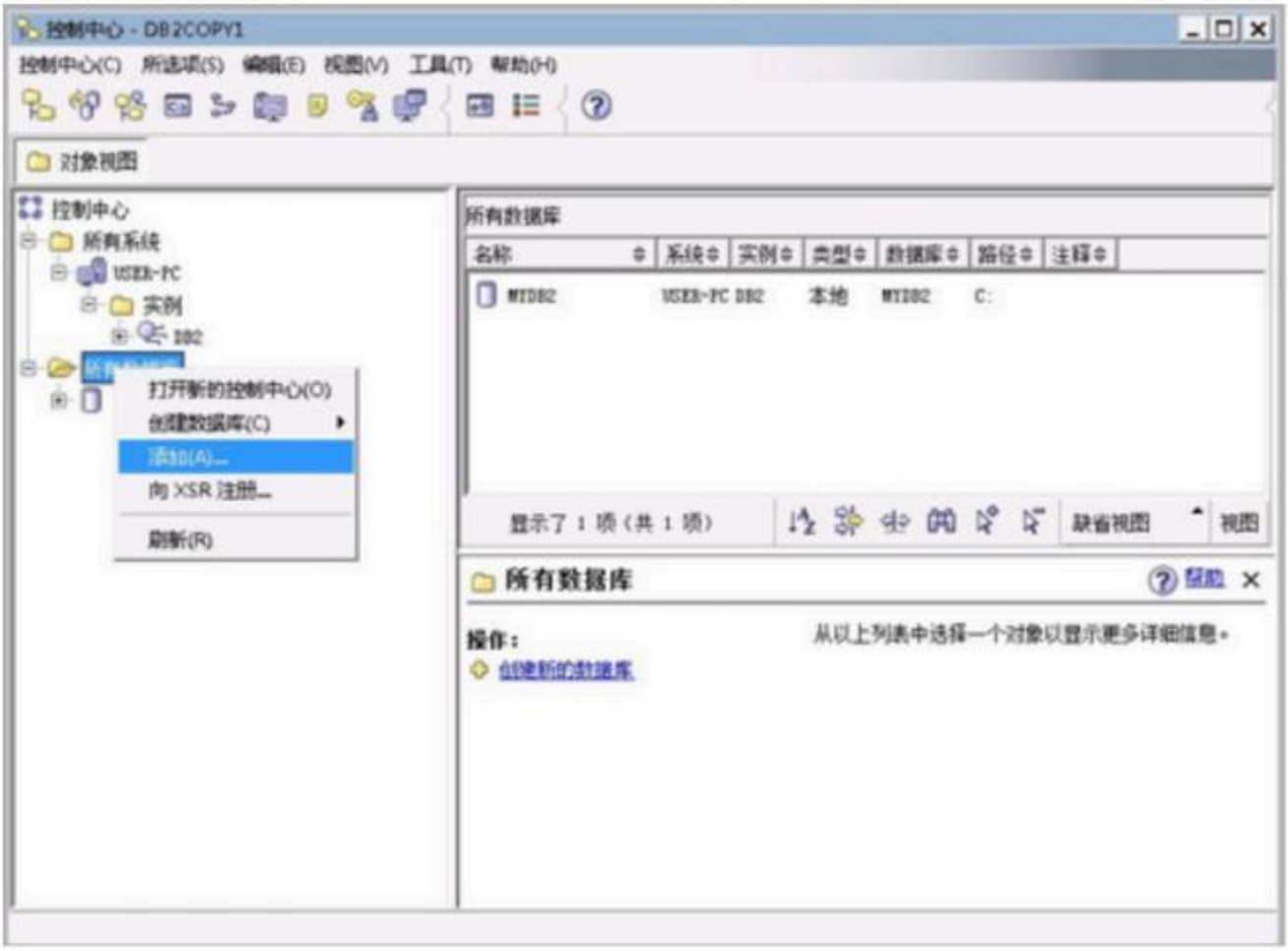
注意：

◇在完成配置后，要签入配置才能生效。

◇若要进行sql转储，则在编码集中设置适用编码集

如何在系统中配置DB2数据源

1. 安装DB2客户端后，打开DB2控制中心，如下图：



2. 右键点击"所有数据库"，点击"添加"按钮，如下图：

❖主机名：输入需要连接的DB2数据库的IP地址。点击"搜索"按钮后，会将所有的数据库显示在数据库列表中。

❖新的数据库别名：自定义的数据库别名。

注意：此处的数据库别名对应到转储配置中的数据源名称。

3. 配置完成后，点击"确定"按钮，完成配置。

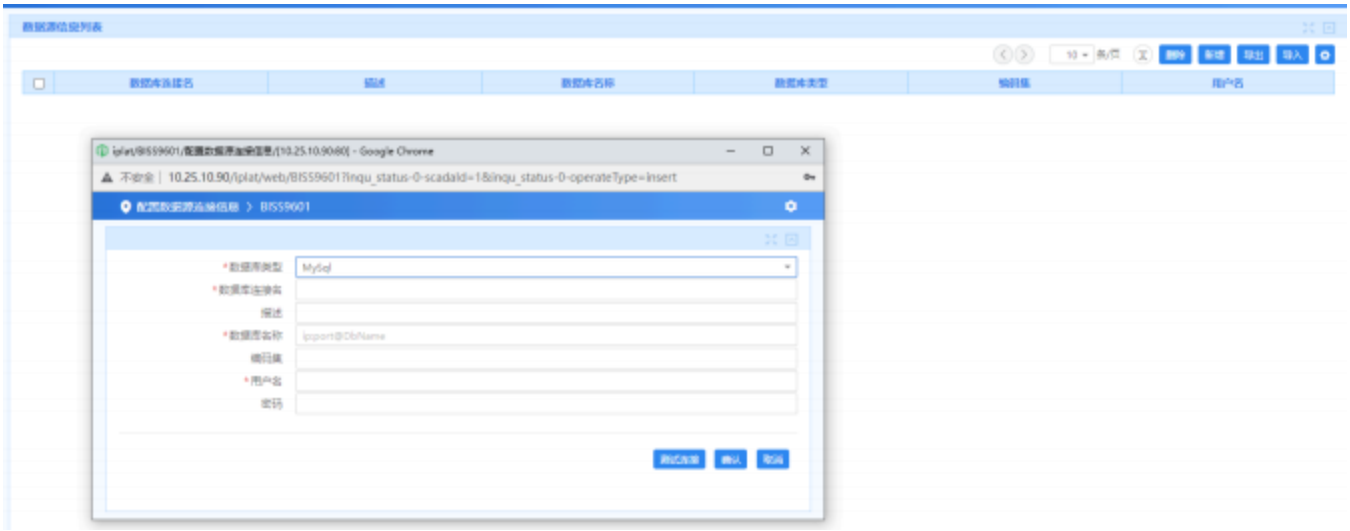
如果要远程采集DB2的数据，需要在采集机器上搭建【本地编目】

搭建DB2客户端数据采集环境

.MySQL数据源

如何在数据源中配置连接的MySQL数据源

转储数据源配置界面如下：



❖数据库连接：数据库连接的名称，必填，自定义字符串。

❖描述：数据源的描述，选填，自定义字符串。

❖数据源类型：选择MySQL。

❖数据源名称：格式为"数据库IP@数据库名称"。

❖用户名：登陆该数据库使用的用户名。

❖密码：登陆该数据库使用的用户密码。

❖编码集：该数据库适用的编码集，选填

注意：

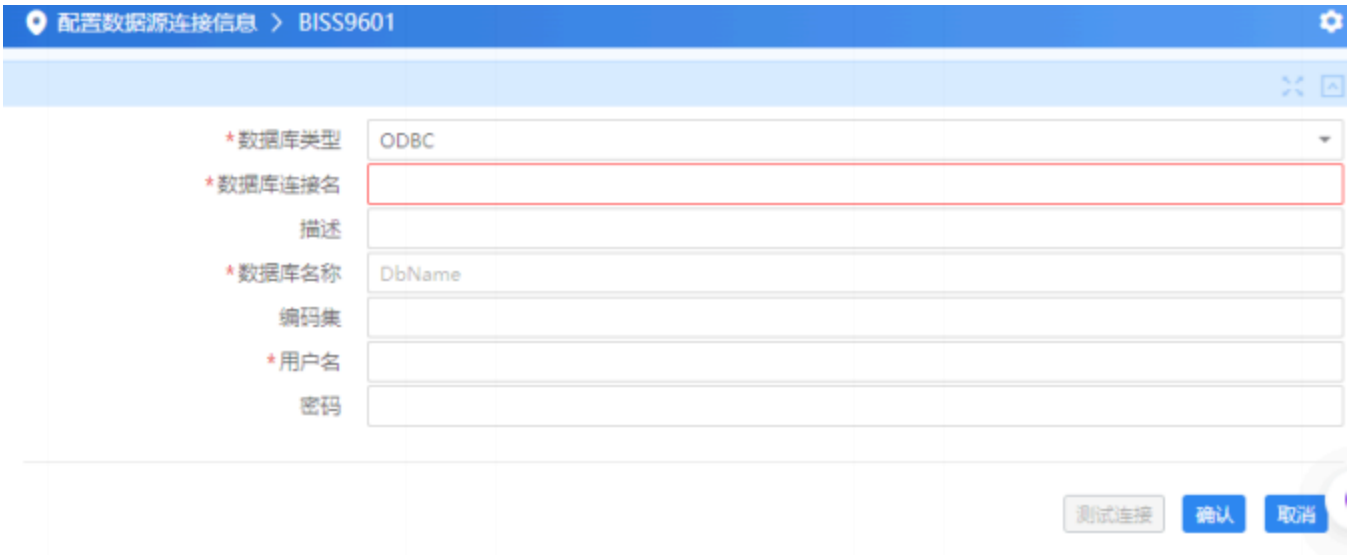
◇在完成配置后，要签入配置才能生效。
◇若要进行sql转储，则在编码集中设置适用编码集

◇要实现MySQL的转储，必须依赖于与服务器mysql版本对应的libmysql.dll，请注意是32位的还是64位的。由于MySQL版本较多，为了保证匹配，在工程启动前，需要将对应的libmysql.dll（MYSQL安装目录下的libmysql.dll）拷贝到iplatda安装目录下的executable目录下。

.ODBC数据源

如何在数据源中配置连接的ODBC数据源

转储数据源配置界面如下：



- 数据源类型：选择ODBC。
- 数据库连接名：数据库连接的名称，必填，自定义字符串。
- 描述：数据源的描述，选填，自定义字符串。
- 数据库名称：
- 数据源名称：填入在系统ODBC中配置的ODBC名称。参见如何在系统中配置ODBC数据源
- 用户名：登陆该数据库使用的用户名。
- 密码：登陆该数据库使用的用户密码。
- 编码集：该数据库适用的编码集，选填

注意：

1. 在完成配置后，要部署配置才能生效。
2. 若要进行sql转储，则在编码集中设置适用编码集

如何在系统中配置ODBC数据源

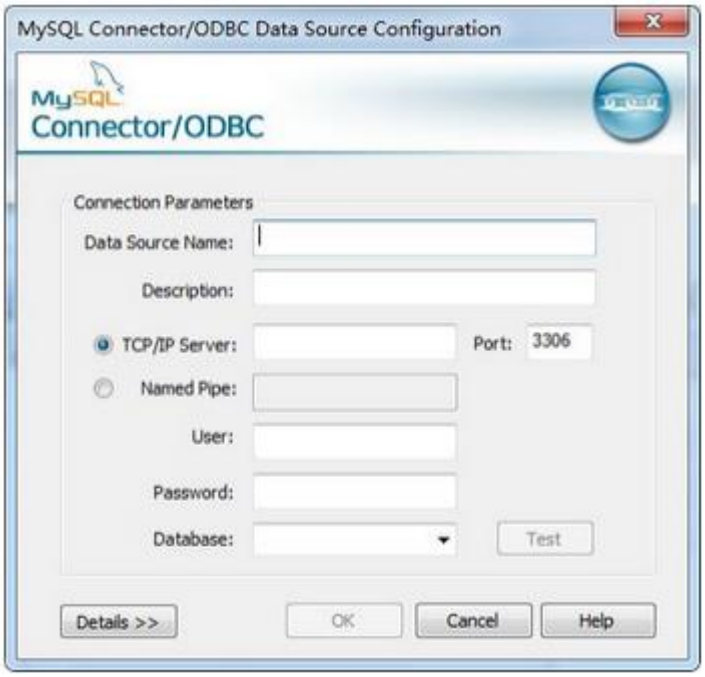
注：在64位Windows系统中，默认数据源(ODBC)是64位的，包括控制面板-》管理工具-》数据源或在运行中直接运行ODBCAD32程序。

1. 打开操作系统的"控制面板"-"管理工具"-"数据源ODBC"（或在命令行中输入odbc），打开ODBC配置界面。如下图：

2.点击"添加"按钮，显示"数据源驱动程序选择"界面，如下图：

注意：若需要将Oracle或DB2的数据库配置成为ODBC数据源，则需要首先在系统中安装对应的客户端，才能在此处进行驱动程序选择。

3.选择驱动程序后，点击"完成"按钮，进入"数据源"配置界面，这里以Mysql为例，如下图：



4.在配置界面中填入对应的信息，即完成ODBC源的配置。

注意：此处填入的Data Source Name即为需要填入到宝信基础自动化软件中转储配置中的数据源名称。

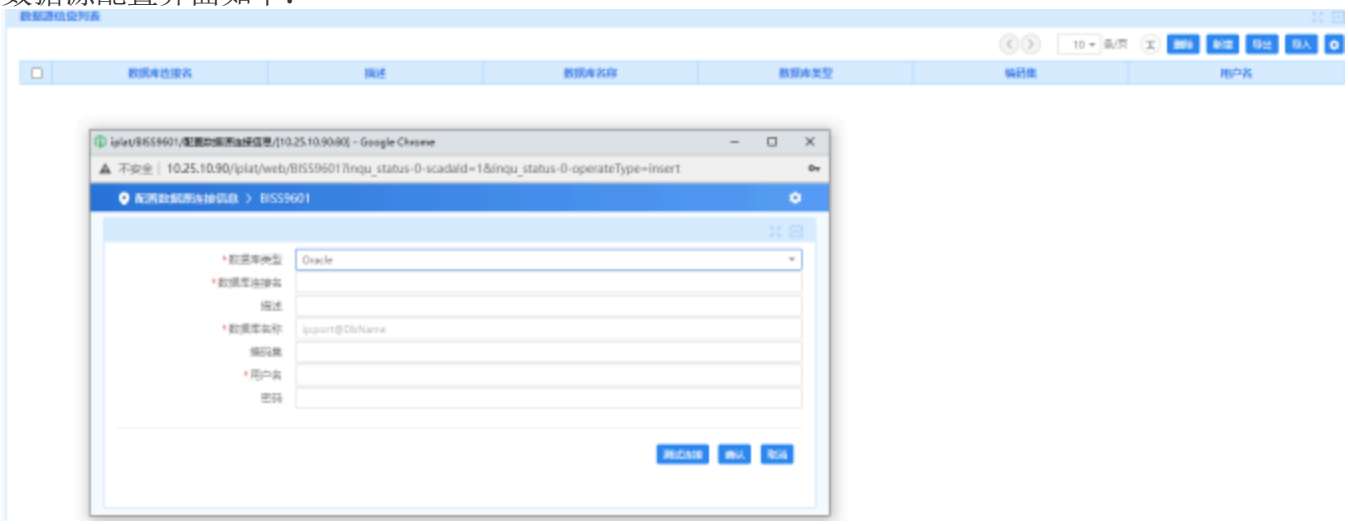
.Oracle数据源

本章主要介绍两部分内容：

- 如何在数据源中配置连接的Oracle数据源
- 如何在系统中配置Oracle数据源

如何在数据源中配置连接的Oracle数据源

数据源配置界面如下：



❖数据库连接：数据库连接的名称，必填，自定义字符串。

❖描述：数据源的描述，选填，自定义字符串。

❖数据源类型：选择Oracle。

❖数据库名称：填入在本地Oracle客户端中配置的Oracle网络服务名。参见如何在系统中配置
Oracle数据源。若配置完后，连接成功了，但是数据转储不了，请尝试在网络服务器名称前面加上ip，如127.0.0.1@orcl

若无法连接,请尝试格式**//ip:port/SID,SID为

oracle数据库名.**❖用户名：登陆该数据库使用的用

户名。

❖密码：登陆该数据库使用的用户密码。

❖编码集：该数据库适用的编码集，选填

注意：

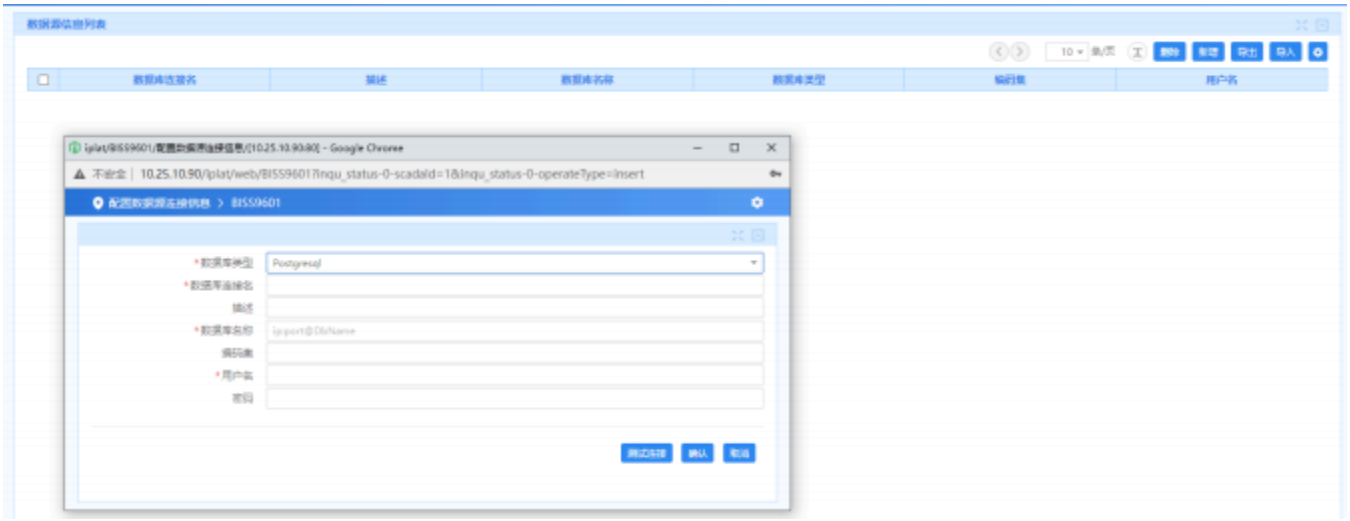
◇在完成配置后，要签入配置才能生效。

◇若要进行sql转储，则在编码集中设置适用编码集

.Postgresql数据源

如何在数据源中配置连接postgresql的数据源

转储数据源配置界面如下：



❖数据库连接：数据库连接的名称，必填，自定义字符串。

❖描述：数据源的描述，选填，自定义字符串。

❖数据源类型：选择PostgreSQL。

❖数据源名称：格式为"数据库IP@数据库名称"。

❖用户名：登陆该数据库使用的用户名。

❖密码：登陆该数据库使用的用户密码。

❖编码集：该数据库适用的编码集。当遇到有中文需要转码的时候，需要填上GBK或者utf-8编码集形式。

注意：

◇若要进行sql转储，则在编码集中设置适用编码集

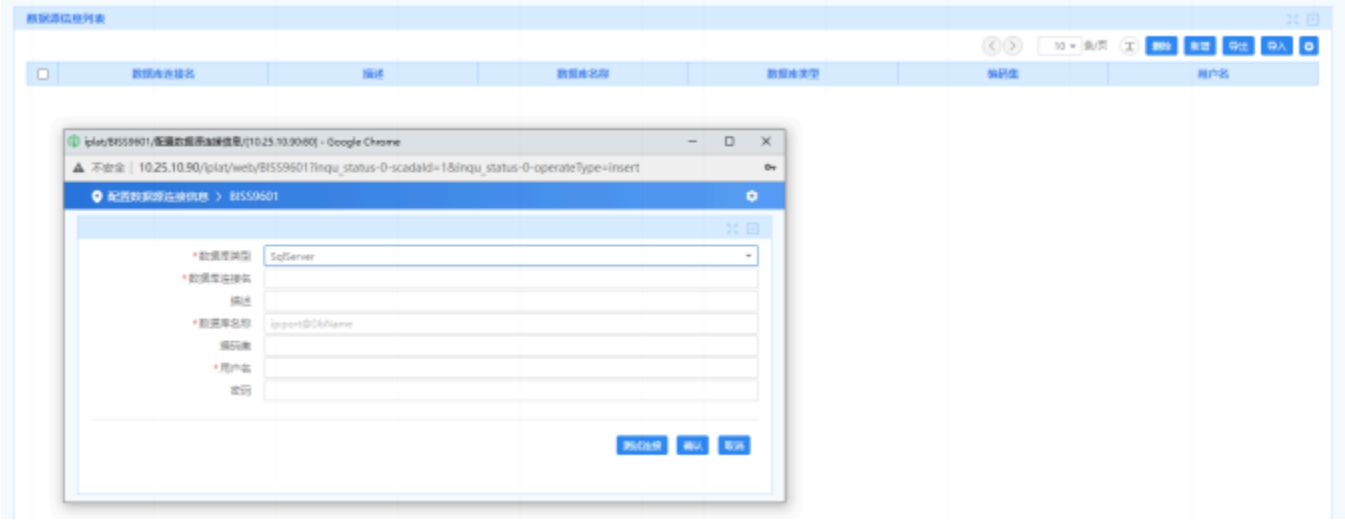
◇要实现postgresql的转储，必须依赖于与服务器版本对应的libpq.dll，请注意是32位的还是64位的。在工程启动前，需要将对应的libpq.dll（pgsql安装目录下的libpq.dll）拷贝到iplat安装目录下的executable目录下。

◇如果需要iplat远程连接postgreSQL，则需要在服务端安装VC2013runtime的环境。且postgres的配置文件pg_hba需要添加远程ip的信任，在IPv4下面添加host all all0.0.0.0/0 trust。

.SQLServer数据源

如何在数据源中配置连接的SQLServer数据源

转储数据源配置界面如下：



❖数据库连接：数据库连接的名称，必填，自定义字符串。

❖描述：数据源的描述，选填，自定义字符串。

❖数据源类型：选择SQLServer。

❖数据源名称：格式为"数据库IP@数据库名称"，如果测试连接失败的话也可以试试"数据库IP，端口号@数据库名称"（逗号为英文版的逗号）

❖用户名：登陆该数据库使用的用户名。

❖密码：登陆该数据库使用的用户密码。

❖编码集：该数据库适用的编码集，选填

注意：

◇若要进行sql转储，则在编码集中设置适用编码集

.StringN类型案例

选择dbdriver驱动。

添加设备：

首先要进行数据库连接

数据库连接：选择已连接的数据库，本例以MYSQL为例。

连接参数：默认设置，不可修改。

添加数据块：

如图中配置。

元素个数：数据库表的行数；

元素大小（字节）：每行的总长度；

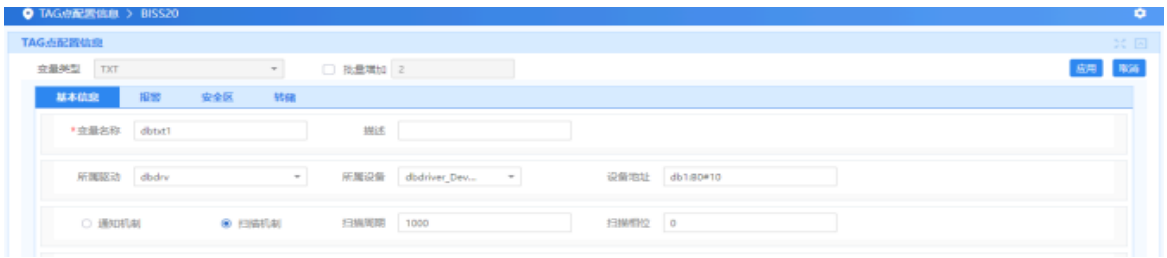
SQL语句；SQL语句，填写想要获得的列；

数据类型：填写stringN类型，赋予每列长度；string后数字表示数据有多少字节。

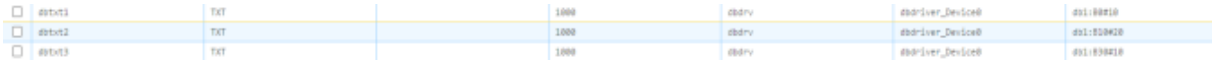
点击确定，并签入dbdriver驱动。

配置变量：

添加txt类型变量，



按此配置三个变量如下图：

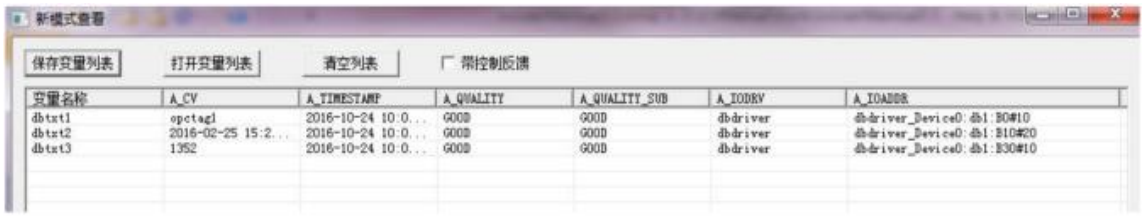


部署。

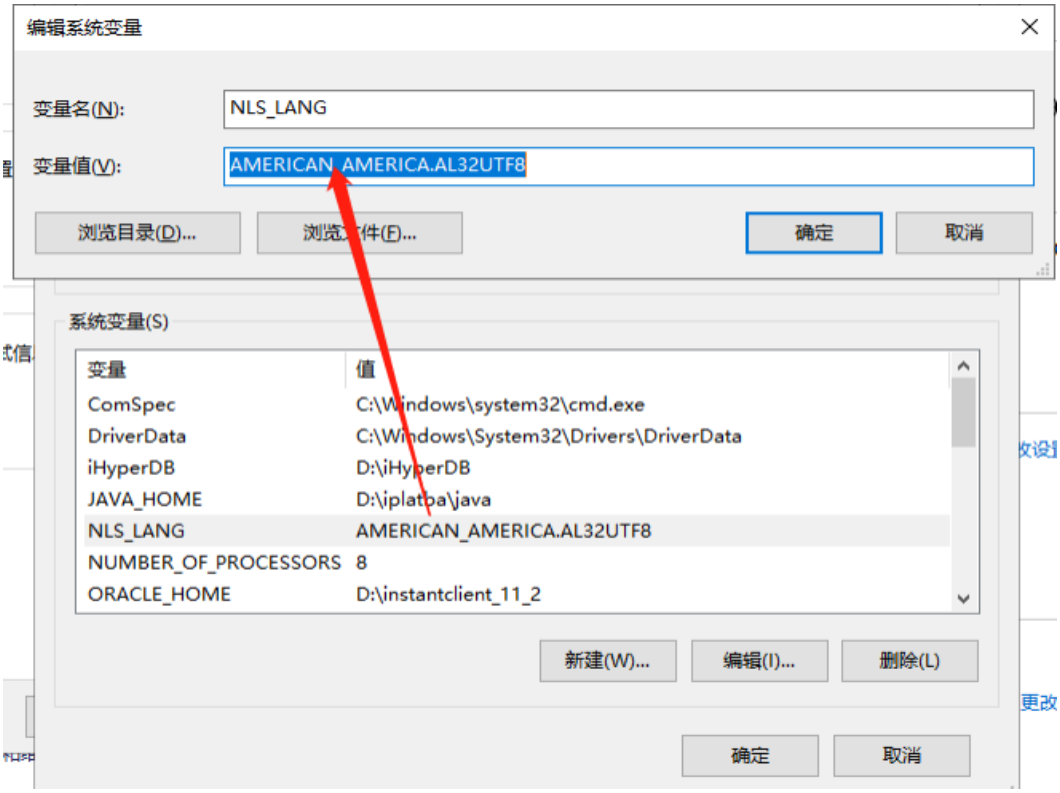
数据库中表为：

	name	time	quality	value
▶	opctag1	2016-02-25 15:21:27.739	192	1352
	opctag1	2016-02-25 15:21:32.739	192	1942
	opctag1	2016-02-25 15:21:37.740	192	2336
	opctag1	2016-02-25 15:21:42.740	192	2929

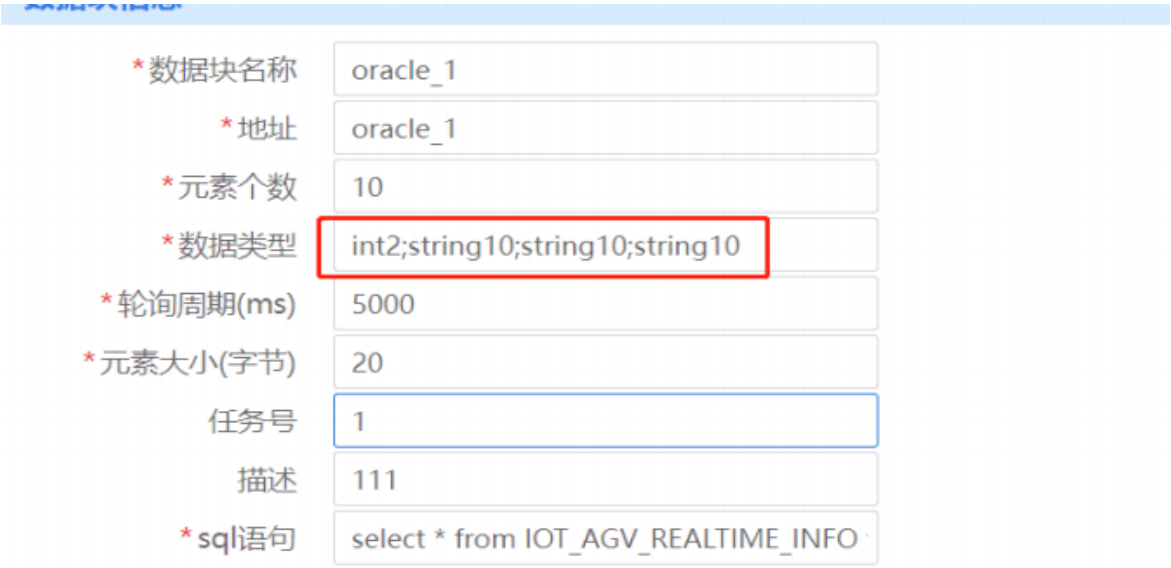
即可读出表中的数据：



注：在dbdriver驱动采集oracle数据库时，若采集的是中文会出现乱码，原因是中文编码集处理有问题，需要在da所在的服务器上配置环境变量，如下图：



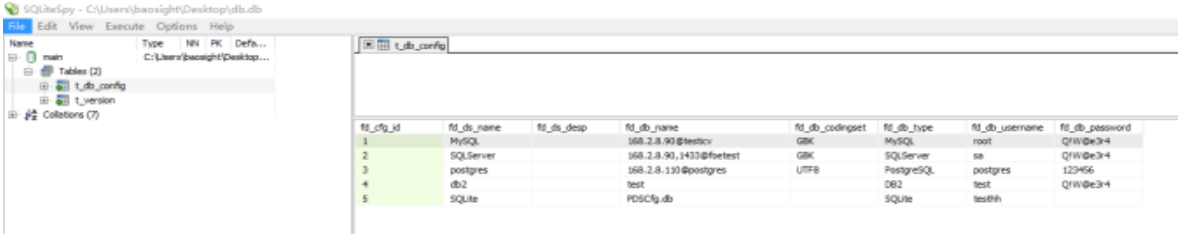
并且在配置数据块的数据类型时需要保证该数据类型的个数和sql语句所查字段相等，且类型也要保证相同（int的类型是int2）

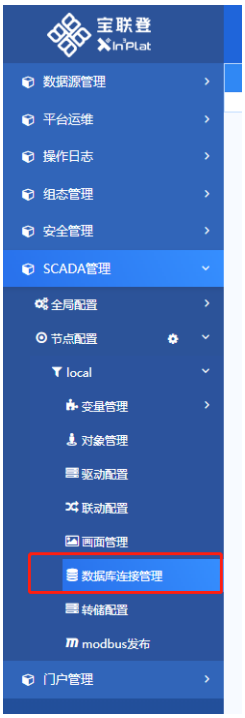


.数据库连接

数据库连接是一个通用的功能。无论是使用数据库作为采集数据的数据源或者是采集数据后转储到数据库，其中涉及到连接数据库，都是由这一块来进行统一管理。

数据库连接配置完成后，配置信息将会生成一个SQLite文件，保存在iplat-da\projects\defaultproject\config目录下：





数据源连接内容如下：

- 添加数据源
- 修改数据源
- 删除数据源
- 导出导入
- 测试连接
- 查看日志

各数据库的配置参见：

ODBC数据源

Oracle数据源

DB2数据源

MySQL数据源

SQLServer数据源

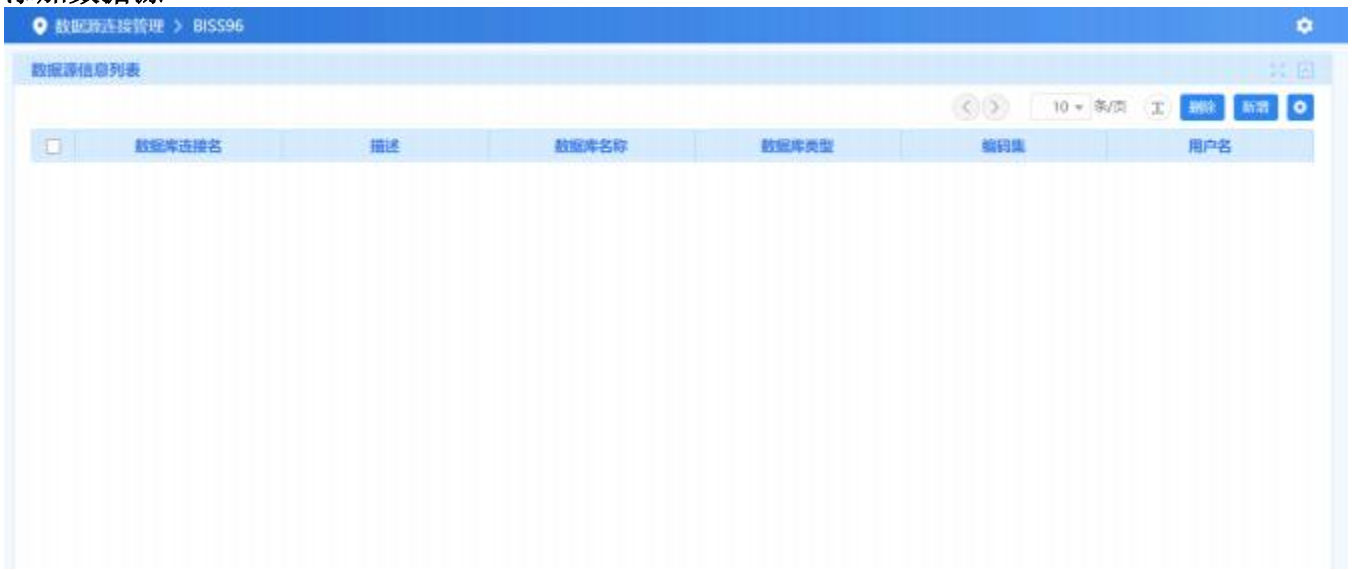
Postgresql数据源

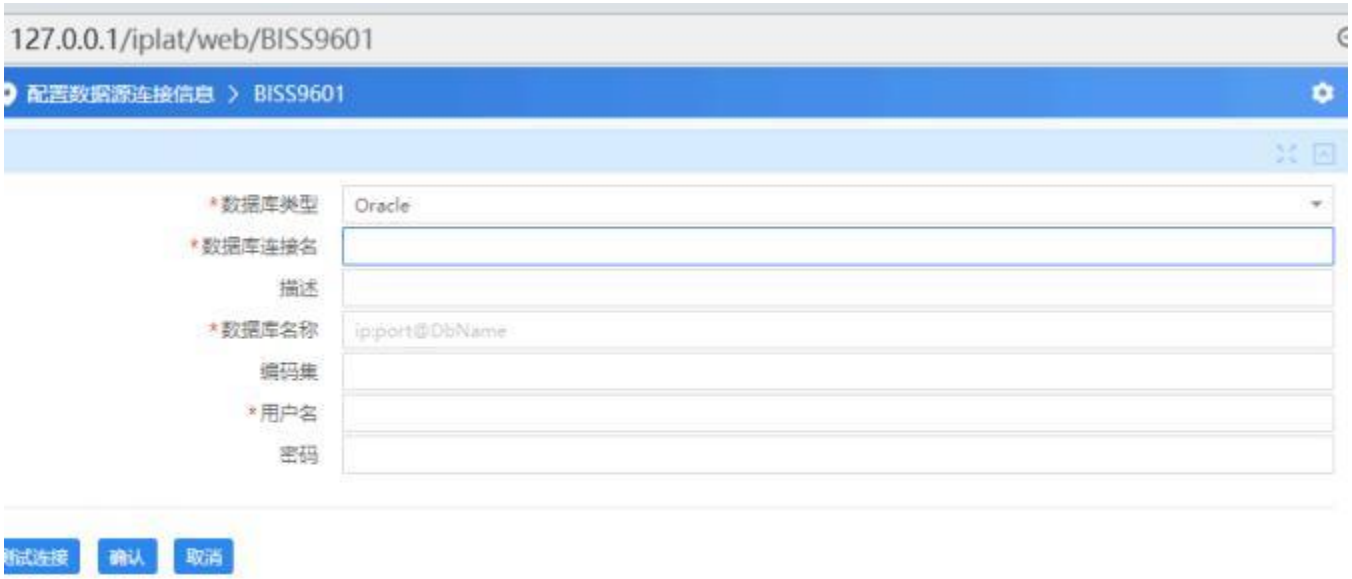
在iplat配置数据库连接的流程如下，每个scada节点的数据库连接信息保存在该scada的节点配置下：

数据库连接内容如下：

- 添加数据源
- 修改数据源
- 删除数据源
- 导出导入
- 测试连接

添加数据源





点击新增按钮，在弹出的新页面中配置数据源信息：

数据库类型：可选择Oracle,MySQL,DB2,Sql Server,Postgresql类型，通过下拉框选择。

数据库连接名：用户对这条数据库连接信息的自定义名称。

描述：用户对这条数据库连接信息的描述，选填。

数据库名称：该数据库的ip和名称，写成ip@name的写是保存。
编码集：一般为UTF-8，选填。

用户名：该数据库的用户名。

密码：该数据库的密码。

在配置完成所有信息后，点击保存，新增成功。

修改数据源

双击一条数据库信息，弹出与新增信息相同的页面，修改完成后点击确认，修改成功。

删除数据源

选中一条数据库信息，点击删除。

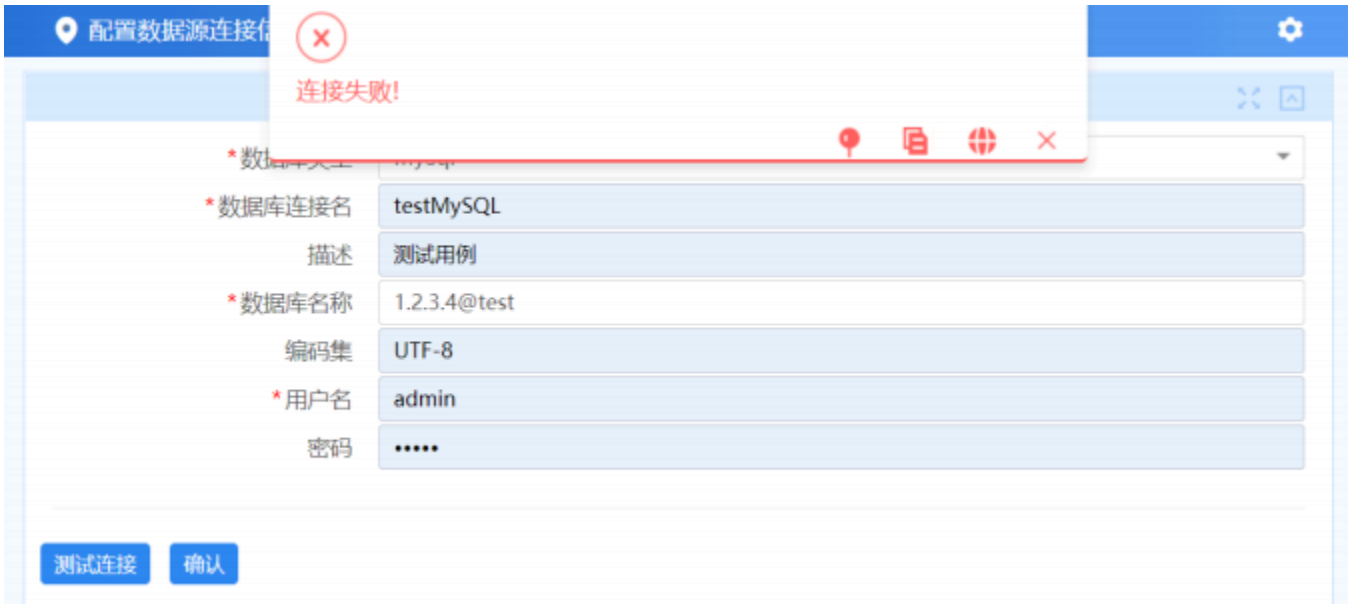
导出导入

导出：点击导出按钮，会将该页面下的所有连接信息以Excel形式通过浏览器下载。

导入：点击导入按钮，选择要导入的文件，导入完成。

测试连接

在新增、修改的BISS9601页面配置完成数据库信息，点击测试连接，返回成功或失败结果。



.配置db驱动

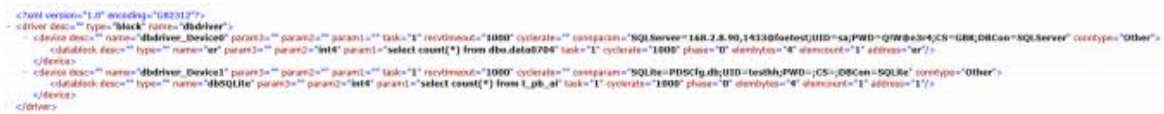
配置db驱动分为3步：

（1）配置驱动（driver）

（2）配置设备（device）

（3）配置数据块（block）

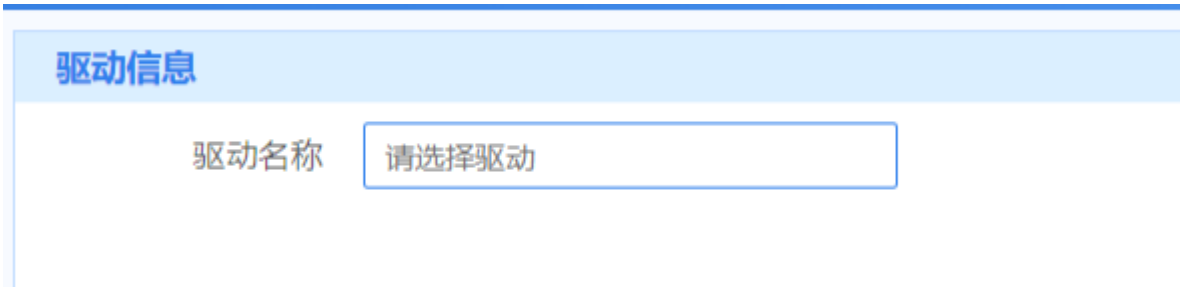
最终生成驱动配置文件格式在工程目录\iplat-da\projects\defaultproject\config\drivers\dbdrv.xml中，如下：



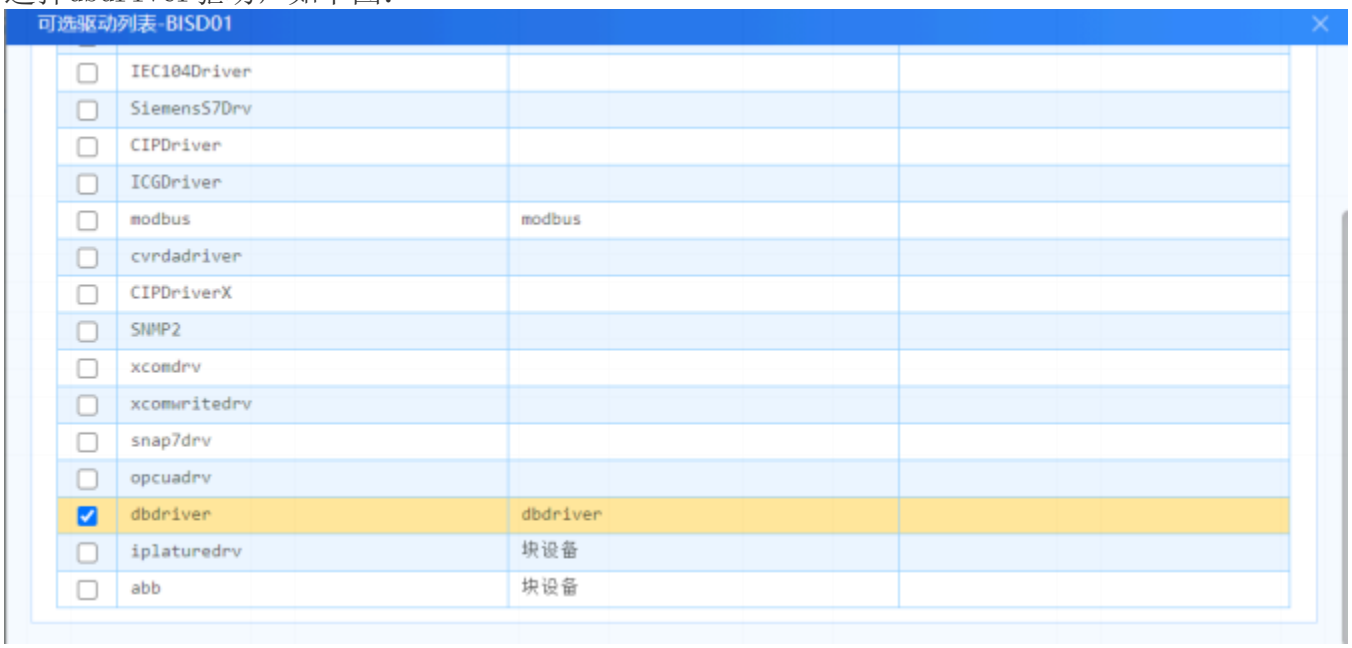
第一步（配置驱动）：

右键点击<当前SCADA下的驱动列表>,点击添加驱动，如下图：

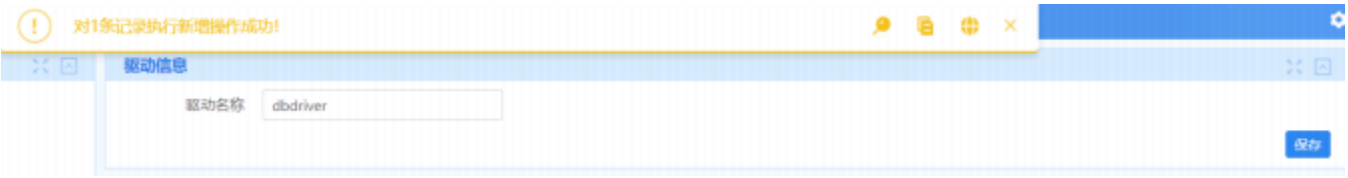
点击请选择驱动，如下图：



选择dbdriver驱动，如下图：



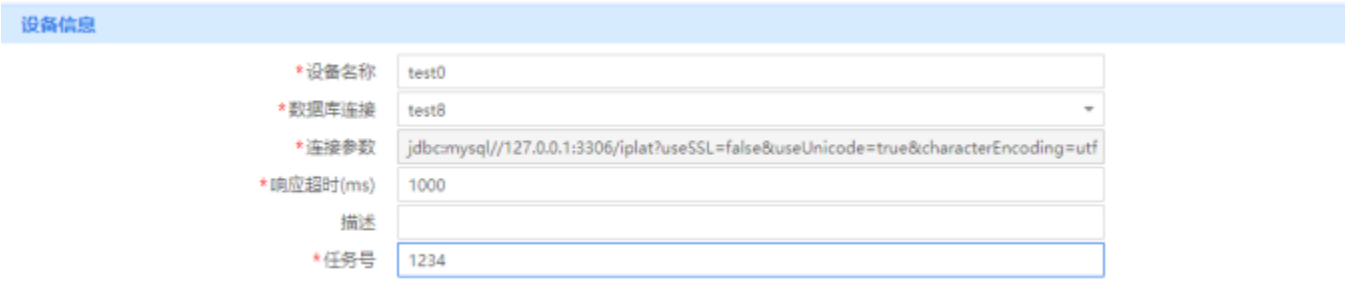
点击右上角的X进入保存界面，点击保存，如下图：



第二步（添加、配置设备）：

右键点击dbdriver选择添加设备，如下图

如下图，在弹出的"设备配置"对话框中配置相关参数：



设备名称：保证名字唯一，由用户指定；

数据库连接：数据库连接中配置好的数据库。数据库连接参见数据库连接。

连接参数：不可修改，自动加载数据库连接中的配置。数据库连接参见数据库连接。

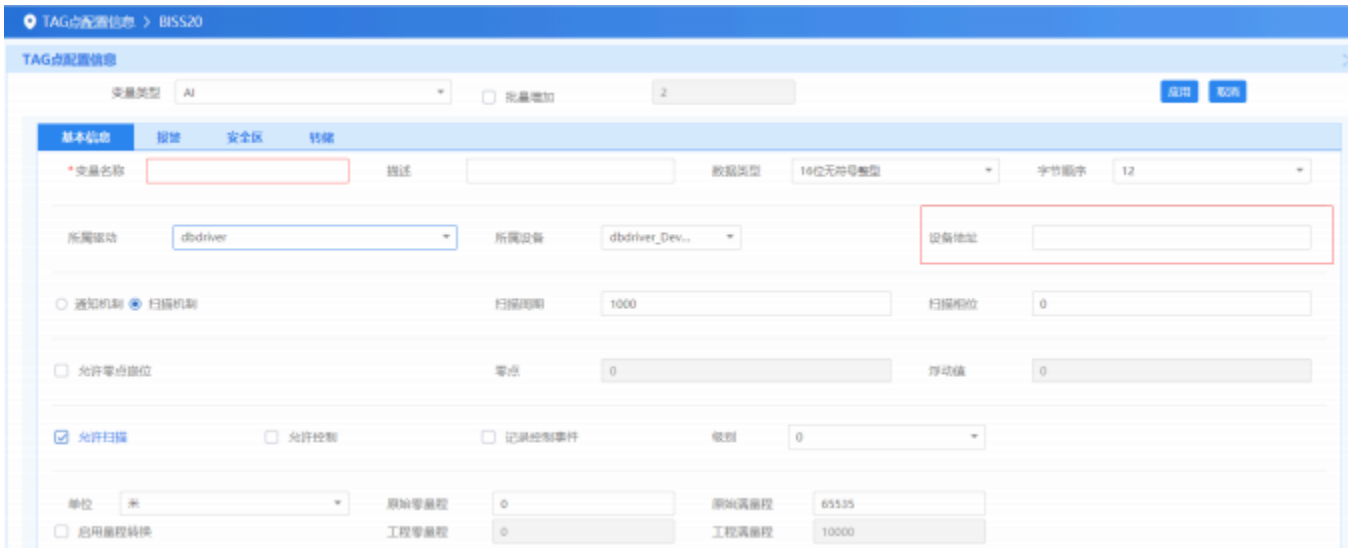
任务号：同一个驱动下的不同设备建议分配不同的任务号。不同的任务号之间是以多线程的并行方式运行，不会相互影响。若通讯正常的设备与通讯故障的设备使用同一个任务号，那么就会降低正常设备的运行速度。任务号的设定，可以用来调整系统中多个通讯的性能。

第三步：选择设备，配置此设备的数据块。

右键点击设备名称选择添加数据块，如下图

弹出"数据块配置"对话框对话框根据用户实际情况进行配置，如下图：

名称：名字唯一，由用户指定。这个名字是变量管理中的设备地址。



地址：地址和数据块名称相同，在变量配置中，使用数据块名称作为变量配置中的设备地址。

元素个数：用户要在数据块中保存的查询记录的最大条数。

元素大小：用户查询的每条记录的元素大小。

任务号：同一个任务号的数据块之间是按照顺序以单线程的方式运行；而不同任务号的数据块之间是以多线程的并行的方式运行，不会相互影响。

SQL语句：需要查询的SQL查询语句。（由于编码集的问题，SQL语句暂不支持中文）

例子：

```
select count(*)from mydb2.table1;
```

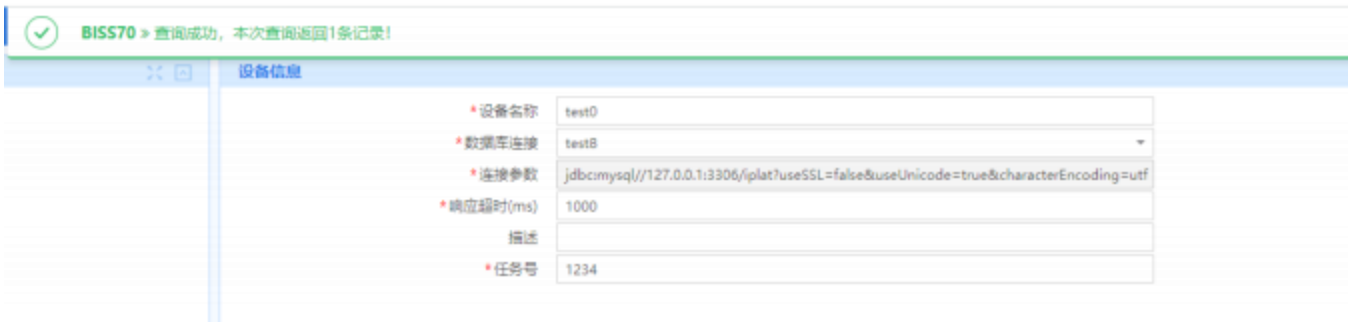
注意：oracle数据库和mysql数据库的可选参数1的配置如同db2数据库，但是oracle数据库的sql语句后面不需要添加分号

数据类型：SQL语句中配置的字段类型，字段类型支持int1，int2，int4，uint1，uint2，uint4，float，double和stringN，stringN意思为最大长度为N个字节的字符串，N的值为大于1且小于等于256

注意：字段类型大小的和不能小于所配置的元素大小；元素大小不能超过256个字节。

轮询周期：SQL语句的执行周期。

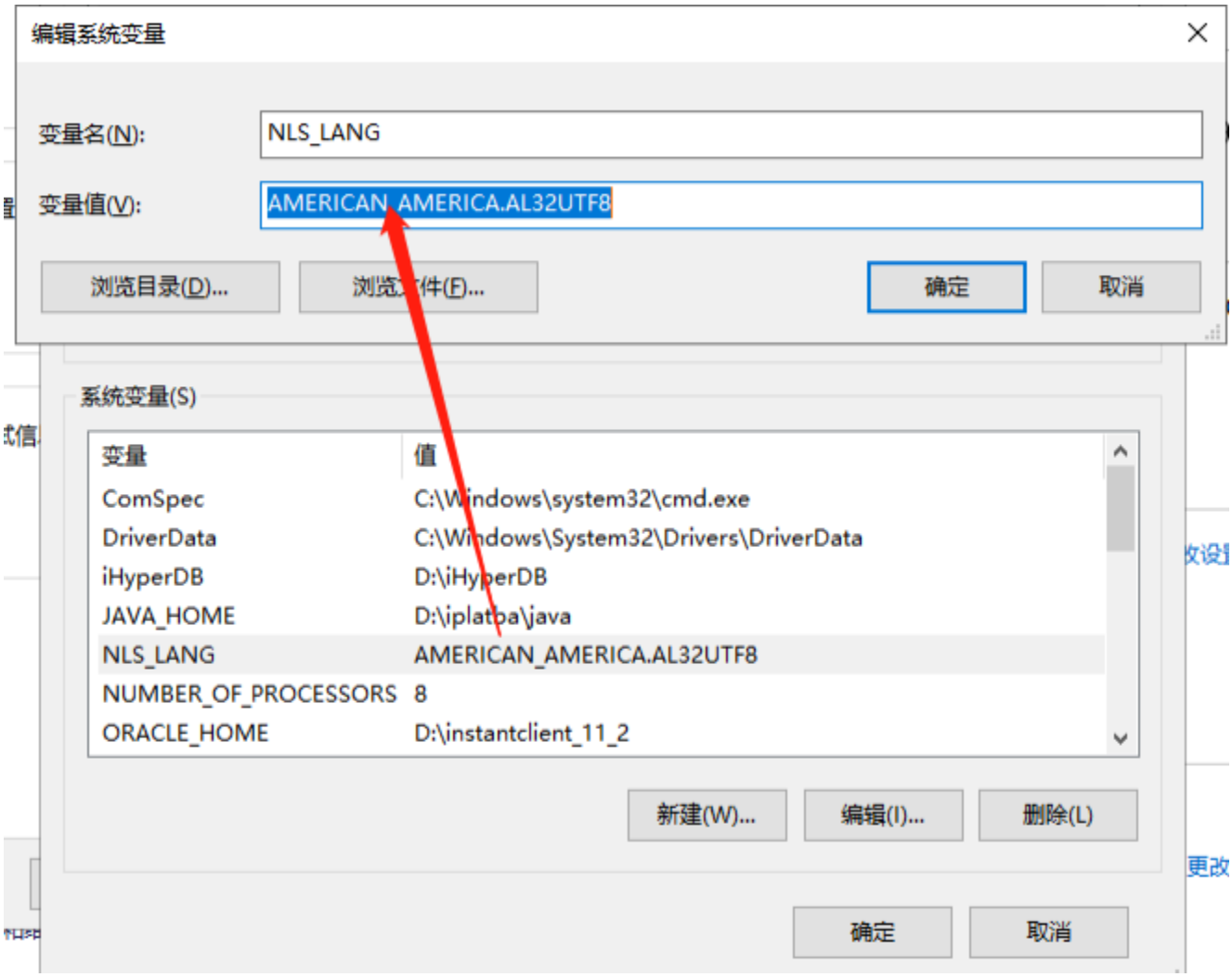
配置结果如下，dbdriver下包含一个device，一个device连接MySQL数据库。每一个device下包含一个数据块。



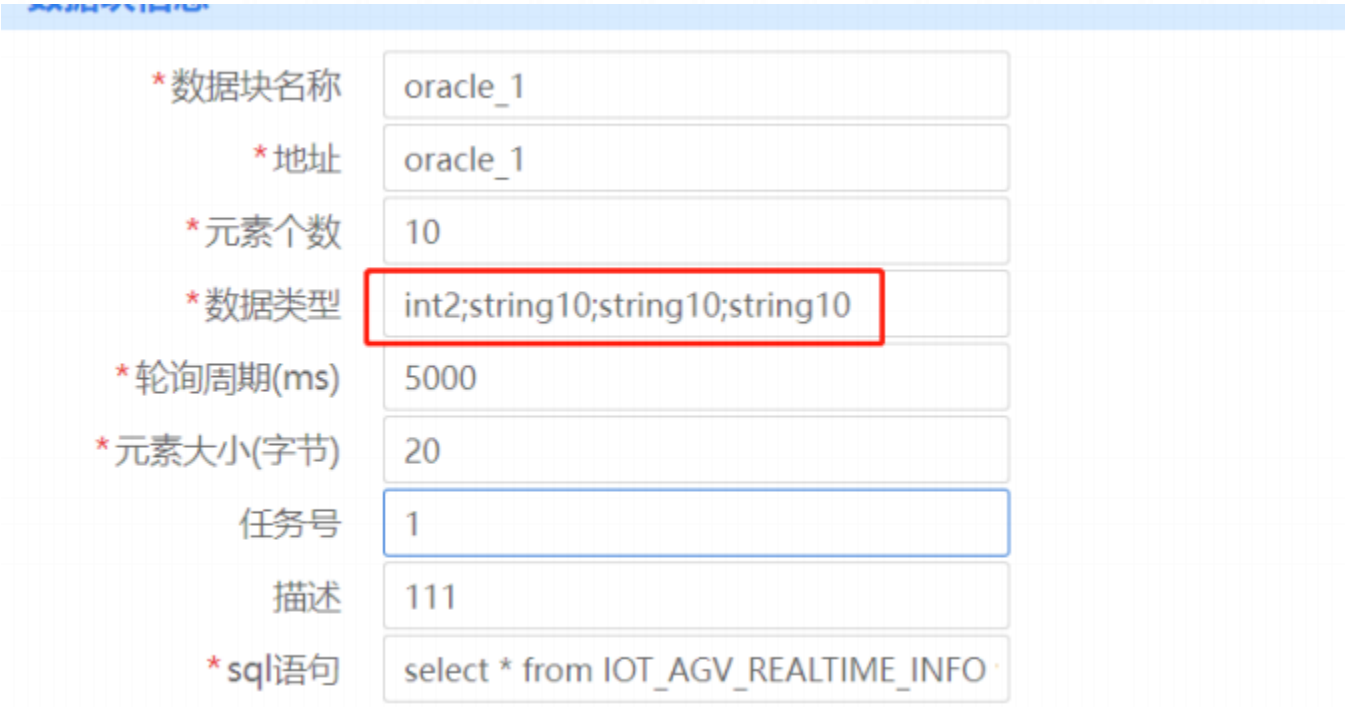
选择设备和数据块列表，右击可以对已经创建的设备和数据块的任意记录进行删除和修改。

设备和数据块列表都支持快捷方式，如：Delect删除、Ctrl+C复制、Ctrl+V粘贴。

注：在dbdriver驱动采集oracle数据库时，需要在服务器上安装oracle客户端，若采集的是中文会出现乱码，原因是中文编码集处理有问题，需要在da所在的服务器上配置环境变量，如下图：



并且在配置数据块的数据类型时需要保证该数据类型的个数和sql语句所查字段相等，且类型也要保证相同（int的类型是int2）



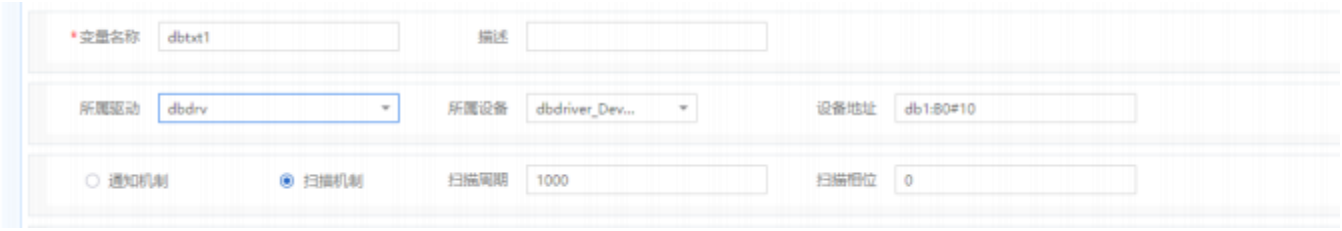
. 配置对应变量

添加变量参见变量管理。

使用db驱动的变量中的设备地址的配置格式为：

blockname:Bx#y数据类型要设置为与block中配置的一致

。



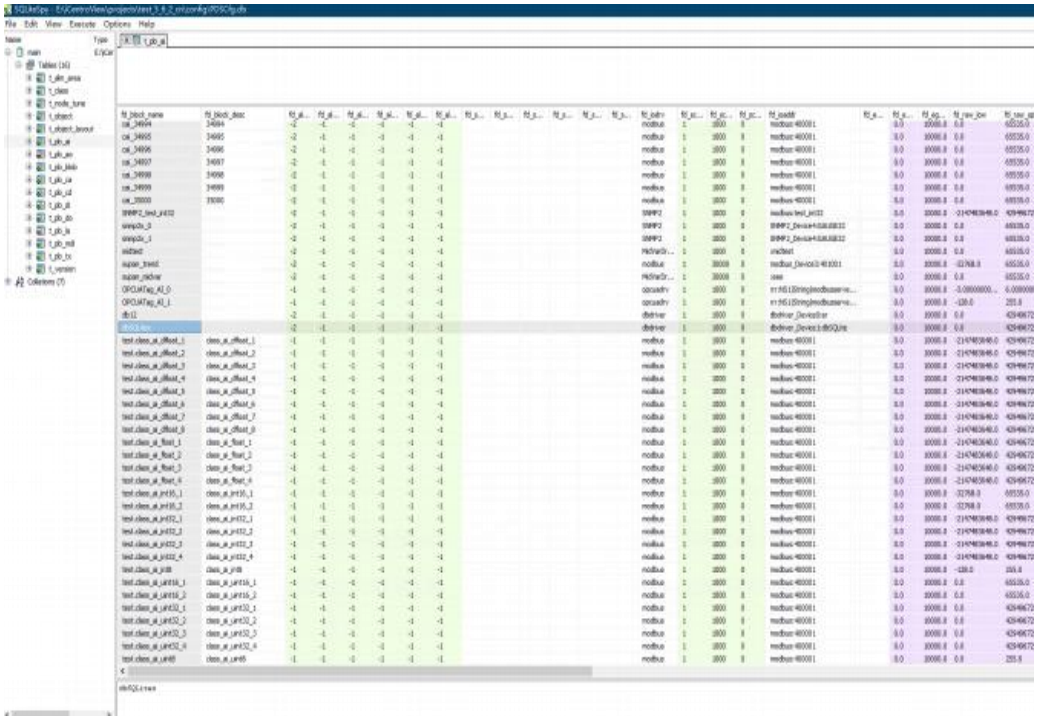
其中blockname表示驱动中对应设备下的对应数据块名；Bx#y，其中B表示Byte，x为偏移量，
表示该数据块的起始地址，最小从0开始；#为特殊标示符，y表示读取的字节个数。上图中的设备地址dbdata1::B3#2就表示为dbdata1数据块下从第三位开始读取2位数据。

具体的数据请根据数据块配置中的元素个数、元素大小和可选参数2的相关配置进行配置。

配置完之后会在PDSCfg.db中生成采集点信息。需要注意的是这几个

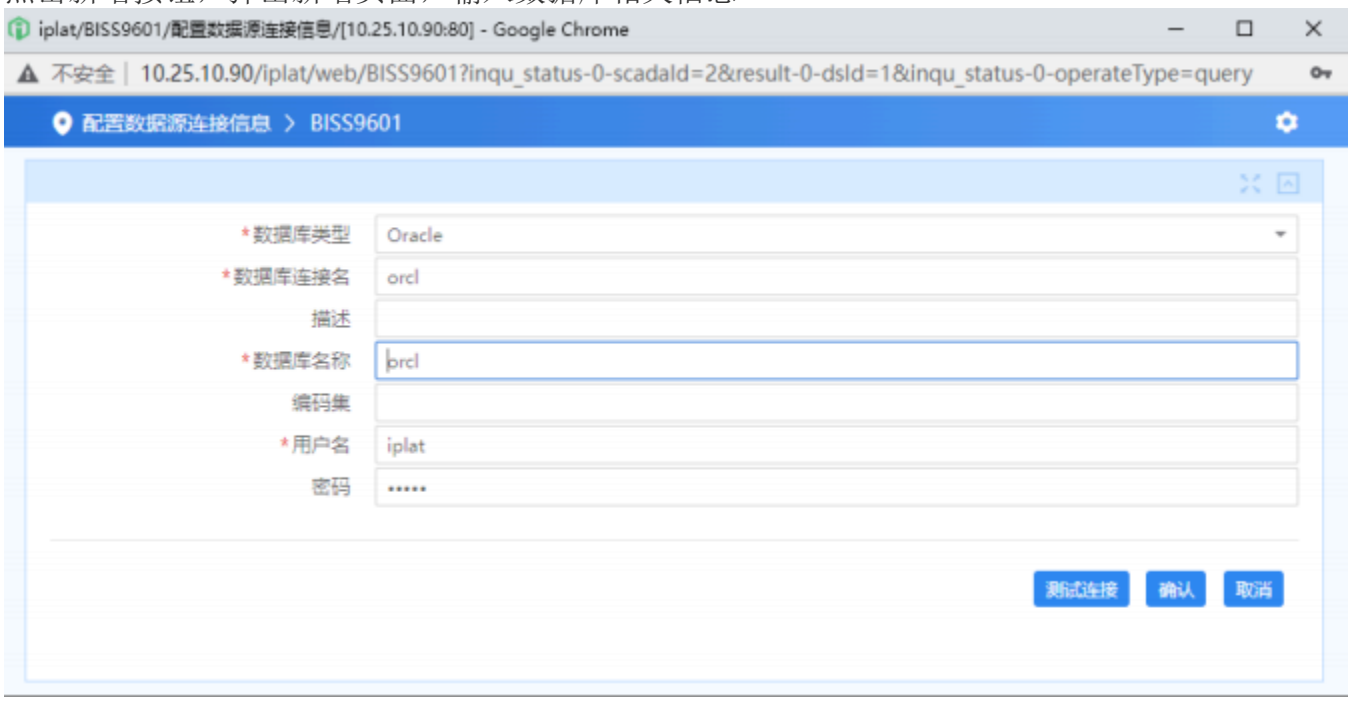
字段：fd_block_name,配置的变量名称

fd_iodrv,驱动名称
fd_ioaddr,设备名称：块名称fd_data_type，数据类型



注意：配置完成后需要部署到scada节点，使得配置生效。
连接oracle数据库
使用dbdrv采集oracle数据库的数据，需要做如下配置：
1. 配置数据库连接

点击新增按钮，弹出新增页面，输入数据库相关信息



数据库类型：选择Oracle
数据库连接名：自定义

描述：自定义

数据库名称：为数据库的SID(实例名)

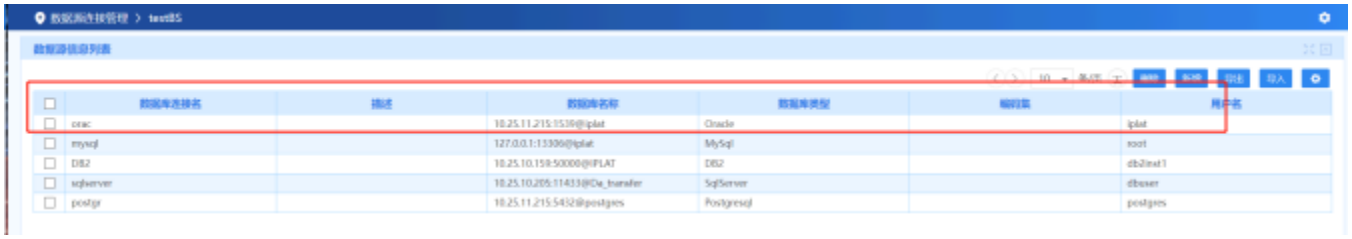
用户名：数据库用户名

密码：数据库密码

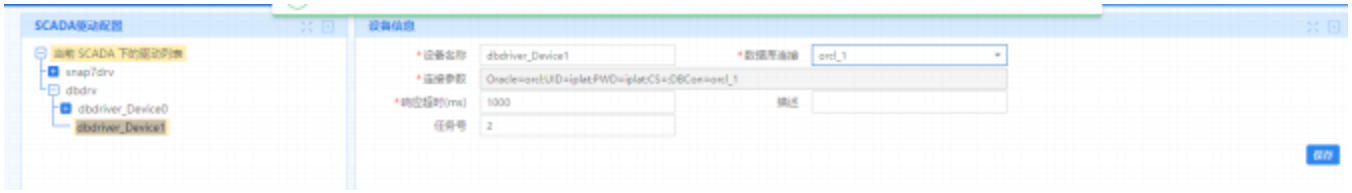
注意：页面上使用JDBC接口，名数库名称的格式为ip:port@SID，使用TNS Name连接。

但驱动使用的是sqlapi，连接字符串格式与JDBC不同，使用ip@SID格式的字符串可能会导致驱动无法连接上Oracle。

因此此处需要按照sqlapi的格式配置，格式为SID或ip:port@SID



数据源配置完成后，还需正确配置驱动。在dbdrv设备配置页面选择数据库连接下拉框，选择之前配置的数据源，页面会自动填充连接参数，如下：

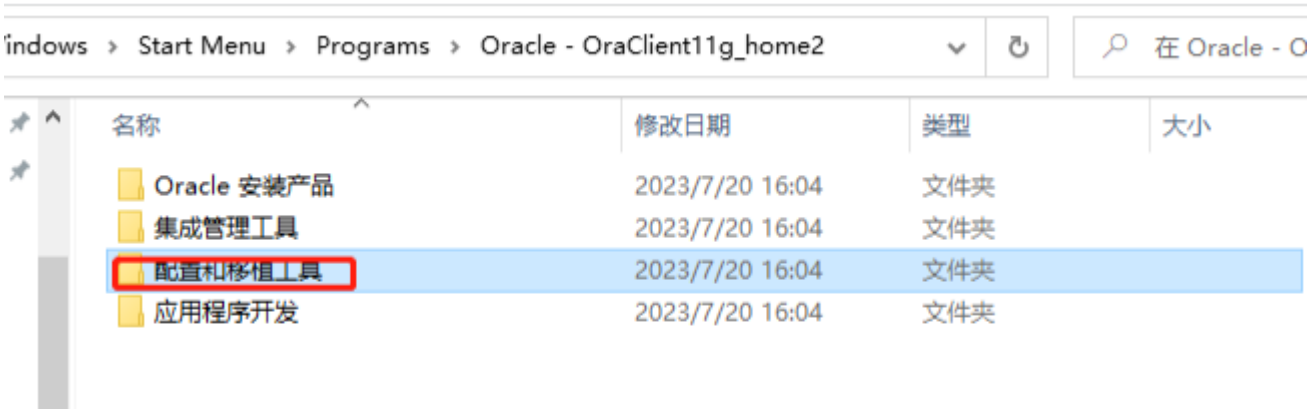


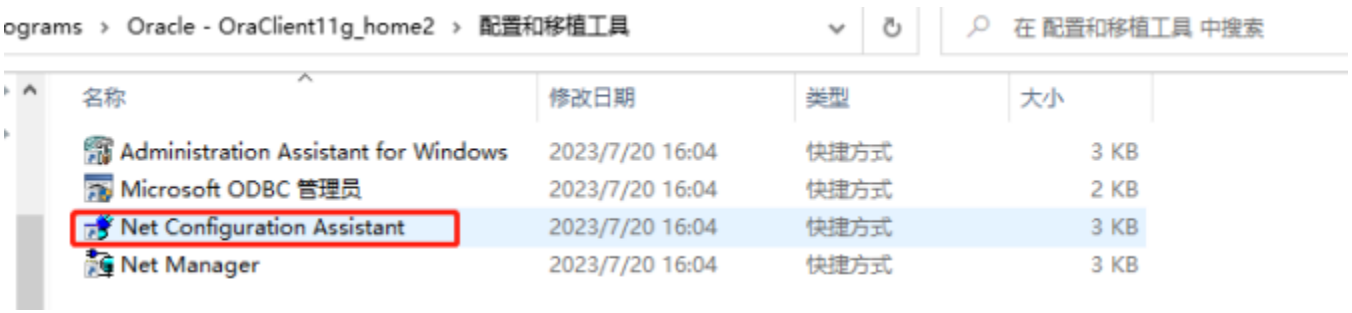
驱动配置完成后，还需检查驱动组件sqlapi依赖的一些动态库是否能够正确加载。

sqlapi连接oracle需要加载一些动态库，因此使用dbdrv采集oracle数据库，还需要安装oracle客户端进行一些配置。

安装oracle客户端

打开Net Configuration Assistant。





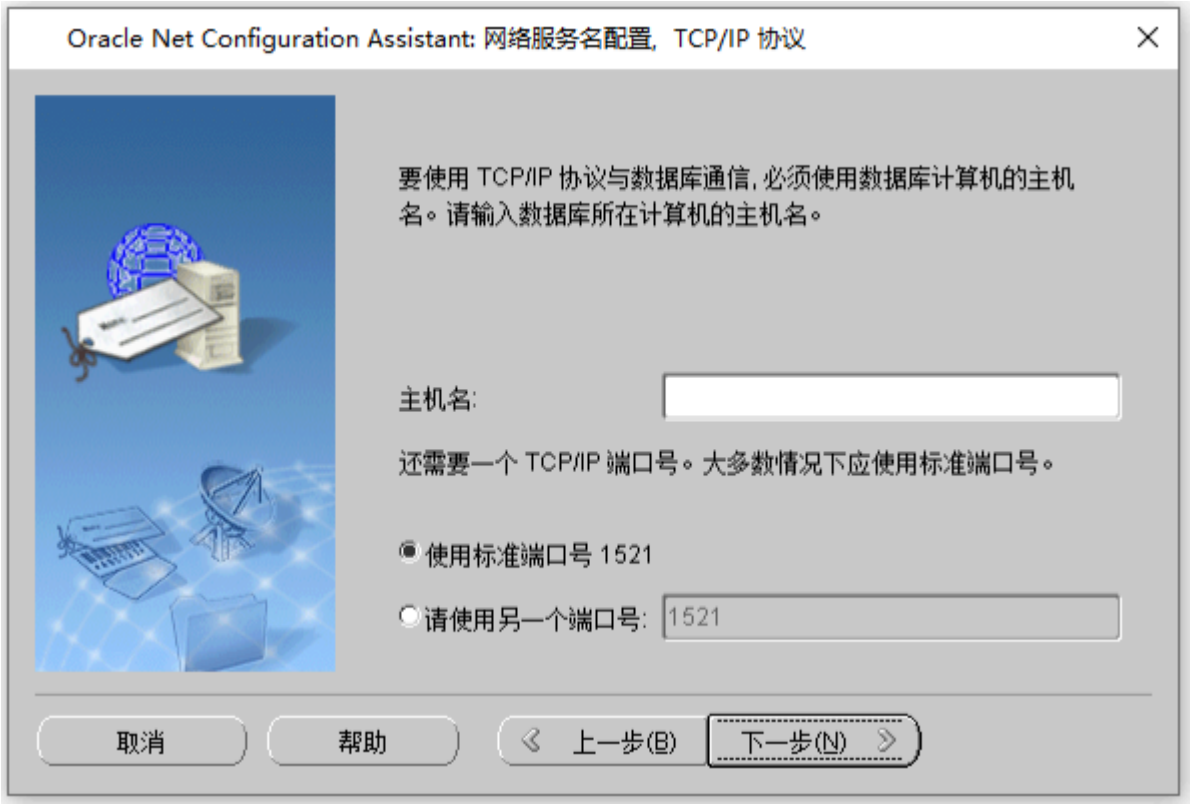
选择"本地网络服务配置"，点击下一步。

选择"添加"，点击下一步，

输入服务名，**此处服务名应配置为oracle数据库的实例名**，例如：orcl，点击下一步，

选择TCP，点击下一步，

输入主机名，**主机名应设置为oracle数据库的ip地址**，设置端口号，默认为1521，点击下一步，



此处可以测试数据库是否连通，选择"是，进行测试"，点击下一步，

点击"更改登录"，输入用户名，密码，点击"确定"，点击下一步。

网络服务名，由用户自定义，点击下一步。

选择"否",点击下一步，

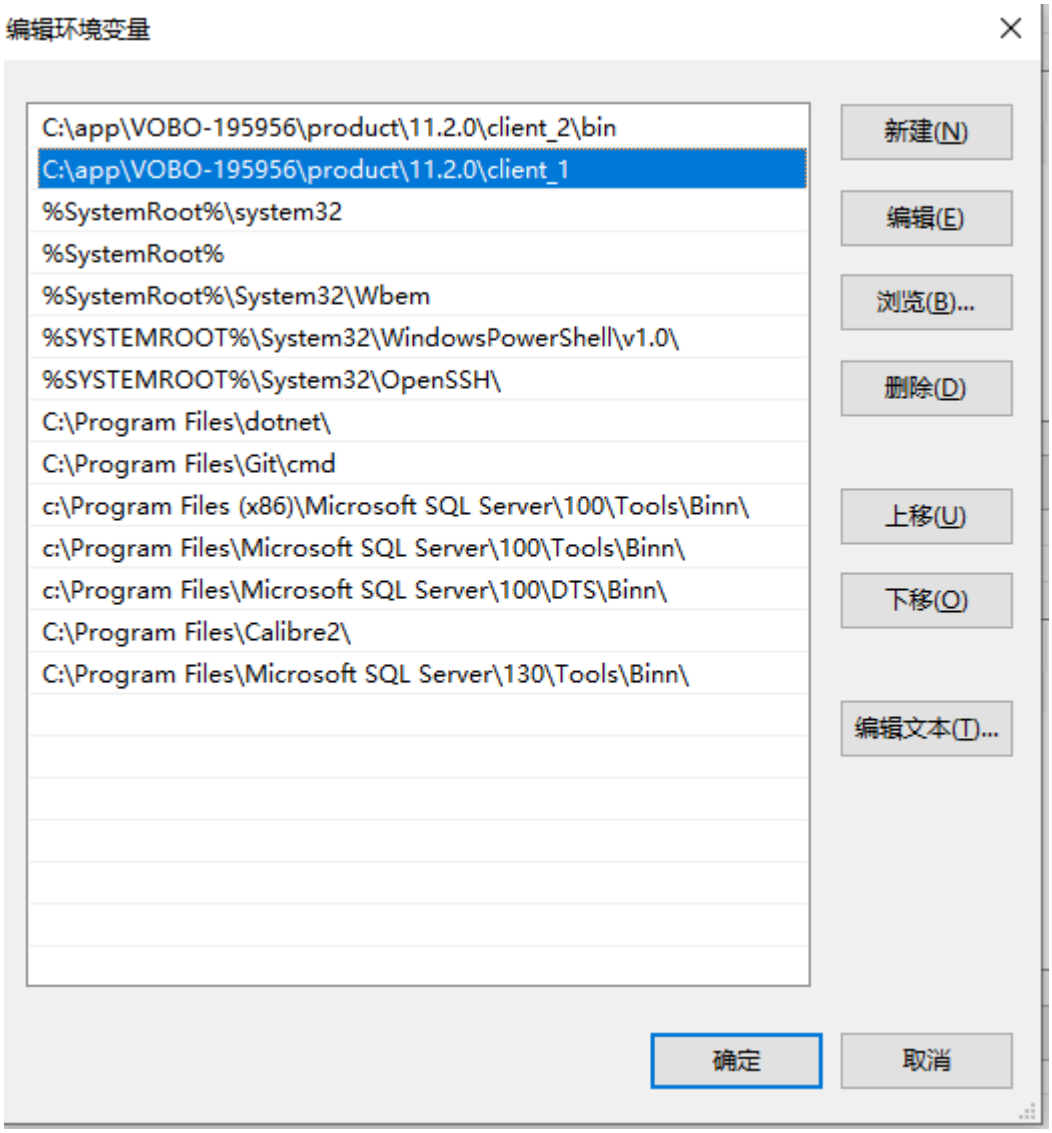
点击下一步，配置完成。

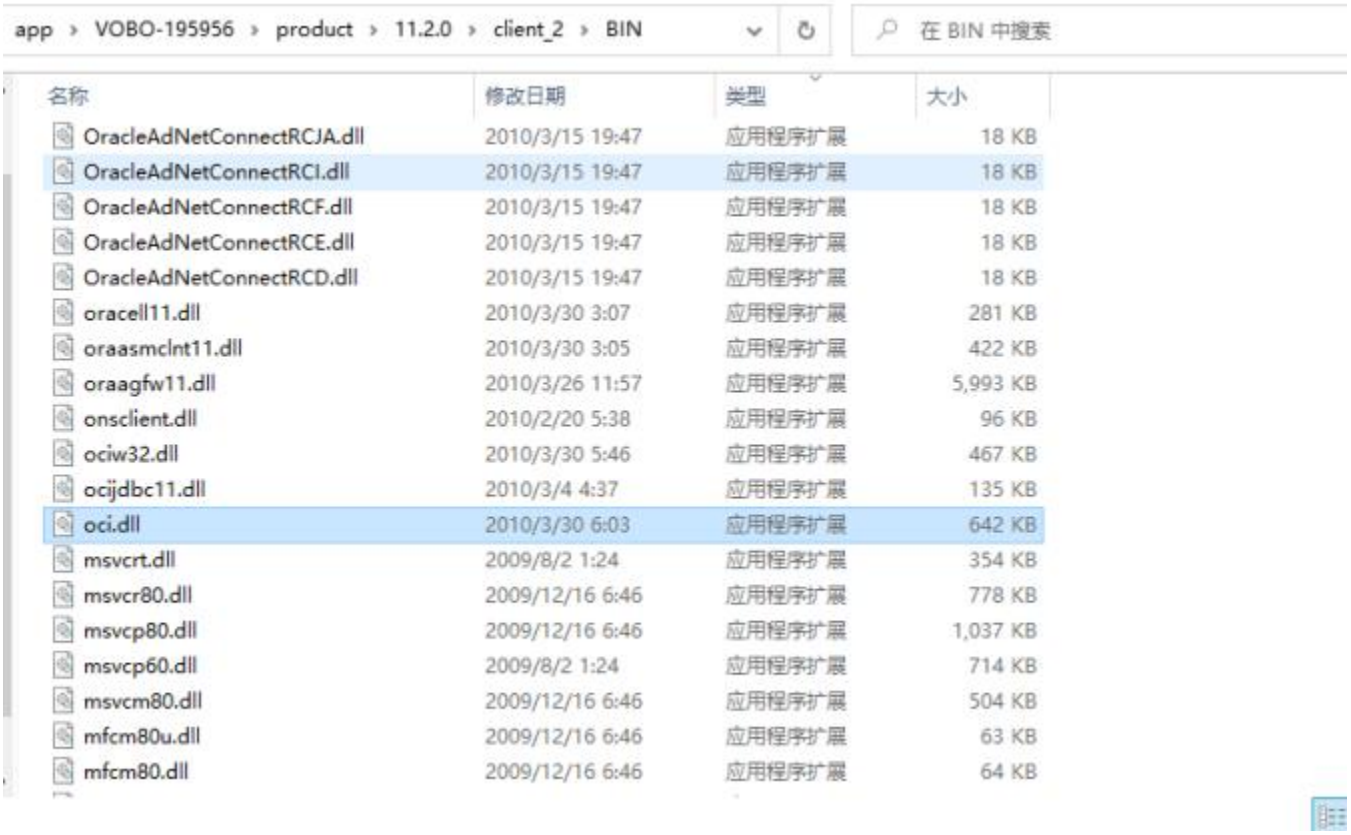
上述操作本质是修改tnsnames.ora文件，文件路径一般为为
X:\..\..\product\11.2.0\client_X\network\admin，

tnsnames.ora

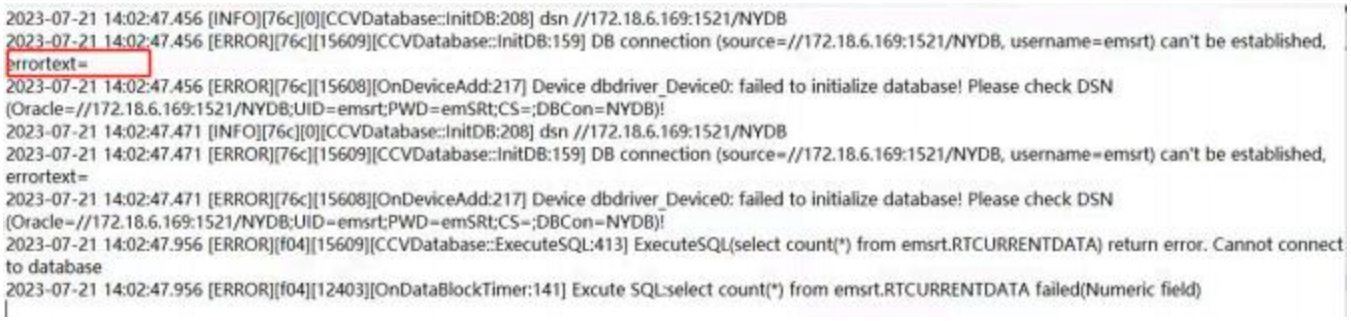
```
# tnsnames.ora Network Configuration File: C:\app\VOBO-195956\product\11.2.0\client_2\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

LOCALTEST =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 10.25.10.130)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
    )
  )
```

客户端配置完成后查看下系统环境变量，查看是否有oracle客户端的环境变量目录是否设置正确，一般来说应设置到bin目录下。



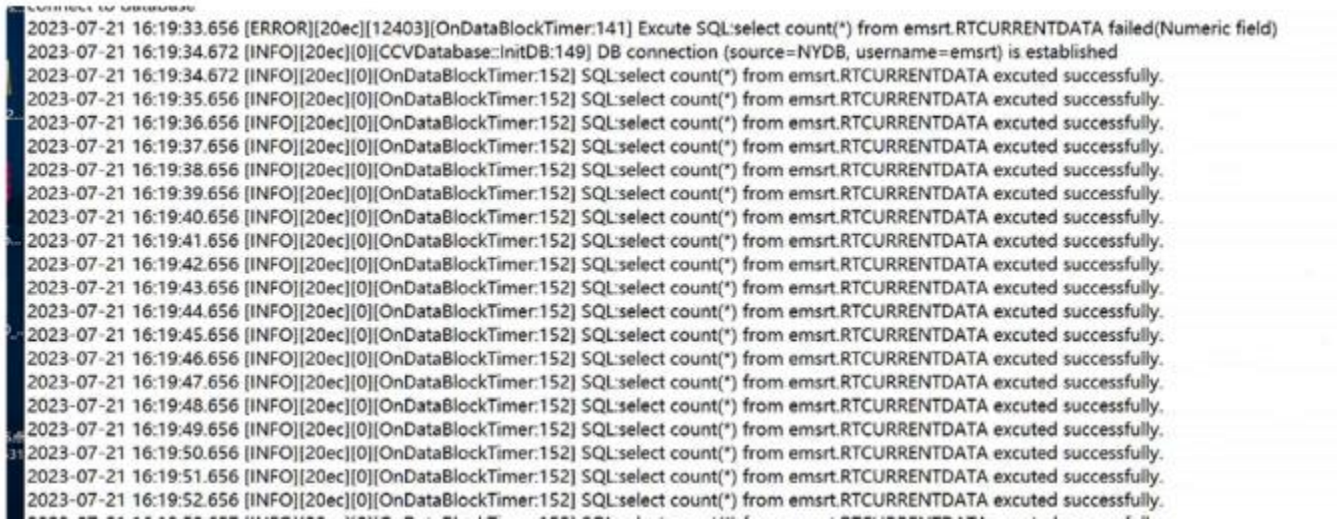


dbdrv访问oracle数据库需要加载oci.dll等oracle相关的动态库，若没有配置环境变量或环境变量配置有误，驱动将无法连接oracle，此时驱动日志将会报错，如下：



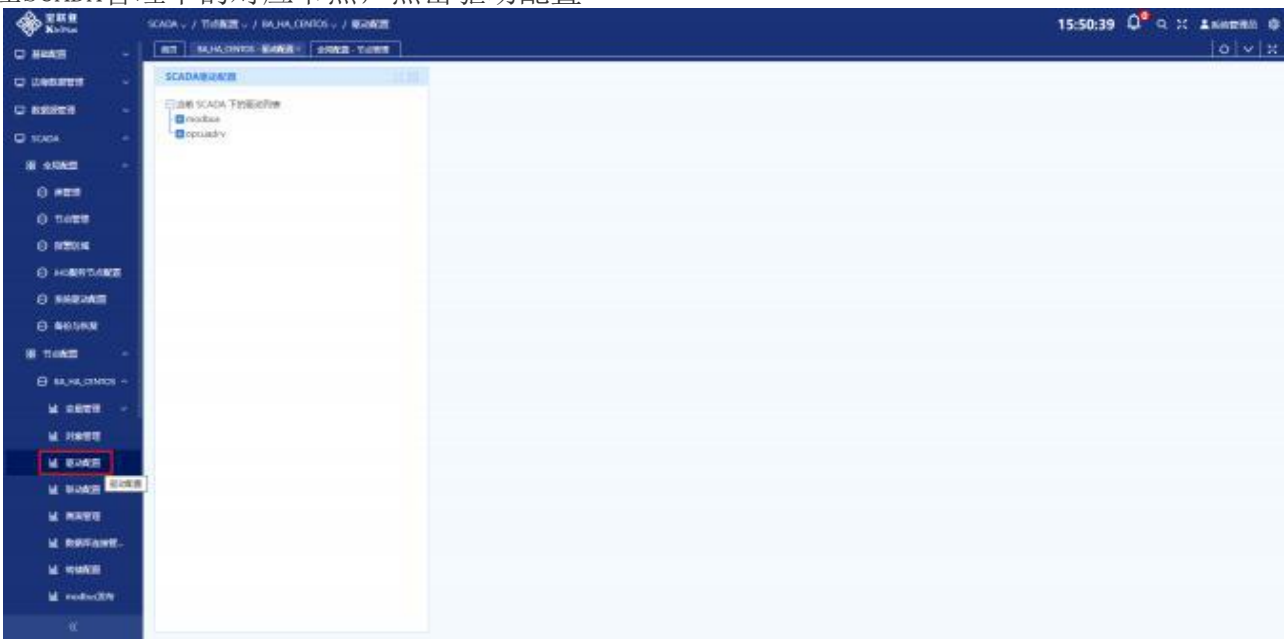
如图可见"errortext="后的内容为空，原因就是没有正确加载动态库，驱动捕获异常的信息显示不全。
若正确安装了oracle客户端且环境变量配置正确，则"errortext"能够正常输出异常信息，例如：

上图错误信息含义为用户被锁，需要进行解锁，解锁后连接成功：



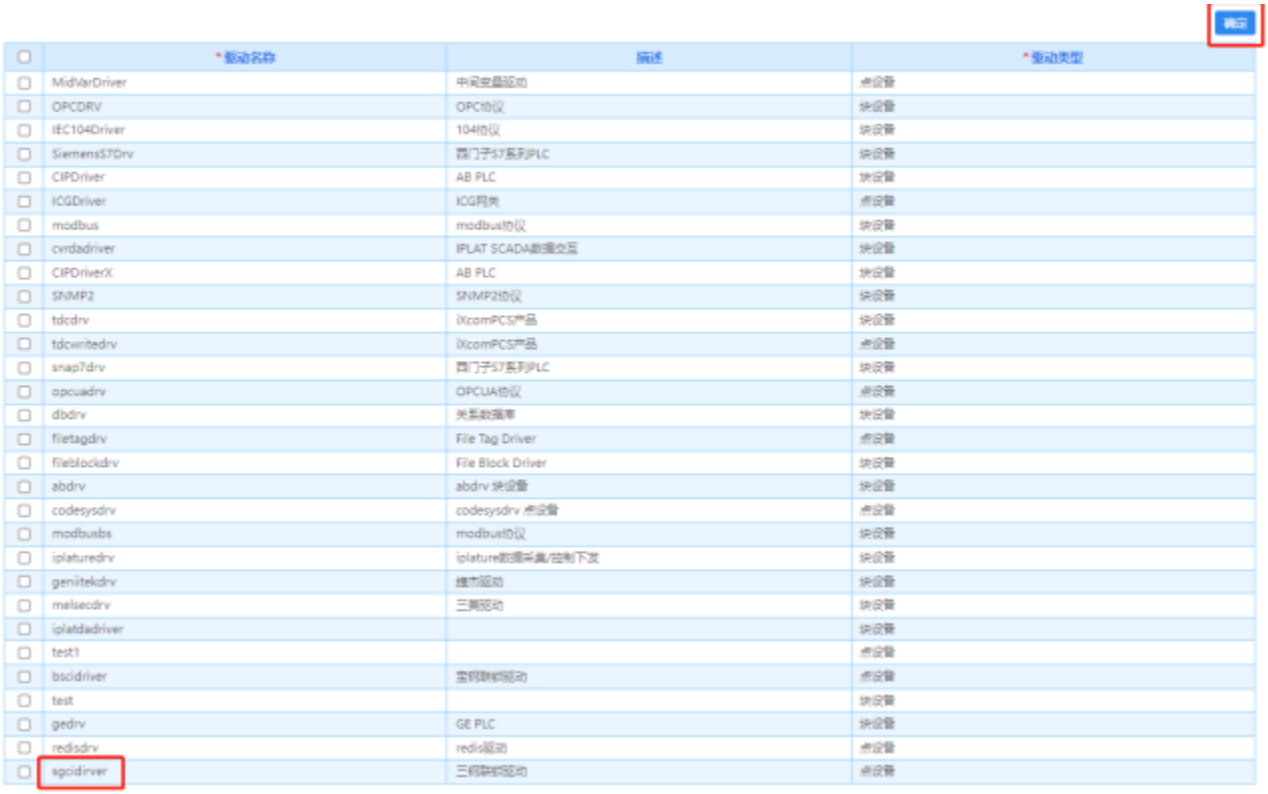
添加sgcidriver驱动

1. 在SCADA管理下的对应节点，点击驱动配置



2. 在驱动配置页面中，点击添加驱动

3. 勾选sgcidriver驱动，再点击确定



	*驱动名称	描述	*驱动类型
	MidVarDriver	中间变量驱动	点设备
	OPCDRV	OPC协议	块设备
	IEC104Driver	104协议	块设备
	Siemens57Drv	西门子S7系列PLC	块设备
	CIPDriver	AB PLC	块设备
	ICGDriver	ICG网关	点设备
	modbus	modbust协议	块设备
	cvrdadriver	IPLAT SCADA数据交互	块设备
	CIPDriverX	AB PLC	块设备
	SNMP2	SNMP2协议	块设备
	tdcdrv	iXcomPCS产品	块设备
	tdcwritedrv	iXcomPCS产品	点设备
	snap7drv	西门子S7系列PLC	块设备
	opcuadrv	OPCUA协议	点设备
	dbdrv	关系数据库	块设备
	filetagdrv	File Tag Driver	点设备
	fileblockdrv	File Block Driver	块设备
	abdrv	abdrv 块设备	块设备
	codesysdrv	codesysdrv 点设备	点设备
	modbusbs	modbust协议	块设备
	iplaturedrv	iplature数据采集/控制下发	块设备
	geniitekdrv	维控驱动	块设备
	melsecdrv	三菱驱动	块设备
	iplatdadriver		块设备
	test1		点设备
	bscidriver	宝钢联机驱动	点设备
	test		块设备
	gedrv	GE PLC	块设备
	redisdrv	redis驱动	点设备
	sgcidirver	三钢联机驱动	点设备

添加sgcidriver设备

1. 右键点击添加设备

2. 右键sgcidriver，选择添加设备，如下图

在右侧设备信息中填写设备相关参数：

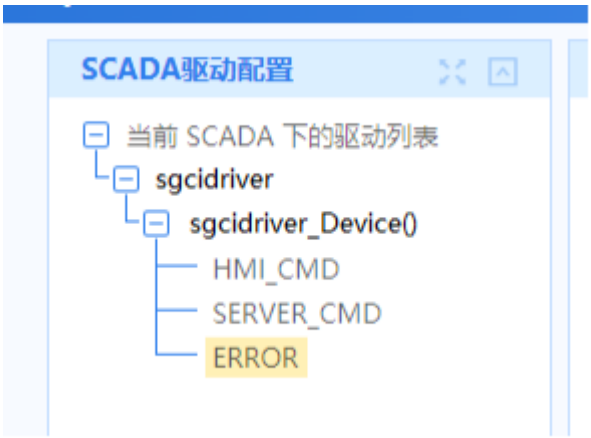
❖以TCP方式进行采集，按照以下方式配置

参数说明：

- **设备名称**：不能为空，不能以数字开头；
- **描述**：自定义的设备描述信息；
- **轮询周期**：读取对应值得周期时间；
- **响应超时**：连接设备和发送数据的响应超时时间；
- **IP地址（主）**：欲连接主设备的IP；
- **端口号（主）**：欲连接主设备使用的端口号，添加设备时默认填入10000；
- **IP地址（备）**：欲连接备设备的IP；
- **端口号（备）**：欲连接备设备使用的端口号，添加设备时默认填入10000；
- **任务号**：同一任务号下的设备消息放在同一个进程中；
- **支持多连接**：该设备是否支持多连接，1表示支持多连接；
- **扩展参数**：用户自定义；

配置数据块

添加设备后会自动生成3个数据块，HMI_CMD、SERVER_CMD、ERROR


SCADA驱动配置
当前 SCADA 下的驱动列表
sgcidriver
sgcidriver_Device()
HMI_CMD
SERVER_CMD
ERROR

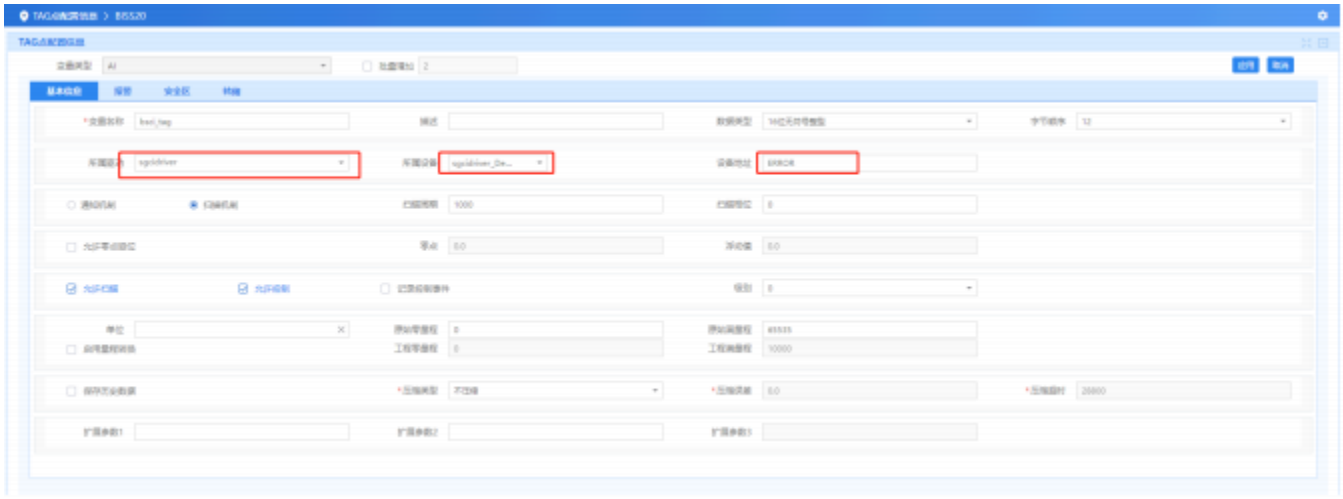
数据项信息
*地址
HMI_CMD
*元素个数
256
元素大小(字节)
1
描述
描述
可选参数1
可选参数1
可选参数2
可选参数2
可选参数3
可选参数3
保存

数据项信息
*地址
SERVER_CMD
*元素个数
256
元素大小(字节)
1
描述
描述
可选参数1
可选参数1
可选参数2
可选参数2
可选参数3
可选参数3
保存

也可以选择自行添加、删除、修改数据块。

添加数据块：选中已经配置的设备，右键设备，选择"添加数据块"，弹出数据块信息配置界面，如下图：

- **地址**：设备地址。
- **元素个数**：该数据块的长度,即，用户要在数据块中保存的查询记录的最大条数。本驱动默认填入256，可以根据需求修改。
- **元素大小[字节]**：用户查询的每条记录的元素大小。本驱动默认填入1，可以根据需求修改。
- **描述**：自定义的描述信息。
- **扩展参数**：用户自定义。



添加变量步骤与一般设备相同，这里主要描述变量关键信息:

所属驱动:选择sgcidriver;

所属设备:选择sgcidriver对应设备;

数据类型:理论上可以选择任意数据类型，驱动会将实际接收数据转成配置的类型，但还是建议根据报文里的数据类型来配置;

设备地址:

- **DI变量/DO变量的配置:**

设备地址格式为：数据块名:b偏移量。（b代表以位为单位进行偏移）

例如：Datablock1:b61，表示读取数据块Datablock1中的第62位。

- **AI变量/AO变量配置:**

设备地址格式为：数据块名:B（W）偏移量。（B代表以字节为单位进行偏移,W表示以字为单位进行偏移）

例如：Datablock0:B23，表示从数据块Datablock0中的第24字节开始读取，读取的长度根据AI变量的数据类型自动选取，8位整形读取1个字节，16位整形读取2个字节，32位整形和单精度浮点型读取4个字节，双精度浮点型读取8个字节。

Datablock0:W23，表示从数据块Datablock0中的第24字开始读取，读取的长度根据AI变量的数据类型自动选取，由于W单位是字，一个字是2个字节，所以配置的数据类型不能是8位的整形，从16位整形开始读取1个字，32位整形和单精度浮点型读取2个字，双精度浮点型读取4个字。

- **TEXT、BLOB:**

设备地址格式为：数据块名:偏移字节#字节长

度例：DataBlock0:2#50

表示从数据块DataBlock0的第3字节开始读取50字节的长度。

注：变量配点的时，也支持W，以16位为一个字，也支持按字去采集，一个字是两个字节。两个字节一个字，在采集string或者blob的时候会用到。

读取数据库连接信息

如果三钢联锁驱动要连接数据库，写入发生错误时的信息、接受报文时的信息，需要读取数据库连接信息文件wrh.ini

文件放置的路径为iplatda\executable\config\wrh.ini

```
[DB]
IP=10.25.79.115
UserName=sa
DBName=sg
Password=Baosight!

[DBLog]
IP=10.25.79.115
UserName=sa
DBName=sgLog
Password=Baosight!

[BaseInfo]
SCADA=SRL
SERVER=TS

[FlatShunting]
port = 6005
time_out = 3
version = 1000

[BJDCA]
fd_deployment=T01
fd_car_no=T02
motor_state=D02
fd_car_lsattr=D07

[DTSGA]
fd_tpc_no=A02
fd_tpc_type=A01
fd_tpc_status=A03
fd_tpc_gw=A04
fd_tpc_tare=A05
fd_tpc_nw=A06

[DTSCA]
fd_wagon_no=A01
fd_wagon_status=D01
fd_wagon_use_status=A02
```

```
[DZBCA]
fd_wagon_no=T03
fd_type_name=A02
fd_wagon_use_status=D01
fd_customize_state=D02
fd_wagon_status=D03
fd_goods_code=T02
fd_goods_name=T01
fd_lsattr=D04

[DLCPA]
wagon_num=T03
type_name=A02
wagon_use_status=D01
customize_state=D02
wagon_status=D03
goods_code=T02
goods_name=T01
lsattr=D04
```

《SiemensS7驱动》该驱动支持与西门子的S7系列PLC之间的通讯，通讯方式通过以太网进行通讯。

支持的硬件型号有：

- SIMATIC S7-200；
- SIMATIC S7-300；
- SIMATIC S7-400(H)；（冗余的主备通讯由驱动程序根据PLC状态信息自动判断并切换连接。）
- SIMATIC S7-1200；
- SIMATIC S7-1500；
- S7-200 SMART

该驱动无需SIMATIC NET软件及CP卡。

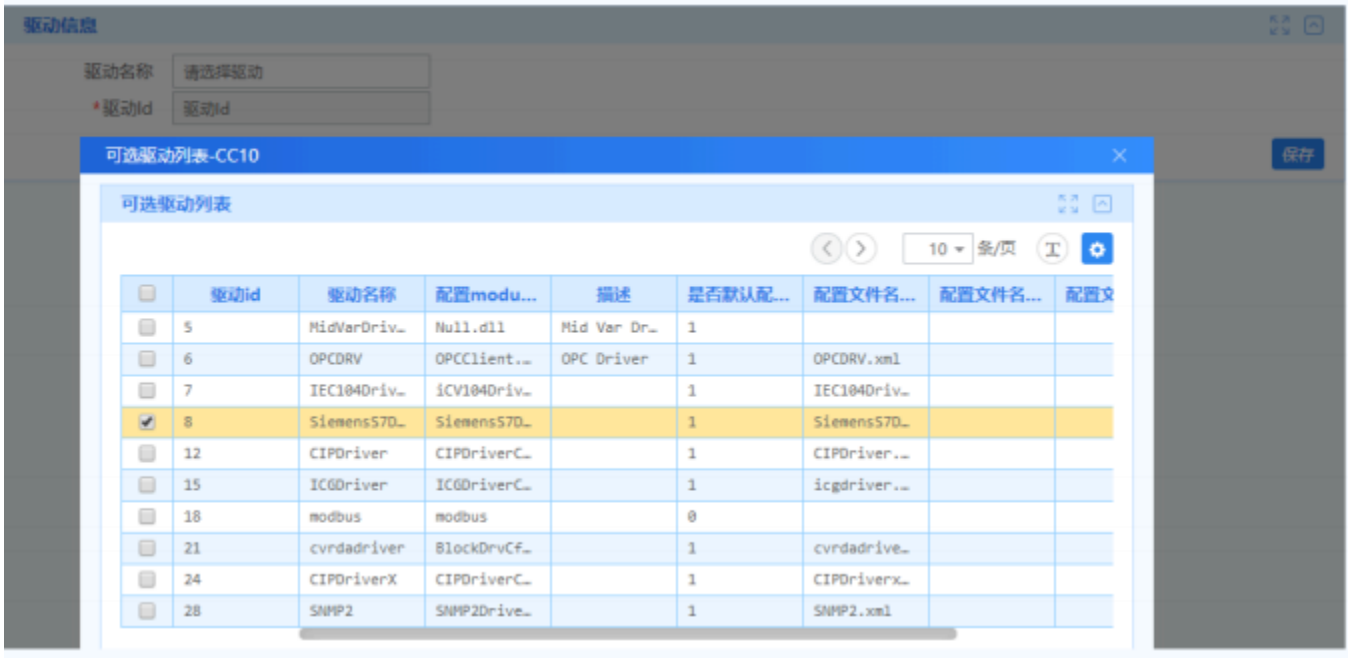
主要包含以下步骤：

添加SiemensS7Drv驱动

首先，需要配置好相应的SCADA节点。点击边缘管理->节点配置->某节点->驱动配置，如下图：

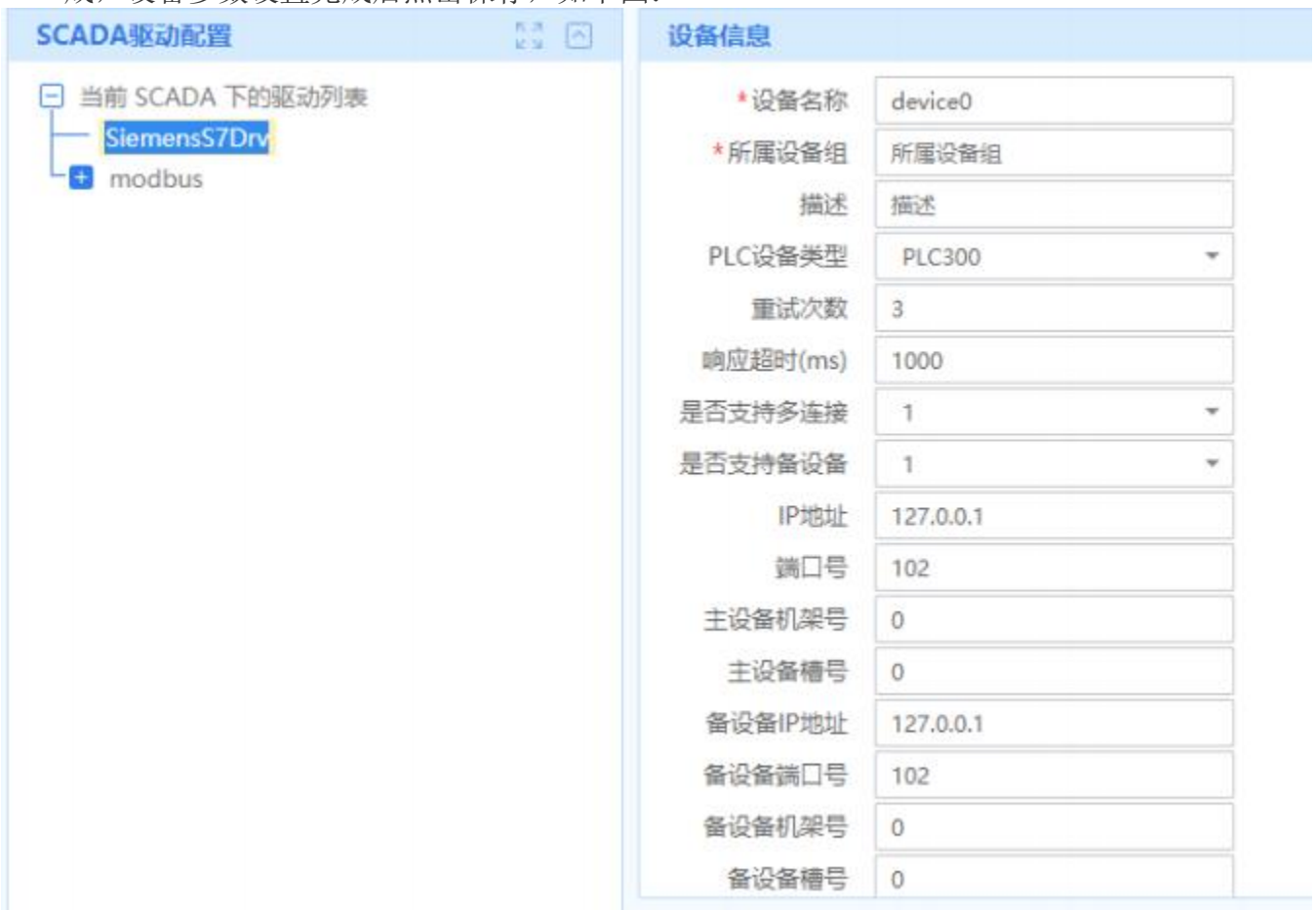
•新增驱动

进入驱动配置页面，右键点击"添加驱动"的按钮，弹出"可选驱动列表"的选择对话框，如下图。选中"SiemensS7Drv"，关闭可选驱动列表页面，点击"保存"。



·添加SiemensS7Drv驱动设备

1. SiemensS7Drv驱动配置界面，列表的右侧显示新添加设备的配置界面，设备的名称自动生成，设备参数设置完成后点击保存，如下图：



如果现场设备存在冗余，需要在"备设备"那几列中填入备机的ip地址、端口号等；除此之外，还需要配置各项参数，驱动配置工具中所列的默认配置参数为推荐的配置值，并非所有的设备均合适，如果通讯出现问题，需要调整配置参数，请咨询设备厂商。

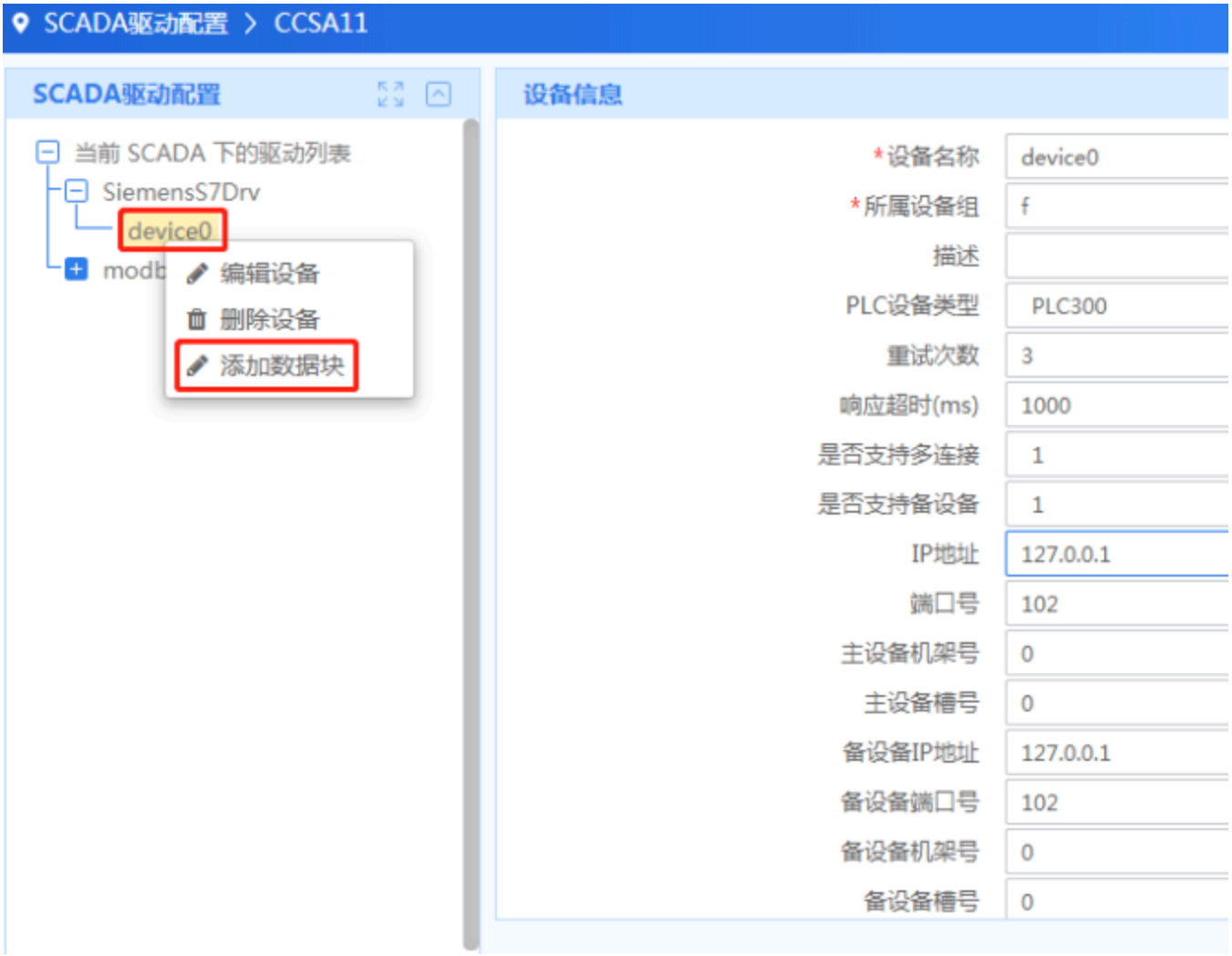
参数配置的意义如下：

- 设备名称：自定义设备名称。
- 所属设备组：当一个工程中，需要连接不同工艺组的众多PLC设备时，可以用"设备组"来进行
分组管理。如果所连PLC设备不多，也可归为一个设备组。
- 描述：添加对设备的描述信息。
- PLC设备类型：请填选实际的西门子PLC的设备类型。(除S7-200 SMART请选S7-300)。
- 重试次数：连接或发送失败时的重试次数。
- 响应超时：连接和发送时的响应超时时间。
- 是否支持多连接：默认选1，支持；选0为不支持。
- 是否支持备设备：默认选1，支持主备冗余；选0为不支持。
- IP地址/备设备IP地址：设备地址是PLC的IP地址：即为通过Step7编程软件为通讯模块(CP443-1)
设定的IP地址。
- 端口号/备设备端口号：设备连接的端口号。默认为102。
- 机架号/备设备机架号：设备插入的机架的机架号。
- 槽号/备设备槽号：设备CPU插入机架中的槽号。

SiemensS7Drv数据块配置

1. 设备参数设置完成点击保存后，驱动列表中自动生成设备，右键设备名，点击添加数据块，

如下：



2. 在SiemensS7Drv驱动的数据块配置如下图：

- **数据块名称**：自定义的数据块名称。
- **数据块描述**：自定义的数据库描述信息。
- **轮询周期**：扫描数据块的周期。
- **相位**：数据块的刷新相位。
- **类型**：对应到S7PLC中的寄存器类型。
- **起始地址**：数据块的起始位置。
- **长度**：数据块的读取长度。

注意：

1. 数据块所配置长度不能大于PLC DB/I/Q/M区实际长度；
2. 本驱动一个报文长度有限制，例如PLC寄存器是DB类型，s7-400长度限制416字节,s7-300长度限制188字节。对于其他寄存器类型的数据块长度一般会小于DB类型，具体长度需要查阅西门子相关手册。因而，对于大于416/188字节的PLC DB块，需要断成多个数据块对应。

- **数据排列方式**：对应到S7寄存器中的数据排列方式。

数据排列方式(dataorder)与iCV变量配置管理中数据类型相

关:AI 8位,DI:dataorder=0

AI 16位:dataorder=2

AI 32位,单精度浮点型:dataorder=4

- **DB分区号**：寄存器类型为DB时有效，对应到S7中的DB分区号。

SiemensS7Drv变量配置

添加变量参见变量管理快速搭建手册。

注意：配置完成后需要部署到边缘节点，使得配置生效。

- **寄存器为I、Q区时对应变量的变量设置**
  - 可配置AI8位无符号整形，设备地址为："数据块名:字节偏移"，例如：datablock1:2，表示读取datablock1这个数据块中的第3个字节。
  - 可配置DI变量，设备地址为："数据块名:字节偏移:位偏移"，例如：datablock1:2:3，表示读取datablock1这个数据块中的第3个字节的第4位。
- **寄存器为M区时对应变量的变量设置**
  - 可配置AI 8位无符号整形，AI 16位无符号整形，AI 16位有符号整形，设备地址为："数据块名:字节偏移"，例如：datablock1:2，表示读取datablock1这个数据块中的第3个字节（若为16位整形，读取第3、4两个字节）。
  - 可配置DI变量，设备地址为："数据块名:字节偏移:位偏移"，例如：datablock1:2:3，表示读取datablock1这个数据块中的第3个字节的第4位。
- **寄存器为DB区时对应变量的变量设置**
  - 可配置模拟量（AI/AO）所有类型（除双精度浮点型）变量，设备地址为："数据块名:字节偏移"，例如：datablock1:2，表示读取datablock1这个数据块中的第3个字节（若为16位数据类型，读取第3、4两个字节，若为32位数据类型，读取3、4、5、6四个字节）。
  - 可配置数字量（DI/DO）变量，设备地址为："数据块名:字节偏移:位偏移"，例如：datablock1:2:3，表示读取datablock1这个数据块中的第3个字节的第4位。
  - 可配置TXT变量，设备地址为："数据块名:偏移字节#字节长度"，例如：

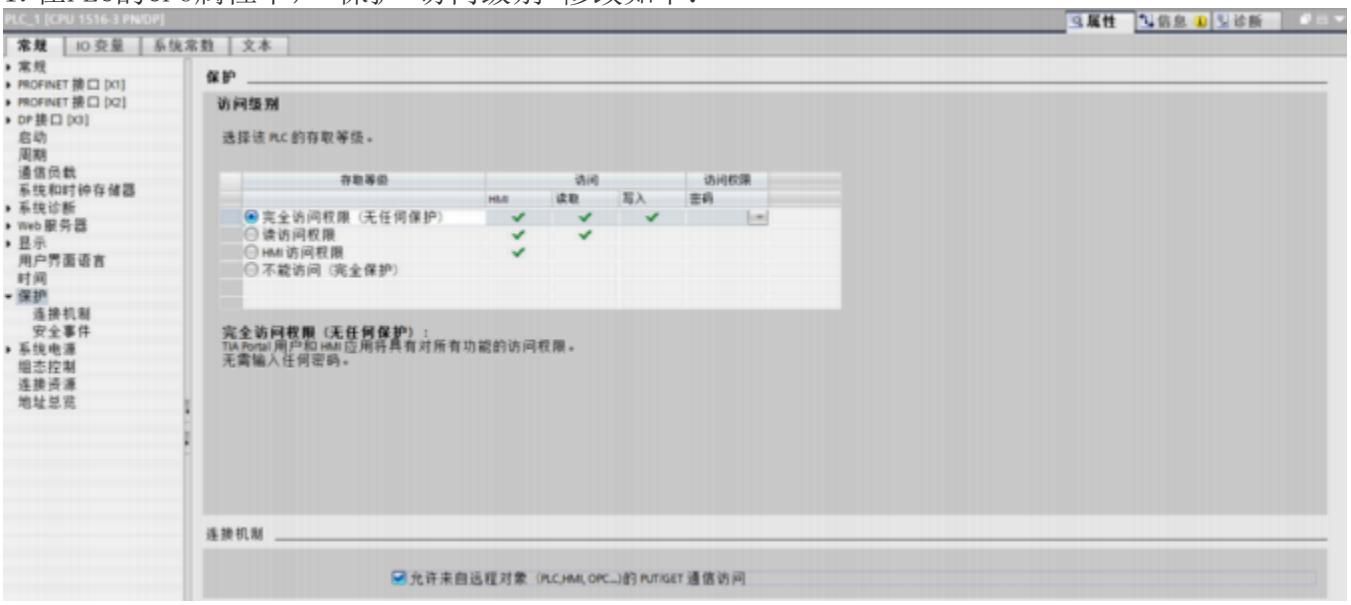
datablock1:2#50，表示读取datablock1这个数据块中第3个字节开始读取50个字节的长度。

- **寄存器为PI、T、C区时对应变量的变量设置**
  - 暂时不支持PI、T、C类型。

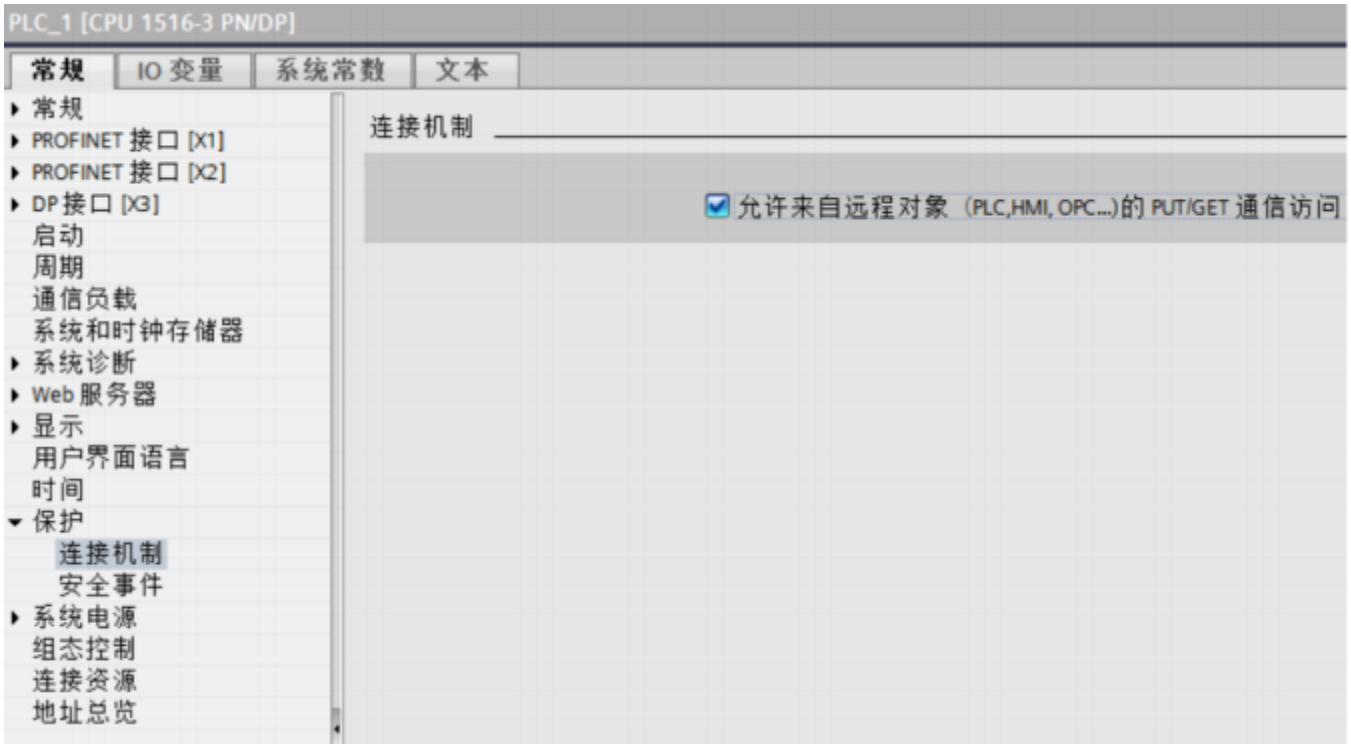
PLC1200、PLC1500侧配置修改

为了能够远程访问PLC，需要在PLC侧进行以下配置修改。

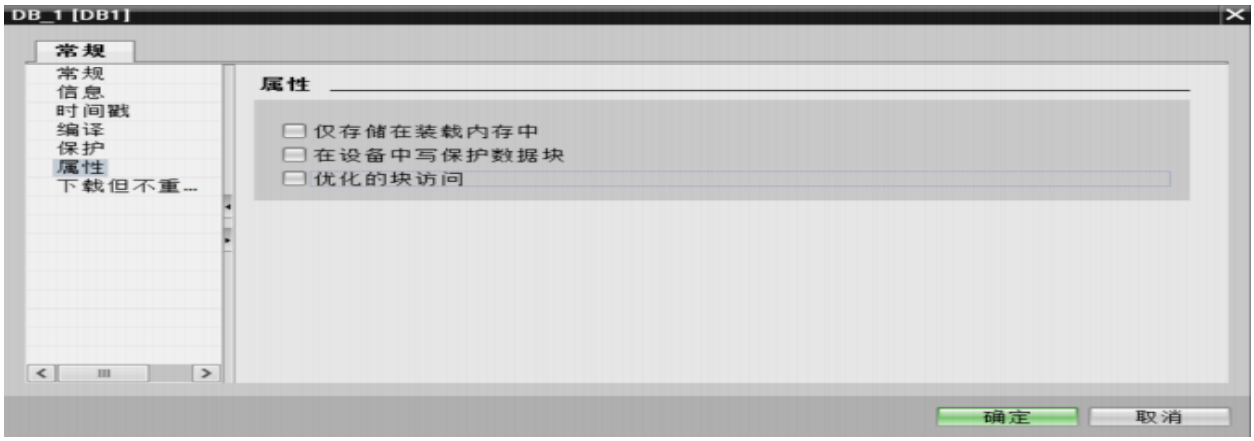
1.在PLC的CPU属性中，"保护-访问级别"修改如下：



2.在PLC的CPU属性中，"保护-连接机制"修改如下：



3.需要被icv读写的所有DB块的属性都需要将"优化的块访问"的勾去掉。（去掉这个选项，才可以使用绝对地址访问块。）



•设备配置

当前版本，请尝试选择"PLC300"。

•数据块配置

本驱动单数据块对S7-200 SMART读写支持最大长度为220个字节，即需要读取长度大于220个字节

时，请创建多个数据块用于读取。

S7-200 SMART VW寄存器类型请在数据块寄存器类型配置中选择"DB"。驱动Snap7Driver对数据块

最大长度支持65535字节。

•变量地址格式

本驱动为非免赔块驱动，因而都需要配置数据块。变量地址格式即为数据块寄存器地址的访问格

式，写法同其他PLC。

SiemensS7Drv.xml

```
<?xml version="1.0"encoding="GB2312"?>
<DriverCfg name="SiemensS7Drv"devGrpCount="1">
<DevGrpCfg name="DeviceGroup0"devCount="1">
<Description></Description>
<DeviceCfg
name="S7400"dataBlockCount="1"type="400"shelf="0"slot="0"ip="127.0.0.1"po
<Description></Description>
<DataBlockCfg name="DataBlock0"devAddress=""startaddr="0"length="100"registerType=
<Description></Description>
</DataBlockCfg>
</DeviceCfg>
</DevGrpCfg>
</DriverCfg>
```

添加omron欧姆龙驱动

1. 进入系统驱动配置界面，按照红框中的设置添加欧姆龙驱动

配置omron欧姆龙驱动设备

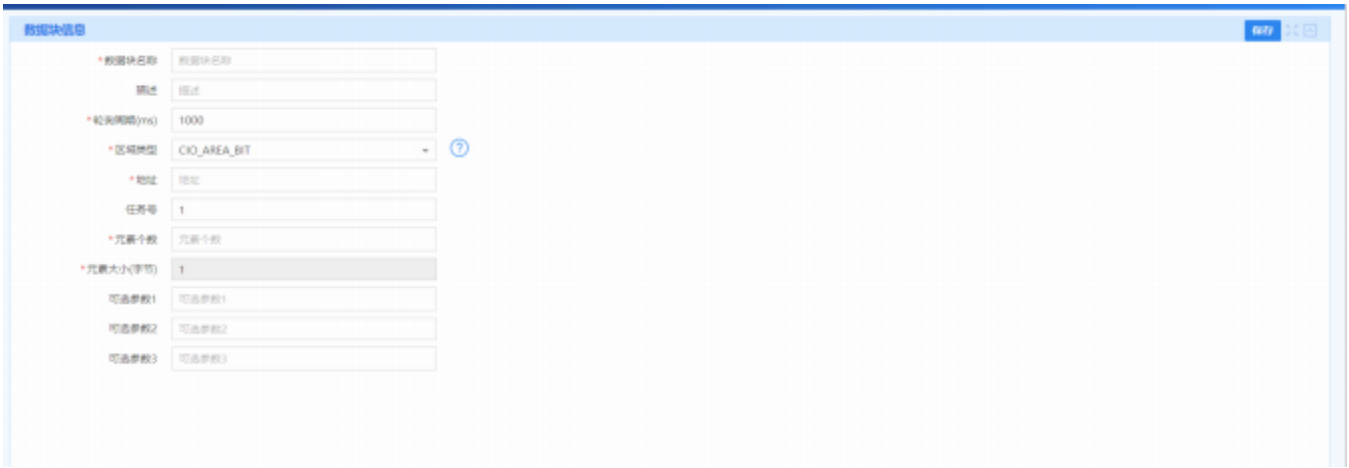
右击Omrondrv选择添加设备

参数说明：

- 设备名称：不能为空，不能以数字开头；
- 描述：自定义的设备描述信息；
- 任务号：同一任务号下的设备消息放在同一个进程中；
- IP地址（主）：欲连接主设备的IP；
- 端口号（主）：欲连接主设备使用的端口号；
- 重试次数（主）：欲连接主设备重试次数，默认值为3；
- IP地址（备）：欲连接备设备的IP；
- 端口号（备）：欲连接备设备使用的端口号；
- 重试次数（备）：欲连接主设备重试次数；
- 响应超时：连接设备和发送数据的响应超时时间；
- 支持多连接：该设备是否支持多连接，1表示支持多连接；
- 扩展参数：用户自定义；

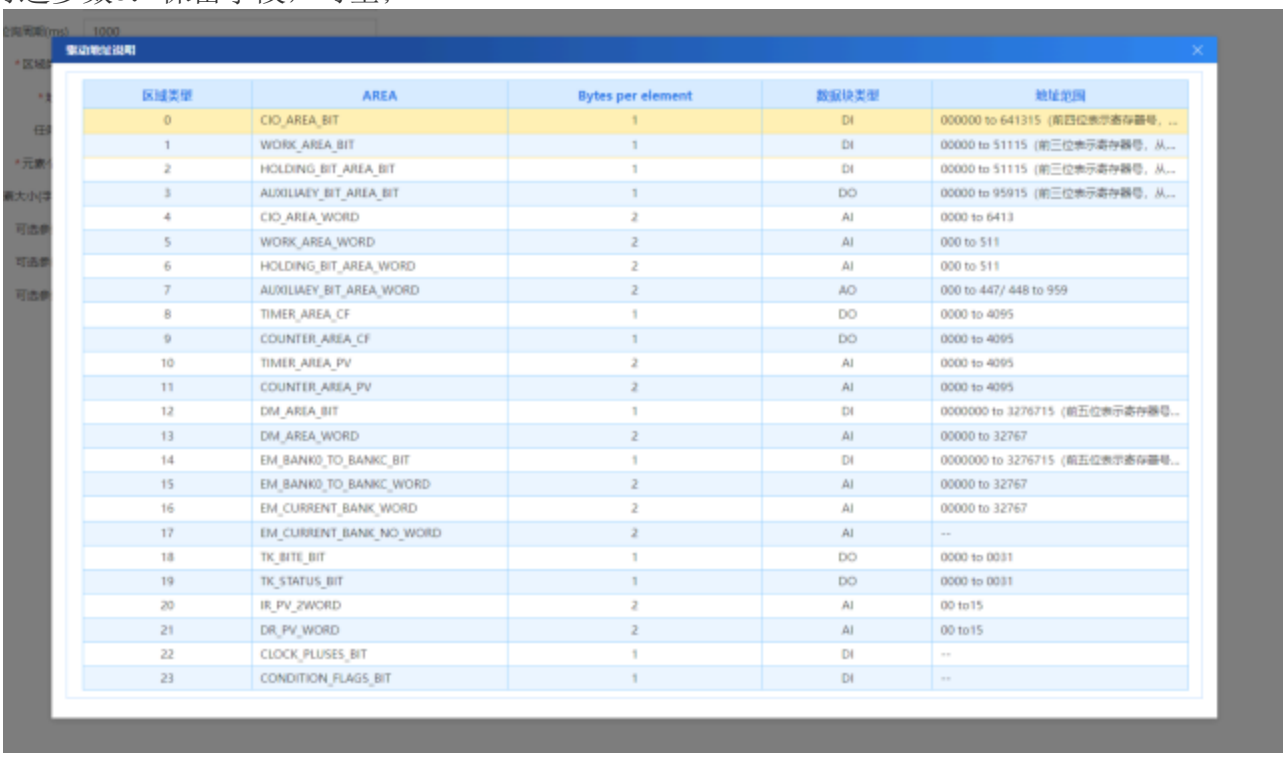
配置omron欧姆龙驱动数据块

右击需要添加数据块的设备



参数说明：

- 数据块名称：自行定义，不能为空，不能以数字开头；
- 描述：自定义的描述信息；
- 轮询周期[MS]：数据块的刷新周期；
- 区域类型：下拉框进行选择；点击?按钮会有释意
- 地址：根据选择的区域类型配置相对应的地址；点击?按钮会有地址范围
- 任务号：为所添加数据块分配任务号（线程号）。
- 元素个数：数据块的长度；
- 元素大小[字节]：根据区域类型的不同默认填写值；值对应为?按钮弹出窗口表格中的Bytes per element列；
- 可选参数1：保留字段，可空；
- 可选参数2：保留字段，可空；
- 可选参数3：保留字段，可空；



区域类型	AREA	Bytes per element	数据块类型	地址范围
0	CIO_AREA_BIT	1	DI	00000 to 641315 (前四位表示寄存器号，...
1	WORK_AREA_BIT	1	DI	00000 to 51115 (前三位表示寄存器号，从...
2	HOLDING_BIT_AREA_BIT	1	DI	00000 to 51115 (前三位表示寄存器号，从...
3	AUXILIAEY_BIT_AREA_BIT	1	DO	00000 to 95915 (前三位表示寄存器号，从...
4	CIO_AREA_WORD	2	AI	0000 to 6413
5	WORK_AREA_WORD	2	AI	000 to 511
6	HOLDING_BIT_AREA_WORD	2	AI	000 to 511
7	AUXILIAEY_BIT_AREA_WORD	2	AO	000 to 447/ 448 to 959
8	TIMER_AREA_CF	1	DO	0000 to 4095
9	COUNTER_AREA_CF	1	DO	0000 to 4095
10	TIMER_AREA_PV	2	AI	0000 to 4095
11	COUNTER_AREA_PV	2	AI	0000 to 4095
12	DM_AREA_BIT	1	DI	0000000 to 3276715 (前五位表示寄存器号...
13	DM_AREA_WORD	2	AI	00000 to 32767
14	EM_BANK0_TO_BANKC_BIT	1	DI	0000000 to 3276715 (前五位表示寄存器号...
15	EM_BANK0_TO_BANKC_WORD	2	AI	00000 to 32767
16	EM_CURRENT_BANK_WORD	2	AI	00000 to 32767
17	EM_CURRENT_BANK_NO_WORD	2	AI	--
18	TK_BITE_BIT	1	DO	0000 to 0031
19	TK_STATUS_BIT	1	DO	0000 to 0031
20	IR_PV_2WORD	2	AI	00 to15
21	DR_PV_WORD	2	AI	00 to15
22	CLOCK_PLUSES_BIT	1	DI	--
23	CONDITION_FLAGS_BIT	1	DI	--

【区域类型】对应的是xml配置文件的datablock元素中的type属性

【AREA】是PLC的区域名称

【Bytes per element】每个元素所使用的byte数

【地址范围】数据块DI、DO类型：配地址时填四位PLC地址，不写0~15的偏移地址。element count按需填写，每个element为1byte。element count每增16相当于PLC地址加1。例如:【地址=6400 elmentcount=31】的数据块，相当于640000640015、640100640115

数据块AI、AO类型：配地址时填四位PLC地址。element count按需填写，每个element为2byte。

最终计算出来的PLC地址，不能超过表中的最大地址。

xml示例

- 添加iplaturedrv驱动

iplaturedrv驱动支持推送iplature端tag点数据,批量下发控制tag点的值.

iplature和iplatda数据通过zmq消息中间件交互数据,共有两个通道,分别为发布订阅模式通道和请
求应答模式通道,

实时数据推送走的是发布订阅通道,全量数据和下发控制走的是请求应答模式通道

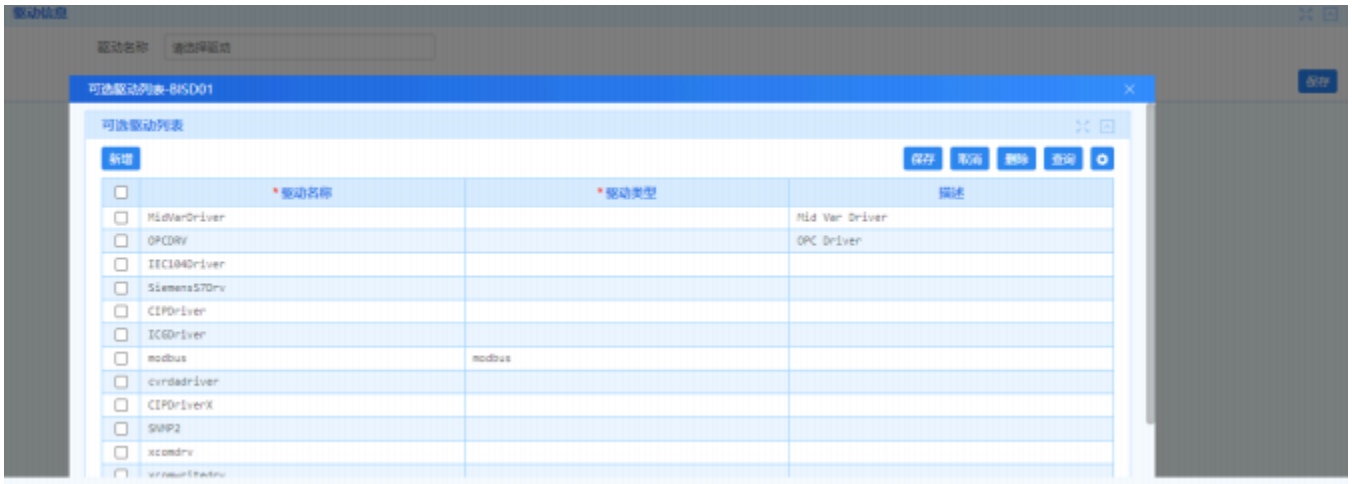
.iplaturedrv配置步骤:

1. 添加iplaturedrv驱动
2. 添加iplaturedrv设备
3. 配置变量
4. 控制下发

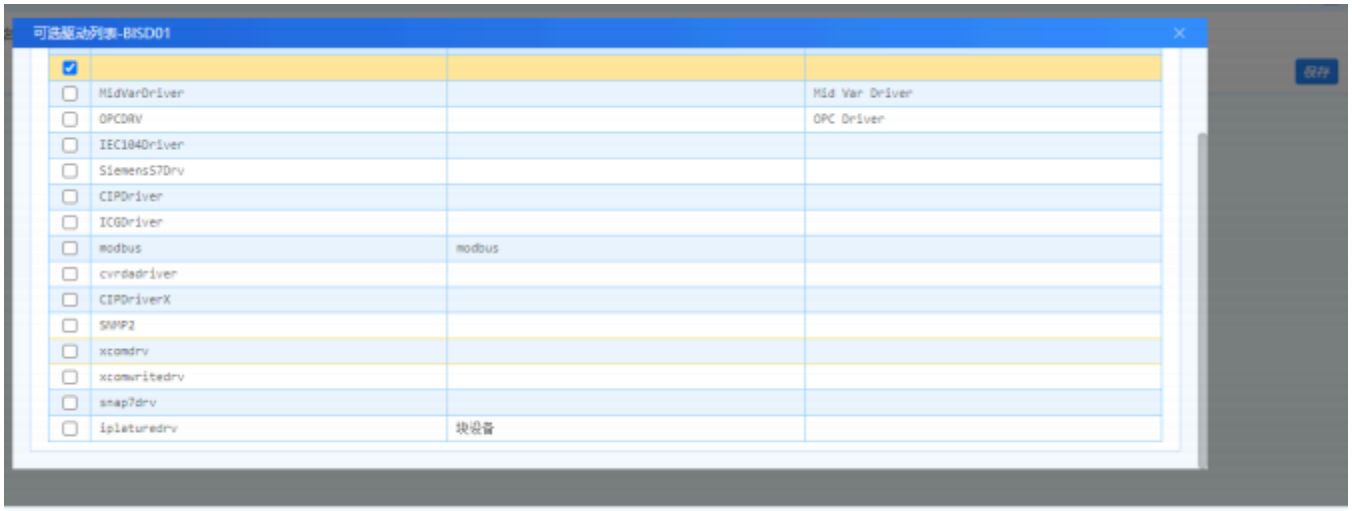
添加iplaturedrv驱动

驱动配置页面添加驱动,

点击新增,输入"iplaturedrv",若已经存在此驱动,点击选中,



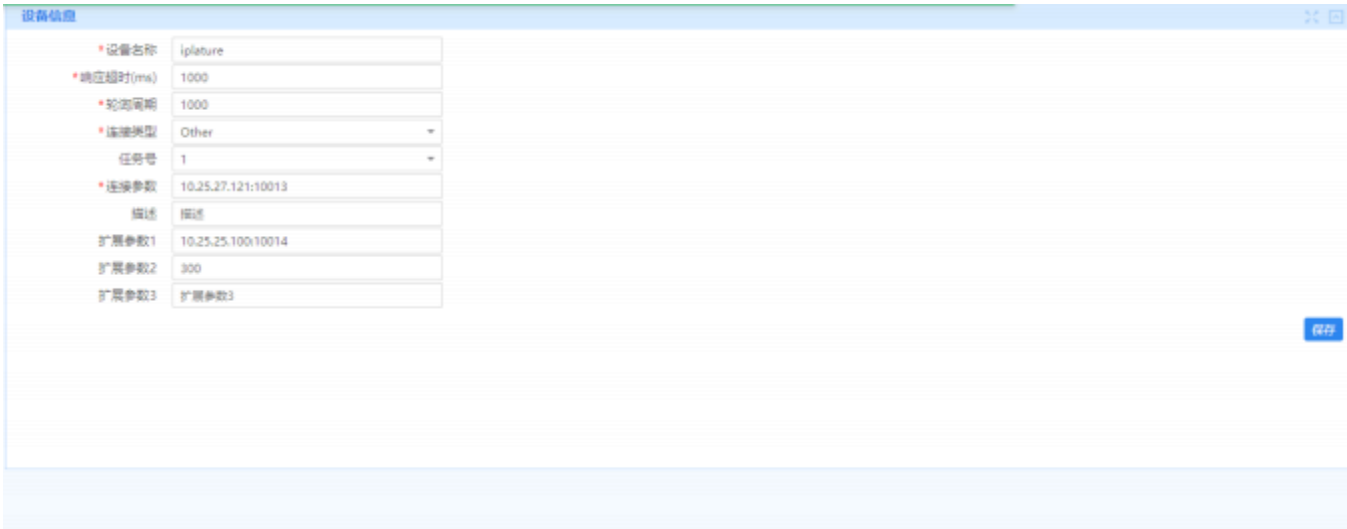
点击保存,



添加iplaturedrv设备

右键点击添加设备,

iplaturedrv为免配块驱动,只需要配置相关设备信息,如下



- 设备名称：自行定义，不能为空，不能以数字开头；
- 响应超时：连接设备和发送数据的响应超时时间，单位：毫秒；
- 连接类型：选择Other。不可选其他类型，本驱动只支持这一种连接类型；
- 连接参数：格式为ip地址:port端口号,xxx.xxx.xxx.xxx:xxx。其中ip和port为iplature端数发布订阅模式通道的ip地址,端口为iplature端数发布订阅模式通道的端口号
- 描述：自定义的设备描述信息；
- 任务号：为所添加设备分配任务号（线程号）。目前版本支持配置一个iplature设备,不支持多设备；
- 可选参数1：请求应答模式通道的ip/port端口号,格式为xxx.xxx.xxx.xxx:xxx；请求应答模式通道指iplat请求指定的tag点,iplature收到请求后返回响应tag点值的一条专有数据链路
- 可选参数2：全量数据推送的周期频率,单位为秒,缺省值为300；全量推送指iplature在指定周期(如300s)推送iplature所有tag点到iplature驱动
- 可选参数3：暂未使用；

变量配置

该驱动为免配块驱动,定义了变量地址为iplature的Tag点的点名,例如iplature端tag点为"mtag1",则iplat端配置的变量地址为mtag1.

iplaturedrv驱动控制变量值

iplature固定的控制有五种CylicPut,EventPut,MDPut,MDPutEnd,Write2Level1,参数个数也不一样，除此之前还工程部门也有自定义的其它控制方法

控制方法	参数1	参数2	参数3
CyclicPut	map<string,string>点，值		
EventPut	map<string,string>点，值	string	string
MDPut	map<string,string>点，值	map<string,string>	
MDPutEnd	map<string,string>点，值		
Write2Level1	map<string,string>点，值		
自定义	自定义	自定义	自定义

iplatscada的控制只能通过点名写值，组态页面调用的写值方法为WriteValue(TagName,Value)

为了能实现下发控制到iplature的功能，并兼容工程部门的使用习惯，iplature的控制下发方案如下

1、配置iplature控制点，对于每一种控制方法Scada中建一个字符串类型（TXT）的点，点的地址配置为iPlature的控制方法名，例如Scada中建立一个TXT类型的任意名字的点（如
TagName_CyclicPut）地址配置为CyclicPut

2、使用js开发类似于alarm，trend和sys等对象的方法，开发一个新的对象iplature，用户调用iplature.CylicPutt(map<string,string>)等方法进行控制，map的key-value是点值对,如
tag1:value1,tag2:value2，iplature.CylicPut是基于sys的
WriteValue(tagname,value)方法进行封装。

3、页面脚本调用CyclicPut()方法，传入参数tag1:value1 tag2:value2用来控制iplature中的点tag1，tag2

使用方式如下：
```
Button8087521_onClick=function(event,data,view){iplature.CylicPutt("TagName_CyclicPut","[{\"ta
```
4、驱动收到控制命令，需要控制点名为TagName_CyclicPut，值为"{"tag1":"value1","tag2":"value2"}"，驱动最终生成发送到iplature的控制字符串
```
{"msgtype":"1","transmitid":"100001","method":"CyclicPut","parameter":"[{"tag1":"value1","tag2":"v
```

5、iplature收到控制字符串，解析出控制方法名为CyclicPut,控制参数为tag:value1 tag2:

value2
进行控制，返回控制结果{"msgtype":"1","transmitid":"100001","result":"sucess"}

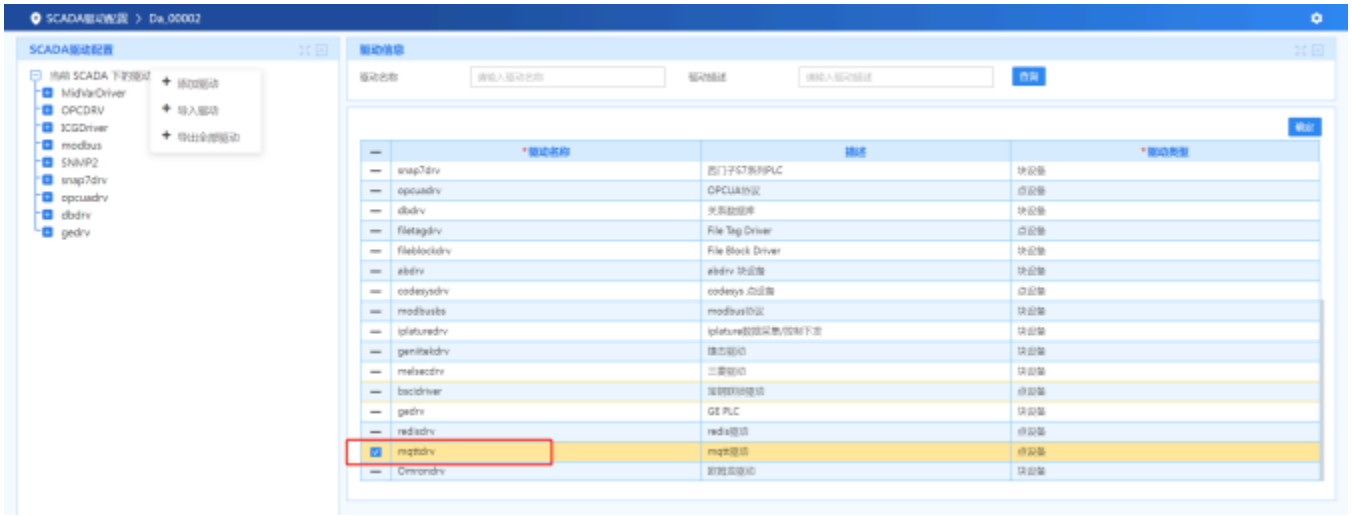
iplature和iplatda数据通过zmq消息中间件交互数据,共有两个通道,分别为发布订阅模式通道和请求应答模式通道,

实时数据推送走的是发布订阅通道,全量数据和下发控制走的是请求应答模式通道.

- 添加mqttdrv驱动
- 配置mqtt设备
- 配置变量
- mqtt驱动支持的通信标准

添加mqttdrv驱动

右键scada驱动列表中选择添加驱动，在可选驱动列表中勾选mqttdrv驱动，点击确定按钮。



配置mqtt设备

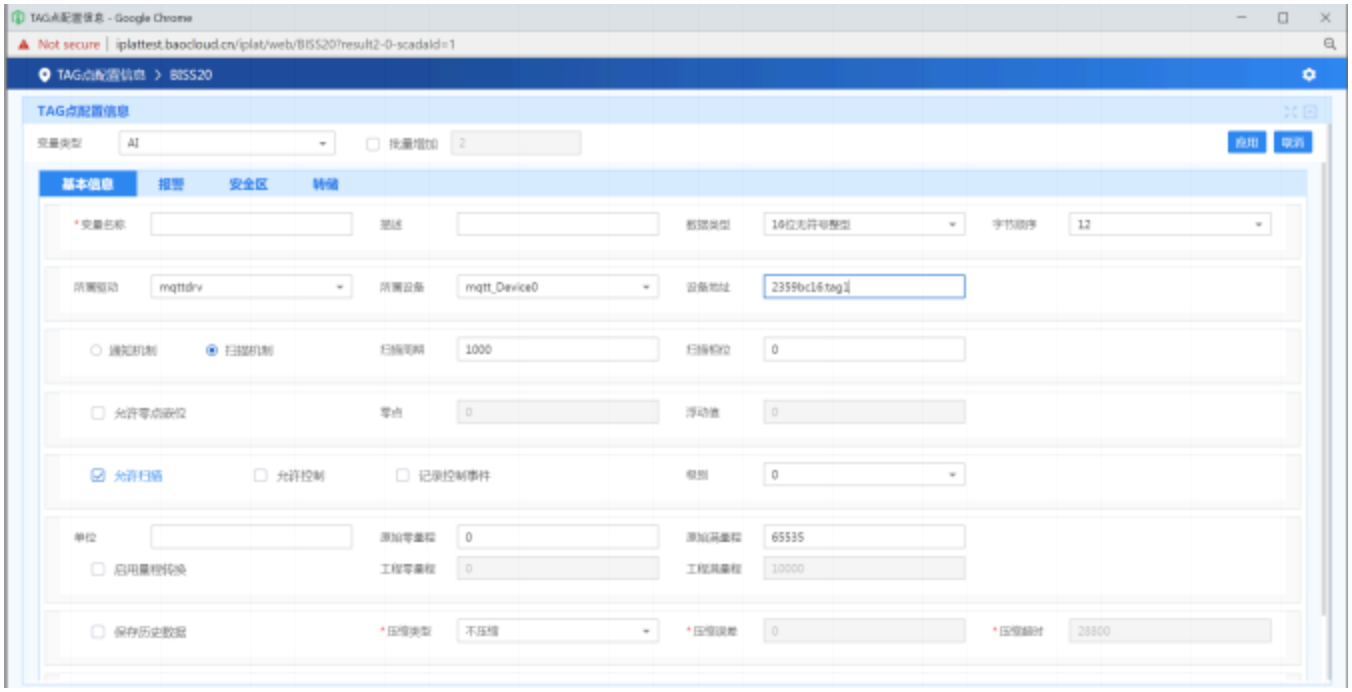
右键mqttdrv中选择添加设备

参数说明：

- 设备名称：mqtt broker的名称,不能为空，不能以数字开头
- 描述：设备描述
- IP地址：mqtt broker的IP
- 端口号：mqtt broker使用的端口号，默认为1883
- 用户名:mqtt broker设置的用户名
- 密码:mqtt broker用户的密码
- 任务号:按需求填写，同一任务号的设备会放在同一线程进行读取
- 连接超时:连接mqtt broker的超时,默认10s

配置变量

进入节点配置对应节点的设备变量管理页面,新增



参数说明:

- 所属驱动：mqttdrv
- 所属设备：mqttdrv添加的设备
- 设备地址：变量的地址,格式为device_sn:tagname,device_sn表示采集设备的序列号,tagname

表示变量名,如2359bc16:tag1,2359bc16是采集设备的序列号,tag1是变量名;

String类型变量配置格式：2359bc16:tag1#30；30是字符串长度

其他参数参见:新建变量

mqtt驱动支持的通信标准

宝信-保罗云-数据标准

- 接入设备注册

  设备使用mqtt接入平台时，第一步必须先注册。注册成功平台才会订阅此设备的所有topic，网关发布的数据才能被平台接收。网关发送注册命令的同时，会订阅此序列号的所有

  topic，接收平台下发的消息，如果收到注册失败，设备应该取消订阅此序列号的topic。平台

  启动时，会订阅topic"/xin3plat_iot/up/sys/active"，接收网关的注册请求(注意，topic最前面有/),该topic QOS=1

注册请求

（平台订阅，网关发布）：

```
 topic: /xin3plat_iot/up/sys/active
body: {
"id": 0,
"version": "1.0",
"device_sn": "12345667890123456788012345678 90ab",
"device_type": "MQTT",
"model_type": "robot",
"code": "60ceade02196fe537854f5c0f8c5bdemodemodemo"
}
```

注册请求返回
（平台发布，网关订阅）：

```
topic: /xin3plat_iot/<device_sn>/down/sys/active_ack
body: {
"id": 10000,
"status": 0
}
```

字段	类型	说明
id	int	消息发布的ID号，从0开始自增。
		每个消息发布者各自独立计数。
		设备A第一个发布的消息号为0，A后续继续发布的消息的ID则依次自增。
		平台发布的第一个消息号为0，平台后续继续发布的消息的ID则依次自增。
version	String	本协议的协议版本号，目前取值为1.0。
device_sn	String	网关序列号。接入平台校验的字段之一。
device_type	String	网关类型。固定为"MQTT"。接入平台校验的字段之一。
model_type	String	网关模型类型目前支持"robot"
code	String	激活码
status	integer	0/1：成功
		-1：网关发送的激活码在平台中找不到。

		-2：网关发送的序列号在平台中找不到。
		-3:网关已经接入，无需重复激活接入。
		-100：未知错误。

注册流程：

（a）网关订阅topic"/xin3plat_iot/<device_sn>/down/sys/active_ack"

（b）平台收到注册请求消息，取出body中的device_sn，下发**active_ack**消息给设备

（c）网关收到返回消息后，解析body中的状态，如果是1表示注册成功，如果是-3则代表已经初次接入激活过，无需重复激活接入

•网关数据推送

（平台订阅，网关发布）

网关**严格**按照以下格式推送数据，一次推送一个数据点消息或多个数据点的消息。如果平台接收的数据格式，未严格按以下格式定义则平台不接收。

QOS=0

```
topic: /xin3plat_iot/<device_sn>/up/data/tagvalue
    body:{
     "id" : 3,
     "time":"timestamp:",  //单位秒，应用层使用时自行补充成毫秒
     "data":[
     {"<tag1>"  : [<"value1">, <"quality1">]},
      {"<tag2>"  : [<"value2">, <"quality2">]},
        ......,
      {"<tagn>"  : [<"valuen">, <"qualityn">]},
    ]
}
```

timestamp	**String**	**数据时间戳，单位秒，自1970年以来的秒数。应用层使用时自行补充成毫秒**
data	二维数组	网关推送数据。
tag 1...tag n	String	网关采集点名称，要保证唯一
value 1...value n	String	数据值

quality 1...quality n	String	表明数据质量，3代表数据可信，其他表示数据可信值较低,如下质量戳表格

质量戳表格

数据质量	含义
-1	timestamp不存在的情况下,quality设置-1,代表时间戳是dm赋值的
0	BAD
1	不确定质量是GOOD还是BAD
3	GOOD

•控制命令

平台下行命令（平台发布，网关订阅）

网关注册成功后订阅控制命令topic，接收平台下发的控制命令QOS=1

```
topic: /xin3plat_iot/<device_sn>/down/sys/control
body:{
"id":10005,
"type":"tag",
"value":[
{`"<key1>":"<value1>"`},
{`"<key2>":"<value2>"`},
------
{`"<keyn>":"<valuen>"`}
]
}
```

| 字段|类型|说明
---|---|---|---
type|String|控制命令类型：
||(1)设置tag点的值，type为"tag"
||(2)设置心跳间隔，type为"heartbeat"
||**(3)设置网关软件整体重启type为"softallreboot"**
||**(4)设置网关软件模块重启type为"softmodreboot"**
||**(5)设置日志上送开启或关闭type为"sentlog"**
||**(6)设置日志等级更改type为"setloglevel"**
||**(7)要求设备全量推送一次数据type为"callall"**
||**(8)设置存储到边缘redis数据的保存时间type为"expirat
key1...keyn|String|(1)设置tag点的值，key为tag名
||(2)设置心跳间隔，key为"heartbeat"

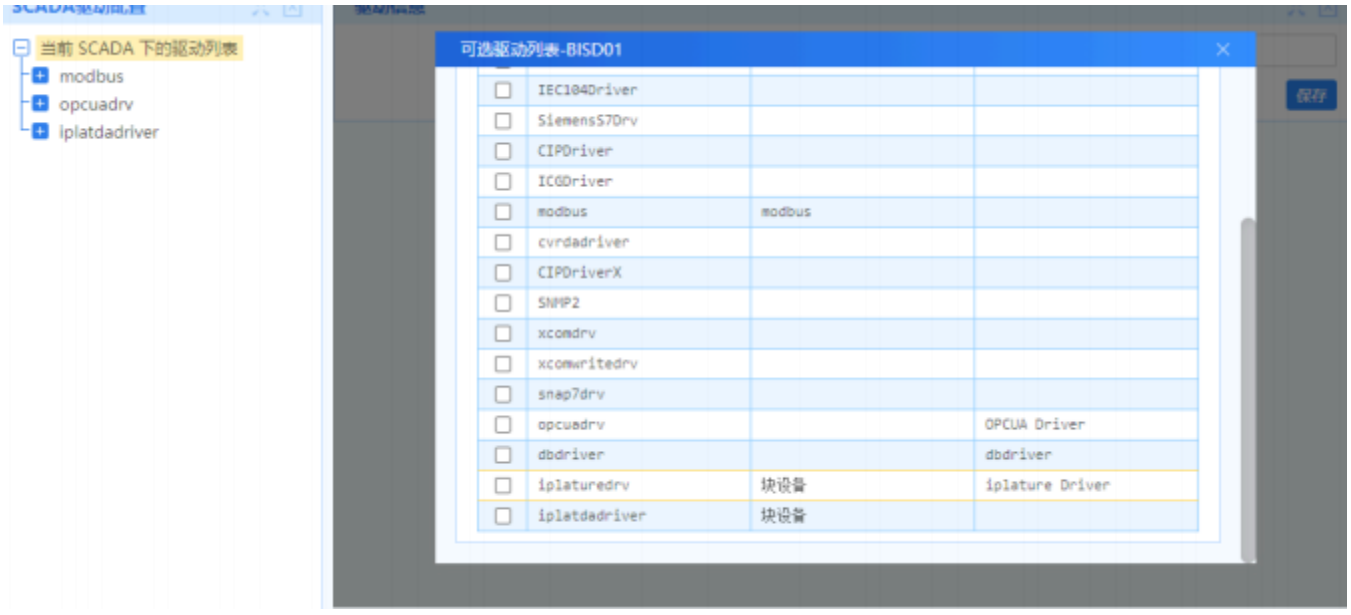
||**(3)设置网关软件整体重启key为"softallreboot"**
||**(4)设置网关软件模块重启key为"网关软件进程名"，如mod
||**(5)设置日志上送开启或关闭key为"sentlog"**
||**(6)设置日志等级更改key为"网关软件进程名"，如modbus
||**(7)设置存储到边缘redis数据的保存时间key为"expirati
value1...valuen|String|(1)设置tag点的值，value为想要设置的tag的值
||(2)设置心跳间隔，value为想要设置的心跳间隔（单位：秒）
||**(3)设置网关软件整体重启，value为预期的超时时间（单位
||**(4)设置网关软件模块重启，value为预期的超时时间（单位
||**(5)设置日志上送开启或关闭value为"enable"或者"disab
||**(6)设置日志等级value为"debug"、"info"、"warn"等
||**(7)设置存储到边缘redis数据的保存时间（单位为秒）**

- 驱动配置
- Tag点配置
- 驱动初始化参数配置
- 如何启动驱动

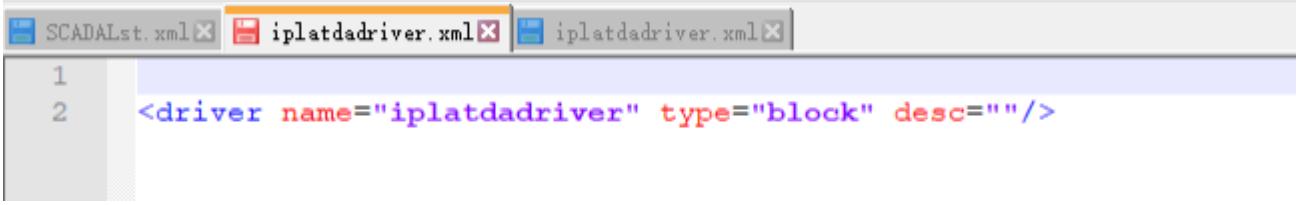
数据模拟器，用于测试iplatda的一些功能，如报警，数据发布等等。数据类型**支持AI,DI,TXT**,不支持AO,DO,CA等类型。

驱动配置

在scada驱动列表中新增"iplatdadriver"驱动，点击保存。



注意：本驱动无需添加设备以及数据块等信息。生成的配置文件如下：



Tag点配置

只需要在所选驱动下拉框中选择iplatdadriver即可，无需配置设备，设备地址等。

驱动初始化参数配置

本驱动提供初始参数配置，在projects\defaultproject\config目录下的iplatdadriver.xml中可以配置如下参数：

1. mode：数据模拟模式，mode 0表示数据保持不变，mode 1表示数据线性增长
2. refreshtime：数据更新周期，默认为1000ms
3. startvalue：数据起始值，默认为0
4. step：数据步进值，数据每次更新时的涨幅
5. number：tag点数量

iplatdadriver.xml

```
<?xml version="1.0" encoding="GB2312"?>
<iplatdadriver version="1.0">
    <mode mMode="1"/>
    <refreshtime mSecond="1000"/>
    <startvalue mStartValue="0"/>
    <step mStep="1"/>
    <number mNumber="15000"/>
</iplatdadriver>
```

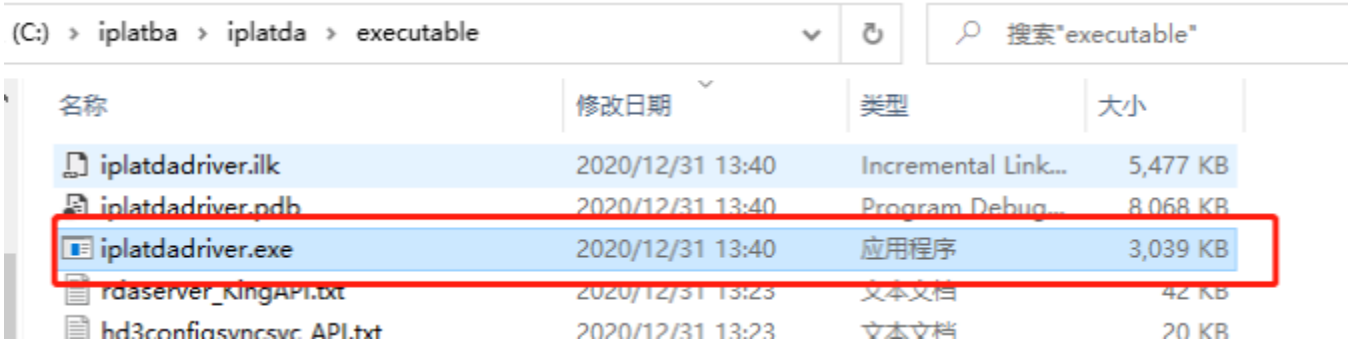
如何启动驱动

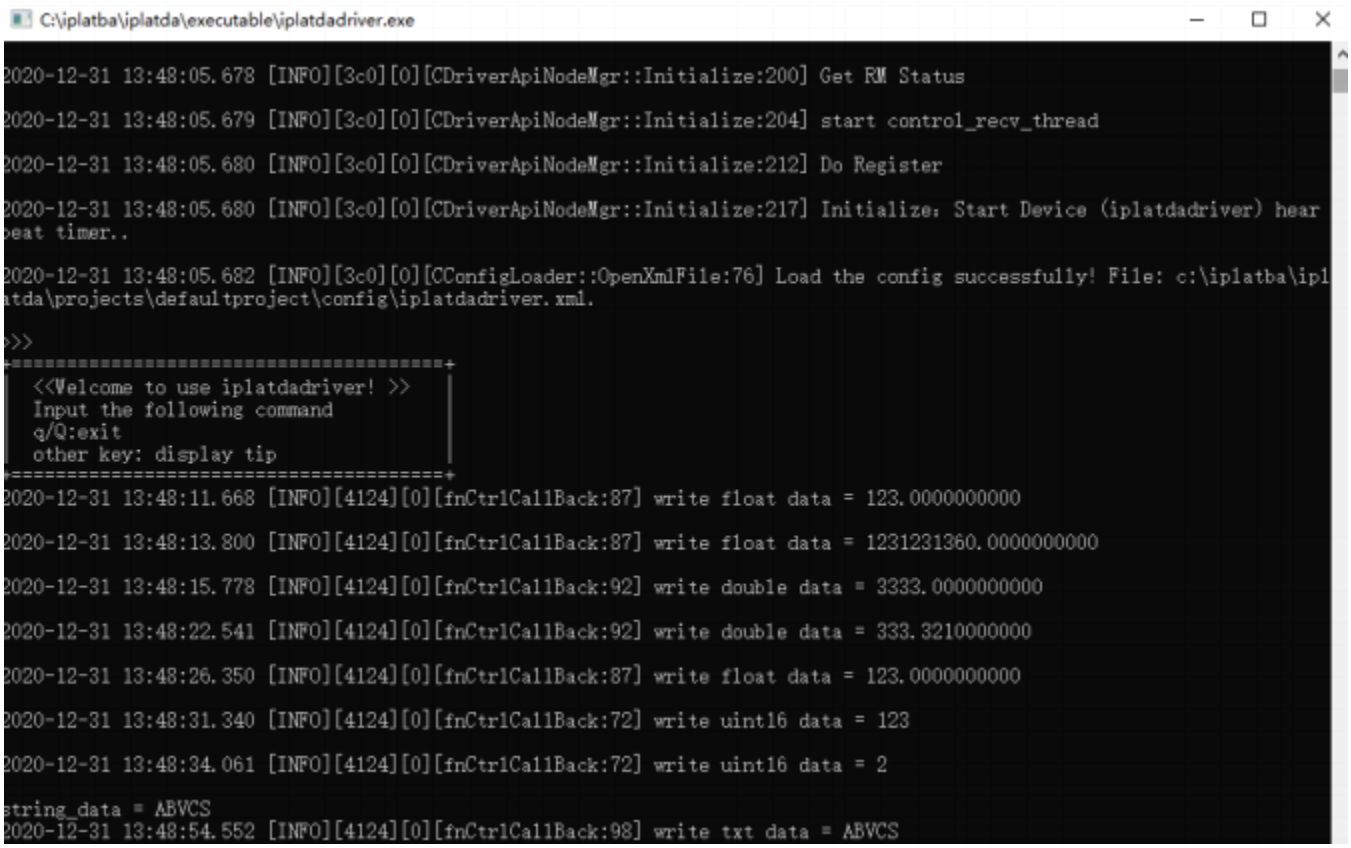
Linux下启动：

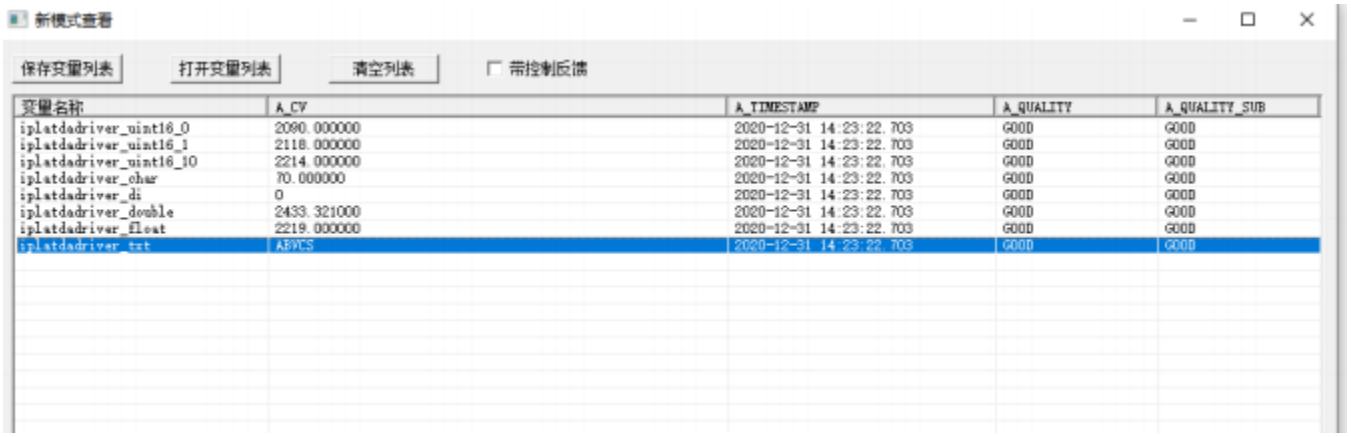
在executable目录下执行：./iplatdaadriver tdrv

Windows下启动：

在..\iplatda\executable目录下找到iplatdadriver.exe,双击启动，若已经配置了iplatdadriver的tag点，通过rdamonitor可以读/写相关tag点.







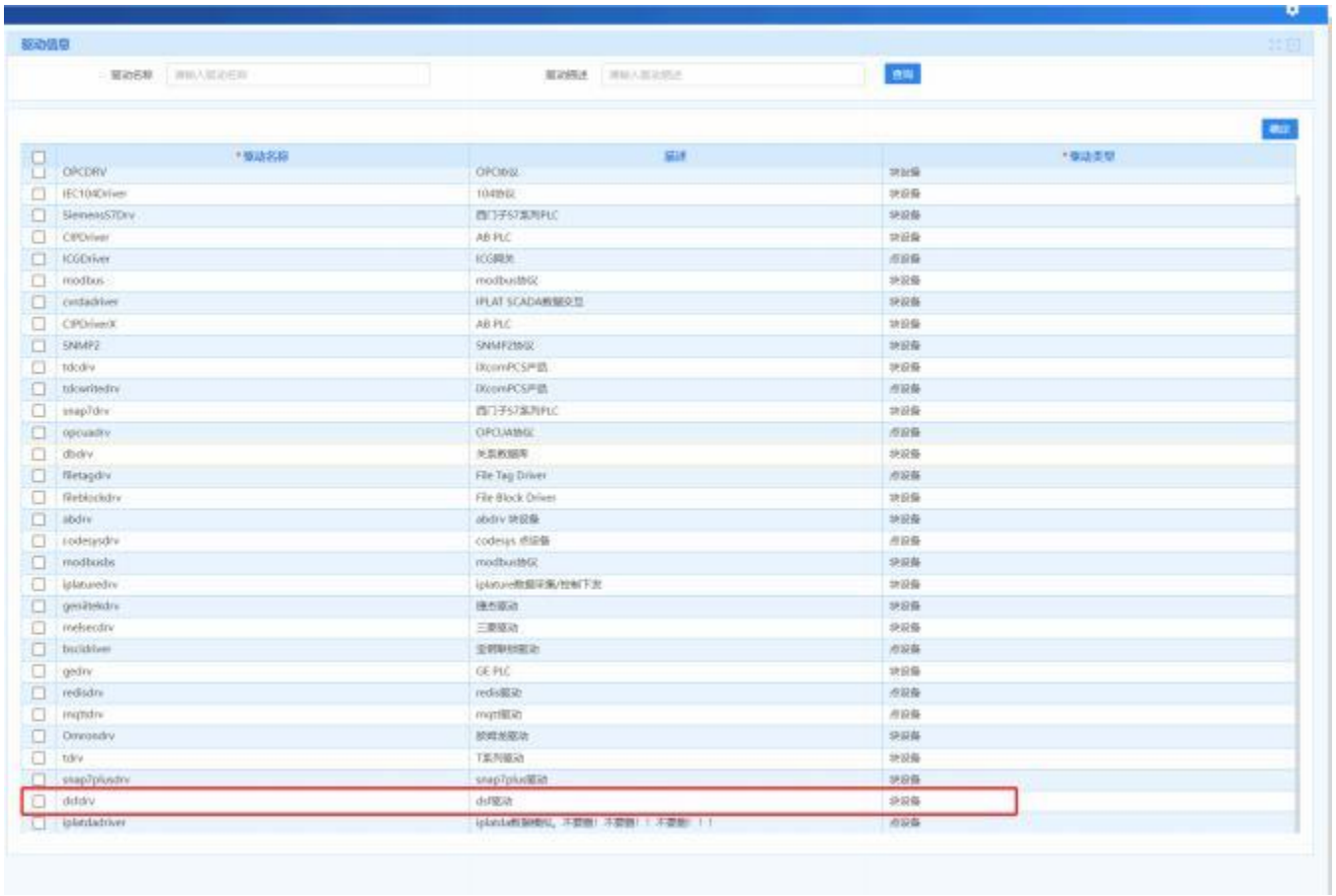
- 添加dsf驱动
- 配置dsf驱动设备
- dsf驱动的部署xml

添加dsf驱动

添加scada节点后，点击配置好的scada节点，在下拉列表中点击驱动配置。

在scada驱动配置中，右键驱动列表，点击添加驱动。

进入系统驱动配置界面，按照红框中的设置添加dsf驱动。



配置dsf驱动设备

•设备名称：设备的名称，不可以重复。只能输入数字字符串和下划线。设备名自增；
•描述：该设备的描述；
•IP地址：IP地址；
•端口：c端口号，默认为1224；
•响应超时：接收超时，单位毫秒，默认1000ms。校验规则正整数；
•轮询周期：单位毫秒，默认1000ms
•任务号：大于0小于101的正整数，自增；
•支持多连接：默认值否，不支持修改

dsf驱动的部署xml

```
  <driver name="dsfdrv"type="block"desc="">
<device
name="dsfdrv_Device0"conntype="Other"connparam="ip=127.0.0.1;port=1224;multilink=0
;"c
<device
name="dsfdrv_Device11"conntype="Other"connparam="ip=127.0.0.1;port=12245;multilink
=0;"
<device
name="dsfdrv_Device23"conntype="Other"connparam="ip=127.0.0.1;port=12244;multilink
=0;"
  </driver>
```

•块设备驱动配置.
•点设备驱动配置

通用驱动配置分为块配置(Block)和点配置(Item).

块设备驱动配置.

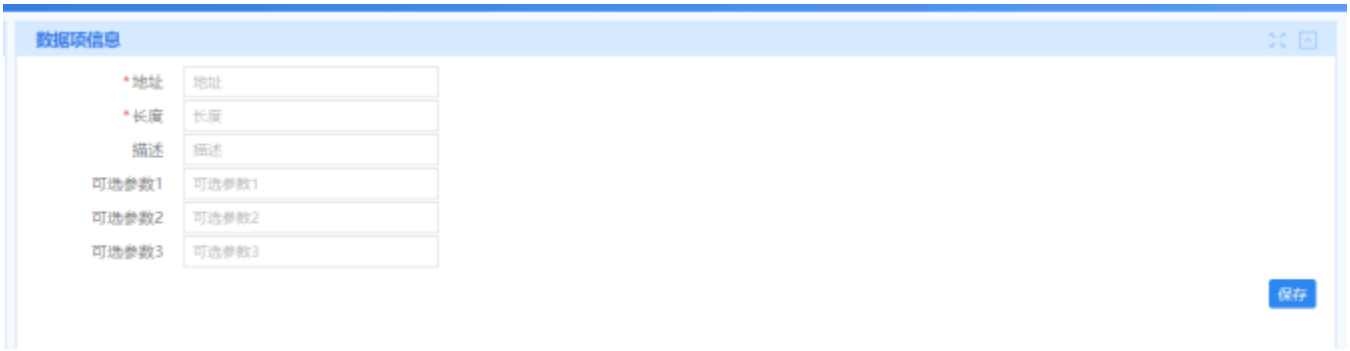
块设备配置文件格式如下:

```
  <?xml version="1.0"encoding="GB2312"?>
  <driver name="BlockDrvCfg"type="block"desc="">
<device
name="BlockDrvCfg_Device0"conntype="TCPClient"connparam="ip=127.0.0.1;port=502;mul
tiLi
<datablock name="123"type="123"address="123"elemcount="123123"elembytes="
"cyclerate="1000
    </device>
  </driver>
```

点设备页面配置如下:

设备配置:

块配置:



点设备驱动配置

点设备配置文件格式如下:

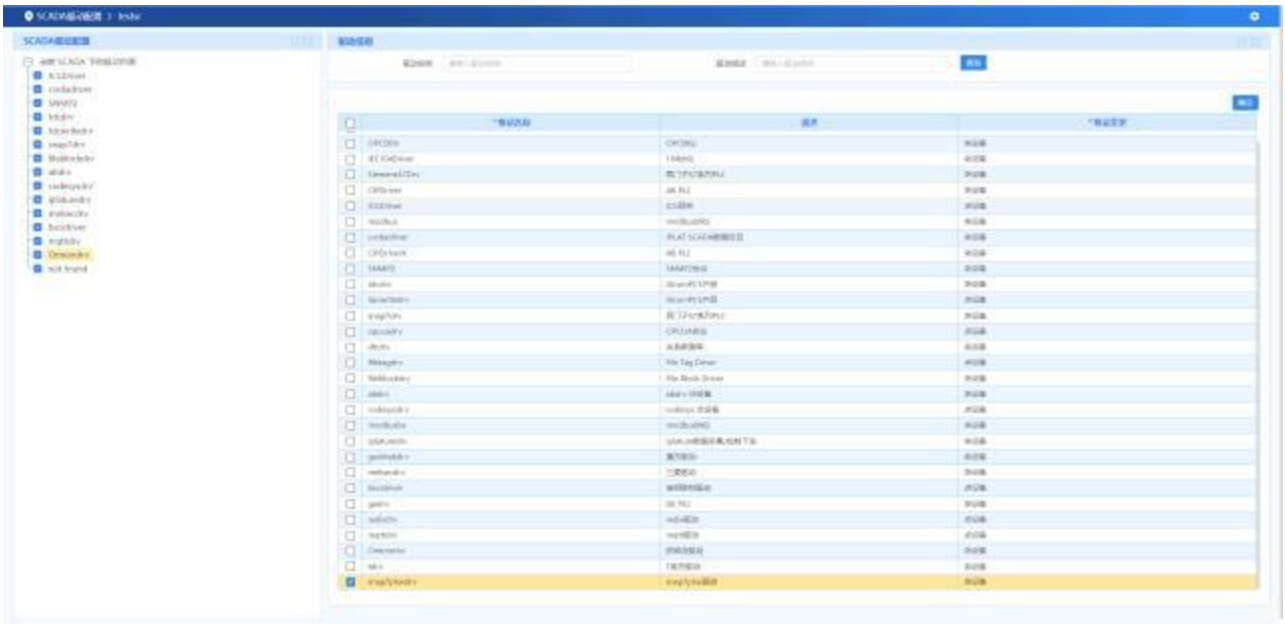
```
 <?xml version="1.0"encoding="GB2312"?>
 <driver name="ItemDrvCfg"type="item"desc="">
<device
name="ItemDrvCfg_Device0"conntype="TCPClient"connparam="ip=127.0.01;port=502;
multiLink
<datablock address="123"elemcount="123"elembytes=""desc=""param1=""param2="
"param3=""/>
   </device>
 </driver>
```

块设备配置页面如下:

设备配置:

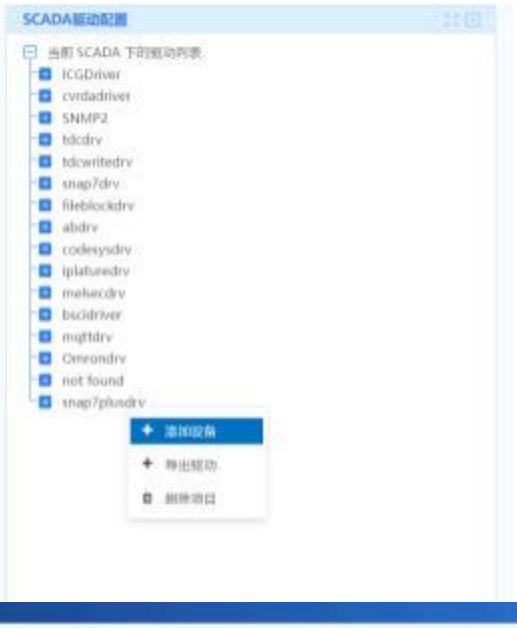
块配置:

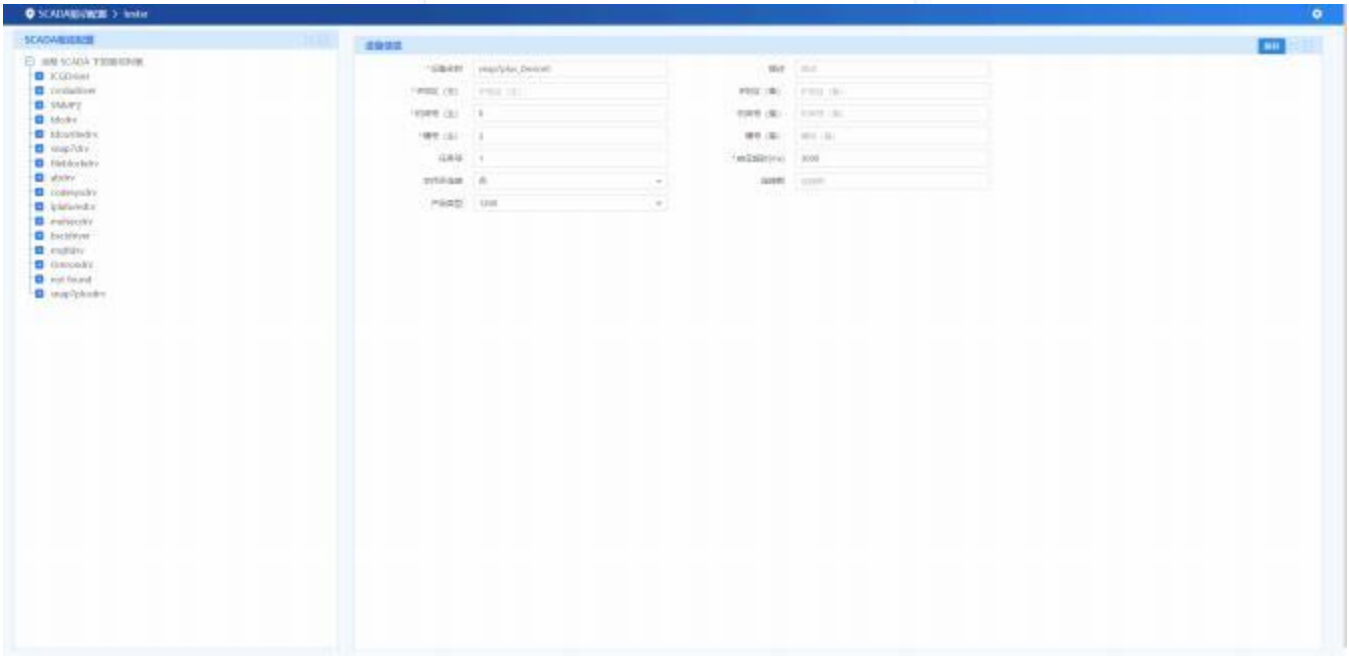
进入系统驱动配置界面，按照红框中的设置添加snap7plusdrv驱动



配置snap7plusdrv驱动设备

右击snap7plusdrv选择添加设备

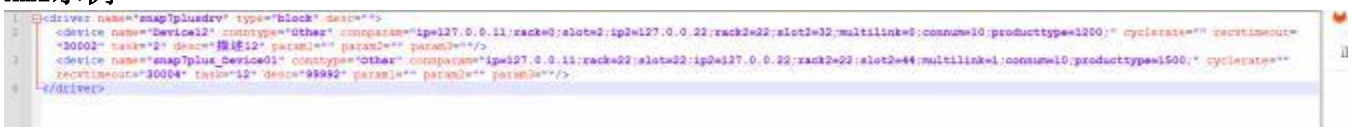




参数说明：

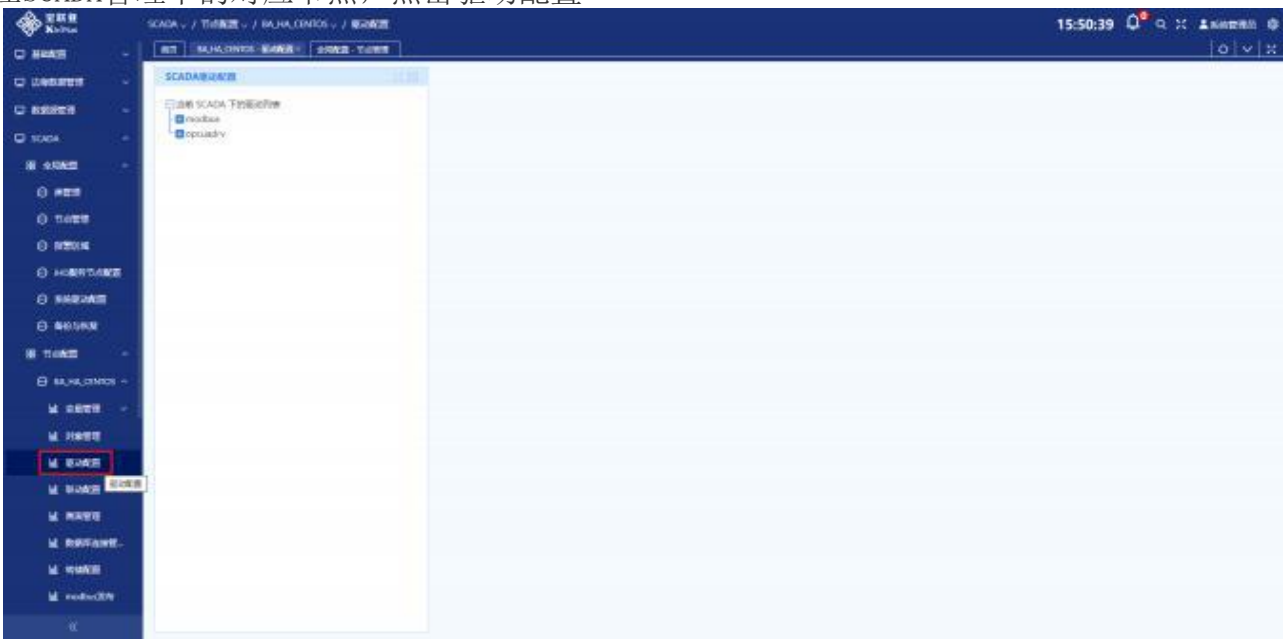
- 设备名称：不能为空，不能以数字开头；
- 描述：自定义的设备描述信息；
- IP地址（主）：欲连接主设备的IP；
- IP地址（备）：欲连接备设备的IP；
- 机架号（主）：欲连接主设备的机架号，默认0；
- 机架号（备）：欲连接备设备的机架号；
- 槽号（主）：欲连接主设备的槽号，默认2；
- 槽号（备）：欲连接备设备的槽号；
- 任务号：同一任务号下的设备消息放在同一个进程中；
- 响应超时：连接设备和发送数据的响应超时时间；
- 支持多连接：该设备是否支持多连接，1表示支持多连接；
- 连接数：连接数；
- 产品类型：产品类型；

xml示例



添加tdrv驱动

1. 在SCADA管理下的对应节点，点击驱动配置



2. 在驱动配置页面中，点击添加驱动，勾选tdrv驱动，再点击确定

添加tdrv设备

1. 右键点击添加设备

2. 填写设备配置，如下图

参数说明：

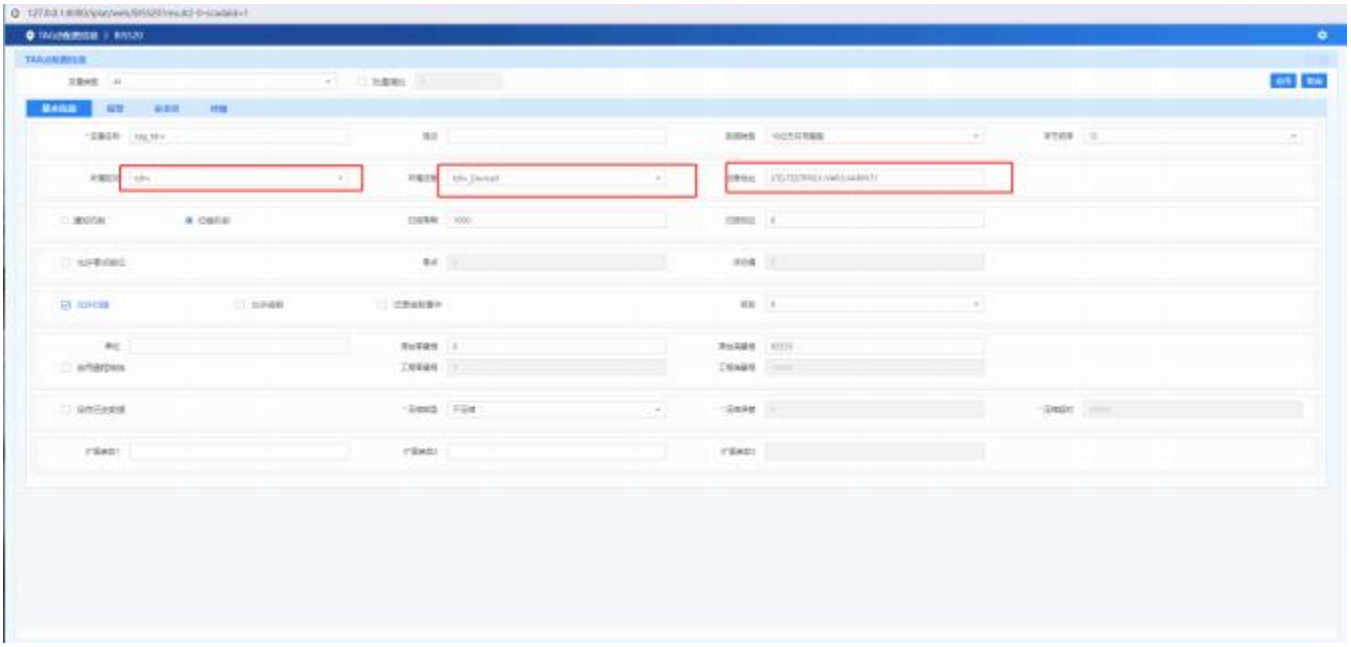
- 设备名称：不能为空，不能以数字开头；
- 描述：自定义的设备描述信息；
- 轮询周期：读取对应值得周期时间，tdrv的轮询周期只有设备上的配置有效，tag点配置的轮询周期无效；
- 响应超时：连接设备和发送数据的响应超时时间；
- IP地址：欲连接的PLC的IP；
- 用户名：欲连接的PLC的用户名；
- 密码：欲连接的PLC的密码；
- 任务号：同一任务号下的设备消息放在同一个进程中；
- 支持多连接：该设备是否支持多连接，【是】驱动的主备机采用多连接的方式连接PLC，【否】驱动的主备机采用单连接的方式连接PLC；
- 产品类型：选择对应的plc类型，驱动会用不同的策略进行数据读取，目前的类型有"T3""T4"；

***注：【产品类型】选择类型与PLC型号不符，也可以部署、读写数据，只是会对读取性能造成一些影响。**

PLC会根据自身的【产品类型】对可以成功读取的点数进行限制T3所有连接加起来能读取5000点，T4所有连接加起来能读取50000超出数量的部分会无法读取。也就是说，如果tdrv采集T3plc，配置了读取6000点的数据，T3plc只会返回至多5000点的数据。

*注：天行PLC采用符号访问的方式，因此不需要配置数据块

添加变量



添加变量步骤与一般设备相同，这里主要描述变量关键信息:

所属驱动:选择tdrv;

所属设备:选择tdrv对应设备;

设备地址:PLC提供的符号名

数据类型:

PLC类型	C++ 类型	da对应类型	备注
BOOL	bool	DI点	
SINT	int8_t	AI点 8位有符号整型	
INT	int16_t	AI点 16位有符号整型	
DINT	int32_t	AI点 32位有符号整型	
TIME	int32_t	AI点 32位有符号整型	

LINT	int64_t		da暂不支持，建议使用AI点双精度浮点型
LTIME	int64_t		da暂不支持，建议使用AI点双精度浮点型
BYTE	uint8_t	AI点 8位无符号整型	
USINT	uint8_t	AI点 8位无符号整型	
WORD	uint16_t	AI点 16位无符号整型	
UINT	uint16_t	AI点 16位无符号整型	
DWORD	uint32_t	AI点 32位无符号整型	
UDINT	uint32_t	AI点 32位无符号整型	
DATE	uint32_t	AI点 32位无符号整型	
TOD	uint32_t	AI点 32位无符号整型	
TIME_OF_DAY	uint32_t	AI点 32位无符号整型	
DT	uint32_t	AI点 32位无符号整型	
DATE_AND_TIME	uint32_t	AI点 32位无符号整型	
LWORD	uint64_t		da暂不支持，建议使用AI点双精度浮点型
ULINT	uint64_t		da暂不支持，建议使用AI点双精度浮点型
LDATE	uint64_t		da暂不支持，建议使用AI点双精度浮点型
LTOD	uint64_t		da暂不支持，建议使用AI点双精度浮点型

LDT	uint64_t		da暂不支持，建议使用AI点双精度浮点型
CHAR	char	AI点 8位有符号整型	
REAL	float	AI点 单精度浮点型	
LREAL	double	AI点 双精度浮点型	
STRING	string	TXT点	plc中的string设定【系统中实际占用存储空间=[N+N是常数，最大254;缺省值N=80。】 da中的tdrv的txt点设定【一个txt点可以用的总长度
?	string		点不存在时，类型返回的值

```
EntryNotFound (FileSystemError): Error: ENOENT: no such file or directory,

 open 'd:
```

数据状态查询工具功能说明

1. 该程序的主要功能为一次性拉取指定驱动下所有点的数据，并显示多少点为BAD，目前暂时还不支持区分设备。
2. 目前仅支持在windows上运行本程序。
3. 要读取linux上运行工程的点值状态，需要提前进行一些额外的操作，下文会详细阐述。

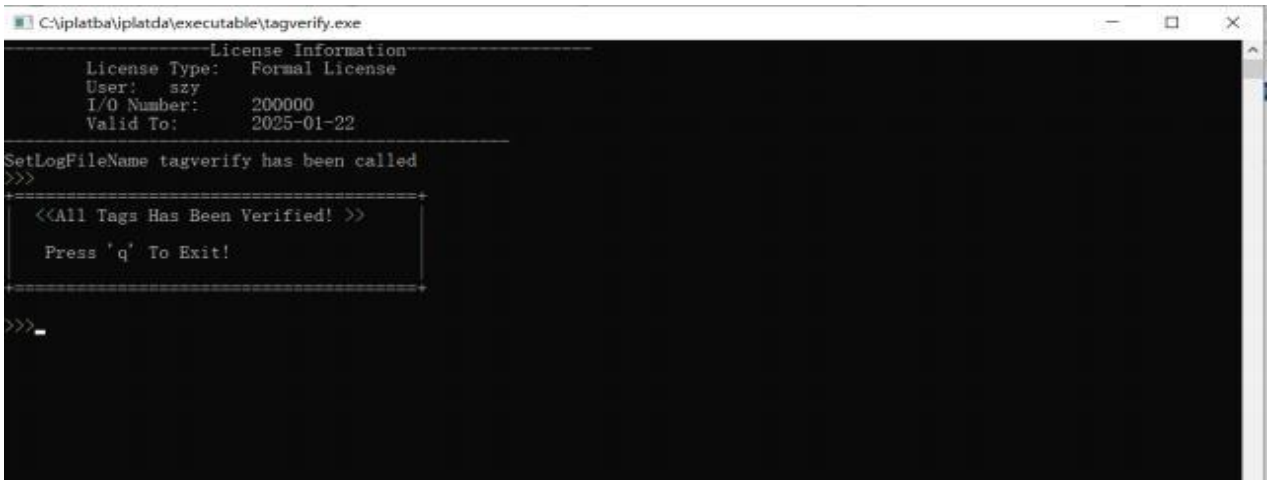
Windows

1. 首先确认安装目录下的executable目录中是否包含了tagverify.exe该文件。有该文件再开始执行以下步骤。
2. 前往projects\defaultproject\config目录，找到tagverify.xml文件，打开后界面如下图：

```
<?xml version="1.0" encoding="GB2312" ?>
<!-- tagverify.exe可将指定驱动下的所有变量读一次并将点名，
     数据质量等信息保存为csv文件输出到executable目录下
     mdriver的值为指定驱动，mode的值可填写0或1，
     0为同步读，1为异步读，建议使用异步读，性能比同步读高 -->
<tagverify version="1.0">
    <simdriver mdriver="melsecdrv" />
    <mode mMode="1" />
</tagverify>
```

其中红框处修改为需要监控的驱动名称，下面这行的mMode通常使用默认值1（异步读）即可。

2. 回到executable目录，双击执行tagverify.exe，会自动进行读值操作，该过程时间会根据点数规模大小变化，读值完成后黑窗口显示如下图所示：



此时程序执行完成，按Q退出程序即可。

1. 在executable目录下寻找最新生成的csv文件，文件名称格式通常为tagverify_驱动名称_yymmdd_hhmmss.csv，打开后内容如下：

	A	B	C	D	E	F
1	TAGNAME	TAGVALUE	TIMESTAMP	QUALITY	TAGADDR	TAGTYPE
2	Total Tags 505 BAD Tags 505					
3	melsec.AEX_A02_2F_E_PdT_B01_OFS	???		33.10.4 BAD	A02_4_AsEX_W_PLC01:D3512	AI
4	melsec.CR_A02_3F_E_HCV_B02A_2_DEVN_VALUE	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7440	AI
5	melsec.CR_A02_3F_E_HCV_B02A_2_HMI	???		00:00:0 BAD	A02_4_CR_E_PLC01:D3042	AI
6	melsec.CR_A02_3F_E_HCV_B02A_2_OV	???		00:00:0 BAD	A02_4_CR_E_PLC01:D4042	AI
7	melsec.CR_A02_3F_E_HCV_B02B_1_DEVN_DELAY	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7452	AI
8	melsec.CR_A02_3F_E_HCV_B02B_1_DEVN_VALUE	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7450	AI
9	melsec.CR_A02_3F_E_HCV_B02B_1_HMI	???		00:00:0 BAD	A02_4_CR_E_PLC01:D3044	AI
10	melsec.CR_A02_3F_E_HCV_B02B_1_OV	???		00:00:0 BAD	A02_4_CR_E_PLC01:D4044	AI
11	melsec.CR_A02_3F_E_HCV_B02B_2_DEVN_DELAY	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7462	AI
12	melsec.CR_A02_3F_E_HCV_B02B_2_DEVN_VALUE	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7460	AI
13	melsec.CR_A02_3F_E_HCV_B02B_2_HMI	???		00:00:0 BAD	A02_4_CR_E_PLC01:D3046	AI
14	melsec.CR_A02_3F_E_HCV_B02B_2_OV	???		00:00:0 BAD	A02_4_CR_E_PLC01:D4046	AI
15	melsec.CR_A02_3F_E_HCV_B02C_1_DEVN_DELAY	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7582	AI
16	melsec.CR_A02_3F_E_HCV_B02C_1_DEVN_VALUE	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7580	AI
17	melsec.CR_A02_3F_E_HCV_B02C_1_HMI	???		00:00:0 BAD	A02_4_CR_E_PLC01:D3126	AI
18	melsec.CR_A02_3F_E_HCV_B02C_1_OV	???		00:00:0 BAD	A02_4_CR_E_PLC01:D4126	AI
19	melsec.CR_A02_3F_E_HCV_B02C_2_DEVN_DELAY	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7592	AI
20	melsec.CR_A02_3F_E_HCV_B02C_2_DEVN_VALUE	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7590	AI
21	melsec.CR_A02_3F_E_HCV_B02C_2_HMI	???		00:00:0 BAD	A02_4_CR_E_PLC01:D3128	AI
22	melsec.CR_A02_3F_E_HCV_B02C_2_OV	???		00:00:0 BAD	A02_4_CR_E_PLC01:D4128	AI
23	melsec.CR_A02_3F_E_HCV_B02D_1_DEVN_DELAY	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7602	AI
24	melsec.CR_A02_3F_E_HCV_B02D_1_DEVN_VALUE	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7600	AI
25	melsec.CR_A02_3F_E_HCV_B02D_1_HMI	???		00:00:0 BAD	A02_4_CR_E_PLC01:D3130	AI
26	melsec.CR_A02_3F_E_HCV_B02D_1_OV	???		00:00:0 BAD	A02_4_CR_E_PLC01:D4130	AI
27	melsec.CR_A02_3F_E_HCV_B02D_2_DEVN_DELAY	???		00:00:0 BAD	A02_4_CR_E_PLC01:D7612	AI

第二行第一列中显示了统计的总点数和BAD的点的个数，TIMESTAMP具体显示需要选中查看。

Linux

1. 将Linux上运行工程的projects\defaultproject\config目录覆盖Windows机器上的同名目录，确认Windows机器与Linux机器网络能连通。
2. 根据实际情况修改config目录中如下几个文件，SCADALst.xml，RMService.xml，文件内容中对应scada的IP按照实际修改
3. 按照Windows操作步骤执行，最后生成的csv文件为读取目标Linux机器的结果。

3.5 变量类型

变量的类型

以下对iPlat中的各种类型的变量做一个简介：

变量类型	变量名称	变量分类	变量描述
设备变量	AI	模拟量输入	提供对模拟量数据的读写访问，并允许设置
	DI	数字量输入	提供对数字量数据的读写访问，并允许设置
	AO	模拟量输出	向I/O驱动发送模拟量数据

	DO	数字量输出	向I/O驱动发送数字量数据
	MDI	多态数字量输入	支持0~7共8种状态，可设置报警
	TXT	文本	支持最大256字符的文本读写
	BLOB	二进制大对象	用于存储大型文件（如图片、声音）
	CA	模拟计算量	基于表达式计算模拟量结果
	CD	数字计算量	输出为DI量，基于表达式计算
	LA	模拟量输入	支持16种区间报警，含死区特性
对象变量	_ 创建类模板及自定义属性		将设备抽象为类，适用于同类设备较多的工
	-添加子系统		
	-创建对象		
	-对象变量查询		
	-对象点使用		
中间变量	-中间变量		用于内部值传递，不与实际设备交互
本地变量	-本地变量		仅存在于当前浏览器，不同步至服务器或其
全局变量	-全局变量		可在全局变量标签页中选择关联的TAG点
系统变量	-系统变量		系统内置变量，表示系统状态

3.6 变量属性

变量的属性

变量的属性包括名称、描述、类型、所属驱动等。以下介绍基本属性，其他特有属性在各变量类型中详述。

基本信息tab页配置

- **变量名称**：变量的名字，用于标识Tag点。因为变量可以用于iplat系统中脚本的编写，因此，命名变量名称需根据一定的规则：
  - 只能包含中文、数字、英文字母或下划线“_”和点“.”(只能包含一个点)
  - 名字不能为空，长度不超过64（含结尾“()”
  - 不能以数字或"_"开头，不可全数字组成
  - 名称不能等于表达式里面的关键字
    （
    "in","max","min","sin","cos","tan","abs","asin","acos","atan","log","ceil","pow","sqrt","eq","neq","conret","strconcat","todecimal","int","exp","fmod"等）
  - 类自定义属性不能以F_或A_开头
  - 类的自定义属性不能以数字或"_"开头，不可全数字组成
  - 变量名称第二部分不能以F_或A_开头
- **描述**：对变量的补充说明，同时也是变量报警时的报警描述。在作为报警描述显示时，可以通过配置iplat-da中的配置文件AlmServiceCfg.xml，选择显示/不显示报警条件（例如在报

警描述的结尾添加，显示报警条件：[高高报警值]:10），默认配置是不显示报警条件。

- **数据类型**：配置的变量所用的类型，DI变量不用选择，AI是模拟量，有浮点型、整型等各种类型需要选择。
- **所属驱动**：选择所新增的这个变量所要采集的数据，来自哪种驱动。
- **设备名称**：选择所新增的这个变量所要采集的驱动下对应的设备。
- **设备地址**：填写新增的这个变量的寄存器具体地址。如果选择了是驱动协议是免块方式的配置，请直接填写寄存器地址，如果选择的是驱动中配置数据块的方式，请按照“数据块：寄存器地址”的格式填写。
- **扫描周期**：点采集的周期，单位毫秒。
- **启用量程转换**：原始量程按照比例转换成工程量程，原始量程和工程量程一致时，限制的是上下限范围值。
- **保存历史数据**：组态中历史趋势点数据曲线展示的前提。
- **压缩类型**：不可选（不压缩，压缩，对历史数据值进行压缩）。
- **压缩误差**：压缩可接受的误差值范围；例如：填“0.1”，即变化误差为0.1的值会被压缩。DI类型数据，此项请填“0”，系统会自动对DI类型数据进行压缩。
- **压缩超时**：超过该时间范围强制记录历史数据。
- **扩展参数**：存储类的属性的扩展信息，该扩展信息可完全由用户自定义。报警tab页配置

概述：根据是否需要报警进行报警的配置，配置后可以在组态的报警控件中过滤显示。

注意：由于AI点报警配置和DI点配置报警项有些不同，请到各自的页面查看具体的报警解释：

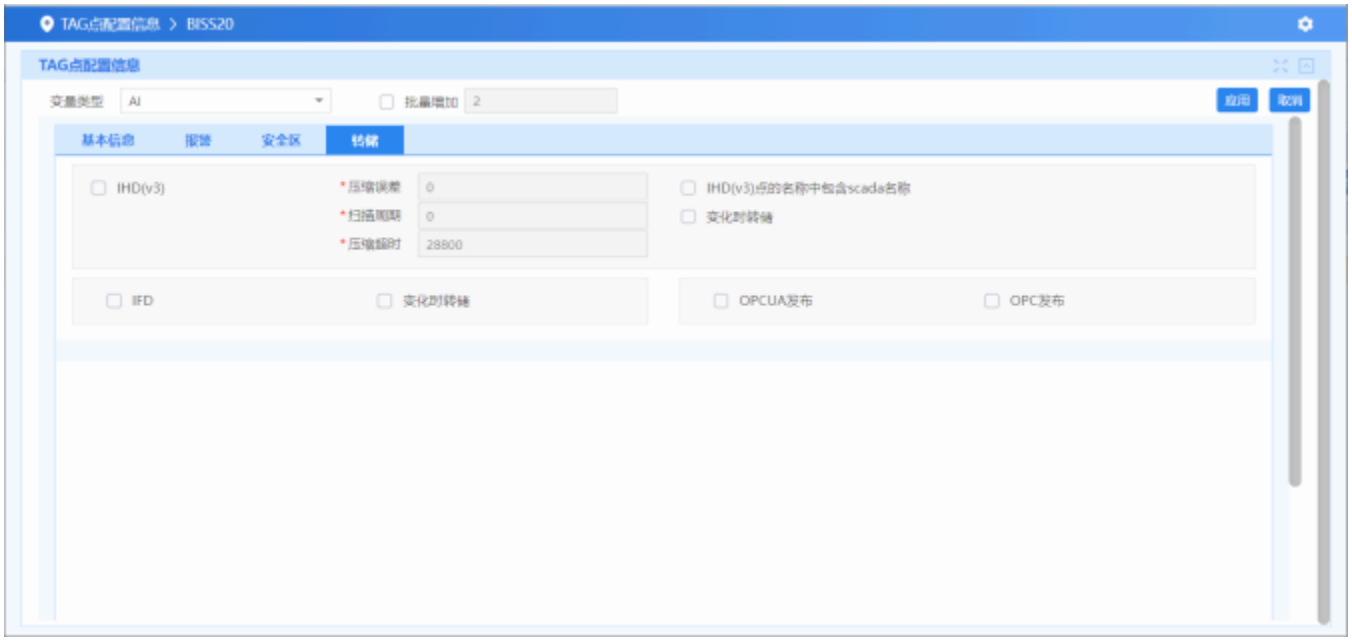
AI：AI点配置信息详解
DI：DI点配置信息详解

- **启用报警处理**：是否启用报警；
- **报警区域名称**：选择已配置的报警区域；
- **报警值死区**：避免报警频繁触发；
- **时间死区**：需维持报警状态一段时间才触发（变化率报警除外）。

详细配置请参考：

- AI点配置详解
- DI点配置详解

转储tab页配置



- **IHD**：勾选后，部署会进行转储
- **压缩误差**：压缩可接受的误差值范围，填“0”表示不开压缩；若要对点进行压缩，可以在此项中填写压缩误差值。例如：填“0.1”，即变化误差为0.1的值会被压缩。DI类型数据，此项请填“0”，系统会自动对DI类型数据进行压缩。
- **扫描周期**：两次扫描周期所间隔的时间
- **压缩超时**：超过该时间范围强制记录历史数据
- **IHD点名称包含SCADA名称**：转储到ihd的点名称含有scada节点名称
- **变化时转储**：勾选后，数据变化时进行转储，不勾选根据采集周期转储
- **IFD**：勾选后，iplatda系统会将该点转储到目标influxdb中，具体配置参考转储数据到时序数据库
- **OPCUA发布**：勾选后iplatda系统会将数据以opcua的形式进行发布，可以用opcua的客户端进行查看
- **OPC发布**：选后iplatda系统会将数据以opc的形式进行发布，，可以用opc的客户端进行查看

3.7 变量命名规则

变量的名字，用于标识Tag点。因为变量可以用于iPlat系统中脚本的编写，因此，命名变量名称需根据一定的规则如下

设备点名的命名规则

- 长度≤64，不为空；
- 可含中文、数字、字母、英文标点（`-:._()[]+=%@#`）和中文标点（`-：。（）【】+=%@#`）；
- 最后一个点后不能含`F_`或`A_`开头字符串。

对象点的命名规则

- 可含中文、数字、字母、下划线“_”；
- 名字不能为空，长度不超过64（含结尾`\0`）；
- 不能以数字或“_”开头，不可全数字组成；
- 不能使用表达式关键字
- 名称不能等于表达式里面的关键字

（"in","max","min","sin","cos","tan","abs","asin","acos","atan","log","ceil","pow","sqrt","eq","neq","conret","strconcat","todecimal","int","exp","fmod"等）；

- 类自定义属性不能以`F_`或`A_`开头；

•类的自定义属性不能以数字或“_”开头，不可全数字组成
•变量名称第二部分不能以F_或A_开头；
•变量名称不能含点.。

3.8 配置AI变量基本属性

1. AI的属性如下图所示：

注意

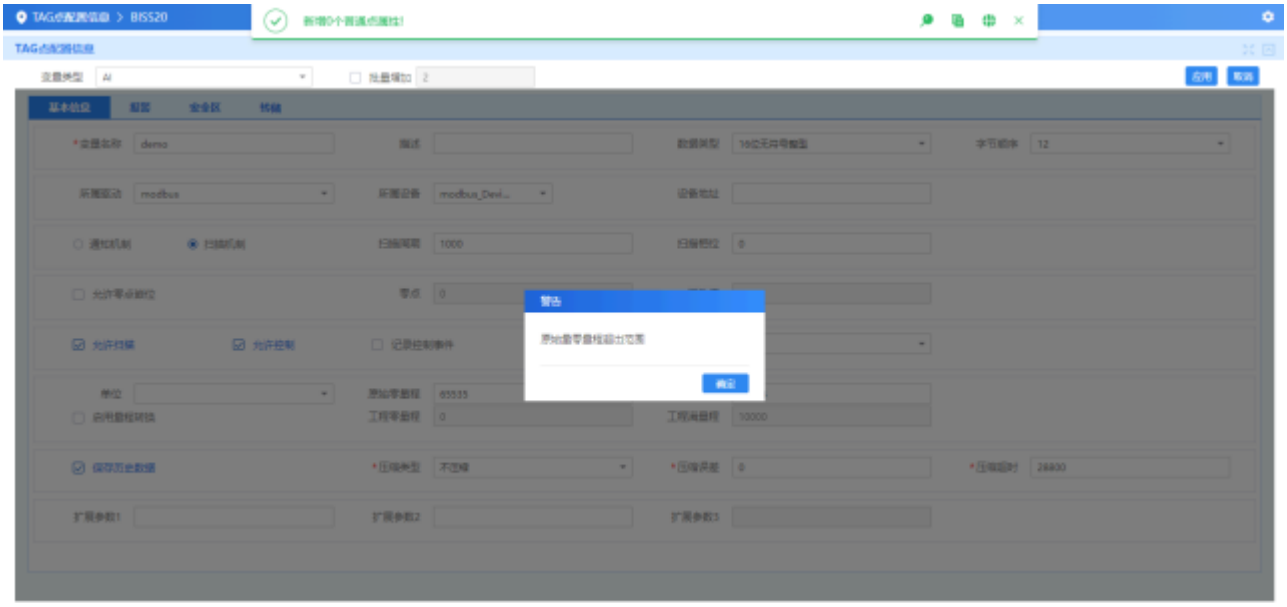
如果用户没有配置"所属驱动"、"设备名称"和"设备地址"，则添加的变量默认为全局变量，全局变量可以在所有客户端中通用。

•数据类型：

数据类型	取值范围	
16位无符号整型	0~65535	
16位有符号整型	-32768~32767	
32位无符号整型	0~4294967295	
32位有符号整型	-2147483648~2147483647	
8位无符号整型	0~255	
8位有符号整型	-128~127	
单精度浮点型	-2147483648~2147483647	32位的4个字节，这里驱动数据块的寄存器也需要配
双精度浮点型	-2147483648~2147483647	64位的8个字节;小数点后不能超过6位有效数字

•字节顺序：通常规情况中，用户不用修改选项。该功能用以实现某些设备寄存器高低位翻转的特殊情况的读写，例如16位无符号整形258，在x86结构机器上其16进制表示为0x0102，但是从设备读取该数据时其顺序可能为0x0201，此时需要作"21"转换，即将第一个字节与第二个字节进行交换，从而使得从设备读取的数据可以在终端上正确显示

•扫描设置：
  ◦通知机制：指变量值有变化的时候进行通知读取
  ◦扫描机制：通过轮循扫描的方式对变量进行读取，可对"扫描周期"和"扫描相位"进行设置。这里的扫描周期是过程数据库读取驱动的时间
  ◦扫描周期：两次扫描周期所间隔的时间
  ◦扫描相位：用于设置起始扫描时间。避免多组同时扫描造成的资源不足风险。注意：为了避免丢点现象，如果用户知道设备的数据变化周期，建议将驱动的刷新周期配置为设备的数据变化周期的一半，点的扫描周期配置为数据变化周期的4分之一，如设备的数据变化周期为4000毫秒，那么驱动刷新周期为2000毫秒，点的扫描周期为1000毫秒
•零点嵌位：当数据值在"零点"附近"浮动值"范围内，那么这个数值就当成为"零点"值，如"零点"基准值设置为100，浮动值设置为1，那么>99或<101的值都会转换成100。选择"允许零点嵌位"复选框，可对"零点"和"浮动值"进行设置
•启动选项：
  ◦允许控制：用户如果要对变量进行写控制，需要勾选"允许控制"，iplat提供了"允许扫描"和"禁止扫描"的选项，方便用户设置是否启用扫描机制
  ◦记录控制事件：当我们对过程量进行写值控制的时候，用于记录控制的事件，此功能在"允许控制"被勾选的情况下可用，并且可以设定事件级别，方便查找时筛选
•启用量程转换：
  ◦单位：变量值的单位可通过"单位"下拉框进行设定，软件提供了米、分、秒、千克、克、米/秒、公里/小时、摄氏度、华氏度等常见单位选项，也可自行输入单位(为兼容老版本，数字数字1到10导入后会根据老版本对应关系进行展示"1"对应"米"，"2"对应"分"，"3"对应"秒"，"4"对应"千克"，"5"对应"克"，"6"对应"米/秒"，"7"对应"公里/小时"，"8"对应"摄氏度"，"9"对应"华氏度"，"10"对应空值)。"原始量"根据实际需求配置。选择"启用量程转换"复选框，可对"工程量"中的"零量程"与"满量程"进行设置
  ◦原始量：从驱动获取到的原始值
  ◦工程量：转换工程单位后的值
  ◦零量程：测量范围的下限值
  ◦满量程：测量范围的上限值
  ◦量程转换：量程是测量范围上限值与下限值之差，而从一个量程变换成另一个量程就叫量程转换（支持小树格式）
•注意：原始量的零量程和满量程值，应在所选的数据类型的范围内（16位无符号整型：0~65535；16位有符号整型：-32768~32767；32位无符号整型：0~4294967295；32位有符号整型：-2147483648~2147483647；8位无符号整型：0~255；8位有符号整型：-128~127；单精度浮点型：-2147483648~2147483647，但小数点后不能超过6位有效数字；双精度浮点型：-2147483648~2147483647，但小数点后不能超过6位有效数字）。当量程超限时会自动修正为该数据类型下的最大值，然后成功保存，并给出提示。如下图：



- 保存历史数据：iplatda系统会将这个点的历史数据保存下来，用来绘制趋势图、脚本查询历史数据等
- 压缩类型：可选（不压缩，压缩，对历史数据值惊醒压缩）
- 压缩误差：压缩可接受的误差值范围；例如：填"0.1"，即变化误差为0.1的值会被压缩。DI类型数据，此项请填"0"，系统会自动对DI类型数据进行压缩。
- 压缩超时：超过该时间范围强制记录历史数据。
- 扩展参数：存储类的属性的扩展信息，该扩展信息可完全由用户自定义

配置AI变量报警属性

1.AI的模拟量报警属性如下图所示：

- 启用报警处理：用户如果需要对变量进行报警处理，需要勾选"启用报警处理"选项，报警区域默认显示为ALL。
- 报警区域名称：为了便于用户监控报警时筛选，我们提供了报警区域配置，具体参考报警区域配置。
- 报警设置：
  - 低低报警，低报警：低于该值则报警。
  - 高报警，高高报警：高于该值则报警。

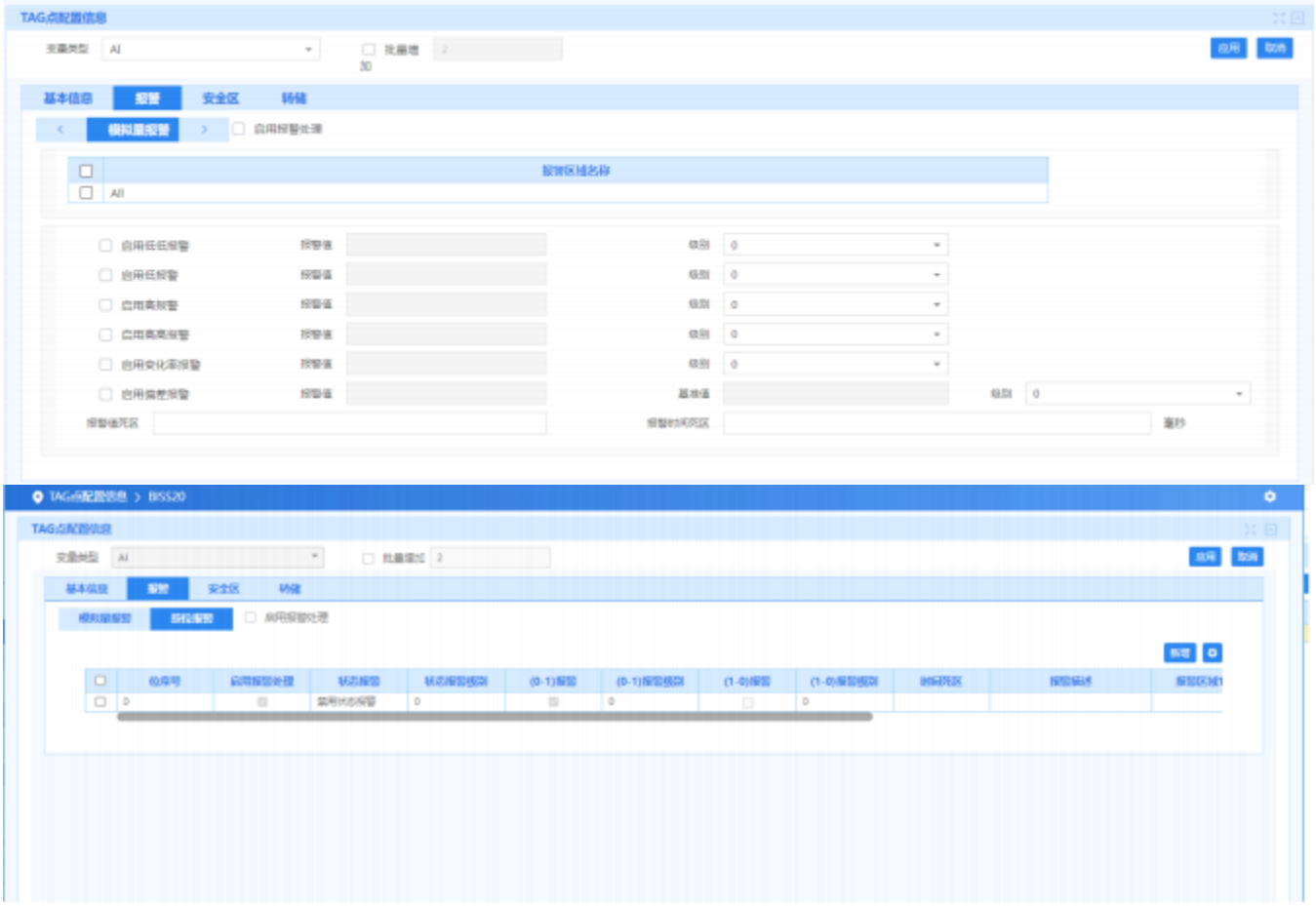
各级报警级别：低低报警值 < 低报警值 < 高报警值 < 高高报警值。报警值级别可选择0-255级别。数字越小，级别越高，级别为0的报警级别是最高的。

- 变化率报警：当过程量的当前值与前一次的值的差除以时间间隔（以秒为单位）再取绝对值大于等于变化率限值时，触发报警。
- 偏差报警：当过程量的当前值与基准值的差取绝对值大于等于偏差限值时，触发报警。
- 报警值死区：配置报警值死区后，报警的触发和恢复都要在触发限值和恢复限制上加上或者减去报警值死区的大小才判断触发或者恢复，避免报警在限值附近波动时频繁发生的报警和恢复消息。例如：当高报警值为24，且设置报警值死区为1时，超过25报警，低于23恢复。
- 时间死区：配置报警时间区后，需要报警在一段时间死区内维持报警状态，才触发报警，否则不触发报警，其中变化率报警不受时间死区限制。

补充

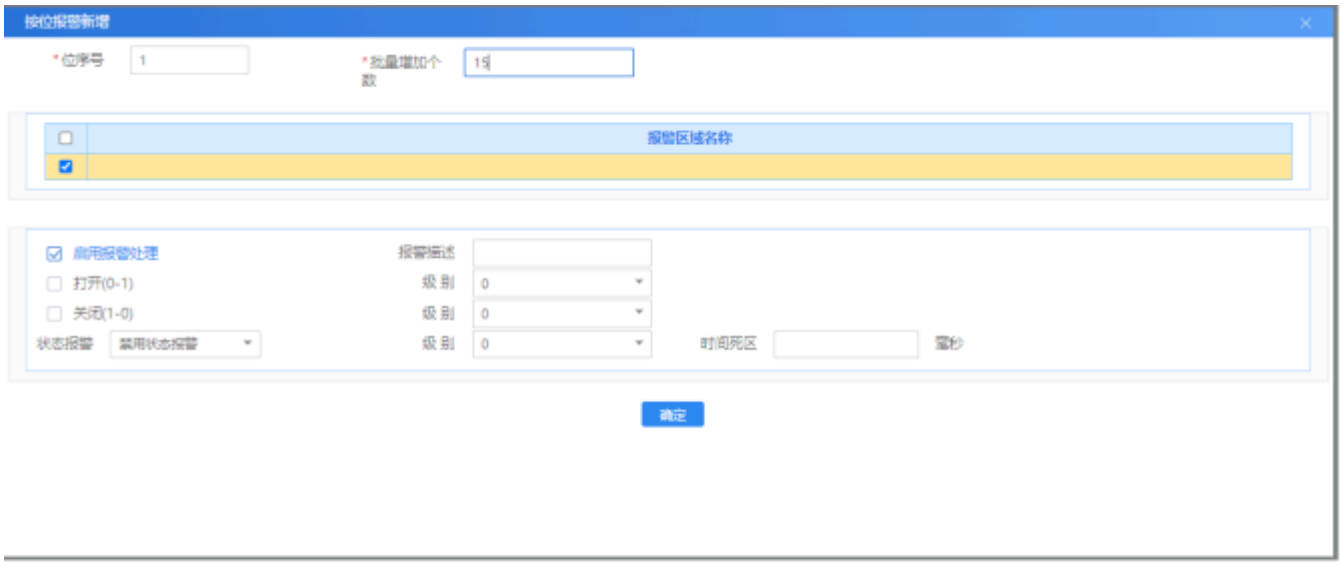
配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

2.AI的按位报警属性如下图所示

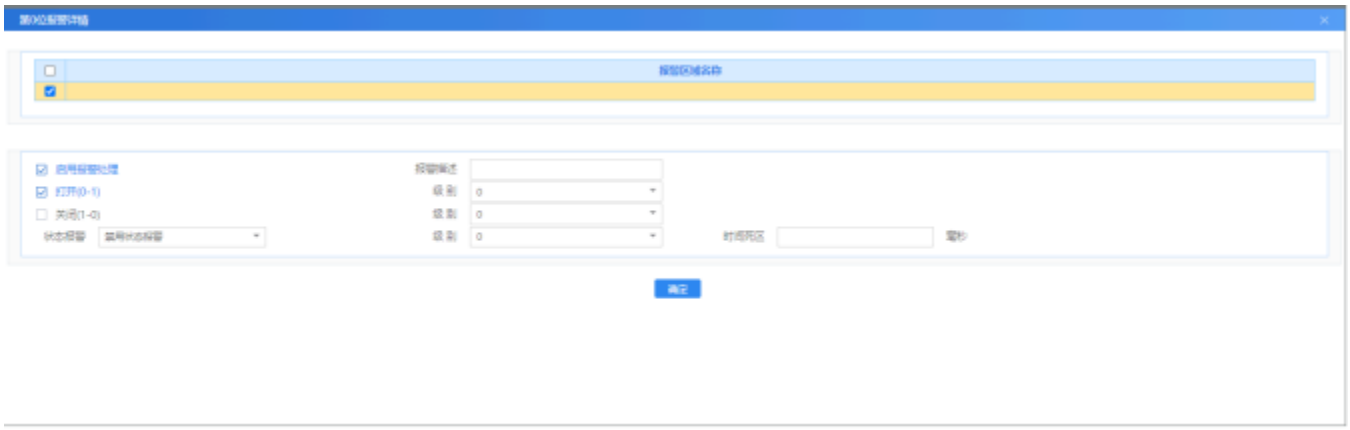


按位报警配置图1

•启用报警处理：用户如果需要对变量进行报警处理，需要勾选"启用报警处理"选项。
•点击【新增】按钮，弹出图2【按位报警新增】界面，新增按位报警
•双击已经配置的一列，弹出图3【第n位报警详情】界面，修改这一位的报警配置



按位报警配置图2



按位报警配置图3

- 报警区域名称：为了便于用户监控报警时筛选，我们提供了报警区域配置，具体参考报警区域配置
- 报警设置：
  - 启用报警处理：属于状态报警类型选择
    - 禁用状态报警：禁用此功能将不产生状态报警
    - 状态0报警：当前值为0时报警
    - 状态1报警：当前值为1时报警
  - 打开/关闭报警：属于变位报警类型选择，支持多选
    - "打开"报警：当前值从0变为1时触发报警
    - "关闭"报警：当前值从1变为0时触发报警
  - 报警时间死区：对启用报警处理的"状态报警"有效，当前值到达报警触发条件，并保持一段时间（报警时间死区设定值），才触发报警，否则不触发报警
- 报警描述：此BIT代表的报警含义的文字描述
- 级别：包括报警级别和事件级别，数字越小，级别越高，级别为0的选项设置，级别最高

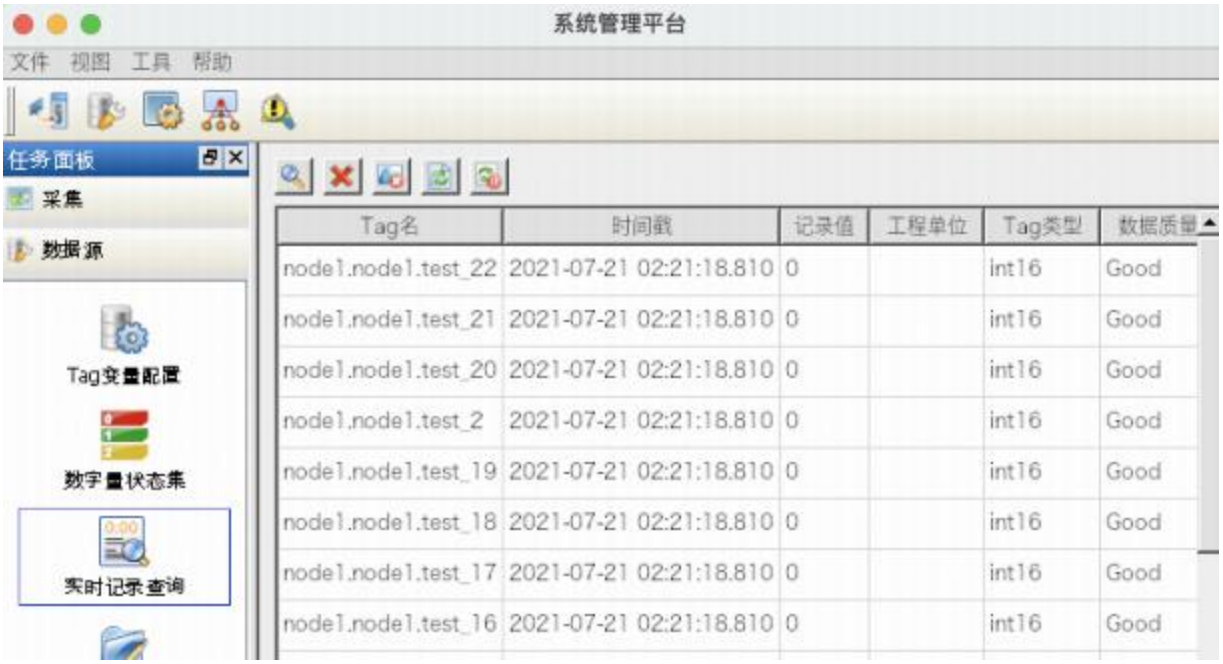
补充

配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

配置AI变量转储属性

•AI的转储属性如下图所示：

- IHD：勾选后，iplatda系统会将该点转储到目标ihd中
- 压缩误差：压缩可接受的误差值范围，例如：填"0.1"，即变化误差为0.1的值会被压缩。DI类型数据，此项请填"0"，系统会自动对DI类型数据进行压缩。
- 扫描周期：两次扫描周期所间隔的时间
- 压缩超时：超过该时间范围强制记录历史数据
- IHD(v3)点的名称中包含SCADA名称：勾选IHD(v3)点的名称中包含SCADA名称后，转出到ihd的中点的名字包含SCADA节点名称
- 变化时转储：勾选变化时转储，则当SCADA点变化是才会转储到IHD，转储后结果如图所示：



- 只转储GOOD数据：勾选后iplatda系统向ihd转出数据时会过滤掉质量不为Good的数据
- iFD：勾选后，iplatda系统会将该点转储到目标influxdb中，具体配置参考转储数据到时序数据库
- OPCUA发布：勾选后iplatda系统会将数据以opcua的形式进行发布，可以用opcua的客户端进行查看
- OPC发布：选后iplatda系统会将数据以opc的形式进行发布，可以用opc的客户端进行查看

模拟量输入AI的属性表

•变量属性名称以"A_"开头的为字符型变量；变量属性名称以"F_"开头的为数值型变量；

模拟输入量属性

属性名称	类型（长度）	
A_NAME	DT_ASCII	Readonly。127字节，由中文字符、字母、数字、_字符
A_DESC	DT_ASCII	127字节
F_BLKTYPE	DT_UCHAR	DisplayOnly PB_AI=1
F_TAGID	DT_SLONG	Readonly,
属性名称	**类型（长度）**	
F_ISTREND	DT_SINT16	1:保存历史趋势，0不保存历史趋势
F_COMPTYPE	DT_SINT16	保存历史数据后的配置属性
F_COMPDEV	DT_SINT16	保存历史数据后的配置属性
F_COMPMAXTIME	DT_SLONG	保存历史数据后的配置属性
A_ALM_AREA0	DT_SINT16	-1=无报警区；-2="ALL"报警区，只可能为报警区0的ID；

A_ALM_AREA1	DT_SINT16	
A_ALM_AREA2	DT_SINT16	
A_ALM_AREA3	DT_SINT16	
A_ALM_AREA4	DT_SINT16	
A_ALM_AREA5	DT_SINT16	
A_SEC_AREA0	DT_ASCII	每个安全区最长31字节。
A_SEC_AREA1	DT_ASCII	
A_SEC_AREA2	DT_ASCII	
A_SEC_AREA3	DT_ASCII	
A_SEC_AREA4	DT_ASCII	
A_SEC_AREA5	DT_ASCII	
A_BLK_SN	DT_SLONG	
A_EXT_PARAM0	DT_ASCII	63字节
A_EXT_PARAM1	DT_ASCII	
A_EXT_PARAM2	DT_ASCII	
A_EXT_PARAM3	DT_ASCII	
A_SCAN_MODE	DT_ASCII	1=扫描机制；0=通知机制

属性名称	类型（长度）	
F_SCAN_MODE	DT_BIT	
F_SCAN_INTV	DT_ULONG	毫秒数
F_SCAN_PHS	DT_ULONG	毫秒数
A_IODRV	DT_ASCII	63字节
A_IOADDR	DT_ASCII	63字节
A_EGU_LBL	DT_ASCII	15字节
F_EGU_LOW	DT_DBL	浮点数
F_EGU_SPAN	DT_DBL	浮点数
F_RAW_LOW	DT_DBL	浮点数
F_RAW_SPAN	DT_DBL	浮点数
F_EGU_ENABLE	DT_BIT	1=允许；0=禁止
A_DATA_TYPE	DT_ASCII	
F_DATA_TYPE	DT_UCHAR	
F_SCAN_ENABLE	DT_BIT	1/0
F_OVERRIDE_ENABLE	DT_BIT	1/0
F_OVERRIDE_VALUE	DT_DBL	
F_ZP_ENABLE	DT_BIT	1/0
F_ZP_CENT	DT_DBL	
A_ZP_THRES	DT_DBL	
F_ALM_ENABLE	DT_BIT	1/0
F_ALM_V_DEADBAND	DT_DBL	
F_ALM_TM_DEADBAND	DT_ULONG	毫秒数
F_ALM_LL_ENABLE	DT_BIT	
F_ALM_LL_THRESH	DT_DBL	浮点数，位于工程下限和工程上限之间，下同。 LL,L,H,

F_ALM_L_ENABLE	DT_BIT	
F_ALM_L_THRESH	DT_DBL	
F_ALM_H_ENABLE	DT_BIT	
F_ALM_H_THRESH	DT_DBL	
F_ALM_HH_ENABLE	DT_BIT	
F_ALM_HH_THRESH	DT_DBL	
F_ALM_DEV_ENABLE	DT_BIT	
F_ALM_ROC_ENABLE	DT_BIT	
F_ALM_DEV_THRESH	DT_DBL	
F_ALM_DEV_CENT	DT_DBL	
F_ALM_ROC_THRESH	DT_DBL	
F_ALM_HH_PRIOR	DT_UCHAR	0～255
F_ALM_H_PRIOR	DT_UCHAR	0～255
F_ALM_L_PRIOR	DT_UCHAR	0～255
F_ALM_LL_PRIOR	DT_UCHAR	0～255
F_ALM_DEV_PRIOR	DT_UCHAR	0～255
F_ALM_ROC_PRIOR	DT_UCHAR	0～255
F_ALM_HH_STATE	DT_UCHAR	0=无报警/报警已恢复已确认；1=报警未恢复已确认；2=
F_ALM_H_STATE	DT_UCHAR	
F_ALM_L_STATE	DT_UCHAR	
属性名称	**类型（长度）**	
F_ALM_LL_STATE	DT_UCHAR	
F_ALM_DEV_STATE	DT_UCHAR	
F_ALM_ROC_STATE	DT_UCHAR	
F_OUTPUT_ENABLE	DT_BIT	1/0

F_VTQ	DT_ASCII	值、时间戳和数据质量
A_CV	DT_VTQ	
F_CV	DT_VTQ	
A_TIMESTAMP	DT_ASCII	YYYY-MM-DD HH:mm:ss.xxx
F_TIMESTAMP	DT_CV_TIME	64BIT,TCV_TimeStamp，使用接口读取后，转化成当前
A_QUALITY	DT_ASCII	兼容OPC Quality
F_QUALITY	DT_UINT16	
A_QUALITY_SUB	DT_ASCII	
F_QUALITY_SUB	DT_UINT16	
A_ALM_PIC	DT_ASCII	32字节，该点所有报警
F_ACK_ALM_TYPE	DT_UINT16	WriteOnly，写入待确认报警类型
F_DEL_ALM_TYPE	DT_UINT16	WriteOnly，写入待删除报警类型
F_ALM_CNT	DT_SLONG	报警状态为1或3
F_ALM_UNACK_CNT	DT_SLONG	报警状态为2或3
F_ALM_ALL_CNT	DT_SLONG	只要报警状态不为0

- 配置DI变量基本属性
- 配置DI变量报警属性
- 配置DI变量转储属性
- 数字量输入DI的属性表

3.9 配置DI变量基本属性

1. DI的属性如下图所示：

注意：如果用户没有配置"所属驱动"、"设备名称"和"设备地址"，则添加的变量默认为全局变量，全局变量可以在所有客户端中通用。

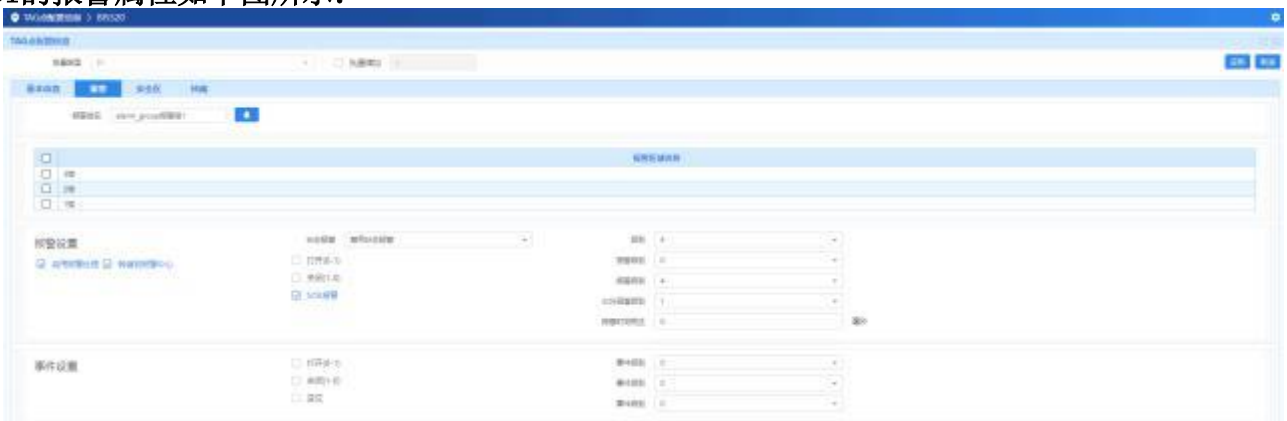
- 扫描设置：
  - 通知机制：指变量值有变化的时候进行通知读取
  - 扫描机制：通过轮循扫描的方式对变量进行读取，可对"扫描周期"和"扫描相位"进行设置。这里的扫描周期是过程数据库读取驱动的时间
    - 扫描周期：两次扫描周期所间隔的时间
    - 扫描相位：用于设置起始扫描时间。避免多组同时扫描造成的资源不足风险

注意：为了避免丢点现象，如果用户知道设备的数据变化周期，建议将驱动的刷新周期配置为设备的数据变化周期的一半，点的扫描周期配置为数据变化周期的4分之一，如设备的数据变化周期为4000毫秒，那么驱动刷新周期为2000毫秒，点的扫描周期为1000毫秒。

- 启动选项：
  - 允许控制：用户如果要对变量进行写控制，需要勾选"允许控制"，ipIat提供了"允许扫描"和"禁止扫描"的选项，方便用户设置是否启用扫描机制
  - 记录控制事件：当我们对过程量进行写值控制的时候，用于记录控制的事件，此功能在"允许控制"被勾选的情况下可用，并且可以设定事件级别，方便查找时筛选
- 标注：设定"开""关"状态时所显示的标注，用户可选入下拉菜单中的状态选项
- 保存历史数据：ipIatda系统会将这个点的历史数据保存下来，用来绘制趋势图、脚本查询历史数据等
- 压缩类型：可选（不压缩，压缩，对历史数据值惊醒压缩）
- 压缩误差：压缩可接受的误差值范围；例如：填"0.1"，即变化误差为0.1的值会被压缩。DI类型数据，此项请填"0"，系统会自动对DI类型数据进行压缩。
- 压缩超时：超过该时间范围强制记录历史数据。
- 扩展参数：存储类的属性的扩展信息，该扩展信息可完全由用户自定义

配置DI变量报警属性

1. **DI的报警属性如下图所示：**



- 报警区域名称：为了便于用户监控报警时筛选，我们提供了报警区域配置，具体参考报警区域配置
- 报警设置：
  - 启用报警处理：属于状态报警类型选择
    - 禁用状态报警：禁用此功能将不产生状态报警
    - 状态0报警：当前值为0时报警
    - 状态1报警：当前值为1时报警
  - 打开/关闭报警：属于变位报警类型选择，支持多选
    - "打开"报警：当前值从0变为1时触发报警

.″关闭″报警：当前值从1变为0时触发报警

。SOE(Sequence Of Event)报警(通常用于104驱动SOE点的报警展示)：

.SOE报警：只要该点值或时间戳发生了更新(即驱动收到了对应SOE报文)就会触发该报警

.SOE报警级别：包括报警级别和事件级别，数字越小，级别越高，级别为0的选项设置，级别最高SOE报警

。报警时间死区：对启用报警处理的″状态报警″有效，当前值到达报警触发条件，并保持一段时间（报警时间死区设定值），才触发报警，否则不触发报警

•事件设置：

◦″打开″事件：当前值从0变为1时，记录设备变位事件

◦″关闭″事件：当前值从1变为0时，记录设备变位事件

◦″变位″事件：当前值发生变位，从0变为1时或从1变为0时均记录设备变位事件

•级别：包括报警级别和事件级别，数字越小，级别越高，级别为0的选项设置，级别最高

补充

配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

配置DI变量转储属性

•DI变量转储设置基本同AI变量，参考AI变量转储属性

数字量输入DI的属性表

•变量属性名称以″A_″开头的为字符型变量；变量属性名称以″F_″开头的为数值型变量数字输入量属性

属性名称	类型（长度）	
A_NAME	DT_ASCII	Readonly。127字节，由中文字符、字母、数字、‘,’、‘_’字
A_DESC	DT_ASCII	127字节
F_BLKTYPE	DT_UCHAR	DisplayOnly PB_DI=5
F_TAGID	DT_SLONG	Readonly,
F_ISTREND	DT_SINT16	1:保存历史趋势，0不保存历史趋势
F_COMPTYPE	DT_SINT16	保存历史数据后的配置属性
F_COMPDEV	DT_SINT16	保存历史数据后的配置属性

F_COMPMAXTIME	DT_SLONG	保存历史数据后的配置属性
A_ALM_AREA0	DT_SINT16	-1=无报警区；-2="ALL"报警区，只可能为报警区0的I
A_ALM_AREA1	DT_SINT16	
A_ALM_AREA2	DT_SINT16	
A_ALM_AREA3	DT_SINT16	
A_ALM_AREA4	DT_SINT16	
A_ALM_AREA5	DT_SINT16	
A_SEC_AREA0	DT_ASCII	每个安全区最长31字节
A_SEC_AREA1	DT_ASCII	
A_SEC_AREA2	DT_ASCII	
A_SEC_AREA3	DT_ASCII	
A_SEC_AREA4	DT_ASCII	
A_SEC_AREA5	DT_ASCII	
A_BLK_SN	DT_SLONG	
A_EXT_PARAM0	DT_ASCII	63字节
A_EXT_PARAM1	DT_ASCII	
A_EXT_PARAM2	DT_ASCII	
A_EXT_PARAM3	DT_ASCII	
A_SCAN_MODE	DT_ASCII	1=扫描机制；0=通知机制
属性名称	**类型（长度）**	
F_SCAN_MODE	DT_BIT	
F_SCAN_INTV	DT_ULONG	毫秒数
F_SCAN_PHS	DT_ULONG	毫秒数
A_IODRV	DT_ASCII	63字节
A_IOADDR	DT_ASCII	63字节

A_EGU_OPEN	DT_ASCII	31字节
A_EGU_CLOSE	DT_ASCII	31字节
F_SCAN_ENABLE	DT_BIT	1/0
F_SOE_ENABLE	DT_BIT	1/0
F_EVENT_ENABLE	DT_BIT	1/0
F_ALM_ENABLE	DT_BIT	1/0
F_ALM_OPEN	DT_BIT	1/0
F_ALM_CLOSE	DT_BIT	1/0
F_ALM_KEEPON	DT_BIT	1/0
F_ALM_KEEPOFF	DT_BIT	1/0
F_ALM_OPEN_PRIORITY	DT_BIT	1/0
F_ALM_CLOSE_PRIORITY	DT_BIT	1/0
F_EVT_OPEN	DT_BIT	1/0
F_EVT_CLOSE	DT_BIT	1/0
F_EVT_CHG	DT_BIT	1/0
F_EVT_CTRL	DT_BIT	1/0
F_EVT_OPEN_PRIORITY	DT_UCHAR	0～255
F_EVT_CLOSE_PRIORITY	DT_UCHAR	0～255
F_EVT_CHG_PRIORITY	DT_UCHAR	0～255
F_EVT_CTRL_PRIORITY	DT_UCHAR	0～255
F_ALM_TDB	DT_ULONG	毫秒数
F_OUTPUT_ENABLE	DT_BIT	1/0
F_OVERRIDE_ENABLE	DT_BIT	1/0
F_OVERRIDE_VALUE	DT_DBL	
F_ALM_OPEN_STATE	DT_UCHAR	0：无报警/报警已恢复已确认；1：报警未恢复已确认

F_ALM_CLOSE_STATE	DT_UCHAR	0：无报警/报警已恢复已确认；1：报警未恢复已确认
F_ALM_KEEPON_STATE	DT_UCHAR	
F_ALM_KEEPOFF_STATE	DT_UCHAR	
F_VTQ	DT_ASCII	值、时间戳和数据质量
A_CV	DT_VTQ	
F_CV	DT_VTQ	
A_TIMESTAMP	DT_ASCII	YYYY-MM-DD HH:mm:ss.xxx
F_TIMESTAMP	DT_CV_TIME	64BIT,TCV_TimeStamp，使用接口读取后，转化成当
A_QUALITY	DT_ASCII	兼容OPC Quality
F_QUALITY	DT_UINT16	
A_QUALITY_SUB	DT_ASCII	
F_QUALITY_SUB	DT_UINT16	
A_ALM_PIC	DT_ASCII	32字节，该点所有报警
F_ACK_ALM_TYPE	DT_UINT16	WriteOnly，写入待确认报警类型
F_DEL_ALM_TYPE	DT_UINT16	WriteOnly，写入待删除报警类型
F_ALM_CNT	DT_SLONG	报警状态为2或3
F_ALM_UNACK_CNT	DT_SLONG	报警状态为1或3
F_ALM_ALL_CNT	DT_SLONG	只要报警状态不为0

- 模拟量输出AO
  - 配置AO变量基本属性
  - 配置AO变量报警属性
  - 配置AO变量转储属性

- 模拟量输出AO的属性表
- 数字量输出DO
  - 配置DO变量基本属性
  - 配置DO变量报警属性
  - 配置DO变量转储属性
  - 数字量输出DO的属性表
- 二进制大对象量BLOB
  - 配置BLOB变量基本属性
- 配置BLOB变量报警属性
- 配置BLOB变量转储属性
- 二进制大对象量BLOB的属性表
- 文本量TXT
  - 配置TXT变量基本属性
  - 配置TXT变量报警属性
  - 配置TXT变量转储属性
  - 文本量TXT的属性表
- 模拟计算量CA
  - 配置CA变量基本属性
  - 配置CA变量报警属性
  - 配置CA变量转储属性
  - 模拟计算量CA的属性表
- 数字计算量CD
  - 配置CD变量基本属性
  - 配置CD变量报警属性
  - 配置CD变量转储属性
  - 数字计算量CD的属性表
- 分级报警量LA
  - 配置LA变量基本属性
- 配置LA变量报警属性
- 配置LA变量转储属性
- 分级报警量LA的属性表
- 多数字量输入MDI
- 配置MDI变量基本属性
- 配置MDI变量报警属性
- 配置MDI变量转储属性
- 配置MDI变量编址属性
- 多数字量输入MDI的属性表

3.10 模拟量输出AO

配置AO变量基本属性

1. AO的属性如下图所示：

注意：如果用户没有配置"所属驱动"、"设备名称"和"设备地址"，则添加的变量默认为全局变量，全局变量可以在所有客户端中通用。

•数据类型：

数据类型	取值范围	
16位无符号整型	0~65535	
16位有符号整型	-32768~32767	
32位无符号整型	0~4294967295	
32位有符号整型	-2147483648~2147483647	
8位无符号整型	0~255	
8位有符号整型	-128~127	
单精度浮点型	-2147483648~2147483647	32位的4个字节，这里驱动数据块的寄存器也需
双精度浮点型	-2147483648~2147483647	64位的8个字节；小数点后不能超过6位有效数字

•字节顺序：字节顺序用来解决设备返回数据的字节顺序与iplat读取数据字节顺序不一致的情况，例如16位无符号整型258，在x86结构机器上其16进制表示为0x0102，但是从设备读取该数据时其顺序可能为0x0201，此时需要作"21"转换，即将第一个字节与第二个字节进行交换，从而使得从设备读取的数据可以在终端上正确显示，通常规情况中，用户不用修改选项。

•启动选项：

◦允许设置初值：用户可以通过勾选"允许设置初值"为变量设置初始值，此时若启用量程转换复选框未勾选，则初值的范围应在所选数据类型范围之内。若启用量程转换复选框

勾选，则初值取值范围应在零量程与满量程之间。

◦允许控制：用户如果要对变量进行写控制，需要勾选"允许控制"。

◦记录控制事件：当我们对过程量进行写值控制的时候，用于记录控制的事件，此功能在"允许控制"被勾选的情况下可用，并且可以设定事件级别，方便查找时筛选。

•启用量程转换：

◦单位：变量值的单位可通过"单位"下拉框进行设定，软件提供了米、分、秒、千克、克、米/秒、公里/小时、摄氏度、华氏度等常见单位选项，也可自行输入单位(为兼容老版本，数字数字1到10导入后会根据老版本对应关系进行展示"1"对应"米"，"2"对应"分"，"3"对应"秒"，"4"对应"千克"，"5"对应"克"，"6"对应"米/秒"，"7"对应"公里/小时"，"8"对应"摄氏度"，"9"对应"华氏度"，"10"对应空值)。"原始量"根据实际需求配置。选择"启用量程转换"复选框，可对"工程量"中的"零量程"与"满量程"进行设置。如上图所示，原始量的测量范围是12分到10000分之间，所有小于12分值转换为12，所有大于10000分的值转换为500，在工程应用是，装换为12到500之间。

◦原始量：从驱动获取到的原始值。

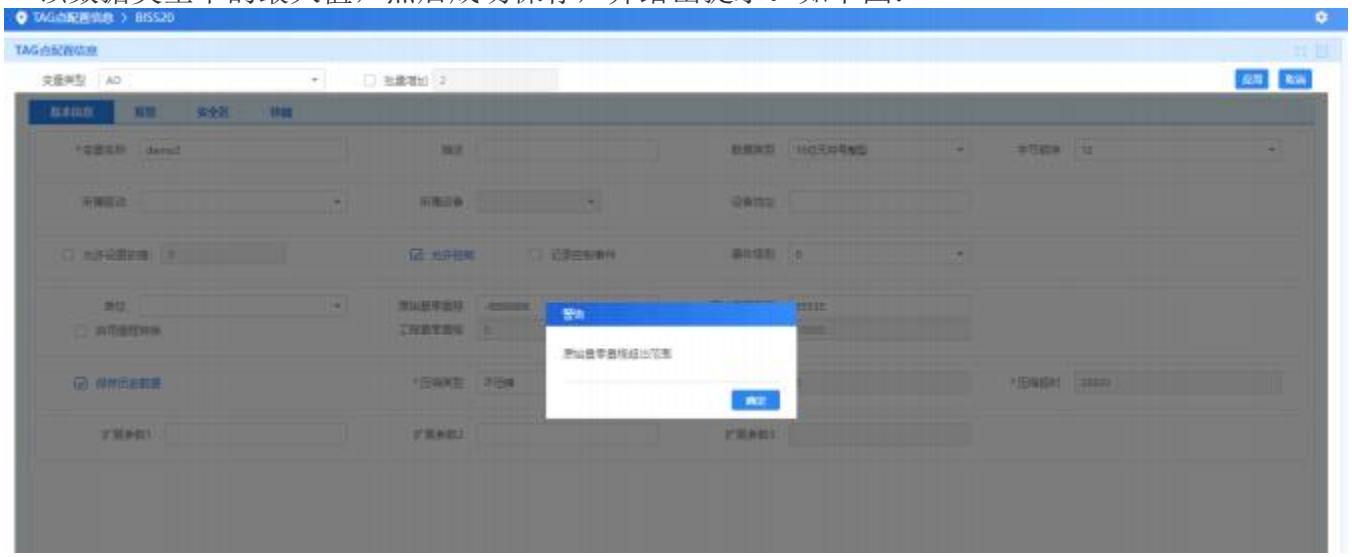
◦工程量：转换工程单位后的值

◦零量程：测量范围的下限值

◦满量程：测量范围的上限值

◦量程转换：量程是测量范围上限值与下限值之差，而从一个量程变换成另一个量程就叫量程转换（支持小数格式）

•注意：原始量的零量程和满量程值，应在所选的数据类型的范围内（16位无符号整型：065535；16位有符号整型：-3276832767；32位无符号整型：04294967295；32位有符号整型：-21474836482147483647；8位无符号整型：0255；8位有符号整型：-128127；单精度浮点型：-21474836482147483647，但小数点后不能超过6位有效数字；双精度浮点型：-21474836482147483647，但小数点后不能超过6位有效数字）。当量程超限时会自动修正为该数据类型下的最大值，然后成功保存，并给出提示。如下图：



•保存历史数据：iplatda系统会将这个点的历史数据保存下来，用来绘制趋势图、脚本查询历史数据等

•扩展参数：存储类的属性的扩展信息，该扩展信息可完全由用户自定义

配置AO变量报警属性

•AO的报警属性如下图所示：

补充

配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

配置AO变量转储属性

- AO变量转储设置基本同AI变量，参考AI变量转储属性。

模拟量输出AO的属性表

- 变量属性名称以"A_"开头的为字符型变量；变量属性名称以"F_"开头的为数值型变量；

模拟输出量属性

属性名称	类型（长度）	
A_NAME	DT_ASCII	Readonly。127字节，由中文字符、字母、数字、’_’字符，’
A_DESC	DT_ASCII	127字节
F_BLKTYPE	DT_UCHAR	DisplayOnly PB_AO=2
F_TAGID	DT_SLONG	Readonly,
F_ISTREND	DT_SINT16	1:保存历史趋势，0不保存历史趋势
F_COMPTYPE	DT_SINT16	保存历史数据后的配置属性
F_COMPDEV	DT_SINT16	保存历史数据后的配置属性
F_COMPMAXTIME	DT_SLONG	保存历史数据后的配置属性
A_ALM_AREA0	DT_SINT16	-1=无报警区-2="ALL"报警区，只可能为报警区0的ID1～327
A_ALM_AREA1	DT_SINT16	
A_ALM_AREA2	DT_SINT16	

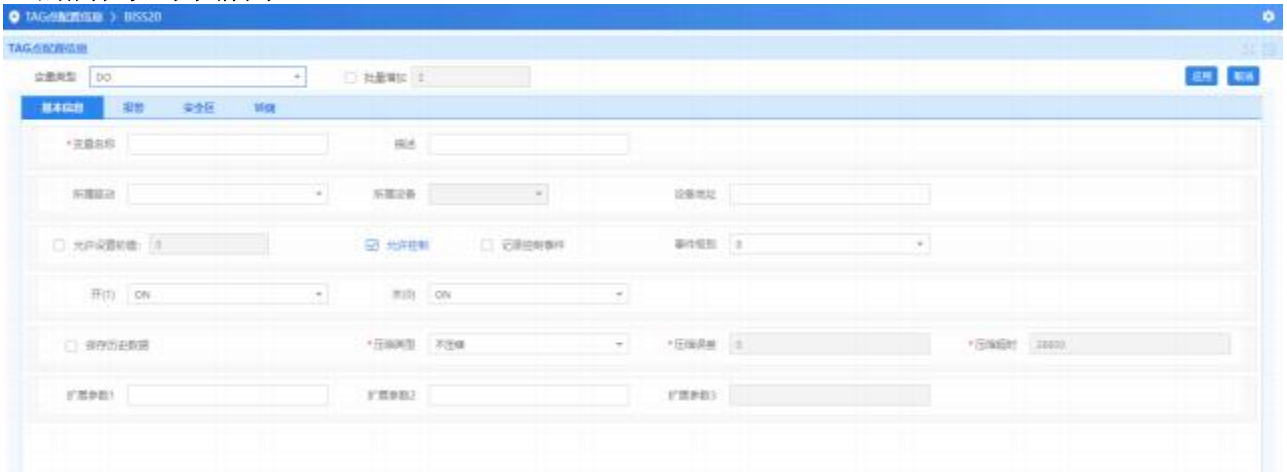
A_ALM_AREA3	DT_SINT16	
A_ALM_AREA4	DT_SINT16	
A_ALM_AREA5	DT_SINT16	
属性名称	**类型（长度）**	
A_SEC_AREA0	DT_ASCII	每个安全区最长31字节
A_SEC_AREA1	DT_ASCII	
A_SEC_AREA2	DT_ASCII	
A_SEC_AREA3	DT_ASCII	
A_SEC_AREA4	DT_ASCII	
A_SEC_AREA5	DT_ASCII	
A_BLK_SN	DT_SLONG	
A_EXT_PARAM0	DT_ASCII	63字节
A_EXT_PARAM1	DT_ASCII	
A_EXT_PARAM2	DT_ASCII	
A_EXT_PARAM3	DT_ASCII	
A_IODRV	DT_ASCII	63字节
A_IOADDR	DT_ASCII	63字节
A_EGU_LBL	DT_ASCII	15字节
F_EGU_LOW	DT_DBL	浮点数
F_EGU_SPAN	DT_DBL	浮点数
F_RAW_LOW	DT_DBL	浮点数
F_RAW_SPAN	DT_DBL	浮点数
F_EGU_ENABLE	DT_BIT	1=允许0=禁止
A_DATA_TYPE	DT_ASCII	
F_DATA_TYPE	DT_UCHAR	

F_INIT_VALUE	DT_DBL	
F_INITVAL_ENABLE	DT_BIT	1/0
F_EVT_ENABLE	DT_BIT	1/0
F_EVT_PRIORITY	DT_UCHAR	0～255
F_OUTPUT_ENABLE	DT_BIT	1/0
F_VTQ	DT_ASCII	值、时间戳和数据质量
A_CV	DT_VTQ	
F_CV	DT_VTQ	
A_TIMESTAMP	DT_ASCII	YYYY-MM-DD HH:mm:ss.xxx
F_TIMESTAMP	DT_CV_TIME	64BIT,TCV_TimeStamp，使用接口读取后，转化成当前时
A_QUALITY	DT_ASCII	兼容OPC Quality
F_QUALITY	DT_UINT16	
A_QUALITY_SUB	DT_ASCII	
F_QUALITY_SUB	DT_UINT16	

3.11 数字量输出DO

配置DO变量基本属性

1. **DO的属性如下图所示：**



注意：如果用户没有配置"所属驱动"、"设备名称"和"设备地址"，则添加的变量默认为全局变量，全局变量可以在所有客户端中通用。

•启动选项：

- 允许设置初值：用户可以通过勾选"允许设置初值"为变量设置初始值。
- 允许控制：用户如果要对变量进行写控制，需要勾选"允许控制"。
- 记录控制事件：当我们对过程量进行写值控制的时候，用于记录控制的事件，此功能在"允许控制"被勾选的情况下可用，并且可以设定事件级别，方便查找时筛选。

•**标注：**设定"开""关"状态时所显示的标注，用户可选入已有项中一项或者自行输入添加一项。

•**保存历史数据：**iplatda系统会将这个点的历史数据保存下来，用来绘制趋势图、脚本查询历史数据等。

•**扩展参数：**存储类的属性的扩展信息，该扩展信息可完全由用户自定义。

配置DO变量报警属性

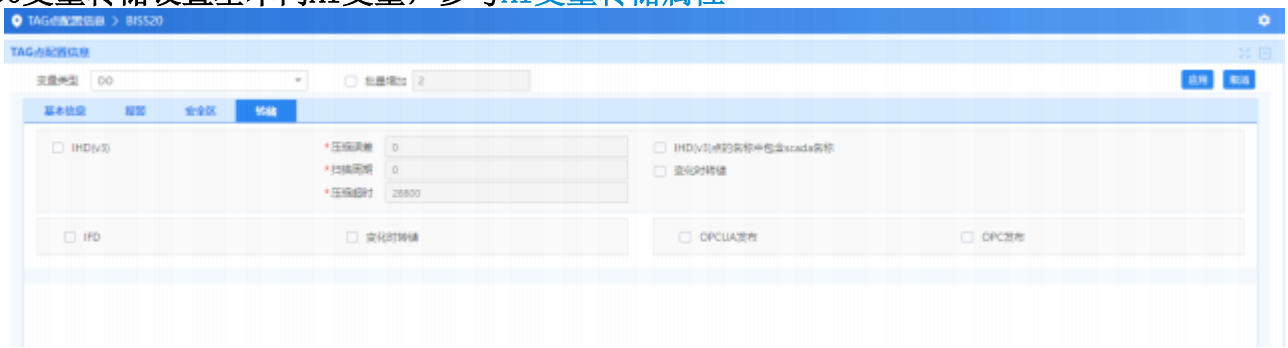
•**DO变量报警属性如下：**

补充

配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

配置DO变量转储属性

- DO变量转储设置基本同AI变量，参考AI变量转储属性



数字量输出DO的属性表

- 变量属性名称以"A_"开头的为字符型变量；变量属性名称以"F_"开头的为数值型变量数字输出量属性

属性名称	类型（长度）	
A_NAME	DT_ASCII	Readonly。127字节，由中文字符、字母、数字、'_'字符，'
A_DESC	DT_ASCII	127字节
F_BLKTYPE	DT_UCHAR	DisplayOnly PB_DO=6
F_TAGID	DT_SLONG	Readonly,
F_ISTREND	DT_SINT16	1:保存历史趋势，0不保存历史趋势
F_COMPTYPE	DT_SINT16	保存历史数据后的配置属性
F_COMPDEV	DT_SINT16	保存历史数据后的配置属性
F_COMPMAXTIME	DT_SLONG	保存历史数据后的配置属性
A_ALM_AREA0	DT_SINT16	-1=无报警区；-2="ALL"报警区，只可能为报警区0的ID；1~
A_ALM_AREA1	DT_SINT16	
A_ALM_AREA2	DT_SINT16	
A_ALM_AREA3	DT_SINT16	
属性名称	类型（长度）	
A_ALM_AREA4	DT_SINT16	
A_ALM_AREA5	DT_SINT16	
A_SEC_AREA0	DT_ASCII	每个安全区最长31字节

A_SEC_AREA1	DT_ASCII	
A_SEC_AREA2	DT_ASCII	
A_SEC_AREA3	DT_ASCII	
A_SEC_AREA4	DT_ASCII	
A_SEC_AREA5	DT_ASCII	
A_BLK_SN	DT_SLONG	
A_EXT_PARAM0	DT_ASCII	63字节
A_EXT_PARAM1	DT_ASCII	
A_EXT_PARAM2	DT_ASCII	
A_EXT_PARAM3	DT_ASCII	
A_IODRV	DT_ASCII	63字节
A_IOADDR	DT_ASCII	63字节
A_EGU_OPEN	DT_ASCII	31字节
A_EGU_CLOSE	DT_ASCII	31字节
F_INIT_VALUE	DT_ASCII	
F_INITVAL_ENABLE	DT_BIT	1/0
F_SCAN_ENABLE	DT_BIT	1/0
F_EVT_ENABLE	DT_BIT	1/0
F_EVT_PRIORITY	DT_UCHAR	0~255
属性名称	**类型（长度）**	
F_OUTPUT_ENABLE	DT_BIT	1/0
A_CV	DT_VTQ	
F_CV	DT_VTQ	
A_TIMESTAMP	DT_ASCII	YYYY-MM-DD HH:mm:ss.xxx
F_TIMESTAMP	DT_CV_TIME	64BIT,TCV_TimeStamp，使用接口读取后，转化成当前时

A_QUALITY	DT_ASCII	兼容OPC Quality
F_QUALITY	DT_UINT16	
A_QUALITY_SUB	DT_ASCII	
F_QUALITY_SUB	DT_UINT16	

3.12 二进制大对象量BLOB

配置BLOB变量基本属性

1. **BLOB的属性如下图所示：**

- 扫描设置：
- 通知机制：指变量值有变化的时候进行通知读取
- 扫描机制：通过轮循扫描的方式对变量进行读取，可对"扫描周期"和"扫描相位"进行设置。
- 扫描周期：两次扫描周期所间隔的时间
- 扫描相位：用于设置起始扫描时间。避免多组同时扫描造成的资源不足风险

注意：
– 为了避免丢点现象，如果用户知道设备的数据变化周期，建议将驱动的刷新周期配置为设备的数据变化周
-Blob变量配置长度应小于或等于实际长度，反之，数据无法推送！

- 启动选项：
  - 允许控制：用户如果要对变量进行写控制，需要勾选"允许控制"，iplat提供了"允许扫描"和"禁止扫描"的选项，方便用户设置是否启用扫描机制
  - 记录控制事件：当我们对过程量进行写值控制的时候，用于记录控制的事件，此功能在"允许控制"被勾选的情况下可用，并且可以设定事件级别，方便查找时筛选
- 保存历史数据：iplatda系统会将这个点的历史数据保存下来，用来绘制趋势图、脚本查询历史数据等
- 扩展参数：存储类的属性的扩展信息，该扩展信息可完全由用户自定义

3.13 配置BLOB变量报警属性

- **BLOB变量报警属性如下：**

补充
配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

配置BLOB变量转储属性

•BLOB变量转储设置基本同AI变量，参考AI变量转储属性

二进制大对象量BLOB的属性表

•变量属性名称以"A_"开头的为字符型变量；变量属性名称以"F_"开头的为数值型变量BLOB量属性

属性名称	类型（长度）	输入规则
A_NAME	DT_ASCII	Readonly。127字节，由中文字符、字母、数字、'_'字符，
A_DESC	DT_ASCII	127字节
F_BLKTYPE	DT_UCHAR	DisplayOnly PB_BLOB=3
F_TAGID	DT_SLONG	Readonly,
F_ISTREND	DT_SINT16	1:保存历史趋势，0不保存历史趋势
F_COMPTYPE	DT_SINT16	保存历史数据后的配置属性
F_COMPDEV	DT_SINT16	保存历史数据后的配置属性
F_COMPMAXTIME	DT_SLONG	保存历史数据后的配置属性
A_ALM_AREA0	DT_SINT16	-1=无报警区；-2="ALL"报警区，只可能为报警区0的ID；1~
A_ALM_AREA1	DT_SINT16	

A_ALM_AREA2	DT_SINT16	
A_ALM_AREA3	DT_SINT16	
A_ALM_AREA4	DT_SINT16	
A_ALM_AREA5	DT_SINT16	
A_SEC_AREA0	DT_ASCII	每个安全区最长31字节
A_SEC_AREA1	DT_ASCII	
A_SEC_AREA2	DT_ASCII	
A_SEC_AREA3	DT_ASCII	
A_SEC_AREA4	DT_ASCII	
A_SEC_AREA5	DT_ASCII	
A_BLK_SN	DT_SLONG	
A_EXT_PARAM0	DT_ASCII	63字节
A_EXT_PARAM1	DT_ASCII	
A_EXT_PARAM2	DT_ASCII	
A_EXT_PARAM3	DT_ASCII	
A_SCAN_MODE	DT_ASCII	1=扫描机制0=通知机制
F_SCAN_MODE	DT_BIT	
F_SCAN_INTV	DT_ULONG	毫秒数
F_SCAN_PHS	DT_ULONG	毫秒数
A_IODRV	DT_ASCII	63字节
A_IOADDR	DT_ASCII	63字节
F_SCAN_ENABLE	DT_BIT	1/0
F_EVT_ENABLE	DT_BIT	1/0
F_EVT_PRIORITY	DT_UCHAR	0~255
F_OUTPUT_ENABLE	DT_BIT	1/0

F_VTQ	DT_ASCII	值、时间戳和数据质量
A_CV	DT_VTQ	
F_CV	DT_VTQ	
A_TIMESTAMP	DT_ASCII	YYYY-MM-DD HH:mm:ss.xxx
F_TIMESTAMP	DT_CV_TIME	64BIT,TCV_TimeStamp，使用接口读取后，转化成当前时
A_QUALITY	DT_ASCII	兼容OPC Quality
F_QUALITY	DT_UINT16	
A_QUALITY_SUB	DT_ASCII	
F_QUALITY_SUB	DT_UINT16	

3.14 文本量TXT

配置TXT变量基本属性

1. TXT的属性如下图所示：

•扫描设置：

◦通知机制：指变量值有变化的时候进行通知读取

◦扫描机制：通过轮循扫描的方式对变量进行读取，可对"扫描周期"和"扫描相位"进行

设置。这里的扫描周期是过程数据库读取驱动的时间

◦扫描周期：两次扫描周期所间隔的时间

◦扫描相位：用于设置起始扫描时间。避免多组同时扫描造成的资源不足风险

> 注意：为了避免丢点现象，如果用户知道设备的数据变化周期，建议将驱动的刷新周期
>
> 配置为设备的数据变化周期的一半，点的扫描周期配置为数据变化周期的4分之一，如
>
> 设备的数据变化周期为4000毫秒，那么驱动刷新周期为2000毫秒，点的扫描周期为1000

毫秒。

•启动选项：

◦允许控制：用户如果要对变量进行写控制，需要勾选"允许控制"，iplat提供了"允许

扫描"和"禁止扫描"的选项，方便用户设置是否启用扫描机制

◦记录控制事件：当我们对过程量进行写值控制的时候，用于记录控制的事件，此功能

在"允许控制"被勾选的情况下可用，并且可以设定事件级别，方便查找时筛选

•保存历史数据：iplatda系统会将这个点的历史数据保存下来，用来绘制趋势图、脚本查

询历史数据等

•扩展参数：存储类的属性的扩展信息，该扩展信息可完全由用户自定义

配置TXT变量报警属性

•TXT变量报警属性如下：

补充
配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

配置TXT变量转储属性

•TXT变量转储设置基本同AI变量，参考AI变量转储属性

文本量TXT的属性表

•变量属性名称以"A_"开头的为字符型变量；变量属性名称以"F_"开头的为数值型变量文本量属性

属性名称	类型（长度）	
A_NAME	DT_ASCII	Readonly。127字节，由中文字符、字母、数字、'_'字符，'
A_DESC	DT_ASCII	127字节

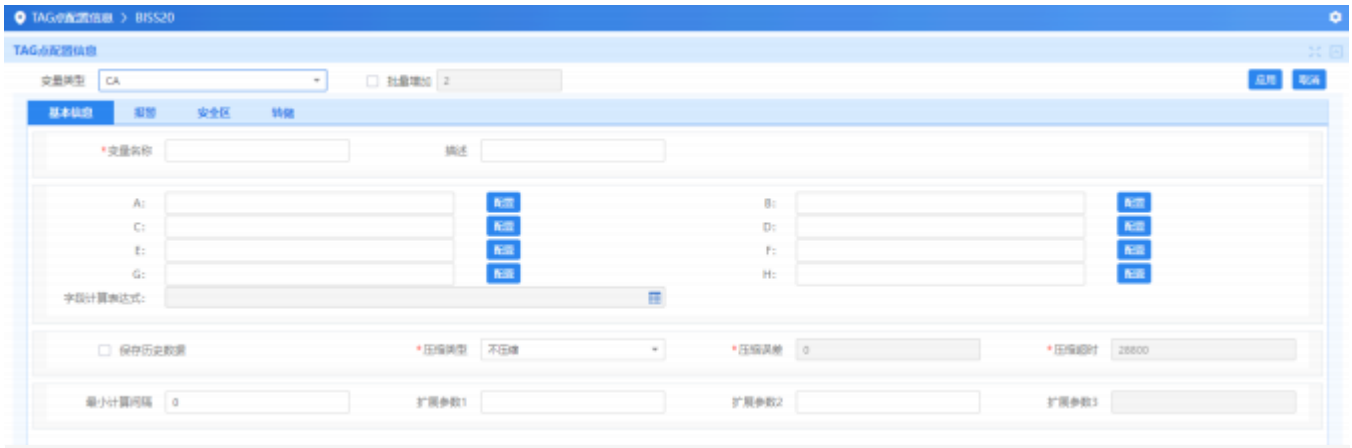
F_BLKTYPE	DT_UCHAR	DisplayOnly PB_TX=9
F_TAGID	DT_SLONG	Readonly,
F_ISTREND	DT_SINT16	1:保存历史趋势，0不保存历史趋势
F_COMPTYPE	DT_SINT16	保存历史数据后的配置属性
F_COMPDEV	DT_SINT16	保存历史数据后的配置属性
F_COMPMAXTIME	DT_SLONG	保存历史数据后的配置属性
A_ALM_AREA0	DT_SINT16	-1=无报警区；-2="ALL"报警区，只可能为报警区0的ID；1~
A_ALM_AREA1	DT_SINT16	
A_ALM_AREA2	DT_SINT16	
A_ALM_AREA3	DT_SINT16	
A_ALM_AREA4	DT_SINT16	
A_ALM_AREA5	DT_SINT16	
A_SEC_AREA0	DT_ASCII	每个安全区最长31字节
A_SEC_AREA1	DT_ASCII	
A_SEC_AREA2	DT_ASCII	
A_SEC_AREA3	DT_ASCII	
A_SEC_AREA4	DT_ASCII	
A_SEC_AREA5	DT_ASCII	
A_BLK_SN	DT_SLONG	
A_EXT_PARAM0	DT_ASCII	63字节
A_EXT_PARAM1	DT_ASCII	
A_EXT_PARAM2	DT_ASCII	
A_EXT_PARAM3	DT_ASCII	
A_SCAN_MODE	DT_ASCII	1=扫描机制0=通知机制
F_SCAN_MODE	DT_BIT	

F_SCAN_INTV	DT_ULONG	毫秒数
F_SCAN_PHS	DT_ULONG	毫秒数
A_IODRV	DT_ASCII	63字节
A_IOADDR	DT_ASCII	63字节
F_SCAN_ENABLE	DT_BIT	1/0
F_EVT_ENABLE	DT_BIT	1/0
F_EVT_PRIORITY	DT_UCHAR	0~255
F_OUTPUT_ENABLE	DT_BIT	1/0
F_INIT_VALUE	DT_DBL	
F_INITVAL_ENABLE	DT_BIT	1/0
F_VTQ	DT_ASCII	值、时间戳和数据质量
A_CV	DT_VTQ	
F_CV	DT_VTQ	
A_TIMESTAMP	DT_ASCII	YYYY-MM-DD HH:mm:ss.xxx
F_TIMESTAMP	DT_CV_TIME	64BIT,TCV_TimeStamp，使用接口读取后，转化成当前时
A_QUALITY	DT_ASCII	兼容OPC Quality
F_QUALITY	DT_UINT16	
A_QUALITY_SUB	DT_ASCII	
F_QUALITY_SUB	DT_UINT16	

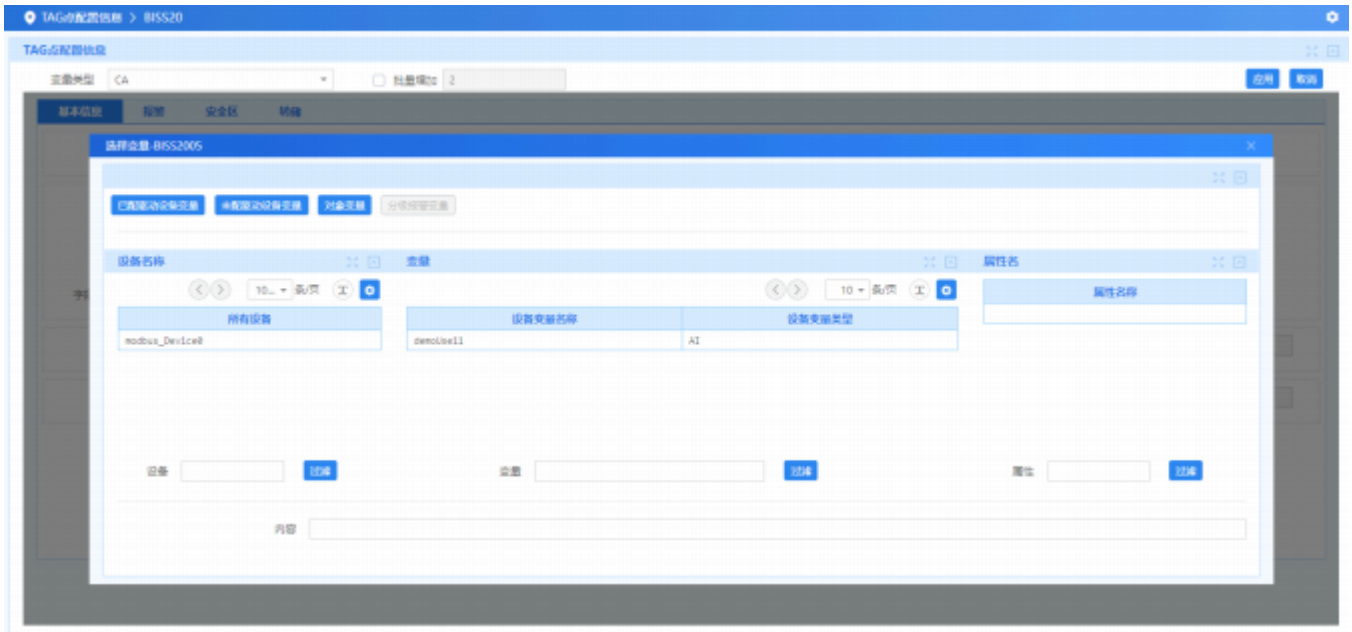
3.15 模拟计算量CA

配置CA变量基本属性

1. **CA的属性如下图所示：**



- 输入：用户可以在A---H，8个文本框中输入变量名称，也可以点击配置按钮在弹出的对话框中进行选



•选择变量对话框打开后，初始状态下设备名称类表中默认显示所有的设备，变量列表和属性名列表显示为空

◦点击设备名称按钮可以在设备名称列表中显示所有的设备名称

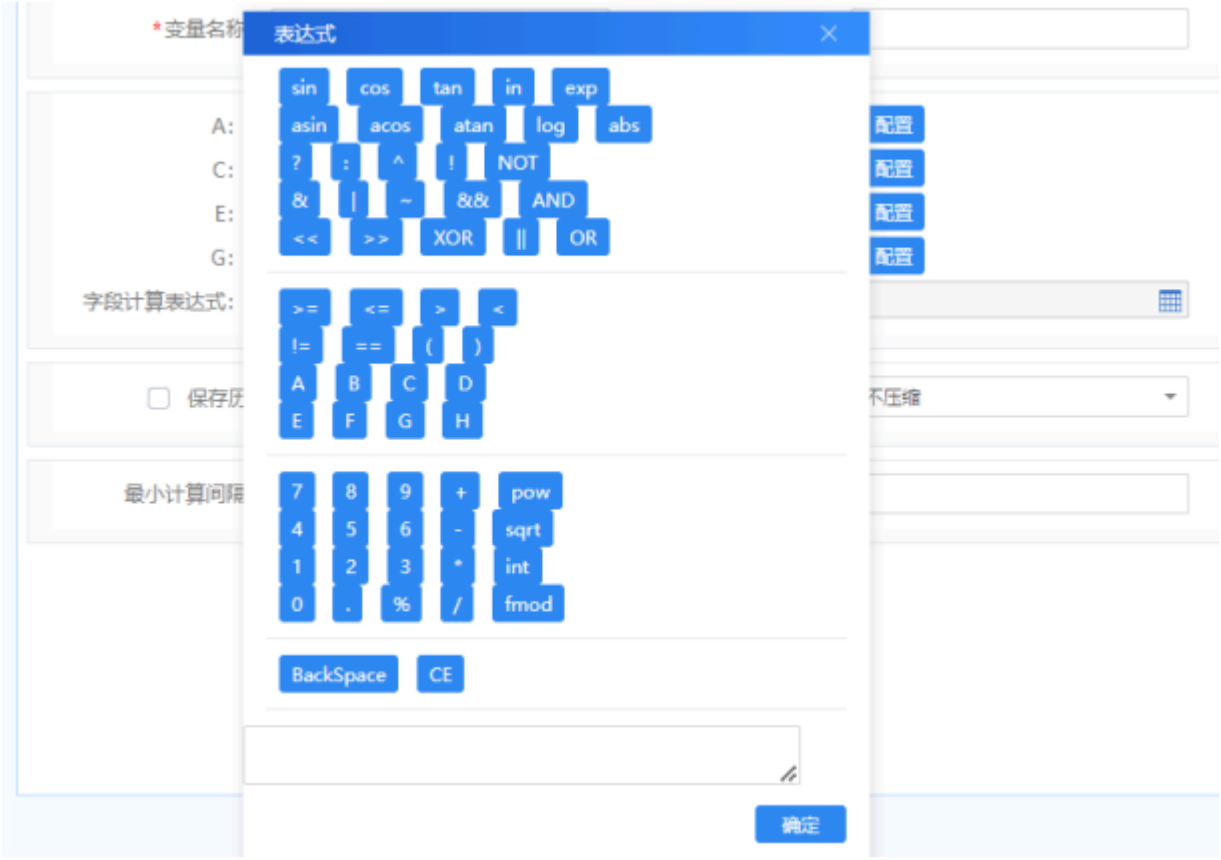
◦ 点击对象变量按钮可以在变量列表中显示所有对象变量的变量名称及变量类型

◦ 点击分级报警变量按钮可以在变量列表中显示所有LA变量的变量名称及变量类型

◦ 注意

.在点击全局变量、分级报警变量三个按钮中的任何一个后设备名称列表变灰不可用
.目前属性名列表中仅支持：变量值(F_CV、A_CV)、时间戳(F_TIMESTAMP、A_TIMESTAMP)，时间戳获取到秒、数据质量(F_QUALITY、A_QUALITY)，数据质量获取到的是数值，比如GOOD时为192

。字段计算表达式：用户可以在计算表达式文本框中进行变量的表达式编辑，或者在计算表达式文本框中双击，在弹出的表达式对话框中进行表达式编辑，如下图所示：



。在表达式对话框中编辑完表达式后可以点击初步检查按钮对表达式的合法性进行检查

> 注意:
> 如果选择三角函数，要将变量名填入函数括号中，否则表达式不合法。若不正确，请用户手动调整为正确的数学表达式。再次进行检查。

•最小计算间隔：毫秒为单位，计算表达式执行的间隔时间
•保存历史数据：ipIatda系统会将这个点的历史数据保存下来，用来绘制趋势图、脚本查询历史数据等
•扩展参数：存储类的属性的扩展信息，该扩展信息可完全由用户自定义

配置CA变量报警属性

•CA变量报警属性如下：

注：CA变量用Double类型存储，报警与AI点一样，根据配置报警等级进行报警。

补充
配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

配置CA变量转储属性

- CA变量转储设置基本同AI变量，参考AI变量转储属性

模拟计算量CA的属性表

- 变量属性名称以"A_"开头的为字符型变量；变量属性名称以"F_"开头的为数值型变量计算量属性

属性名称	类型（长度）	输入规则
A_NAME	DT_ASCII	Readonly。127字节，由中文字符、字母、数字、'_'字符
A_DESC	DT_ASCII	127字节
F_BLKTYPE	DT_UCHAR	DisplayOnly PB_CA=4
F_TAGID	DT_SLONG	Readonly,
F_ISTREND	DT_SINT16	1:保存历史趋势，0不保存历史趋势
F_COMPTYPE	DT_SINT16	保存历史数据后的配置属性
F_COMPDEV	DT_SINT16	保存历史数据后的配置属性
F_COMPMAXTIME	DT_SLONG	保存历史数据后的配置属性
A_ALM_AREA0	DT_SINT16	-1=无报警区；-2="ALL"报警区，只可能为报警区0的ID;
A_ALM_AREA1	DT_SINT16	
A_ALM_AREA2	DT_SINT16	
A_ALM_AREA3	DT_SINT16	
A_ALM_AREA4	DT_SINT16	
A_ALM_AREA5	DT_SINT16	
A_SEC_AREA0	DT_ASCII	每个安全区最长31字节

A_SEC_AREA1	DT_ASCII	
A_SEC_AREA2	DT_ASCII	
A_SEC_AREA3	DT_ASCII	
A_SEC_AREA4	DT_ASCII	
A_SEC_AREA5	DT_ASCII	
A_BLK_SN	DT_SLONG	
A_EXT_PARAM0	DT_ASCII	63字节
A_EXT_PARAM1	DT_ASCII	
A_EXT_PARAM2	DT_ASCII	
A_EXT_PARAM3	DT_ASCII	
A_CALC_EXPR	DT_ASCII	255字节
A_INPUT_0	DT_ASCII	159字节，参数、变量名、变量+属性
A_INPUT_1	DT_ASCII	
A_INPUT_2	DT_ASCII	
A_INPUT_3	DT_ASCII	
A_INPUT_4	DT_ASCII	
A_INPUT_5	DT_ASCII	
A_INPUT_6	DT_ASCII	
A_INPUT_7	DT_ASCII	
A_RECALC_INTV	DT_ULONG	毫秒数
F_ALM_ENABLE	DT_BIT	1/0
F_ALM_V_DEADBAND	DT_DBL	
F_ALM_TM_DEADBAND	DT_ULONG	毫秒数
F_ALM_LL_ENABLE	DT_BIT	
F_ALM_LL_THRESH	DT_DBL	浮点数，位于工程下限和工程上限之间，下同。LL,L,H,

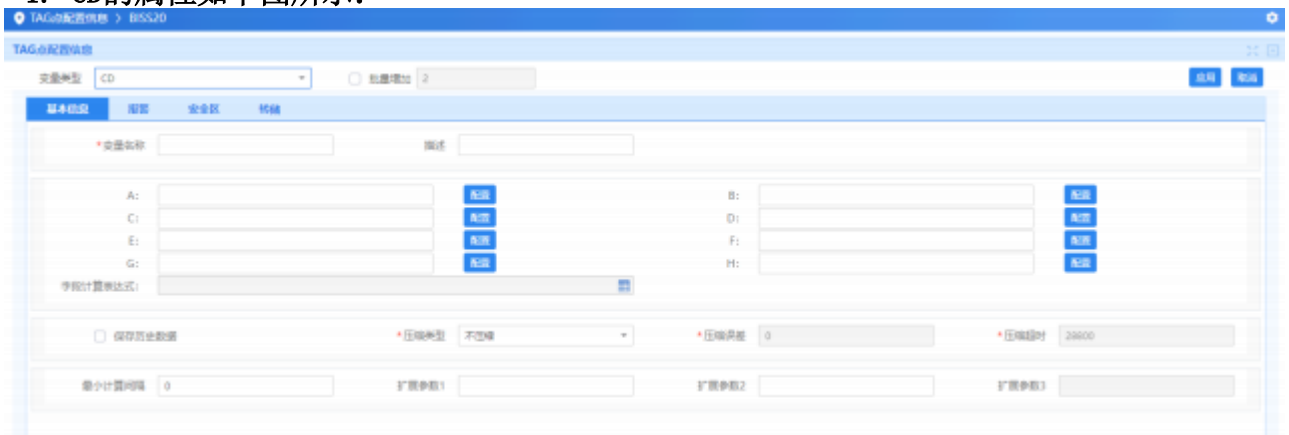
F_ALM_L_ENABLE	DT_BIT	
F_ALM_L_THRESH	DT_DBL	
F_ALM_H_ENABLE	DT_BIT	
F_ALM_H_THRESH	DT_DBL	
F_ALM_HH_ENABLE	DT_BIT	
F_ALM_HH_THRESH	DT_DBL	
F_ALM_DEV_ENABLE	DT_BIT	
属性名称	**类型（长度）**	**输入规则**
F_ALM_ROC_ENABLE	DT_BIT	
F_ALM_DEV_THRESH	DT_DBL	
F_ALM_DEV_CENT	DT_DBL	
F_ALM_ROC_THRESH	DT_DBL	
F_ALM_HH_PRIOR	DT_UCHAR	0～255
F_ALM_H_PRIOR	DT_UCHAR	
F_ALM_L_PRIOR	DT_UCHAR	
F_ALM_LL_PRIOR	DT_UCHAR	
F_ALM_DEV_PRIOR	DT_UCHAR	
F_ALM_ROC_PRIOR	DT_UCHAR	
F_ALM_HH_STATE	DT_UCHAR	0=无报警/报警已恢复已确认；1=报警未恢复已确认；2=
F_ALM_H_STATE	DT_UCHAR	
F_ALM_L_STATE	DT_UCHAR	
F_ALM_LL_STATE	DT_UCHAR	
F_ALM_DEV_STATE	DT_UCHAR	
F_ALM_ROC_STATE	DT_UCHAR	
F_VTQ	DT_ASCII	值、时间戳和数据质量

A_CV	DT_VTQ	
F_CV	DT_VTQ	
A_TIMESTAMP	DT_ASCII	YYYY-MM-DD HH:mm:ss.xxx
F_TIMESTAMP	DT_CV_TIME	64BIT,TCV_TimeStamp，使用接口读取后，转化成当前
A_QUALITY	DT_ASCII	兼容OPC Quality
F_QUALITY	DT_UINT16	
A_QUALITY_SUB	DT_ASCII	
F_QUALITY_SUB	DT_UINT16	
A_ALM_PIC	DT_ASCII	32字节，该点所有报警
F_ACK_ALM_TYPE	DT_UINT16	WriteOnly，写入待确认报警类型
F_DEL_ALM_TYPE	DT_UINT16	WriteOnly，写入待删除报警类型
F_ALM_CNT	DT_SLONG	报警状态为2或3
F_ALM_UNACK_CNT	DT_SLONG	报警状态为1或3
F_ALM_ALL_CNT	DT_SLONG	只要报警状态不为0

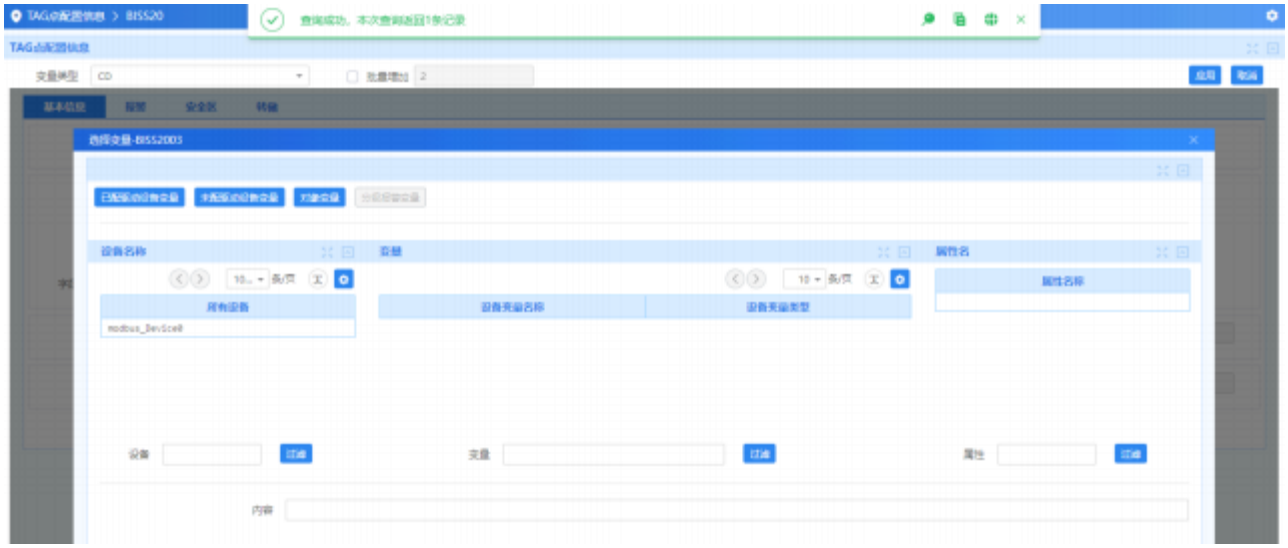
3.16 数字计算量CD

配置CD变量基本属性

1. CD的属性如下图所示：



•输入：用户可以在A---H，8个文本框中输入变量名称，也可以点击配置按钮在弹出的对话框中进行选择，选择变量对话框如下图所示



- 选择变量对话框打开后，初始状态下设备名称类表中默认显示所有的设备，变量列表和属性名列表显示为空
  - 点击设备名称按钮可以在设备名称列表中显示所有的设备名称
  - 点击对象变量按钮可以在变量列表中显示所有对象变量的变量名称及变量类型
  - 点击分级报警变量按钮可以在变量列表中显示所有LA变量的变量名称及变量类型
- 注意
  - 在点击全局变量、分级报警变量三个按钮中的任何一个后设备名称列表变灰不可用
  - 目前属性名列表中仅支持：变量值(F_CV、A_CV)、时间戳(F_TIMESTAMP、A_TIMESTAMP)，时间戳获取到秒、数据质量(F_QUALITY、A_QUALITY)，数据质量获取到的是数值，比如GOOD时为192
- 字段计算表达式：用户可以在计算表达式文本框中进行变量的表达式编辑，或者在计算表达式文本框中双击，在弹出的表达式对话框中进行表达式编辑，如下图所示：

•在表达式对话框中编辑完表达式后可以点击初步检查按钮对表达式的合法性进行检查

注意：如果选择三角函数，要将变量名填入函数括号中，否则表达式不合法。若不正确，请用户手动调整为正确的数学表达式。再次进行检查。

•最小计算间隔：毫秒为单位，计算表达式执行的间隔时间

•保存历史数据：iplatda系统会将这个点的历史数据保存下来，用来绘制趋势图、脚本查询历史数据等

•扩展参数：存储类的属性的扩展信息，该扩展信息可完全由用户自定义

配置CD变量报警属性

1. **CD变量报警属性如下：**

•报警区域名称：为了便于用户监控报警时筛选，我们提供了报警区域配置，具体参考报警区

域配置

•报警设置：

◦启用报警处理：属于状态报警类型选择

▪禁用状态报警：禁用此功能将不产生状态报警

▪状态0报警：当前值为0时报警

▪状态1报警：当前值为1时报警

注：CD变量用BOOL类型存储，报警与DI点一样，CD点会将最终结果转化为0/1（非0即1），

从而根据配置状态进行报警。

•打开/关闭报警：属于变位报警类型选择，支持多选

◦"打开"报警：当前值从0变为1时触发报警

◦"关闭"报警：当前值从1变为0时触发报警

•报警时间死区：对启用报警处理的"状态报警"有效，当前值到达报警触发条件，并保持一段

时间（报警时间死区设定值），才触发报警，否则不触发报警

补充
配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

配置CD变量转储属性

•**CD变量转储设置基本同AI变量，参考AI变量转储属性**

数字计算量CD的属性表

- 变量属性名称以"A_"开头的为字符型变量；变量属性名称以"F_"开头的为数值型变量计算量属性

属性名称	类型（长度）	输入规则
A_NAME	DT_ASCII	Readonly。127字节，由中文字符、字母、数字、'_'字符，
A_DESC	DT_ASCII	127字节
F_BLKTYPE	DT_UCHAR	DisplayOnly PB_CA=4
F_TAGID	DT_SLONG	Readonly,
F_ISTREND	DT_SINT16	1:保存历史趋势，0不保存历史趋势
F_COMPTYPE	DT_SINT16	保存历史数据后的配置属性
F_COMPDEV	DT_SINT16	保存历史数据后的配置属性
F_COMPMAXTIME	DT_SLONG	保存历史数据后的配置属性
A_ALM_AREA0	DT_SINT16	-1=无报警区；-2="ALL"报警区，只可能为报警区0的ID；1
A_ALM_AREA1	DT_SINT16	
A_ALM_AREA2	DT_SINT16	
A_ALM_AREA3	DT_SINT16	
A_ALM_AREA4	DT_SINT16	
A_ALM_AREA5	DT_SINT16	
A_SEC_AREA0	DT_ASCII	每个安全区最长31字节
A_SEC_AREA1	DT_ASCII	
A_SEC_AREA2	DT_ASCII	
A_SEC_AREA3	DT_ASCII	

A_SEC_AREA4	DT_ASCII	
A_SEC_AREA5	DT_ASCII	
A_BLK_SN	DT_SLONG	
A_EXT_PARAM0	DT_ASCII	63字节
A_EXT_PARAM1	DT_ASCII	
A_EXT_PARAM2	DT_ASCII	
A_EXT_PARAM3	DT_ASCII	
A_CALC_EXPR	DT_ASCII	255字节
A_INPUT_0	DT_ASCII	159字节，参数、变量名、变量+属性
A_INPUT_1	DT_ASCII	
A_INPUT_2	DT_ASCII	
A_INPUT_3	DT_ASCII	
A_INPUT_4	DT_ASCII	
A_INPUT_5	DT_ASCII	
A_INPUT_6	DT_ASCII	
A_INPUT_7	DT_ASCII	
A_RECALC_INTV	DT_ULONG	毫秒数
F_ALM_ENABLE	DT_BIT	1/0
F_ALM_V_DEADBAND	DT_DBL	
F_ALM_TDB	DT_ULONG	毫秒数
F_ALM_LL_ENABLE	DT_BIT	
F_ALM_LL_THRESH	DT_DBL	浮点数，位于工程下限和工程上限之间，下同。LL,L,H,
F_ALM_L_ENABLE	DT_BIT	
F_ALM_L_THRESH	DT_DBL	
F_ALM_H_ENABLE	DT_BIT	

F_ALM_H_THRESH	DT_DBL	
F_ALM_HH_ENABLE	DT_BIT	
F_ALM_HH_THRESH	DT_DBL	
F_ALM_DEV_ENABLE	DT_BIT	
F_ALM_ROC_ENABLE	DT_BIT	
F_ALM_DEV_THRESH	DT_DBL	
F_ALM_DEV_CENT	DT_DBL	
F_ALM_ROC_THRESH	DT_DBL	
F_ALM_HH_PRIOR	DT_UCHAR	0～255
F_ALM_H_PRIOR	DT_UCHAR	
F_ALM_L_PRIOR	DT_UCHAR	
F_ALM_LL_PRIOR	DT_UCHAR	
F_ALM_DEV_PRIOR	DT_UCHAR	
F_ALM_ROC_PRIOR	DT_UCHAR	
F_ALM_HH_STATE	DT_UCHAR	0=无报警/报警已恢复已确认；1=报警未恢复已确认；2=报
F_ALM_H_STATE	DT_UCHAR	
F_ALM_L_STATE	DT_UCHAR	
F_ALM_LL_STATE	DT_UCHAR	
F_ALM_DEV_STATE	DT_UCHAR	
F_ALM_ROC_STATE	DT_UCHAR	
F_VTQ	DT_ASCII	值、时间戳和数据质量
A_CV	DT_VTQ	
F_CV	DT_VTQ	
A_TIMESTAMP	DT_ASCII	YYYY-MM-DD HH:mm:ss.xxx
F_TIMESTAMP	DT_CV_TIME	64BIT,TCV_TimeStamp，使用接口读取后，转化成当前时

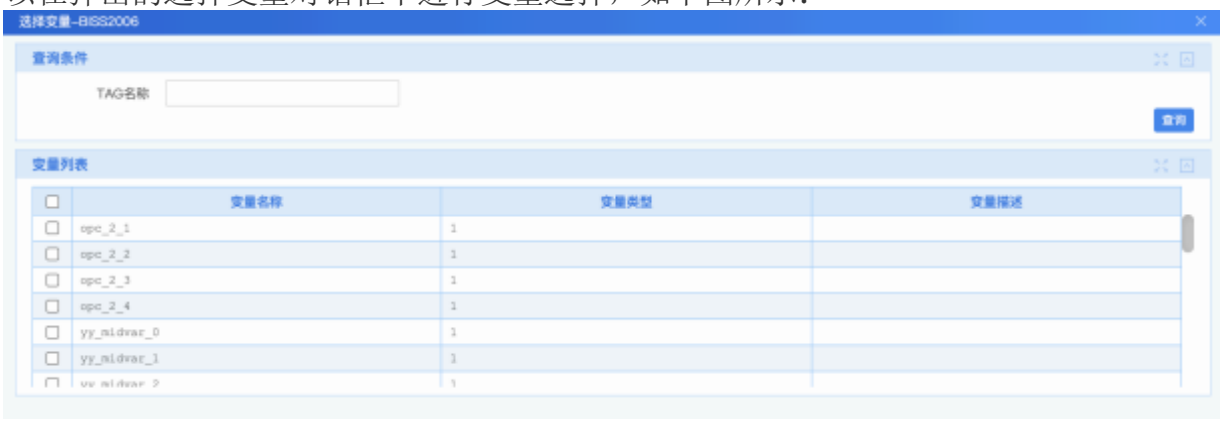
A_QUALITY	DT_ASCII	兼容OPC Quality
F_QUALITY	DT_UINT16	
A_QUALITY_SUB	DT_ASCII	
F_QUALITY_SUB	DT_UINT16	
A_ALM_PIC	DT_ASCII	32字节，该点所有报警
F_ACK_ALM_TYPE	DT_UINT16	WriteOnly，写入待确认报警类型
F_DEL_ALM_TYPE	DT_UINT16	WriteOnly，写入待删除报警类型
F_ALM_CNT	DT_SLONG	报警状态为2或3
F_ALM_UNACK_CNT	DT_SLONG	报警状态为1或3
F_ALM_ALL_CNT	DT_SLONG	只要报警状态不为0

3.17 分级报警量LA

配置LA变量基本属性

1. **LA的属性如下图所示：**

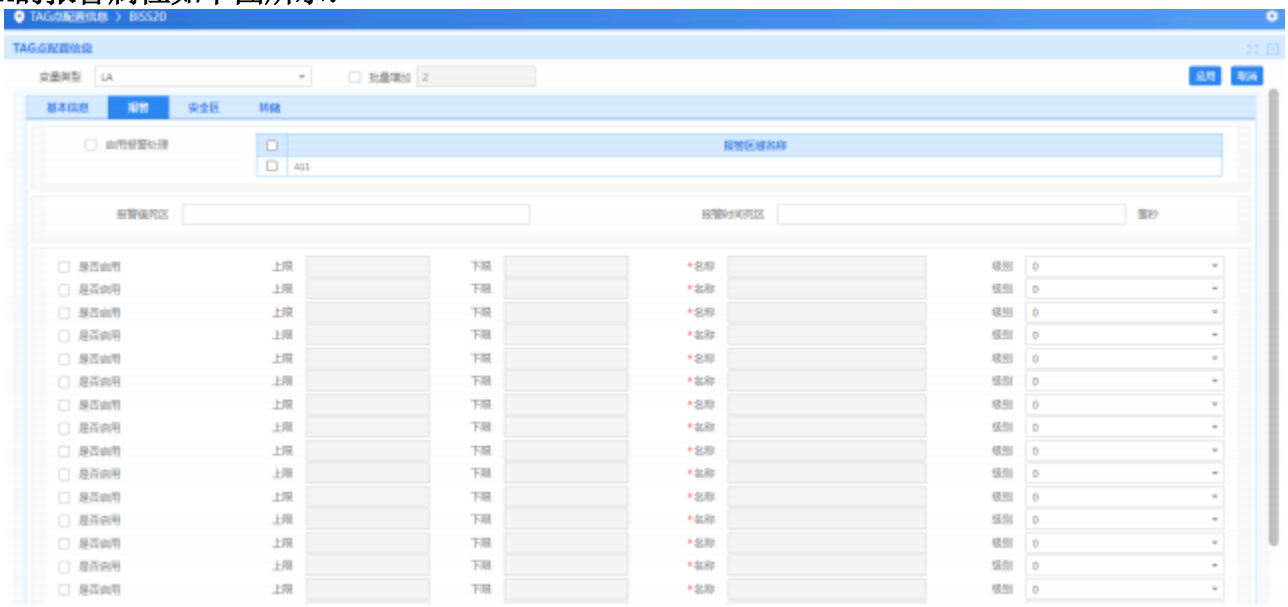
- 输入变量：输入已有的变量名。用户也可以点击输入变量文本框后面的"配置"按钮可以在弹出的选择变量对话框中进行变量选择，如下图所示：



- 保存历史数据：iplatda系统会将这个点的历史数据保存下来，用来绘制趋势图、脚本查询历史数据等。
- 扩展参数：存储类的属性的扩展信息，该扩展信息可完全由用户自定义。

配置LA变量报警属性

1. LA的报警属性如下图所示：



- 启用报警处理：用户如果需要对变量进行报警处理，需要勾选"启用报警处理"选项，报警区域默认显示为ALL
- 报警区域名称：为了便于用户监控报警时筛选，我们提供了报警区域配置，具体参考报警区域配置
- 报警设置：
  - 报警值死区：配置报警值死区后，报警的触发和恢复都要在触发限值和恢复限制上加上或者减去报警值死区的大小才判断触发或者恢复，避免报警在限值附近波动时频繁发生的报警和恢复消息。例如：当高报警值为24，且设置报警值死区为1时，

    超过25报警，低于23恢复
  - 报警时间死区：配置报警时间区后，需要报警在一段时间死区内维持报警状态，才触发报警，否则不触发报警，其中变化率报警不受时间死区限制
- 报警范围设置：报警范围的配置共16个，可以通过点击报警范围列表中第一列的复选框来启用某一行的报警范围配置，其中报警级别为0～255。报警级别数字越小，级别越高，级别为0的报警，报警级别是最高的。报警区域的配置同AI变量报警属性页
- 分级报警（LA变量报警属性）：变量值大于等于下限且小于上限时，就触发此种类型的分级报警

补充
配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

配置LA变量转储属性

- LA变量转储设置基本同AI变量，参考AI变量转储属性

分级报警量LA的属性表

•变量属性名称以"A_"开头的为字符型变量；变量属性名称以"F_"开头的为数值型变量分级报警变量

属性名称	类型（长度）	输入规则
A_NAME	DT_ASCII	Readonly。127字节，由中文字符、字母、数字、’_’字符
A_DESC	DT_ASCII	127字节
F_BLKTYPE	DT_UCHAR	DisplayOnly PB_LA=7
F_TAGID	DT_SLONG	Readonly,
F_ISTREND	DT_SINT16	1:保存历史趋势，0不保存历史趋势
F_COMPTYPE	DT_SINT16	保存历史数据后的配置属性
F_COMPDEV	DT_SINT16	保存历史数据后的配置属性
F_COMPMAXTIME	DT_SLONG	保存历史数据后的配置属性
A_ALM_AREA0	DT_SINT16	-1=无报警区-2="ALL"报警区，只可能为报警区0的ID1~
A_ALM_AREA1	DT_SINT16	
A_ALM_AREA2	DT_SINT16	
A_ALM_AREA3	DT_SINT16	
A_ALM_AREA4	DT_SINT16	
A_ALM_AREA5	DT_SINT16	
A_SEC_AREA0	DT_ASCII	每个安全区最长31字节
A_SEC_AREA1	DT_ASCII	
A_SEC_AREA2	DT_ASCII	
A_SEC_AREA3	DT_ASCII	
A_SEC_AREA4	DT_ASCII	
A_SEC_AREA5	DT_ASCII	
A_BLK_SN	DT_SLONG	
A_EXT_PARAM0	DT_ASCII	63字节
A_EXT_PARAM1	DT_ASCII	

A_EXT_PARAM2	DT_ASCII	
A_EXT_PARAM3	DT_ASCII	
A_INPUT_TAGNAME	DT_ASCII	127字节
F_ALM_ENABLE	DT_BIT	0/1
F_ALM_V_DEADBAND	DT_DBL	
F_ALM_TM_DEADBAND	DT_ULONG	毫秒数
F_ALM_0_ENABLE	DT_BIT	
F_ALM_0_L	DT_DBL	
F_ALM_0_H	DT_DBL	
F_ALM_0_PRIORITY	DT_UCHAR	
F_ALM_1_ENABLE	DT_BIT	
F_ALM_1_L	DT_DBL	
F_ALM_1_H	DT_DBL	
F_ALM_1_PRIORITY	DT_UCHAR	
F_ALM_2_ENABLE	DT_BIT	
F_ALM_2_L	DT_DBL	
F_ALM_2_H	DT_DBL	
F_ALM_2_PRIORITY	DT_UCHAR	
F_ALM_3_ENABLE	DT_BIT	
F_ALM_3_L	DT_DBL	
F_ALM_3_H	DT_DBL	
F_ALM_3_PRIORITY	DT_UCHAR	
F_ALM_4_ENABLE	DT_BIT	
F_ALM_4_L	DT_DBL	
F_ALM_4_H	DT_DBL	

F_ALM_4_PRIORITY	DT_UCHAR	
F_ALM_5_ENABLE	DT_BIT	
F_ALM_5_L	DT_DBL	
F_ALM_5_H	DT_DBL	
F_ALM_5_PRIORITY	DT_UCHAR	
F_ALM_6_ENABLE	DT_BIT	
F_ALM_6_L	DT_DBL	
F_ALM_6_H	DT_DBL	
F_ALM_6_PRIORITY	DT_UCHAR	
F_ALM_7_ENABLE	DT_BIT	
F_ALM_7_L	DT_DBL	
F_ALM_7_H	DT_DBL	
F_ALM_7_PRIORITY	DT_UCHAR	
F_ALM_8_ENABLE	DT_BIT	
F_ALM_8_L	DT_DBL	
F_ALM_8_H	DT_DBL	
F_ALM_8_PRIORITY	DT_UCHAR	
F_ALM_9_ENABLE	DT_BIT	
F_ALM_9_L	DT_DBL	
F_ALM_9_H	DT_DBL	
F_ALM_9_PRIORITY	DT_UCHAR	
F_ALM_10_ENABLE	DT_BIT	
F_ALM_10_L	DT_DBL	
F_ALM_10_H	DT_DBL	
F_ALM_10_PRIORITY	DT_UCHAR	

F_ALM_11_ENABLE	DT_BIT	
F_ALM_11_L	DT_DBL	
F_ALM_11_H	DT_DBL	
F_ALM_11_PRIORITY	DT_UCHAR	
F_ALM_12_ENABLE	DT_BIT	
F_ALM_12_L	DT_DBL	
F_ALM_12_H	DT_DBL	
F_ALM_12_PRIORITY	DT_UCHAR	
F_ALM_13_ENABLE	DT_BIT	
F_ALM_13_L	DT_DBL	
F_ALM_13_H	DT_DBL	
F_ALM_13_PRIORITY	DT_UCHAR	
F_ALM_14_ENABLE	DT_BIT	
F_ALM_14_L	DT_DBL	
F_ALM_14_H	DT_DBL	
F_ALM_14_PRIORITY	DT_UCHAR	
F_ALM_15_ENABLE	DT_BIT	
F_ALM_15_L	DT_DBL	
F_ALM_15_H	DT_DBL	
F_ALM_15_PRIORITY	DT_UCHAR	
A_ALM_0_DESP	DT_ASCII	
A_ALM_1_DESP	DT_ASCII	31字节
A_ALM_2_DESP	DT_ASCII	
A_ALM_3_DESP	DT_ASCII	
A_ALM_4_DESP	DT_ASCII	

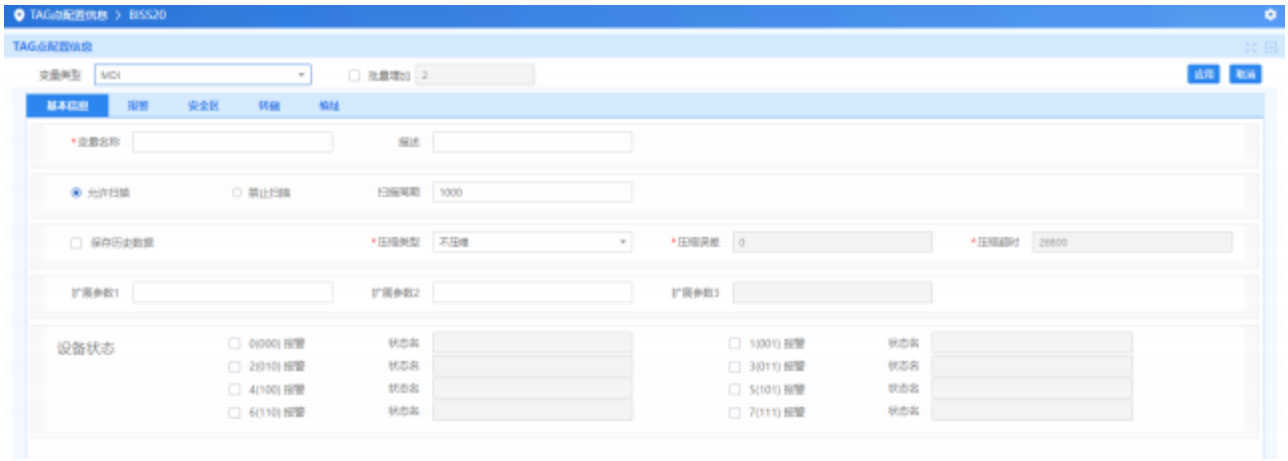
A_ALM_5_DESP	DT_ASCII	
A_ALM_6_DESP	DT_ASCII	
A_ALM_7_DESP	DT_ASCII	
A_ALM_8_DESP	DT_ASCII	
A_ALM_9_DESP	DT_ASCII	
A_ALM_10_DESP	DT_ASCII	
A_ALM_11_DESP	DT_ASCII	
A_ALM_12_DESP	DT_ASCII	
A_ALM_13_DESP	DT_ASCII	
A_ALM_14_DESP	DT_ASCII	
A_ALM_15_DESP	DT_ASCII	
A_ALM_0_STATE	DT_UCHAR	
A_ALM_1_STATE	DT_UCHAR	
A_ALM_2_STATE	DT_UCHAR	
A_ALM_3_STATE	DT_UCHAR	
A_ALM_4_STATE	DT_UCHAR	
A_ALM_5_STATE	DT_UCHAR	
A_ALM_6_STATE	DT_UCHAR	
A_ALM_7_STATE	DT_UCHAR	
A_ALM_8_STATE	DT_UCHAR	
A_ALM_9_STATE	DT_UCHAR	
A_ALM_10_STATE	DT_UCHAR	
A_ALM_11_STATE	DT_UCHAR	
A_ALM_12_STATE	DT_UCHAR	
A_ALM_13_STATE	DT_UCHAR	

A_ALM_14_STATE	DT_UCHAR	
A_ALM_15_STATE	DT_UCHAR	
F_VTQ	DT_ASCII	值、时间戳和数据质量
A_CV	DT_ASCII	
F_CV	DT_FLT	
A_TIMESTAMP	DT_ASCII	YYYY-MM-DD H24:mm:ss.xxx
F_TIMESTAMP	DT_CV_TIME	64BIT，UTC SYSTEMTIME，使用接口读取后，转化成
A_QUALITY	DT_ASCII	参考"5数据质量列表"
F_QUALITY	DT_UINT16	
A_QUALITY_SUB	DT_ASCII	
F_QUALITY_SUB	DT_UINT16	
A_ALM_PIC	DT_ASCII	32字节，该点所有报警
F_ACK_ALM_TYPE	DT_UINT16	WriteOnly，写入待确认报警类型
F_DEL_ALM_TYPE	DT_UINT16	WriteOnly，写入待删除报警类型
F_ALM_CNT	DT_SLONG	报警状态为2或3
F_ALM_UNACK_CNT	DT_SLONG	报警状态为1或3
F_ALM_ALL_CNT	DT_SLONG	只要报警状态不为0

3.18 多数字量输入MDI

配置MDI变量基本属性

1.**MDI的属性如下图所示：**



注意：如果用户没有配置"所属驱动"、"设备名称"和"设备地址"，则添加的变量默认为全局变量，全局变量可以在所有客户端中通用。

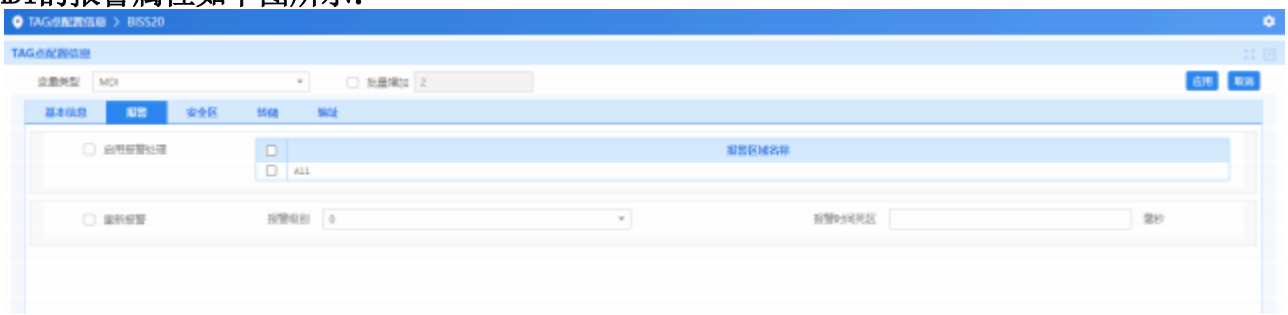
- 扫描设置：
  - 通知机制：指变量值有变化的时候进行通知读取
  - 扫描机制：通过轮循扫描的方式对变量进行读取，可对"扫描周期"和"扫描相位"进行设置。这里的扫描周期是过程数据库读取驱动的时间
    - 扫描周期：两次扫描周期所间隔的时间

注意：为了避免丢点现象，如果用户知道设备的数据变化周期，建议将驱动的刷新周期配置为设备的数据变化周期的一半，点的扫描周期配置为数据变化周期的4分之一，如设备的数据变化周期为4000毫秒，那么驱动刷新周期为2000毫秒，点的扫描周期为1000毫秒。

- 保存历史数据：iplatda系统会将这个点的历史数据保存下来，用来绘制趋势图、脚本查询历史数据等
- 扩展参数：存储类的属性的扩展信息，该扩展信息可完全由用户自定义
- 设备状态：MDI变量可由1到3个数字量组成，可以配置8种报警状态

配置MDI变量报警属性

1. **MDI的报警属性如下图所示：**



- 启用报警处理：用户如果需要对变量进行报警处理，需要勾选"启用报警处理"选项
- 报警区域名称：为了便于用户监控报警时筛选，我们提供了报警区域配置，具体参考报警区域配置
- 重新报警：如果配置了重新报警,当过程变量当前值所在状态发生变化,且该状态设置了报警启用,则触发MDI报警；如果未配置重新报警，而之前报警状态如果为报警的话，

  则如果对应当前状态仍然为报警（状态可以发生了变化，由一个报警状态变为另一个报警状态）则不产生新报警
- 报警级别：数字越小，级别越高，级别为0的选项设置，级别最高
- 报警时间死区：对启用报警处理的"状态报警"有效，当前值到达报警触发条件，并保持一段时间（报警时间死区设定值），才触发报警，否则不触发报警

补充
配置完成，部署后报警信息推送到报警控件，相关的展示样式配置要在报警控件中进行

配置MDI变量转储属性

•MDI变量转储设置基本同AI变量，参考AI变量转储属性

配置MDI变量编址属性

•在编址属性页上，用户可配置此属性页上所有配置项。配置页如下图所示：

•MDI变量可由1到3个数字量组成，每个编址对应到一个数字量

多数字量输入MDI的属性表

•变量属性名称以"A_"开头的为字符型变量；变量属性名称以"F_"开头的为数值型变量多状态数字输入量属性

属性名称	类型（长度）	输入规则
A_NAME	DT_ASCII	Readonly。127字节，由中文字符、字母、数字、’_’字符，
A_DESC	DT_ASCII	127字节
F_BLKTYPE	DT_UCHAR	DisplayOnly PB_MDI=8
F_TAGID	DT_SLONG	Readonly，
F_ISTREND	DT_SINT16	1:保存历史趋势，0不保存历史趋势
F_COMPTYPE	DT_SINT16	保存历史数据后的配置属性
F_COMPDEV	DT_SINT16	保存历史数据后的配置属性
F_COMPMAXTIME	DT_SLONG	保存历史数据后的配置属性
A_ALM_AREA0	DT_SINT16	-1=无报警区；-2="ALL"报警区，只可能为报警区0的ID；

A_ALM_AREA1	DT_SINT16	
A_ALM_AREA2	DT_SINT16	
A_ALM_AREA3	DT_SINT16	
A_ALM_AREA4	DT_SINT16	
A_ALM_AREA5	DT_SINT16	
A_SEC_AREA0	DT_ASCII	每个安全区最长31字节
A_SEC_AREA1	DT_ASCII	
A_SEC_AREA2	DT_ASCII	
A_SEC_AREA3	DT_ASCII	
A_SEC_AREA4	DT_ASCII	
A_SEC_AREA5	DT_ASCII	
A_BLK_SN	DT_SLONG	
A_EXT_PARAM0	DT_ASCII	63字节
A_EXT_PARAM1	DT_ASCII	
A_EXT_PARAM2	DT_ASCII	
A_EXT_PARAM3	DT_ASCII	
A_SCAN_MODE	DT_ASCII	1=扫描机制0=通知机制
F_SCAN_MODE	DT_BIT	
F_SCAN_INTV	DT_ULONG	毫秒数
F_SCAN_PHS	DT_ULONG	毫秒数
A_IODRV0	DT_ASCII	63字节
A_IOADDR0	DT_ASCII	63字节
A_IODRV1	DT_ASCII	63字节
A_IOADDR1	DT_ASCII	63字节
F_IOENABLE_1	DT_BIT	1/0

A_IODRV2	DT_ASCII	63字节
A_IOADDR2	DT_ASCII	63字节
F_IOENABLE_2	DT_BIT	
A_EGU_0	DT_ASCII	31字节
A_EGU_1	DT_ASCII	31字节
A_EGU_2	DT_ASCII	31字节
A_EGU_3	DT_ASCII	31字节
A_EGU_4	DT_ASCII	31字节
A_EGU_5	DT_ASCII	31字节
A_EGU_6	DT_ASCII	31字节
A_EGU_7	DT_ASCII	31字节
F_SCAN_ENABLE	DT_BIT	1/0
F_SOE_ENABLE	DT_BIT	1/0
F_ALM_ENABLE	DT_BIT	1/0
F_REALARM_ENABLE	DT_BIT	1/0
F_OVERRIDE_ENABLE	DT_BIT	1/0
F_OVERRIDE_VALUE	DT_DBL	
F_ALM_0	DT_BIT	
F_ALM_1	DT_BIT	
F_ALM_2	DT_BIT	
F_ALM_3	DT_BIT	
F_ALM_4	DT_BIT	
F_ALM_5	DT_BIT	
F_ALM_6	DT_BIT	
F_ALM_7	DT_BIT	

F_ALM_0_STATE	DT_UCHAR	
F_ALM_1_STATE	DT_UCHAR	
F_ALM_2_STATE	DT_UCHAR	
F_ALM_3_STATE	DT_UCHAR	
F_ALM_4_STATE	DT_UCHAR	
F_ALM_5_STATE	DT_UCHAR	
F_ALM_6_STATE	DT_UCHAR	
F_ALM_7_STATE	DT_UCHAR	
F_ALM_PRIORITY	DT_UCHAR	
F_ALM_TDB	DT_ULONG	ms
F_OUTPUT_ENABLE	DT_BIT	1/0
F_VTQ	DT_ASCII	值、时间戳和数据质量
A_CV	DT_VTQ	
F_CV	DT_VTQ	
A_TIMESTAMP	DT_ASCII	YYYY-MM-DD HH:mm:ss.xxx
F_TIMESTAMP	DT_CV_TIME	64BIT,TCV_TimeStamp，使用接口读取后，转化成当前
A_QUALITY	DT_ASCII	兼容OPC Quality
F_QUALITY	DT_UINT16	
A_QUALITY_SUB	DT_ASCII	
F_QUALITY_SUB	DT_UINT16	
A_ALM_PIC	DT_ASCII	32字节，该点所有报警
F_ACK_ALM_TYPE	DT_UINT16	WriteOnly，写入待确认报警类型
F_DEL_ALM_TYPE	DT_UINT16	WriteOnly，写入待删除报警类型
F_ALM_CNT	DT_SLONG	报警状态为2或3
F_ALM_UNACK_CNT	DT_SLONG	报警状态为1或3

F_ALM_ALL_CNT	DT_SLONG	只要报警状态不为0
注意		
**超驰值：主要用在调试控制的下发，需要手动配置参数F_OVERRIDE_ENABLE，置为1		
,表示启动超驰。这样发送的控制命令将不会直接发送给设备。**		

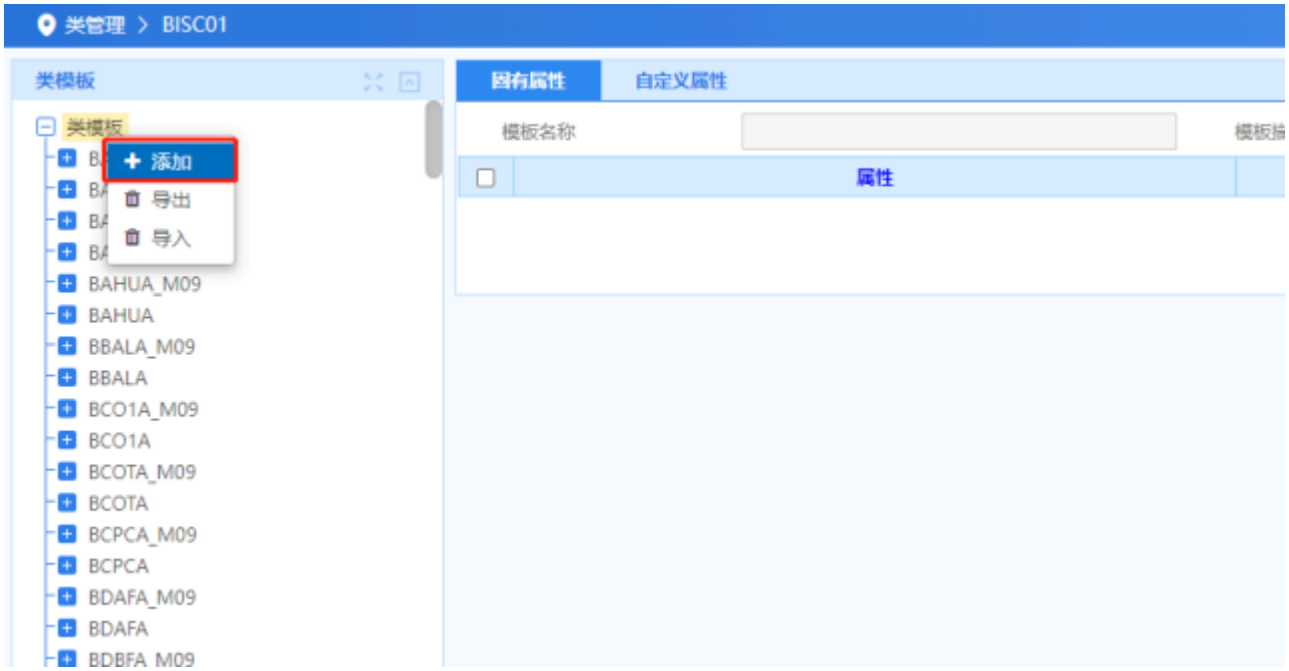
- 创建类模板及自定义属性
  - 创建类模板
  - 配置自定义属性
  - 类模板的导出
  - 类模板的导入
- 添加子系统
  - 步骤演示
- 创建对象
  - 步骤演示
- 对象变量查询
  - 步骤演示
- 对象点使用示例
  - 创建电机参数类模板
  - 实例化对象及建点
  - 对象的自定义属性点完善
  - 创建电机参数图元
  - 画面上使用图元并绑定对象

3.19 创建类模板及自定义属性

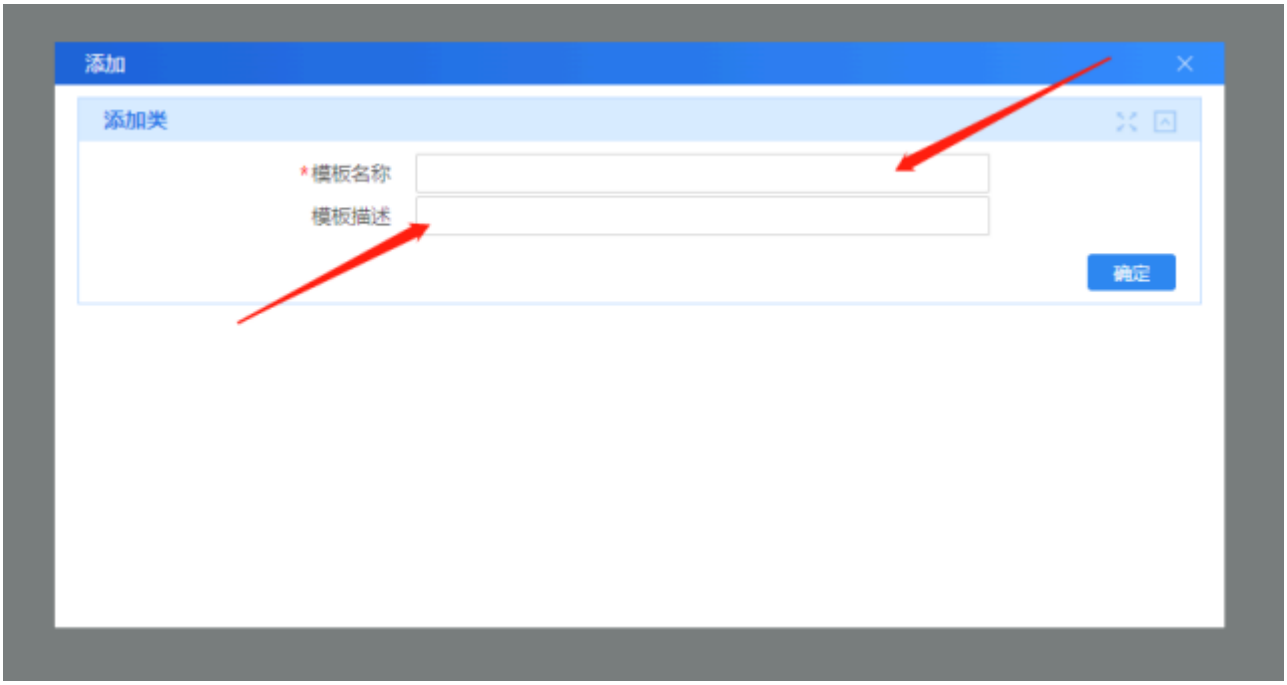
创建类模板

1. 在首页左侧栏中--->SCADA管理--->全局配置--->类管理。

2. 进入到类模板管理页面，右键类模板，点击添加。

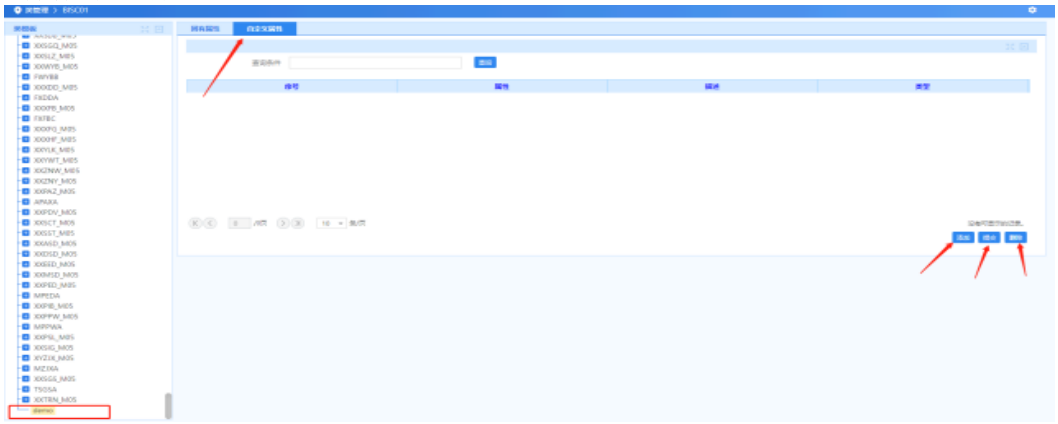


3. 模板名称命名之后不可修改，注意准确输入！模板描述为非必填字段。填好模板名称之后，点击确定，就能在左侧菜单栏里看到新建的类模板。

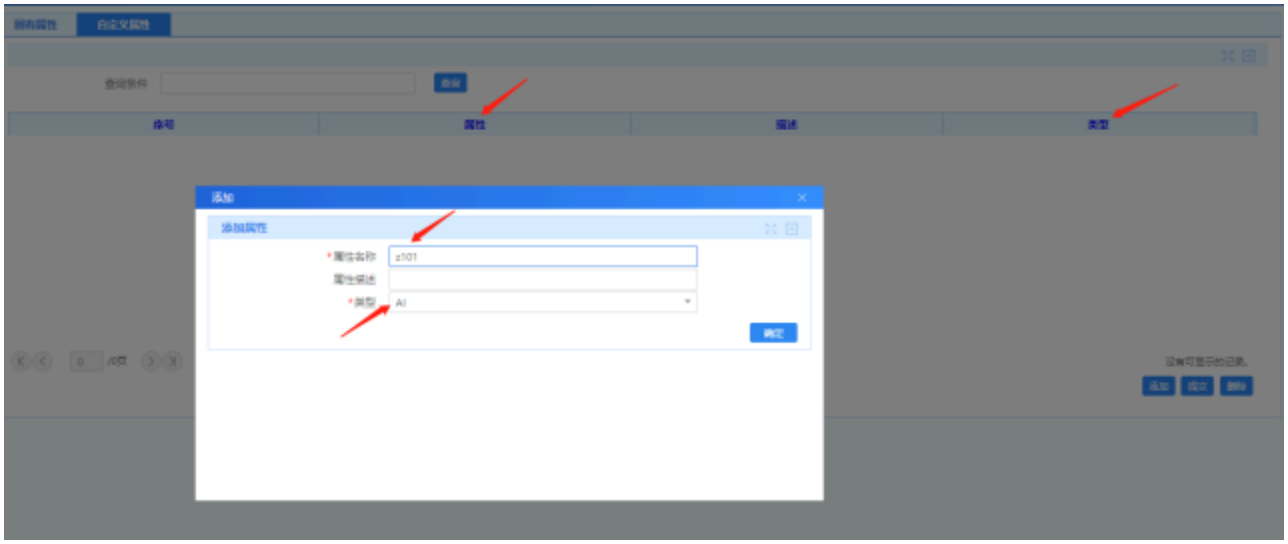


配置自定义属性

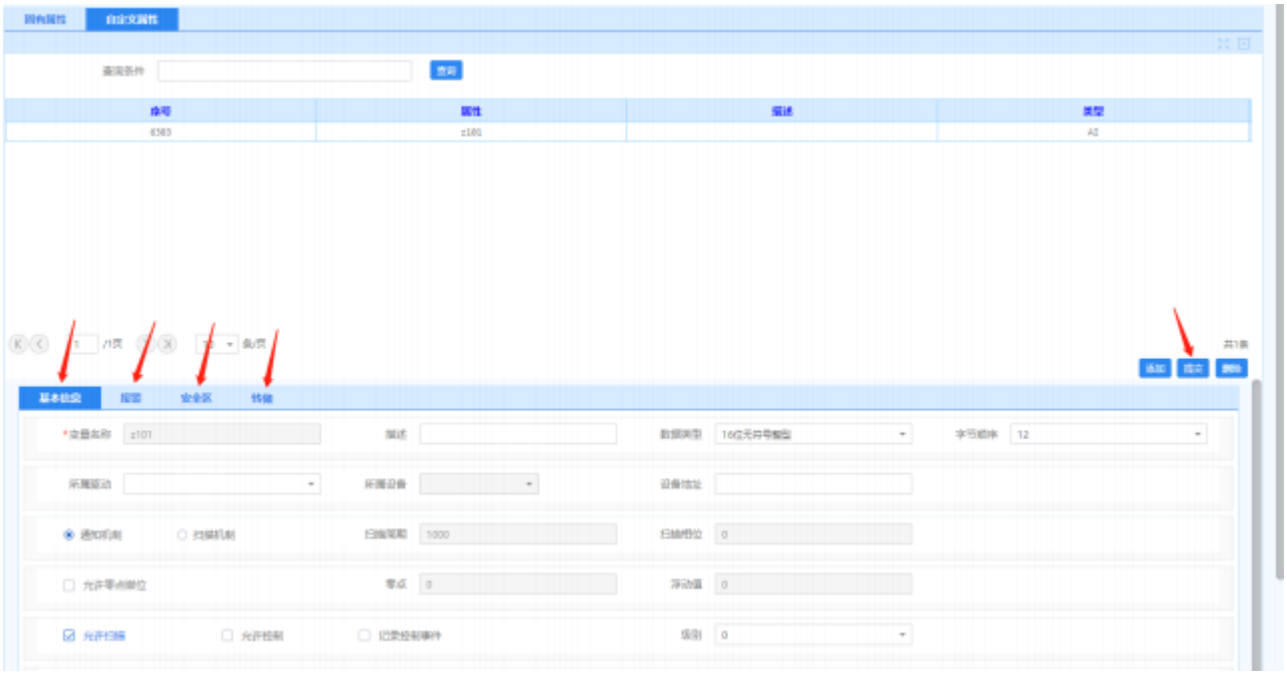
1. 双击左侧新建好的类模板，出现以下两个tab页。因为只做类模板的简单介绍使用。直接看到自定义属性。



2. 点击添加，弹出如下画面：



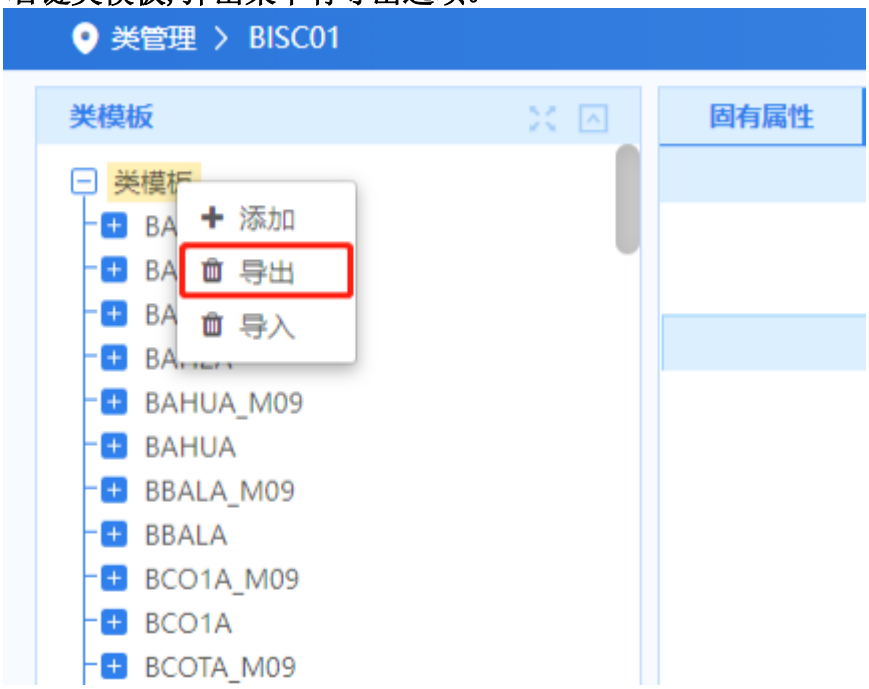
2. 填好属性名称和类型，这里以AI类型为例：



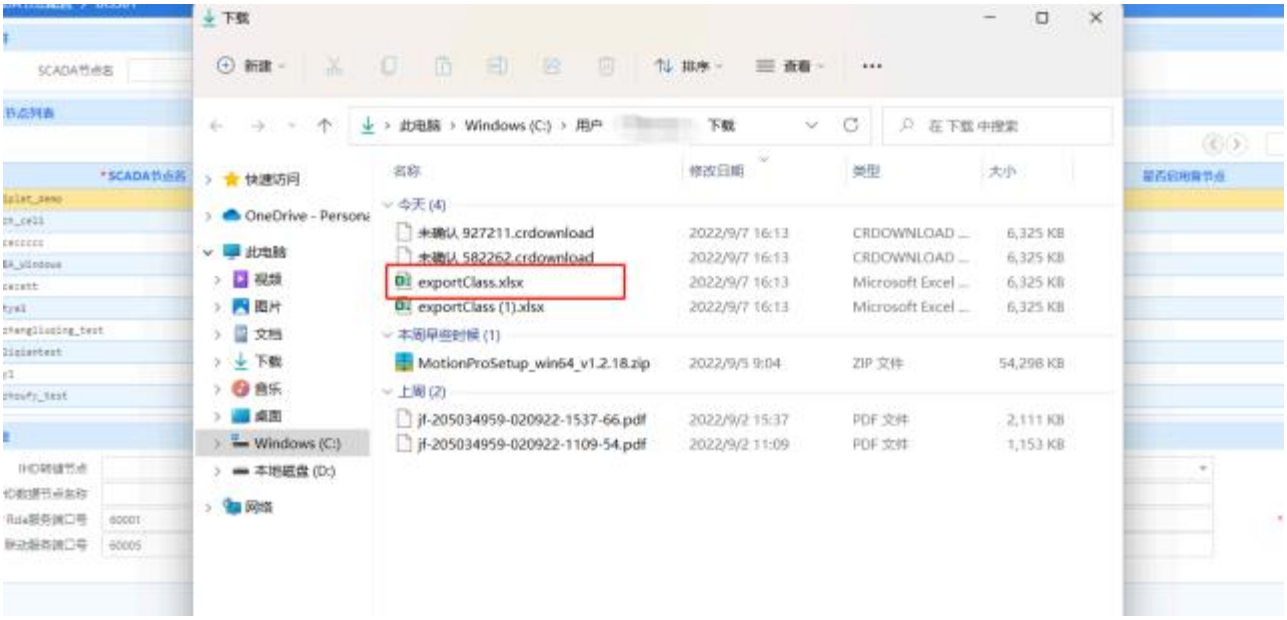
- 单击单元行，可针对每个属性配置4个tab页的不同信息，tab页的信息是在点完提交按钮之后保存的。但是，因为这是类管理页面，到时候示例中用到的最终是由类产生的对象的属性，所以这里可以不用配置

类模板的导出

1. **右键类模板,弹出菜单有导出选项。**



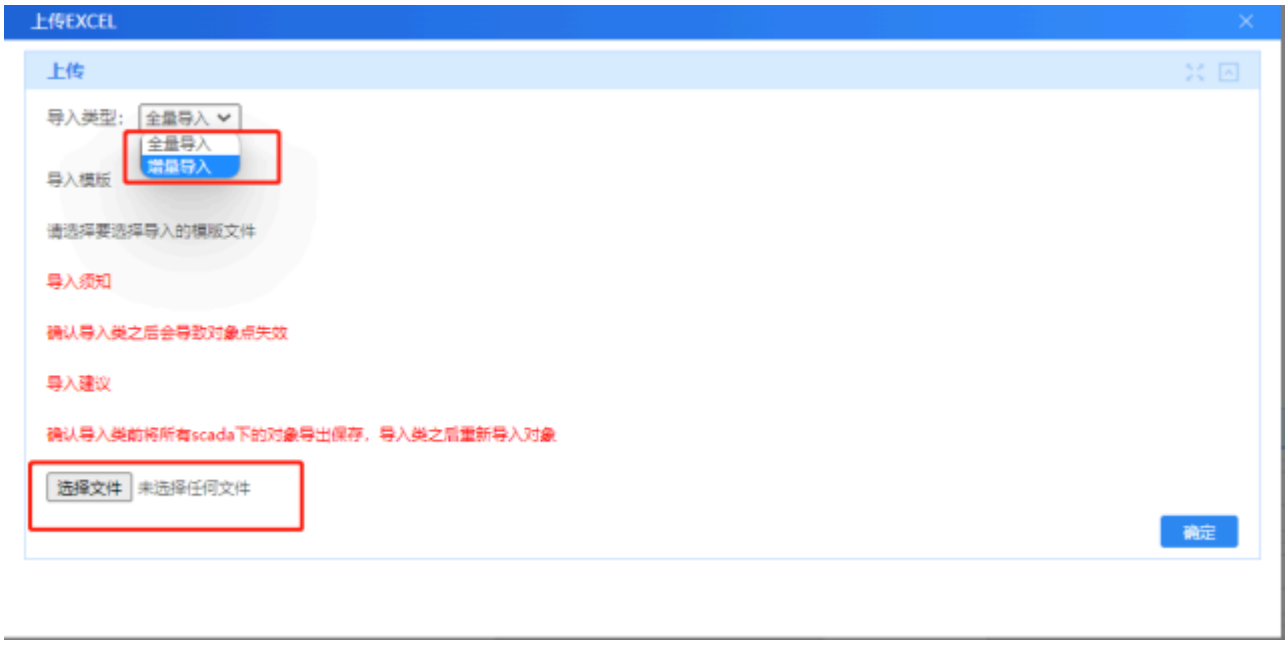
2. 点击导出选项，弹出文件保存窗口；选择保存文件夹，修改文件名（默认文件名为exportClass.xlsx），点击保存即可；保存成功后浏览器底部会出现类模板文件。



类模板的导入

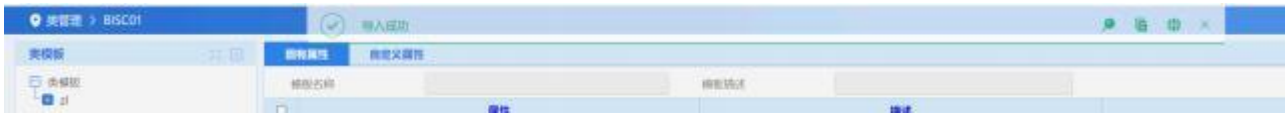
1. 右键类模板，弹出菜单有导入选项。

2. 点击导出选项，弹出导入弹窗；选择全量或增量导入，点击选择文件，选择需要导入的类模板文件，再点击确定。



- 增量导入和全量导入区别：
  - 全量导入：清除系统里的类模板，重新完整导入表格里的类模板
  - 增量导入：只对增量部分导入，同名类模板属性不同时，用表格里的新属性覆盖原来的类模板属性

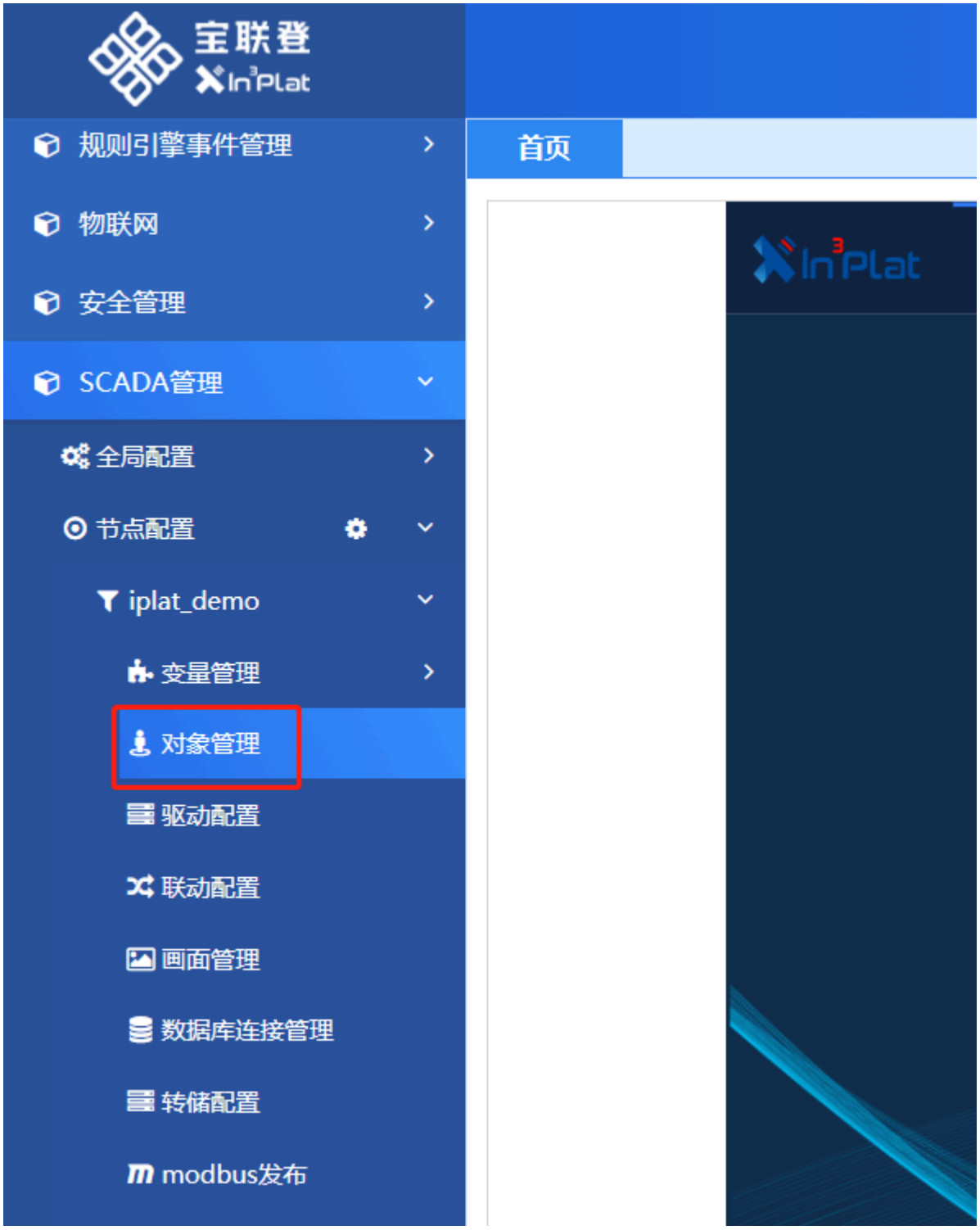
3. 顶部弹出导入成功提示，导入完成。



3.20 添加子系统

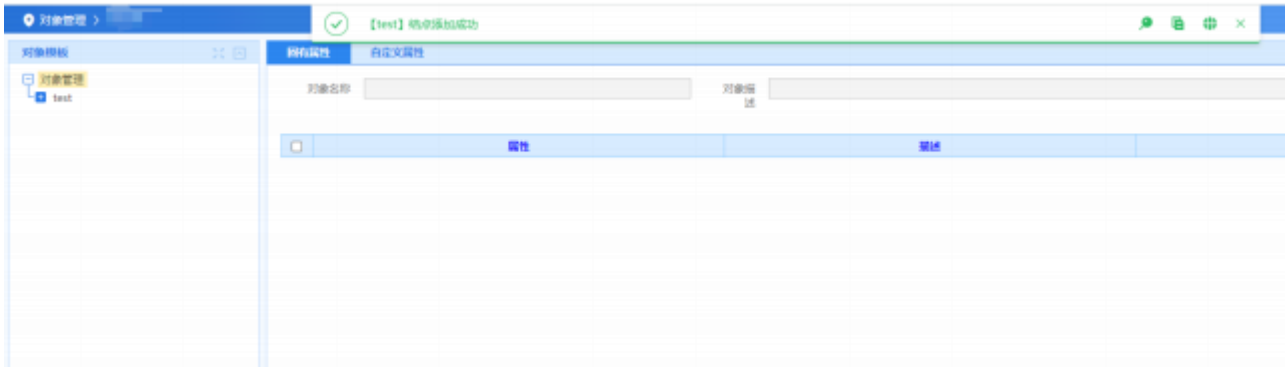
步骤演示

1. 在首页左侧栏中--->SCADA管理--->节点配置--->对象管理。

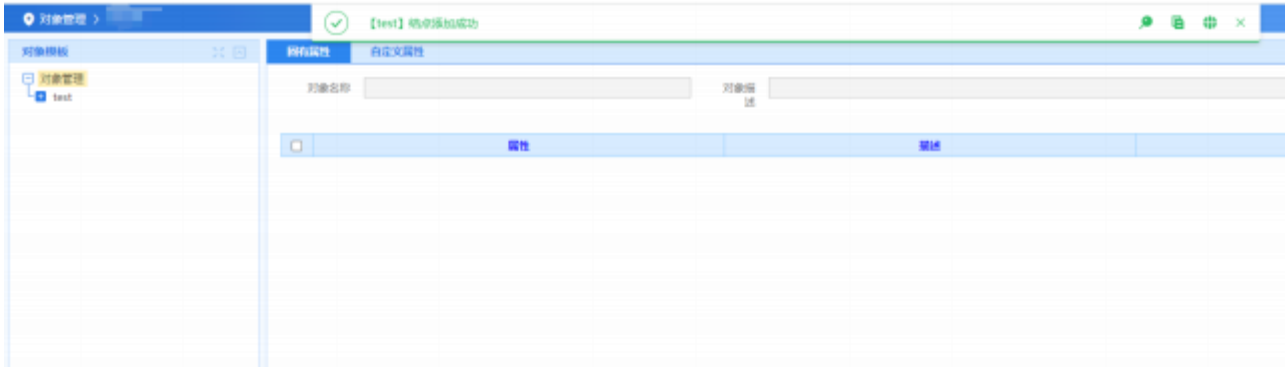


2. 单击对象管理，进入对象管理页面，右键点击新增子系统。

3. 添加必输入的名称，描述可输入可不输，比如此处输入"test"，点击确定。



4. 添加子系统成功。

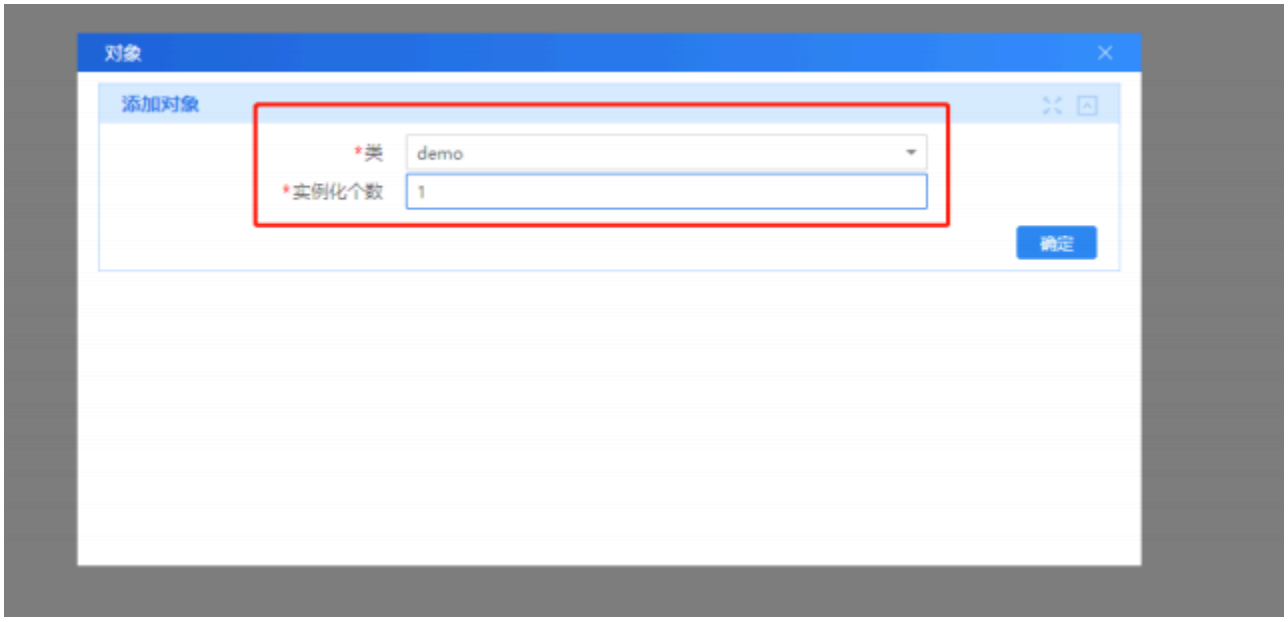


3.21 创建对象

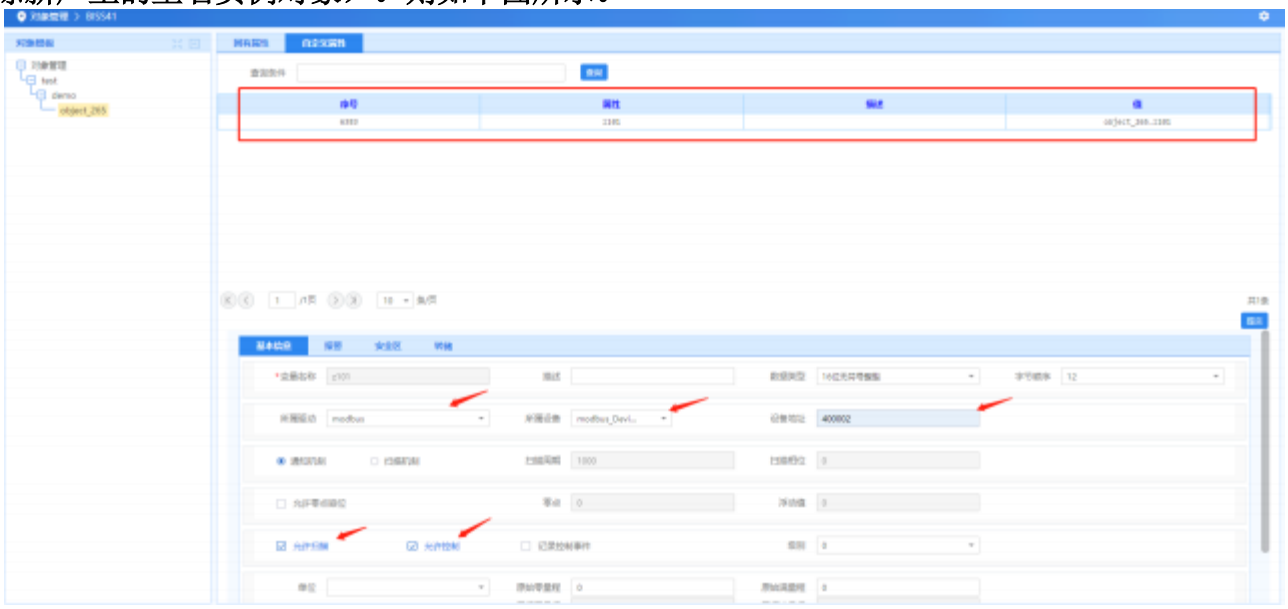
步骤演示

1. 在新增的子系统名称上，右键，可新增对象。

2. 找到所属的类，填写实例化个数。



3.选择之前创好的类（以之前创建类为demo为例），然后填入实例化个数。填入的实例化个数为1个（如果新产生的类文件夹下的对象名与已有的其他类文件夹对象名有重复的，需删除新产生的重名实例对象）。则如下图所示。



- 四个tab页同样是单击单元行弹出。这里为直接演示直观效果，就只需配好基本信息这一个tab页的红色箭头指向的基本信息。
- 每一个object对象的每一个属性都需要配基本信息，而且这些基本信息的设备地址这个字段的值不能重复，且有规定在某一个区间。
- 具体可以参看modbus驱动配完基本信息之后，点击提交按钮。

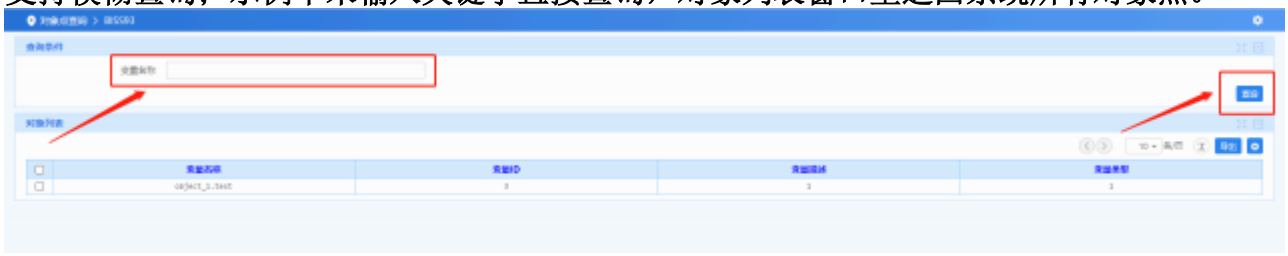
一个类模板的简单配置就基本完成了。

3.22 对象变量查询

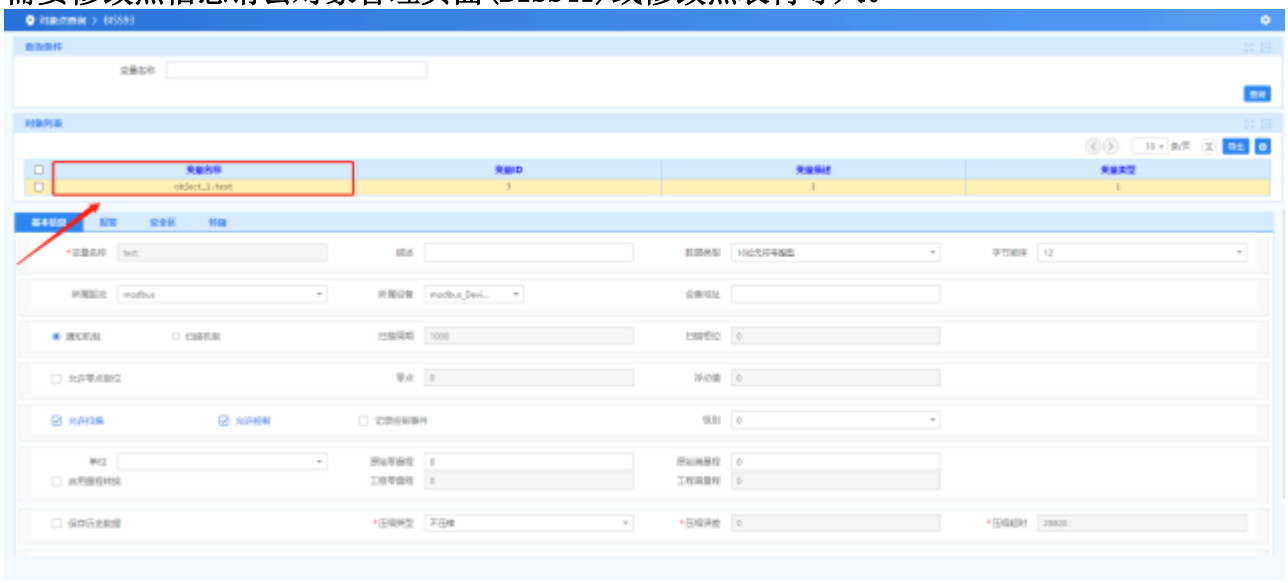
步骤演示

1. 通过【SCADA管理】→【节点配置】→节点名(示例中的iplat_demo)→【对象变量查询】。

2. 对象点查询页面中，变量名称可以输入关键字，然后点击右侧的查询按钮，进行点名查询，
支持模糊查询；示例中未输入关键字直接查询，对象列表窗口里返回系统所有对象点。



3. 左键单击单元行,弹出点的4个属性tab页;此处的属性tab页仅提供查看功能,不可修改;
需要修改点信息请去对象管理页面(BISS41)或修改点表再导入。



3.23 对象点使用示例

我们使用对象点描述2台或者N台电机的参数页面，完成后的优点：

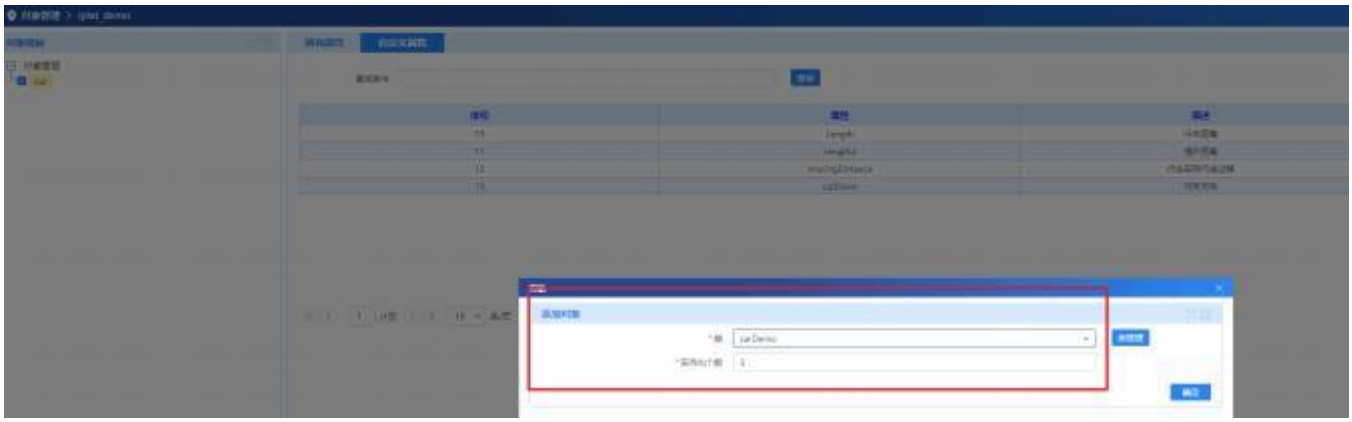
1. 不同电机不用绑点，直接绑定对象名，就可实现电机参数的展示；
2. 只需要维护点表或者点地址即可；

创建电机参数类模板

按照类模板创建步骤创建电机类模板及三个电机自定义属性：【电流】、【转速】、【转矩】

实例化对象及建点

添加对象，选择类模板并实例化，（有多少电机就实例化多少个）

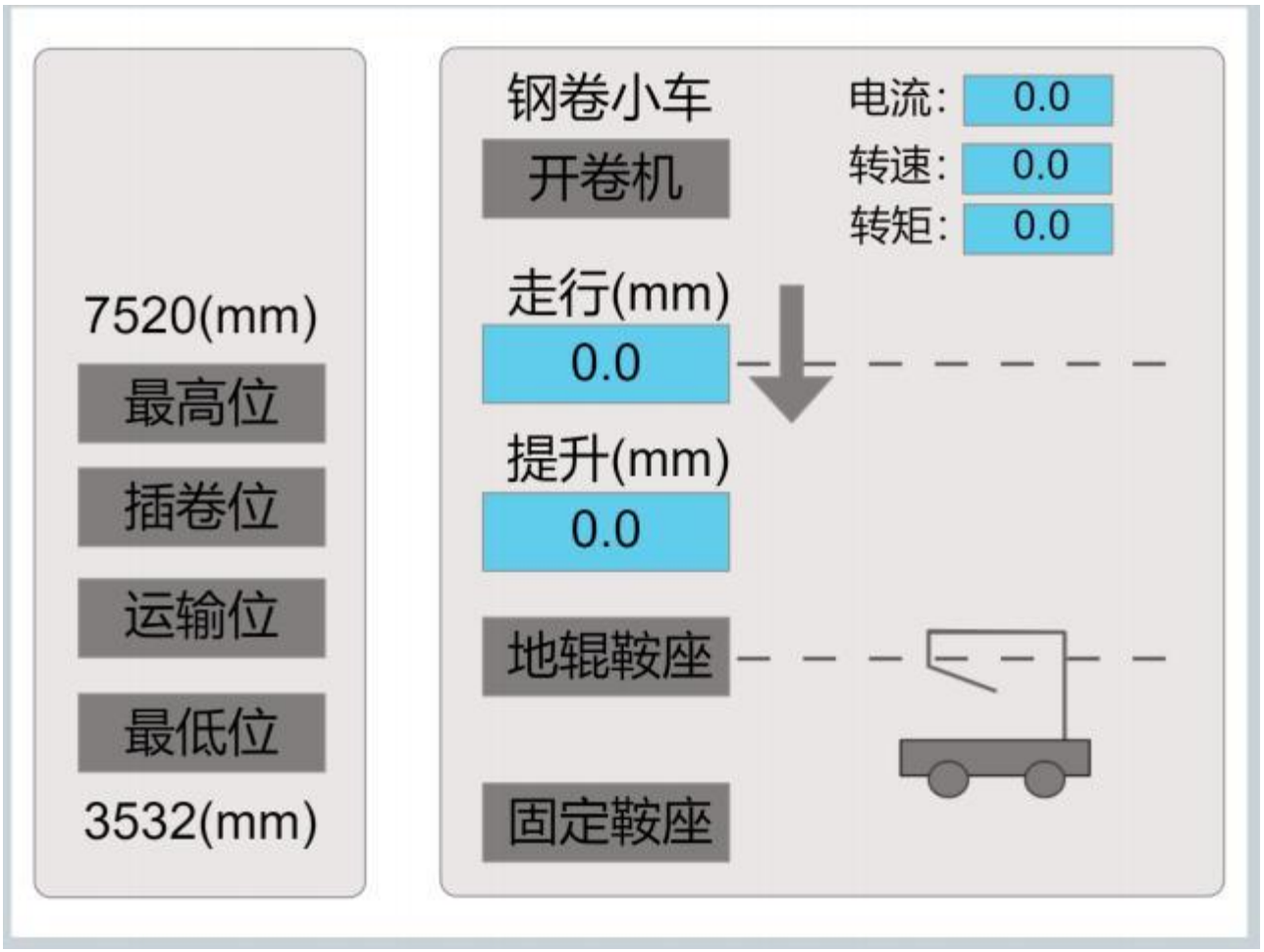


对象的自定义属性点完善

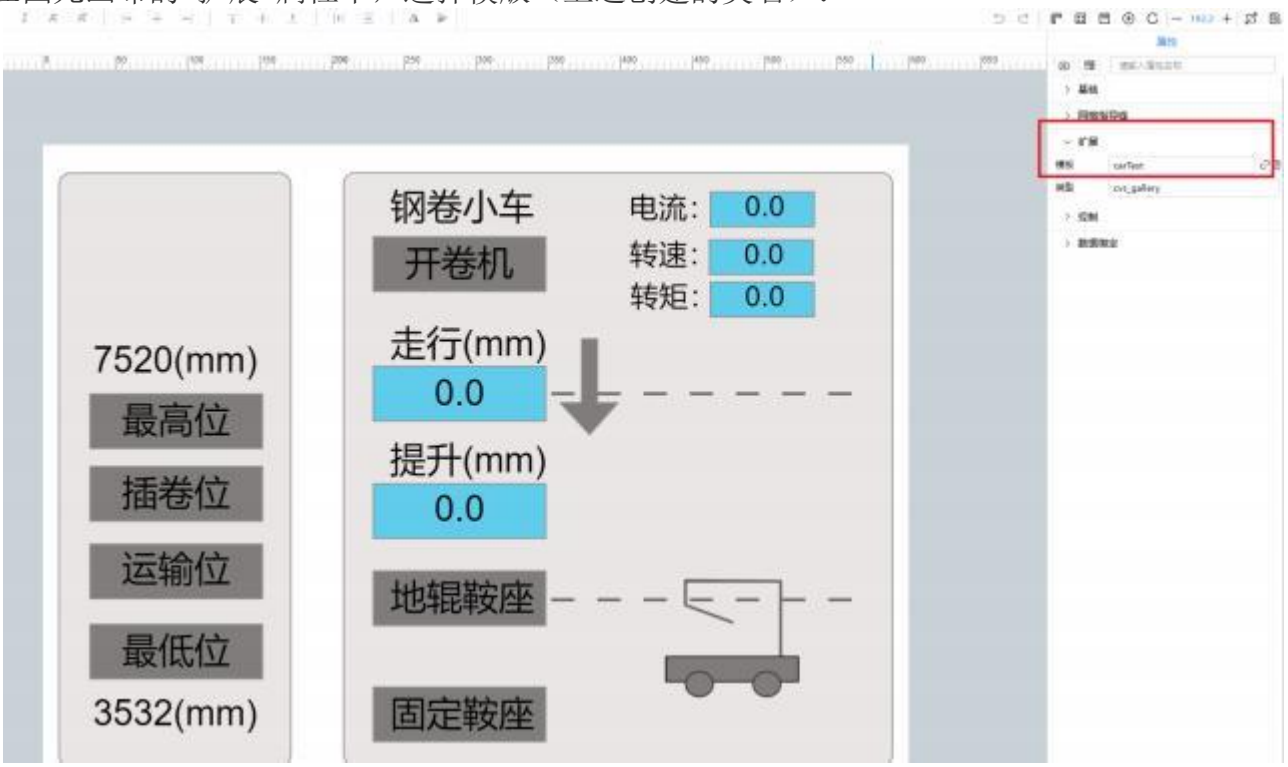
注意
可利用Excel点表完善所有点的信息，然后部署节点完成点的创建。

创建电机参数图元

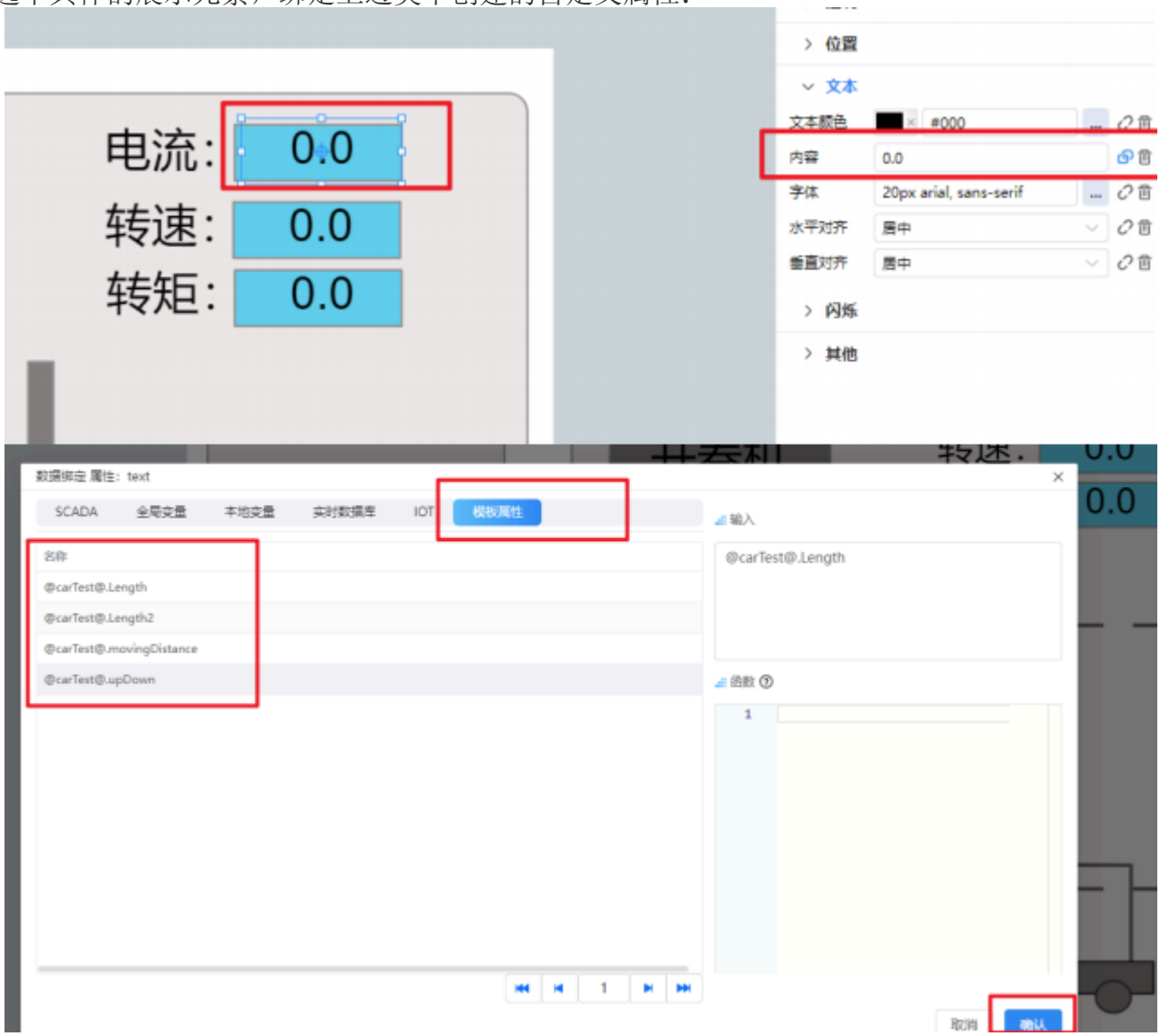
•在图元设计器中，编辑好展示图元：



•在图元画布的"扩展"属性中，选择模版（上边创建的类名）：

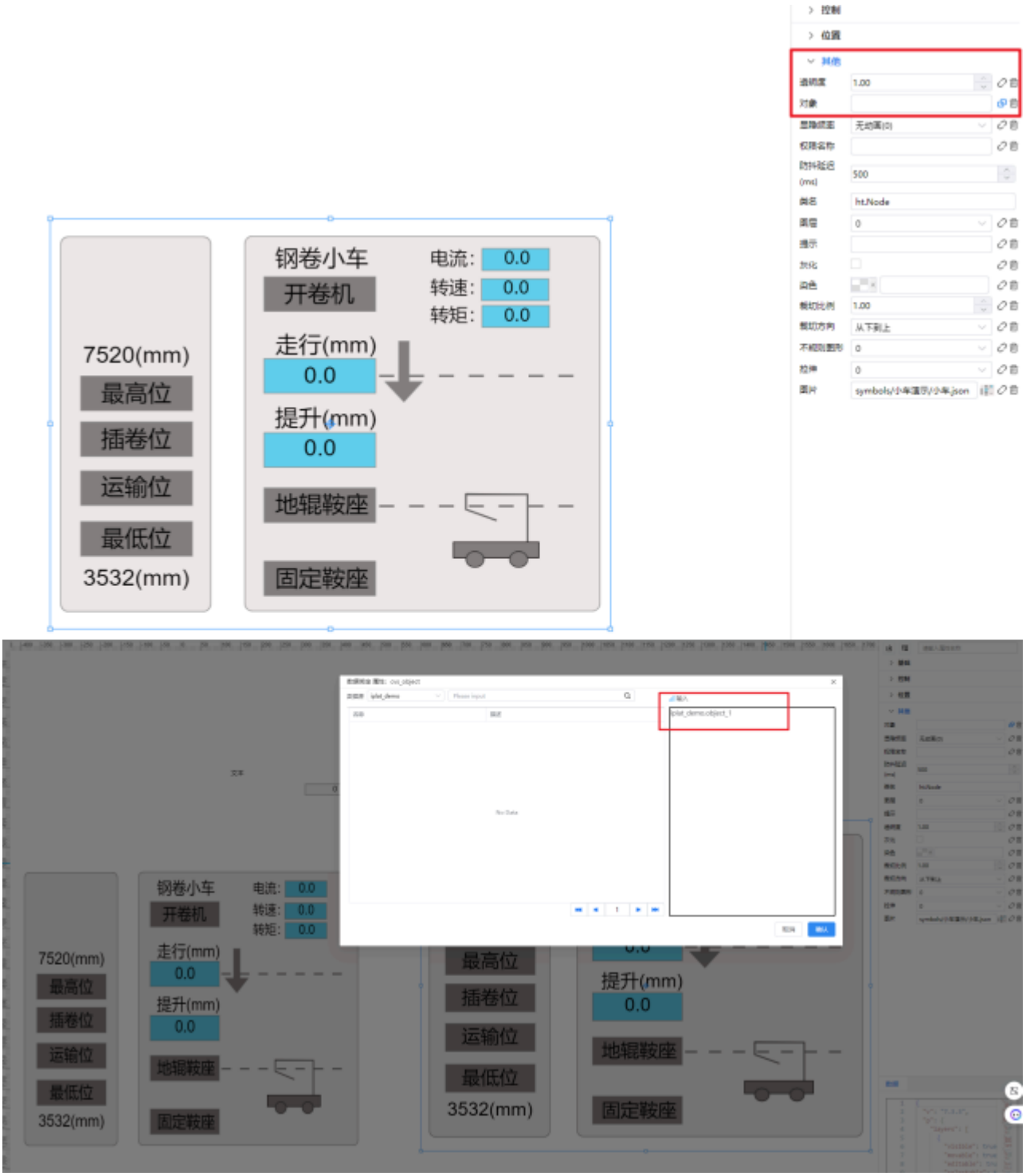


•选中具体的展示元素，绑定上边类中创建的自定义属性：



画面上使用图元并绑定对象

•保存并签入之后，将图元**拖到画面**上，绑定具体的对象即可：



这样我们就可以直接通过，绑定不同的电机对象，实现图元内部点的直接绑定，不需要再去单独绑点了！

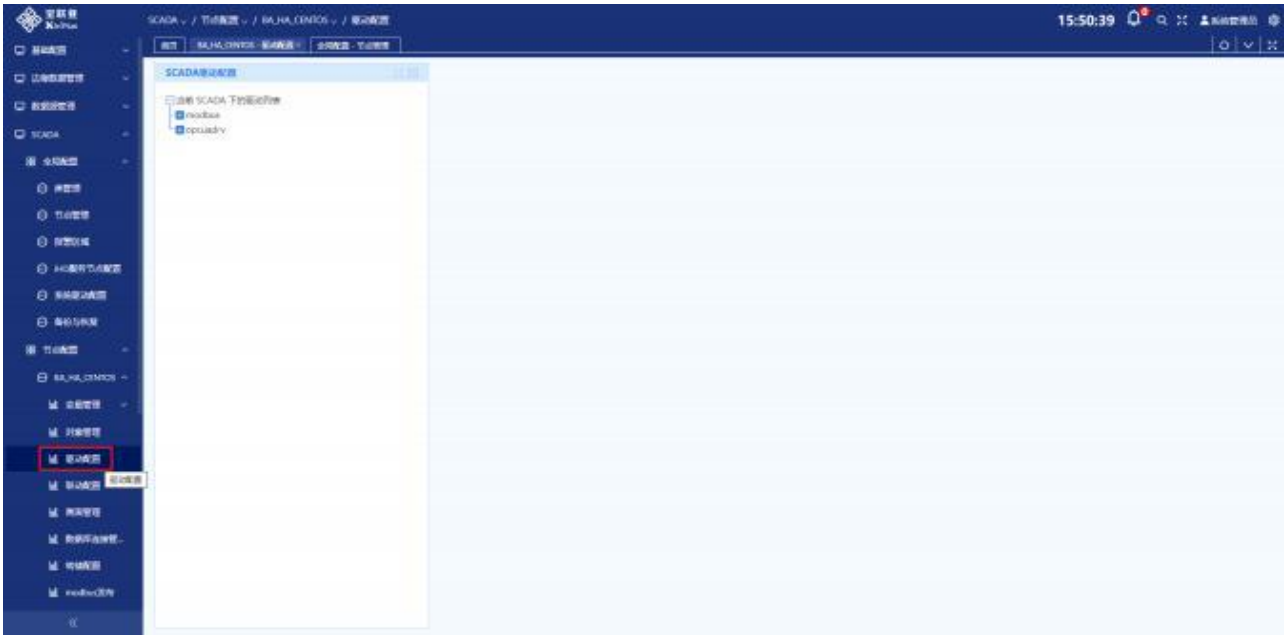
图元下载：

下载数据文件(小车.zip)

- 添加MidVarDriver
- 配置MidVarDriver变量

添加MidVarDriver

1. 首先，需要配置好相应的SCADA节点。MidVarDriver不对应到设备，而用于配置中间变量，主要使用于一些内部值的传递。不与实际设备进行交互。
   点击SCADA节点下的驱动配置，进入驱动配置页面。



2. 右键点击【当前SCADA下的驱动列表】添加驱动弹出可用驱动的选择对话框，在右侧显示的【驱动信息】页面中，勾选MidVarDriver驱动所对应的"MidVarDriver"，然后点击保存。

3. MidVar驱动不需要添加设备。

3.24 配置MidVarDriver变量

点击新增后在如下界面配置所需点的相关信息，配好后点击右上角"应用"按钮，则MidVarDriver点配置完成

注意：配置完成后需要**部署**到SCADA节点，使得配置生效。MidVar点初始值默认为"0"，使用时根据业务需求更改点值，重新部署后点值不受影响，保持部署前最后一次更改的点值。

•为本地变量写值
•绑定本地变量

3.25 为本地变量写值

说明：

- 本地变量是只存在于当前客户端浏览器的变量，其更新不会下发到服务器和其他客户端
- 本地变量是以**$local.**开头的变量，在使用前，需要之前有过设置值，才能在绑点中看见，如使用 sys.setLocalTag(tagname, value)；在脚本编辑器中给点设置过初始值

•使用sys.setLocalTag

sys.setLocalTag("$local.PWD","123");

将添加或设置一个本地变量。

不建议使用，建议使用sys.writeValue

•使用sys.writeValue

•sys.writeValue("$local.PWD","456");

sys.writeValue现在支持写一个本地变量，调用后将添加或设置一个本地变量的值

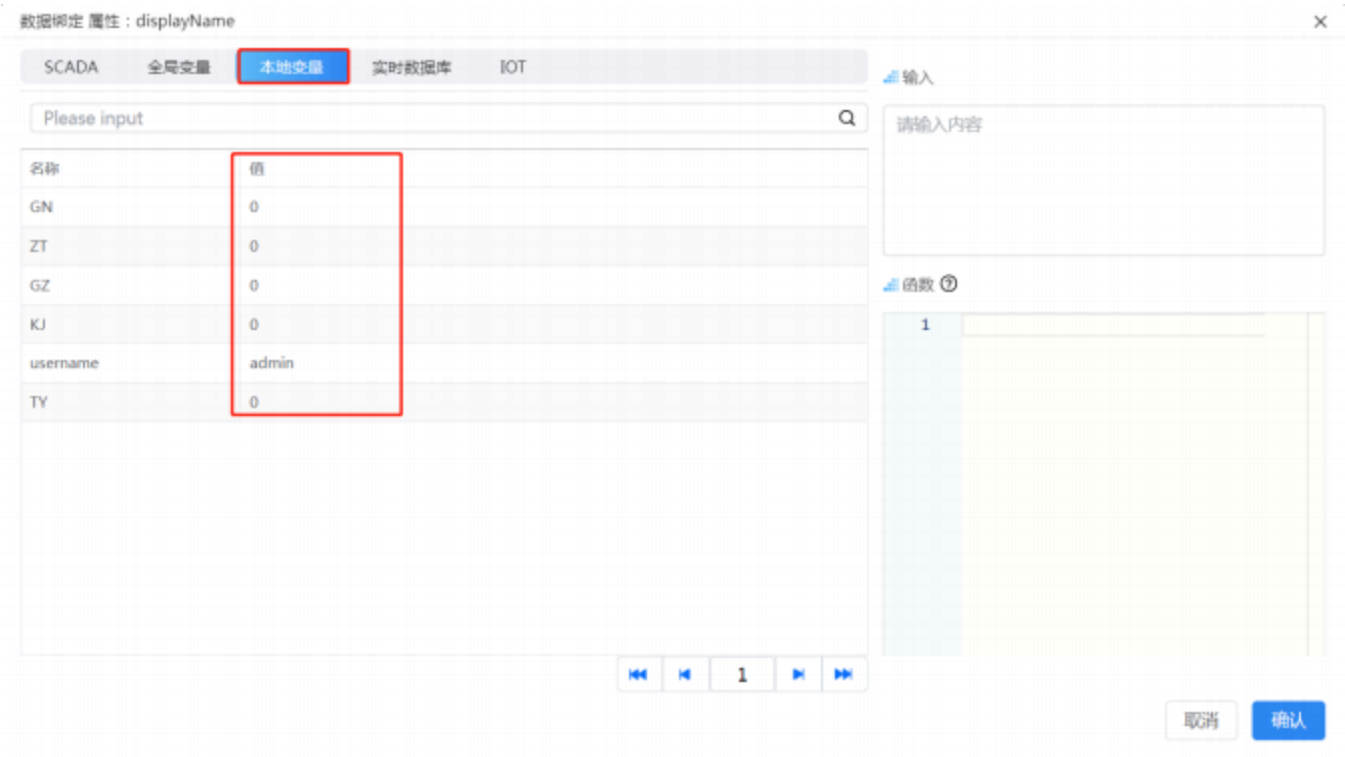
•使用sys.WriteMultiValues

sys.WriteMultiValues现在支持批量写本地变量，和全局变量及scada变量一样可以在一个批量写值中使用

3.26 绑定本地变量

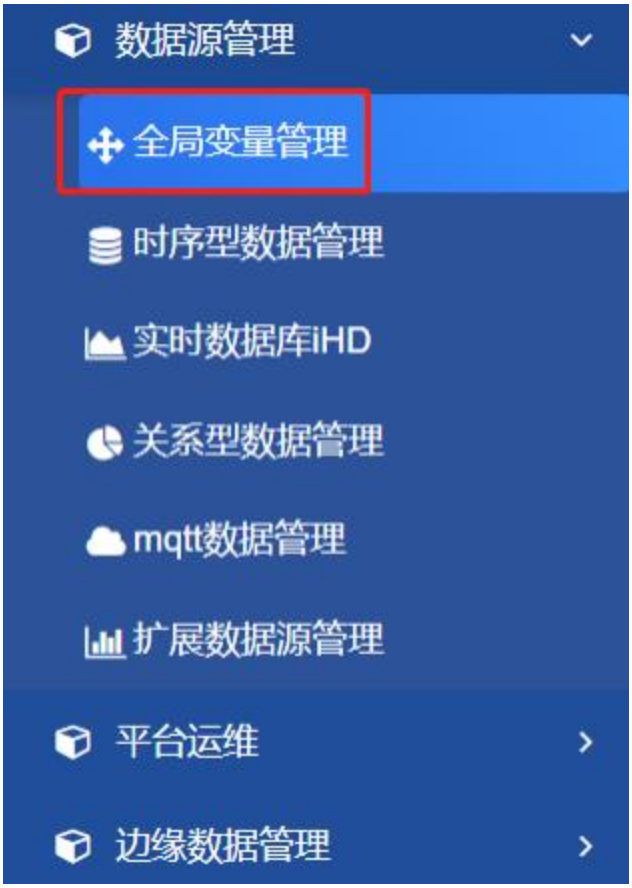
•绑定本地变量和其他点的使用方式相同，对于同一个浏览器客户端的多个客户端来说，其中一个页面更新了本地变量，该页面所在浏览器的另一个页面（包含弹窗、子页面等）若绑定该本地变量，都会监听到值的变化，且更新绑点相关属性

•本地变量的使用方式和其他点完全相同，同一电脑的同一浏览器的不同页面，本地变量是互通的，能够监听到本地变量的变化；区别在于，它是客户端隔离的，即不同客户端和电脑的本地变量之间不互通



3.27　全局变量的配置

1.通过【数据源管理】→【全局变量管理】进入BIDV01全局变量配置界面。

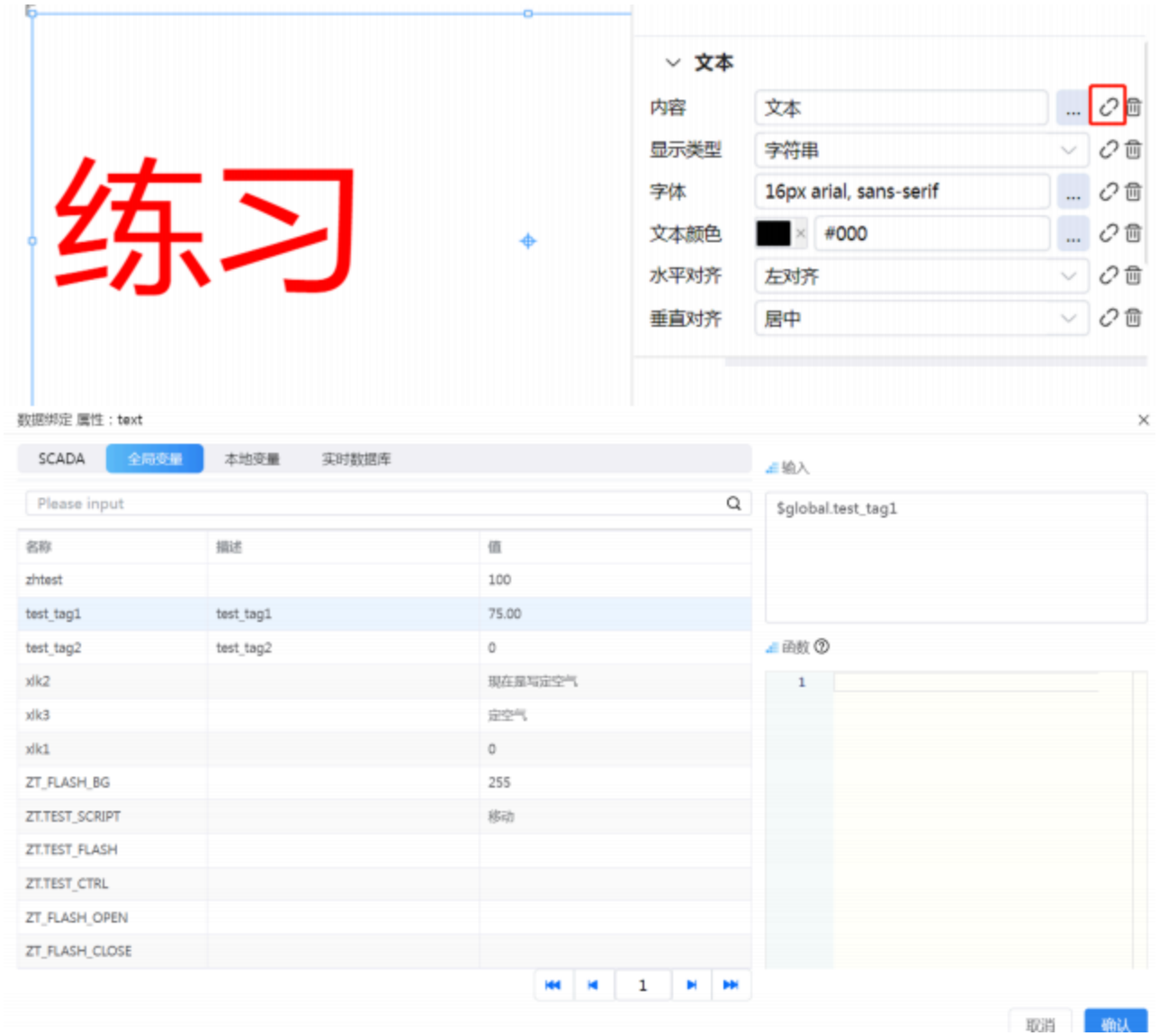


2.配置全局变量。

- 新增：点击全局变量配置左上角【新增】，可以新增一个全局变量tag点，点值的类型为string
- 修改：选中需要修改的tag记录，**tag点名、值均可以修改。**
- 删除：选择需要删除的记录，点击删除即可。
- 查询：在查询条件中输入TAG名称，点击查询可以查询tag点

3. 全局变量的使用

1. 全局变量可以在页面中正常使用，在选点界面的全局变量标签页选择需要关联的tag点即可。



2. 效果如下：

注意：如果用户没有配置"所属驱动"、"设备名称"和"设备地址"，则添加的变量默认为全局变量，全局变量可以在所有客户端中通用。

4.全局变量的读值和写值

1.全局变量支持在BIDV01页面进行查看和修改；

2.脚本对tag值进行读取和修改；

3.28 系统变量的分类

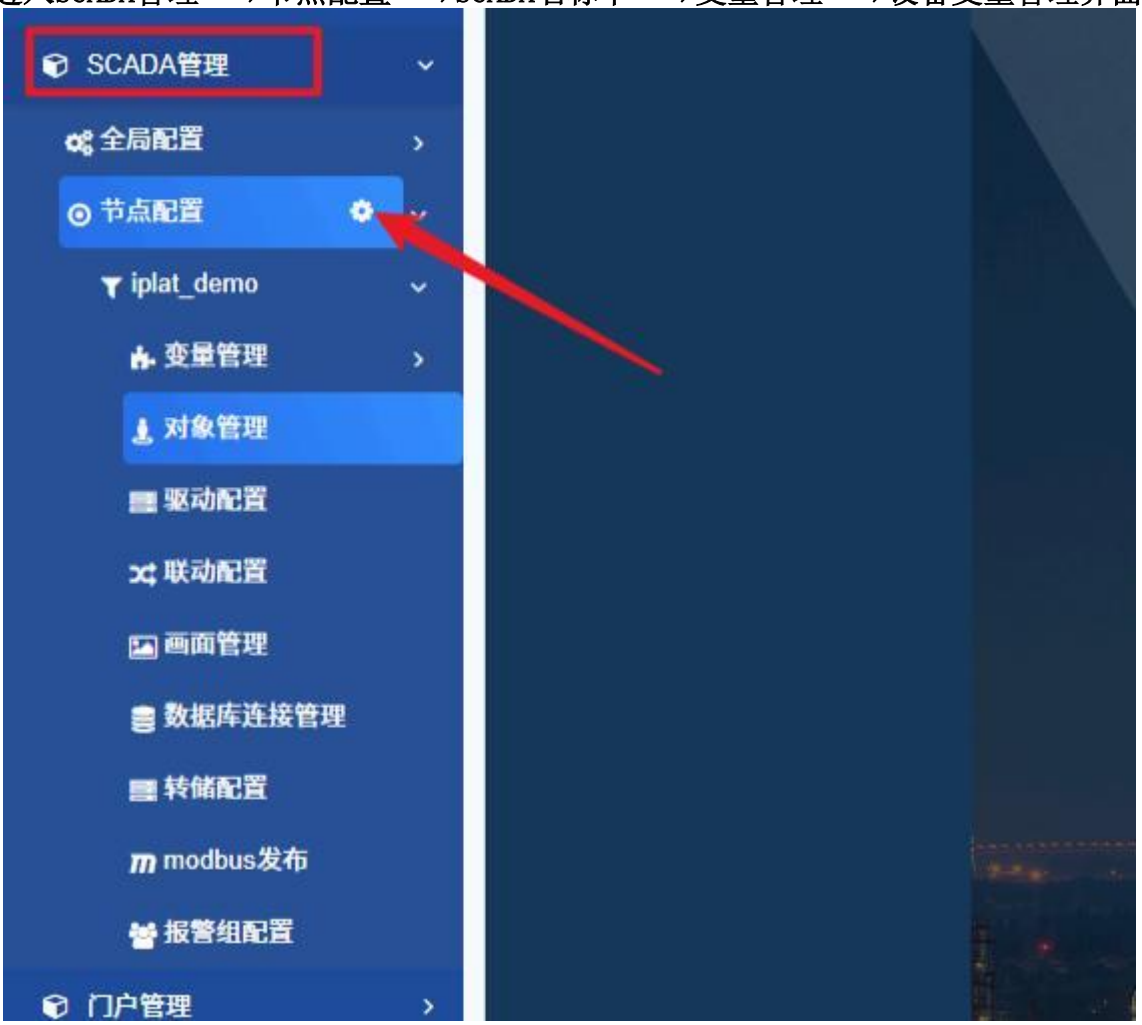
•分类

名称	描述
SYSTEM.ALM	scada下总的报警个数
SYSTEM.ALMFILTER	报警过滤信息
SYSTEM.DEBUG	da内部调试用的tag点
SYSTEM.LICENSE	许可证信息
SYSTEM.MidVarDriver	驱动状态点
SYSTEM.RMSTATUS	冗余状态点
SYSTEM.modbus	驱动状态点
SYSTEM.modbus_modbus_Device0	驱动设备状态点

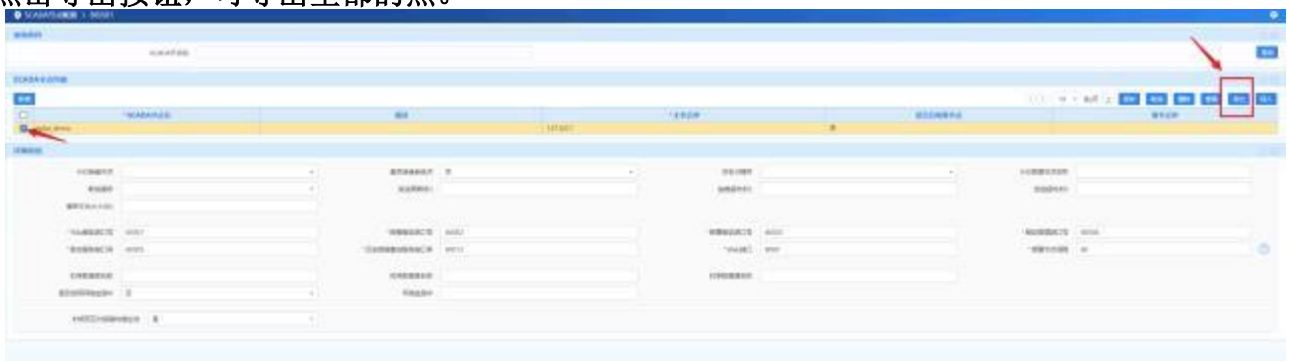
EXCEL批量配置点表

步骤演示

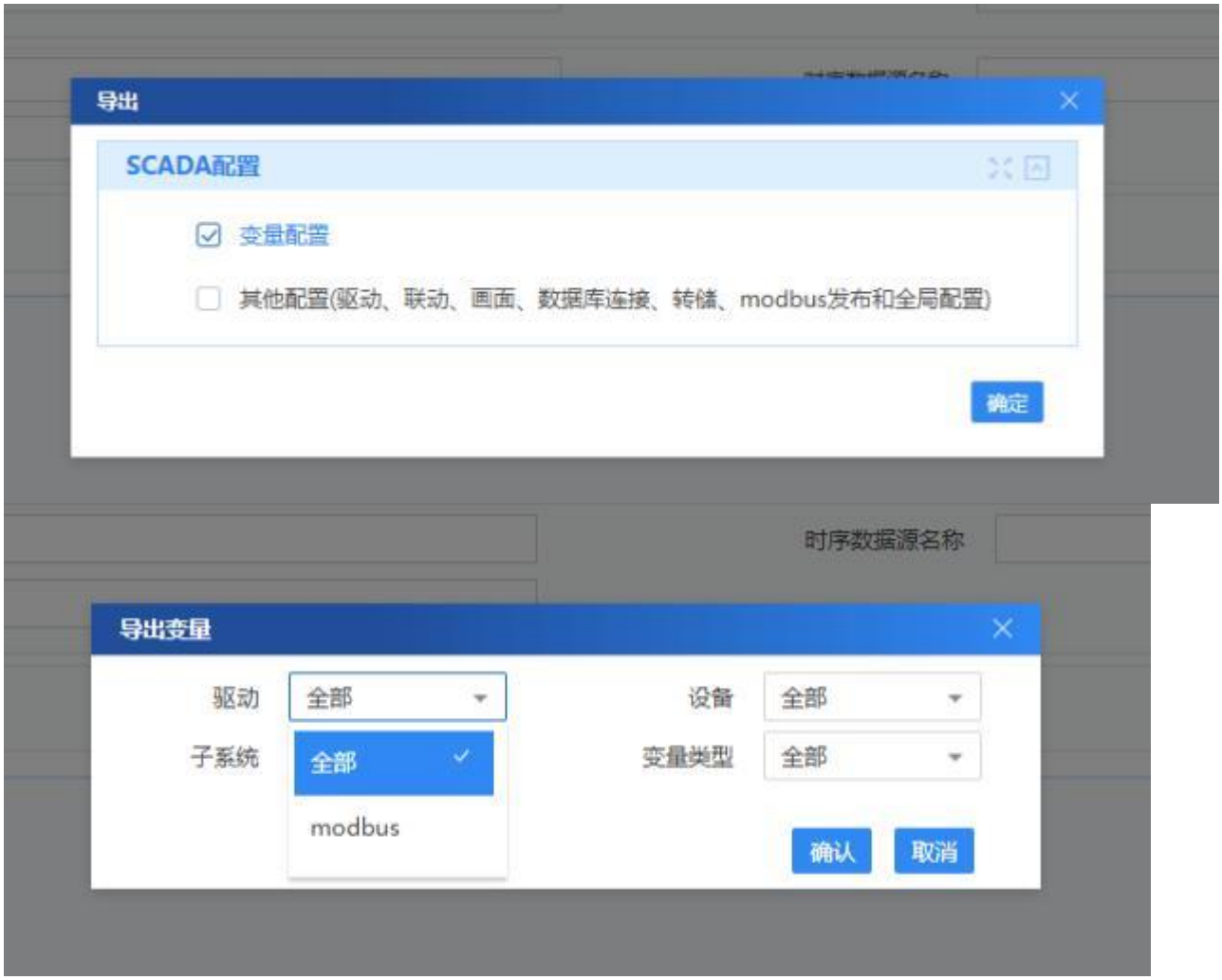
1. 进入SCADA管理--->节点配置--->SCADA名称下--->变量管理--->设备变量管理界面。



2. 点击导出按钮，可导出全部的点。



3. 可以筛选导出内容



4. 页面会出现导出的excel，则导出成功。
5. 导出后，可在下图excel中创建点。

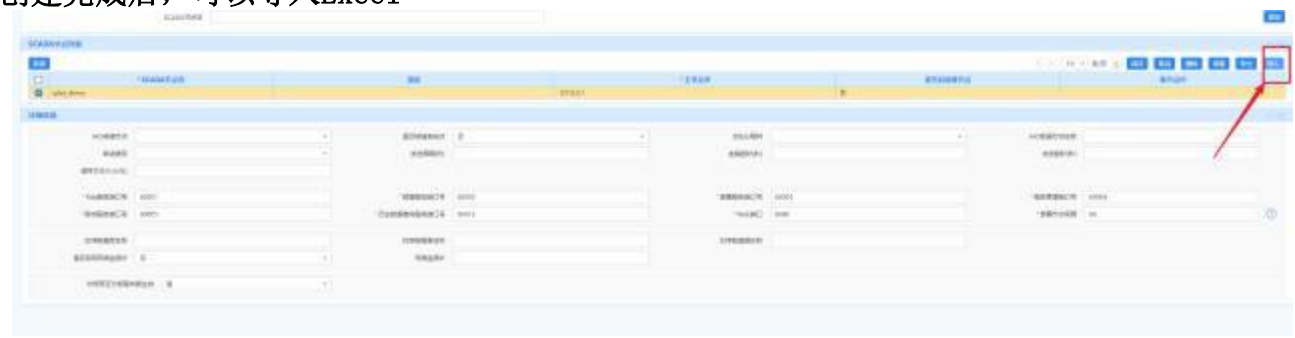
	A	B	C	D	E	F	G
1	变量名称	变量类型	变量描述	数据类型	所属驱动	设备名称	设备地址
2	demoUse1	1		2	modbus	modbus_Device0	400001
3	demoUse11	1	midvar点	7	MidVarDriver		
4	demoUse12	1	u	2	modbus	modbus_Device0	400011
5	demoUse13	1		2	modbus	modbus_Device0	400012
6	demoUse14	1		2	modbus	modbus_Device0	400013
7	demoUse15	1		2	modbus	modbus_Device0	400014
8	demoUse16	1		2	modbus	modbus_Device0	400015
9	demouse17	1		2	modbus	modbus_Device0	400016
10	demouse18	1		2	modbus	modbus_Device0	400017
11	demouse19	1		2	modbus	modbus_Device0	400018
12	demoUse2	1		2	modbus	modbus_Device0	400002
13	demouse20	1		2	modbus	modbus_Device0	400020
14	demouse21	1		2	modbus	modbus_Device0	400021
15	demouse22	1		2	modbus	modbus_Device0	400022
16	demouse23	1		2	modbus	modbus_Device0	400023
17	demouse24	1		2	modbus	modbus_Device0	400024
18	demouse25	1		2	modbus	modbus_Device0	400025
19	demouse26	1		2	modbus	modbus_Device0	400026
20	demouse27	1		2	modbus	modbus_Device0	400027
21	demouse28	1		2	modbus	modbus_Device0	400028
22	demouse29	1		2	modbus	modbus_Device0	400029
23	demoUse3	1		2	modbus	modbus_Device0	400003
24	demouse30	1		2	modbus	modbus_Device0	400030
25	demouse31	1		2	modbus	modbus_Device0	400031
26	demouse32	1		2	modbus	modbus_Device0	400032
27	demoUse4	1		2	modbus	modbus_Device0	400004

- 点表中只有一个"table"sheet表
- 变量名称：变量的名字，用于标识Tag点。因为变量可以用于iplat系统中脚本的编写，因此，命名变量名称需根据一定的规则：
  - 只能包含中文、数字、英文字母或下划线"_"
  - 名字不能为空，长度不超过64（含结尾"\0"
  - 不能以数字或"_"开头，不可全数字组成
  - 名称不能等于表达式里面的关键字
    （
    "in","max","min","sin","cos","tan","abs","asin","acos","atan","log","ceil","pow","sq rt","eq","neq","conret","strconcat","todecimal","int","exp","fmod"等）
  - 类的自定义属性不能以（F_或A_）开头。
  - 类的自定义属性不能以数字或"_"开头，不可全数字组成
  - 变量名称的第二部分不能以（F_或A_）开头
- 变量类型：按照AI、AO、BLOB、CA、DI、DO、LA、MDI、TXT、CD的顺序显示选择的是第几个
- 描述：对变量的补充说明

- 数据类型：配置的变量所用的类型，DI变量不用选择，AI是模拟量，有浮点型、整型等各种类型需要选择，EXCEL批量配置时需根据变量数据类型选择相应数字填充至相应列
  - 16位无符号整型------2
  - 16位有符号整型------1
  - 32位无符号整型------6
  - 32位有符号整型------7
  - 8位无符号整型------11
  - 8位有符号整型------10
  - 单精度浮点型------3
  - 双精度浮点型------8

- 所属驱动：选择所新增的这个变量所要采集的数据，来自哪种驱动
- 设备名称：选择所新增的这个变量所要采集的驱动下对应的设备
- 设备地址：填写新增的这个变量的寄存器具体地址。如果选择了是驱动协议是免块方式的配置，请直接填写寄存器地址，如果选择的是驱动中配置数据块的方式，请按照"数据块：寄存器地址"的格式填写

6. **创建完成后，可以导入Excel**



7. **导入时可进行模式选择**

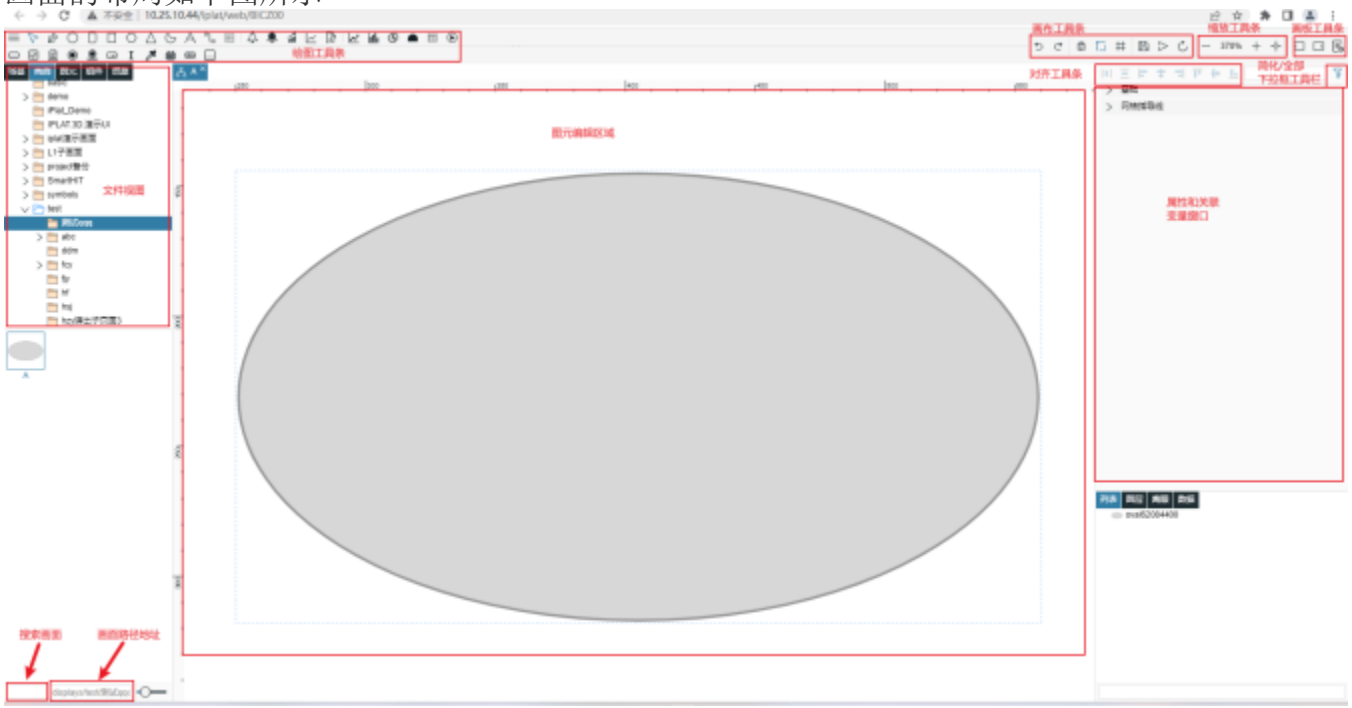
1. 增量导入：新增没有的、覆盖已有的；对相同的tag名的点，进行覆盖，对元来没有的点进行新增
2. 全量导入：删除原来的点表，再导入新点表
3. 循环导入条数：导入时没次导入的数量，可用于定位导入时的点表错误

4. 组态与画面

5.1设计器基本功能

画面布局

画面的布局如下图所示



快捷菜单

点击展开后，功能见图

- 自定义工具栏：
  选择后，弹出自定义工具界面

用户可以

- 工具切换栏，进行工具模式切换：

可以切换，工具栏的位置，其中模式一，如布局图所示，模式二如下：

•对齐方式设置

以位置对齐：例如左对齐，会以最左端元素对齐；顶对齐会以最上方元素对齐
以最后选中元素对齐：例如左对齐，会以最后选择的元素对齐；顶对齐会以最后选择元素对齐

•查看默认菜单：见上图
•变量查询与替换

查询功能：可以查询当前画面中的Tag点，及tag绑定在哪个元素的属性上，支持模糊查询；
替换功能：可以替换tag点或者tag名称中的字符，支持批量替换。

•图元引用列表：

选择画面中某个图元的“标签名”，点击查询，会列出当前环境中，所有使用该图元的画面。

- 新建画面：
  新建一个画面。
- 新建图元：
  新建一个图元。
- 快速定位

  点击快捷菜单右侧该按钮，在打开画面或图元的情况下，可以快捷定位到该文件在目录中的位置

通用工具栏

包含编辑、不规则图形、常用形状、常用图元，可以点击选择元素后，再点击画布即可使用；也可以直接拖拽使用。（通过“自定义工具栏”自定义的工具会显示下标）

- 编辑

点击可回到画图状态

- 不规则图形

可绘制直线、曲线和不规则图形

- 连线

进行元素之间的连线

- 椭圆

绘制椭圆形状

- 圆矩形体

绘制圆矩形形状

- 矩形体

绘制矩形形状

- 正多边形

绘制正多边形形状

- 三角形体

绘制三角形形状

- 圆弧形

绘制圆弧形形状

- 文本

编辑文本

- 文本框

编辑文本框，支持多行文本

- 报警

实时报警控件

- 历史报警

历史报警控件

- 趋势控件

历史趋势控件

- 实时趋势

实时趋势控件

- gif图片

添加gif图片•折线图

添加折线图

•柱状图

添加柱状图

•饼图

添加饼图

•仪表盘

添加仪表盘

•组态回放

组态回放控件

•表格

表格控件

绘制工具栏

可以使用字体格式、对齐方式、反转方式等功能

•字体选择

•粗体

•斜体

•字体缩小

•字体放大

布局快捷工具

快捷工具的触发：

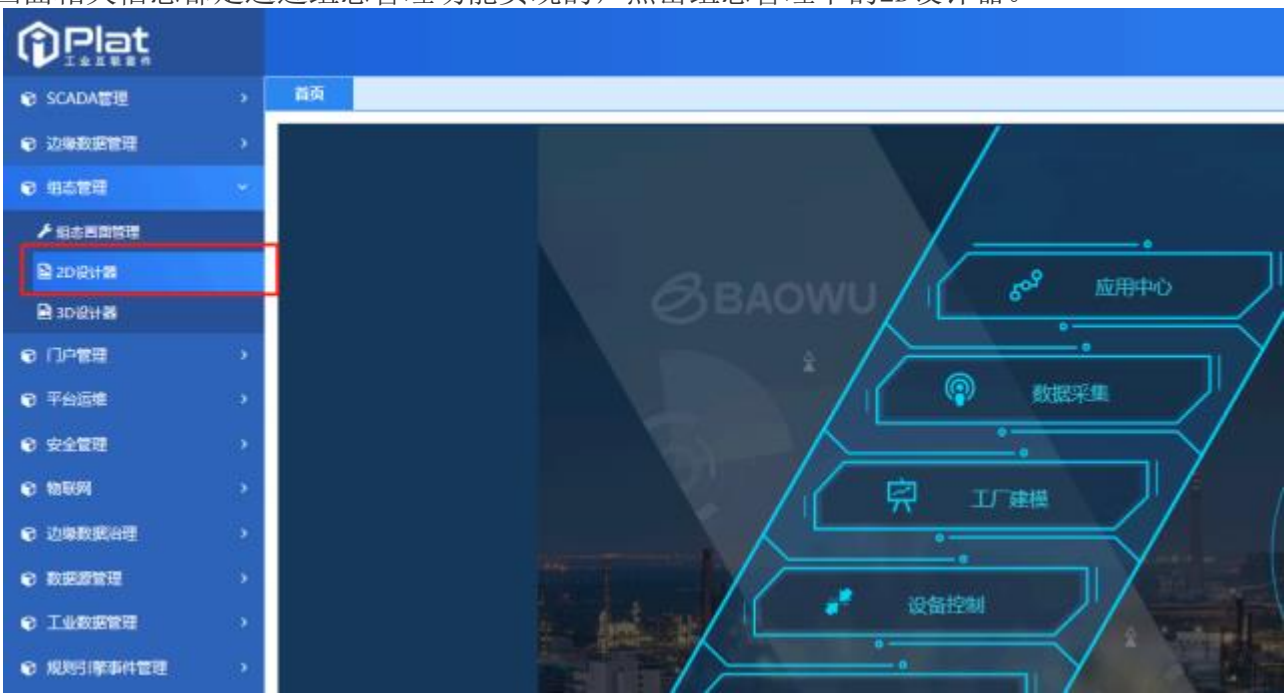
1. 选择一个文本
2. 选择有多个元素

5.2 2D画面搭建与运行

创建画面

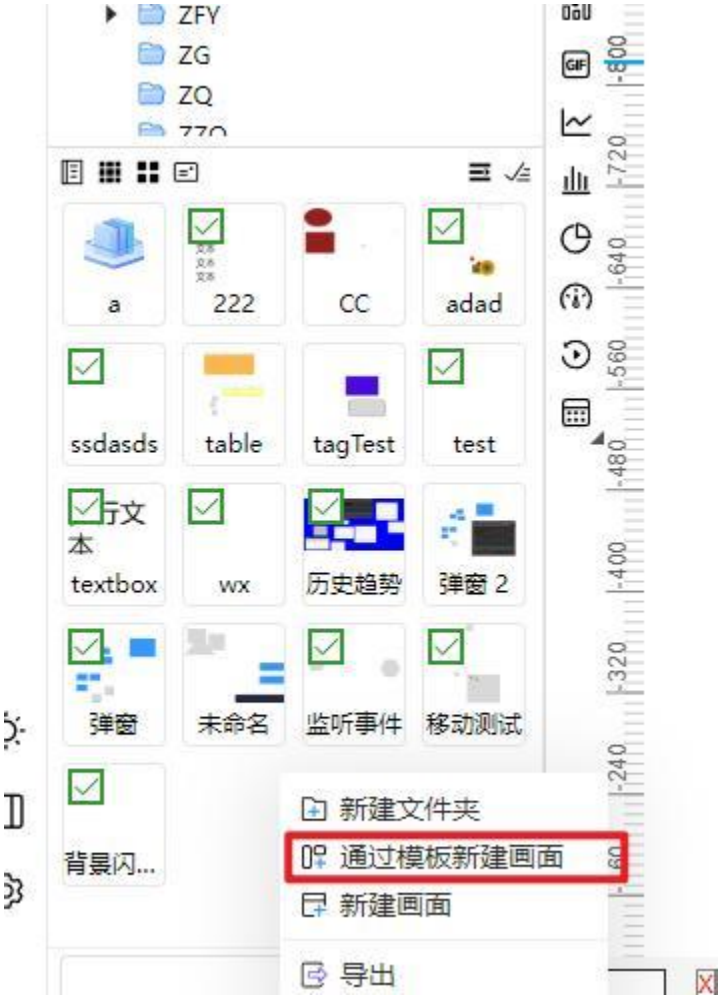
新建画面

1. 画面相关信息都是通过组态管理功能实现的，点击组态管理下的2D设计器。



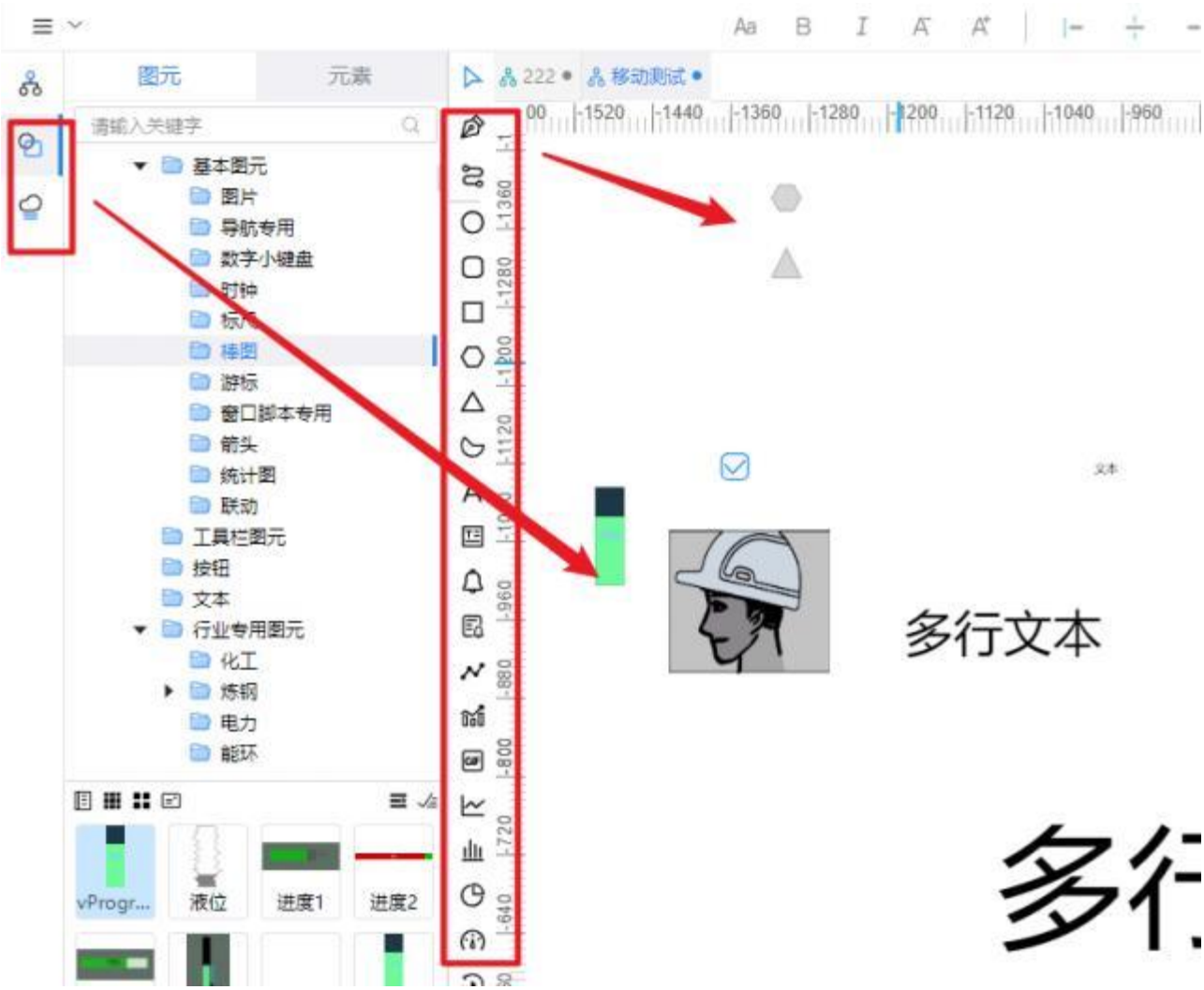
2. 进入2D设计器界面，选择画面选项卡，然后

- 选择一个目录，右键新建画面，也可以根据需要新建目录。
- 也可以选定文件夹后，在文件视图下方“新建画面”
- 也可以“通过模板新建画面”，参见画面模板功能



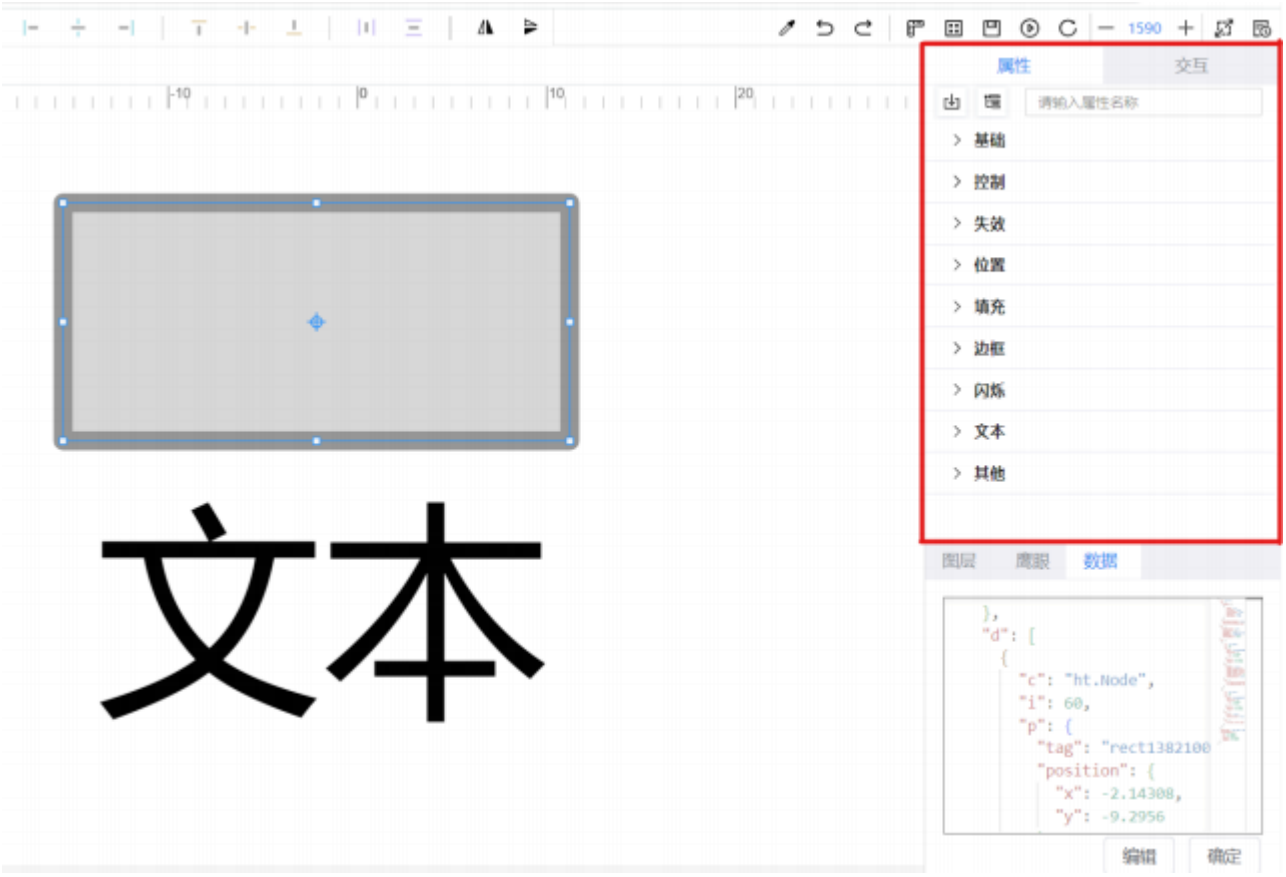
添加元素

1. 从左侧通用工具栏、图元。资源中选中元素放到画面编辑区域

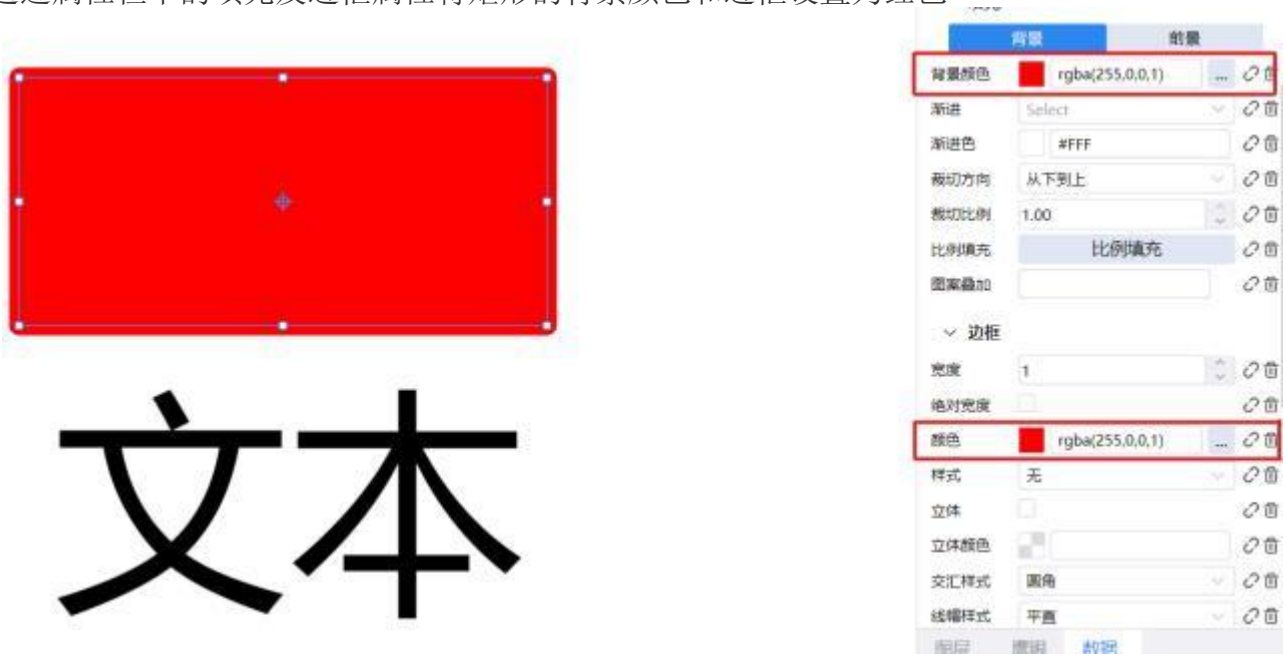


配置属性

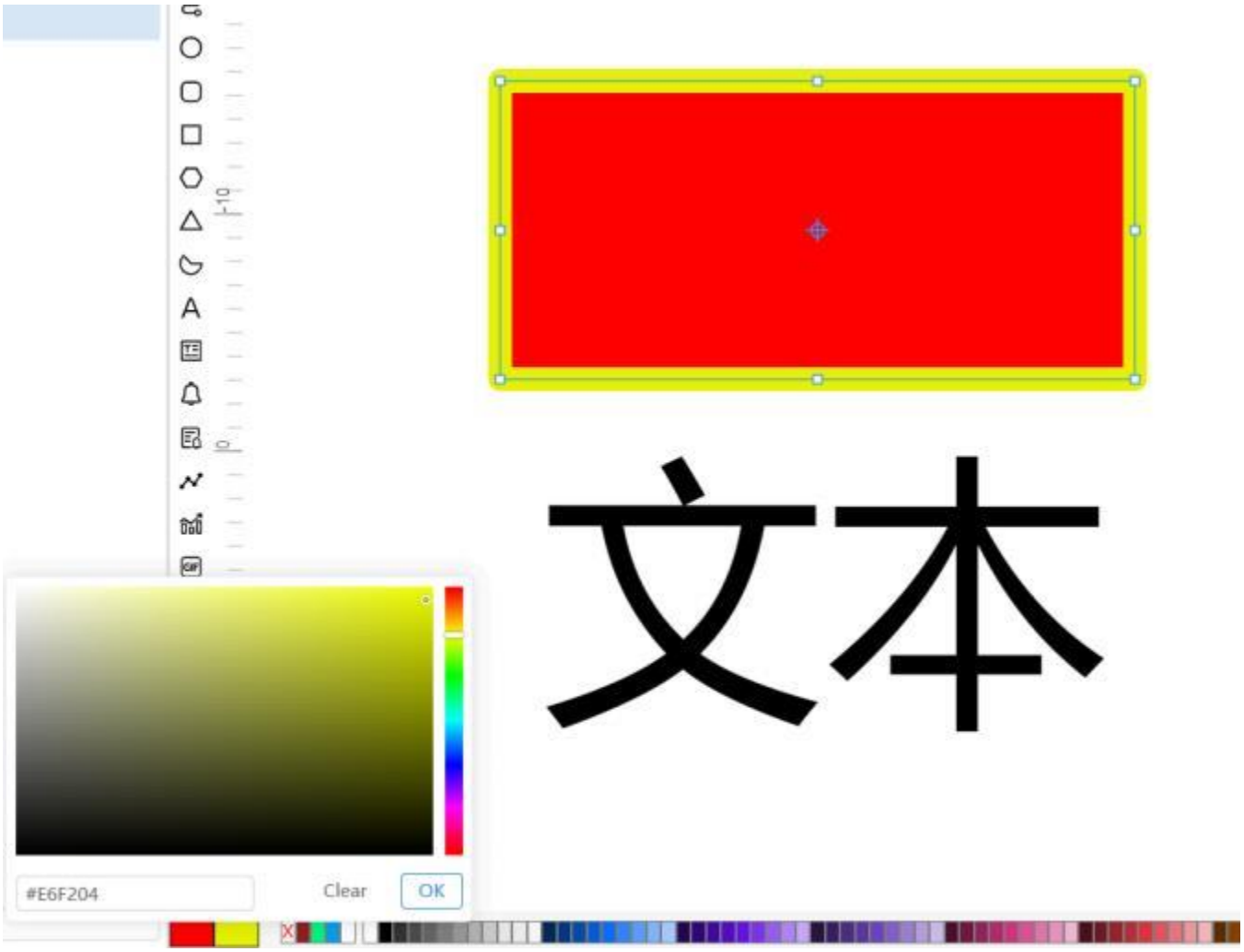
1. 可以在右侧控件的属性栏配置该元素的属性，不同元素属性不同，详见元素功能属性。



2. 通过属性栏中的填充及边框属性将矩形的背景颜色和边框设置为红色

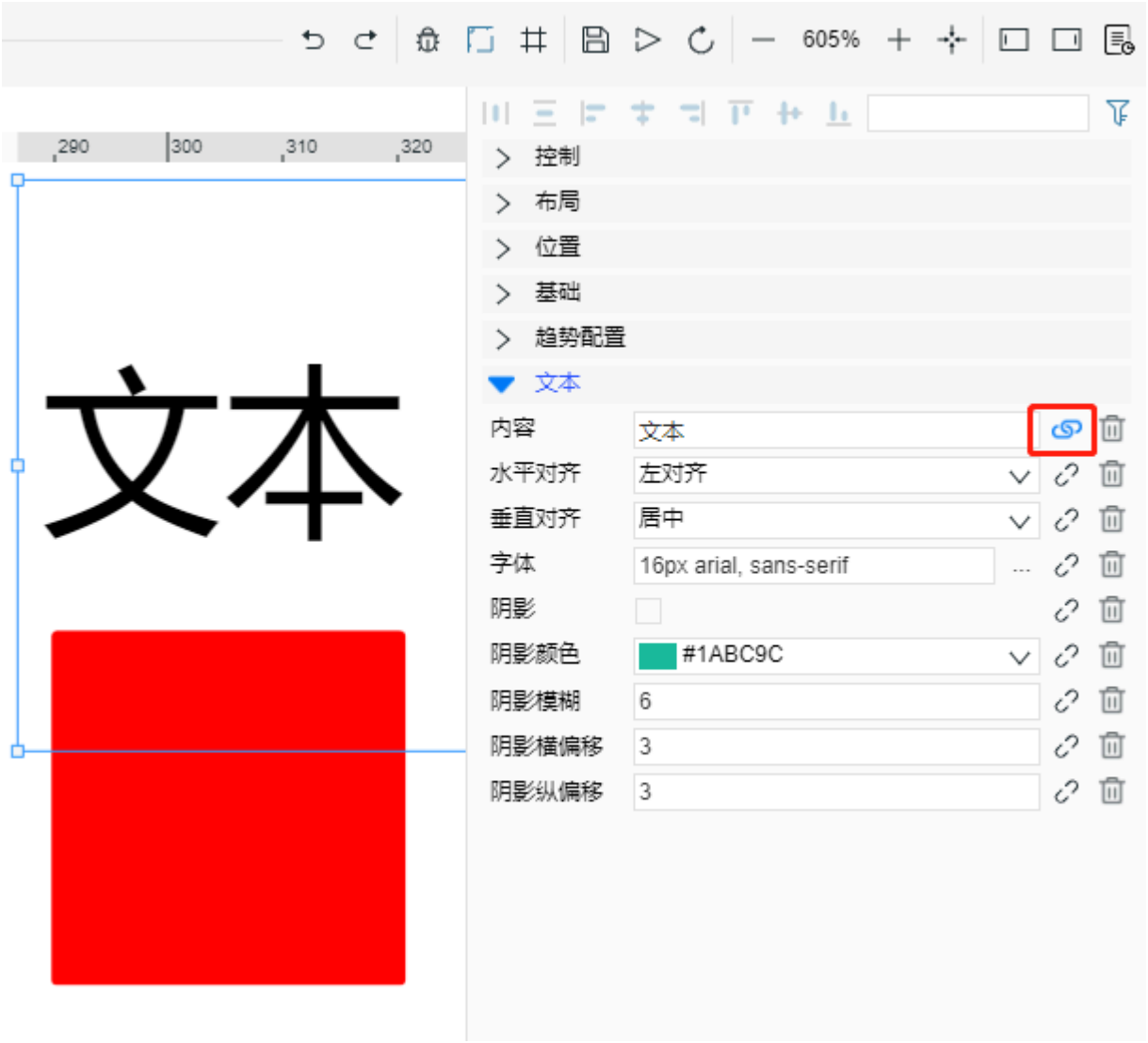


也可通过下方导航框快速修改背景及边框颜色

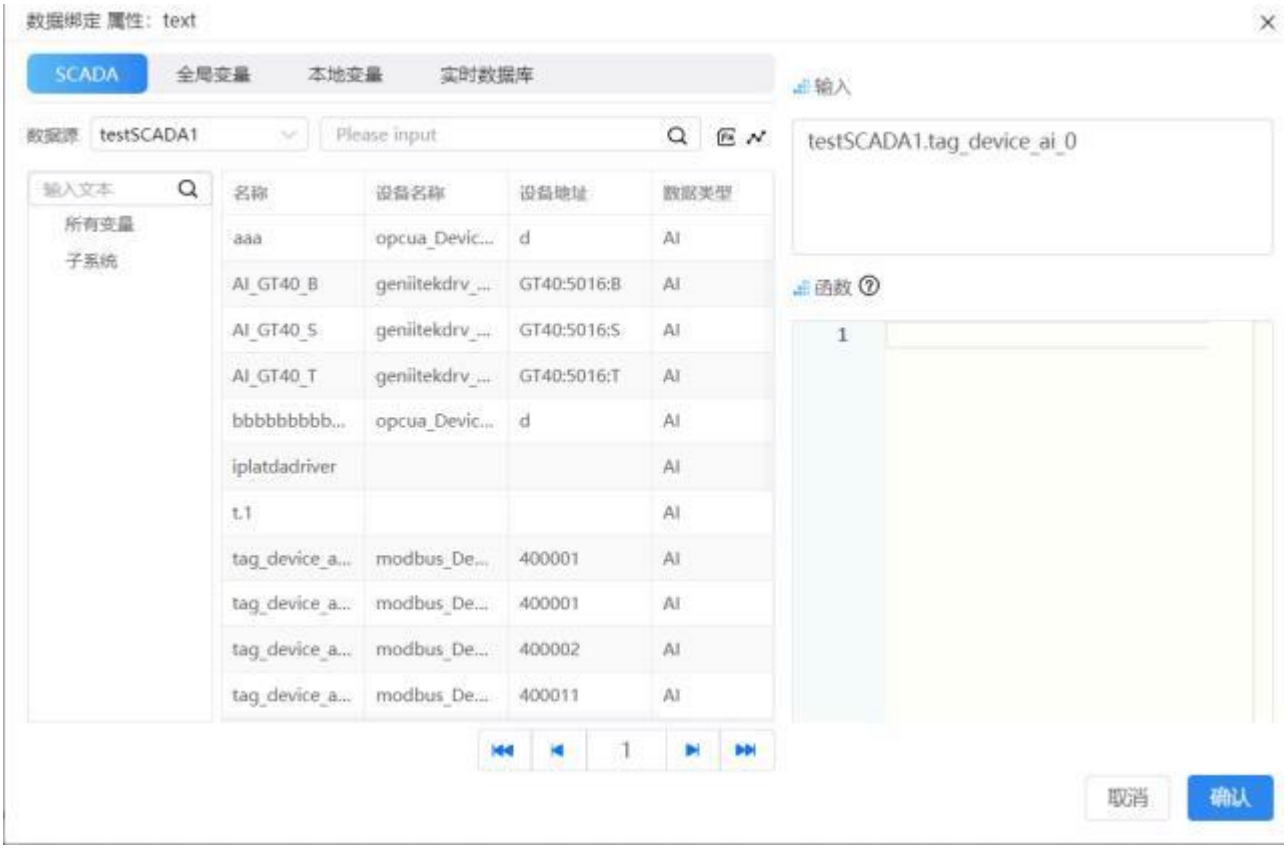


绑点

1. 绑点

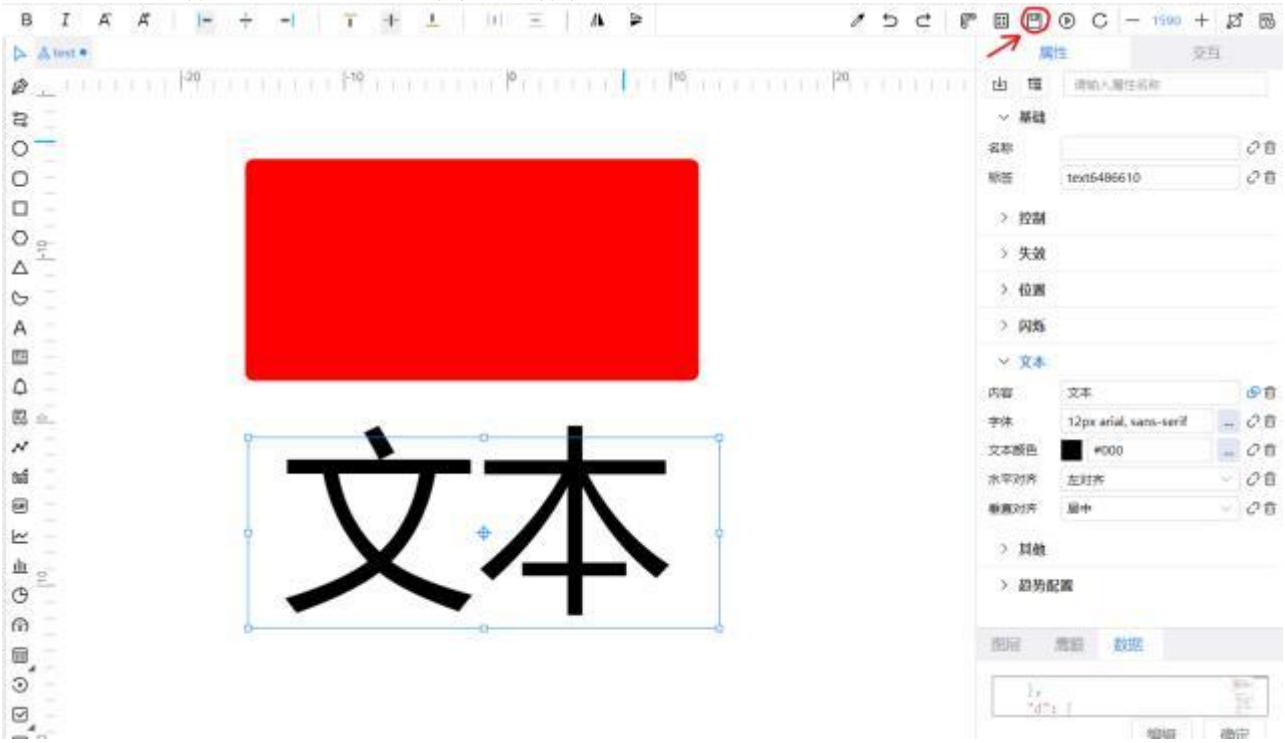

605%
290
300
310
320
文本
控制
布局
位置
基础
趋势配置
文本
内容
文本
水平对齐
左对齐
垂直对齐
居中
字体
16px arial, sans-serif
阴影
阴影颜色
#1ABC9C
阴影模糊
6
阴影横偏移
3
阴影纵偏移
3

2. 选择点，点击确定


数据绑定 属性：text
SCADA
全局变量
本地变量
实时数据库
数据源
testSCADA1
Please input
输入文本
所有变量
子系统
名称
设备名称
设备地址
数据类型
aaa
opcua_Devic...
d
AI
AI_GT40_B
geniitekdrv_...
GT40:5016:B
AI_GT40_S
GT40:5016:S
AI_GT40_T
GT40:5016:T
bbbbbbbbb...
iplatdadriver
t.1
tag_device_a...
modbus_De...
400001
400002
400011
1
输入
testSCADA1.tag_device_ai_0
函数
取消
确认

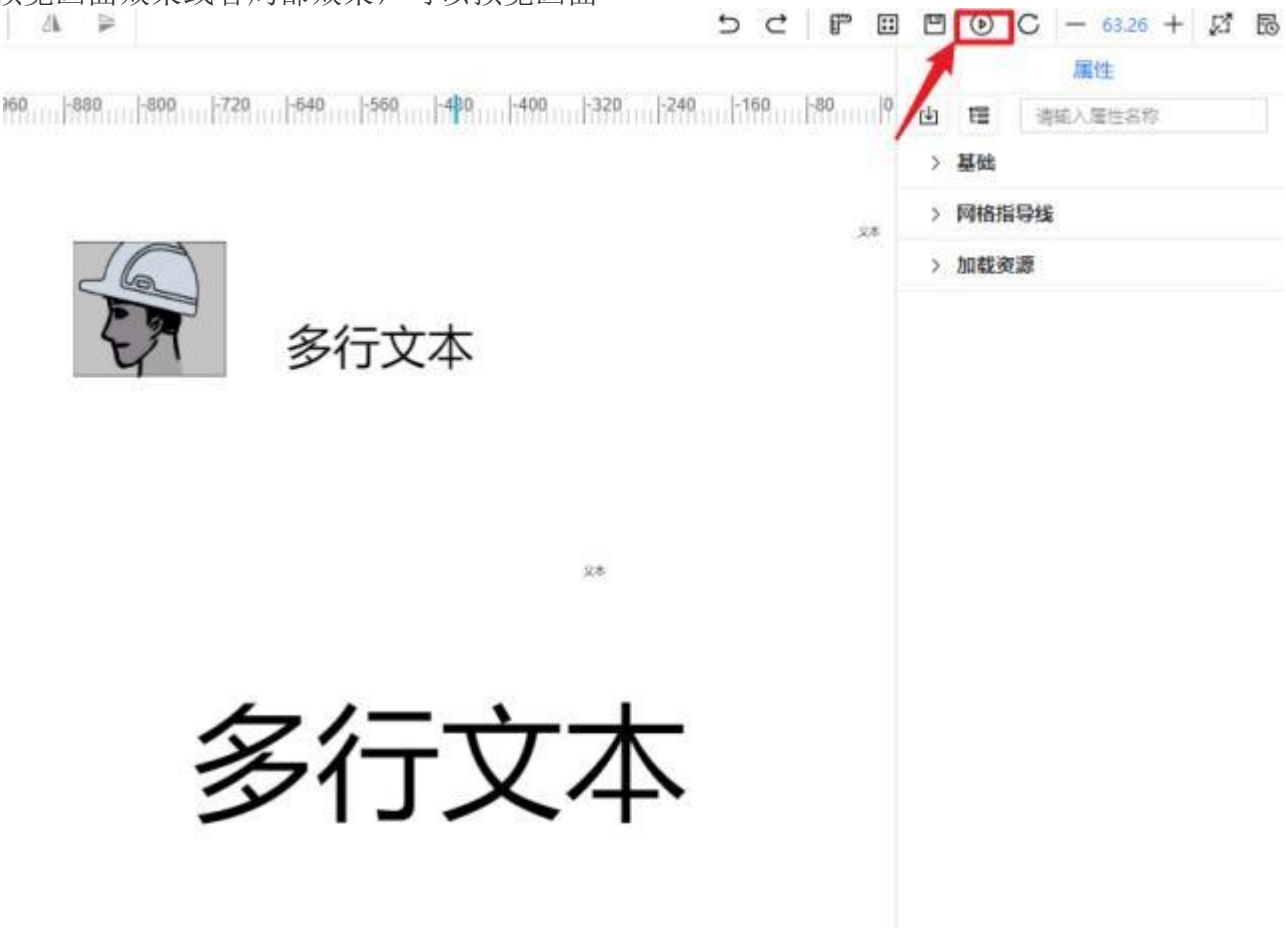
保存画面

1. 可以Ctrl+S保存，或者点击右上角的保存按钮



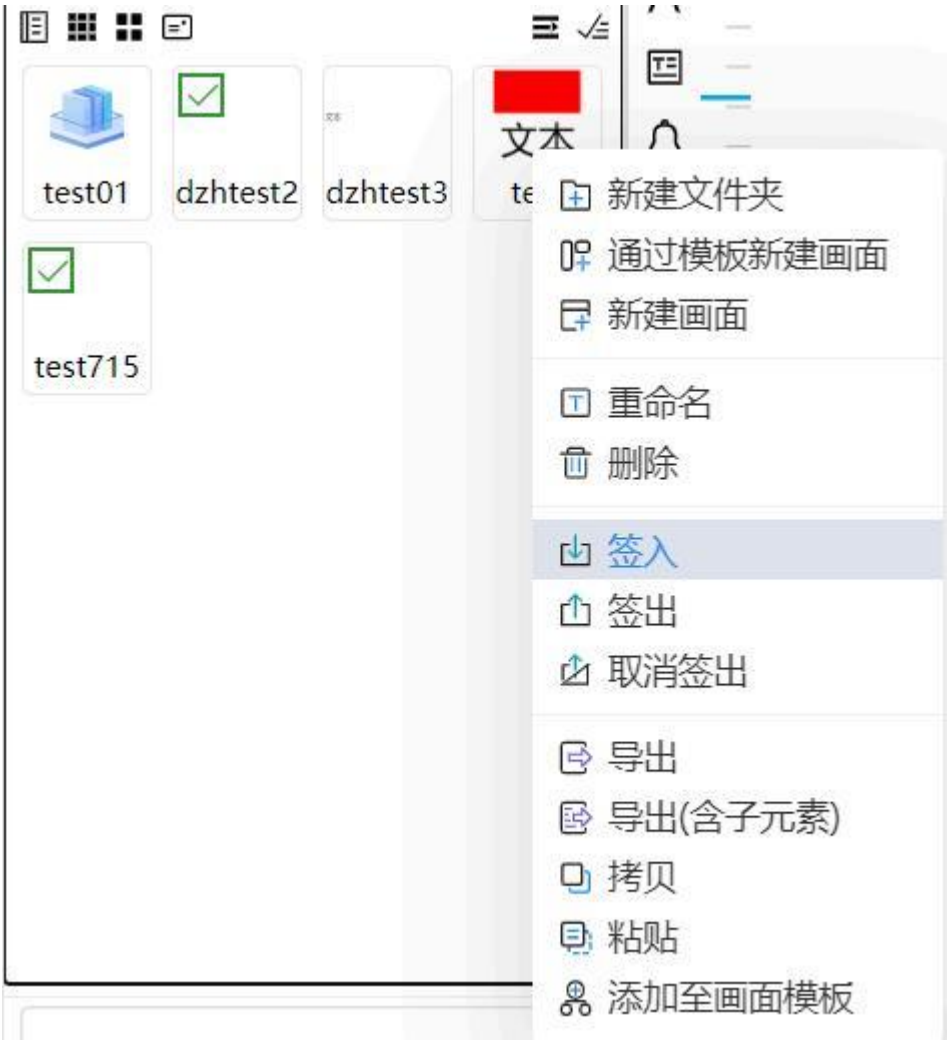
预览画面

1. 预览画面效果或者局部效果，可以预览画面



签入画面

1. 点击右键选择签入



5.3 元素功能属性

属性概述

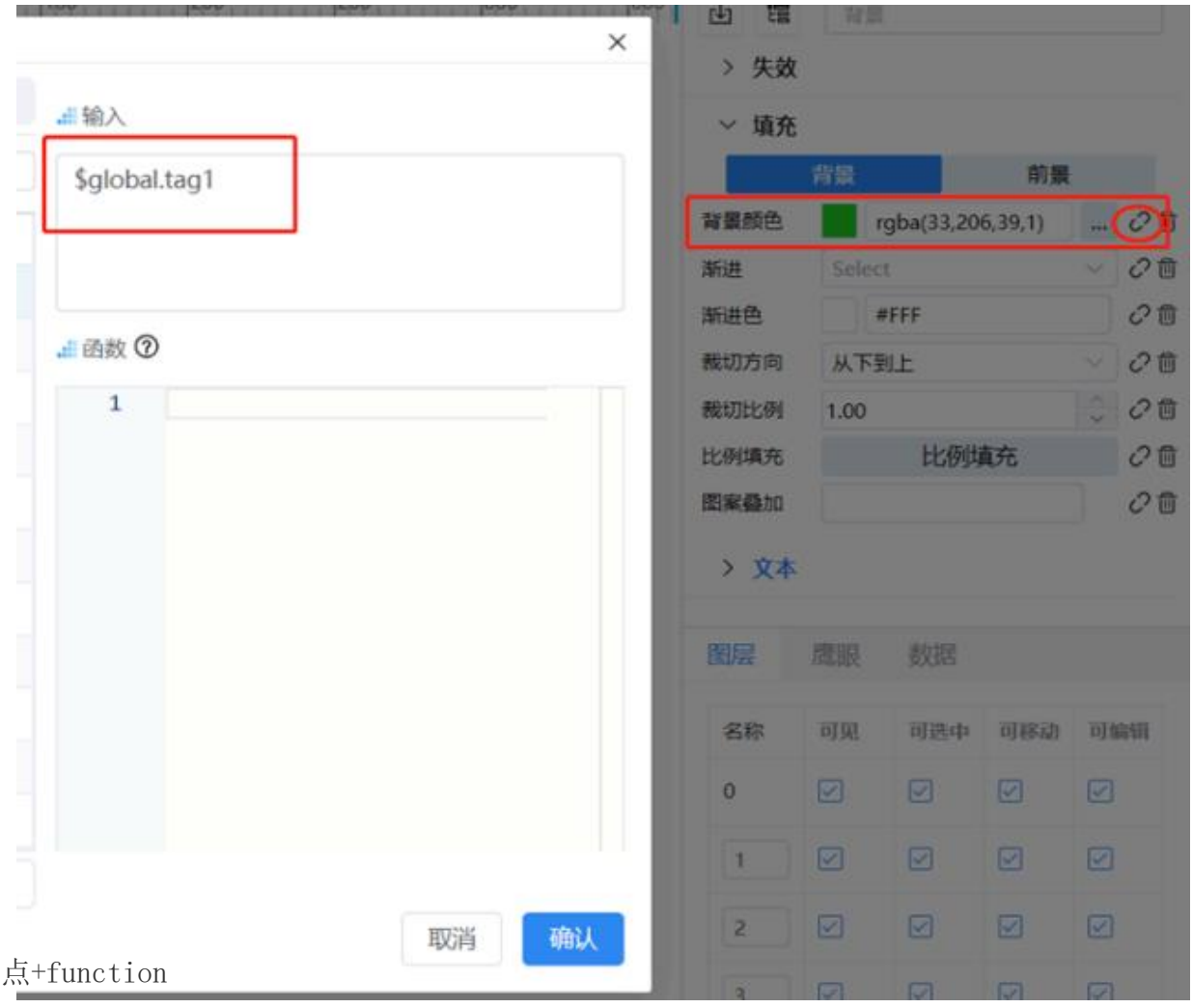
元素的功能属性，包括属性、交互、扩展属性三种：

- **属性**：每个元素都有的基本属性，包括可见性、可选中、可编辑、角度、坐标、宽度等
- **交互**：每个元素都有的基本属性，包括离散值、写值、窗口等
- **自定义属性**：不同类型元素特有的属性，比如，文本元素的文本内容、文本颜色、字体，IO控件的值、上下限等

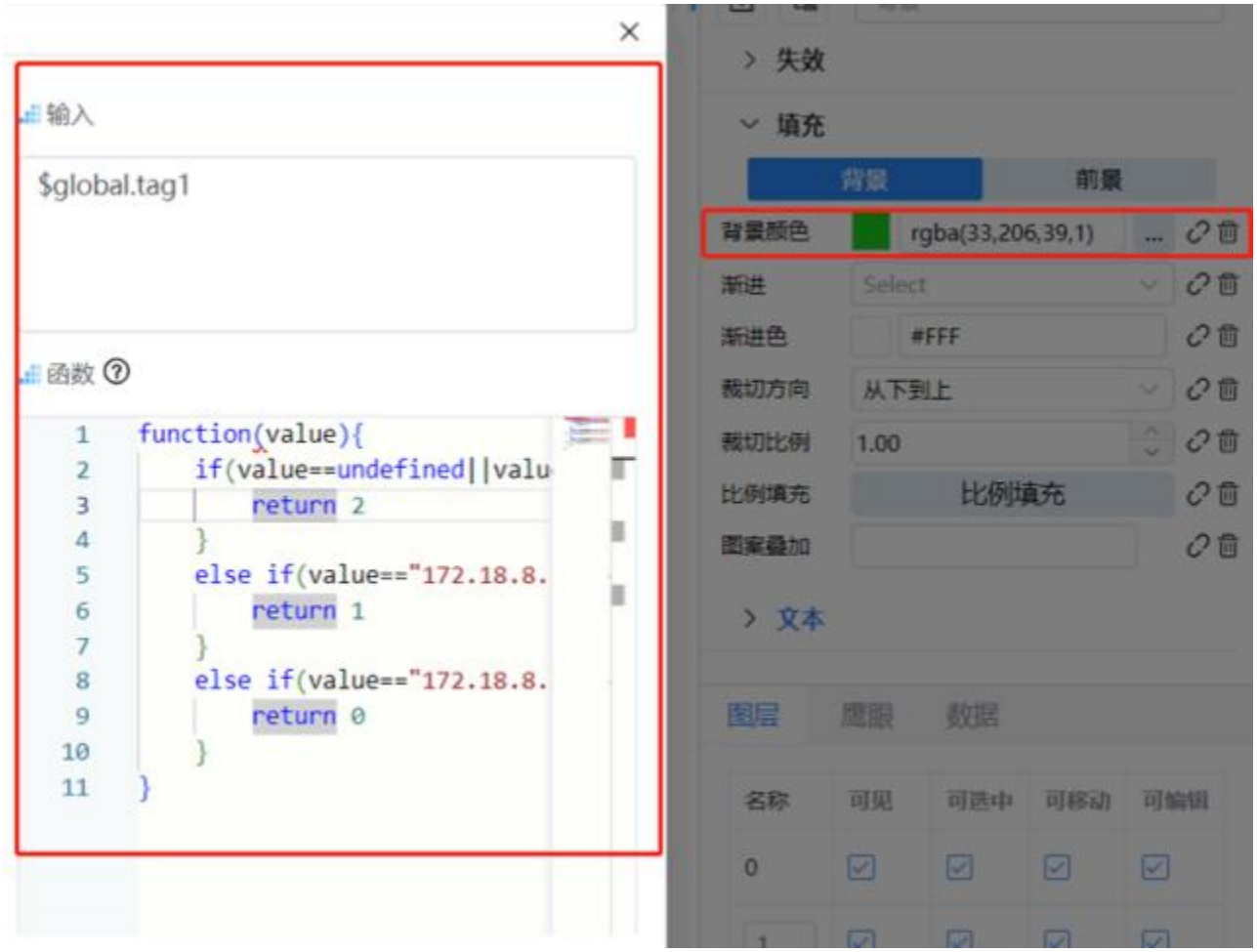
属性使用方式：

- 直接输入内容

•绑点



•绑点+function



•脚本编辑器内脚本操作属性

该编辑器内实现：点击“op10”元素，将“组态回放”元素的“s:2d.visible”（可见属性）的赋值为false。

```
op10_onClick = function(event, data, view){
    sys.sys.setAttribute("组态回放", "s:2d.setVisible", false);

};
```

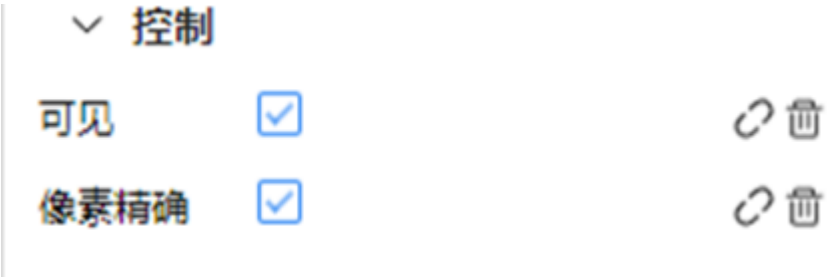
属性

基础

•名称：元素名称，可用于标记元素业务名称

•标签：元素唯一标识，对象名，可用于元素列表中筛选、脚本调用等

控制



•可见：控制元素在运行态的可见、不可见状态，默认可见可通过勾选、绑点、写function方式控制，此外可通过“s:2d.visible”属性进行脚本控制：

```
// 隐藏元素
sys.setAttribute("元素标签", "s:2d.visible", false);
// 显示元素
sys.setAttribute("元素标签", "s:2d.visible", true);
```

•像素精确：勾选表示编辑态选择该元素时，只能点击有像素（非透明）的位置来选中

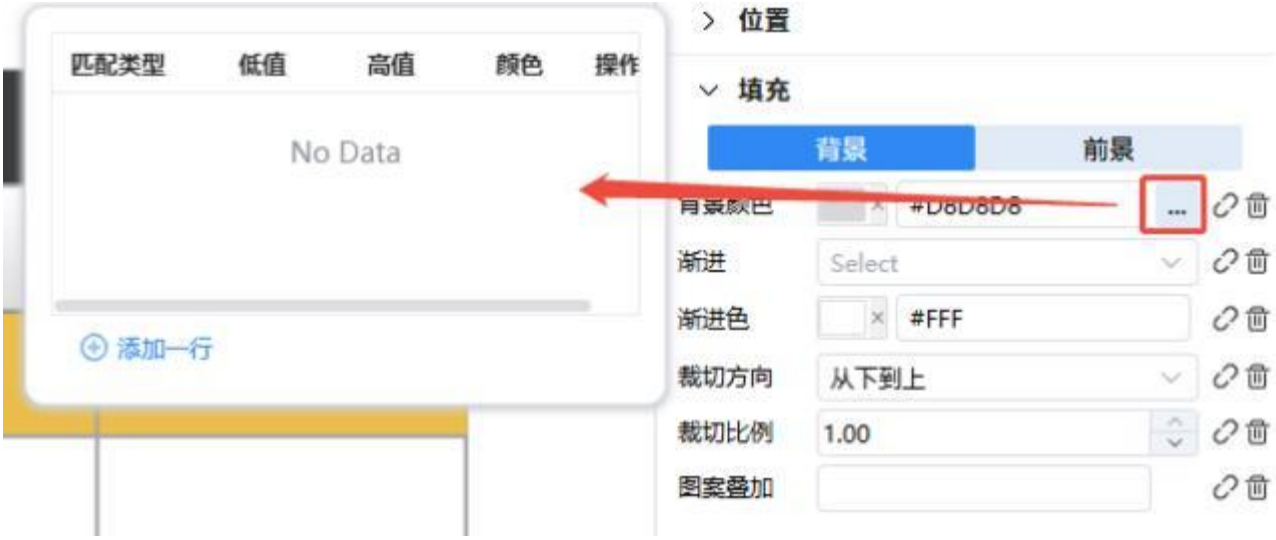
位置

•横坐标：元素的横坐标（p:x）
•纵坐标：元素的纵坐标（p:y）
•锁定比例：锁定元素比例，勾选后元素可按比例缩放
•宽度：元素宽度
•高度：元素高度
•角度：元素旋转的角度(绑定里使用PI计算，例如PI/2表示90°)
•横缩放：元素水平方向缩放(1为原大小，-1翻转)
•纵缩放：元素垂直方向缩放(1为原大小，-1翻转)
•横锚点：元素锚点的水平坐标

•纵锚点：元素锚点的垂直坐标
•垂直：元素纵坐标与点值线性关联
•水平：元素横纵标与点值线性关联
•全屏：元素在运行态全屏展示，有等比、填充两种方式
•节点吸附：在A元素的吸附节点中输入B元素的标签名，之后B元素的位置会跟随A元素改变

填充

•背景颜色：配置元素的背景颜色匹配规则，可以通过绑点、写function方式控制，写function需要返回数字去匹配规则。

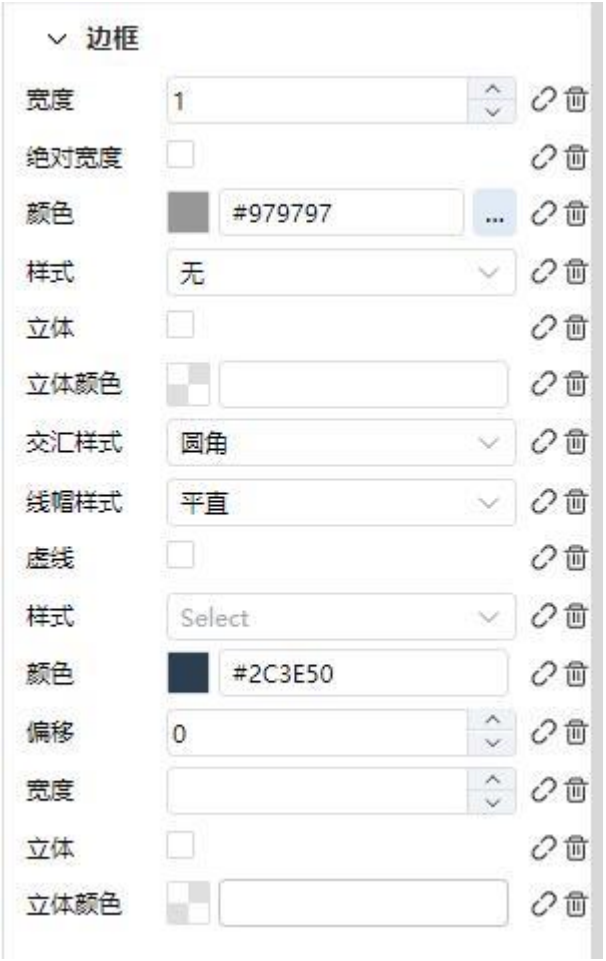


◦添加一行：添加一条颜色匹配规则
◦规则属性：
▪匹配类型：包括范围比较和完全匹配
▪低值：匹配类型选择范围比较时范围的低值或者匹配类型选择为完全匹配时对应的值
▪高值：匹配类型选择范围比较时范围的高值
▪颜色：匹配时的背景颜色
◦置为透明：点击颜色框右侧叉，将当前颜色透明度修改为0
•渐进：元素背景的渐进方式
•渐进色：元素背景的渐进颜色
•裁切方向：元素背景颜色的填充方向
•裁切比例：元素背景颜色的填充比例，区间为（0-1）
•图案叠加：设置元素图案叠加

•前景颜色：配置元素的前景颜色匹配规则
•渐进方向：前景色颜色渐进方向
•渐进色：前景渐进颜色
•裁切方向：元素前景颜色的填充方向

•裁切比例：元素前景颜色的填充比例，区间为（0-1）

边框



•宽度：元素的边框宽度，可以通过绑点、写function方式控制，写function需要返回数字去匹配规则
•绝对宽度：元素的绝对宽度，不受放大缩小影响
•颜色：元素边框颜色
•样式：元素边框的虚线样式
•立体：元素是否展示为立体效果
•立体颜色：元素立体效果颜色
•交汇样式：元素边框交汇处的样式，有圆角、斜角、尖角
•线帽样式：元素边框线帽的样式，有方形、圆形、平直
•虚线：设置元素边框是否为虚线
•样式：虚线样式
•颜色：虚线颜色
•偏移：元素边框的虚线偏移量
•宽度：元素边框虚线宽度
•立体：元素边框虚线是否展示为立体效果
•立体颜色：元素边框虚线的立体效果颜色

其他

- 显隐频率：元素运行态的显隐状态交替的频率，值为0时，无动画；值为-1时，不可见；值取范围是1~9，值越大频率越小
  可通过下拉框选择、绑点、写function方式控制，此外可通过“cvs_freq”属性进行脚本控制显隐的速度：

```
sys.setAttribute ("元素标签", "cvs_freq", 200);// 最后一个参数 200 的单位是毫秒
```

- 权限名称：元素的权限名称
- 防抖延迟：配置元素点击防抖，在配置时间内只触发一次点击事件
- 类名：元素类名
- 图层：元素所属图层
- 提示：配置运行态鼠标悬停元素时的提示内容
- 透明度：元素的透明度，值的范围是0~1，可通过直接输入、绑点、写function方式控制，此外可通过“s:opacity”属性进行脚本控制透明度：

```
sys.setAttribute ("元素标签", "s:opacity", 0.5);
```

- 灰化：配置元素是否灰化
- 染色：配置元素的颜色，优先级在背景色和前景色之间
- 裁切比例：元素的颜色填充比例，可通过直接输入、绑点、写function方式控制，此外可通过“s:shape.foreground.clip.percentage”属性进行脚本控制裁剪比例：

```
sys.setAttribute ("元素标签", "s:shape.foreground.clip.percentage", 0.4);
```

- 裁切方向：元素颜色填充方向
- 不规则图形：元素的图形类型

闪烁

背景闪烁：

- 是否闪烁：配置元素是否闪烁，勾选此项闪烁颜色及闪烁频率配置生效，否则不生效；
- 闪烁颜色：元素的背景闪烁颜色

  - 添加一行：添加一条颜色匹配规则
  - 规则属性：
    - 匹配类型：包括范围比较和完全匹配
    - 低值：匹配类型选择范围比较时范围的低值或者匹配类型选择为完全匹配时对应的值
    - 高值：匹配类型选择范围比较时范围的高值
    - 颜色：匹配时的闪烁颜色
- 闪烁频率：元素的背景闪烁频率，分为慢速、中速、快速

边框闪烁：功能属性同背景闪烁，对应元素边框

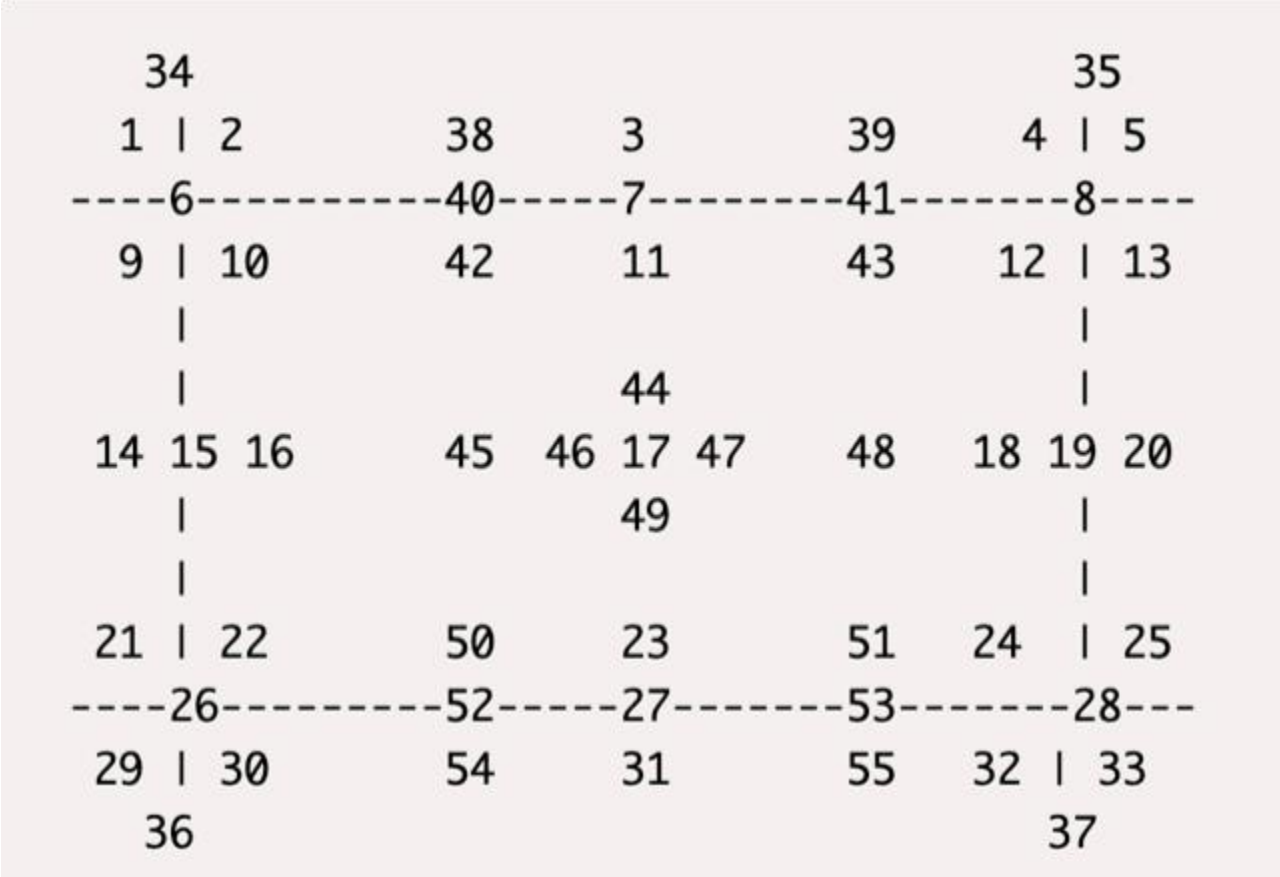
文本

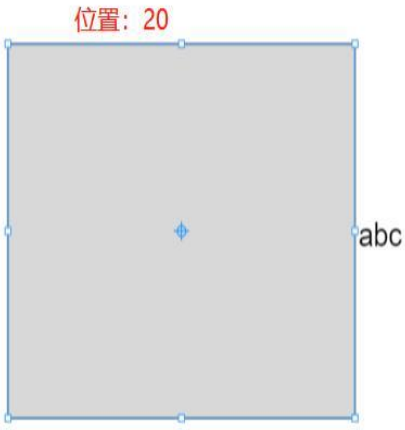
- 内容：文本内容

- 字体：文本字体
- 文本颜色：文本颜色
- 背景：文本背景颜色
- 位置：矩形元素上的文本位置
- 横偏移：元素上文本横向偏移量
- 纵偏移：元素上文本纵向偏移量
- 最大长度：文本显示的最大长度
- 角度：文本旋转角度(绑定里使用PI计算，例如PI/2表示90°)
- 缩放：文本缩放大小

矩形元素文本位置说明：

元素位置是以1~55的数字的信息来描述相文本对于一个矩形区域的位置。比如17代表矩形中心，6代表矩形左上角，20代表矩形右侧





位置可以直接输入数字，也可以点击按钮弹出配置对话框去配置。

交互

交互有效

功能介绍

- 画面（图元）元素可以配置失效，失效的元素各种响应事件被禁用，不能操作，同时一些特殊元素（形状、文本）等*的背景边框文本色可以在失效状态时设置失效颜色，用于区分有效状态。

配置介绍

- 选中元素后，在属性配置里的“失效”分组中，可以配置是否“交互有效”，以及失效状态的颜色。同时支持绑定数据动态设置状态。
- 拥有背景色、边框色的元素（矩形、圆形等）可以配置失效背景色、边框色
- 拥有文本色的元素（文本）可以配置失效文本色

- 交互有效：勾选表示交互动作有效，例如可以点击下发、弹窗等，不勾选表示元素无法交互，点击不会有响应，
- 失效背景色：失效状态下，元素的背景颜色
- 失效边框色：失效状态下，元素的边框颜色

可通过勾选、绑点、写function方式控制，此外可通过“s:2d.selectable”属性进行脚本控制

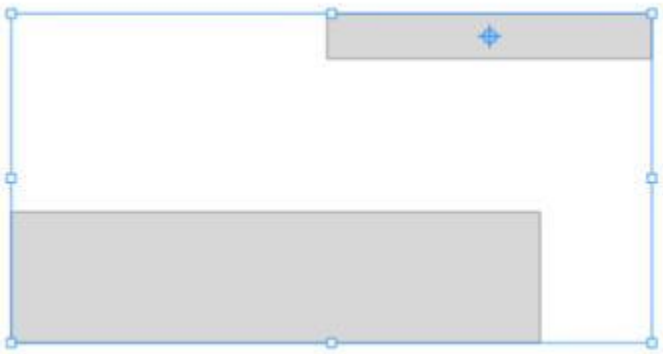
```
sys.setAttribute("元素标签", "s:2d.selectable", false);
sys.setAttribute("元素标签", "s:2d.selectable", true);
```

兼容性

- 画面中的失效，复用旧版本的“2d.selectable”即可选中，升级后可以兼容原属性值及绑点信息。
- 如果不做任何修改，失效颜色默认为undefined，不产生失效效果，完全兼容旧版本。
- 当用户第一设置元素为无效状态时，给用户设置默认背景、边框、文本失效颜色。
- 图元中的失效，没有复用旧版本属性，新增属性“interactiveEnable”。

关于分组

- 当几个可点击的元素组合后，这个组合不可触发内部元素的点击，需要设置组合的“可穿透”性质，才能触发内容元素。如果组合整体设置为失效，可穿透，目前还是可以触发内部元素



离散值

- 启用：离散值配置是否启用
- Tag点：元素离散值绑点
- 动作：绑点触发的元素动作，有直接、取反、切换、复位、置位
  - 直接：鼠标点击元素，鼠标按下写1，松开写0
  - 取反：鼠标点击元素，值为1时写0，值为0时写1
  - 切换：鼠标点击元素，点值1/0切换
  - 复位：鼠标点击元素，点值复位为0
  - 置位：鼠标点击元素，点值置位为1

离散值的下发，可以实现多个点的同时下发，方法是绑点时将要下发的点用英文逗号，隔开

写值

•启用：写值配置是否启用
•Tag点：元素需要写值的点
•值内容：元素需要写的值，也可绑定点或表达式；支持同时绑定多个点，用","分开

窗口

•启用：弹窗属性配置是否启用
•路径：所弹画面路径
•ID：用于关闭画面；仅支持英文和数字，不能用中文，特殊符号，空格；关闭弹窗函数:sys.closePage("弹窗ID");
•窗口类型：选择打开窗口还是关闭窗口
•窗口模式：打开窗口的模式，分8种：
  ◦页面跳转
  ◦新页面跳转
  ◦弹出模态画面
  ◦弹出非模态画面
  ◦新打开一个同级的浏览器tab页
  ◦打开一个新窗口，可以指定大小
  ◦非模态且默认新开页面原大小，此时【弹窗宽度/高度】属性不生效；
  ◦快速弹窗，一般使用在对操作速度有要求的场景，弹窗里的所有sys的函数需要使用view去调用，例如view.sys.setAttribute;
•窗口配置：可配置弹窗基本信息

◦宽高：弹窗宽高
◦位置：弹窗的位置，可配置X,Y坐标轴，也可选择弹窗位置跟随鼠标
◦标题：弹窗的标题
◦启用：相对Top-弹框是否在所有画面最外层，默认为true；显示标题-是否显示弹窗标题；显示关闭-是否显示弹窗关闭按钮；显示边框-选择弹窗是否需要边框
◦变量替换：用于变量名替换，本质是绑定属性里的字符串替换。例如原来绑定变量是"$global.test3"，原变量填"3"，替换的变量填"1"。打开弹窗后,变量"$global.test3"会被替换成"$global.test1"。多变量替换,用";"隔开,原变量和替换的变量一一对应
。传参：父页面传递给子弹窗的对象；例如`{"tag1":"1","tag2":"2"}`
。关闭回调：回调函数，关闭页面时触发

脚本

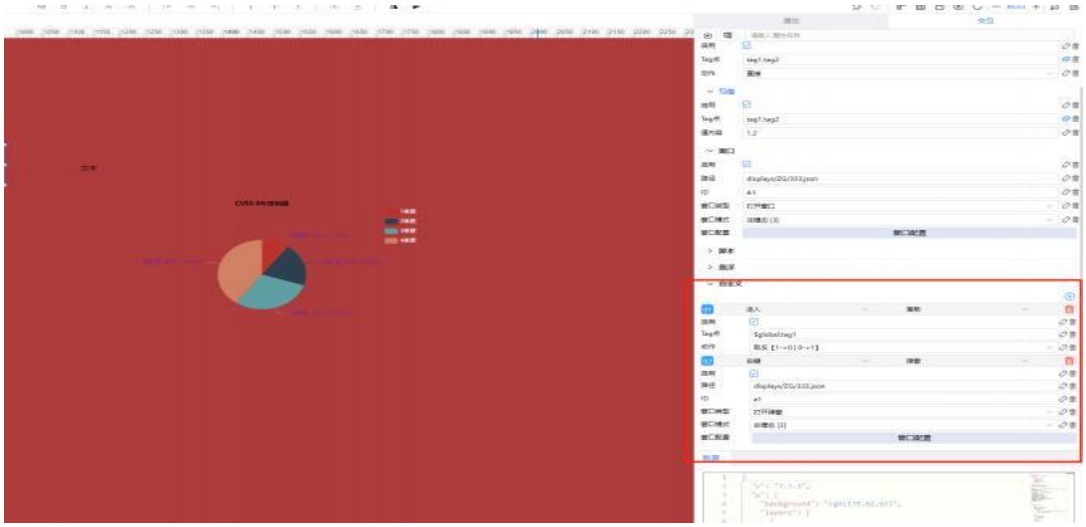
•点击元素，执行脚本

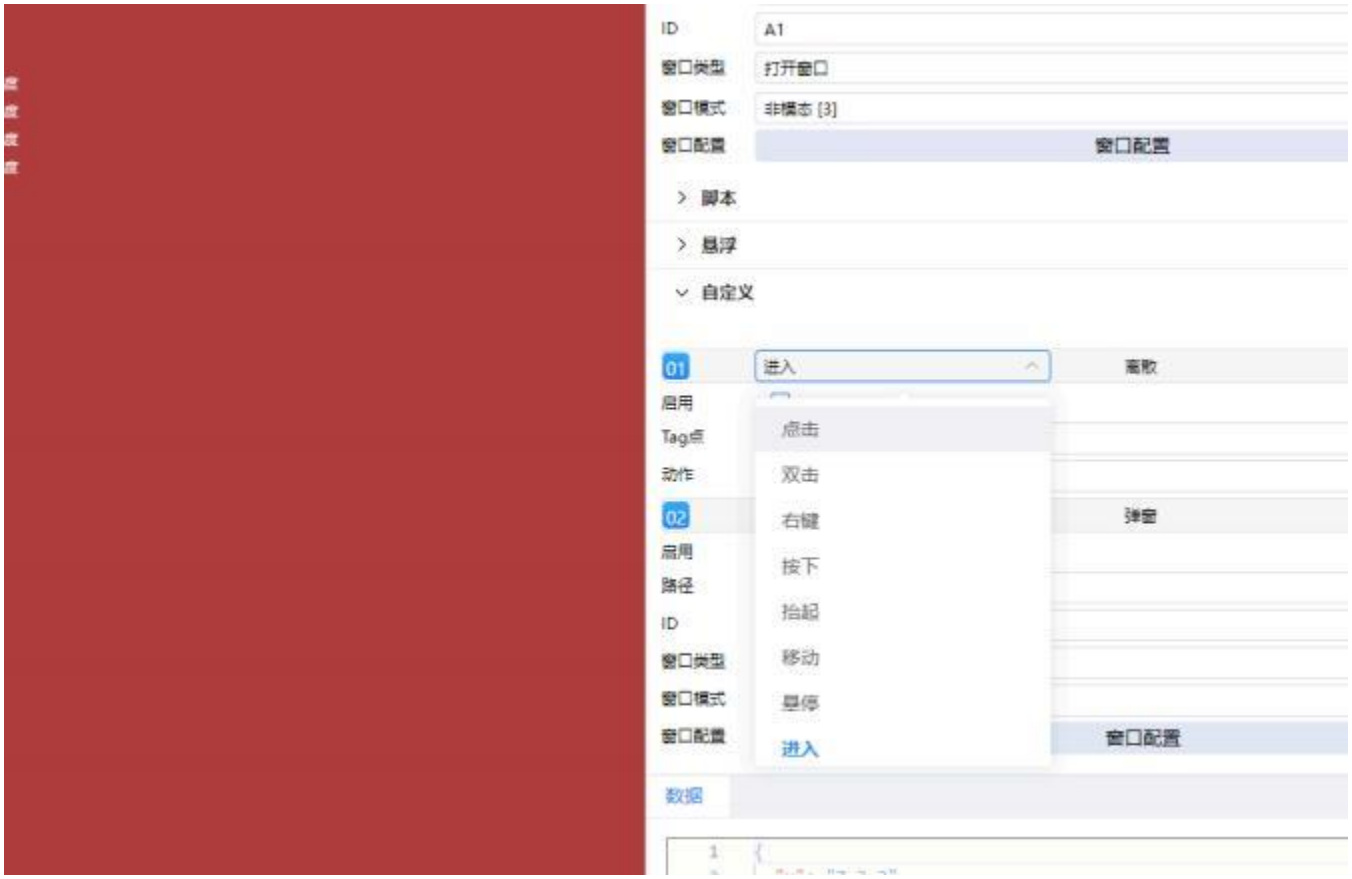
悬浮

•悬浮：运行态鼠标悬停在元素上显示边框；勾选悬浮，以下悬浮功能配置生效，否则不生效
•颜色：运行态鼠标悬停在元素上显示的边框颜色；
•宽度：运行态鼠标悬停在元素上显示的边框宽度；
•边框：运行态鼠标悬停在元素上的边框大小
•阴影：运行态鼠标悬停在元素上的边框阴影；勾选阴影，以下阴影配置生效，否则不生效
•阴影颜色：边框阴影颜色
•阴影模糊：边框阴影效果模糊度
•阴影横偏移：边框阴影横向偏移量
•阴影纵偏移：边框阴影纵向偏移量

自定义交互配置

上述的基本交互功能可能不满足一些交互需求，如触发条件可能时右键，可能左右键都要触发写值等，因此自定义交互对基本交互功能进行了扩展，允许右键、鼠标进入等其他事件来触发交互操作





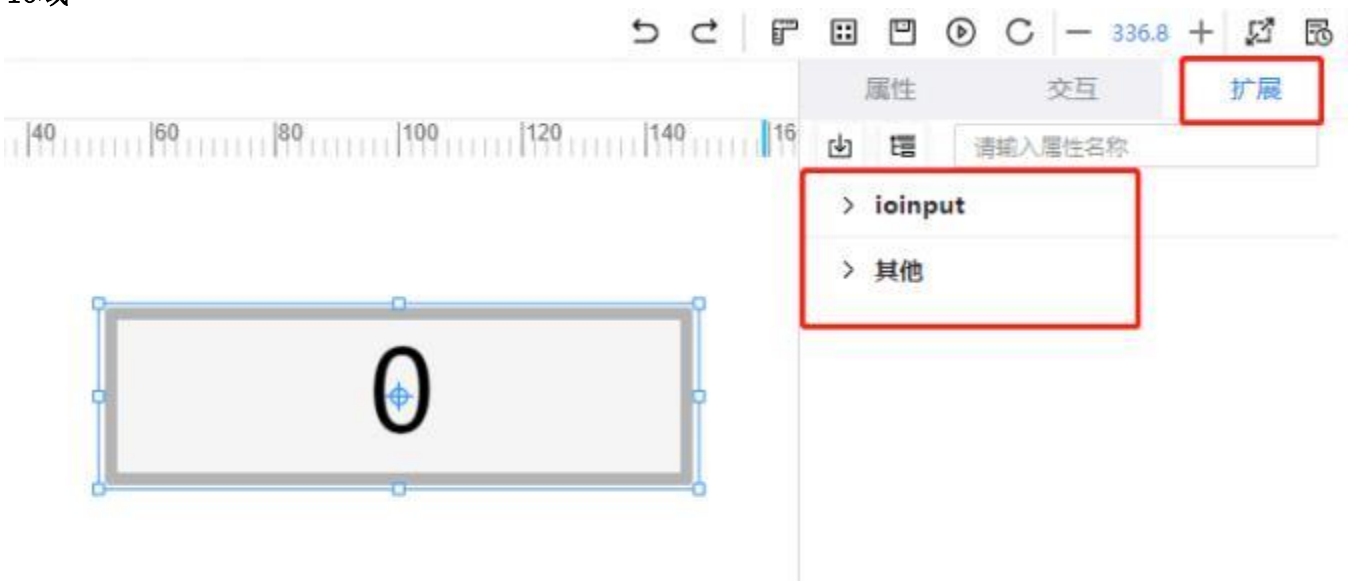
如上图，自定义交互不限制交互的数量，可以任意添加，并允许各种触发条件，因此你可以任意进行组合实现你的交互功能。

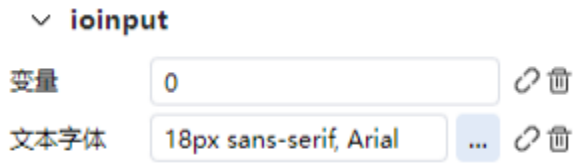
如，你可以实现：
1. 鼠标按下时写100，抬起时写-100；右键时弹出一个指定的弹窗
2. 鼠标进入时弹窗，鼠标移出时关闭弹窗

扩展

不同元素的特有属性

IO域





•变量：可通过IO域编辑控制的变量，同时也会显示
•文本字体：IO域显示文本字体

其他

•输出格式：IO域输入时可以输入或者控制的变量的格式有（字符串，小数，十进制，二进制，八进制，十六进制，指数）
•小数点位数：是指数值精确到小数点后几位
•输出位数：是指整体位置，比如点值是12，精确到小数点后2位，输出位数设置6位，那么就会出现012.00
•补零/补空格：是指比如上方输出位数012.00则是选择的补零，如果是补空格则会是12.00
•上限值/下限值：是指这个变量可控制的一个范围，为全闭，比如上限设置为10，下限设置为1，那么tag点可控制的区间为1<=tag<=10，当超过或者低于这个区间时会提示
•背景色：背景色为IO域的背景颜色，根据需要可进行调整
•文本颜色：对应点值显示出的颜色，根据需要可进行调整
•水平对齐：控制值显示的水平位置，分（左对齐，居中，右对齐）
•边框颜色：是指IO域边框的颜色，根据需要可进行调整
•边框宽度：调整边框的宽度，根据需要可进行调整

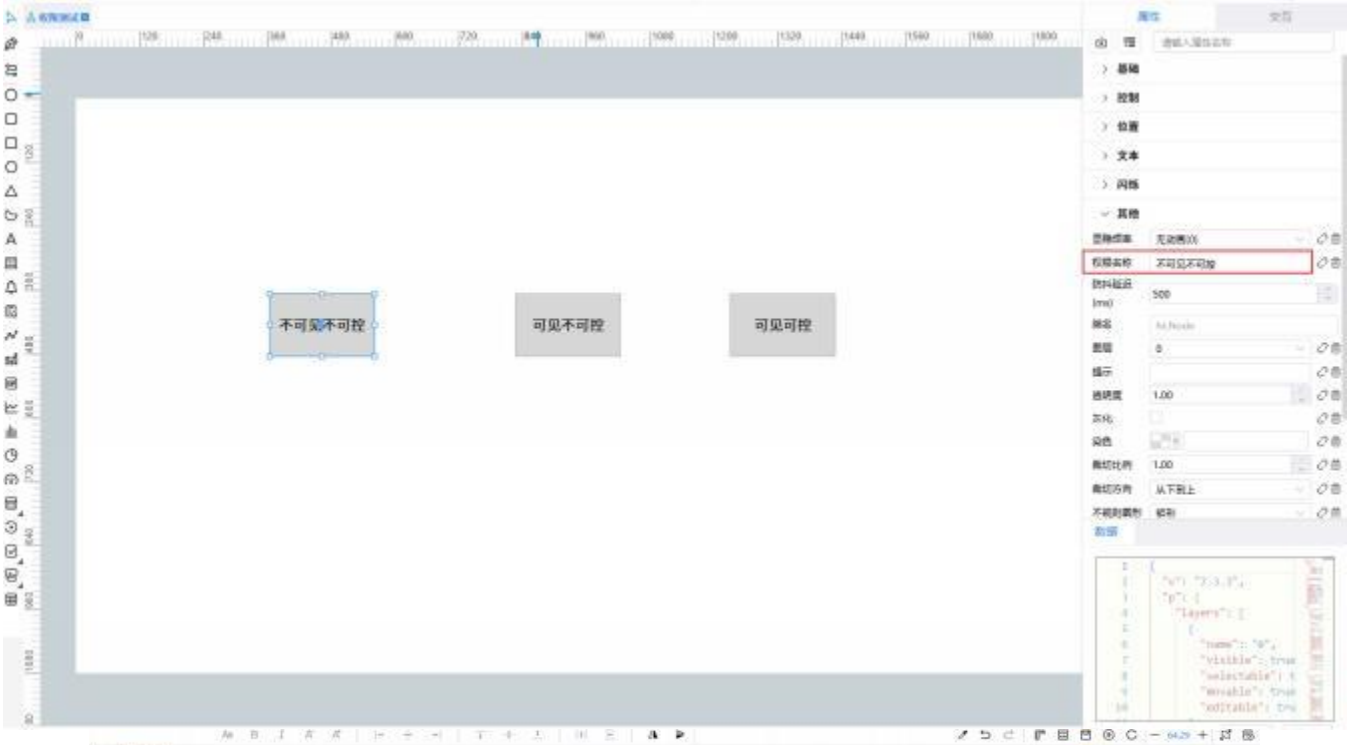
- 禁用输入：当需要条件判断禁用输入时（例如权限不够，不能执行控制下发），在后面类似连接按钮点击绑上需要判断的点或表达式即可
- 输入时背景色/编辑时边框色：当点击IO域准备进行控制时，背景色和边框色会对应配置进行更改
- 输入时清空：勾选后点击io域聚焦后会清空原来内容重新输入，不勾选则保留原来内容并在聚焦时全选所有内容
- 下发前事件：如需要在写值之前做一些处理则可以将处理脚本写在这里，比如写入字符串时，先将字符串的长度写入到另一个点

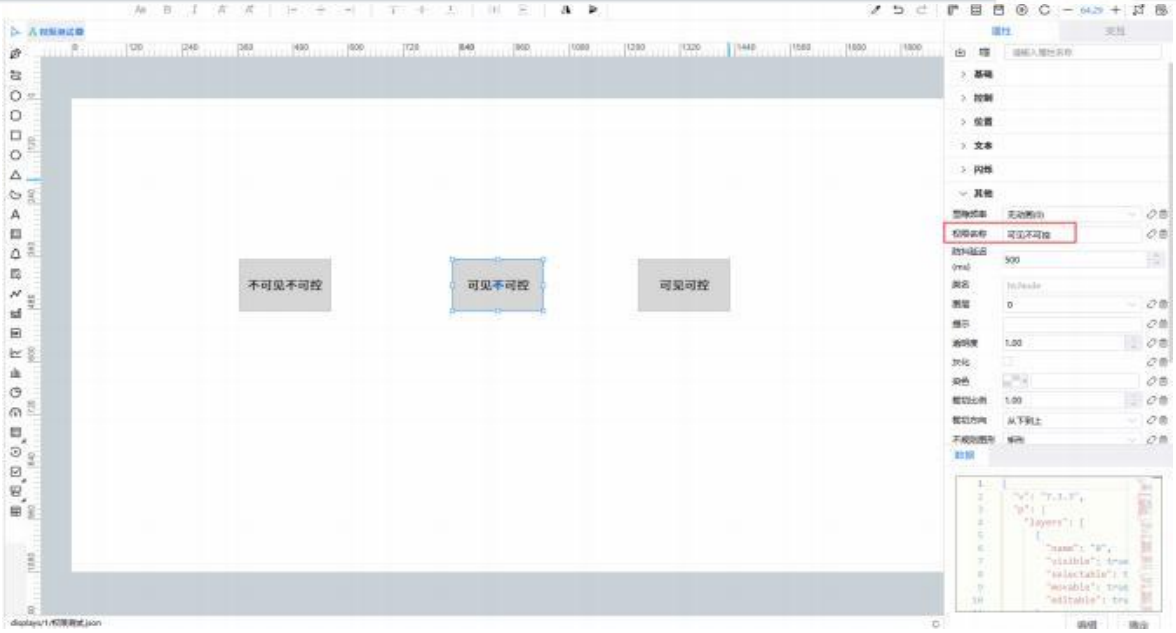
元素的授权与鉴权

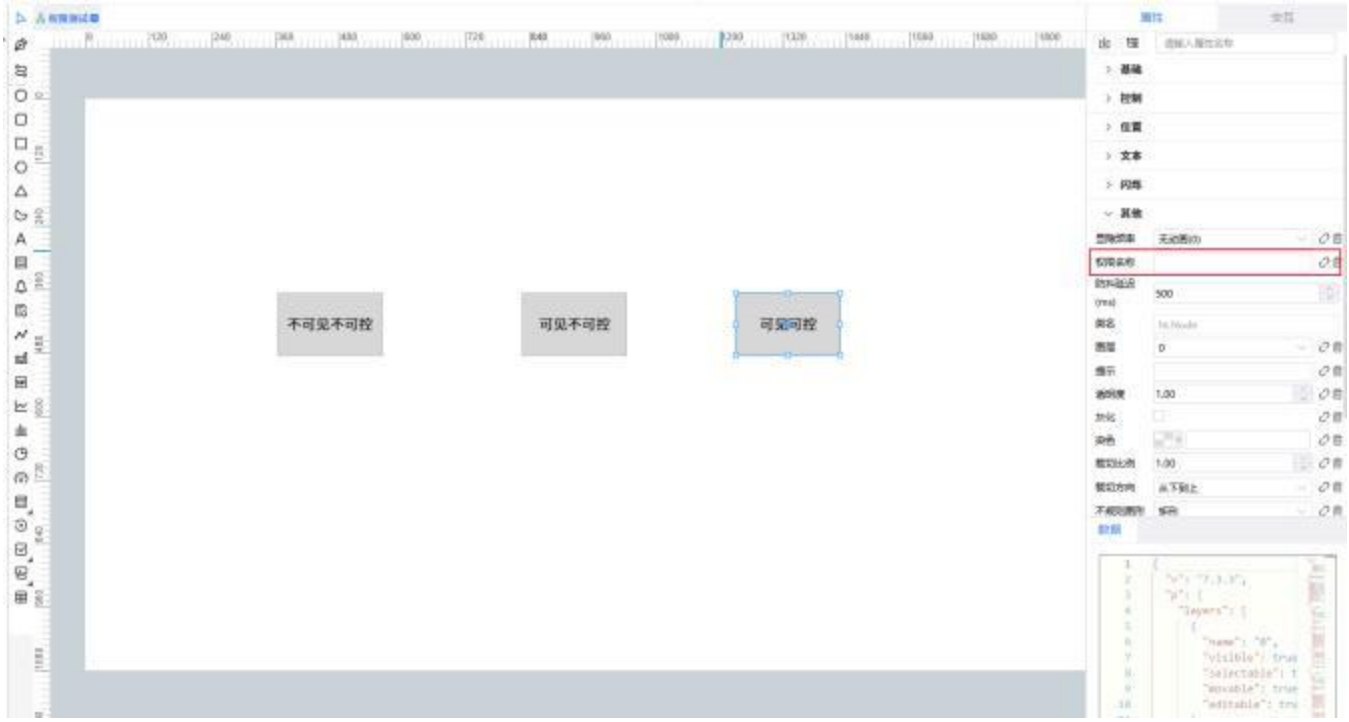
授权

页面元素授权的前提是，页面也是授权的

画面上存在三个元素，他们的权限名称分别是”不可见不可控“，”可见不可控“以及空白。







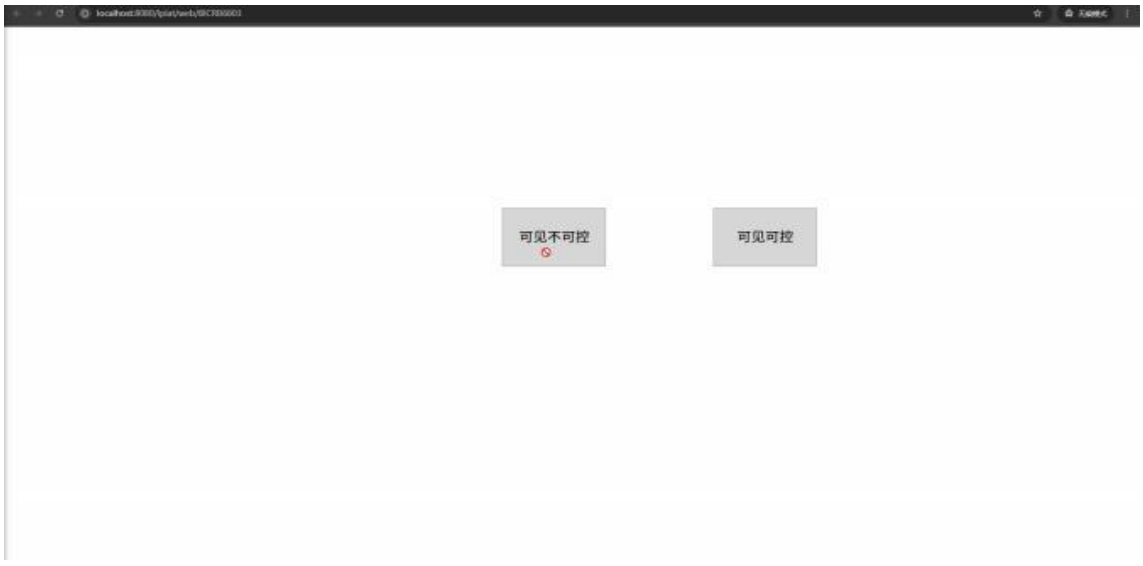
然后去到BIPG01页面，在给运行态给这些资源，设置上对于的操作类型。

授权

运行态鉴权：

”不可见不可控“的元素，不显示。

”可见不可控“的元素，显示，但是无法进行交互。”可见可控“的元素，显示，并且能够进行交互



5.4 元素属性总表

- 设计器属性：包括画面设计器画布属性、画面设计器中元素属性、图元设计器中画布属性、图元设计器中元素属性
- 属性类别：设计器中对属性的分类
- 属性名：代码中给定的元素对象的属性名，用于脚本set、get
- 描述：对属性的功能描述

设计器属性	属性类别	属性	属性名	描述
画面设计器画布属性	基础	启用刷新周期	\	数据推送周期，启用后生效
		刷新周期(ms)	\	数据推送时间间隔，毫秒，需启用"启用刷新周期"
		背景图	\	配置画布的背景图片路径
		显示滚动条	\	画面滚动条
		定时器	\	对脚本编辑器中的，定时器函数进行声明
		背景	\	画布背景色号
		宽度	a:width	画布宽度，超出画布范围的元素，不会被显示
		高度	a:height	画布高度
		反序列化前	\	
		反序列化后	\	
		可缩放	\	画面可用鼠标滚轮进行缩放
		可平移	\	画布元素可被移动
		可框选	\	画布元素可以鼠标左键框选
		适配内容	\	配置画面内容自适应
	网格指导线	网格可见	\	网格是否显示
		网格大小	\	开启显示网格后，调整每个网格的大小
		深线每隔	\	调整深色网格线之间相隔的距离
		浅线颜色	\	调整浅色网格线的颜色
		深线颜色	\	调整深色网格线的颜色
		网格角度	\	调整网格左上角的角度
		网格旋转	\	调整网格整体的旋转
		缩放临界值	\	调整缩放显示网格的临界值，缩放超过临界值则不显示网格
	加载资源	加载资源	\	

画面设计器中元素属性	基础	名称	p:displayName	配置元素在画面中的名称
		标签	p:tag	配置元素在画面中的唯一标识
	控制	可见	s:2d.visible	运行态元素是否显示
		像素精确	s:pixelPerfect	运行态元素是否渲染精确到像素的部分
	失效	交互有效	s:2d.selectable	元素被点击事，是否能够响应
		失效背景色	s:disableBackground	失效状态下，元素背景色
		失效边框色	s:disableBorderColor	失效状态下，元素的边框色
	位置	横坐标	p:x	元素的横坐标
		纵坐标	p:y	元素的纵坐标
		锁定比例	\	锁定元素的横纵比
		宽度	p:width	元素的宽度
		高度	p:height	元素的高度
		角度	p:rotation	调整元素旋转的角度(绑定里使用PI计算,例如 PI/2 表示 90°)
		横缩放	p:scaleX	调整元素水平方向缩放(1为原大小，-1翻转)
		纵缩放	p:scaleY	调整元素垂直方向缩放(1为原大小，-1翻转)
		横锚点	p:anchorX	调整元素锚点的水平坐标
		纵锚点	p:anchorY	调整元素锚点的垂直坐标
		垂直	s:position_v	配置元素纵坐标与点值线性关联
		水平	s:position_h	配置元素横纵标与点值线性关联
		全屏	s:fullscreen	元素在运行态全屏展示，有等比、填充两种方式
		节点吸附	s:host_adsorb	在A元素的吸附节点中输入B元素的标签名，之后B元素的位置会跟随A元素改变
	其他	显隐频率	a:cvs_flashrate	配置元素运行态的显隐状态交替的频率，值的范围是1~9，值越大频率越小
		边框颜色	s:cvs_shape.border.color_tag	配置元素的边框颜色匹配规则，一般通过变量值去匹配，有闪烁颜色和背景颜色两种类型
		权限名称	a:authName	配置元素的权限名称
		防抖延迟	\	配置元素点击防抖，在配置时间内只触发一次点击事件
		类名	\	
		图层	p:layer	元素处在画布的具体层
		提示	s:toolTip	配置运行态鼠标悬停元素时的提示内容
		透明度	s:opacity	配置元素的透明度，值的范围是0~1
		灰化	s:2d.gray	配置元素是否灰化所有颜色
		染色	s:body.color	配置元素的颜色，优先级在背景色和前景色之间
		裁切比例	s:clip.percentage	配置元素的颜色填充比例
		裁切方向	s:clip.direction	配置元素颜色填充方向
		不规则图形	s:shape	配置元素的图形类型。只在画面或者图元种的三角形、矩形等基本图形元素上使用。
		拉伸	p:image.stretch	
		图片	p:image	
	文本	内容	s:label	基本元素-矩形框显示文本
		字体	s:label.font	矩形框文本的字体
		文本颜色	s:label.color	矩形框文本的颜色
		背景	s:label.background	矩形框文本的背景颜色
		位置	s:label.position	矩形框文本的位置
		横偏移	s:label.offset.x	矩形框文本的横向偏移量
		纵偏移	s:label.offset.y	矩形框文本的纵向偏移量
		最大长度	s:label.max	
		角度	s:label.rotation	矩形框文本的角度
		缩放	s:label.scale	矩形框文本的缩放

填充（背景）	背景颜色	s:cvs_shape.background_tag	配置元素的背景颜色匹配规则，一般通过变量值去匹配，有闪烁颜色和背景颜色两种类型
	渐进	:shape.gradient	配置元素背景的渐进方式
	渐进色	s:shape.gradient.color	配置元素背景的渐进颜色
	图案叠加	s:shape.repeat.image	
	比例填充	a:background.fillValue	通过配置大小值与填充比例范围，绑点的点值确定当前填充比例
	裁切比例	s:shape.fill.clip.percentage	配置元素背景颜色的填充比例
	裁切方向	s:shape.fill.clip.direction	配置元素背景颜色的填充方向
填充（前景）	前景颜色	s:shape.foreground	配置元素的前景颜色
	渐进	s:shape.foreground.gradient	配置元素前景色的渐进方式
	渐进色	s:shape.foreground.gradient.color	配置元素前景色的渐进颜色
	裁切比例	s:shape.foreground.clip.percentage	配置元素前景色的填充比例
	裁切方向	s:shape.foreground.clip.direction	配置元素前景色的填充方向
边框	宽度	s:shape.border.width	配置元素的边框宽度
	绝对宽度	s:shape.border.width.absolute	配置元素的绝对宽度，不受放大缩小影响
	颜色	a:cr_shape.border.color_tag	配置边框颜色及闪烁效果
	样式	s:shape.border.pattern	配置边框虚实线样式
	立体	s:shape.border.3d	配置元素是否展示为立体效果
	立体颜色	s:shape.border.3d.color	配置元素立体效果颜色
	交汇样式	s:shape.border.join	配置元素边框交汇处的样式，有圆角、斜角、尖角
	线帽样式	s:shape.border.cap	配置元素边框线帽的样式，有方型、圆形、平直
	虚线	s:shape.dash	配置元素边框样式为虚线
	样式	s:shape.dash.pattern	边框虚线样式
	颜色	s:shape.dash.color	边框虚线颜色
	偏移	s:shape.dash.offset	边框虚线偏移
	宽度	s:shape.dash.width	边框虚线宽度
	立体	s:shape.dash.3d	边框虚线立体效果
	立体颜色	s:shape.dash.3d.color	边框虚线立体效果的颜色
闪烁（背景）	是否闪烁	a:cr_shape.background_isflash_tag	闪烁使能，只有勾选或者触发之后，才有闪烁能力
	闪烁颜色	a:cr_shape.background.flashcolor_tag	闪烁2个颜色之一，可配置，另一个颜色是填充中的背景色
	闪烁频率	\	闪烁的频率
闪烁（边框）	是否闪烁	a:cr_shape.border.color.isflash_tag	边框闪烁使能，只有勾选或者触发之后，才有闪烁能力
	闪烁颜色	a:cr_shape.border.color.flashcolor_tag	边框闪烁2个颜色之一，可配置，另一个颜色是填充中的背景色
	闪烁频率	\	边框闪烁频率
选中	颜色	s:select.color	配置元素选中时的选中效果的颜色
	宽度	s:select.width	配置元素选中效果的宽度
	边距	s:select.padding	配置元素选中效果的边距
	类型	s:select.type	配置元素选中效果的类型

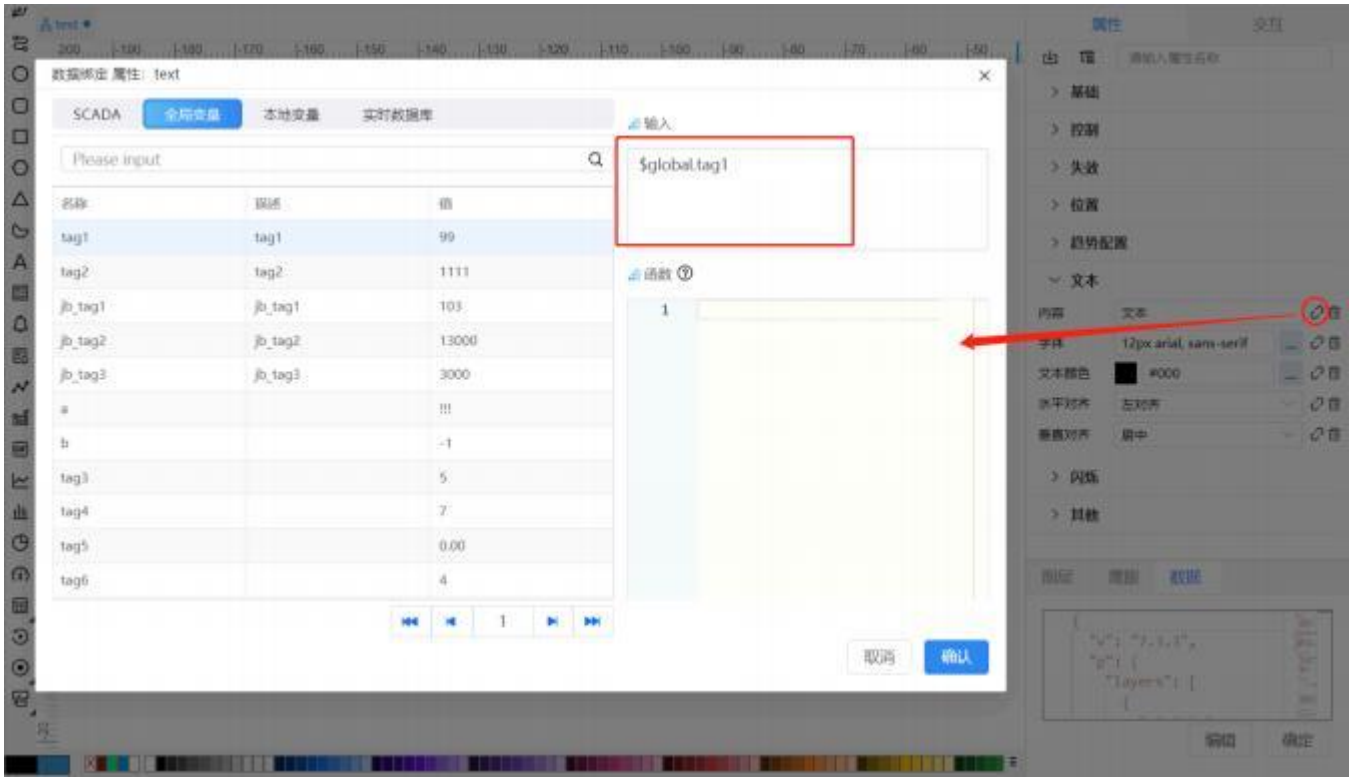
图元设计器中画布属性	扩展	模板	a:cvs_template	配置图元的类模板
		类型	\	配置图元的类型
	控制	渲染元素	\	可写脚本添加图元的渲染元素
		可见	a:visible	配置图元是否可见
		裁剪	a:clip	
		可滚动	a:scrollable	配置图元在运行态是否可以通过鼠标滚动
		可交互	a:interactive	配置运行态是否可以和图元交互
		隐藏选中边框	a:disableSelectedBorder	
		像素精确	a:pixelPerfect	配置运行态图元是否精确像素渲染
		缓存规则	a:cacheRule	配置运行态图元渲染的缓存规则
		贴图缓存	a:textureCache	勾选后会缓存图元中的贴图
		适配大小	a:fitSize	配置运行态图元是否自适应大小
	基础	预览地址	\	
		快照地址	\	
		背景	\	配置图元画布背景色
		连接动作	\	
		宽度	a:width	图元的宽度
		高度	a:height	图元的高度
		边界扩充	a:borderExtend	配置图元渲染边界扩充的像素大小
	网格指导线	网格可见	\	勾选则显示网格
		网格大小	\	配置图元画布的网格大小
		深线每隔	\	配置图元画布的深色网格线间隔
		浅线颜色	\	配置图元画布的浅色网格颜色
		深线颜色	\	配置图元画布的深色网格线颜色
		网格角度	\	配置图元画布的网格左上角的角度
		网格旋转	\	配置图元画布的网格旋转的角度
		缩放临界值	\	调整图元画布的缩放显示网格的临界值，缩放超过临界值则不显示网格

图元设计器中元素属性	控制	可见	s:2d.visible	配置图元元素在运行态是否可见
		像素精确	s:pixelPerfect	配置图元元素是否精确像素渲染
	基础	名称	displayName	配置图元元素的在画面的名称
		分组	prefix	配置图元元素的分组，同一分组共享扩展属性配置
	位置	区域	e:rect	图元元素所在区域的坐标集合
		横坐标	\	图元元素的横坐标
		纵坐标	\	图元元素的纵坐标
		锁定比例	\	勾选则锁定比例
		宽度	\	图元元素的宽度
		高度	\	图元元素的高度
		角度	p:rotation	调整图元元素的角度(绑定里使用PI计算,例如 PI/2 表示 90°)
	失效	交互有效	s:interactiveEnable	元素被点击事，是否能够响应
		失效背景色	s:disableBackground	失效状态下，元素背景色
		失效边框色	s:disableBordercolor	失效状态下，元素的边框色
	填充	背景	s:cr_background_tag	配置图元元素的背景
		图案叠加	s:shape.repeat.image	在原来的图元元素上叠加图案
		渐进	s:shape.gradient	配置元素前景色的渐进方式
		渐进色	s:shape.gradient.color	配置元素前景色的渐进颜色
		裁切比例	s:shape.fill.clip.percentage	配置元素前景色的填充比例
		裁切方向	s:shape.fill.clip.direction	配置元素前景色的填充方向
		fillRule	s:shape.fill.rule	选择非零环绕或者奇偶环绕
	边框	颜色	s:cr_borderColor_tag	配置图元元素边框颜色
		宽度	s:shape.border.width	配置图元元素的边框宽度
		绝对宽度	s:shape.border.width.absolute	配置图元元素的边框宽度是否为绝对宽度(宽度大小不受拉伸影响)
		样式	s:shape.border.pattern	配置图元元素的边框样式
		立体	s:shape.border.3d	配置图元元素是否立体显示
		立体颜色	s:shape.border.3d.color	配置图元元素的立体效果的颜色
		是否闭合	e:closePath	勾选则边框闭合
	闪烁	背景边框	\	选择配置背景或者边框
		是否闪烁	s:cr_background_isflash_tag	选择是否闪烁
		闪烁颜色	s:cr_background_flashcolor_tag	配置闪烁颜色
		闪烁频率	\	配置闪烁频率
	其他	显隐频率	s:cvs_flashrate	配置图元中的元素的显隐频率
		裁剪比例	s:clipPercentage	配置图元中的元素的裁剪比例
		裁剪方向	s:clipDirection	配置图元中的元素的裁剪方向
		透明度	s:opacity	配置图元中的元素的透明度

5.5 元素属性绑点

属性绑点

绑点如下



- 基础：设置名称、标签
- 控制：设置可见、像素精确
- 失效：设置交互有效、失效背景色、失效边框色
- 位置：设置横坐标、纵坐标、锁定比例、宽度、高度、角度、横缩放、纵缩放、横锚点、

纵锚点、垂直、水平、全*屏、节点吸附

- 趋势配置：是否显示趋势、曲线颜色、曲线风格、曲线宽度、显示方式、值标记
- 文本：内容、字体、文本颜色、背景、位置、横偏移、纵偏移、最大长度、角度、缩放
- 闪烁（背景、边框）：是否闪烁、闪烁颜色、闪烁频率
- 其他：显隐频率、权限名称、防抖延迟、类名、图层、提示、透明度、灰化、染色、裁切比

例、裁切方向、不规则图形

- 填充（背景）：背景颜色、渐进、渐进色、裁切方向、裁切比列、图案叠加
- 填充（前景）：前景颜色、渐进方向、渐进色、裁切方向、裁切比列
- 边框：宽度、绝对宽度、颜色、样式、立体、立体颜色、交汇样式、线帽样式、颜色、偏

移、宽度、立体、立体颜色

注释：如果元素绑的是TXT类型的tag点，如果该tag点类型在DA中修改，编辑态需要从重新选点，更改记录的类型。

表达式

注意：组态表达式、联动表达式中的点名包含特殊字符如-:._()[]+=%@#等时，需要用反引号字符``将点名包含起来，如`scada.t+a-g.1`

1. 新增二进制、八进制和十六进制数的解析

其进制转换和JavaScript的对于其他进制语法转换一致。

二进制以0b或0B开头，0b后跟大于1的数是非法的，如0b123非法，八进制以0开头，注意：0后跟的所有如果都小于等于7则其为八进制数，如果其后跟的有大于7的，则为10进制数（和JS数值语法特性致）。077，其为八进制数，值为63。078，其为十进制数，值为78，十六进制以0x或0X开头

```
0b10 & 0b111
073 * 2
13 | 0xFFf
```

2.常量

常量	对应	说明
E	Math.E	自然对数底数
PI	Math.PI	圆周率
true	布尔值true	true
false	布尔值true	false

3.运算符

在绑点的表达式中支持的操作符，对应函数参参考

操作符	对应函数	说明	优先级
+	add	加法	8
-	sub/neg	减法/取相反数	8
*	mul	乘法	9
/	div	除法	9
%	mod	取余	9
^	Math.pow	*在该表达式中表示乘方，若要表示异或，需使用bxor函数*	9
,	append	存储	-1
\	\		or
>	gt	大于	6
<	lt	小于	6
&&	and	逻辑与	1
>=	ge	大于等于	6
<=	le	小于等于	6

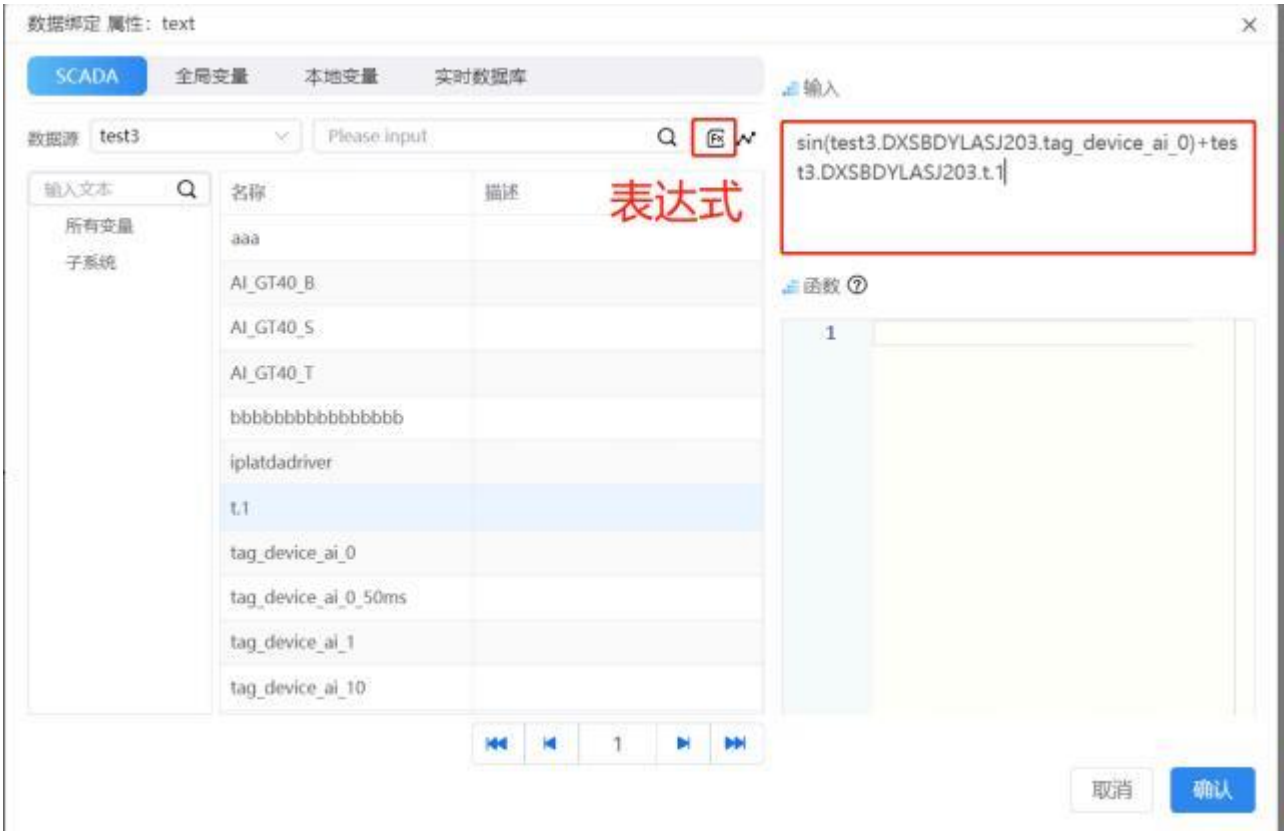
==	equal	JS中的==	5
>>	bright	有符号按位右移	7
<<	bleft	按位左移	7
\		bor	按位或
&	band	按位与	4
!=	notequal	JS中的!=	5
!	not	是否非0数	10
~	bnot	按位取反	10
()		括号	10

4.方法

方法	参数	对应JS操作	说明
sin	(x)	Math.sin	正弦
cos	(x)	Math.cos	余弦
tan	(x)	Math.tan	正切
asin	(x)	Math.asin	反正弦
acos	(x)	Math.acos	反余弦
atan	(x)	Math.atan	反正切
sqrt	(x)	Math.sqrt	平方根
log	(x)	Math.log	自然对数
abs	(x)	Math.abs	绝对值
ceil	(x)	Math.ceil	大于或等于的最小整数
floor	(x)	Math.floor	小于或等于的最大整数
round	(x)	Math.round	四舍五入
exp	(x)	Math.exp	e的x次方
random	(x)	Math.random()*(x|1)	随机数

fac	(x)	x*(x-1)*…*1	x的阶乘
min	(x,y,...)	Math.min	一组数中的最小值
max	(x,y,...)	Math.max	一组数中的最大值
pyt	(x, y)	Math.sqrt(x*x+y*y)	根号下x的平方加上y的平方
pow	(x, y)	Math.pow	乘方
atan2	(x)	Math.atan2	点的偏移角
correl	(x, y, z)	x ? y : z	三元表达式
bnot	(x)	~x	按位取反
band	(x, y)	x & y	按位与
bor	(x, y)	x | y	按位或
bxor	(x, y)	x ^ y	按位异或
bleft	(x, y)	x << y	按位左移
bright	(x, y)	x >> y	有符号按位右移
brightx	(x, y)	x >>> y	无符号按位右移
not	(x)	!Number(x)	判断是否为非0数
gt	(x, y)	x > y	大于
lt	(x, y)	x < y	小于
strconcat	(x, y)	"" + x + y	字符串拼接
eq	(x, y)	x == y	判断是否相等
neq	(x, y)	x != y	是否不相等
todecimal	(x, y)	parseInt(x,y)	进制转换

5.表达式示例



绑域名

1.实时数据库点绑域名规则

不支持注册以域名为结尾的点。比如.value，.quality，.timestamp，F_CV等结尾的点。

举例说明：

实时数据源名称为：ihdtest

点名：hdsys_tag_test.value，hdsys_tag_test.timestamp，hdsys_tag_test.quality，hdsys_tag_test.F_CV注册会报错：110757 Tag点不存在

2.规避方案

注册以域名为结尾的点：比如.value，.quality，.timestamp，F_CV等结尾的点。点的格式需要特殊处理，格式为：数据源名称+.+点名+.+F_CV

举例说明：

实时数据源名称为：ihdtest

点名 hdsys_tag_test.value，hdsys_tag_test.timestamp，hdsys_tag_test.quality，hdsys_tag_test.F_CV需求：推送点的value，quality，timestamp，F_CV域的值

注册点名结尾时域名的点的书写格式：数据源名称+.+点名+.+F_CV

ihdtest.hdsys_tag_test.value.F_CV

ihdtest.hdsys_tag_test.ti

mestamp.F_CV

ihdtest.hdsys_tag_test.qu

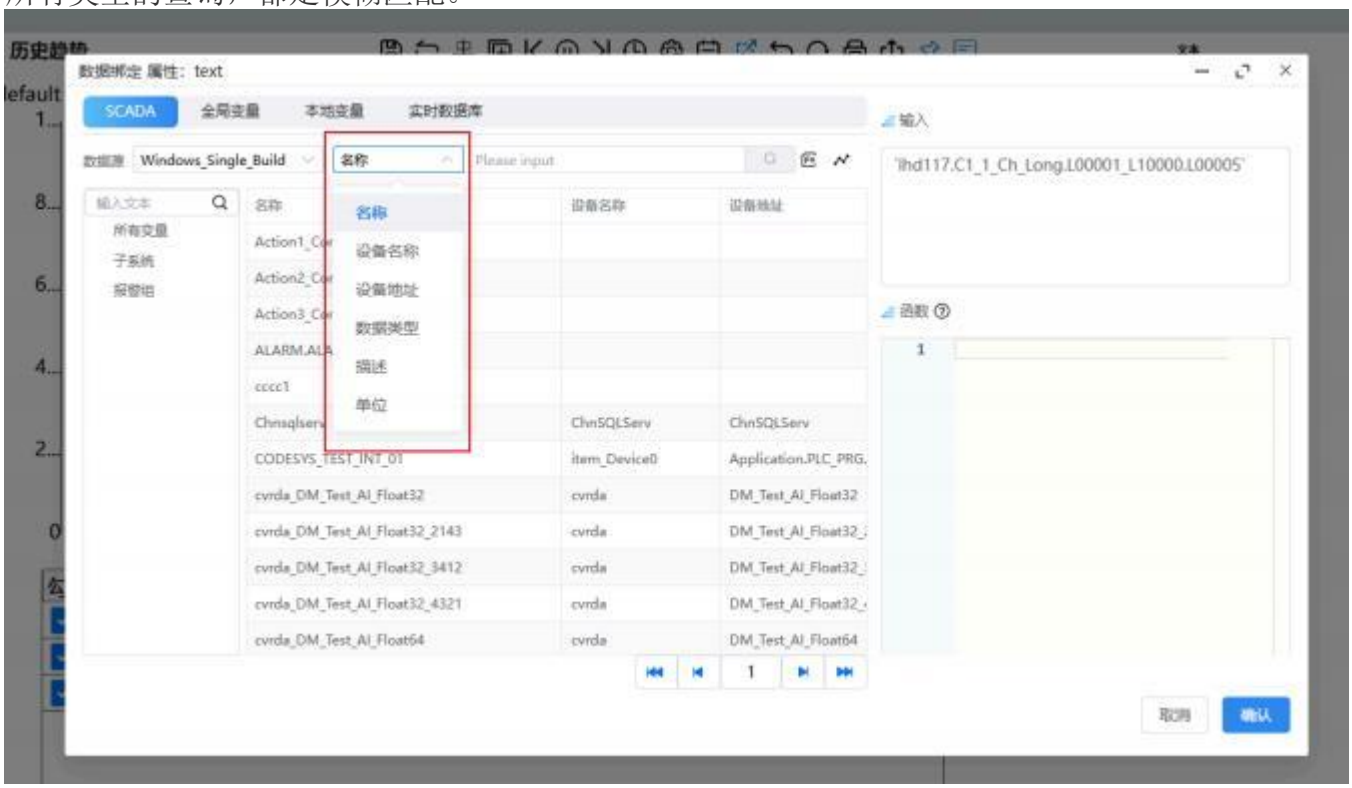
ality.F_CV

ihdtest.hdsys_tag_test.F_

CV.F_CV

选择过滤类型

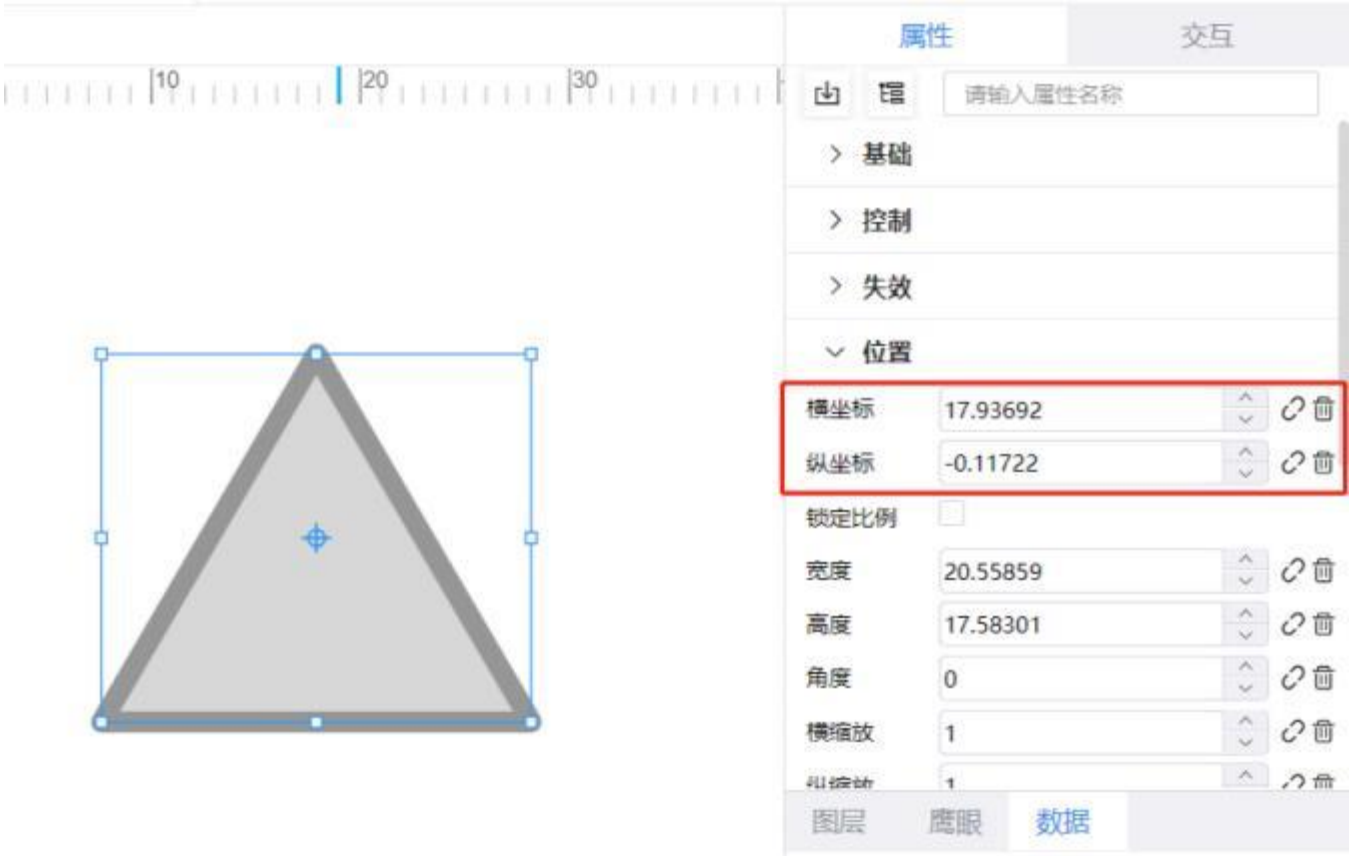
iplat-cm-5.28.0(含)以上版本，SM适配版本iplat-sm-5.13.3(含)以上版本。
在搜索功能输入框前，新增下拉框，选择过滤类型。
目前SCADA支持所有类型属性的过滤。
所有类型的查询，都是模糊匹配。



5.6 元素坐标与定位

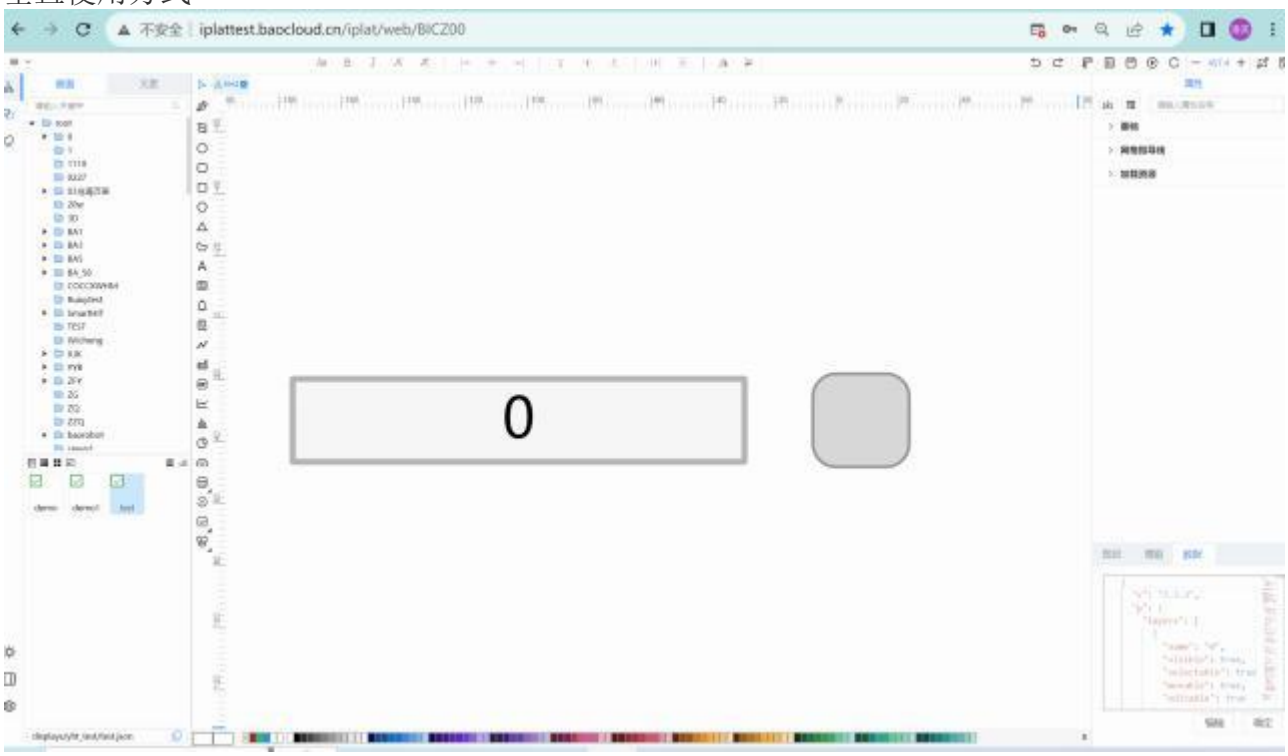
元素坐标

横坐标和纵坐标可设置定位元素的位置



垂直水平使用方式如下：

1. 垂直使用方式

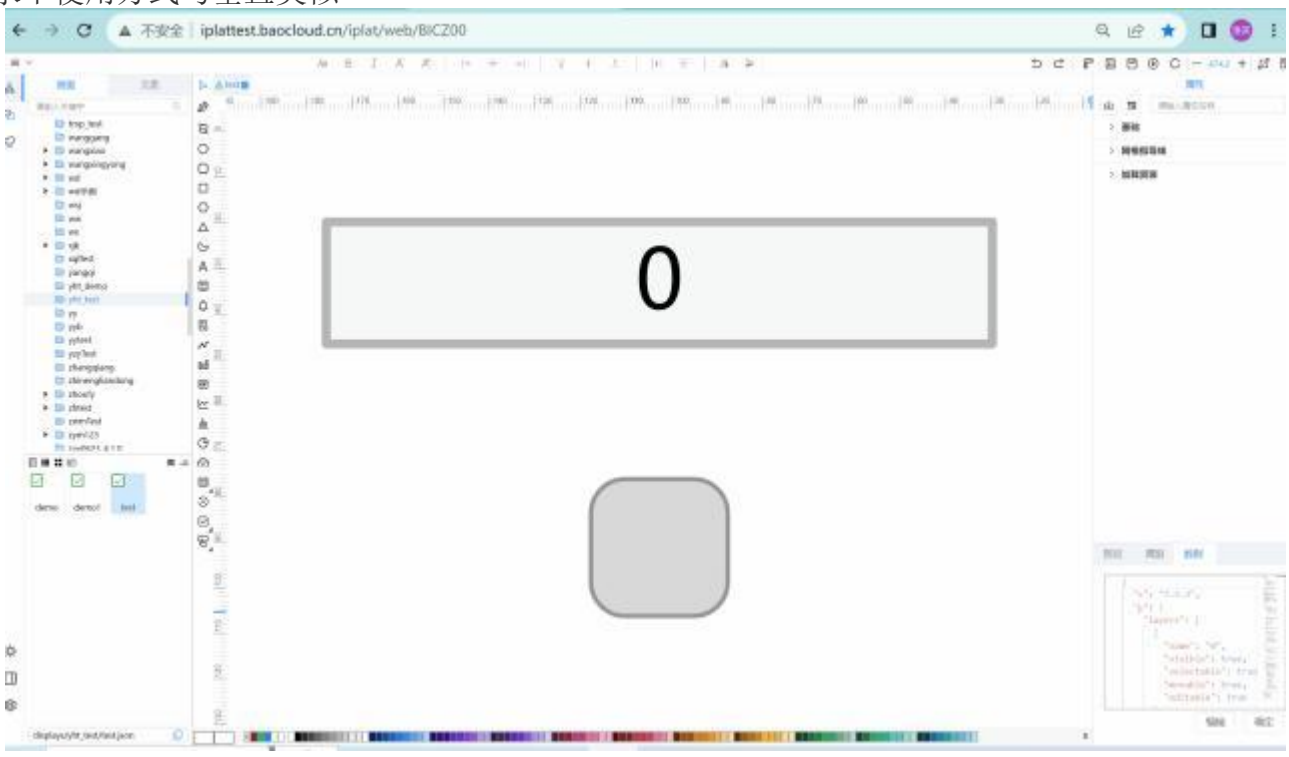


“值”：指绑定点的值变换范围；
“位置”：指元素以自身原本坐标为原点，向上和向下的移动范围
绑定点的值和元素的位置存在一个线性映射关系，每一个值对应元素所在的一个坐标位置。

如上示例所示，当绑定点的值为0时，元素到达原坐标下方50个单位的位置；当绑定点的值为25时，元素到达原坐标下方25个单位的位置；
当绑定点的值为50时，元素到达原坐标的位置；当绑定点的值为75时，元素到达原坐标上方25
个单位的位置；
当绑定点的值为100时，元素到达原坐标上方50个单位的位置。

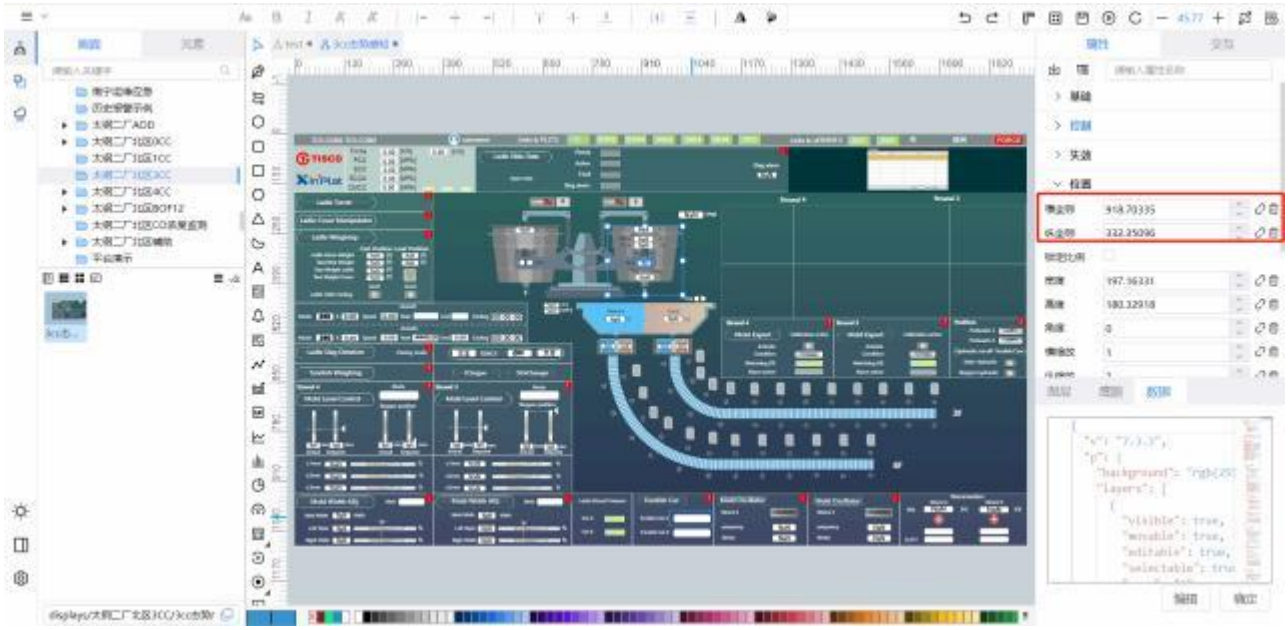
也就意味着随着绑定点的值的变化，元素坐标可以在垂直方向的限定范围内变换。

2. 水平使用方式与垂直类似

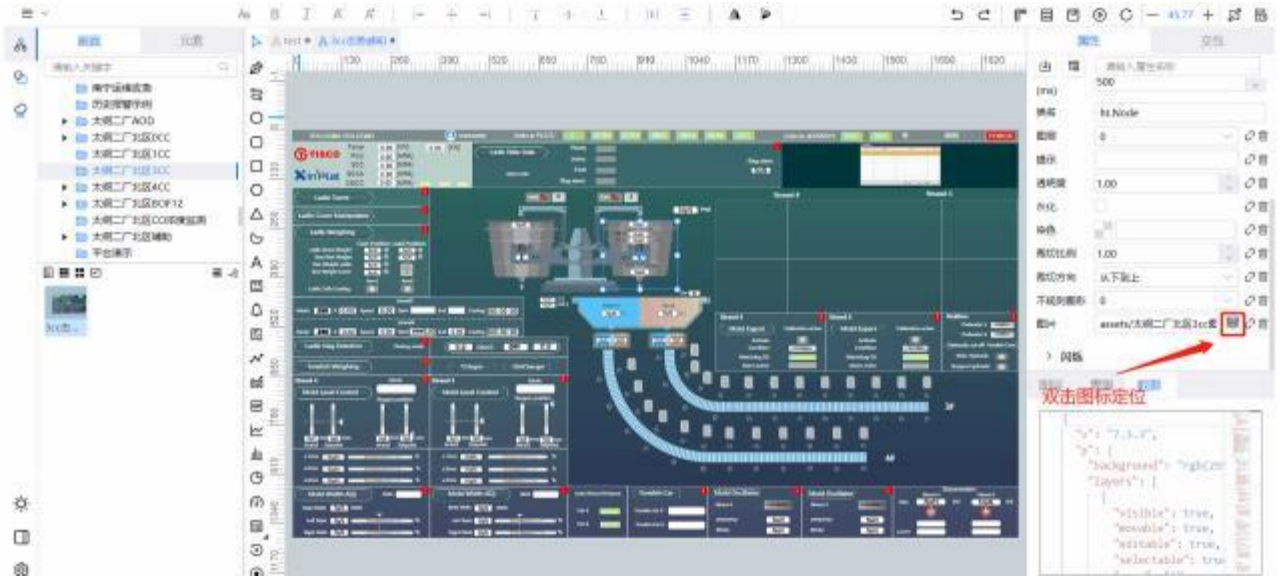


元素定位

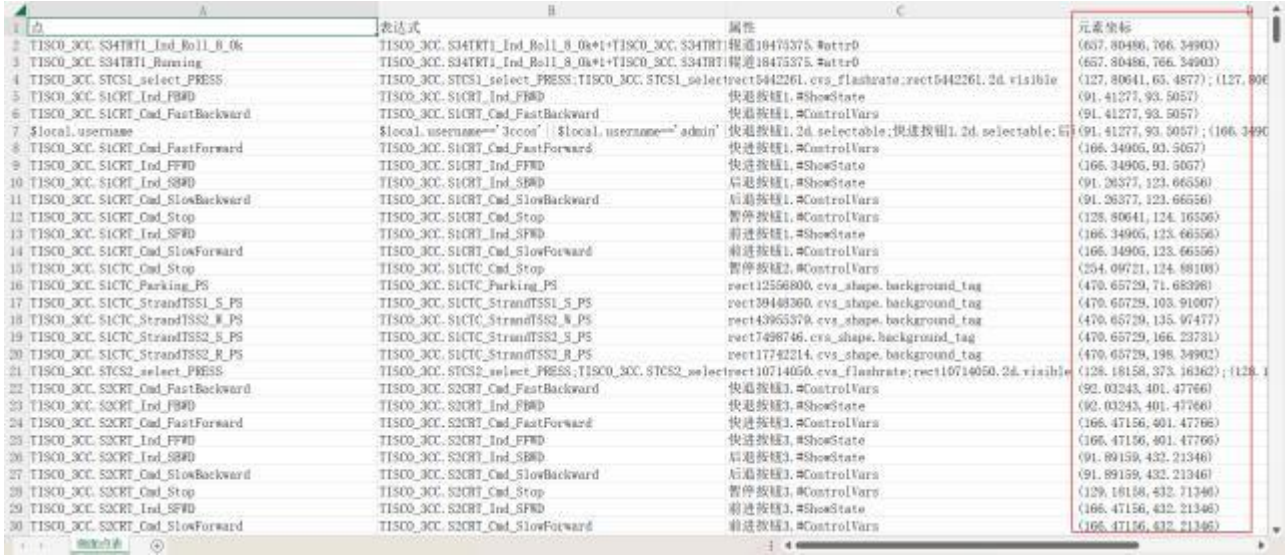
1. 坐标定位



2. 画面中图元、资源双击图标定位



3. 画面右键“导出画面点”中有坐标信息



	点	表达式	属性	元素坐标
2	TISCO_3CC.S34TRT1_Ind_Roll_8_Ok	TISCO_3CC.S34TRT1_Ind_Roll_8_Ok*1+TISCO_3CC.S34TRT1	报通18475375.#attr0	(657.80486,766.34903)
3	TISCO_3CC.S34TRT1_Running	TISCO_3CC.S34TRT1_Ind_Roll_8_Ok*1+TISCO_3CC.S34TRT1	报通18475375.#attr0	(657.80486,766.34903)
4	TISCO_3CC.STCS1_select_PRESS	TISCO_3CC.STCS1_select_PRESS;TISCO_3CC.STCS1_select	rect5442261.cvs_flashrate;rect5442261.2d.visible	(127.80641,65.4877);(127.806
5	TISCO_3CC.S1CRT_Ind_FBWD	TISCO_3CC.S1CRT_Ind_FBWD	快退按钮1.#ShowState	(91.41277,93.5057)
6	TISCO_3CC.S1CRT_Cmd_FastBackward	TISCO_3CC.S1CRT_Cmd_FastBackward	快退按钮1.#ControlVars	(91.41277,93.5057)
7	$local.username	$local.username=='3ccos' || $local.username=='admin'	快退按钮1.2d.selectable;快进按钮1.2d.selectable;后	(91.41277,93.5057);(166.3490
8	TISCO_3CC.S1CRT_Cmd_FastForward	TISCO_3CC.S1CRT_Cmd_FastForward	快进按钮1.#ControlVars	(166.34905,93.5057)
9	TISCO_3CC.S1CRT_Ind_FFWD	TISCO_3CC.S1CRT_Ind_FFWD	快进按钮1.#ShowState	(166.34905,93.5057)
10	TISCO_3CC.S1CRT_Ind_SBWD	TISCO_3CC.S1CRT_Ind_SBWD	后退按钮1.#ShowState	(91.26377,123.66556)
11	TISCO_3CC.S1CRT_Cmd_SlowBackward	TISCO_3CC.S1CRT_Cmd_SlowBackward	后退按钮1.#ControlVars	(91.26377,123.66556)
12	TISCO_3CC.S1CRT_Cmd_Stop	TISCO_3CC.S1CRT_Cmd_Stop	暂停按钮1.#ControlVars	(128.80641,124.16556)
13	TISCO_3CC.S1CRT_Ind_SFWD	TISCO_3CC.S1CRT_Ind_SFWD	前进按钮1.#ShowState	(166.34905,123.66556)
14	TISCO_3CC.S1CRT_Cmd_SlowForward	TISCO_3CC.S1CRT_Cmd_SlowForward	前进按钮1.#ControlVars	(166.34905,123.66556)
15	TISCO_3CC.S1CTC_Cmd_Stop	TISCO_3CC.S1CTC_Cmd_Stop	暂停按钮2.#ControlVars	(254.09721,124.88108)
16	TISCO_3CC.S1CTC_Parking_PS	TISCO_3CC.S1CTC_Parking_PS	rect12556800.cvs_shape.background_tag	(470.65729,71.68398)
17	TISCO_3CC.S1CTC_StrandTSS1_S_PS	TISCO_3CC.S1CTC_StrandTSS1_S_PS	rect39448360.cvs_shape.background_tag	(470.65729,103.91007)
18	TISCO_3CC.S1CTC_StrandTSS2_N_PS	TISCO_3CC.S1CTC_StrandTSS2_N_PS	rect43955379.cvs_shape.background_tag	(470.65729,135.97477)
19	TISCO_3CC.S1CTC_StrandTSS2_S_PS	TISCO_3CC.S1CTC_StrandTSS2_S_PS	rect7498746.cvs_shape.background_tag	(470.65729,166.23731)
20	TISCO_3CC.S1CTC_StrandTSS2_R_PS	TISCO_3CC.S1CTC_StrandTSS2_R_PS	rect17742214.cvs_shape.background_tag	(470.65729,198.34902)
21	TISCO_3CC.STCS2_select_PRESS	TISCO_3CC.STCS2_select_PRESS;TISCO_3CC.STCS2_select	rect10714050.cvs_flashrate;rect10714050.2d.visible	(128.18158,373.16362);(128.1
22	TISCO_3CC.S2CRT_Cmd_FastBackward	TISCO_3CC.S2CRT_Cmd_FastBackward	快退按钮3.#ControlVars	(92.03243,401.47766)
23	TISCO_3CC.S2CRT_Ind_FBWD	TISCO_3CC.S2CRT_Ind_FBWD	快退按钮3.#ShowState	(92.03243,401.47766)
24	TISCO_3CC.S2CRT_Cmd_FastForward	TISCO_3CC.S2CRT_Cmd_FastForward	快进按钮3.#ControlVars	(166.47156,401.47766)
25	TISCO_3CC.S2CRT_Ind_FFWD	TISCO_3CC.S2CRT_Ind_FFWD	快进按钮3.#ShowState	(166.47156,401.47766)
26	TISCO_3CC.S2CRT_Ind_SBWD	TISCO_3CC.S2CRT_Ind_SBWD	后退按钮3.#ShowState	(91.89159,432.21346)
27	TISCO_3CC.S2CRT_Cmd_SlowBackward	TISCO_3CC.S2CRT_Cmd_SlowBackward	后退按钮3.#ControlVars	(91.89159,432.21346)
28	TISCO_3CC.S2CRT_Cmd_Stop	TISCO_3CC.S2CRT_Cmd_Stop	暂停按钮3.#ControlVars	(129.18158,432.71346)
29	TISCO_3CC.S2CRT_Ind_SFWD	TISCO_3CC.S2CRT_Ind_SFWD	前进按钮3.#ShowState	(166.47156,432.21346)
30	TISCO_3CC.S2CRT_Cmd_SlowForward	TISCO_3CC.S2CRT_Cmd_SlowForward	前进按钮3.#ControlVars	(166.47156,432.21346)

5.7 元素闪烁配置

元素闪烁

功能说明

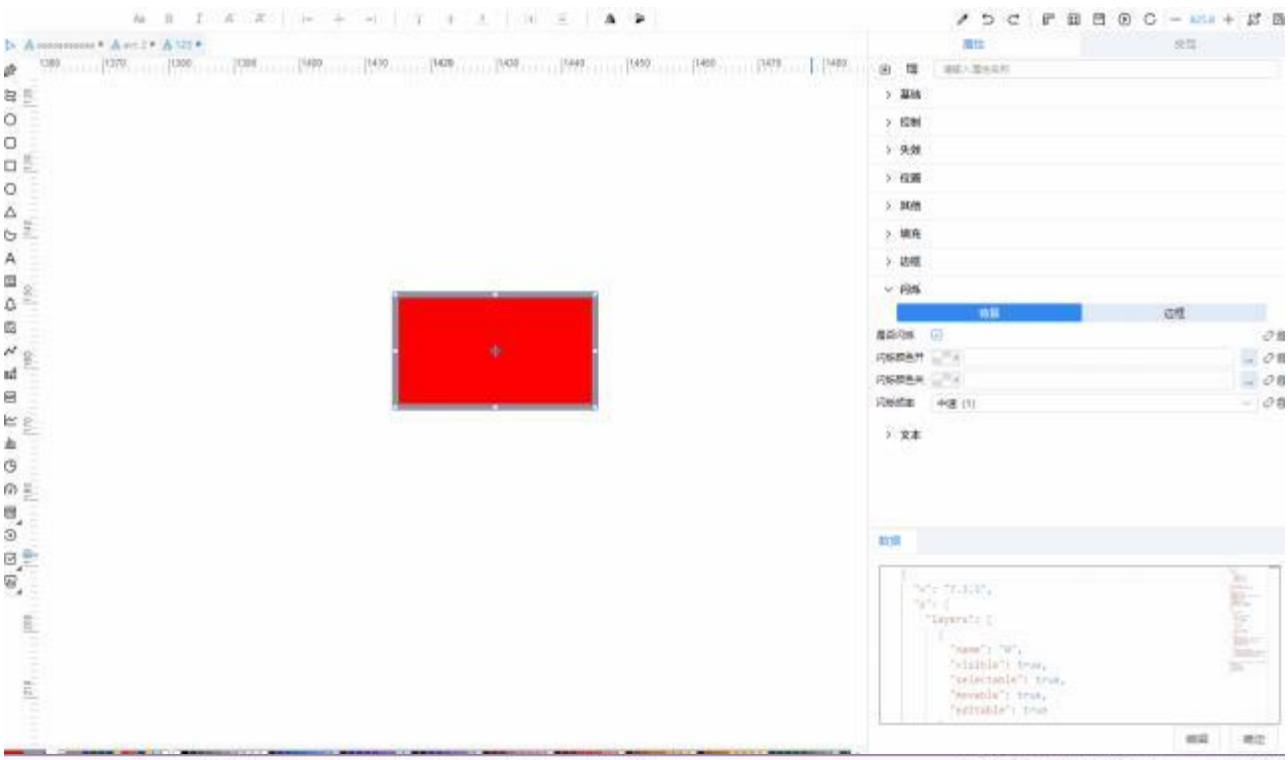
画面中元素可配置闪烁功能，设置颜色匹配规则，与点值关联，类型有背景闪烁和边框闪烁两种，配置好闪烁规则可在运行态查看相应闪烁效果

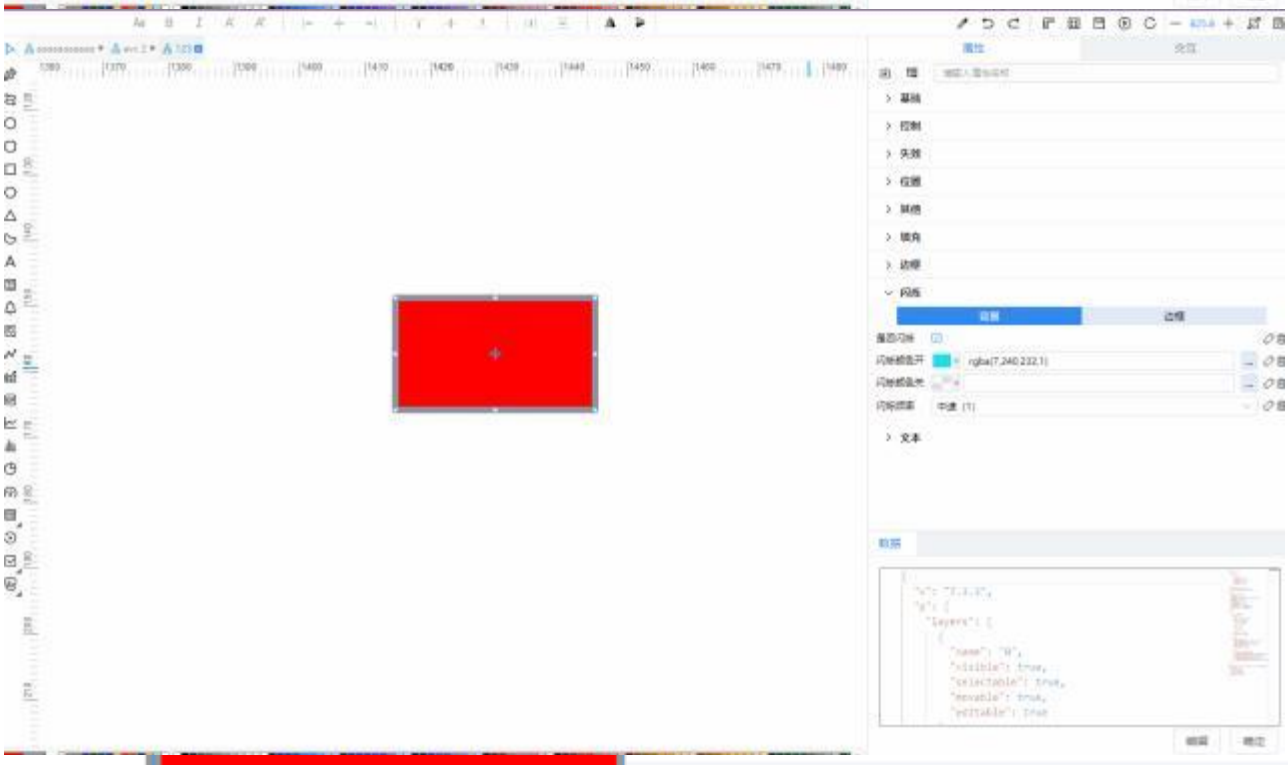
功能示例

背景闪烁：

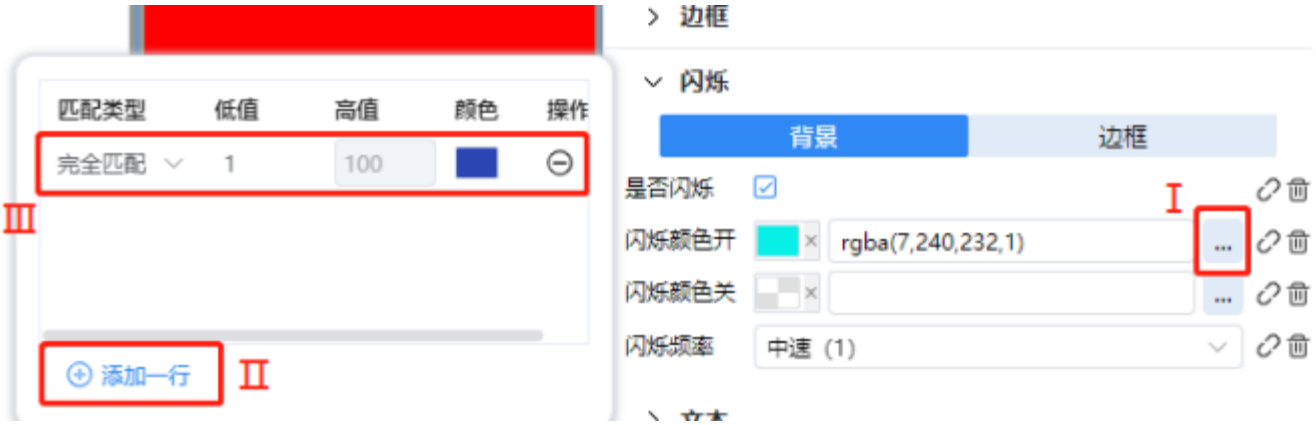
- 点击元素，在属性页选择闪烁分组，在背景配置栏勾选“是否闪烁”，不勾选此项，则背景闪烁配置不生效

- “闪烁颜色开”即为不配置匹配规则时，默认闪烁颜色，若只配置了“闪烁颜色开”，则在闪烁条件满足时，将与元素背景色交替闪烁（上图元素红色为元素背景色）；
  如果配置“闪烁颜色关”（修改默认颜色或者绑点都视为配置过此项），若闪烁条件满足时，则“闪烁颜色开/关”两个颜色交替闪烁；
- 背景色、闪烁颜色开、闪烁颜色关的值为undefined或者null，都不会闪烁。如果用户希望透明色（null值，点颜色选择器右边的X）参与闪烁，请配置rgba色的a值为0。
- 运行效果：

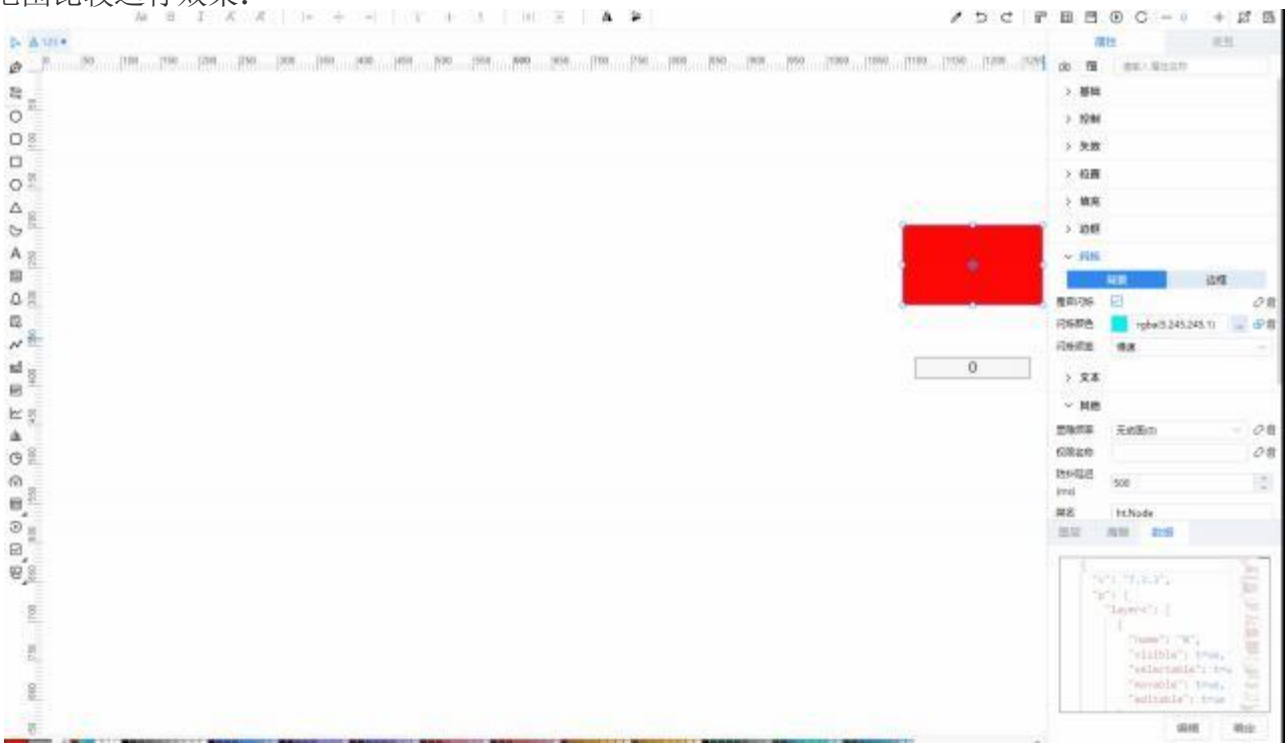




匹配类型
低值
高值
颜色
操作
范围比较
1
100
添加一行
III
II
边框
闪烁
背景
边框
是否闪烁
闪烁颜色开
rgba(7,240,232,1)
闪烁颜色关
闪烁频率
中速（1）
I

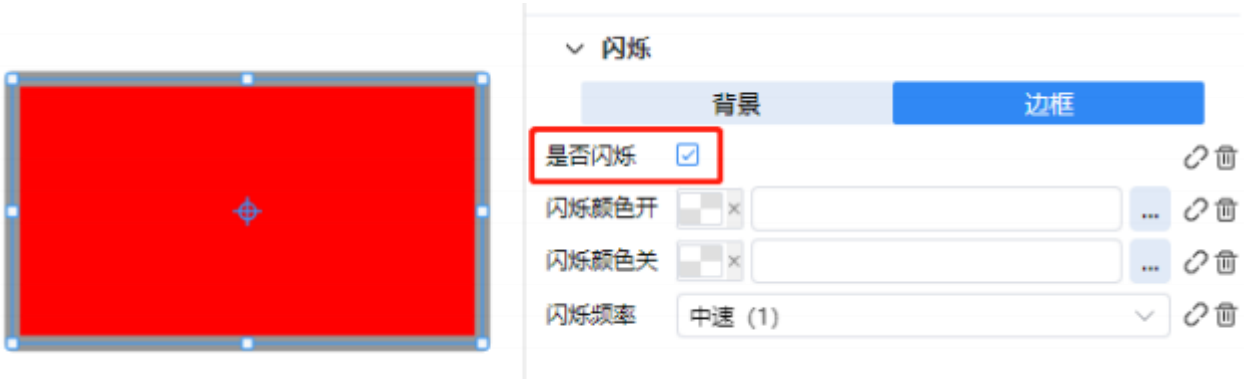


- 此项为配置闪烁规则时，按图示流程点击各项，匹配类型分为范围比较与完全匹配
  - 范围比较：绑点点值在低值-高值范围内，则与背景颜色交替闪烁；注：范围比较的取值区间为前开后闭区间
  - 完全匹配：绑点点值在匹配到某个值时，闪烁生效，与背景颜色交替闪烁；
- 设置好闪烁匹配规则后，可以通过绑点、写function方式控制，写function需要返回数字去匹配规则
- 范围比较运行效果：

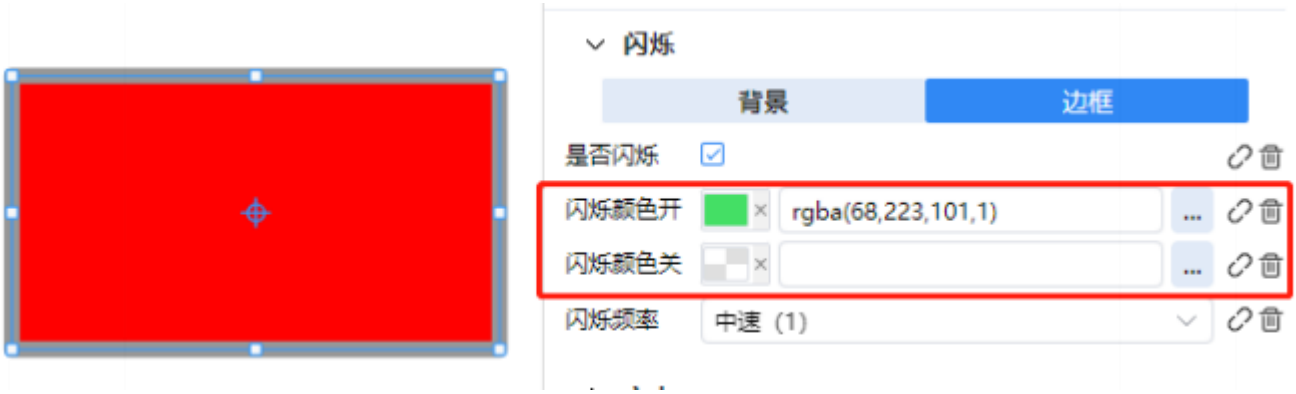


- 完全匹配运行效果：

边框闪烁



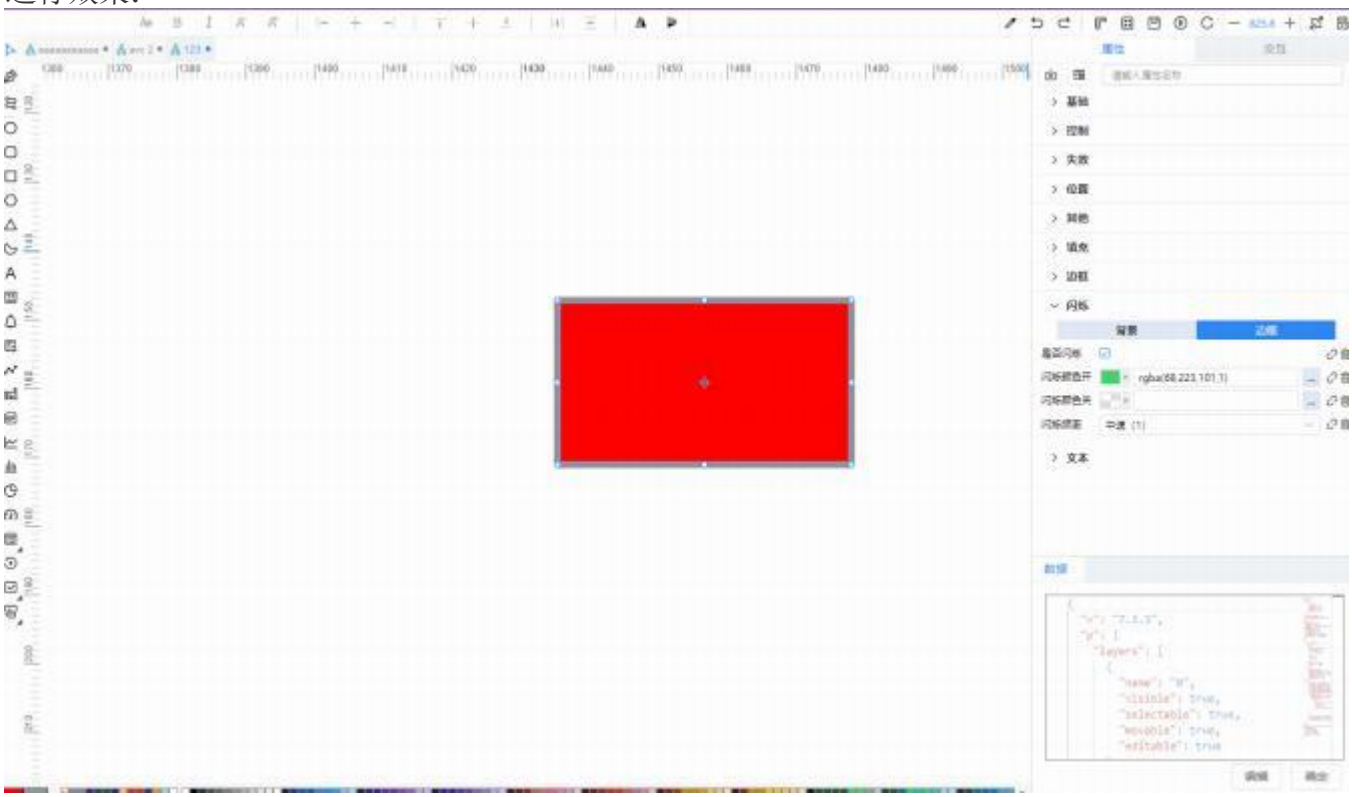
- 点击元素，在属性页选择闪烁分组，在边框配置栏勾选“是否闪烁”，不勾选此项，则边框闪烁配置不生效

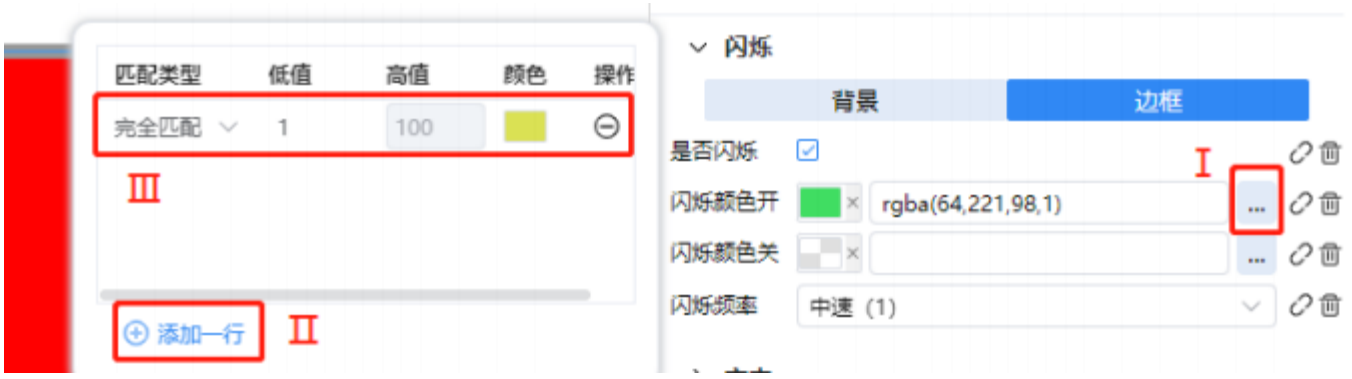


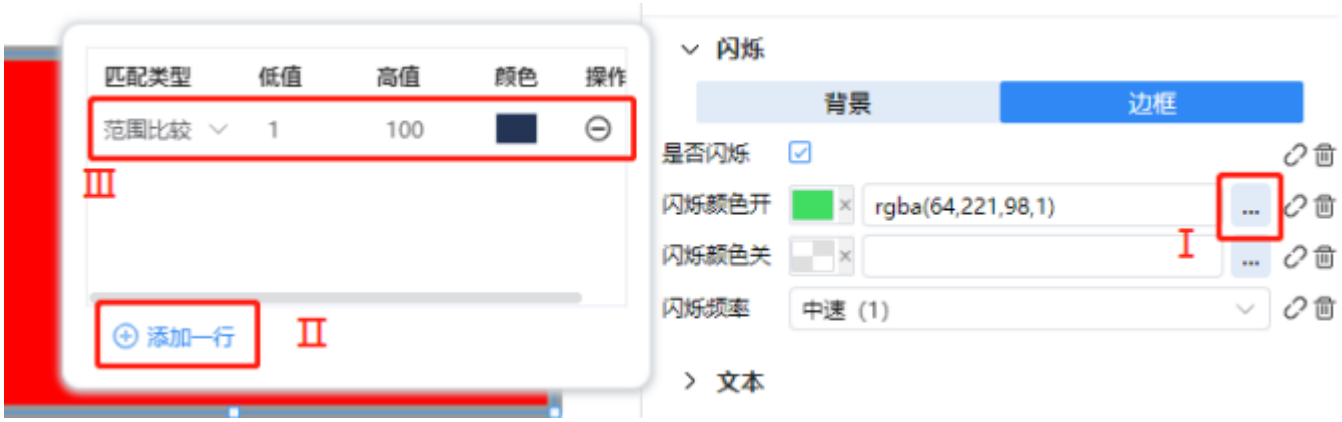
•闪烁颜色开”即为不配置匹配规则时，默认闪烁颜色，

- 若只配置了“闪烁颜色开”，则在闪烁条件满足时，将与元素背景边框色交替闪烁（上图元素灰色为元素编辑态边框色）；
- 如果配置“闪烁颜色关”（修改默认颜色或者绑点都视为配置过此项），若闪烁条件满足时，则“闪烁颜色开/关”两个颜色交替闪烁；

运行效果：

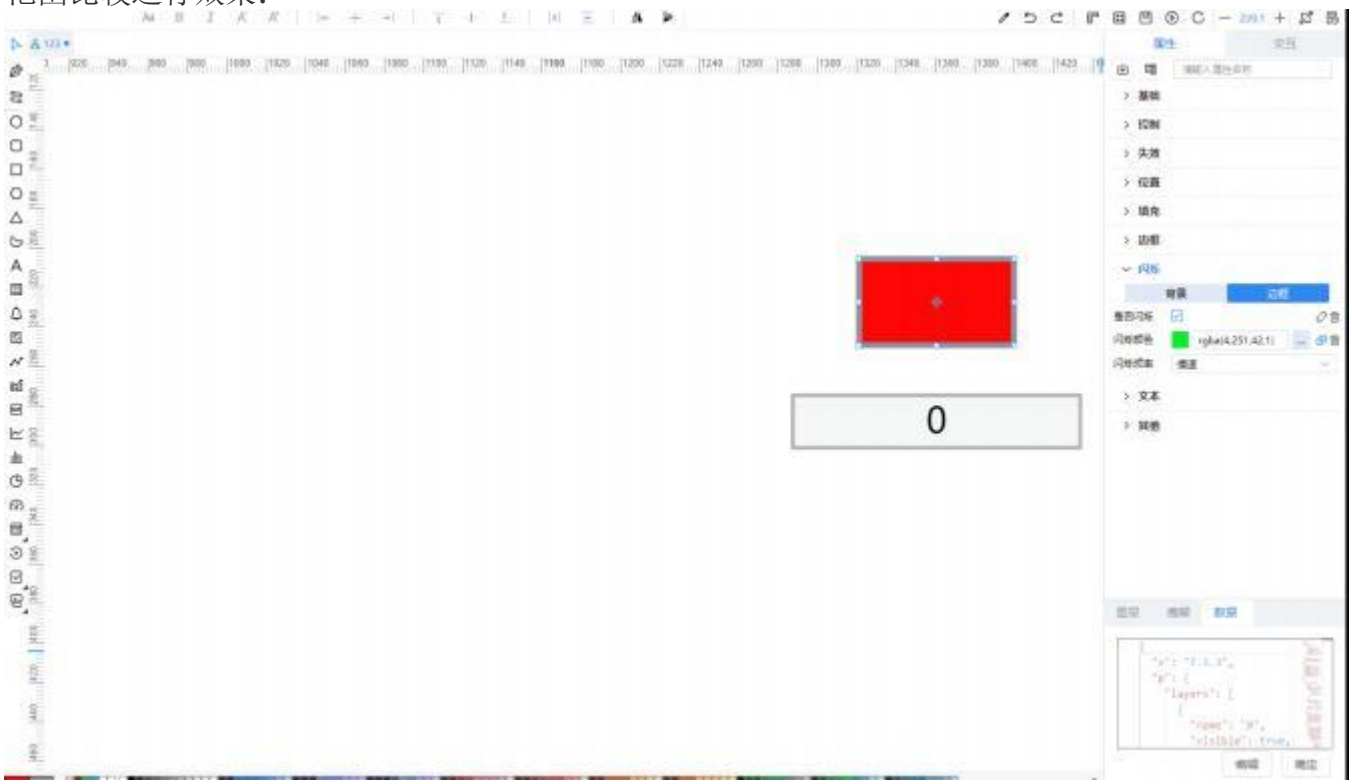


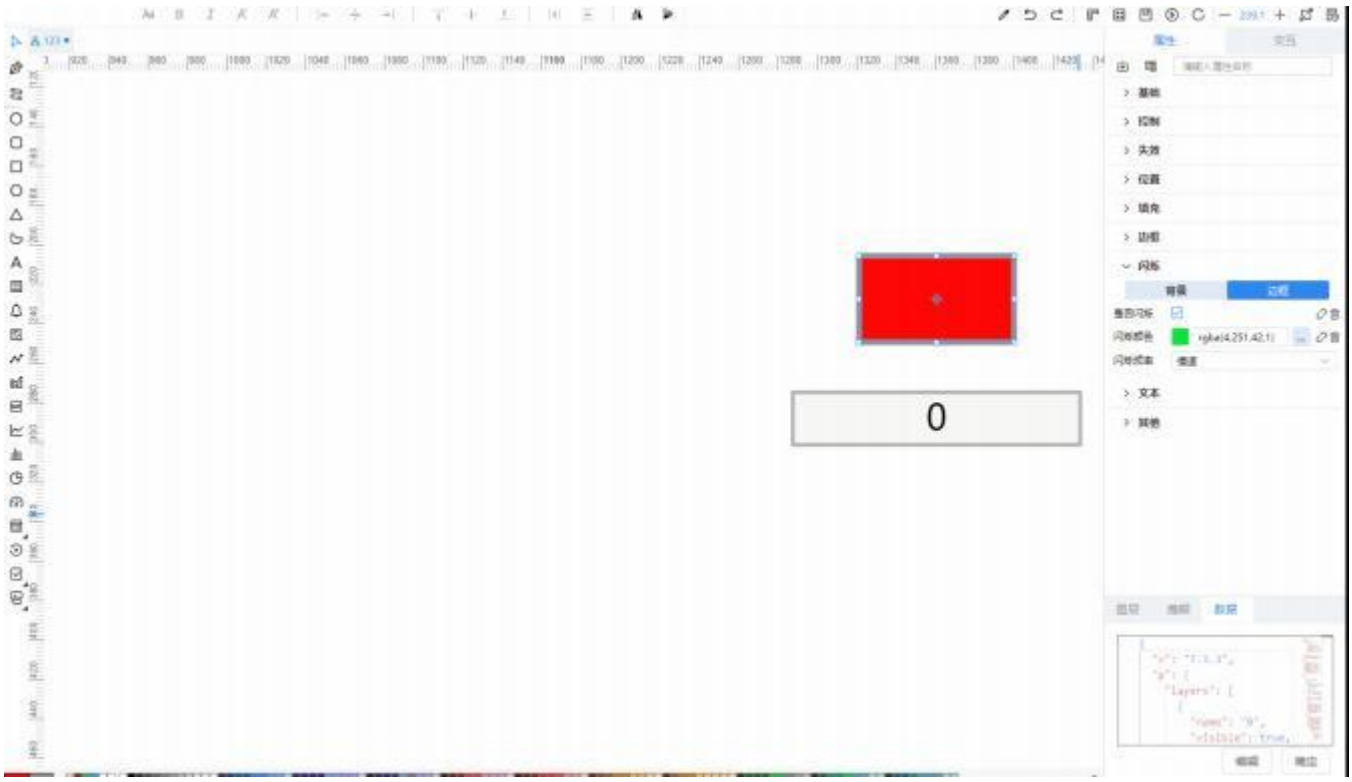




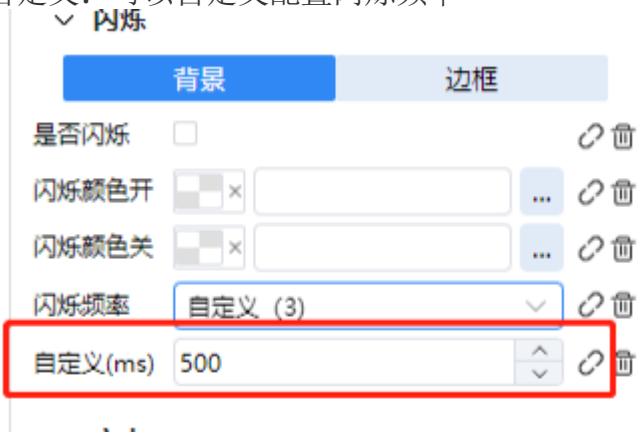
- 此项为配置闪烁规则时，按图示流程点击各项，匹配类型分为范围比较与完全匹配
  - 范围比较：绑点点值在低值-高值范围内，则与元素编辑态边框色交替闪烁；注：范围
    比较的取值区间为前开后闭区间
  - 完全匹配：绑点点值在匹配到某个值时，闪烁生效，与元素编辑态边框色交替闪烁；
- 设置好闪烁匹配规则后，可以通过绑点、写function方式控制，写function需要返回数字去匹配规则

范围比较运行效果：





- 闪烁频率：即元素闪烁速度，可选择慢速、中速、快速
- 自定义：可以自定义配置闪烁频率

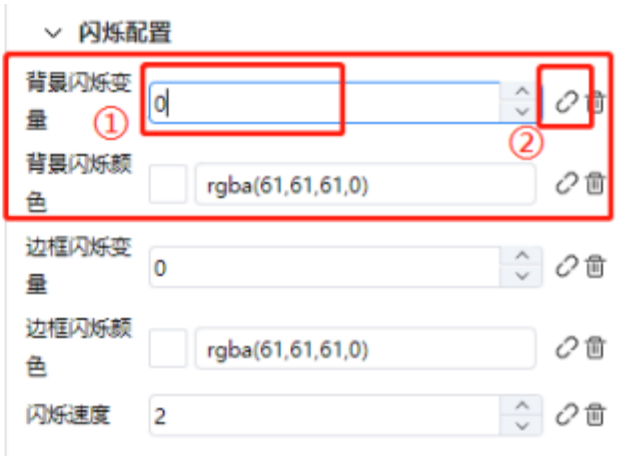


图元闪烁

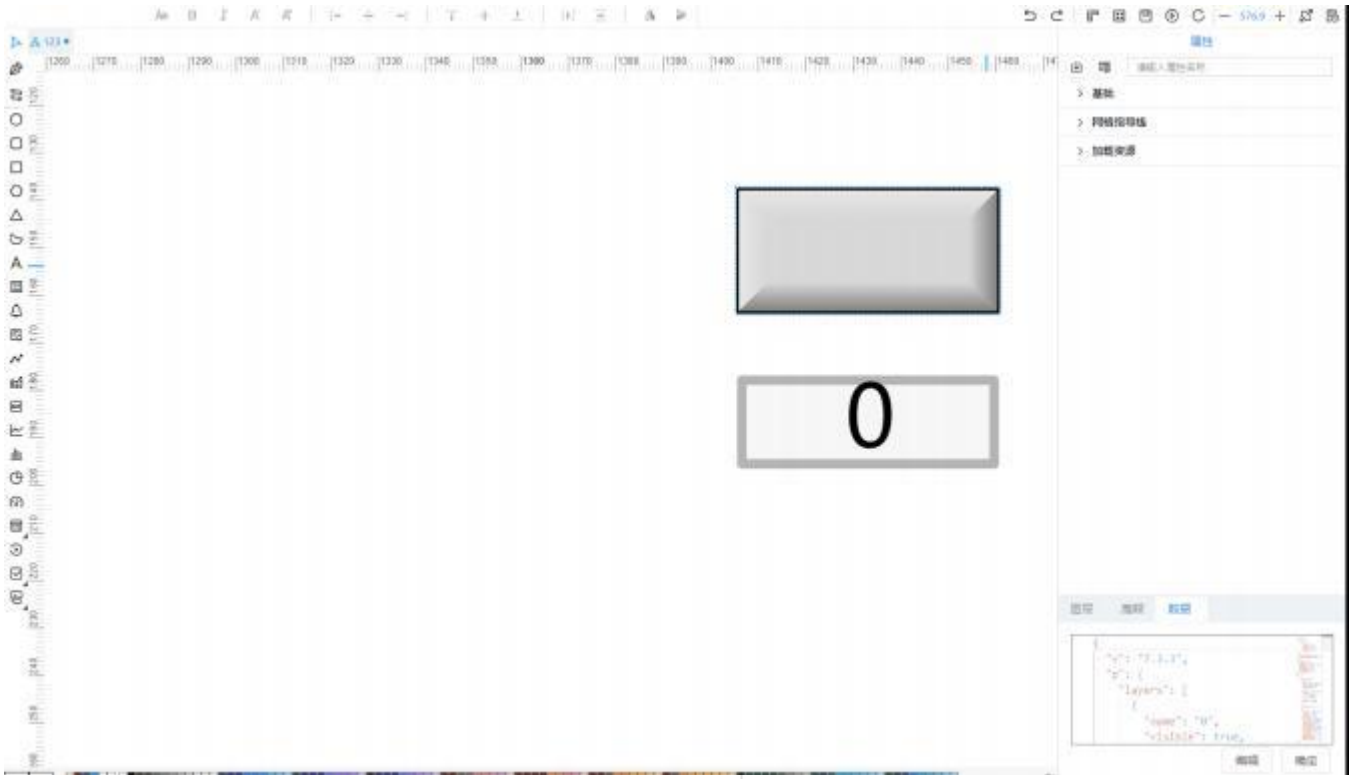
功能说明

画面中图元可配置闪烁功能，设置颜色匹配规则，与点值关联，类型有背景闪烁和边框闪烁两种，配置好闪烁规则可在运行态查看相应闪烁效果

功能示例

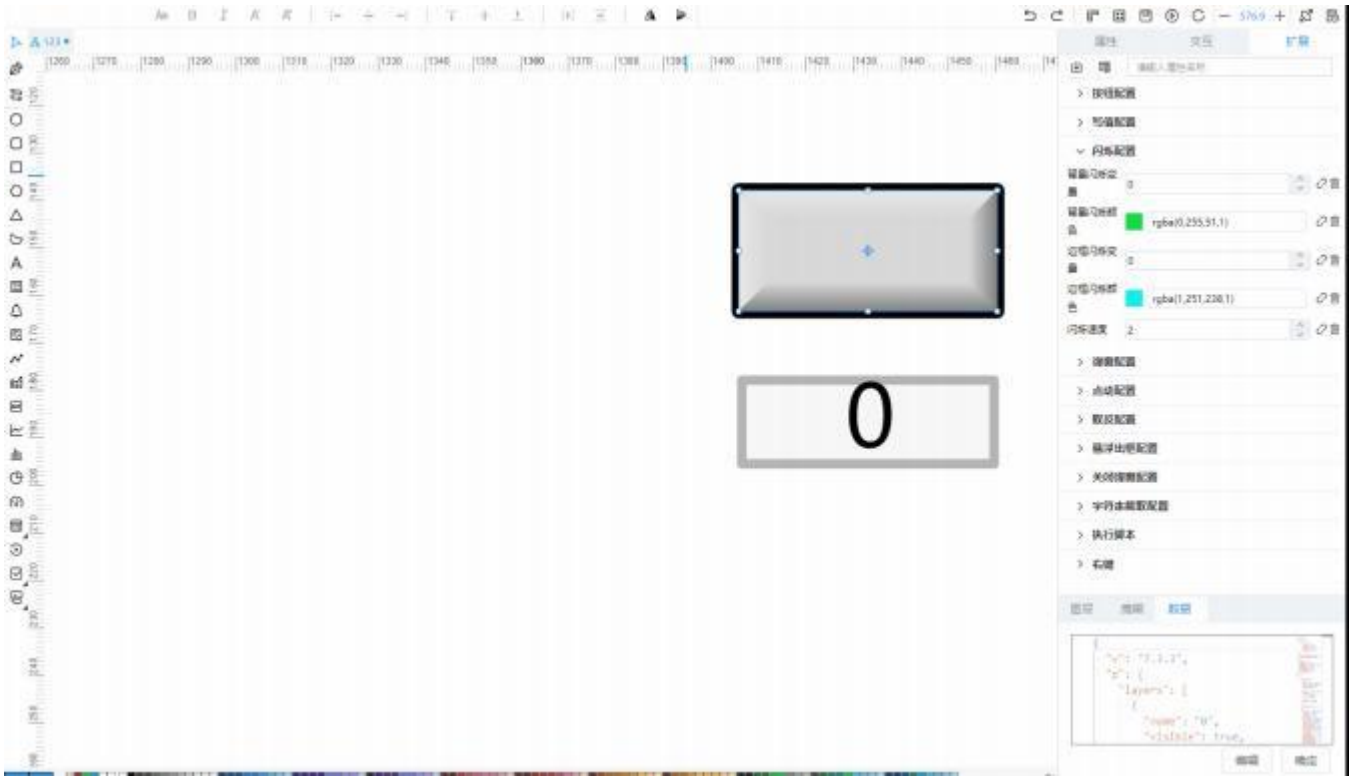


•背景闪烁变量：可在②处绑定图元闪烁点；若①处设置不等于0，则不绑点情况下默认运行
态图元保持闪烁状态；若为0，则不绑点情况下图元不闪烁；若①②处都配置，则优先级②>①,即绑点>静态；
•背景闪烁颜色：设置图元闪烁颜色，配置完成将与图元背景颜色交替闪烁；运行效果示例：



•边框闪烁变量：可在②处绑定图元边框闪烁点；若①处设置不等于0，则不绑点情况下默认运行态图元边框保持闪烁状态；若为0，则不绑点情况下图元边框不闪烁；若①②处都配置，则优先级②>①,即绑点>静态；

•边框闪烁颜色：设置图元边框闪烁颜色，配置完成将与图元编辑态边框色交替闪烁；运行效果示例：



5.8 闪烁速度：图元闪烁的速度

5.9 画面图层使用

图层复制

•制规则，与【新建】元素规则一致：

◦新画面配置了图层，具体如下：

▪新画面中【最新编辑过】的图层，则加入【最新编辑过】图层

▪新画面没有【最新编辑过】的图层，则加到【最后】一个图层

◦新画面没配置过图层：

▪删除元素的layer属性

1. 跨画面复制，不带图层信息，用户可以自定义复制后的图层。
2. 由于不带图层信息，当复制的元素很多，且图层信息复杂时，不能复用原来的图层结构，只能放在新画面的【某一个】图层上。

图层相关函数

```
sys.setLayerVisible(layername,visible);
```

【参数】

layername图层名称String类型

visible可见性，布尔类型（true/false）

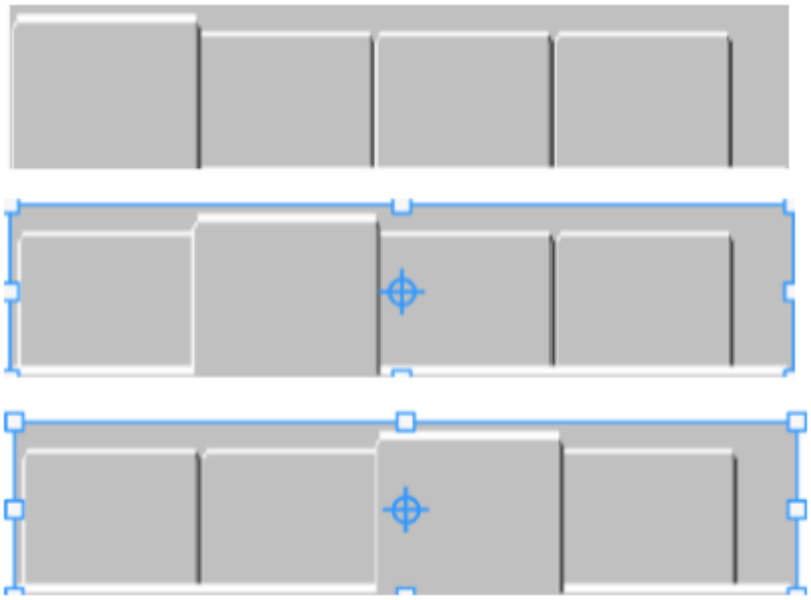
•**功能**：设置页面中该图层名称（layername）对应的图层的可见性（visible）

•**示例**：sys.SetLayerVisible("图层1",false);//图层一，不可见

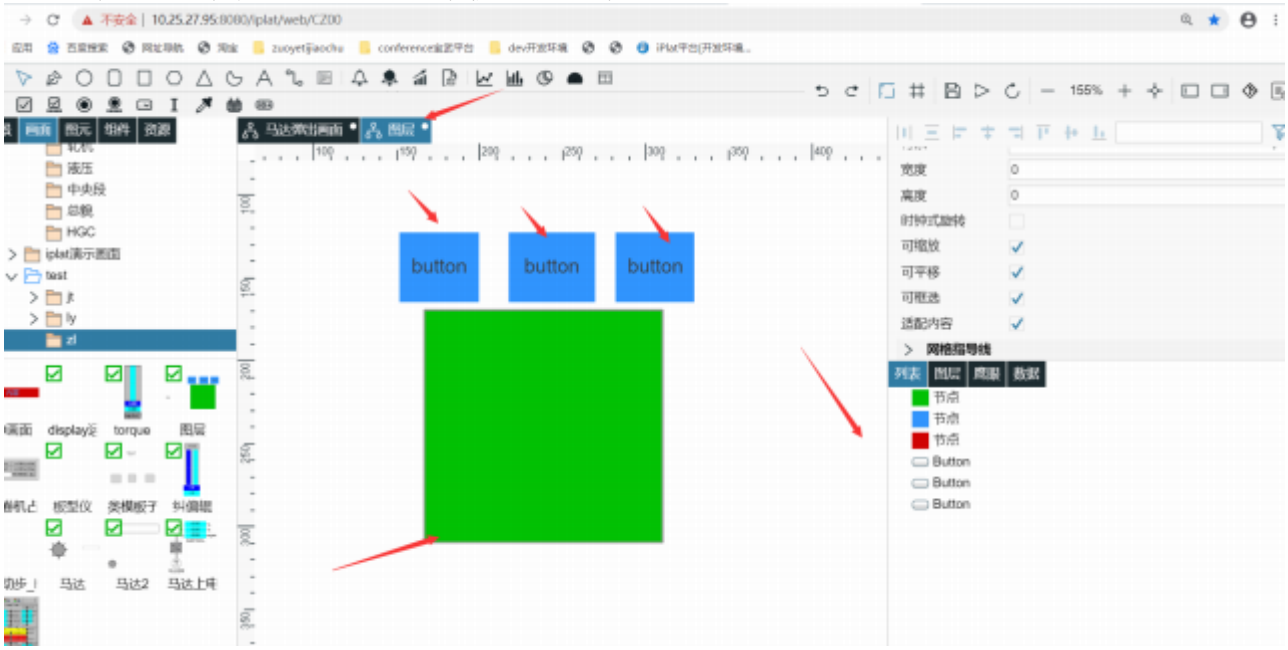
图层使用（页面切换）

功能说明：通过画面按钮触发不同的图层显示, 切换到不同页面时，所点的按钮会相应的放大。

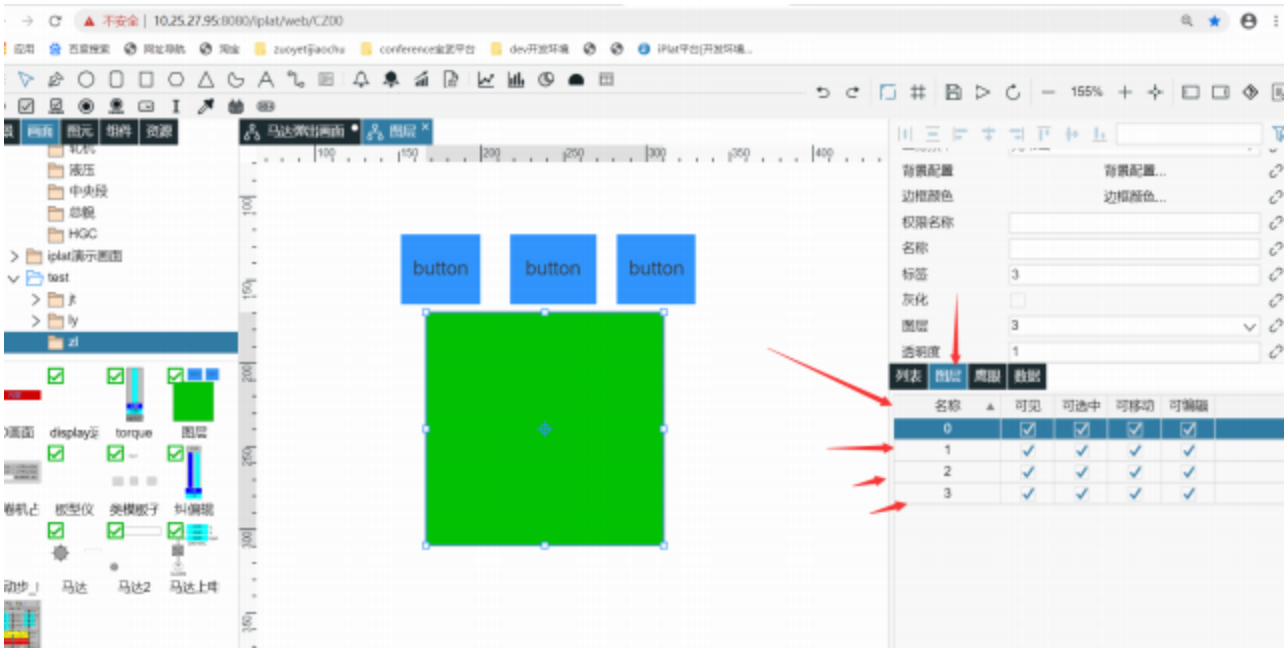
1. 制作三张按钮框，用以展示三个按钮的切换效果三张图片的形状如下图所示：



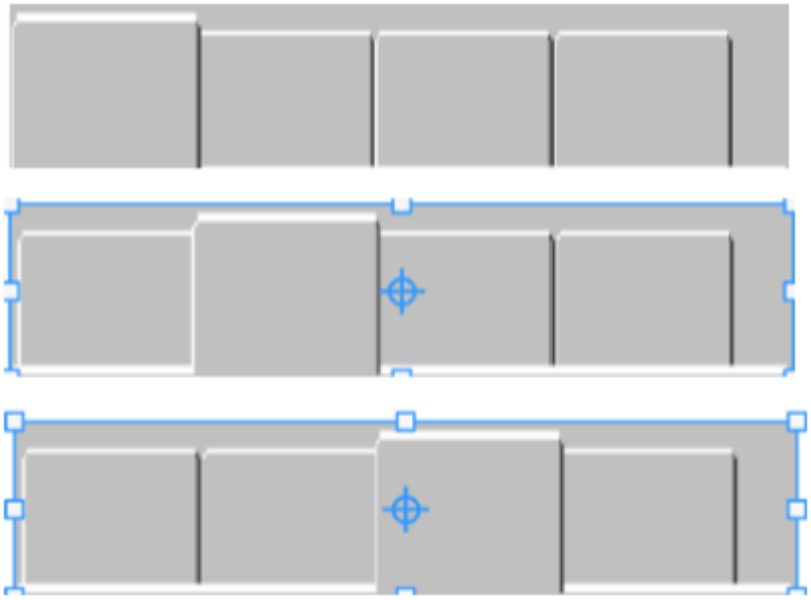
2. 新建一个名为图层的画面，放三个按钮和三个矩形在画面中。



3. 给三个矩形设置不同的背景颜色和不同的图层，图层分别为3，2，1。这三个不同的图层与步骤一的三个按钮框相对应且图层编号一致。给三个按钮设置的图层统一为0。图层在以下tab页添加，2、3图层可以设置为不可见。



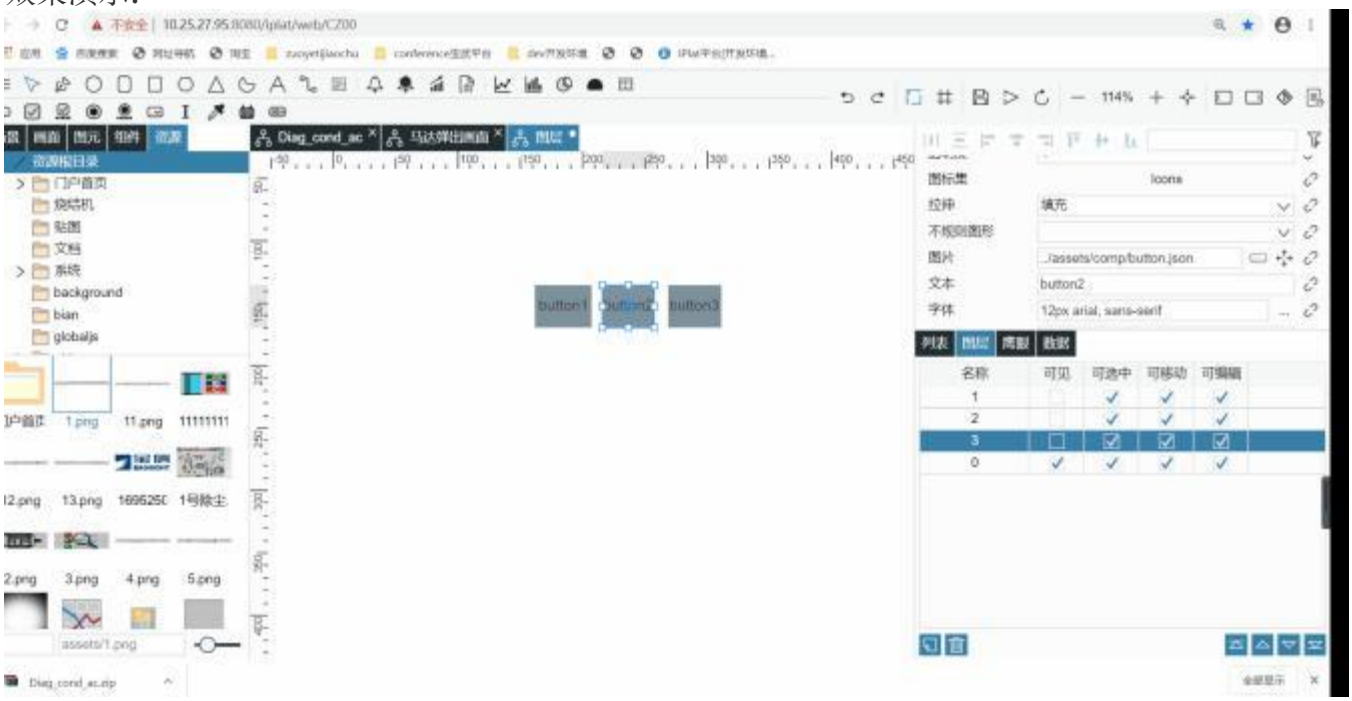
4. 制作三张按钮框，用以展示三个按钮的切换效果三张图片的形状如下图所示：



画面代码：

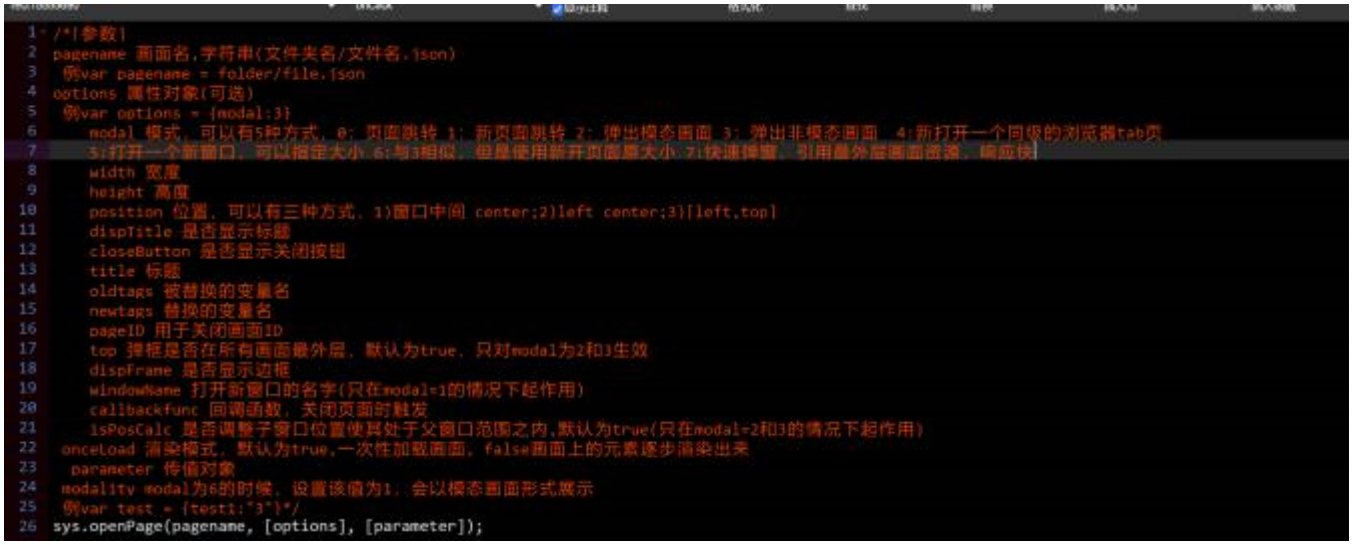
```
Button1_onClick = function(event, data, view){
  sys.SetLayerVisible(1, true);
  sys.SetLayerVisible(2, false);
  sys.SetLayerVisible(3, false);
};

Button2_onClick = function(event, data, view){
  sys.SetLayerVisible(1, false);
  sys.SetLayerVisible(2, true);
  sys.SetLayerVisible(3, false);
};

Button3_onClick = function(event, data, view){
  sys.SetLayerVisible(1, false);
  sys.SetLayerVisible(2, false);
  sys.SetLayerVisible(3, true);
};
```

效果演示：



5.10 画面跳转与弹窗

弹窗函数参数释义



```
/*[参数]
pagename 画面名,字符串(文件夹名/文件名.json)
 例var pagename = folder/file.json
options 属性对象(可选)
 例var options = {modal:3}
    modal 模式，可以有5种方式，0：页面跳转 1：新页面跳转 2：弹出模态画面 3：弹出非模态画面 4:新打开一个同级的浏览器tab页
    5:打开一个新窗口，可以指定大小 6:与3相似，但是使用新开页面最大小 7:快速弹窗，引用最外层画面资源，响应快
    width 宽度
    height 高度
    position 位置，可以有三种方式，1)窗口中间 center;2)left center;3)[left,top]
    dispTitle 是否显示标题
    closeButton 是否显示关闭按钮
    title 标题
    oldtags 被替换的变量名
    newtags 替换的变量名
    pageID 用于关闭画面ID
    top 弹框是否在所有画面最外层，默认为true，只对modal为2和3生效
    dispFrame 是否显示边框
    windowName 打开新窗口的名字(只在modal=1的情况下起作用)
    callbackfunc 回调函数，关闭页面时触发
    isPosCalc 是否调整子窗口位置使其处于父窗口范围之内,默认为true(只在modal=2和3的情况下起作用)
 onceLoad 渲染模式，默认为true,一次性加载画面，false画面上的元素逐步渲染出来
  parameter 传值对象
 modality modal为6的时候，设置该值为1，会以模态画面形式展示
 例var test = {test1:"3"}*/
sys.openPage(pagename, [options], [parameter]);
```

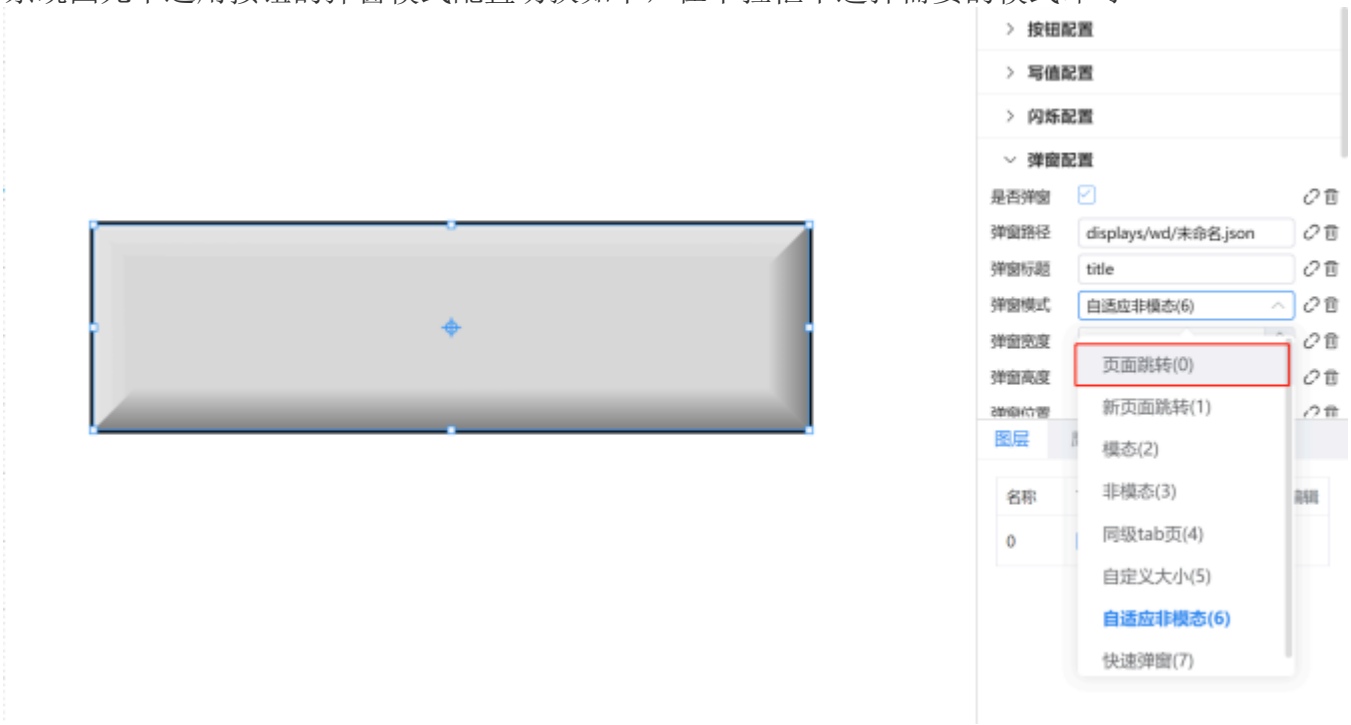
系统图元弹窗

系统图元弹窗属性介绍

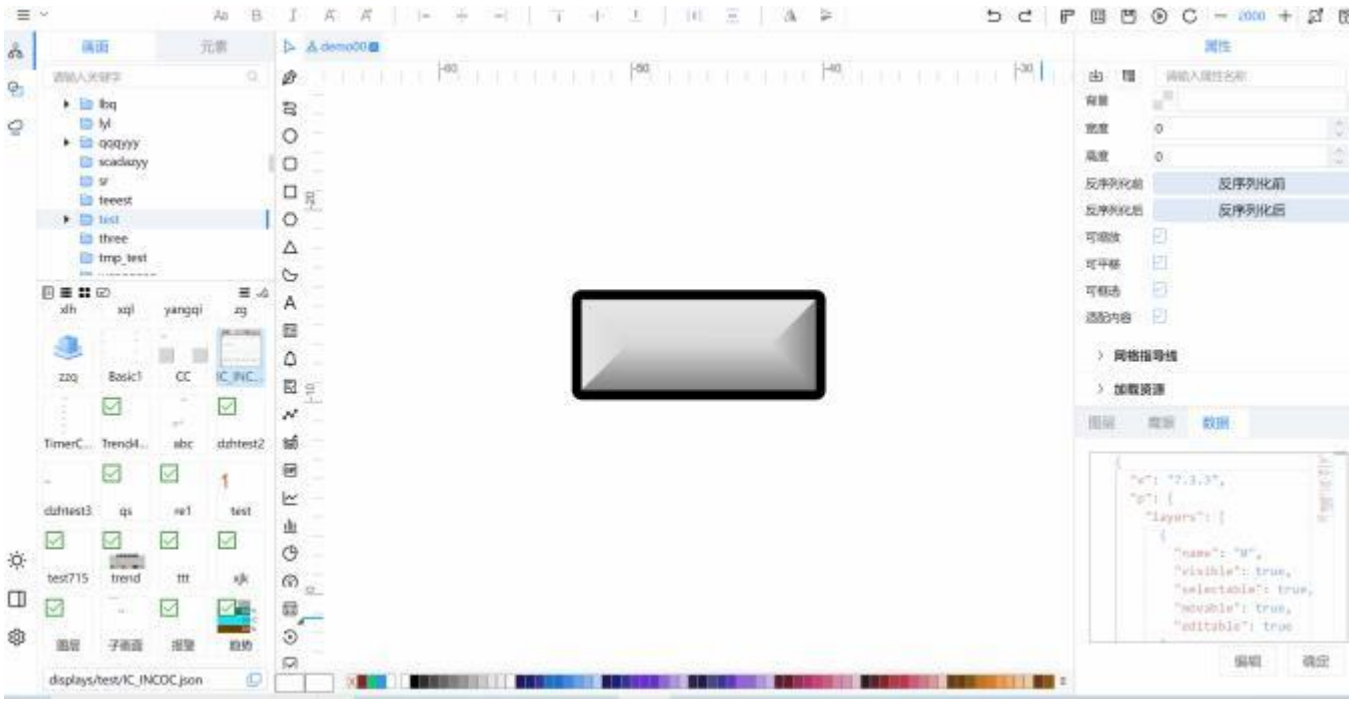
属性名称	属性说明
是否弹窗	勾选，则会出现弹窗
弹窗路径	配置弹窗的路径
弹窗标题	配置弹窗的标题
弹窗模式	配置弹窗的模式，一共包含其中模式，具体实例见后文
弹窗宽度	配置弹窗宽度
弹窗高度	配置弹窗高度

弹窗位置	配置弹窗位置
弹窗ID	配置弹窗ID
原变量	配置弹窗需要被替换的变量
替换的变量	配置弹窗替换的变量
弹窗传参	配置弹窗传递的参数
相对top	勾选，则弹窗相对top
显示标题栏	勾选，则弹窗会显示标题栏
显示关闭按钮	勾选，则弹窗会显示关闭按钮
显示边框	勾选，则弹窗会显示边框
关闭回调	配置关闭回调
openChild	配置子页面调用父页面

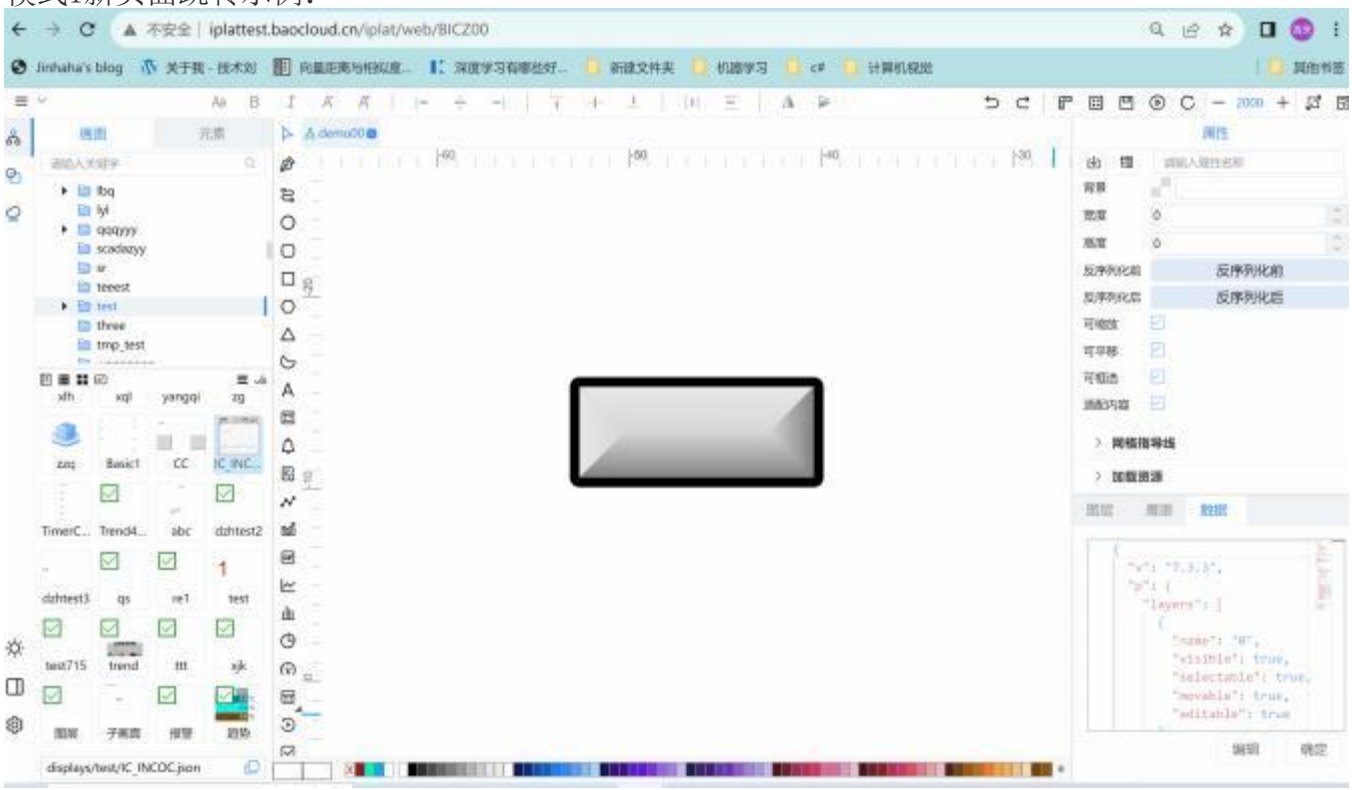
系统图元中通用按钮的弹窗模式配置切换如下，在下拉框中选择需要的模式即可



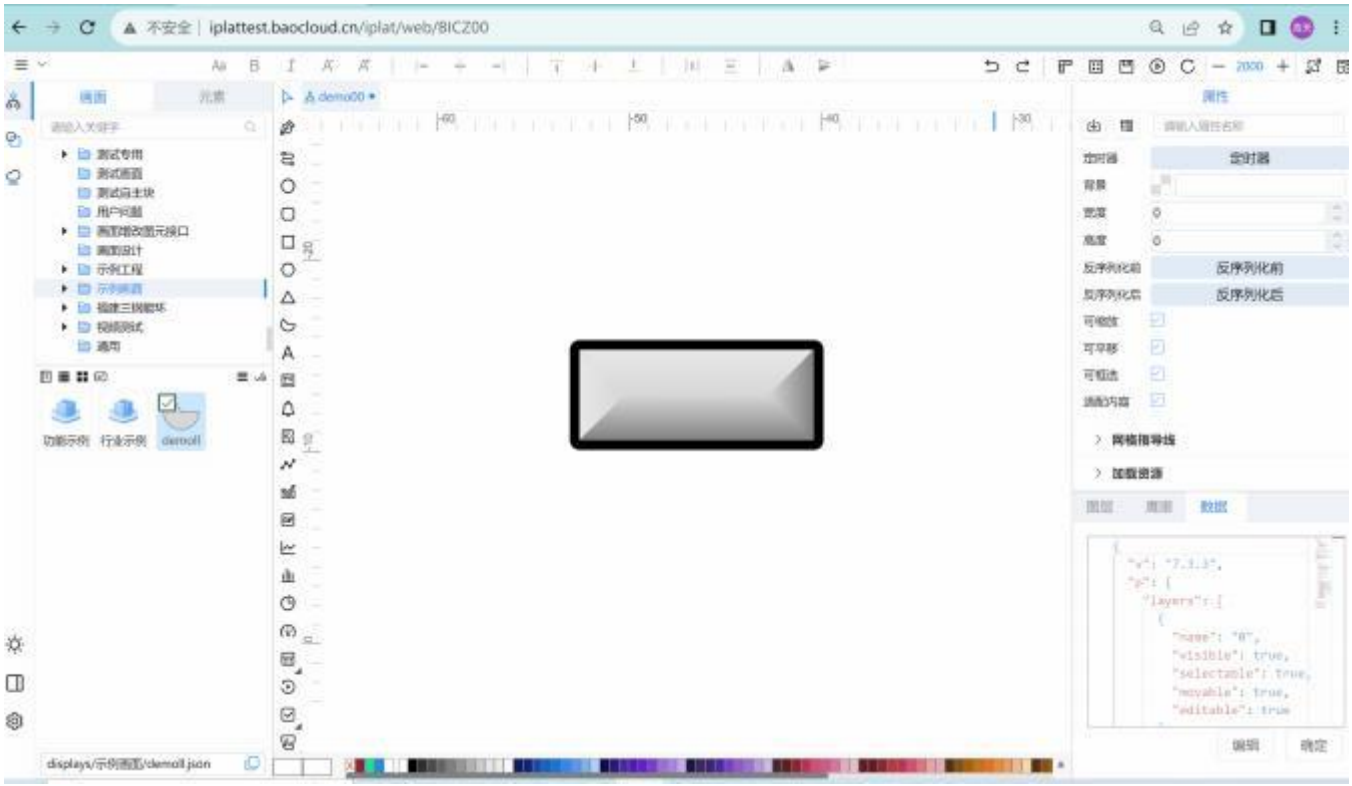
模式0页面跳转示例：



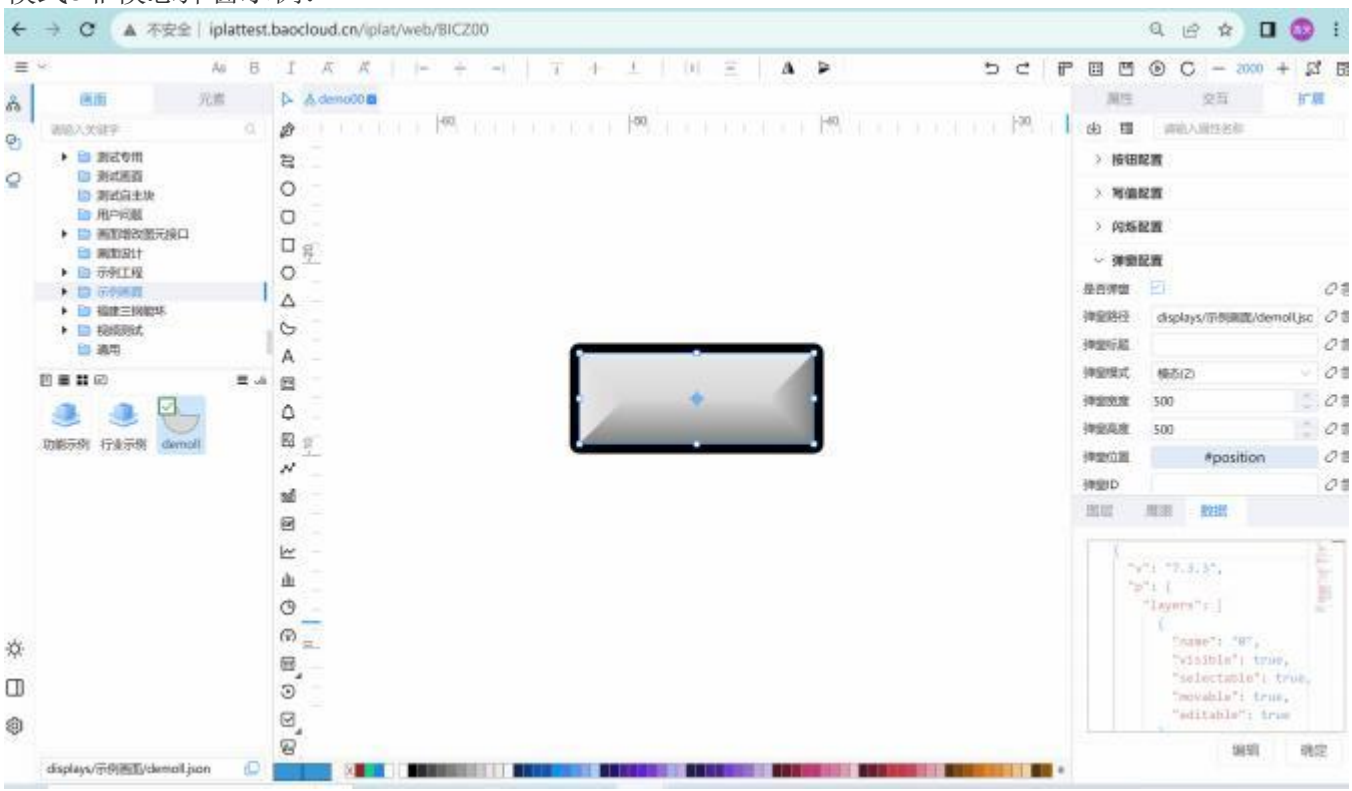
模式1新页面跳转示例：



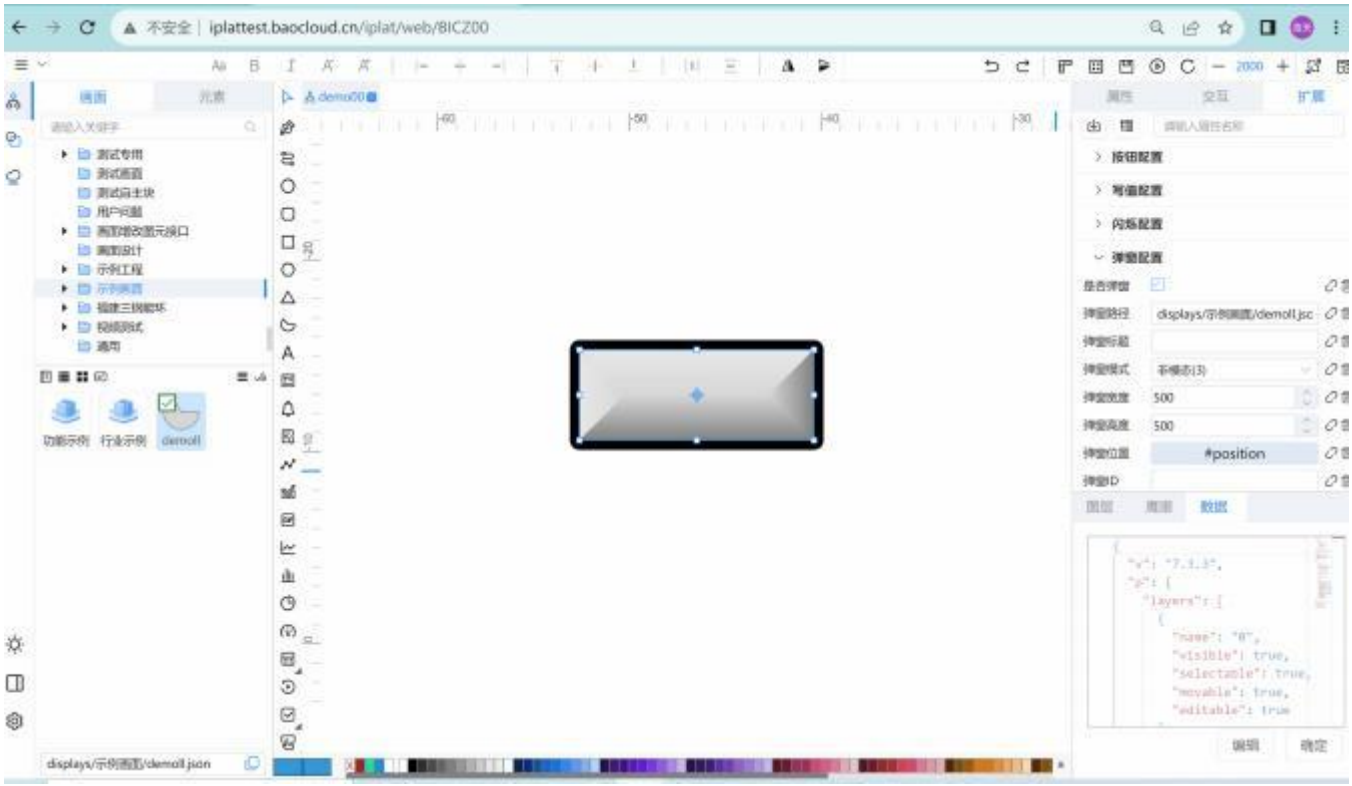
模式2模态弹窗示例：



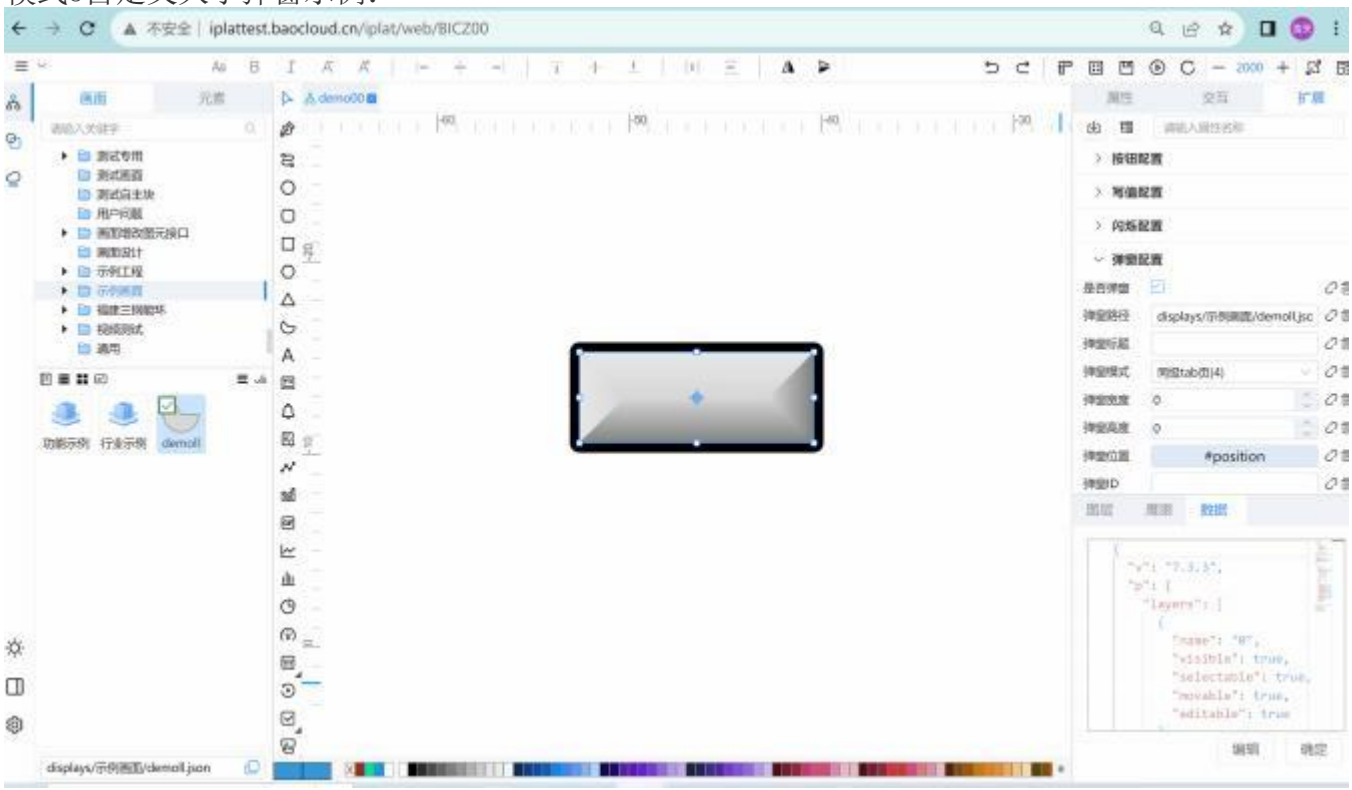
模式3非模态弹窗示例：



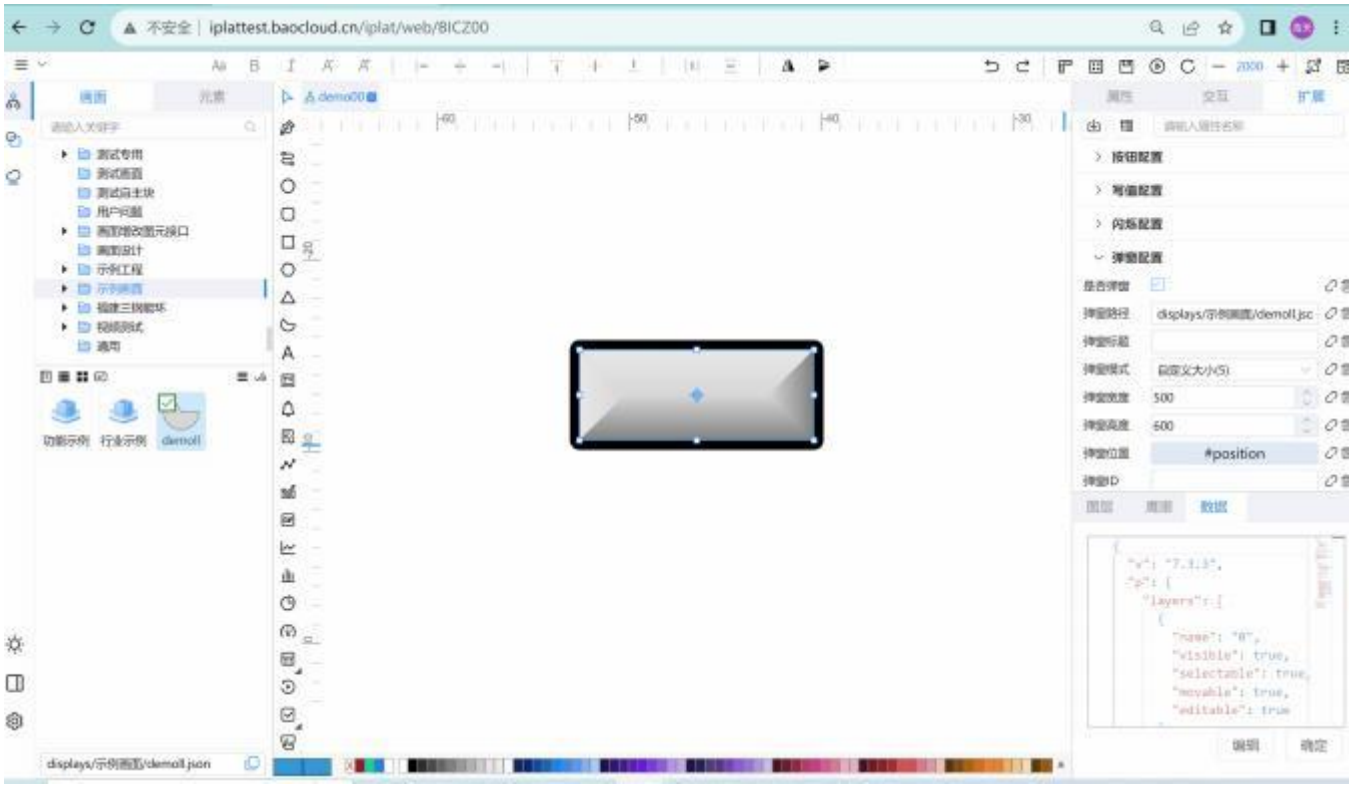
模式4同级tab页弹窗示例：



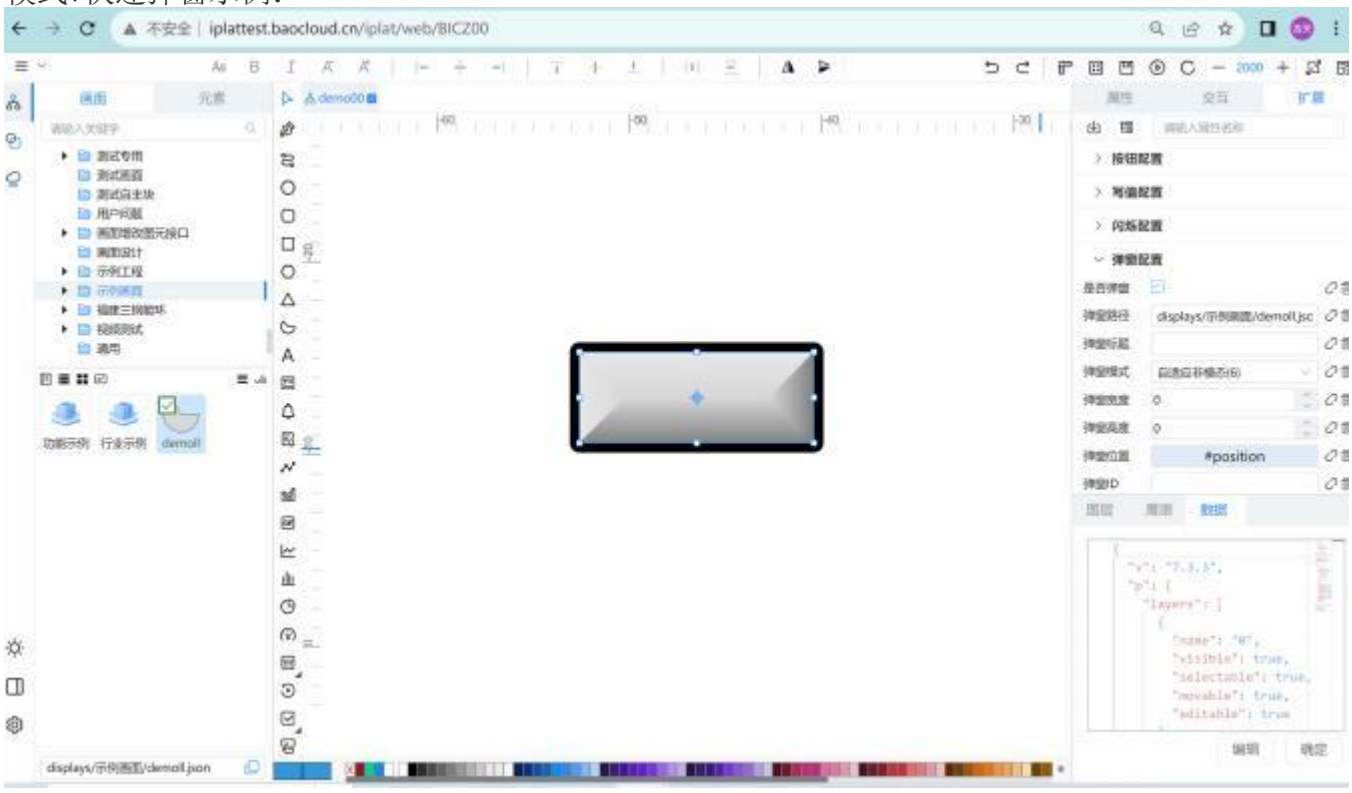
模式5自定义大小弹窗示例：



模式6自适应非模态弹窗示例：



模式7快速弹窗示例：

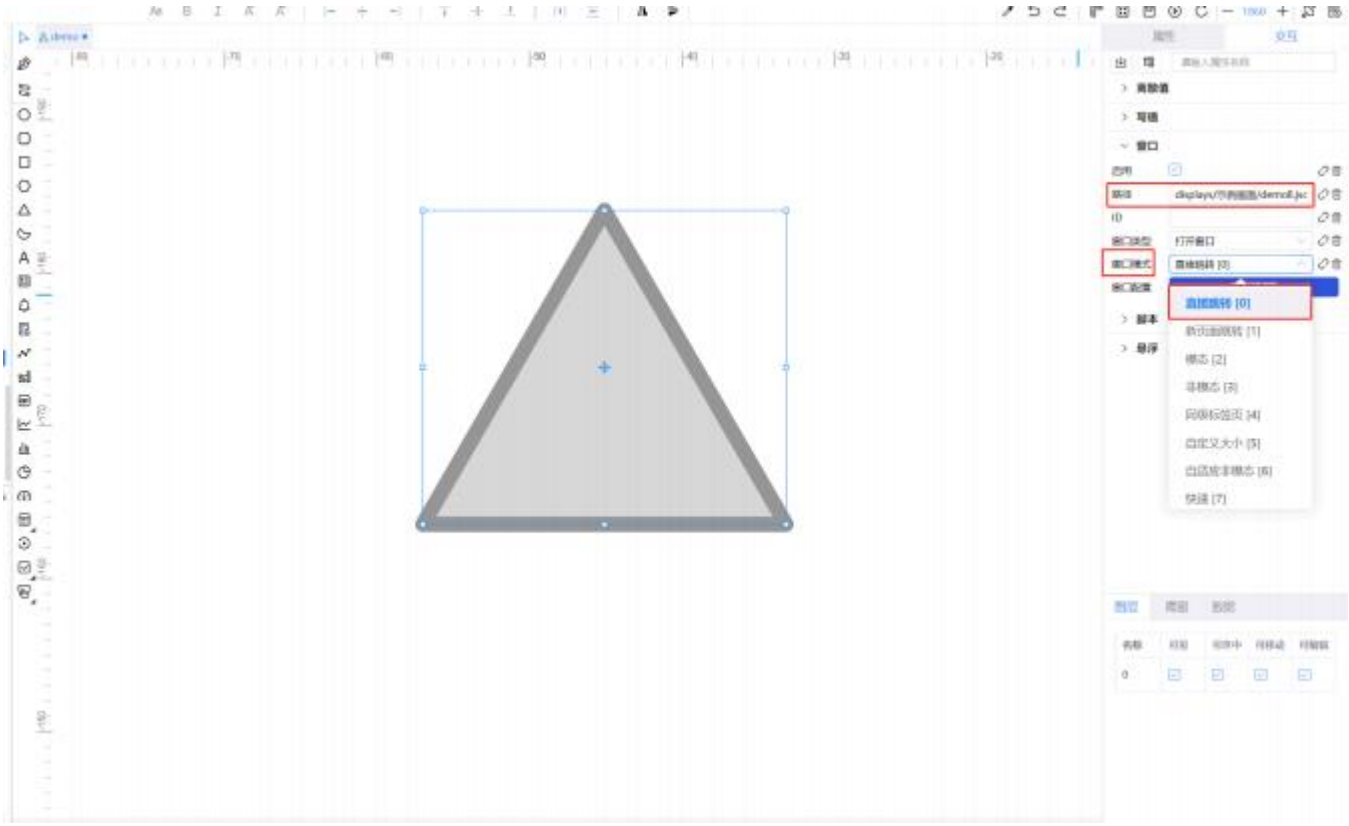


元素弹窗

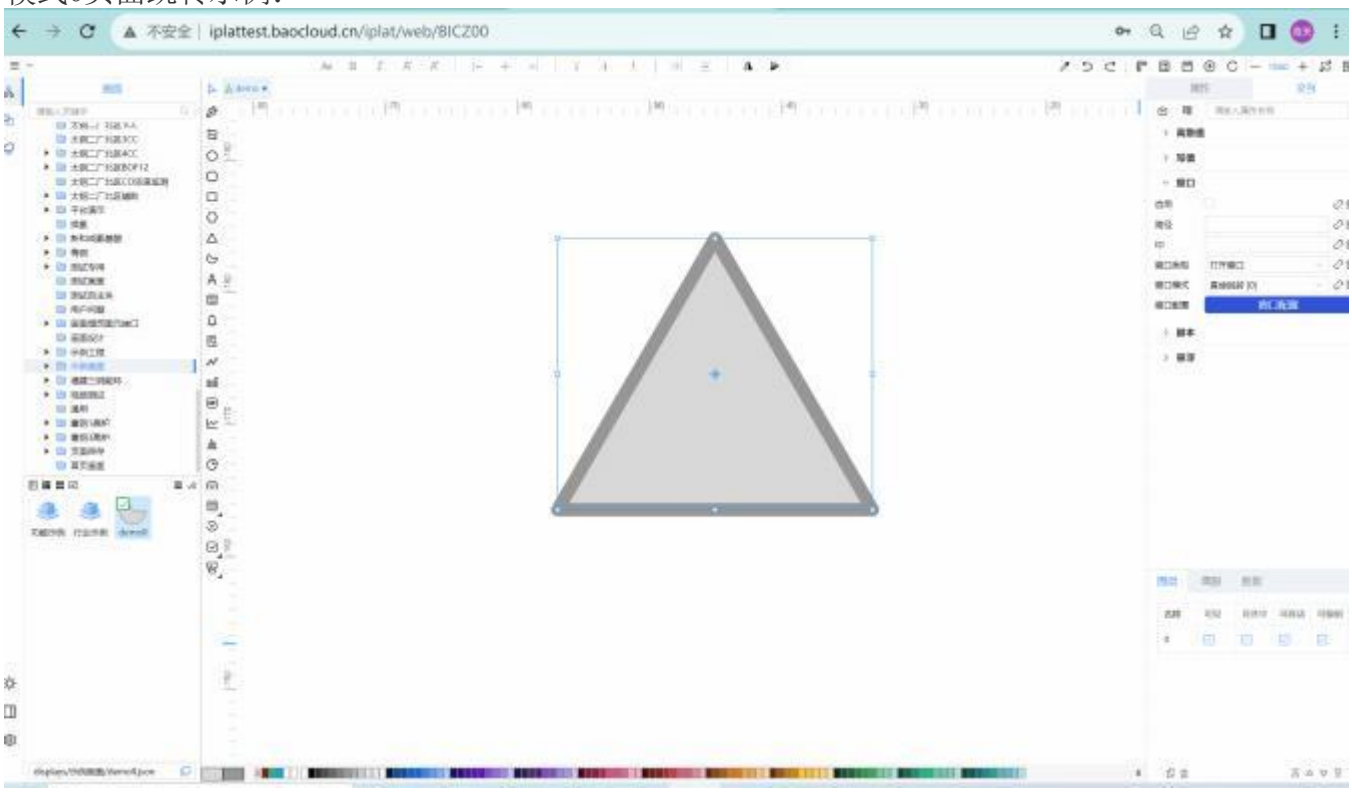
元素弹窗属性介绍

属性名称	
启用	勾选，则
路径	配置弹窗
ID	配置弹窗
窗口类型	选择打开
窗口模式	配置弹窗
窗口配置	配置弹窗

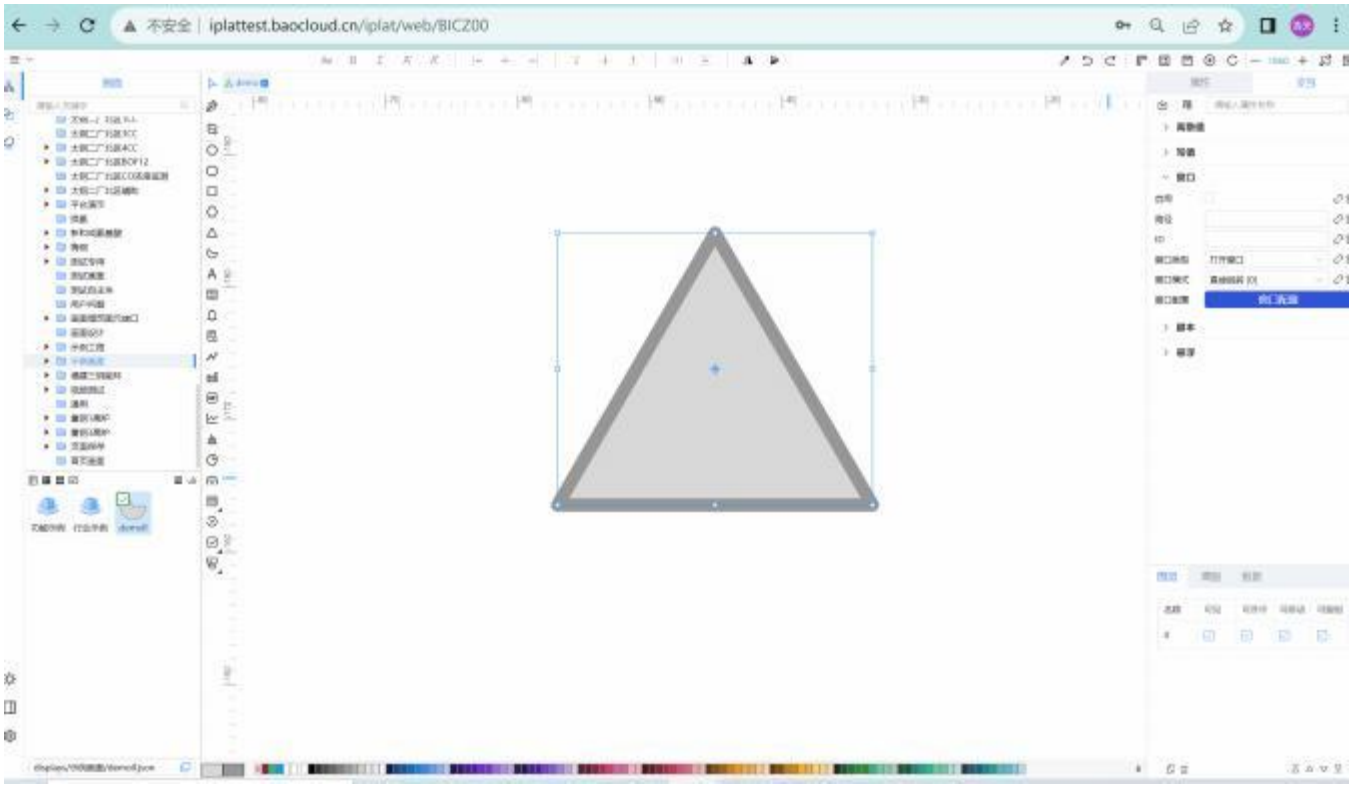
系统图元中通用按钮的弹窗模式配置切换如下，在下拉框中选择需要的模式即可



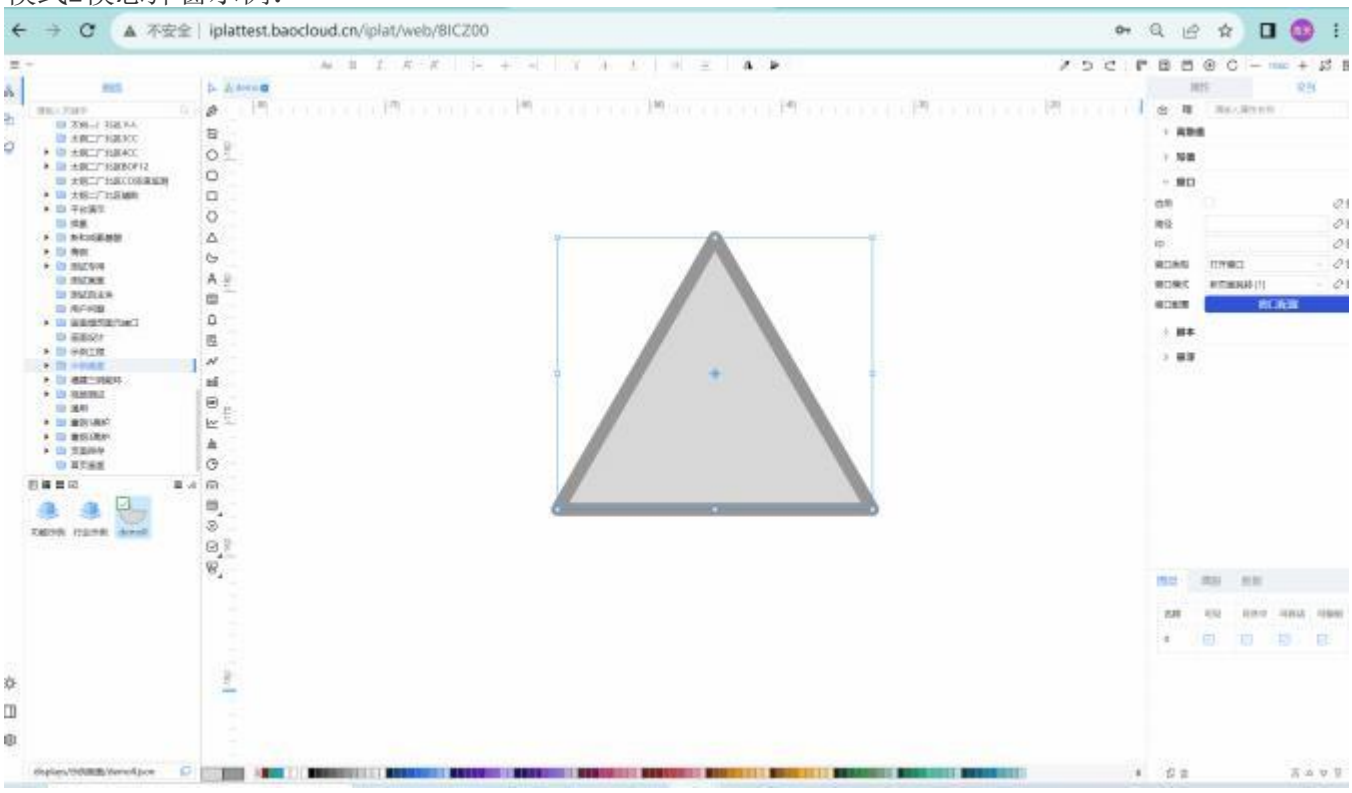
模式0页面跳转示例：



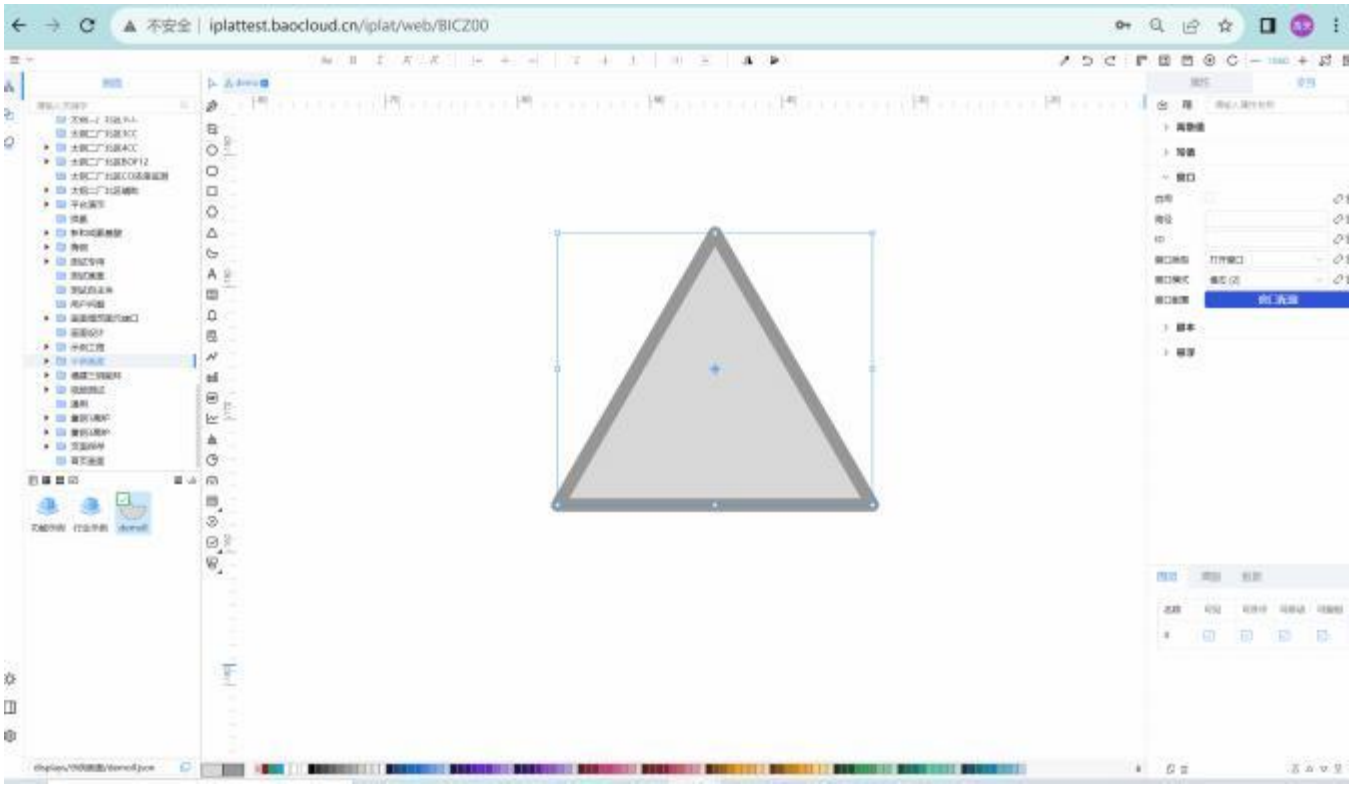
模式1新页面跳转示例：



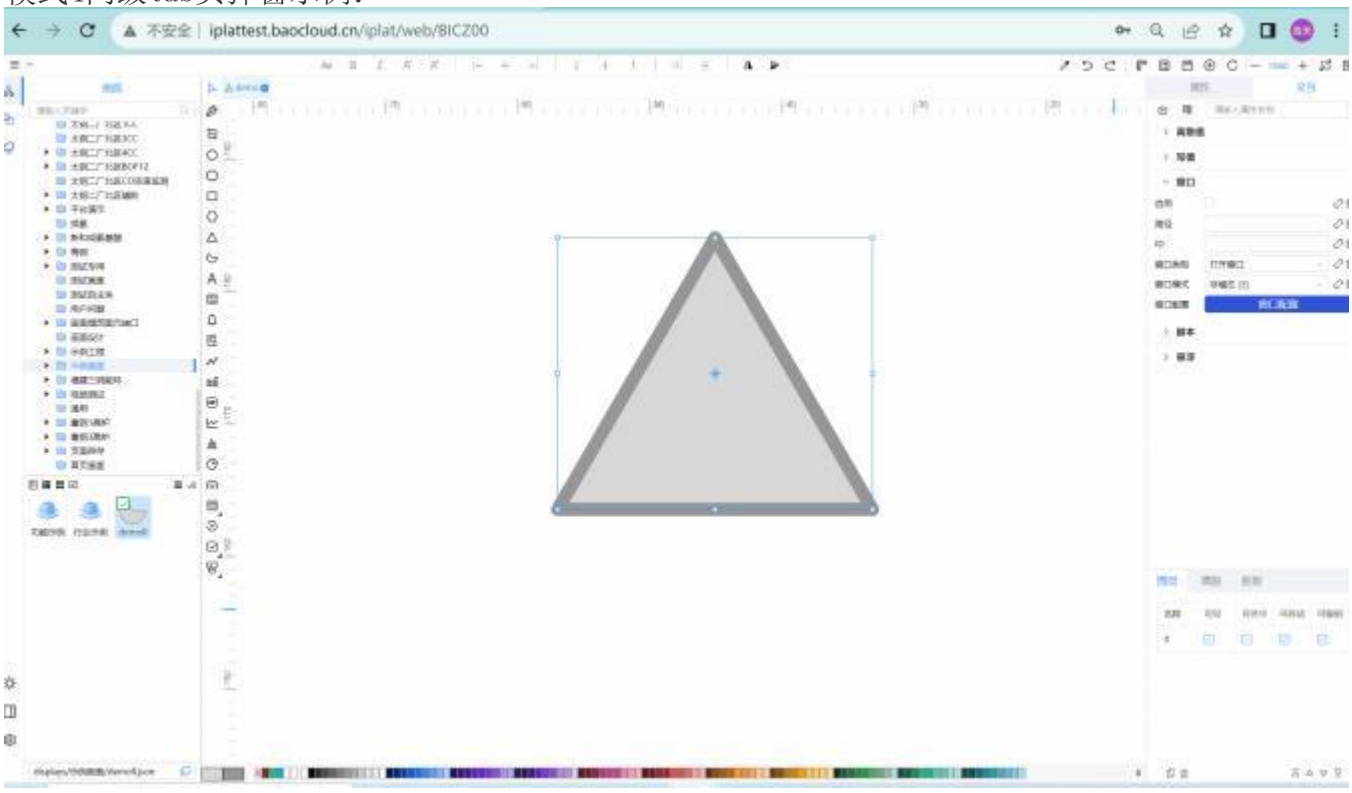
模式2模态弹窗示例：



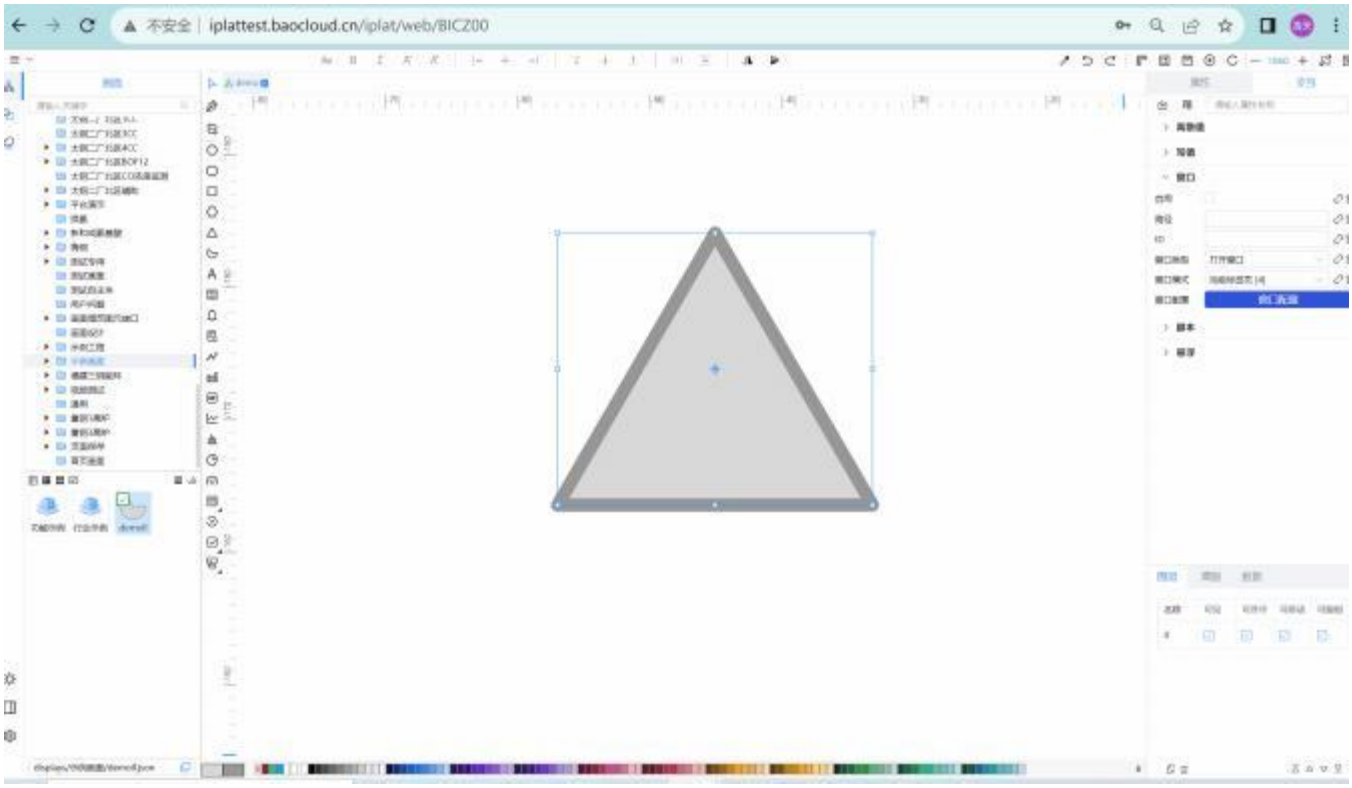
模式3非模态弹窗示例：



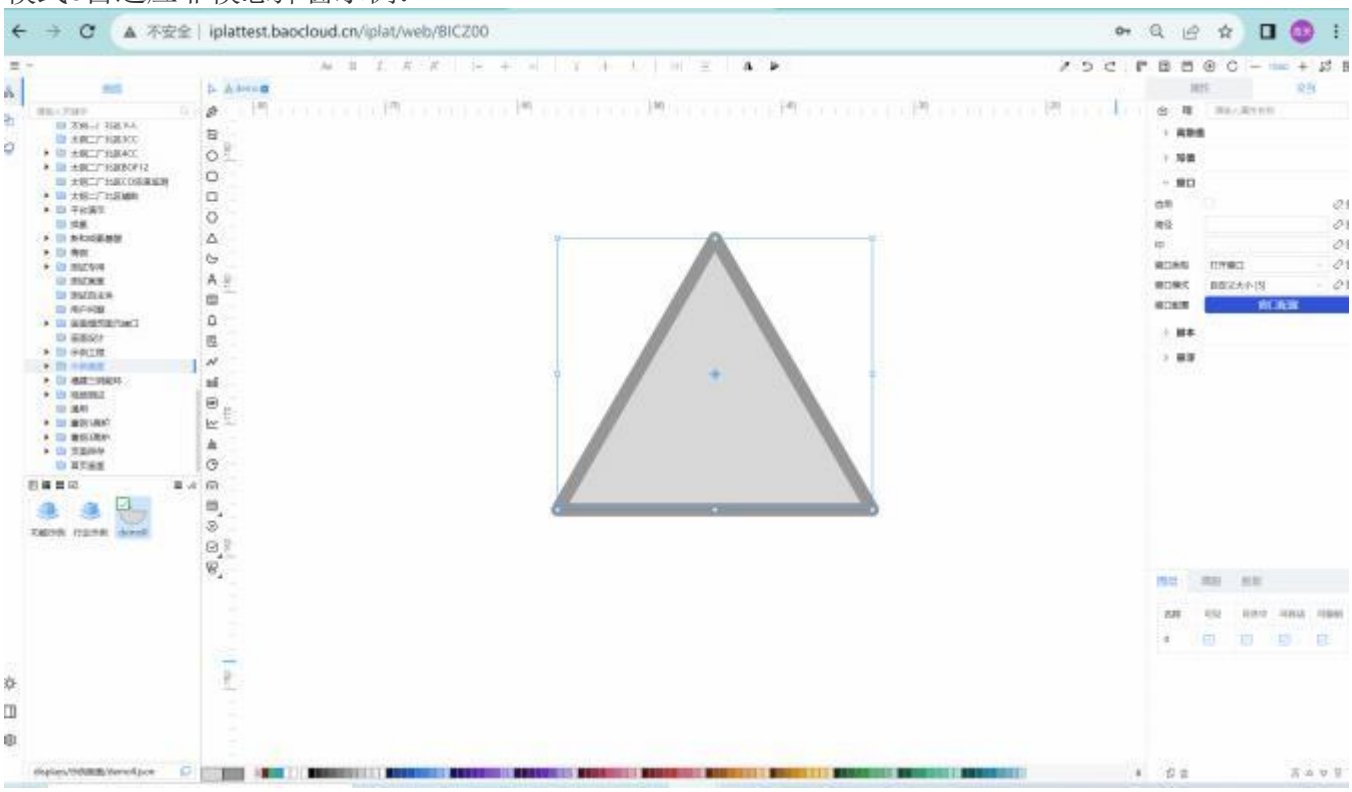
模式4同级tab页弹窗示例：



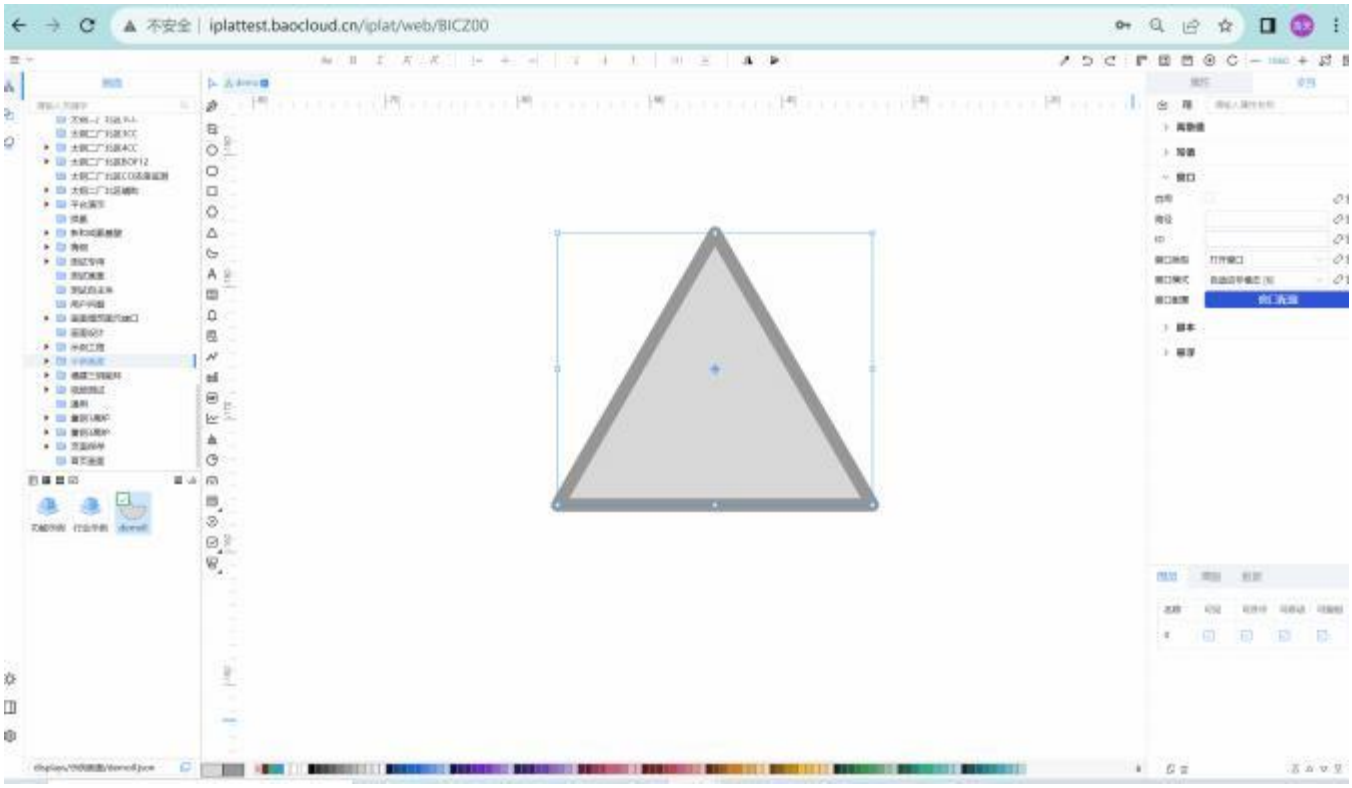
模式5自定义大小弹窗示例：



模式6自适应非模态弹窗示例：



模式7快速弹窗示例：



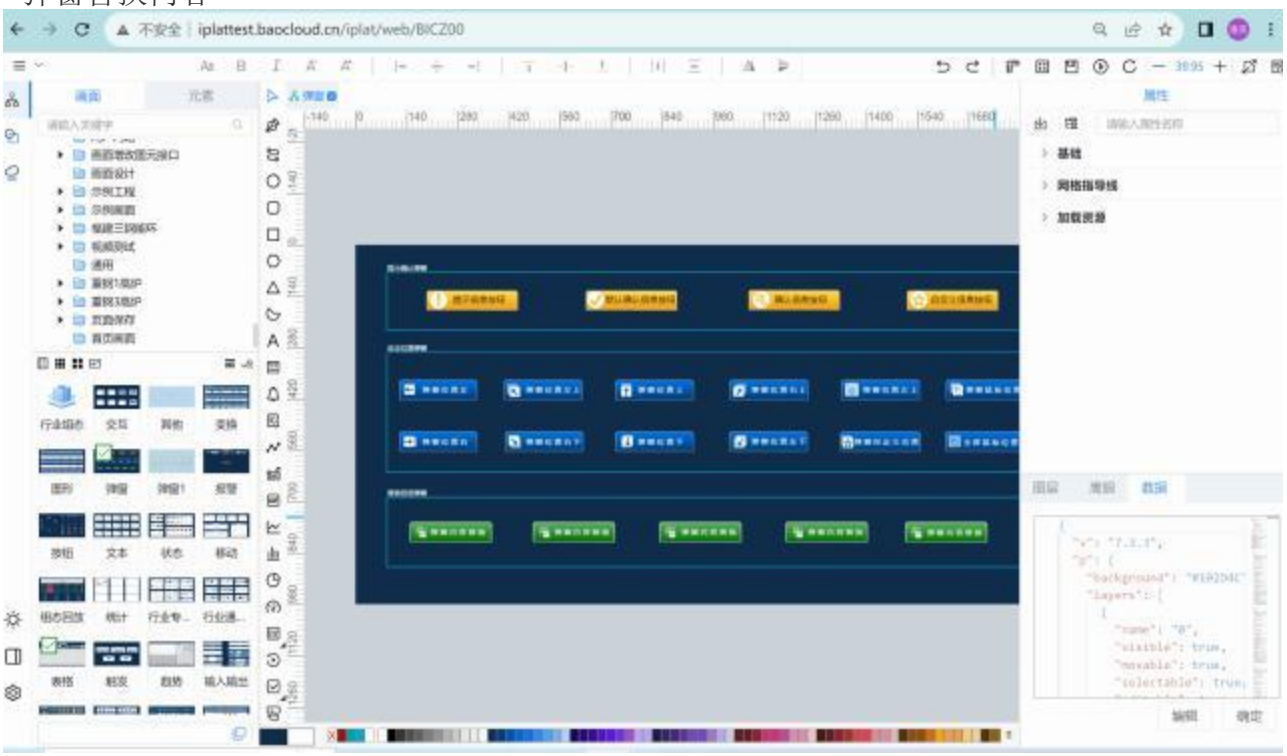
模式8菜单弹窗示例：
该模式配合导航使用。

在导航菜单画面，设置元素弹窗事件，选择模式8并填写路径或画面id，可以在导航中打开相应画面。如果打开的是四级菜单，同时三级菜单能够对应的被高亮。，能够通过上一页，下一页进行前进和后退操作，可缓存的页面有20个。

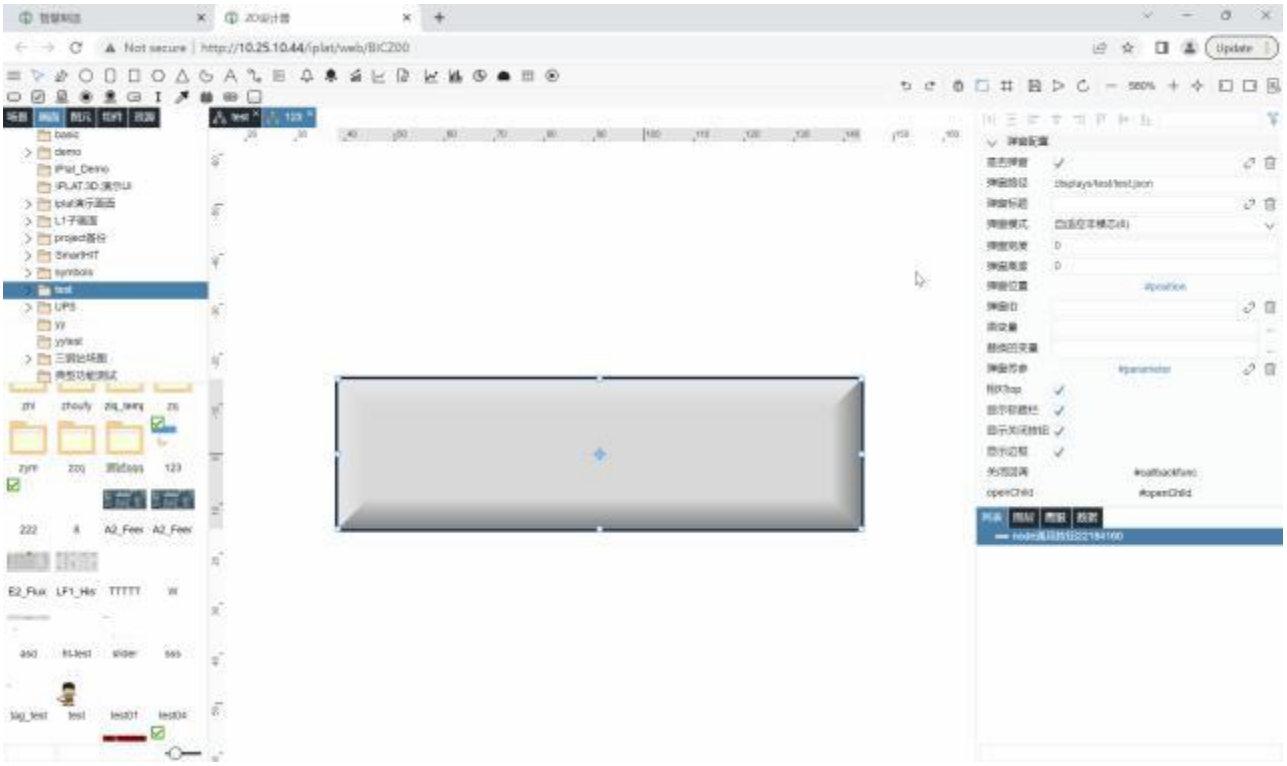
跳转规则：1->2->3->4->5->回退到4->回退到3->点击跳转到6->回退到3->回退到2-回退到1
需要注意的是：跳转到6之后，4和5步骤被删除，不会再走到

典型场景

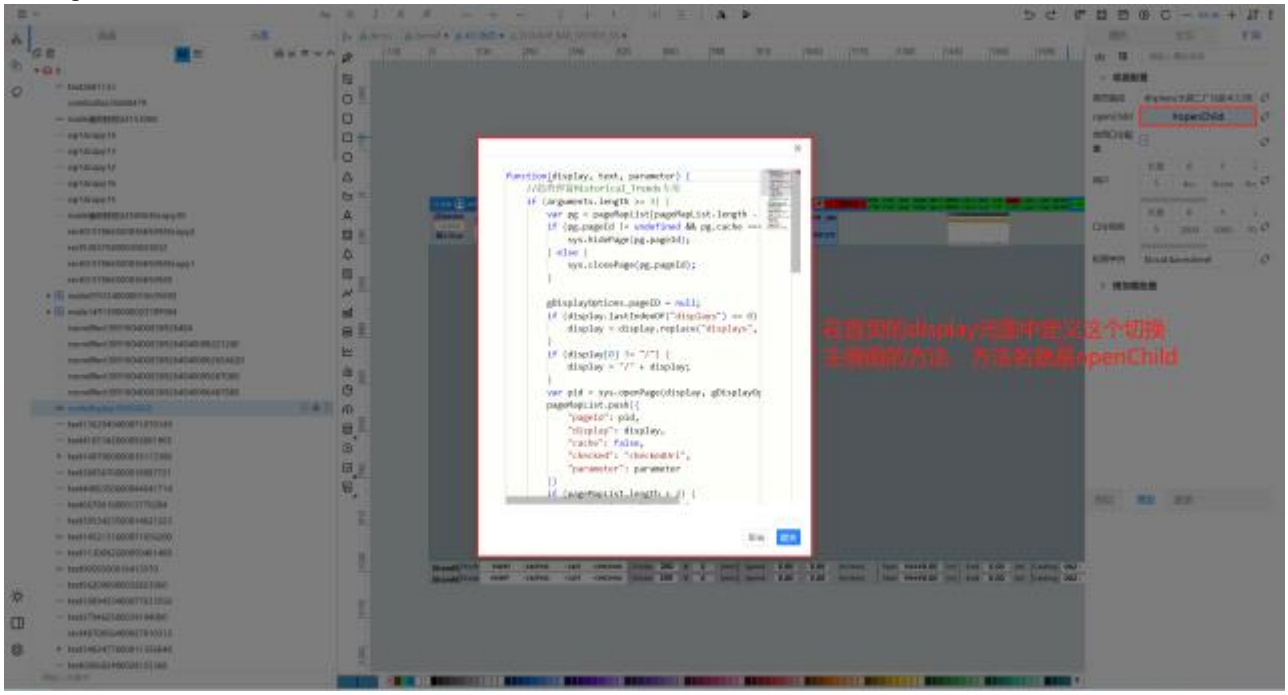
1. 弹窗替换内容

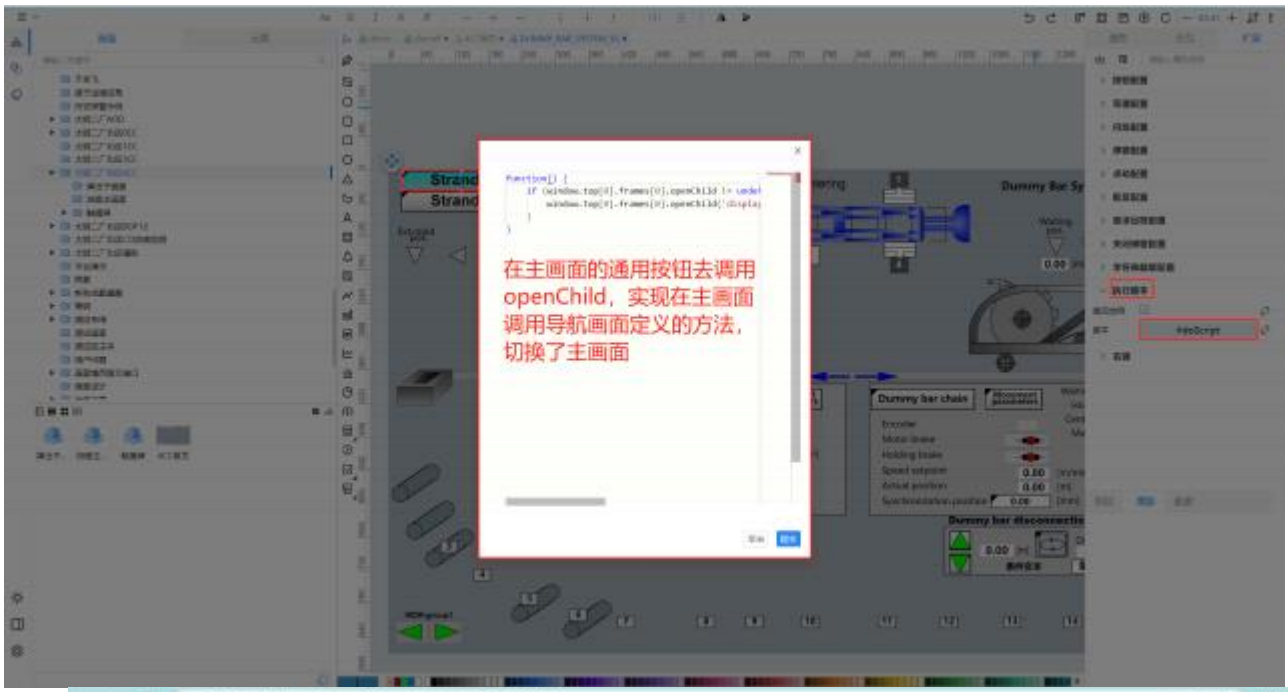


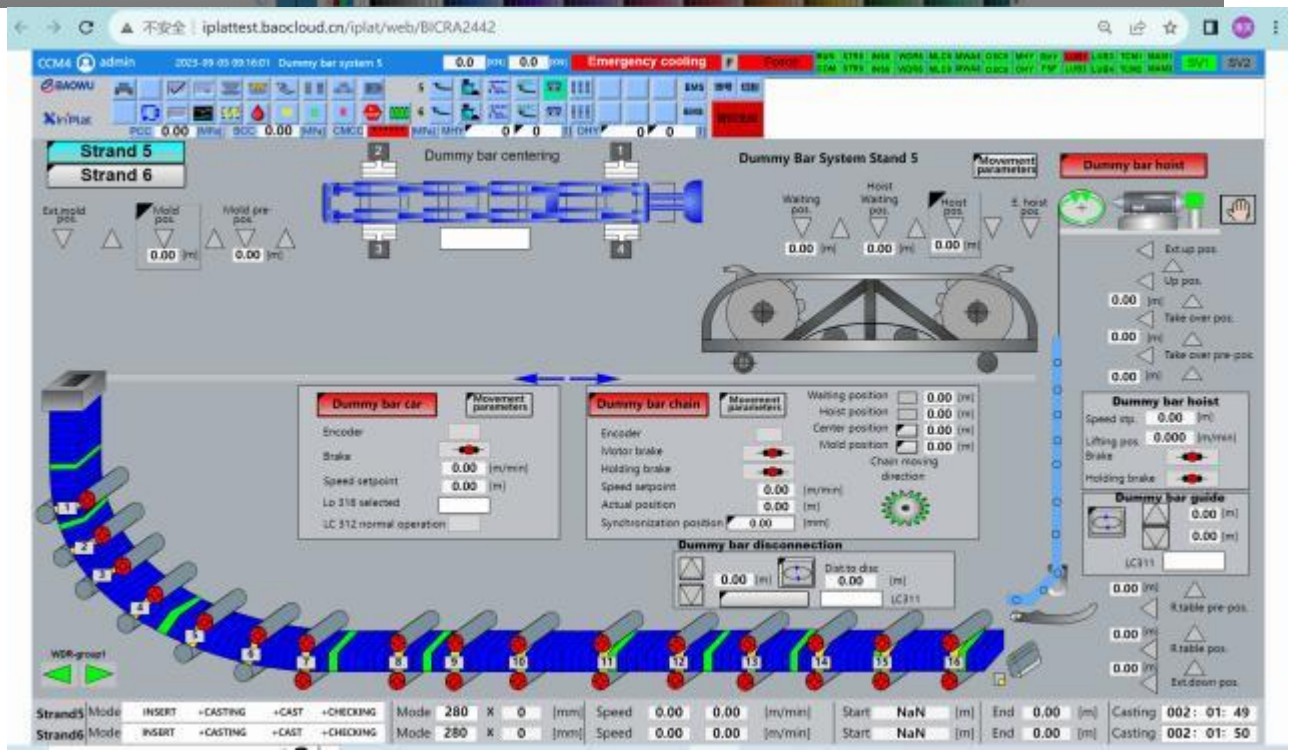
2. 弹窗传值



3. openchild（子页面调用父页面定义的方法）



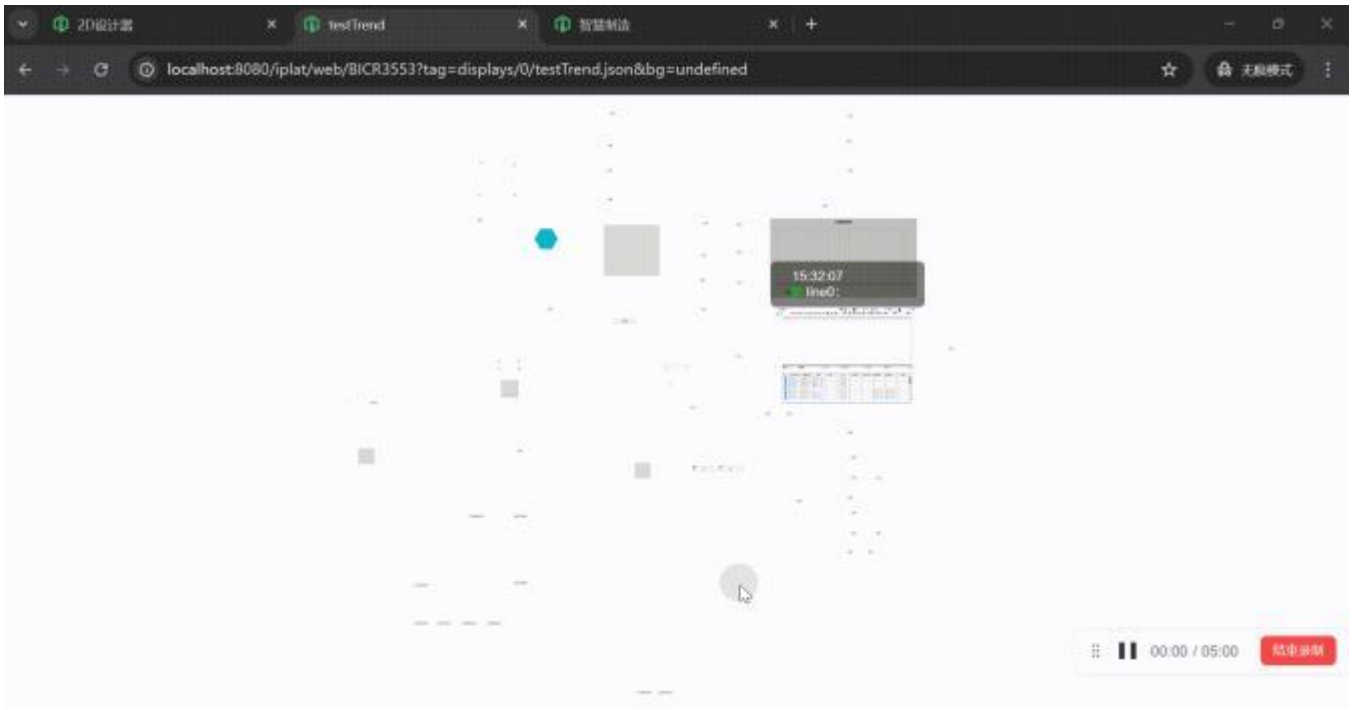




弹窗0和文本趋势窗口

弹窗0导致的当前页面的url变化时，保持文本趋势窗口不关闭，针对新旧文本趋势都生效

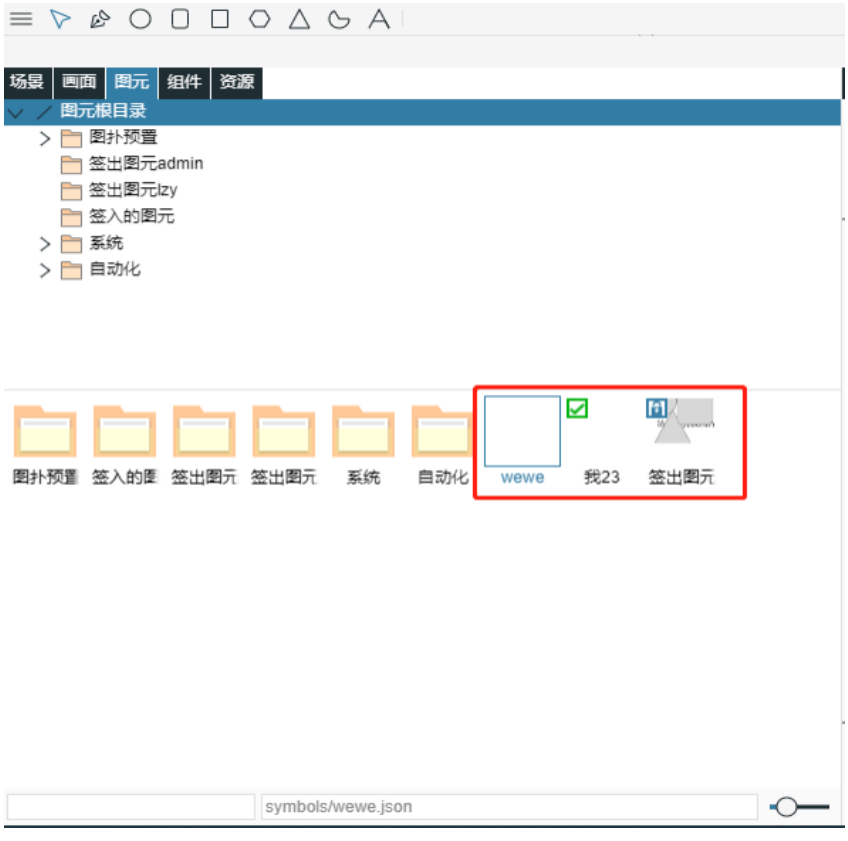
例如，从父页面A通过弹窗0跳转到父页面B时，父页面A打开的文本趋势窗口不会关闭。但是父页面关、刷新或者其它方式导致的url变更还是会关闭文本趋势窗口。



5.11 画面签入签出

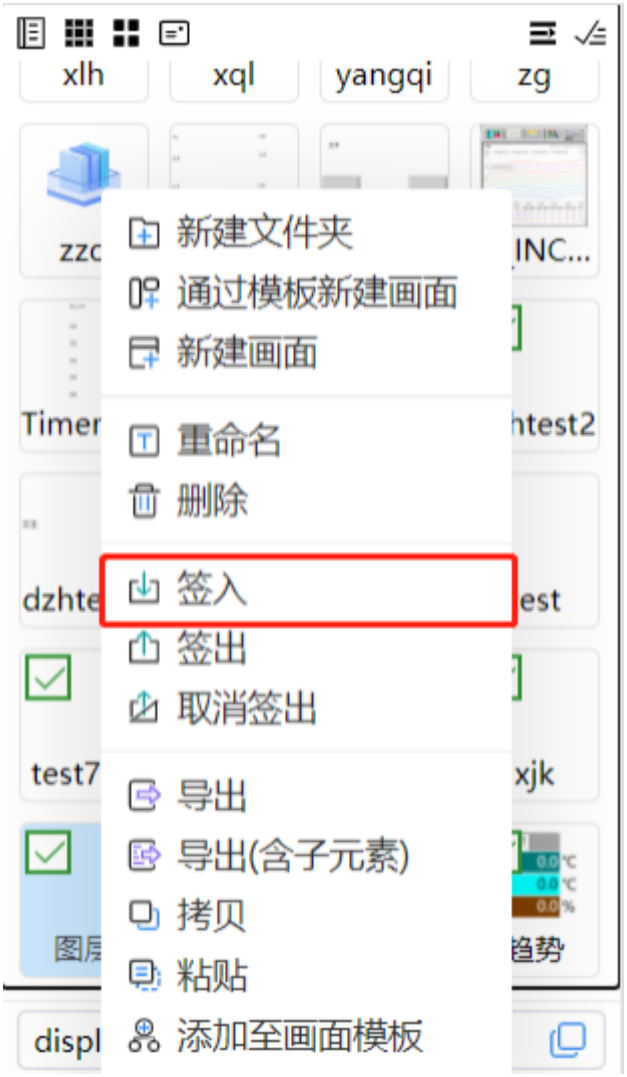
画面属性

- 画面左上方的图标代表图元的三种签入签出状态：
  - 文件左上方为绿色√图标，代表被当前用户签出
  - 文件左上方为蓝色a图标，代表被其他用户签出（或者无画面权限），当前用户不能进行签入操作
  - 文件左上方无图标，代表图元为已签入状态

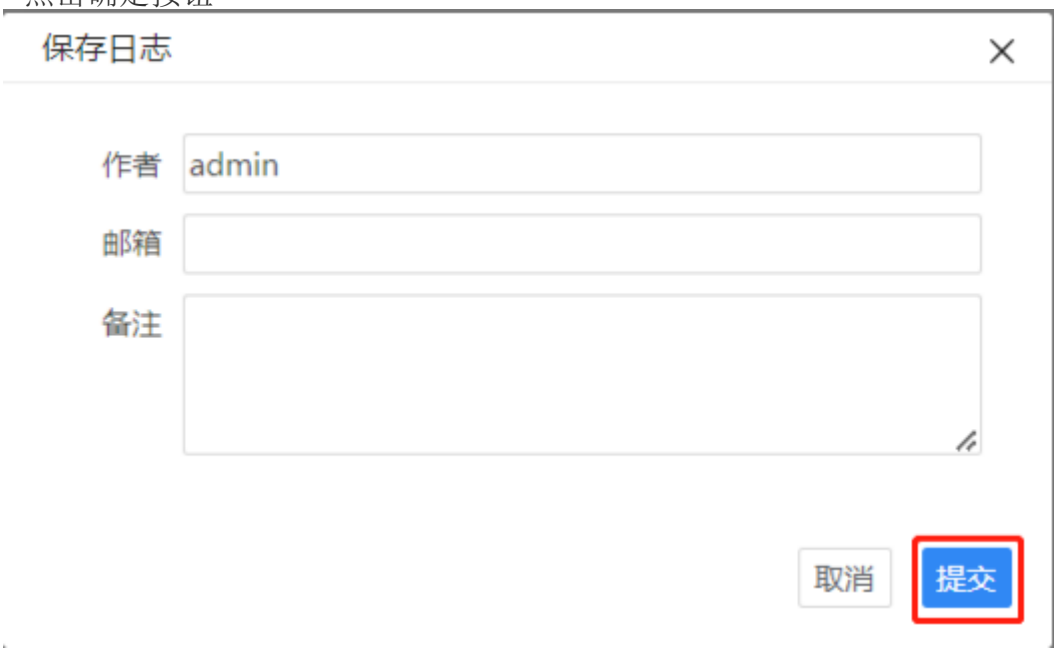


画面签入

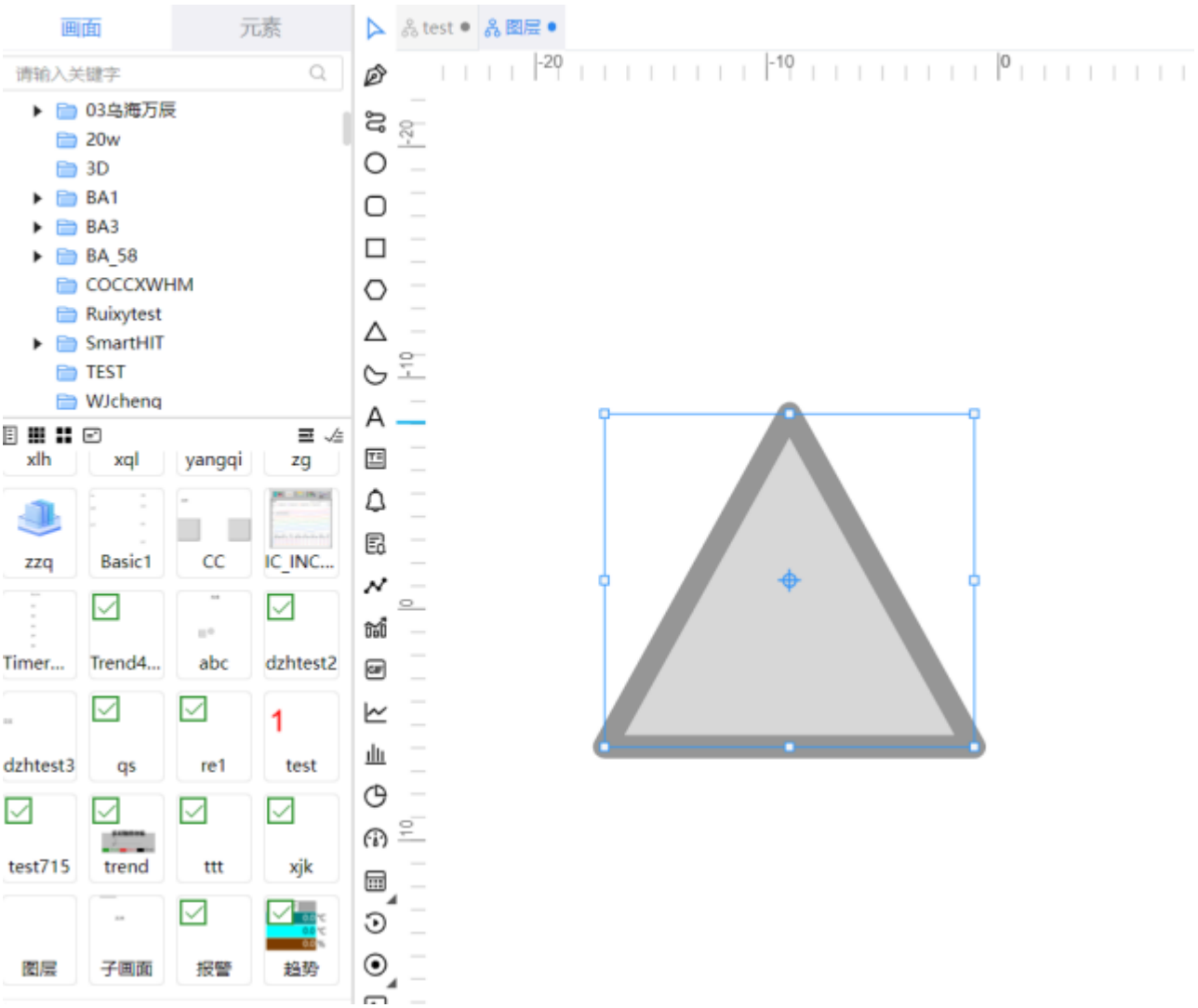
1. 点击右键选择签入



2. 点击确定按钮

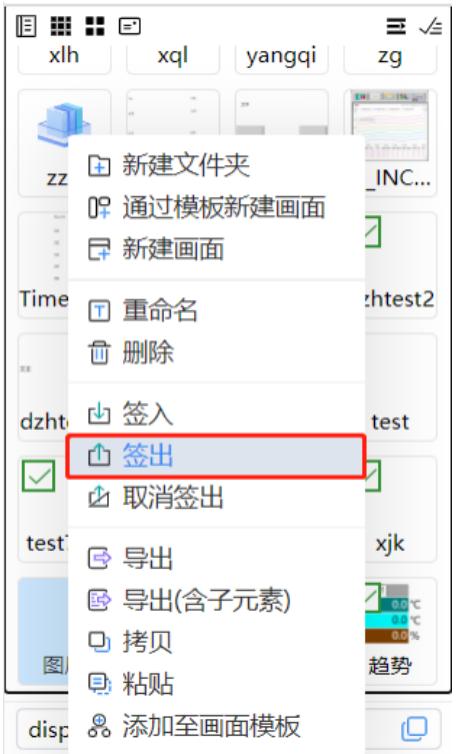


3. 签入成功

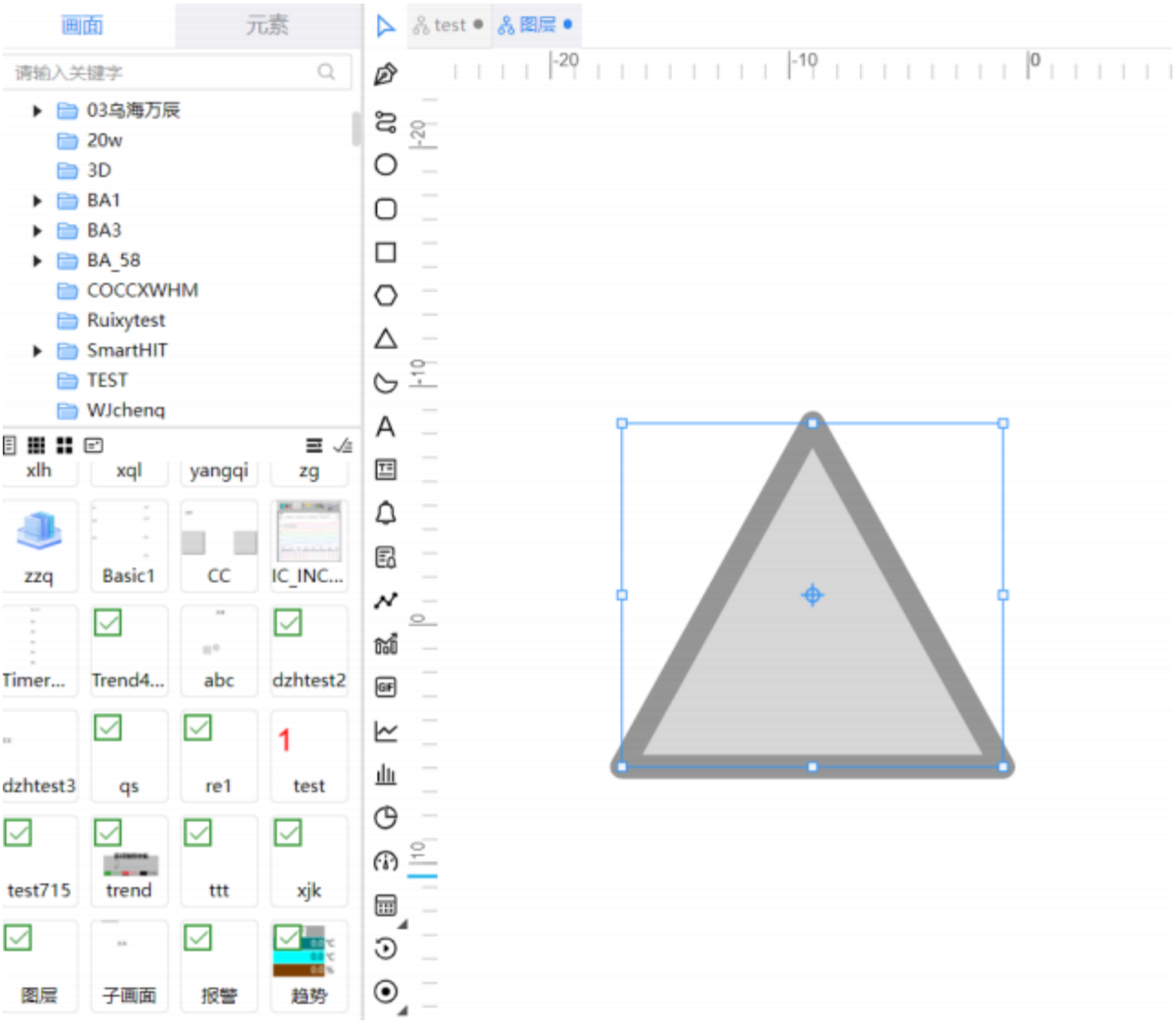


画面签出

1. 点击右键选择签出



2. 签出成功



画面典型操作

1. 打开2D设计器，选择画面Tab页，**画面根目录**下的画面签入签出状态应该为最新状态。
2. **单击**左上方画面文件夹，或者双击左下方**画面文件夹，该文件夹下的画面签入签出状态应该为最新状态。
3. 对于已**被当前用户签出**的画面，点击**签入**，弹出签入对话框进行签入，签入后，画面左上方绿色图标消失。
4. 对于已**被当前用户签出**的画面，点击**签出**，提示"图元已被签出，无法签出，可刷新浏览器获取画面最新状态！"
5. 对于已**被其他用户签出**的图元，点击**签入**，弹出签入对话框进行签入，提示"**画面被他人签出，无法签入！**"(admin对于已**被其他用户签出**的画面，可以**强制签入**)。
6. 对于**已签入**的画面，点击**签出**，签出后，画面左上方出现绿色图标。
7. 只能移动**已签入**的画面，移动画面后到其他文件夹后，该画面状态依然为已签入。
8. 如果文件夹xxx下有**已签出**的画面，**移动该文件夹**，提示"**xxx下存在被签出的画面，无法移动!**"，移动**已签出**的画面"1"，提示"**无法移动画面1，对画面1无权限或该图元被签出！**"。
9. 只能重命名**已签入**的画面，重命名画面后，该画面状态依然为已签入。
10. 重命名**已签出**的画面"1"，提示"**无法重命名画面1，对画面1无权限或该画面被签出！**"。
11. 如果文件夹xxx下有**已签出**的画面，**重命名该文件夹**，提示"**xxx下存在被签出的画面，无法重命名!**"。

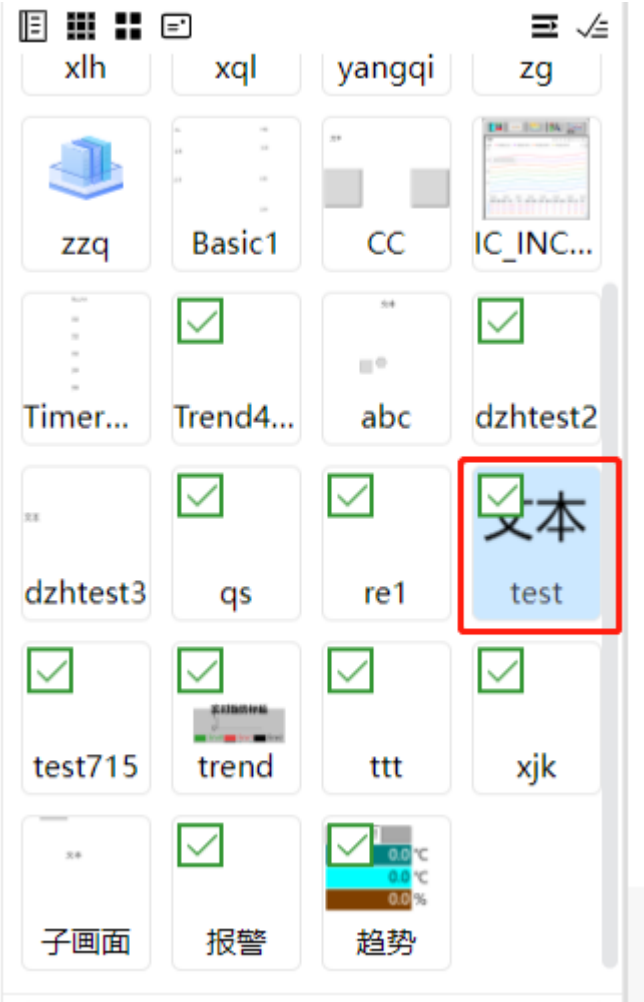
12. 只能删除**已签入**的画面，删除画面后，该图元父文件夹下的画面签入签出状态应该为最新状态。
13. 删除**已签出**的画面"1"，提示"**无法删除画面1,对画面1无权限或该画面被签出！**"。
14. 如果文件夹xxx下有**已签出**的画面，**删除该文件夹**，提示"**xxx下存在被签出的画面，无法删除!**"。
15. 新建画面后，画面为**已签出**状态，该画面父文件夹下的画面签入签出状态应该为最新状态。
16. 导入画面后，画面状态不改变。
17. 粘贴画面，画面为**已签出**状态。
18. 画面的画布右键菜单可以签入图元。

签入后，会记录在历史版本。

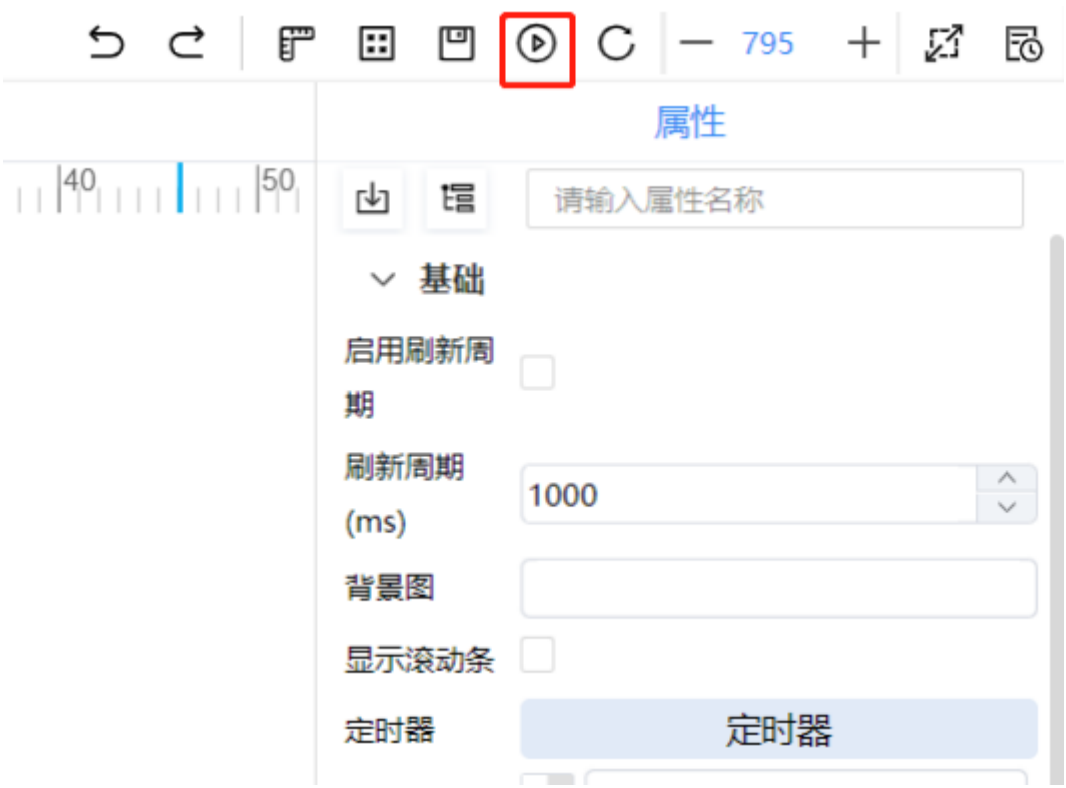
5.12 画面预览

预览画面

1. 找到设计好的画面，先保存画面



2. 预览画面前保存修改并签入，直接点击键盘P即可预览，或者点击右侧三角形预览按钮


795
属性
40
50
请输入属性名称
基础
启用刷新周期
刷新周期(ms)
1000
背景图
显示滚动条
定时器
定时器

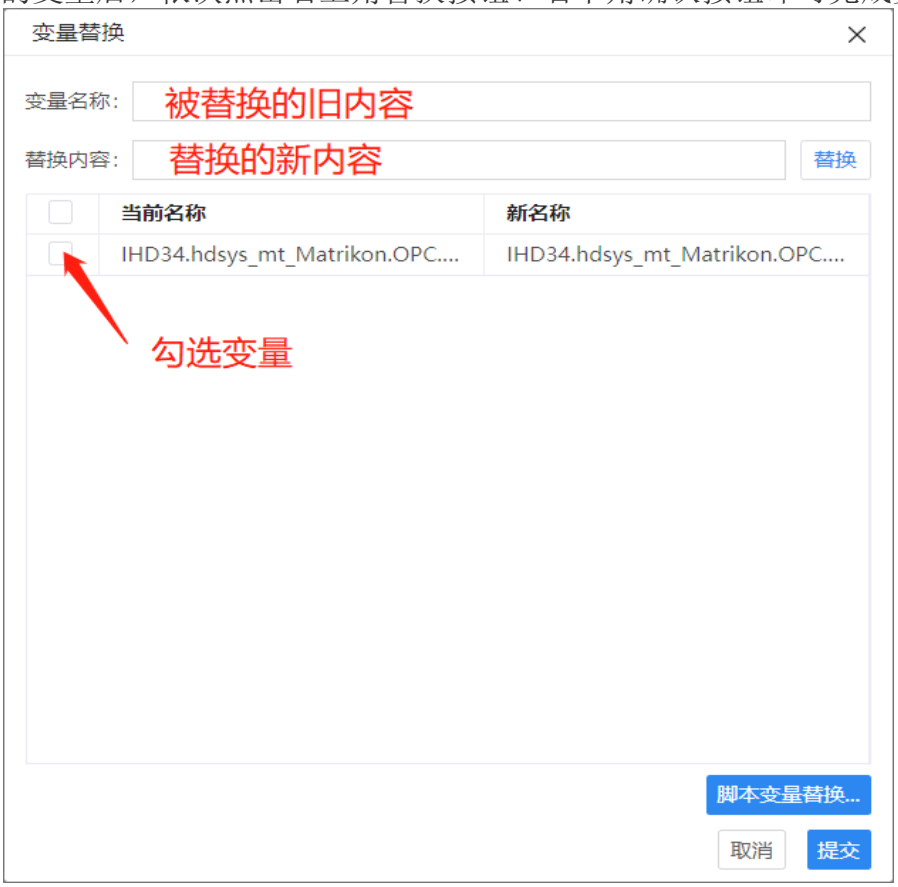
3. 运行结果显示

不安全 | 10.25.10.166:8080/iplat/web/BICR00?tag=tempDisplays/首页画面/demo.json
QQ邮箱
Gmail
YouTube
地图
Gmail
YouTube
地图
表单 | 麦客CRM
智慧制造166
宝信
1

5.13 元素的变量替换

1. 点击元素，可见元素周边出现蓝色边框，在蓝框内右键选择变量替换。

2. 勾选需要替换的变量后，依次点击右上角替换按钮、右下角确认按钮即可完成变量替换。

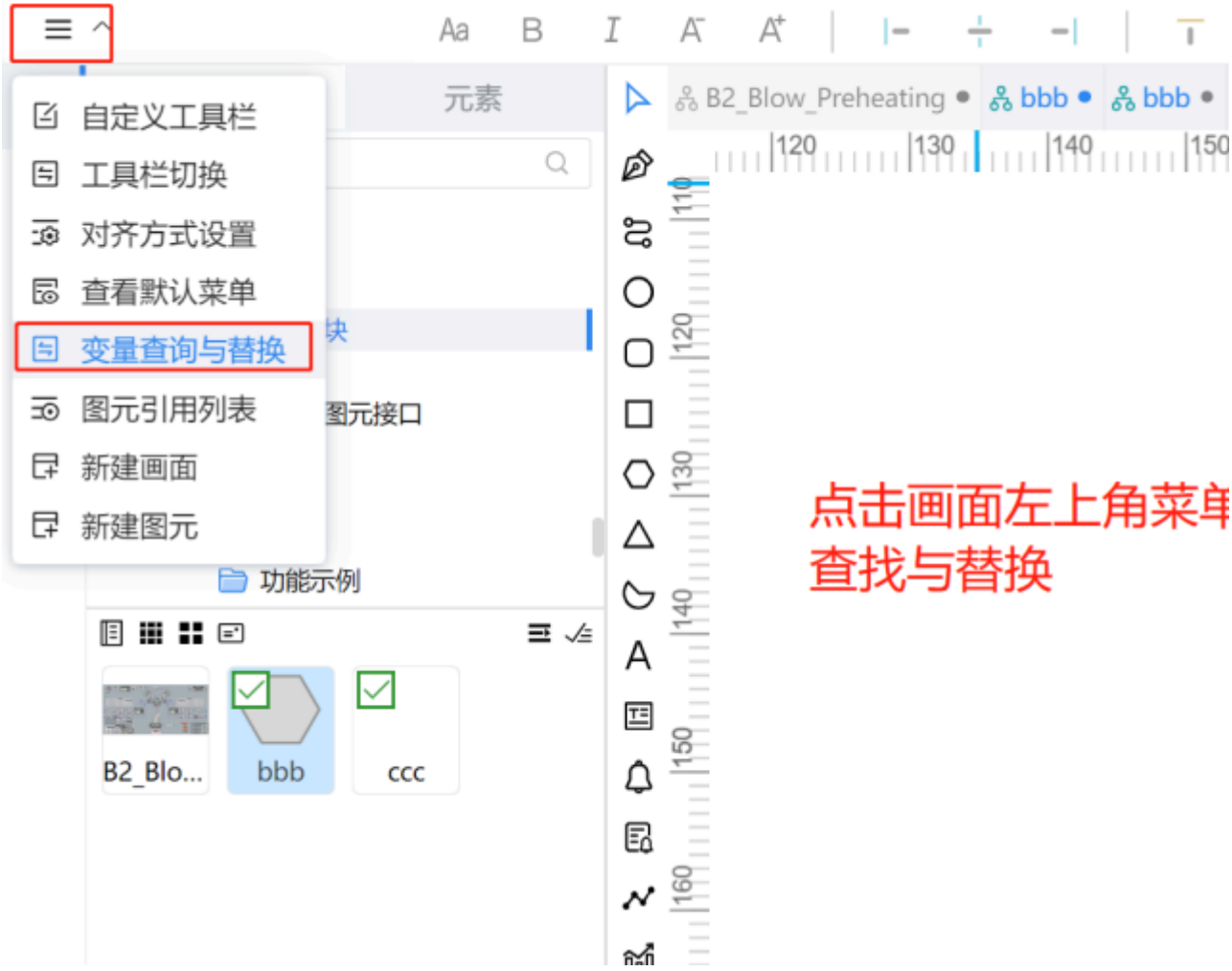


- 变量名称：想要替换的变量内容
- 替换内容：将要替换成的变量内容•整体流程：
  a. 填写变量名称和替换内容

b. 勾选想要替换的变量
c. 点击右上角替换按钮完成替换
d. 点击右下角确认按钮完成修改保存

5.14 画面的变量替换

1. 在画面左上角点击菜单->文件->变量查找与替换弹出变量替换窗口。



2. 点击查询根据查询名称查询变量，接下来操作与元素的变量替换相同。

•操作步骤：

a. 点击查询即可在下方表格看到画面所有变量，也可在查询名称填写条件（支持模糊查询）实现条件查询

b. 填写查询名称和替换内容，注：查询名称＝》替换内容

c. 勾选需要替换的变量

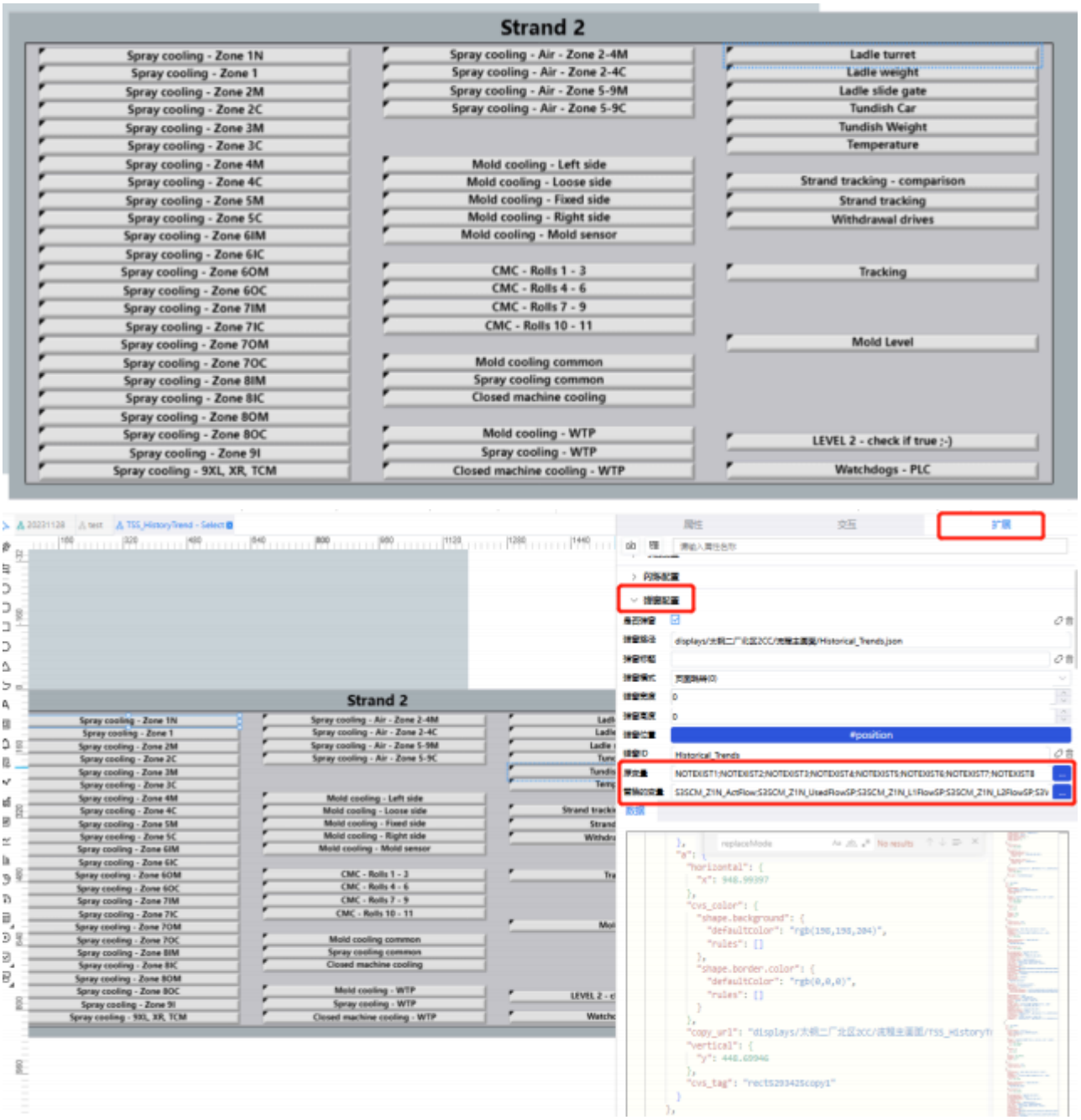
d. 点击右上角替换按钮实现变量替换

e. 点击右下角确认按钮实现修改保存

编辑态配置至运行态的变量替换

1. 全量替换

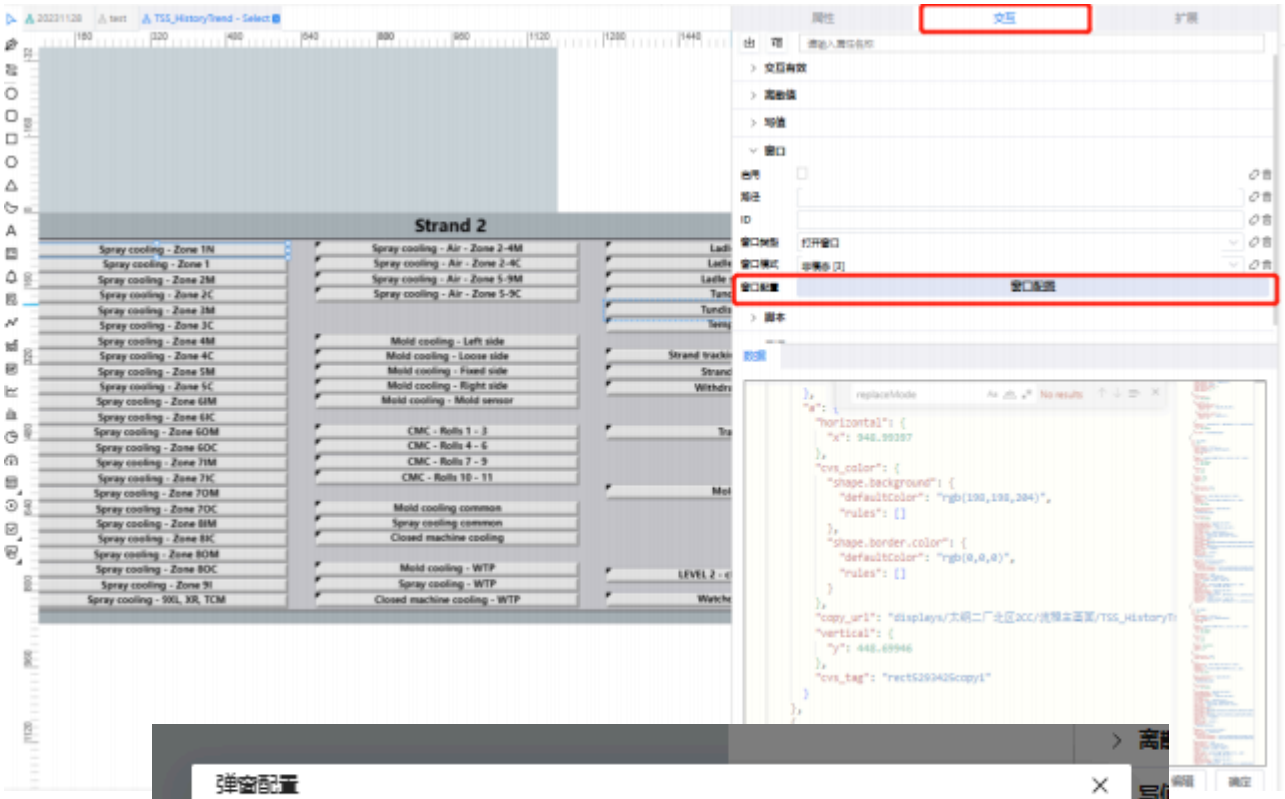
需要弹出多个趋势图时，无需配置多个趋势，而是可以通过弹窗及变量替换的方式，对某个趋势的数据进行替换即可。方法为：扩展->弹窗配置->原变量&替换的变量



注意：这里使用的替换为全量替换，即会将原来趋势中配置的内容全部清除，然后通过变量替换的内容去绘制趋势。eg：原趋势有9条，替换中配置了2条。那么绘制的结果为2条

2. 部分替换

如果想要保留所有曲线，而且希望实现部分内容替换，则可以使用交互中的窗口配置进行变量替换。方法为：交互->窗口->窗口配置


Strand 2
交互
扩展

弹窗配置
宽高:
W
150
H
100
跟随鼠标
位置:
X
Y
相对画布
标题:
启用:
相对Top
显示标题
显示关闭
显示边框
关闭同id弹窗
禁用拖动
变量替换:
原变量
替换变量
操作
No Data
添加一行
传参:
Name
Value
操作
No Data
添加一行
关闭回调:
function () {
}
确定

功能区分

1. 元素的变量替换只对于单个元素进行变量替换
2. 画面的变量替换支持模糊查询，可对画面上的多个元素进行变量替换

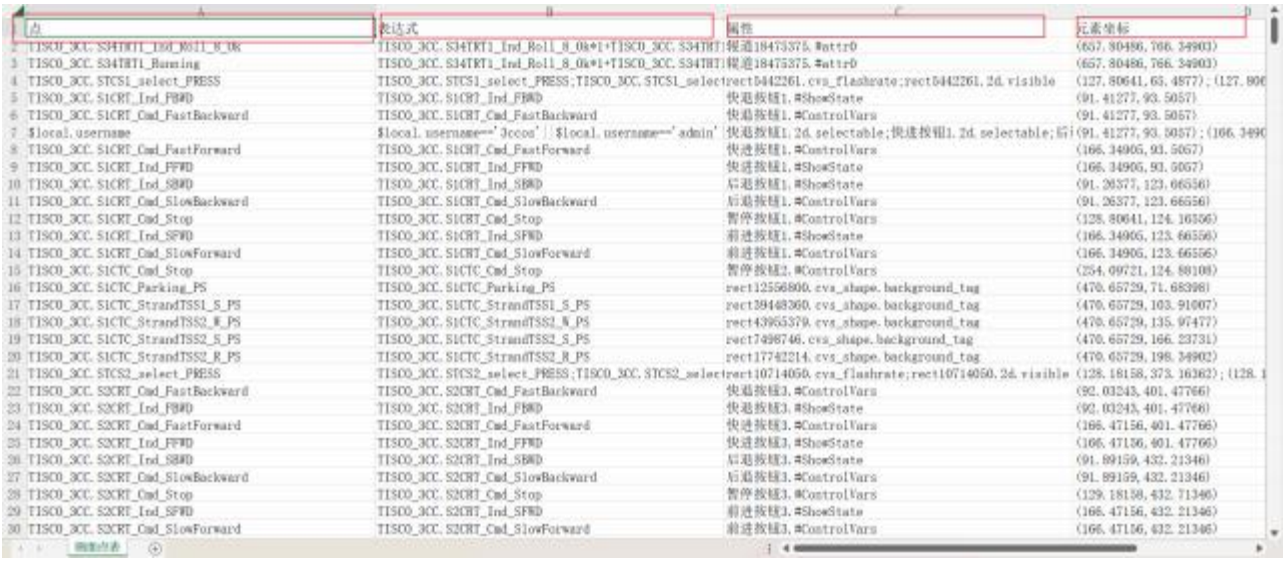
5.15 画面变量导出和导入

画面变量导出

1. 打开画面，右键画布，点击导出画面点，成功导出变量表

2. 打开变量表

第一列：变量名
第二列：该点对应在画面中绑定的表达式
第三列：该表达式绑定于哪个属性
第四列：绑定该变量的元素的坐标
第五列：对应属性的类型



画面变量导入

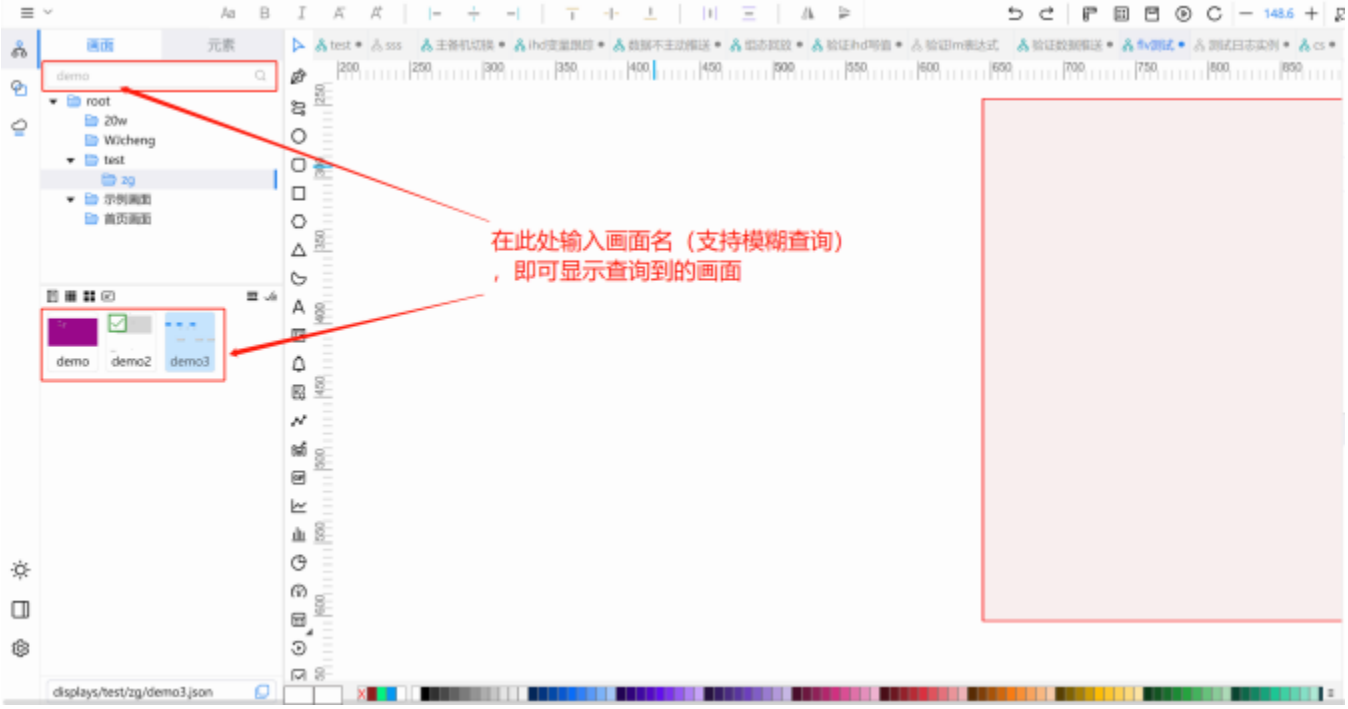
修改导出画面的Excel文件

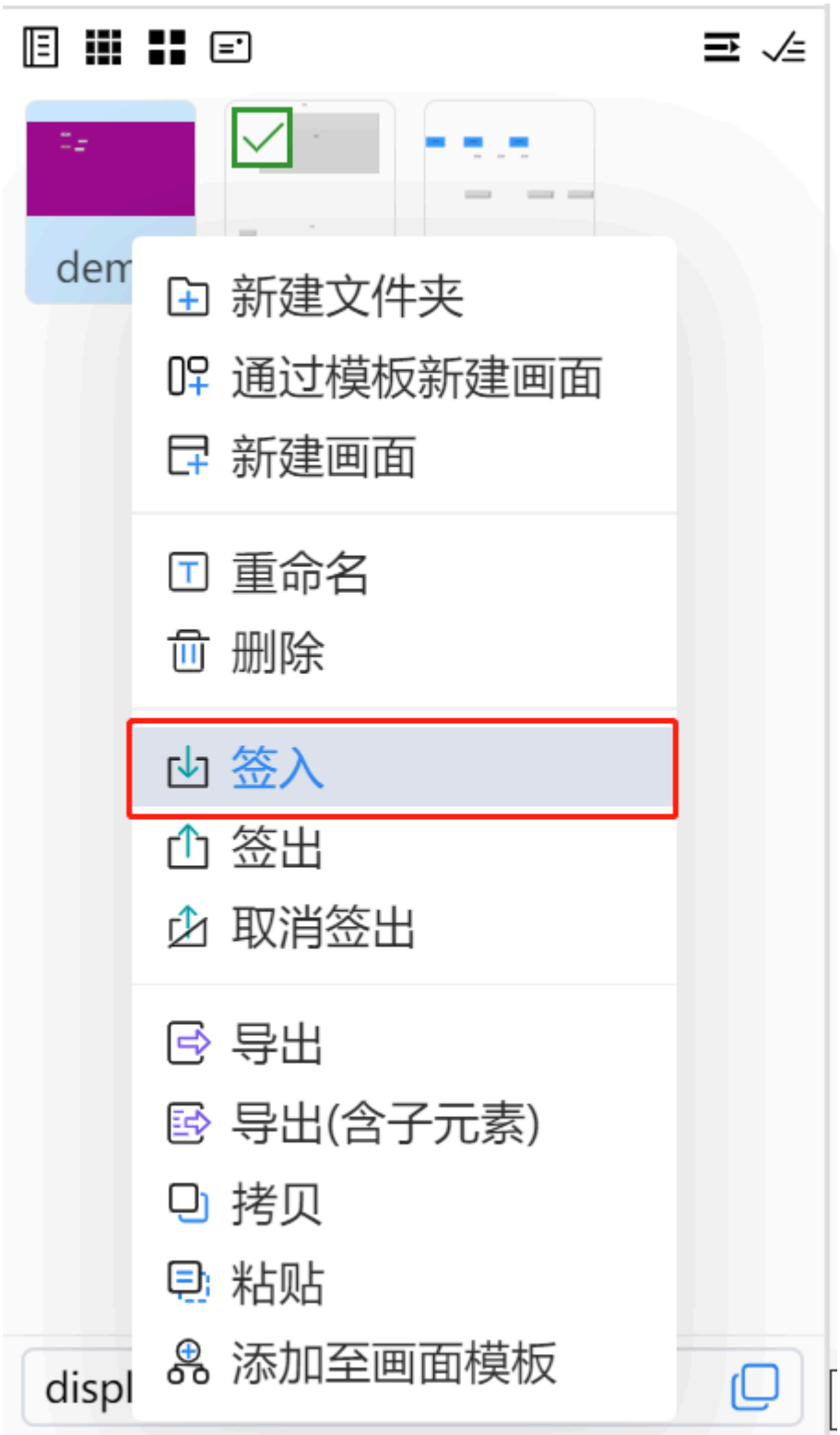
1. 修改表达式那一列，其中不同表达式以;分隔符号分割
2. 导入到画面中，置空的表达式会在画面中将对应元素的属性的绑点删除

5.16　画面查找与定位

画面查找

在左上角输入框输入画面名称即可在下方展示查询到的画面点击右键选择签入

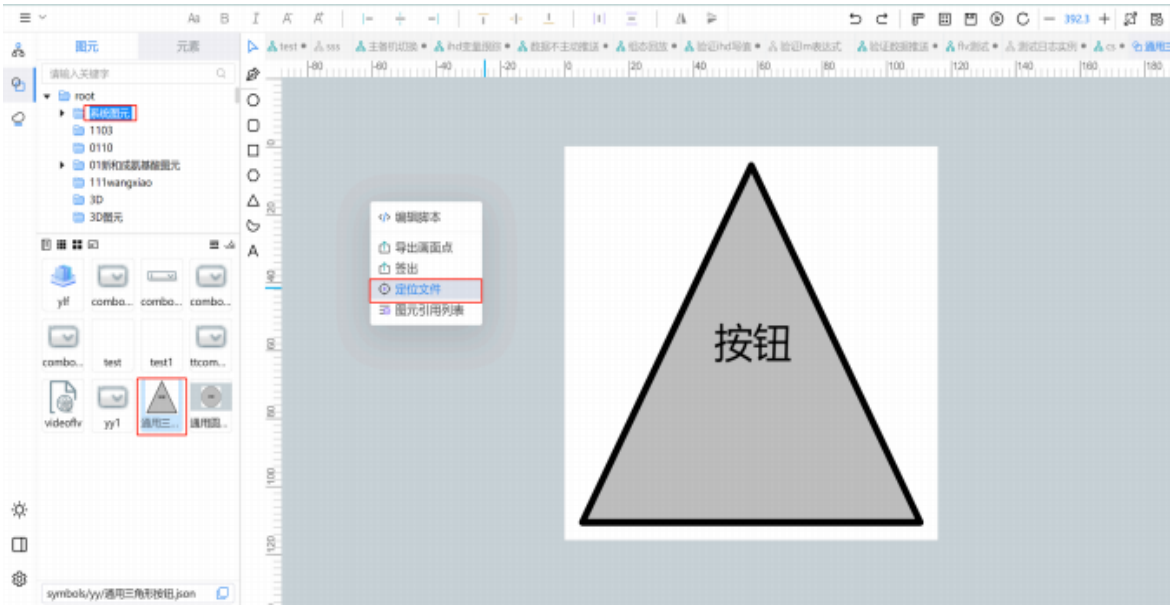




画面定位

打开画面图元过多，这时想在左侧找到该文件就不易找，可以通过右键单击想要定位文件的画布，选择定位文件，左上方文件树就会展开到对应文件的父级文件夹，左下方也会移动滚动条至

文件所在处并将文件选中

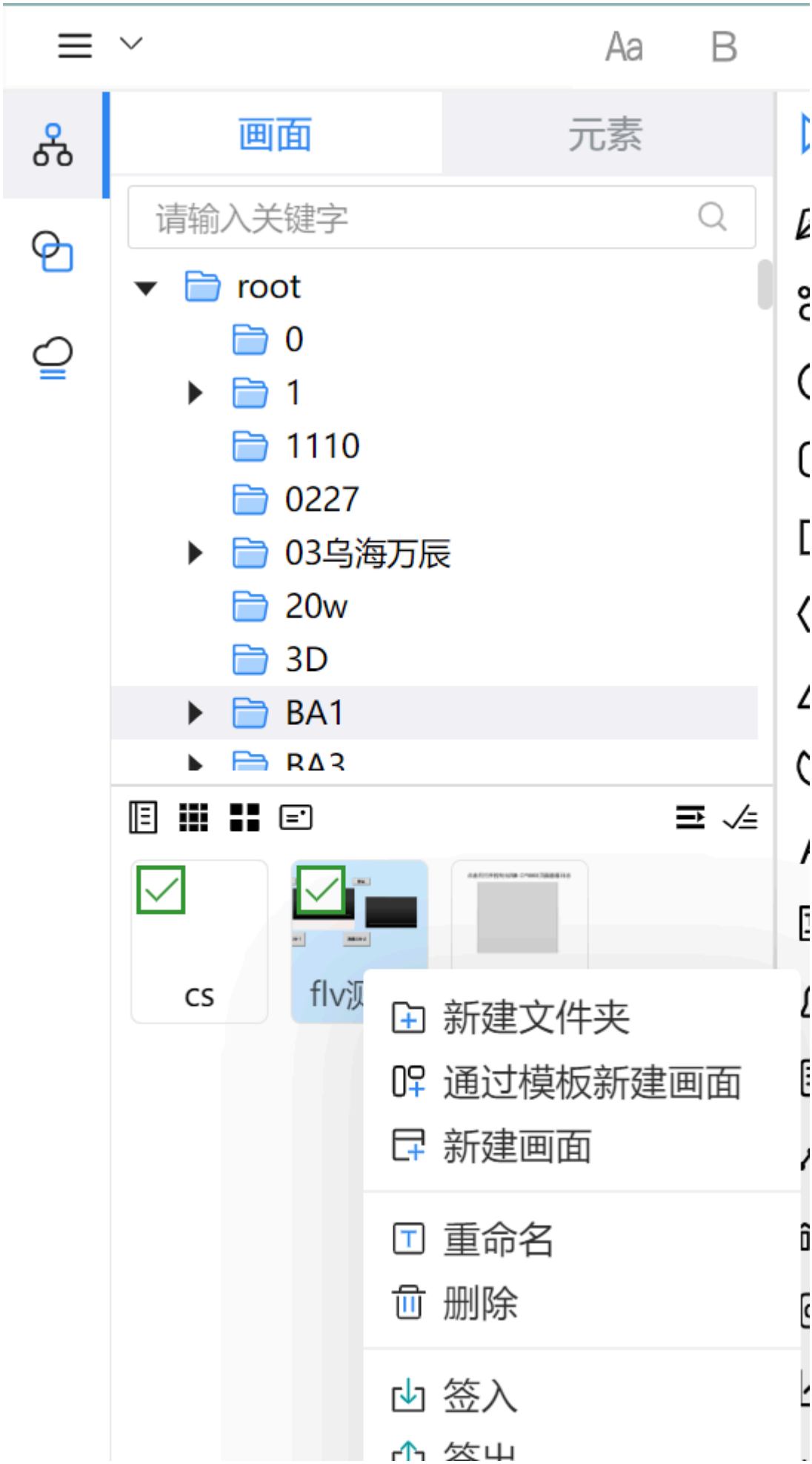


5.17 画面导出导入

注意：导入会覆盖当前的资源，包括画面、图元、资源（系统图元不会被覆盖）

画面导出

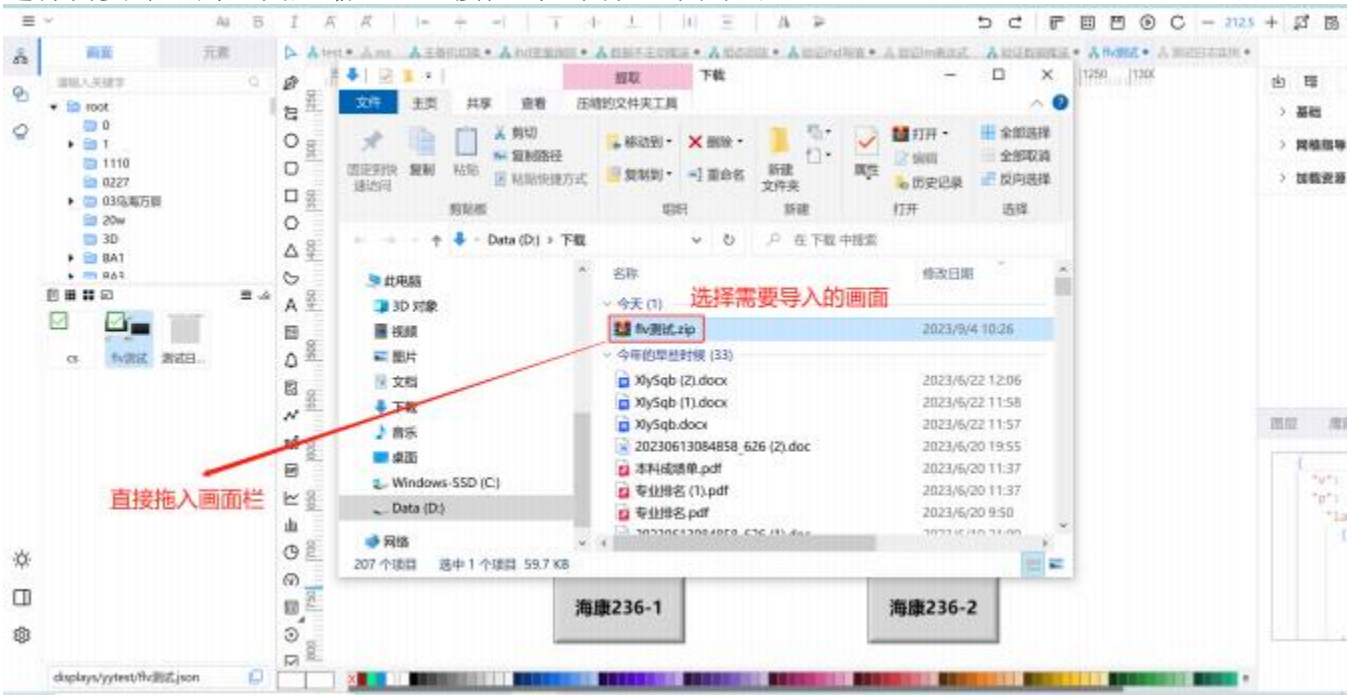
1. 右键点击画面，选择导出，可导出画面为zip形式


Aa
B
画面
元素
请输入关键字
root
0
1
1110
0227
03乌海万辰
20w
3D
BA1
cs
新建文件夹
通过模板新建画面
新建画面
重命名
删除
签入

2. 在浏览器下载内容中显示

画面导入

选择需要导入的画面压缩包，直接拖到画面栏上面即可



画面导入时，会跳过名称中含有非法字符的文件，非法字符包括/:*?"<>|#,.%&导入时，只有合法的路径才能导入

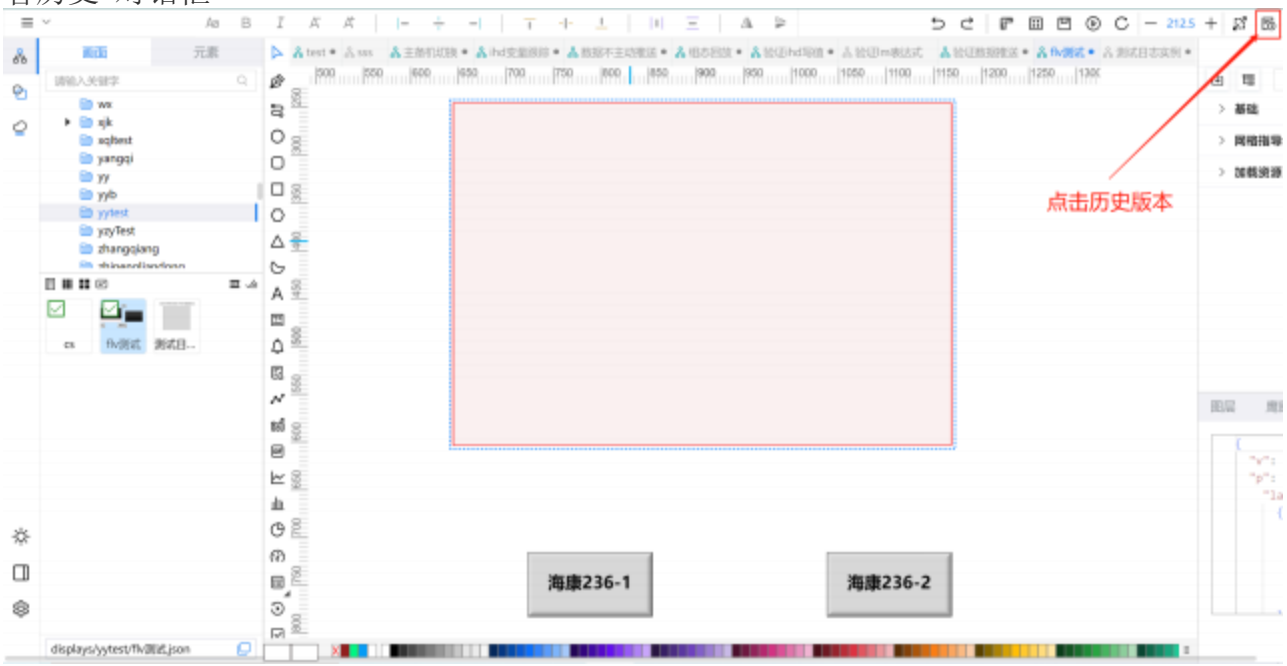
路径	含义
storage/	根目录
storage/displays/	画面路径
storage/symbols/	图元路径
storage/assets/	资源路径

storage/components/	组件路径
storage/scenes/	3D画面路径
storage/models/	3D模型路径
storage/menu/	导航菜单路径
storage/previews/	预览画面路径

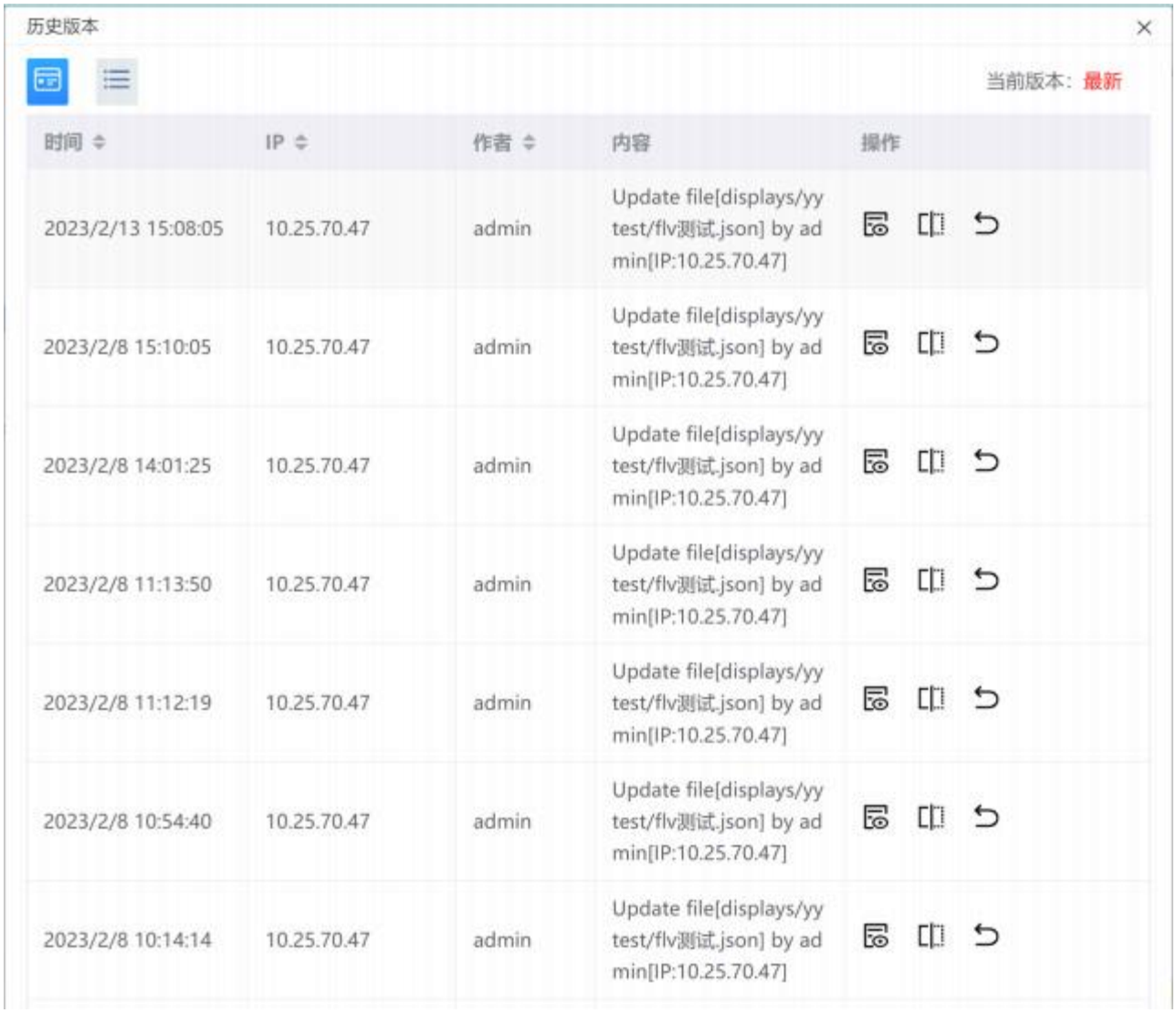
5.18 画面历史版本

画面历史版本

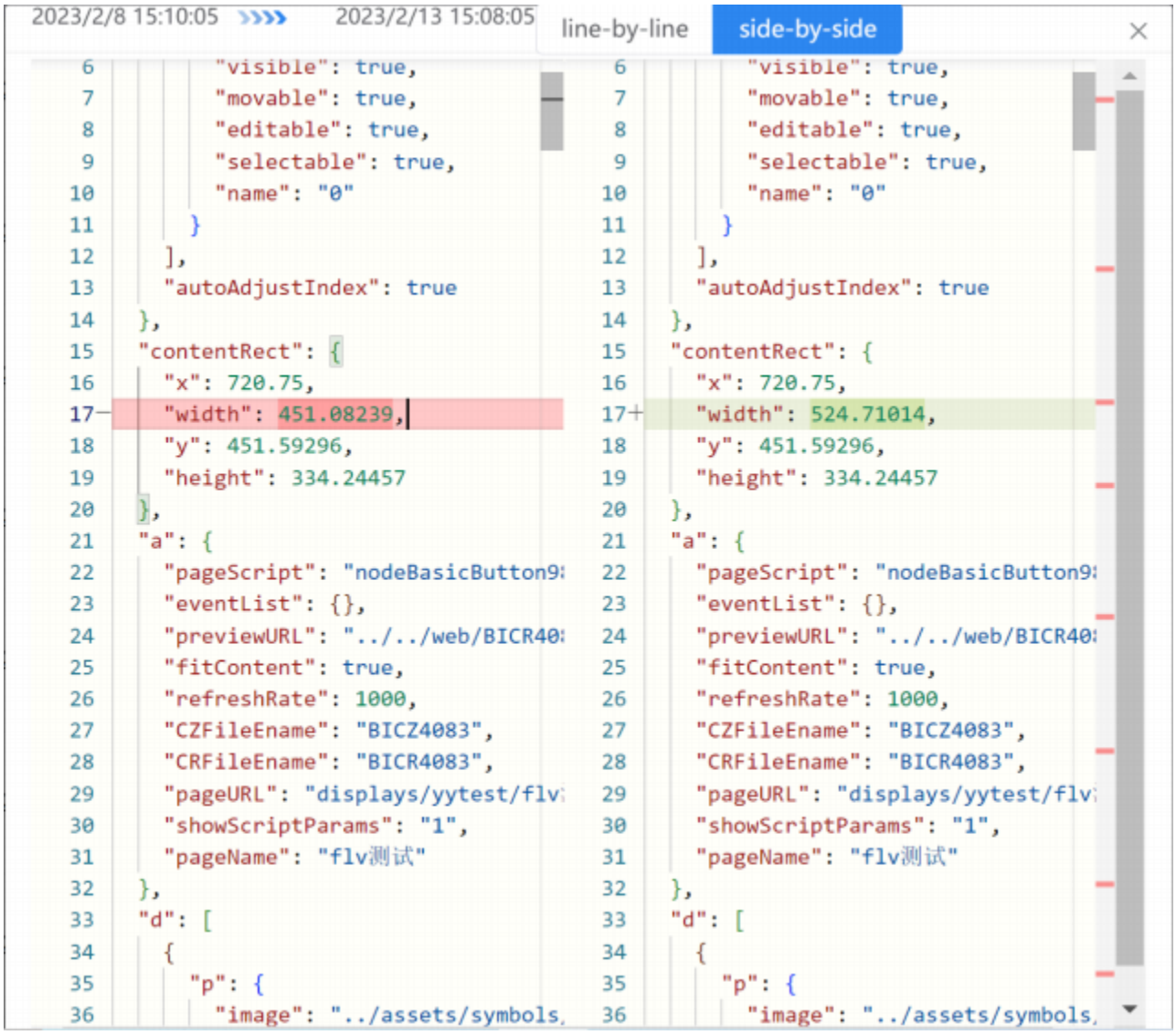
1. 在2D设计器内打开一幅画面或者一个图元，点击工具栏右上角的"历史版本"按钮，弹出"查看历史"对话框



2. 选择一次历史记录，点击"查看此版本"按钮，可以在画布里看到这次历史记录对应的文件内容



3.选择两次历史记录，点击"对比差异"按钮，可以看到两次版本文件内容的差异



•Tips：

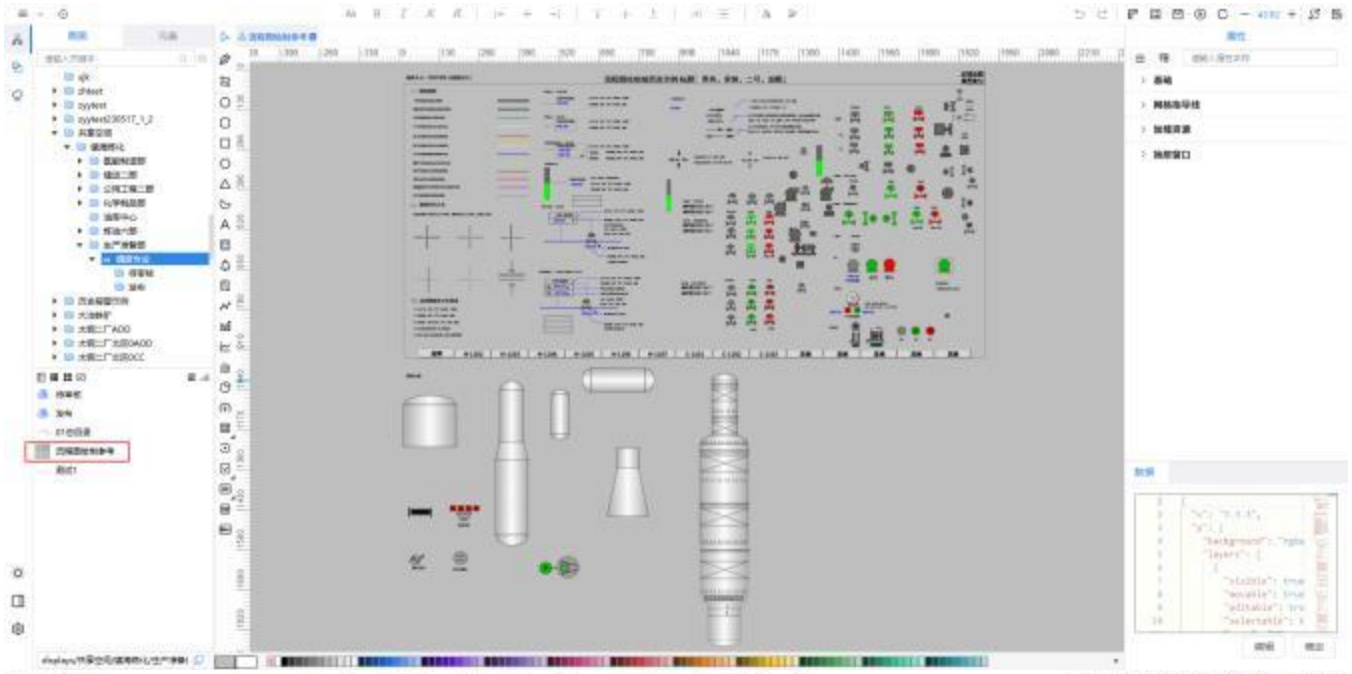
- 。导入文件（画面/图元/场景）会自动记录一次历史版本
- 。新建的文件不会记录历史版本，需要签入记录该文件的第一个历史版本
- 。仅签入，文件内容未作任何改变，不记录历史版本

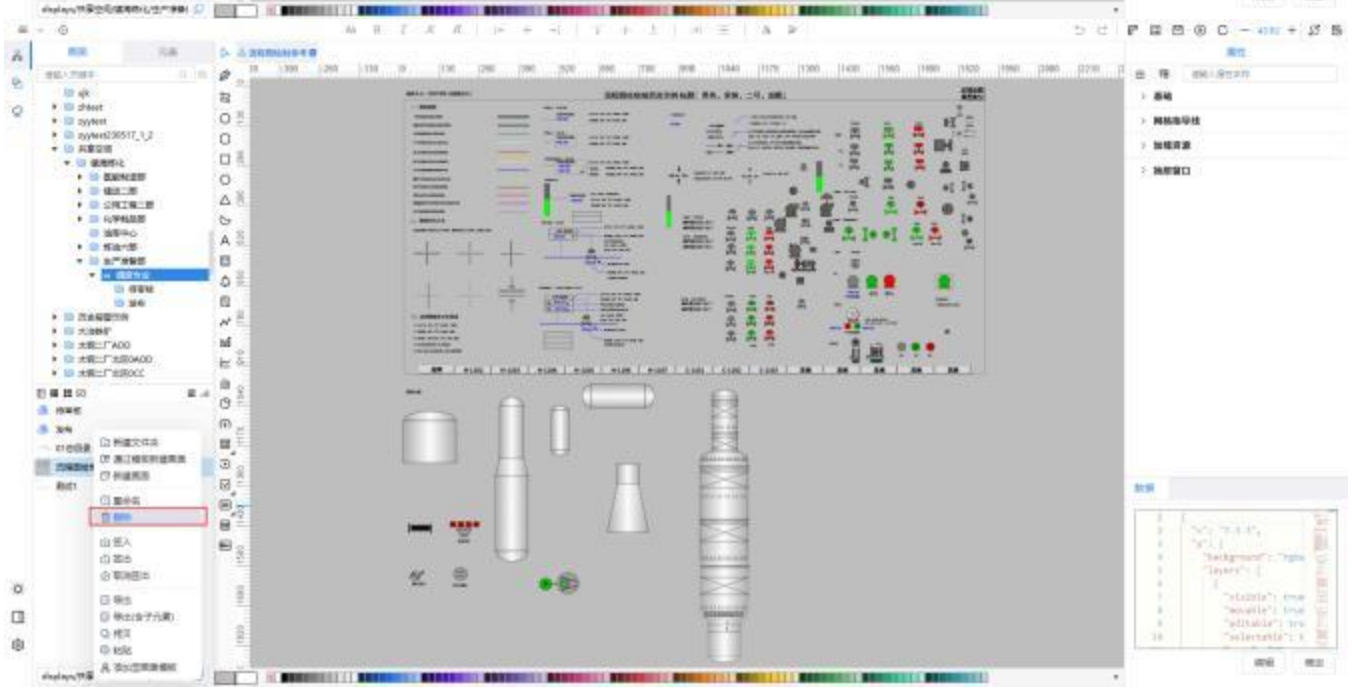
历史版本的另一个使用场景

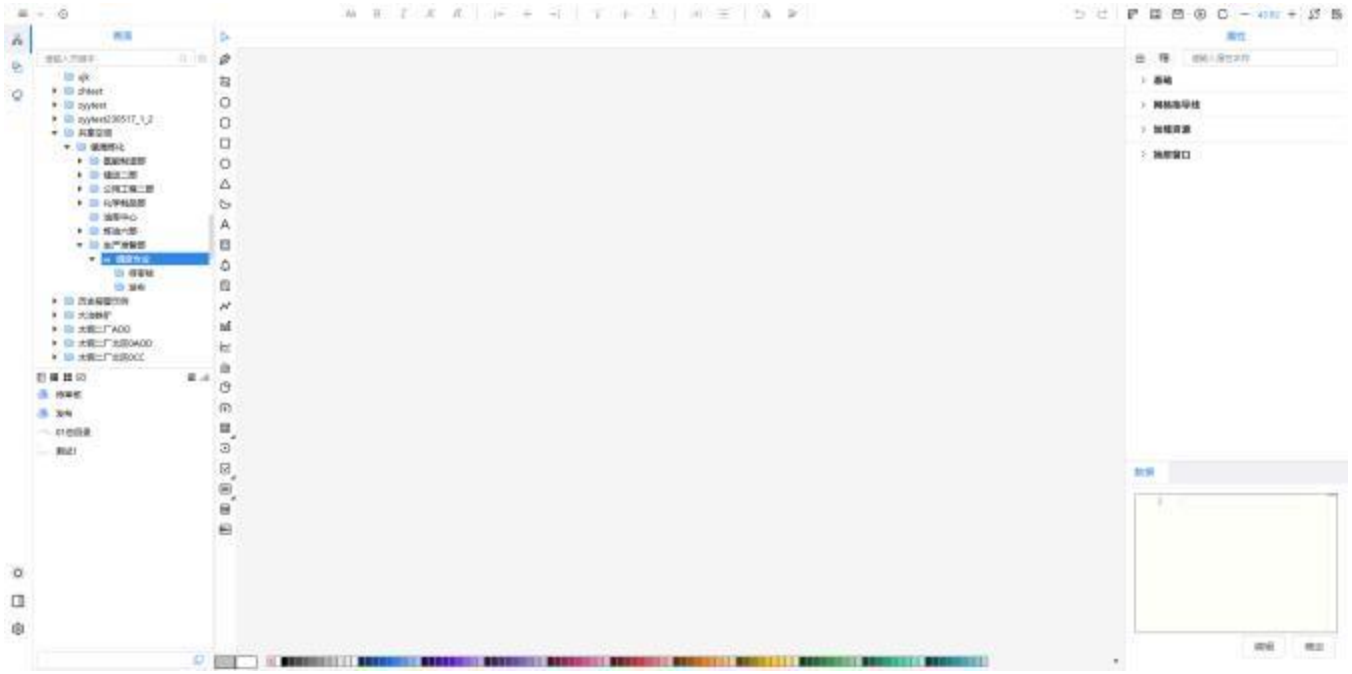
当组态中的画面不小心被误删除后，只要知道画面所在的文件夹，以及画面的名称，就可以通过历史版本找回被删除的画面。

以下是模拟步骤：

删除画面：

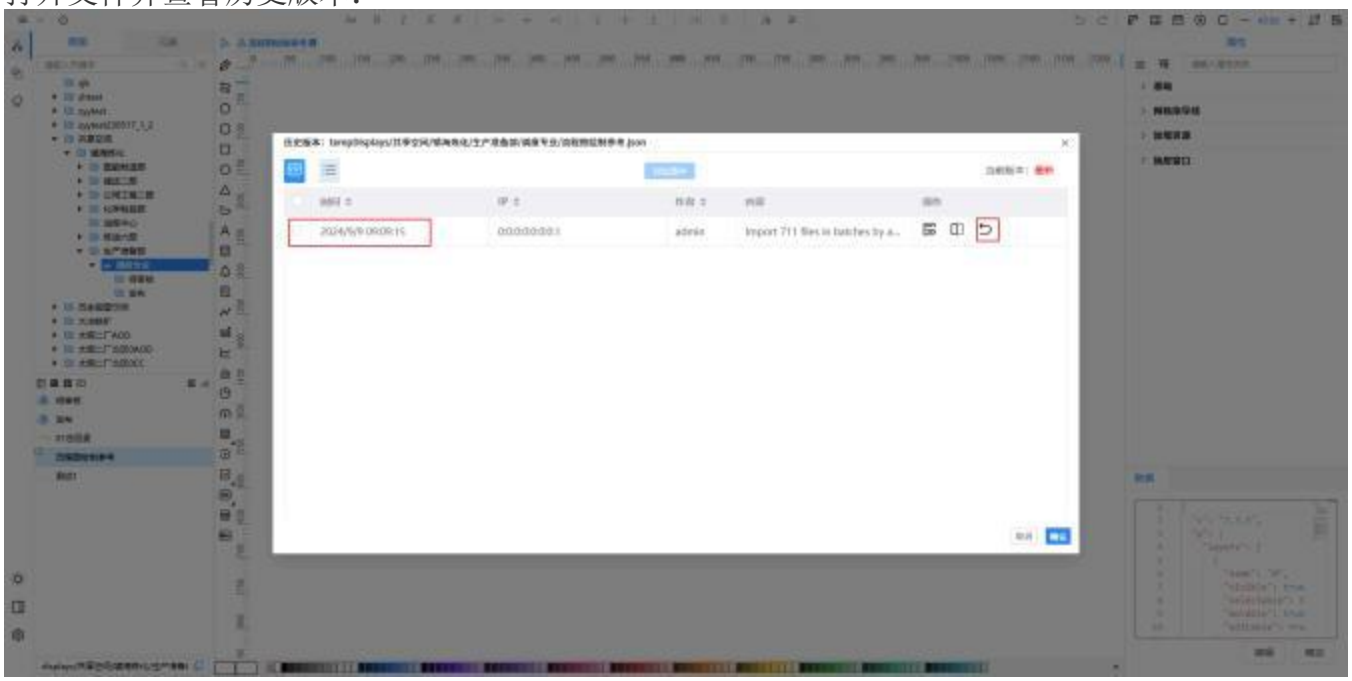




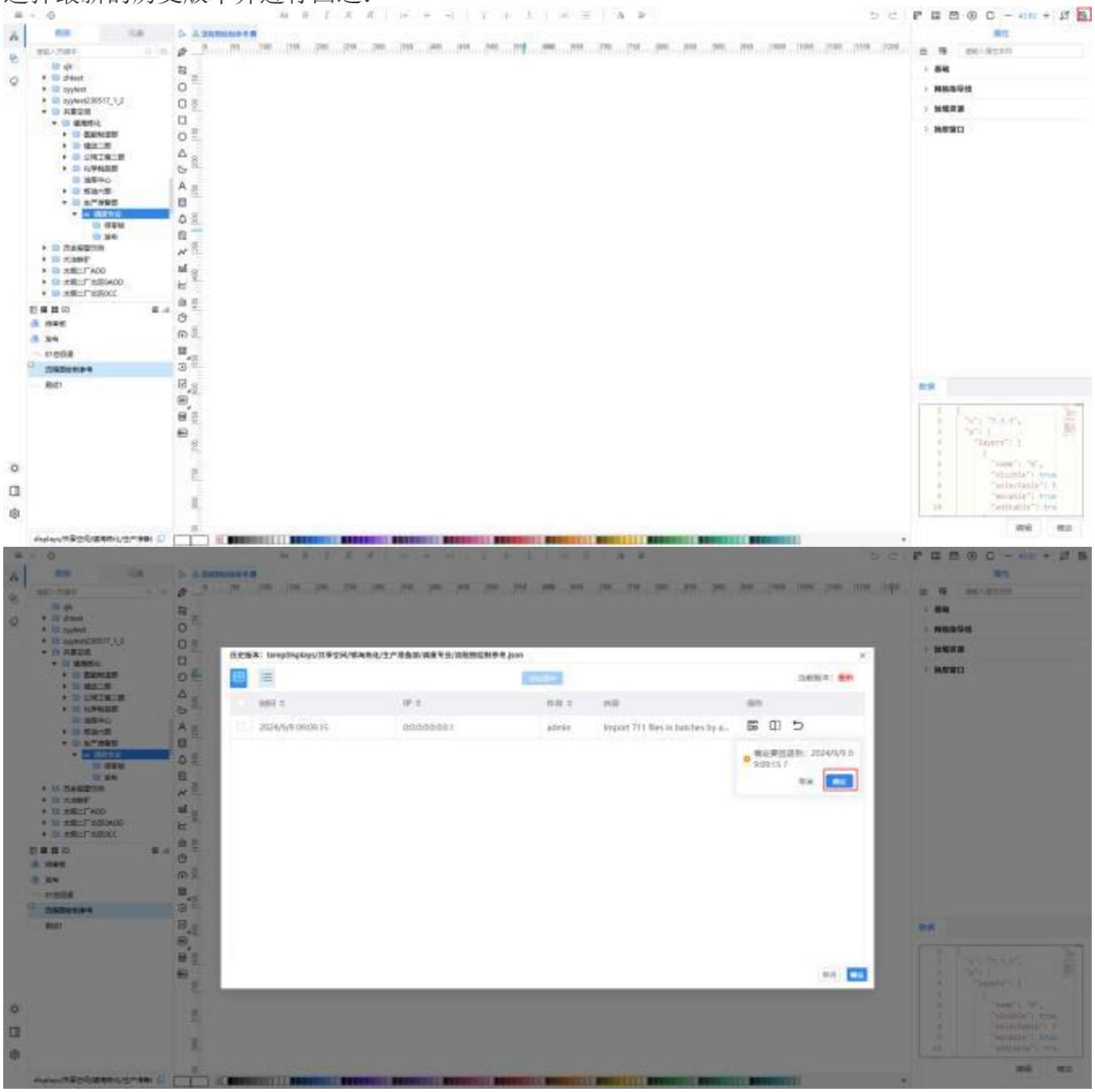


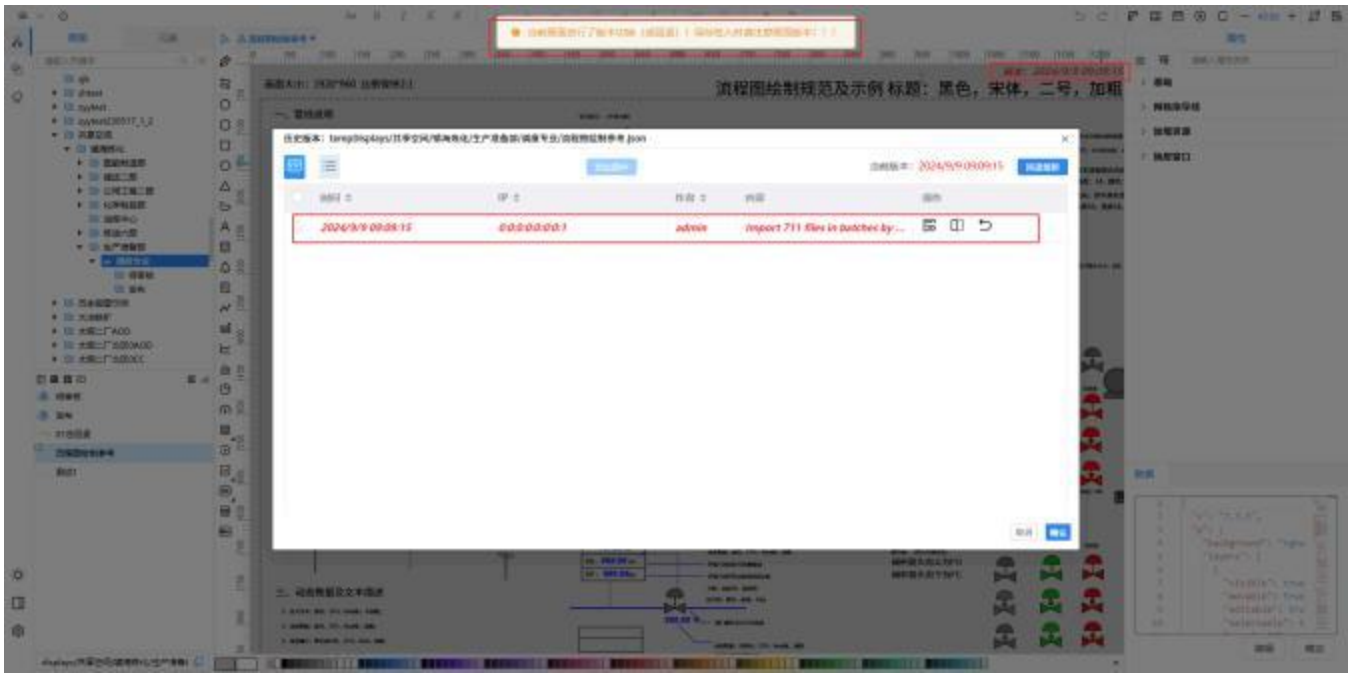
在原路径下新建同名文件：

打开文件并查看历史版本：

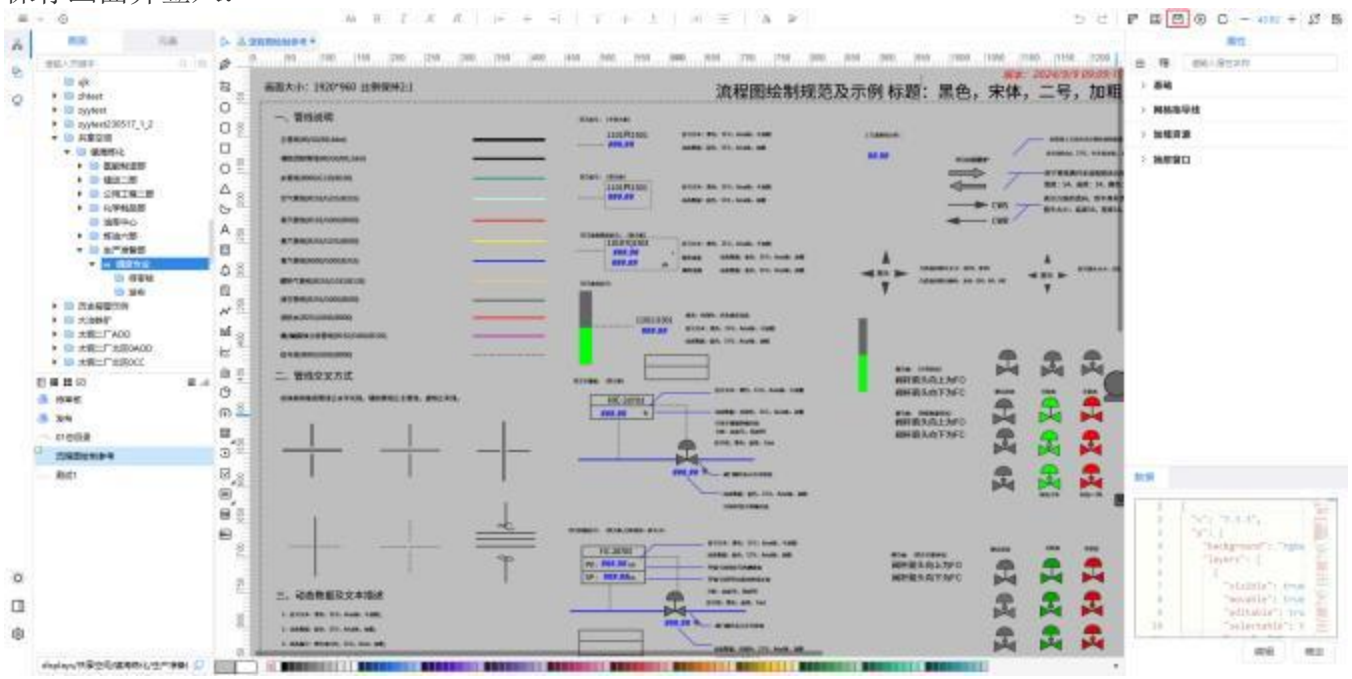


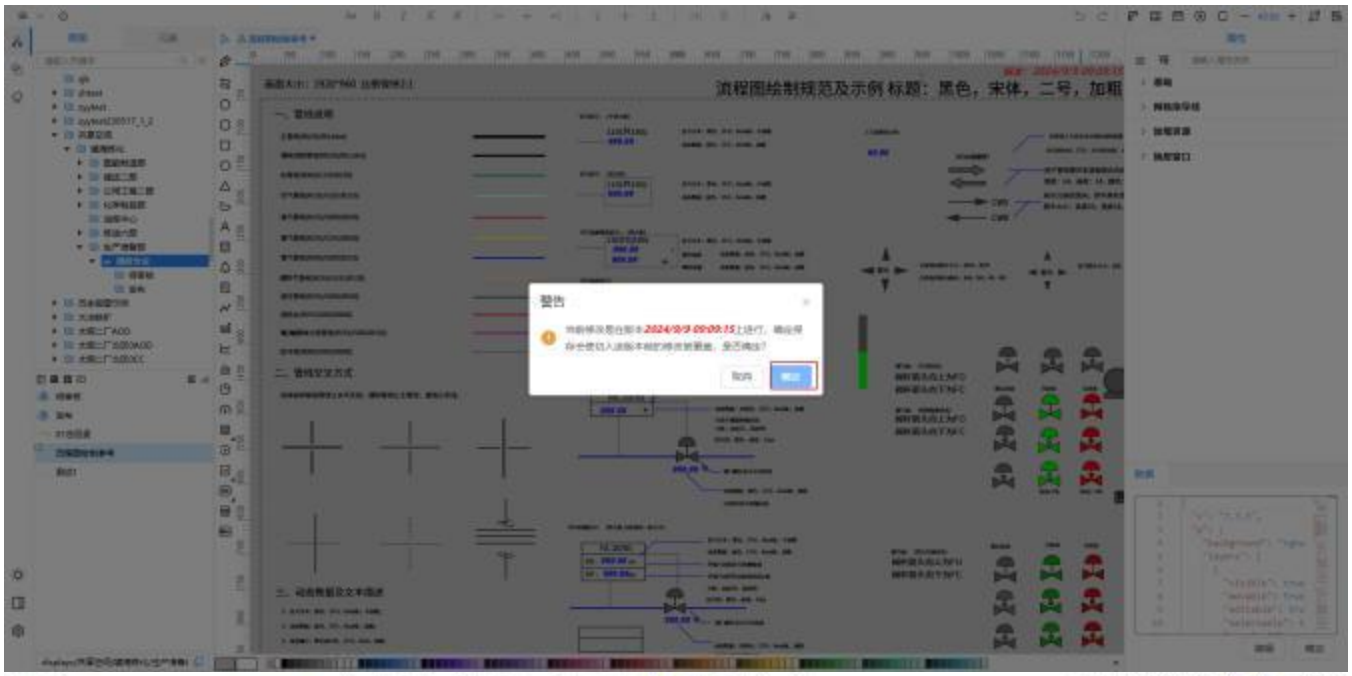
选择最新的历史版本并进行回退：

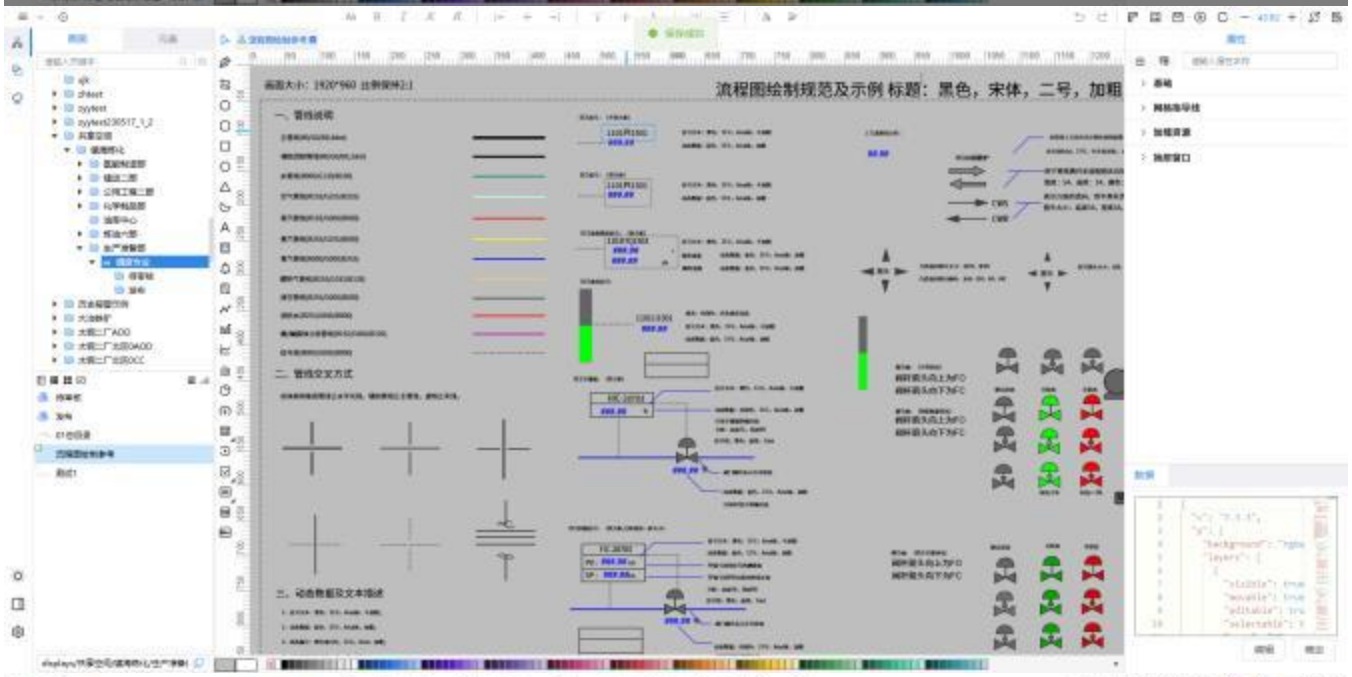


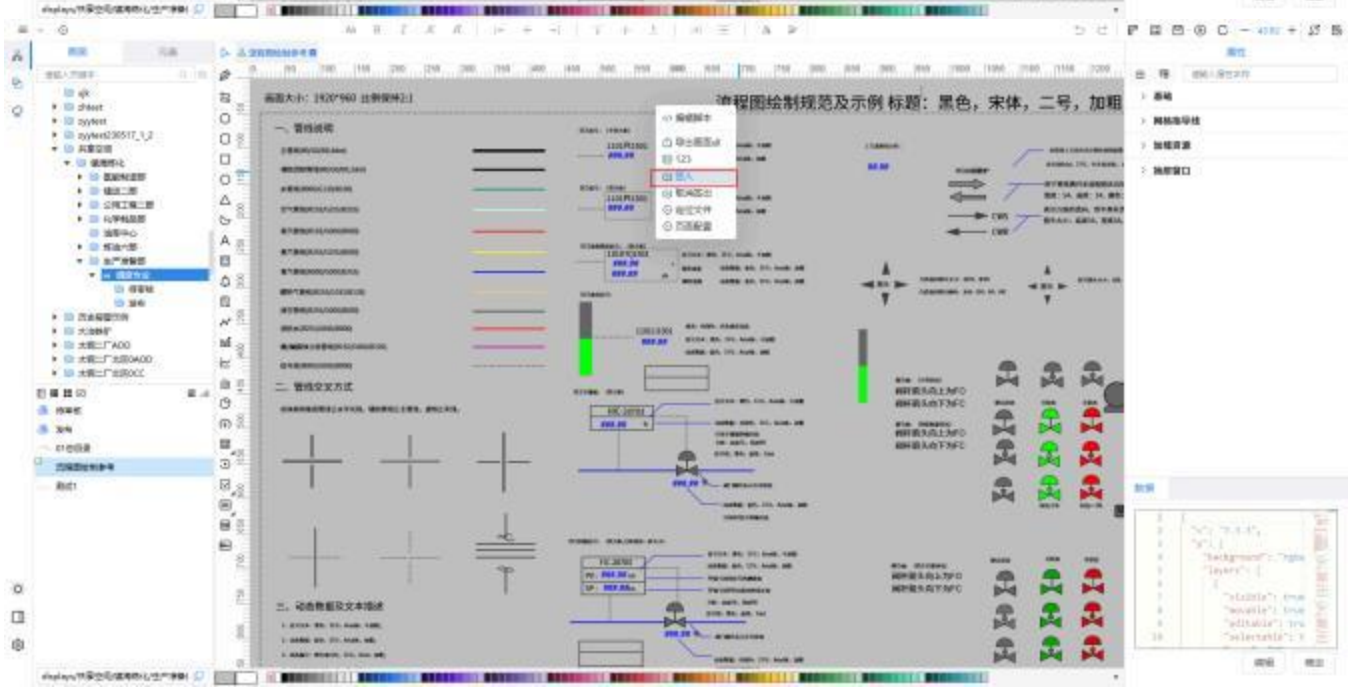

流程图绘制规范及示例 标题：黑色，宋体，二号，加粗
admin
import 711 files in batches by...

保存画面并签入：

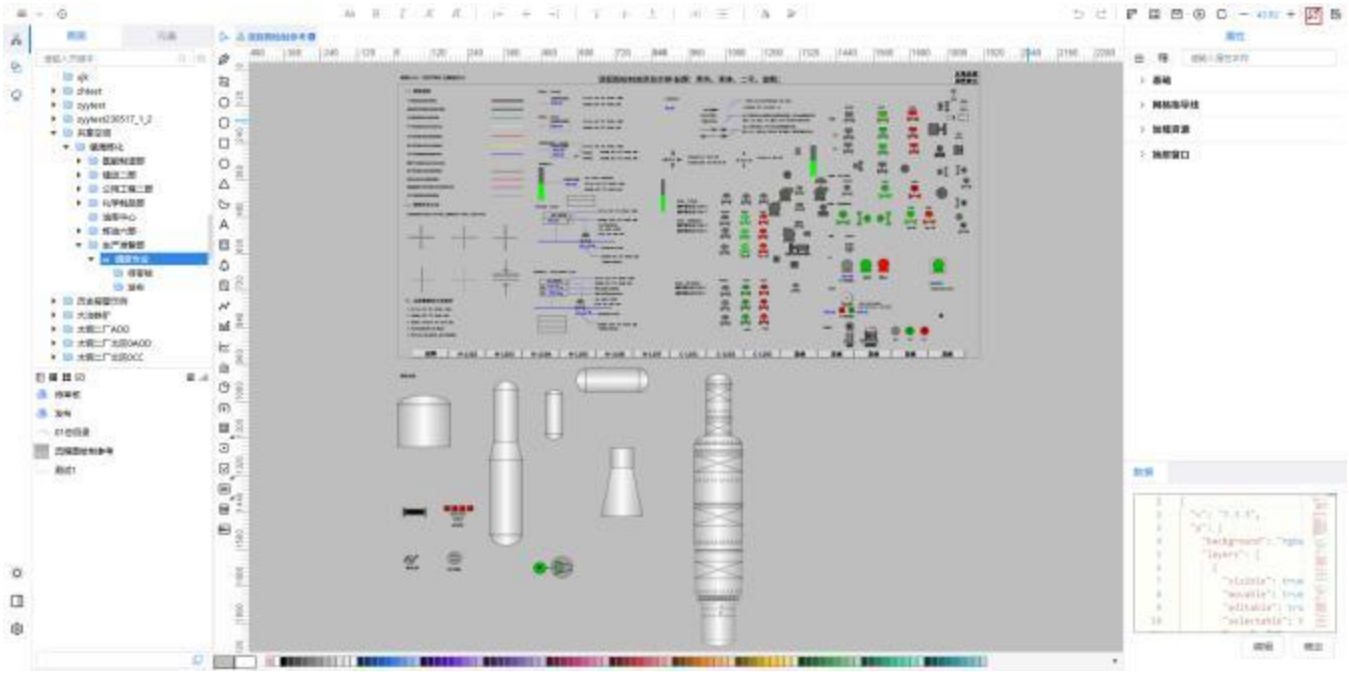

流程图绘制规范及示例 标题：黑色，宋体，二号，加粗


流程图绘制规范及示例 标题：黑色，宋体，二号，加粗
警告
取消
确定


流程图绘制规范及示例 标题：黑色，宋体，二号，加粗


流程图绘制规范及示例 标题：黑色，宋体，二号，加粗

画面的大小可能和默认的不一致，可以通过右上角的自适应，使得画面完全显示出来



5.19画面快捷方式

ALT+P

•使用场景：快速回到指定运行画面编辑态

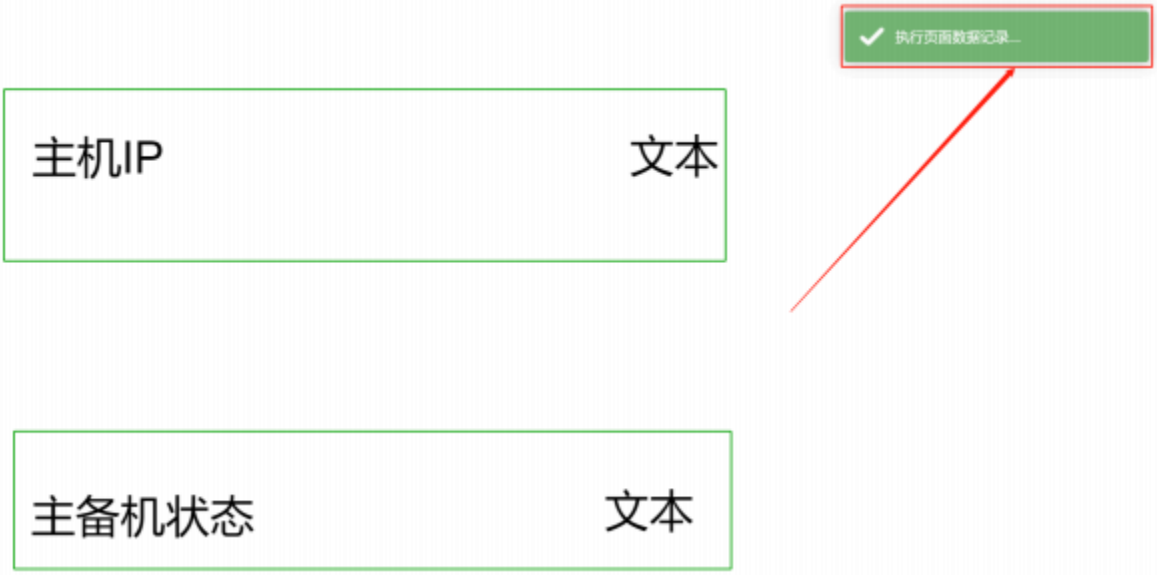
•使用方式：点击当前运行的画面，使用ALT+P回到该画面编辑态

ALT+M

•使用场景：记录当前画面日志

•使用方式：

◦点击当前运行的画面，使用ALT+M，画面右上角弹出"执行页面数据记录"信息提示



◦进入页面：首页搜索BICP0005，进入BICP0005界面

◦日志查询：在BICP0005界面查询存储的日志信息

5.20 自定义字体导入

功能描述类文档主要用于描述某个模块的具体功能，并指导用户使用。
示例内容如下：

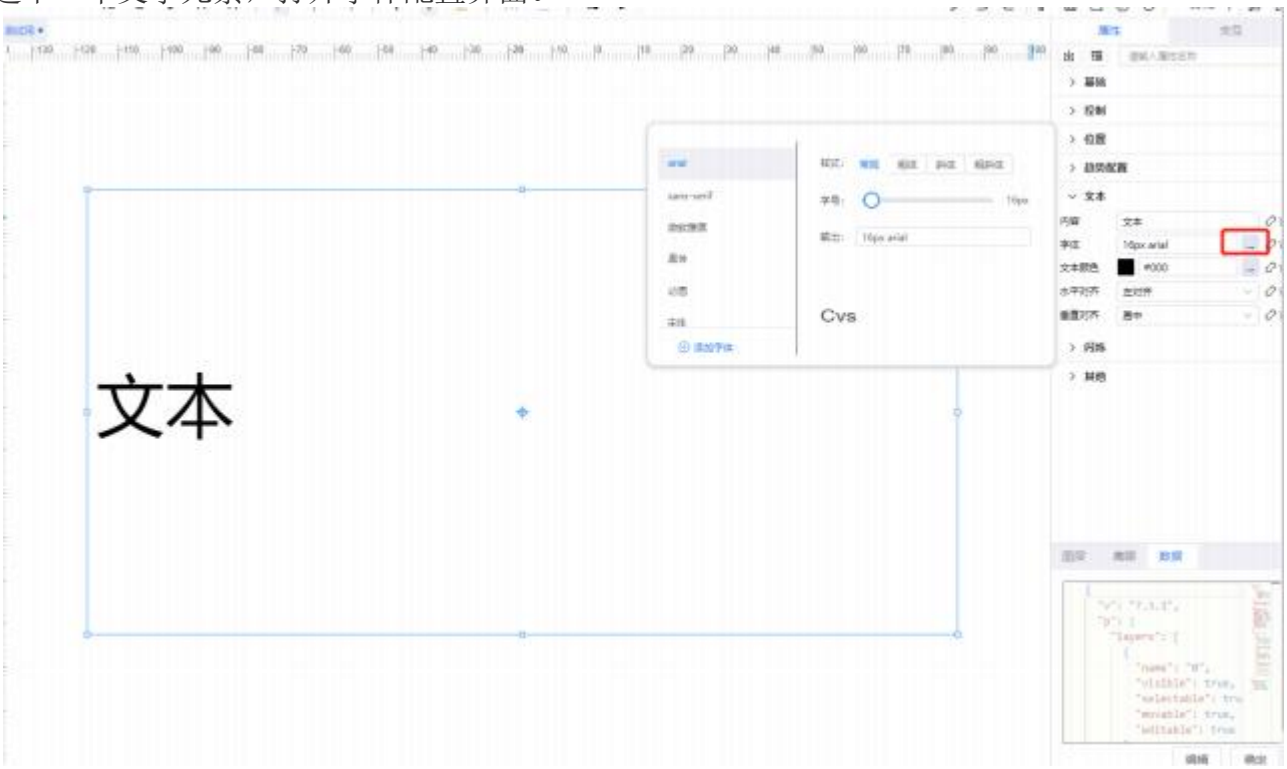
概述

编辑态支持导入ttf字体文件，并且能配置到编辑器中，以供字体选用。以此来满足项目使用的是计师特制的字体，或其他额外字体的需求。
目前支持.ttf,.eot,.woff,.svg,.otf格式

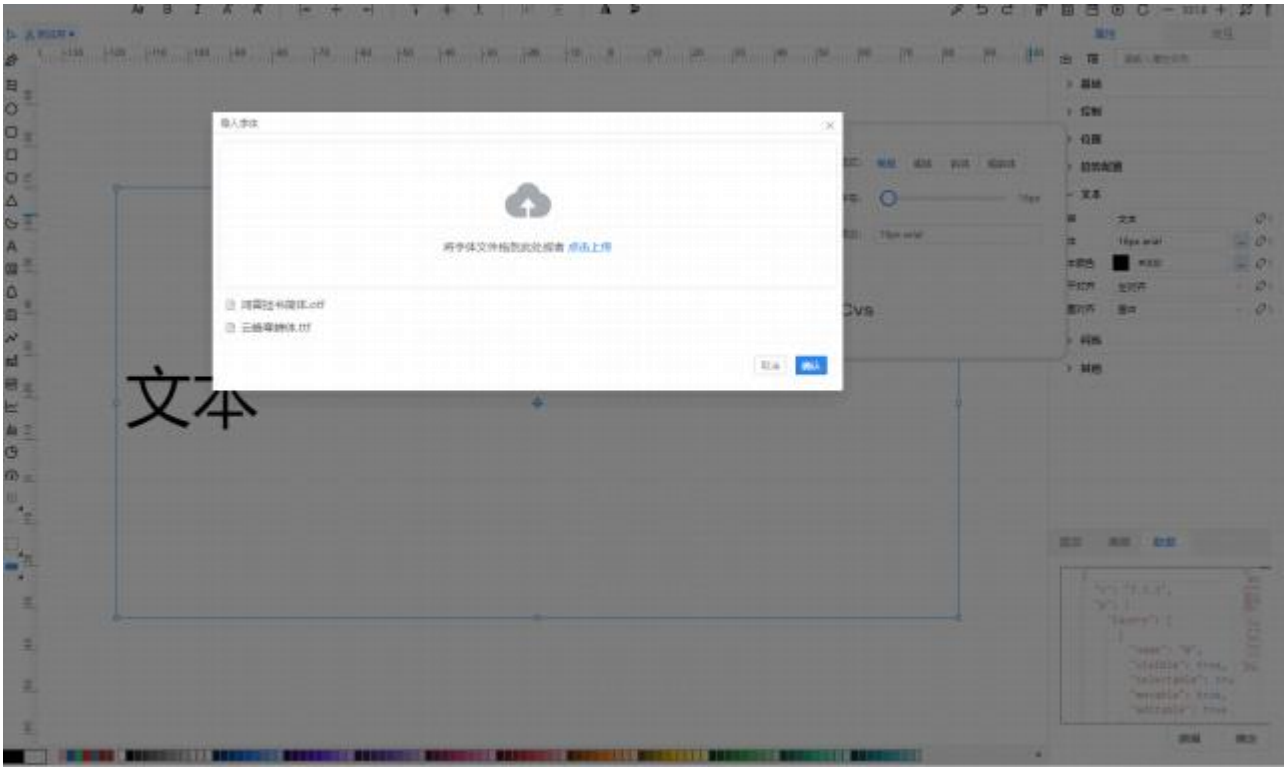
注意:字体文件名不得含有*<>?:"/|'[]%;#!$&(),.@特殊符号

操作步骤

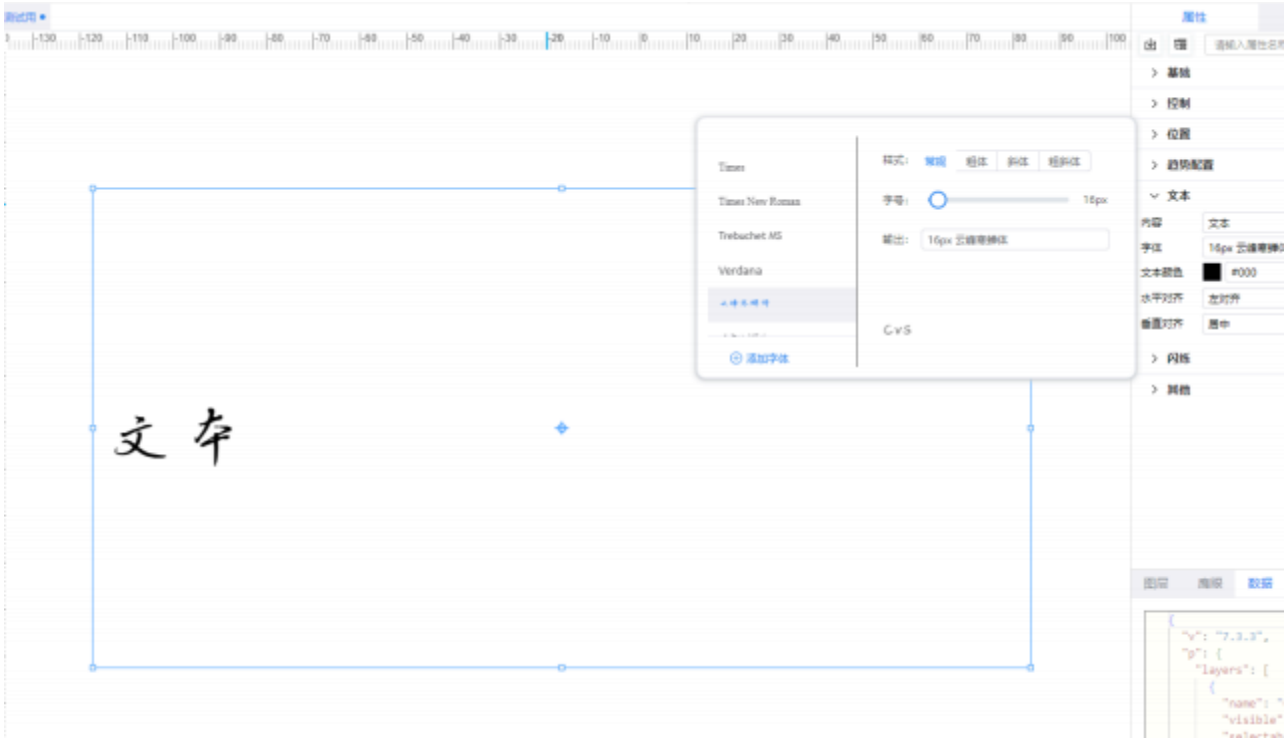
1. 选中一个文字元素，打开字体配置界面。



2. 点击添加字体，将自定义字体文件拖入上传框，点击确认，添加完毕。



3. 可选择配置该字体。



4. 在字体上右键可删除对应字体

5.21 画面页面配置

画面页面配置主要作用是对运行态的画面进行属性设置，包括但不限于页面的宽高，刷新周期，页面的缩放。

右键画面空白处单击，选择页面配置

弹出配置对话框

宽高设置

画面宽度和高度默认值是空。也可以在右侧属性栏配置

刷新周期

刷新周期也可以在右侧属性栏配置

勾选启用表示使用刷新周期

缩放配置

启用缩放
勾选表示可缩放，使下面配置生效。

最小值
在启用缩放的状态下生效，表示鼠标滚轮滚动能将画面缩小的最小限度

- 当勾选最小值的配置为【默认】时，使用系统默认的最小值（0.01）
- 当勾【画面初始】时，此时，使用页面运行态刚打开时的初始大小作为最小值
- 当勾选【自定义】时，使用右侧输入框内的数值作为最小值

最大值
在开启缩放的状态下生效，为鼠标滚轮滚动能将画面缩小的最大限度

- 当勾选最小值的配置为【默认】时，使用系统默认的最大值（20）
- 当勾【画面初始】时，此时，使用页面运行态刚打开时的初始大小作为最大值
- 当勾选【自定义】时，使用右侧输入框内的数值作为最大值

注：缩放配置的自定义最大值为20，最小值为0.01

节流
应用场景：需要经常缩放的复杂页面

开启后，在执行滚动鼠标操作时，在指定的时间内，只会触发一次鼠标滚动事件，降低了页面渲染的次数。

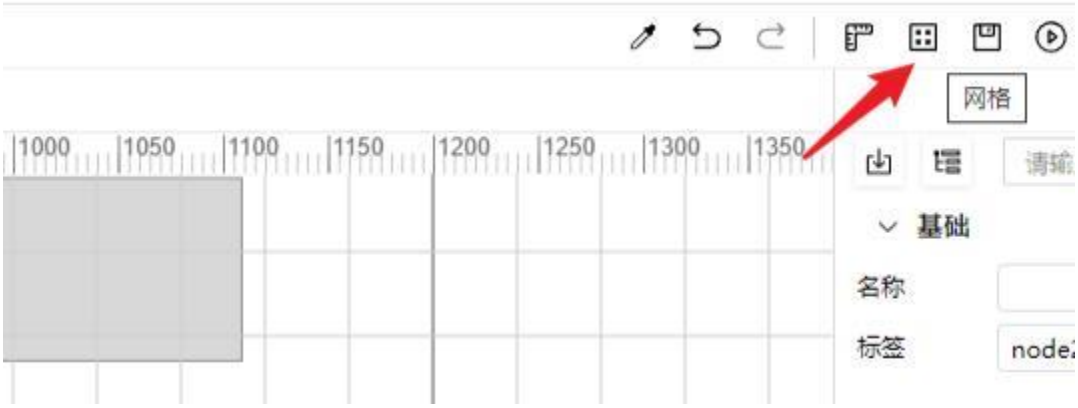
- 当节流为0ms时，认为是关闭状态
- 当节流值不为0ms时，例如100ms，那么表示，滚动鼠标时每100ms，执行一次滚动鼠标触发的页面缩放

5.22 画笔绘制

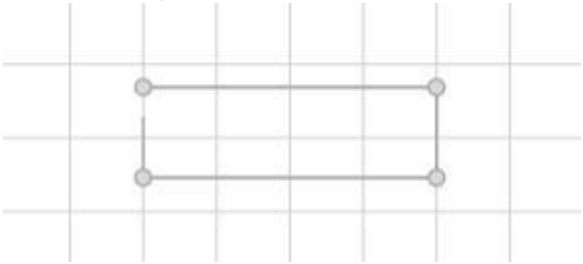
功能入口

绘制图形

- 打开辅助网格

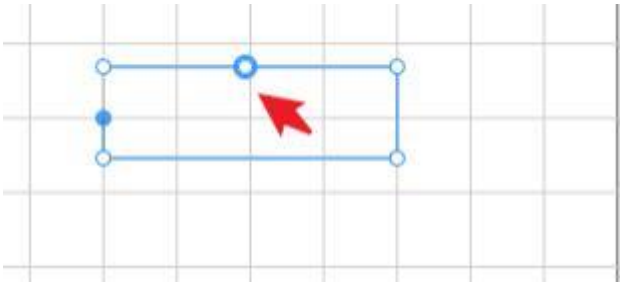


•点击画笔-选择起点-点击（双击可结束绘制）



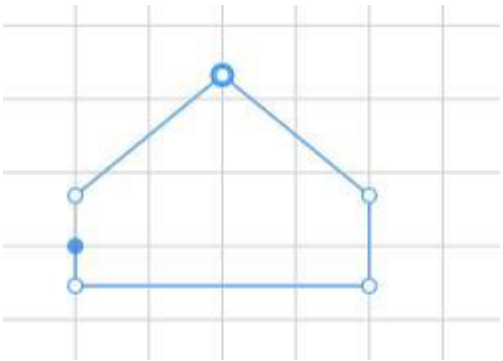
•绘制好的图形后，选中元素，双击元素线条，可进入调整

1. 选择调整点，可进行拖拽调整

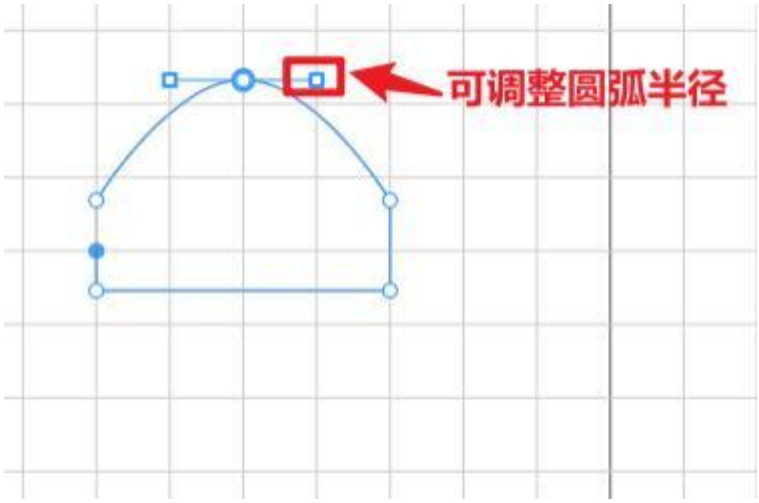


2. 选择控制点类型

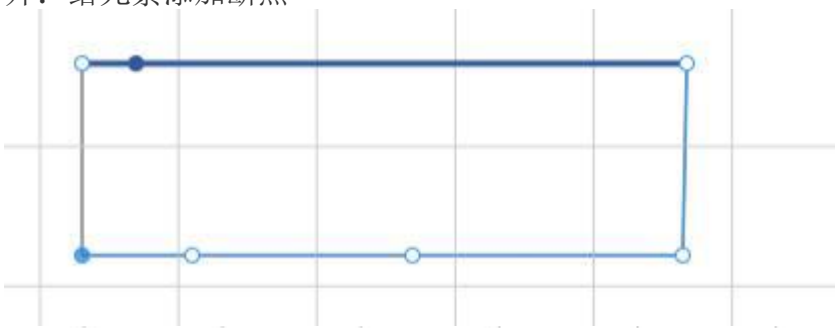
•直线：拖动断点形成直线

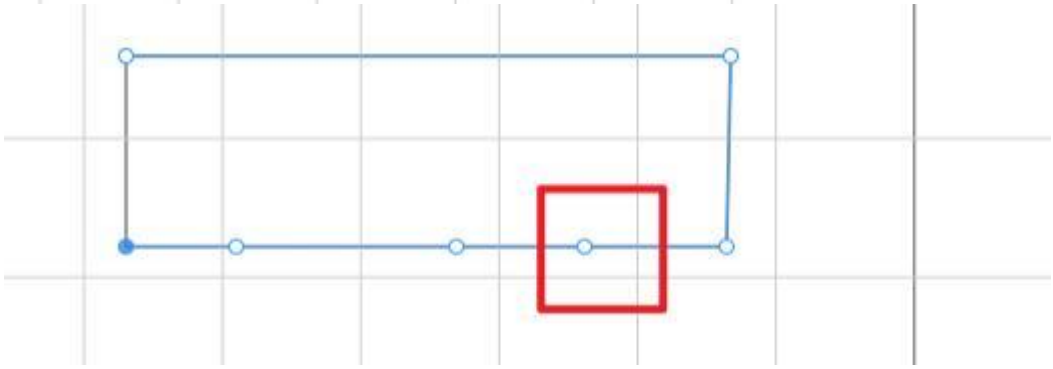


•镜像：拖动断点，形成圆弧

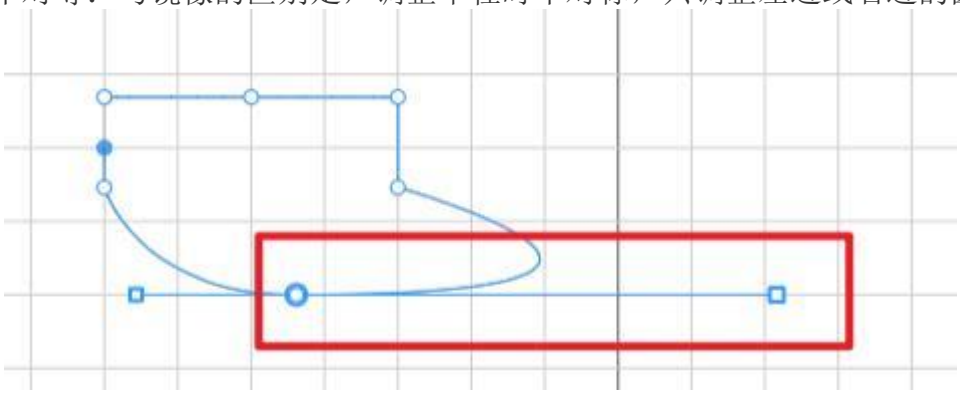


•断开：给元素添加断点



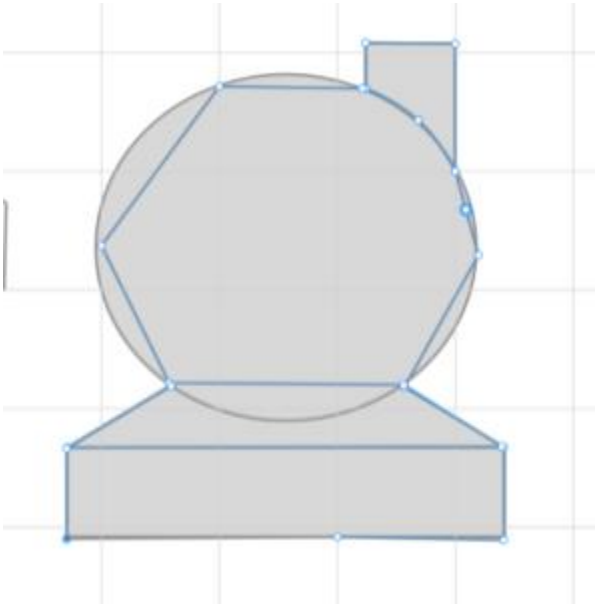


•不对等：与镜像的区别是，调整半径时不对称，只调整左边或右边的圆弧

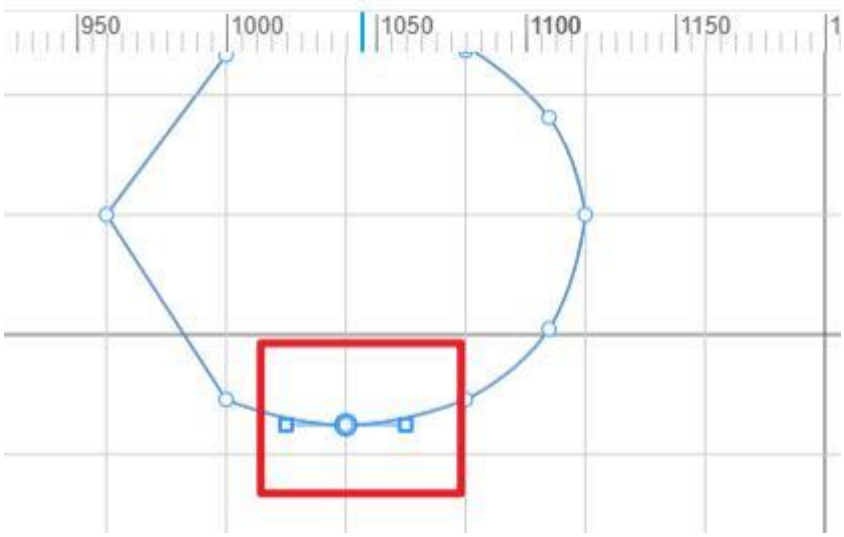


示例：

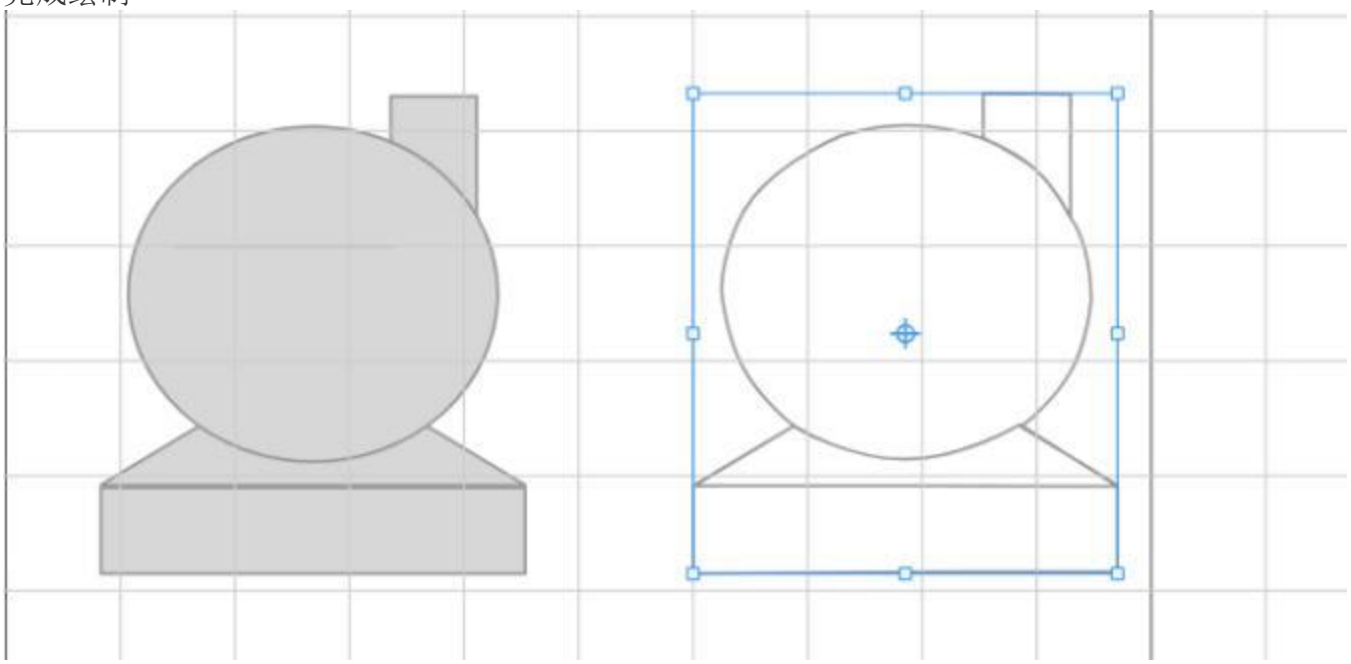
我们将图片置为底图方便临摹



利用断点之间的线段，选择调整点+镜像，形成弧度



完成绘制

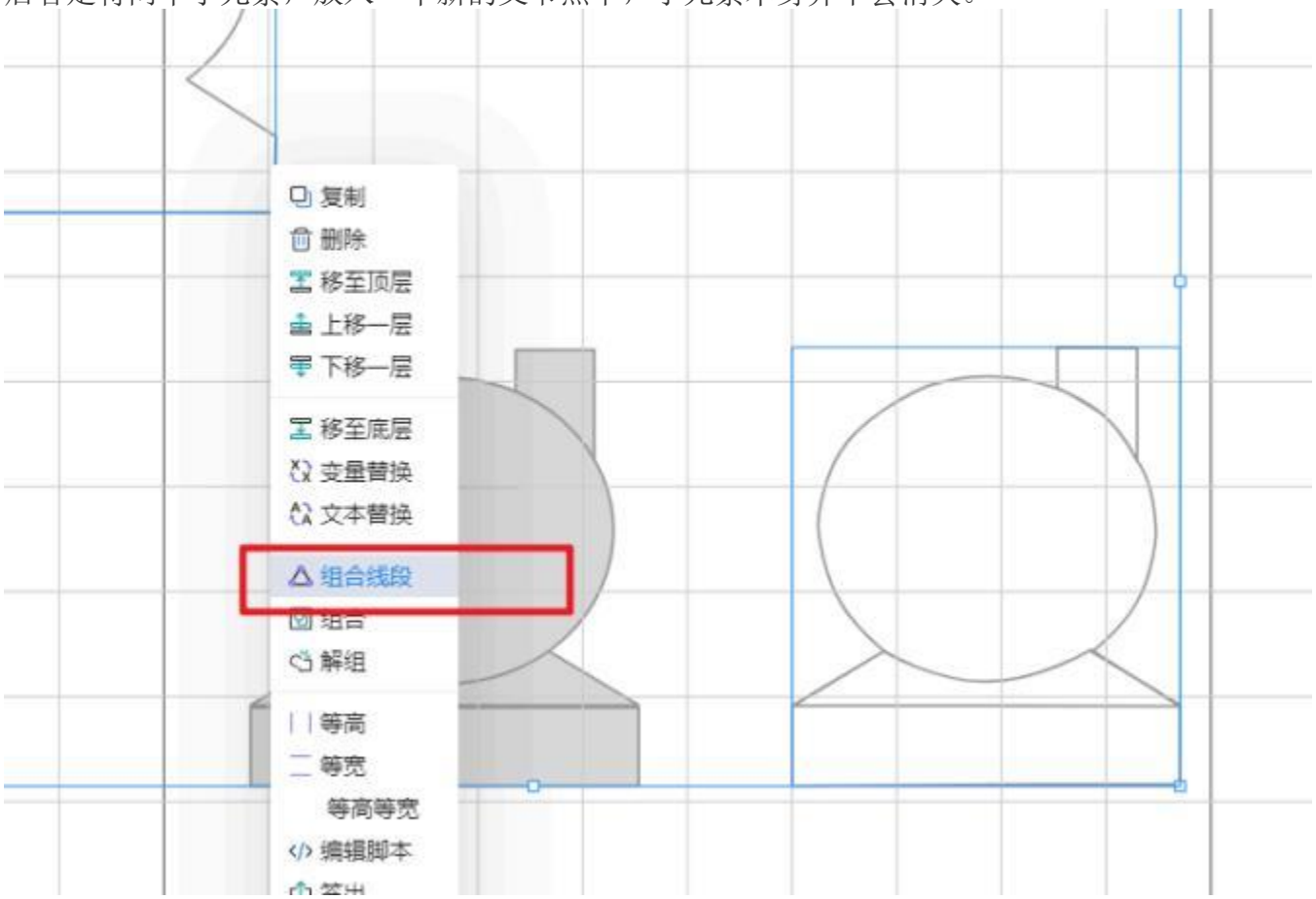


对于此类不规则的图形，掌握下边"线段组合"能更方便的制作

线段组合功能

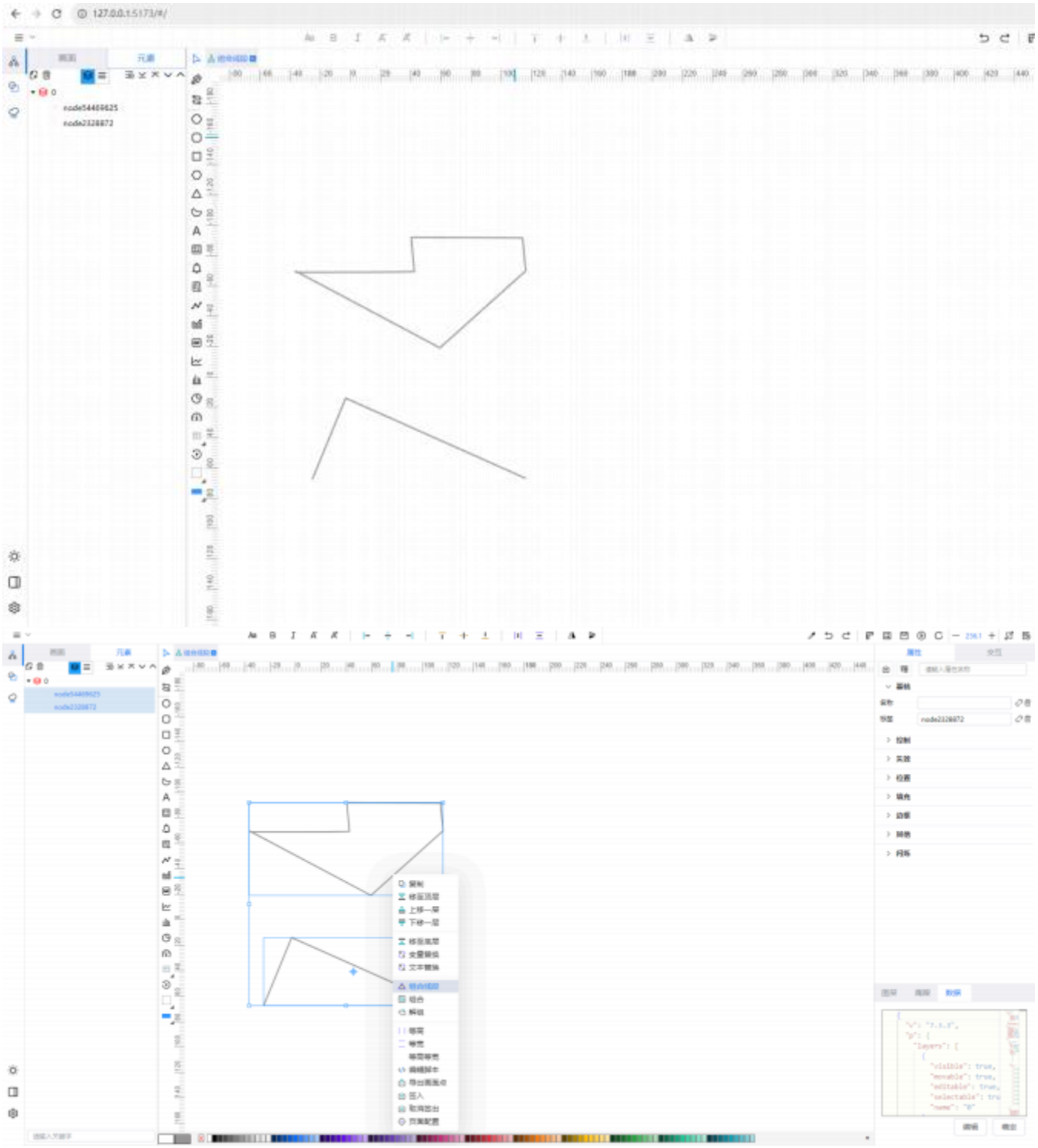
线段的组合与拆解，跟组合/解组是不一样的功能。

前者**会导致node元素个数的变化**，子元素会消失，会创建新的元素。或父元素消失，创建新的子元素。
后者是将两个子元素，放入一个新的父节点下，子元素本身并不会消失。

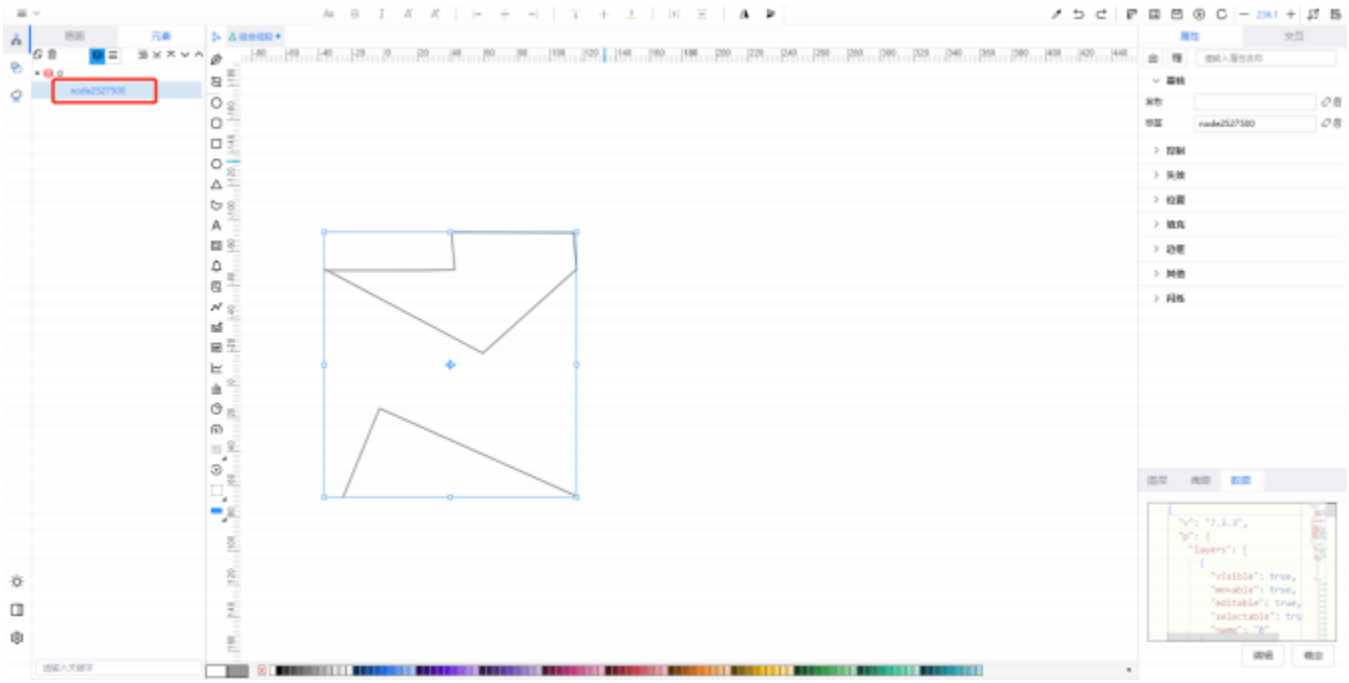


线段的组合

在画布上，使用钢笔工具，绘制出多个图形。选择这些图形后，然后点击鼠标右键，选择【组合线段】

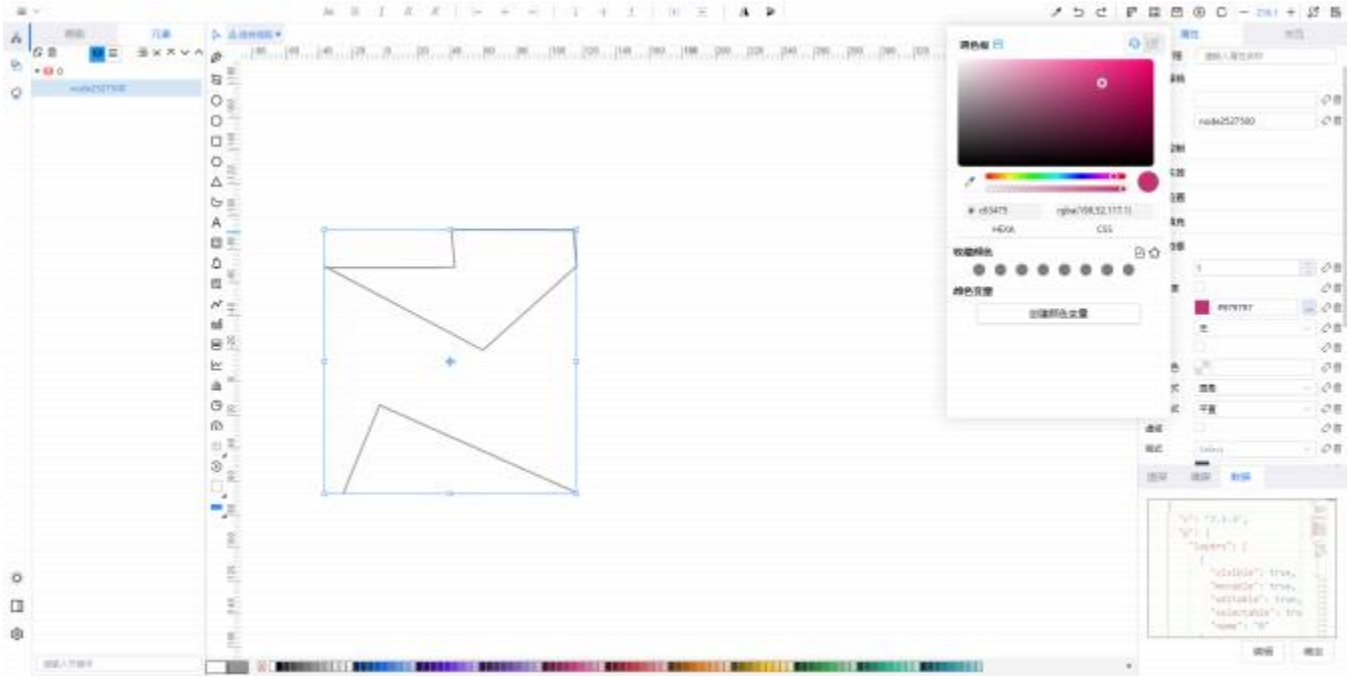


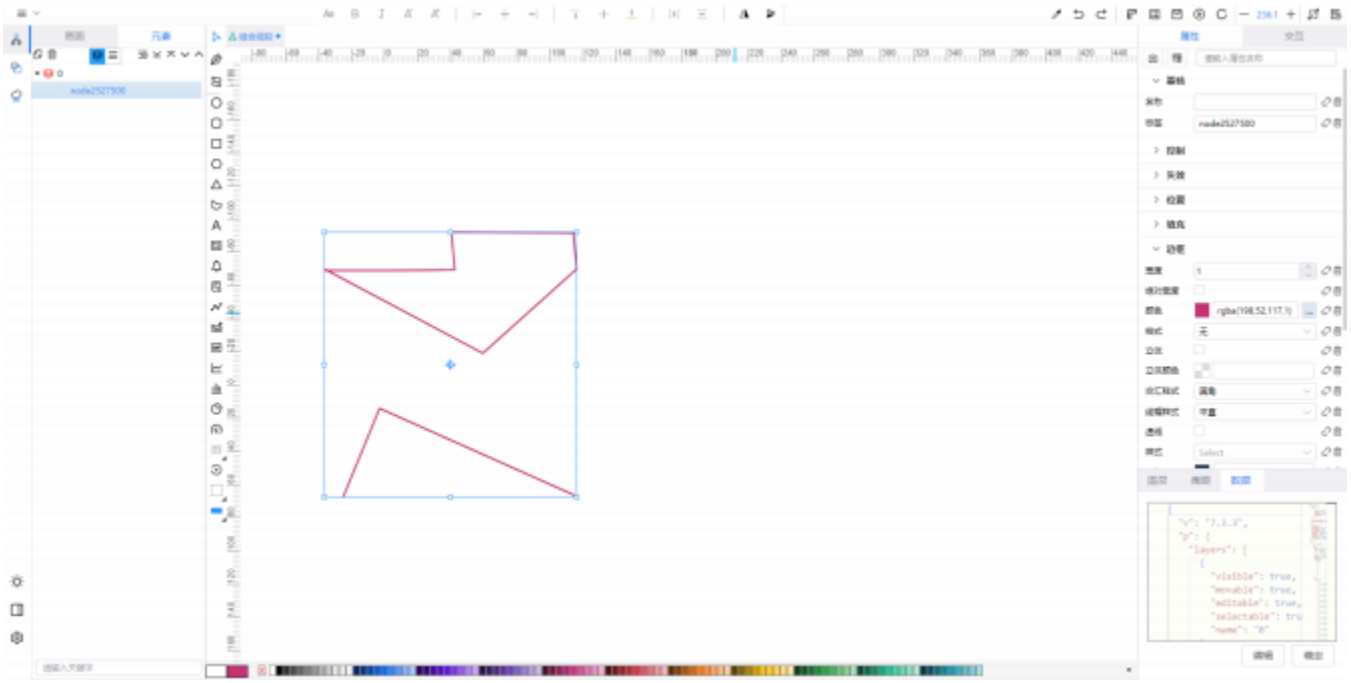
原本是两个node节点，组合后可以发现，变成了一个全新的节点



此时，修改这个元素的属性，会同时反应到被组合的图形上。

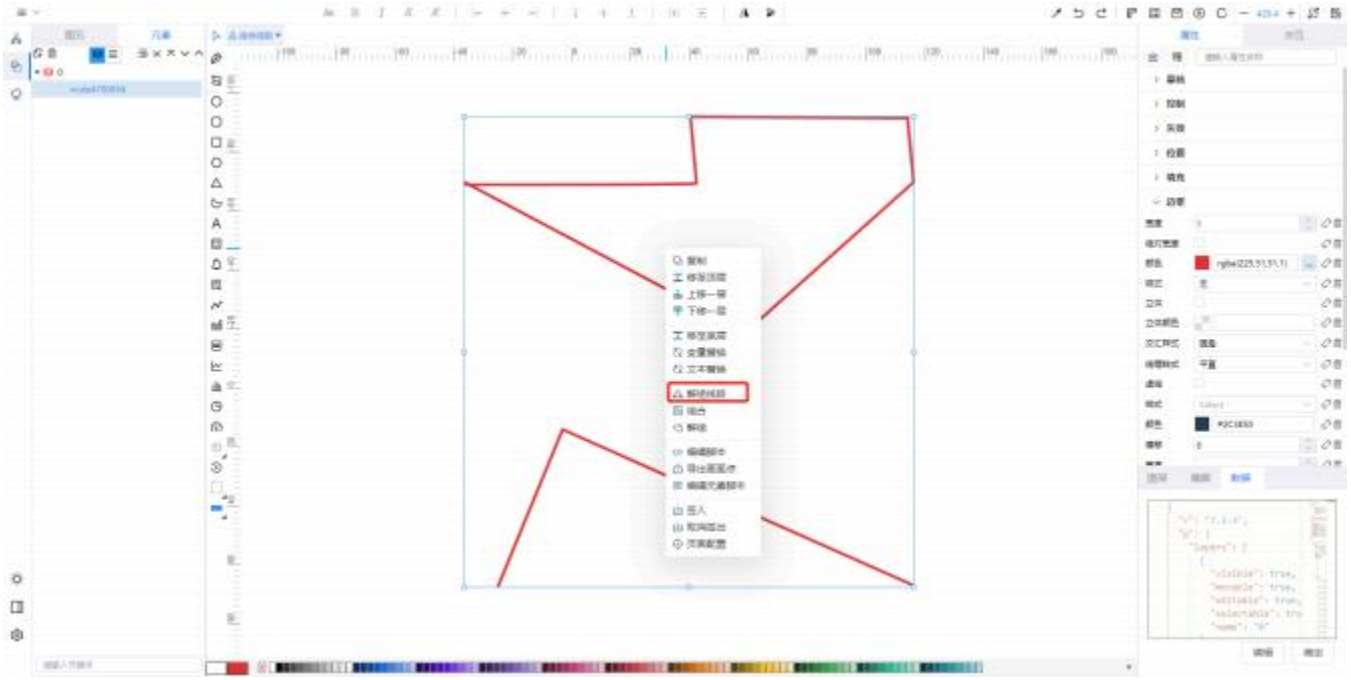
例如：修改边框的颜色，则原来的两个图形的颜色，都发生了变化





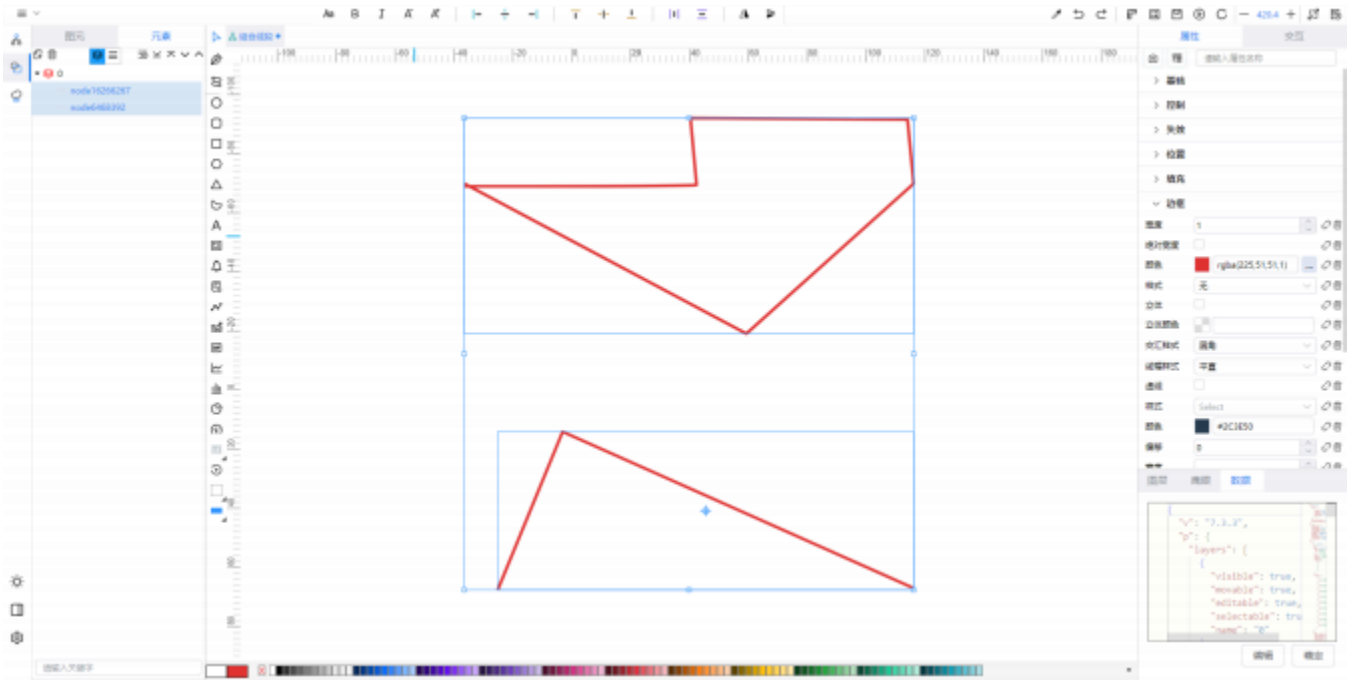
线段的解组

选择图形后，点击鼠标右键，然后选择【解组线段】



此时，会生成两个新的node节点

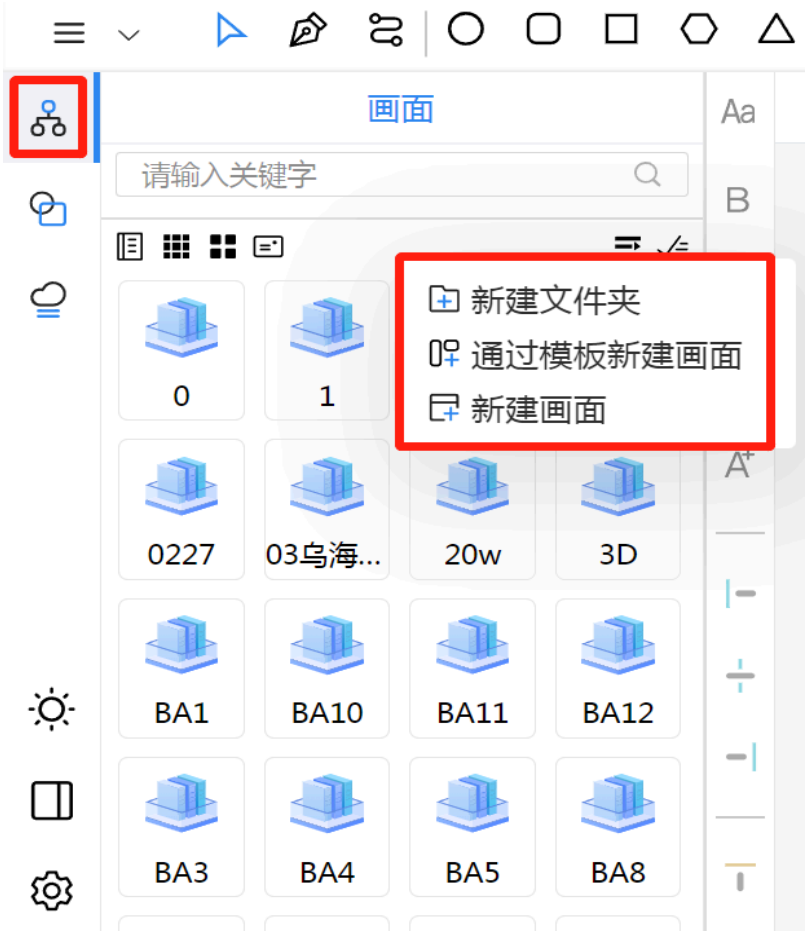
注：新生成的node节点会继承原的节点的属性



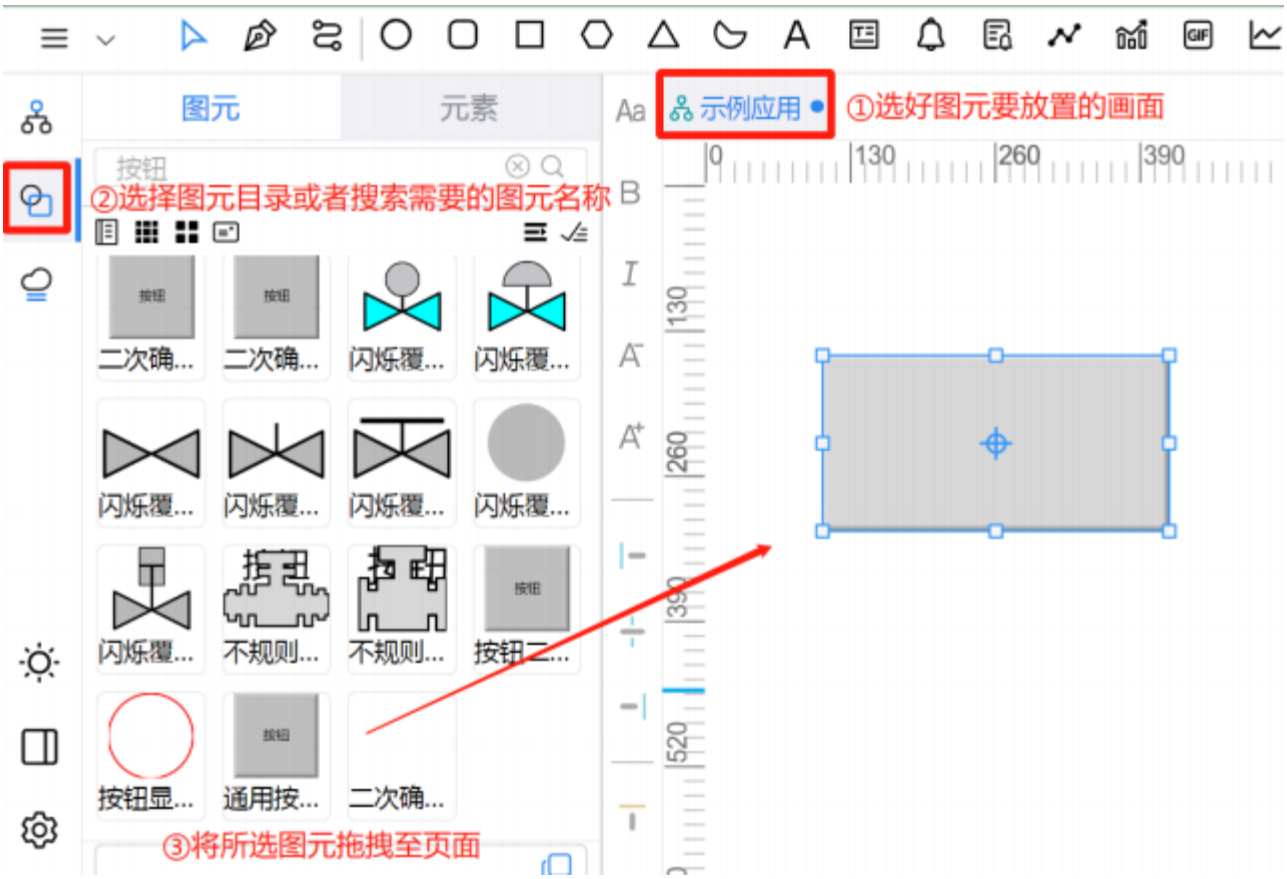
5.23 图元使用

步骤演示

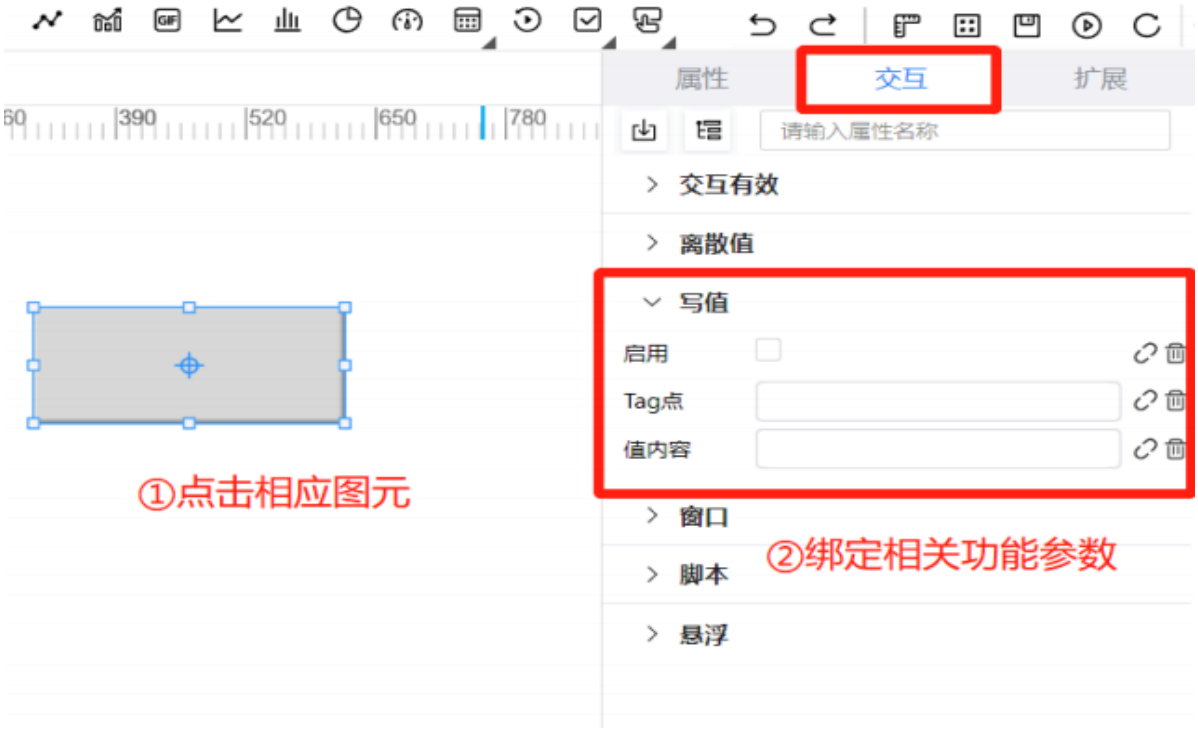
1. 进入2D设计器界面，选择画面选项卡，然后选择一个目录，右键新建画面，也可以根据需要新建目录。



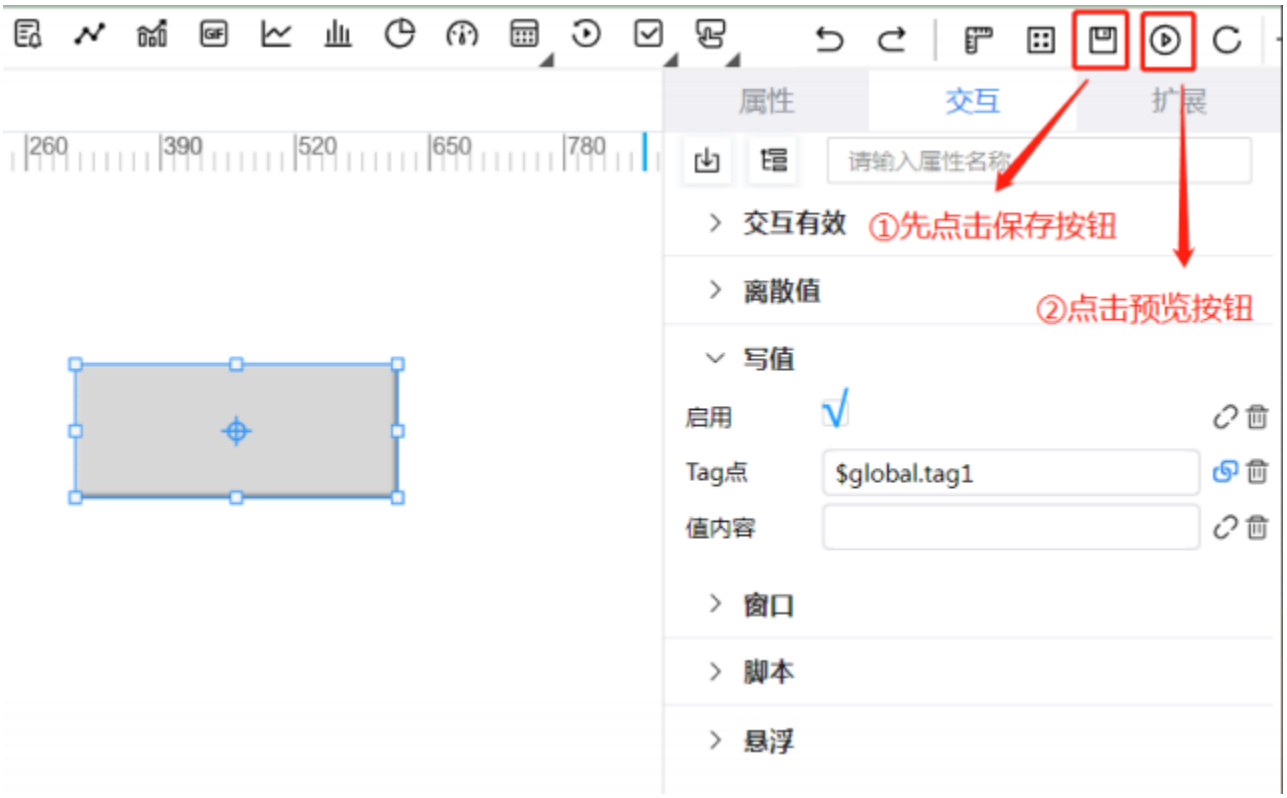
2. 在图元栏中选中IO域，拖拽到画面中。



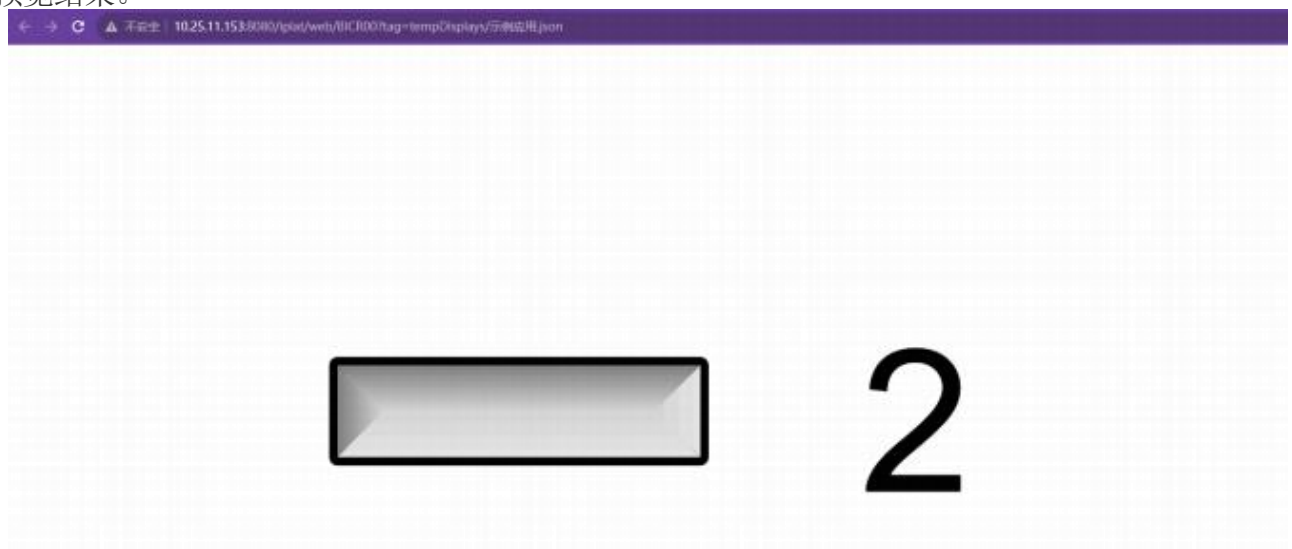
3. 点击画布中的图元，右侧属性窗口，点击"变量"属性的符号，选择一个点进行绑定。



4. 点击右上角菜单栏的"保存"按钮，提示"保存成功"后，再点击"预览"按钮，即可看到运行的预览画面。



5. 预览结果。



5.24 系统图元

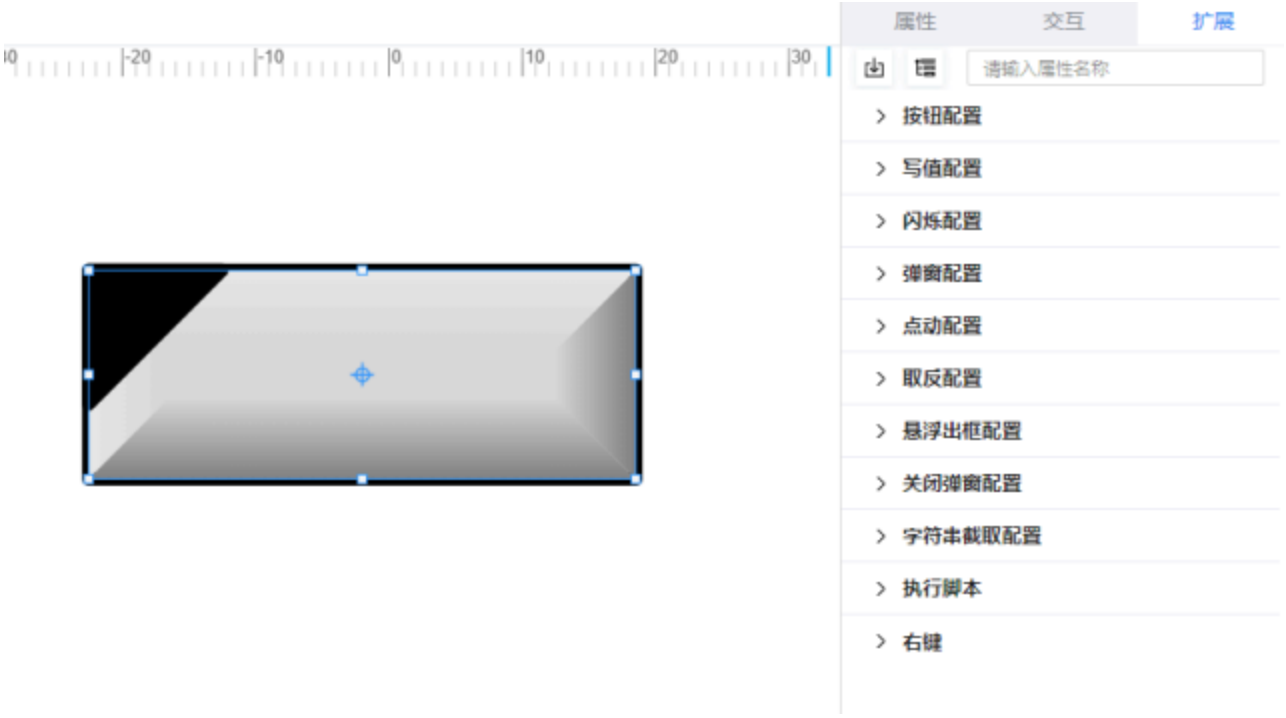
按钮图元

1. 概述

按钮图元目前分以下几种，通用按钮图元(属性基本一致，区别在于形状)、下拉框(comboBox)、单选框等。

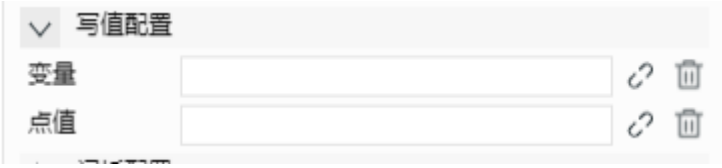
2. 通用按钮图元

2.1. 我们以【通用按钮(角标)】为例介绍通用按钮的使用。
2.2. 新建画面,拖入【通用按钮(角标)】图元;可以看到右侧扩展里有很多按钮专有的属性配置项。

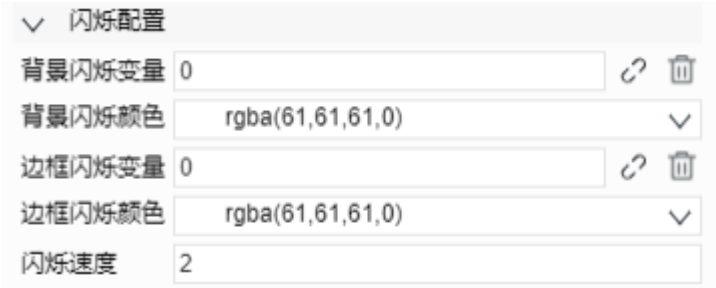


- 按钮配置
  - 提供按钮的文本,颜色,样式等常用配置;

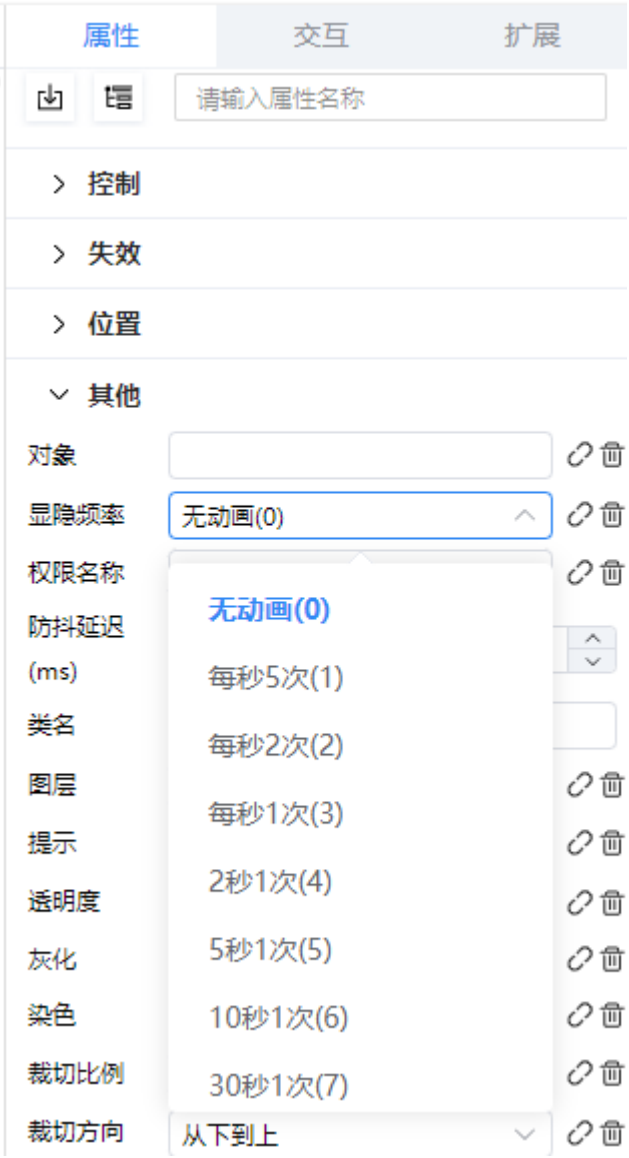
  - 按钮形状：可以修改按钮为15种基本形状
  - 图片填充：将按钮填充地址引用的图片
  - 文本：“文本内容”、“水平对齐”、“垂直对齐”、“字体样式”、“字体颜色”均可对文本进行样式内容修改
  - 默认背景颜色：编辑态时颜色、优先级比“背景颜色”低
  - 背景颜色：可创建匹配规则，例如0-红，1-绿等，与“背景变量”配合使用
  - 深度：可修改边框的深浅样式
  - 渐进：与“渐进色”配合使用，设置按钮样式
  - 圆角矩形半径：如果“按钮形状”是圆角矩形，可以改变圆角程度
- 写值配置
  - 用于给变量写值;
    - 【变量】属性绑定需要写值的点,【点值】属性可以直接输入需要写的值,也可绑定点或表达式;支持同时绑定多个点,用",″分开



- 闪烁配置
  - 提供按钮背景色和边框颜色闪烁配置；



  - 其中【闪烁速度】属性的值配置可以参考【显隐频率】属性配置项



- 弹窗配置
  - 提供弹窗配置选项，这些属性基本是sys.openPage弹窗函数的参数；

◦是否弹窗:勾选开启弹窗功能,否则不生效;
◦弹窗路径:弹窗画面的路径,编辑态选中画面后,在底部可以看到画面路径;
◦弹窗标题:弹窗标题栏内容
◦弹窗模式:有8种方式模式:
- ▪页面跳转
- ▪新页面跳转
- ▪弹出模态画面
- ▪弹出非模态画面
- ▪新打开一个同级的浏览器tab页
- ▪打开一个新窗口，可以指定大小6:非模态且默认新开页面原大小,此时【弹窗宽度/高度】属性不生效;7：快速弹窗，一般使用在对操作速度有要求的场景,弹窗里的所有sys的函数需要使用view去调用，例如view.sys.setAttribute;

◦弹窗宽度/高度:弹窗大小,直接填数字,单位是像素;
◦弹窗位置:默认跟随鼠标,参数值为["event.x","event.y"];也可以直接填坐标,例如[500,300];
◦弹窗ID:用于关闭画面;仅支持英文和数字,不能用中文,特殊符号,空格;关闭弹窗函数:sys.closePage("弹窗ID");
◦原变量/替换的变量:用于变量名替换;本质是绑定属性里的字符串替换;例如原来绑定变量是"*global.test*3";原变量填"3",替换的变量填"1",打开弹窗后,变量"global.test3"会被替换成"$global.test1";多变量替换,用";"隔开,原变量和替换的变量一一对应,示例如下

◦弹窗传参:父页面传递给子弹窗的对象;例如{"tag1":"1","tag2":"2"}

◦相对top:弹框是否在所有画面最外层，默认为true，只对modal为2和3生效;

◦是否需要标题栏:默认勾选，有标题栏

◦是否需要关闭按钮：默认勾选，有弹窗关闭按钮

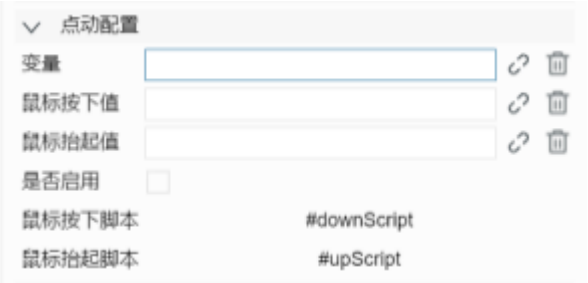
◦是否需要弹窗边框：默认勾选，可选是否需要弹窗边框

◦关闭回调:回调函数,关闭页面时触发

◦openChild:传入子弹窗的函数;可以在子弹窗中调用;调用方式:window.parent.parent.frames[0].openChild();

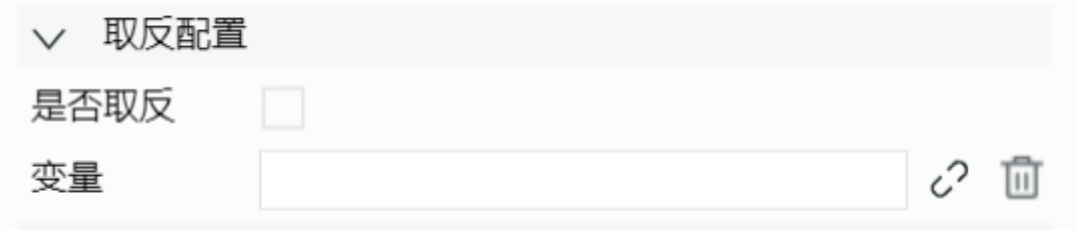
•点动配置

◦用于配置鼠标左键按下和抬起时的操作;【是否启用】勾选时,才会执行下面的【鼠标按下脚本】/【鼠标抬起脚本】;



•取反配置

◦对DI变量进行0-1取反;需要勾选【是否取反】才生效;



•悬浮出框配置

◦鼠标移到按钮上时,显示边框,可以配置边框颜色和宽度

•关闭弹窗配置

◦用于点击按钮时关闭指定弹窗ID的弹窗;

•字符串截取配置

◦用于将截取字符串并用截取结果替换子弹窗里的占位符;其中变量属性值需要是字符串类型;可以直接填字符串,也可以绑定文本型变量,或绑定变量的描述属性等

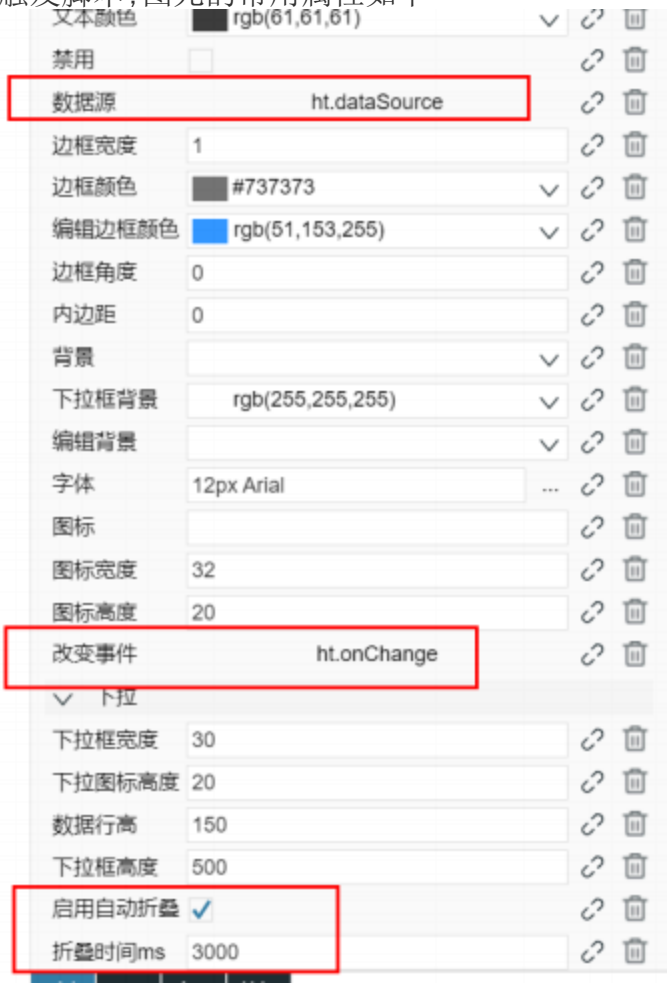
•执行脚本

◦勾选启用后,点击按钮,会执行#doScript里的脚本

- 右键
  - 同点动配置，值不过从左键改为右键触发

3. 下拉框

- 【下拉框】图元提供下拉选择功能，并且可以触发脚本；图元的常用属性如下



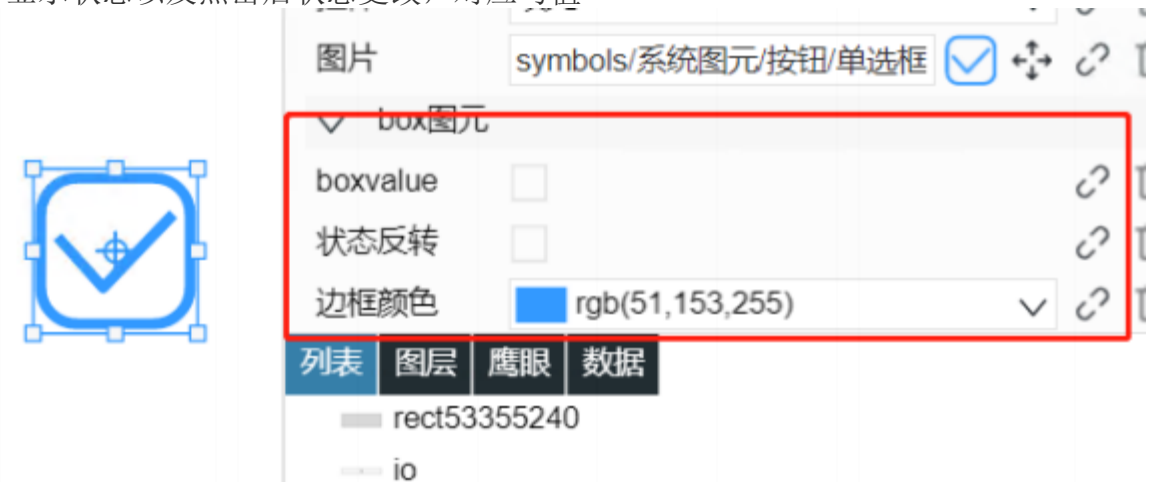
- 【数据源】属性填下拉选项的内容，格式是对象数组；
- 【改变事件】属性里写下拉选项触发的脚本；
- 【启动自动折叠】设置鼠标离开下拉框后是否自动折叠；
- 【折叠时间】设置下拉框自动折叠的时间；
- 其他属性名字比较容易理解，可以自行尝试效果；

4. 标签单选框

- 本身提供标签单选框的功能，额外提供事件改变功能，可以监测值从未选中变成可选中的情况，从而触发自定事件

5. 单选框

- 参数：
  - boxvalue:需要绑定的tag点，控制是否勾选
  - 状态反转:未勾选时：tag点值是1显示勾选状态，是0显示未勾选状态；勾选后：tag点值是1显示未勾选状态，是0显示勾选状态
  - 边框颜色：控制单选框颜色
  - 功能：显示状态以及点击后状态更改，对应写值



文本图元

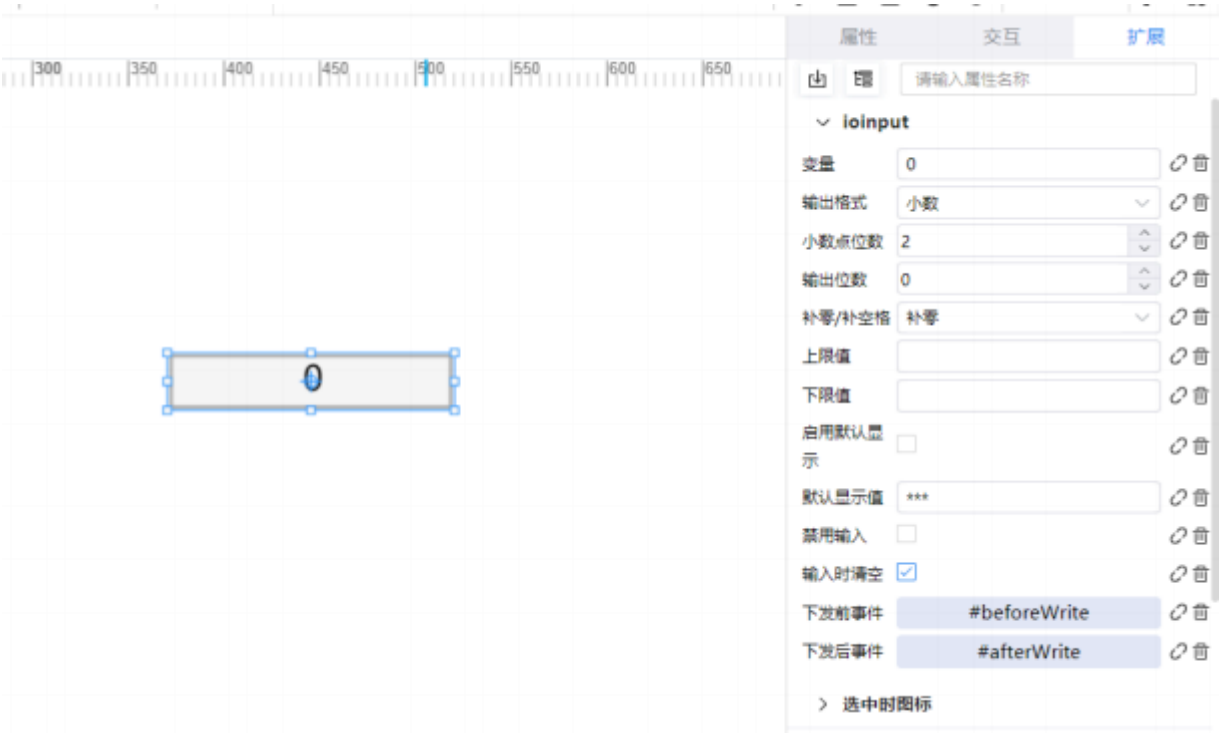
1. IO域（角标IO域可参考IO域）

1.1 概述

实现现场使用IO域控制和显示为同一个点值，和IO域（两变量）不同，这里显示和控制下发为同一个点

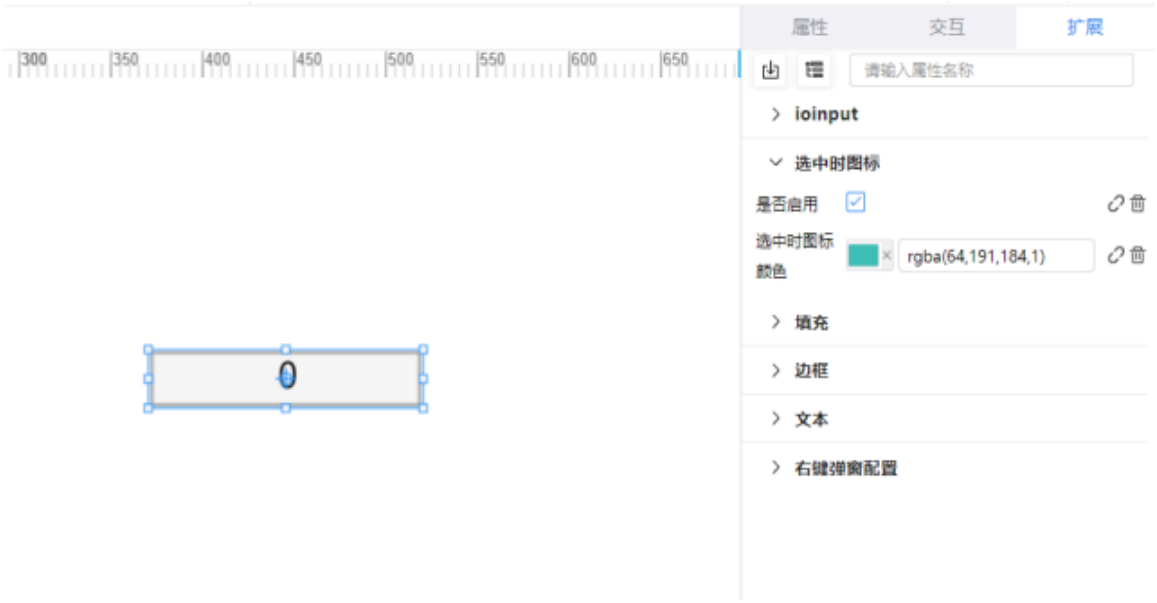
1.2 基本功能

- 首先在系统图元中找到IO域--```路径（symbols/系统图元/文本/IO域.json），拉取到画面中，含有以下配置项



- 变量：是指可通过IO域编辑控制的变量，并会显示
- 输出格式：是指IO域输入时可以输入或者控制的变量的格式有（字符串，小数，十进制，二进制，八进制，十六进制，指数）
- 小数点位数：是指数值精确到小数点后几位
- 输出位数：是指整体位置，比如点值是12，精确到小数点后2位，输出位数设置6位，那么就会出现012.00
- 补零/补空格：是指比如上方输出位数012.00则是选择的补零，如果是补空格则会是12.00
- 上限值/下限值：是指这个变量可控制的一个范围，为全闭，比如上限设置为10，下限设置为1，那么tag点可控制的区间为1<=tag<=10，当超过或者低于这个区间时会提示
- 启用默认显示：可选择在输入超过上下限时是否需要默认显示
- 默认显示值：可设置默认显示值
- 禁用输入：当需要条件判断禁用输入时（例如权限不够，不能执行控制下发），在后面类似连接按钮点击绑上需要判断的点或表达式即可
- 输入时背景色/编辑时边框色：当点击IO域准备进行控制时，背景色和边框色会对应配置进行更改
- 输入时清空：勾选后点击io域聚焦后会清空原来内容重新输入，不勾选则保留原来内容并在聚焦时全选所有内容
- 下发前事件：如需要在写值之前做一些处理则可以将处理脚本写在这里，比如写入字符串时，先将字符串的长度写入到另一个点
- 下发后事件：如需要在写值后做一些处理则可以将处理脚本写在此处。

1.3 选中时图标

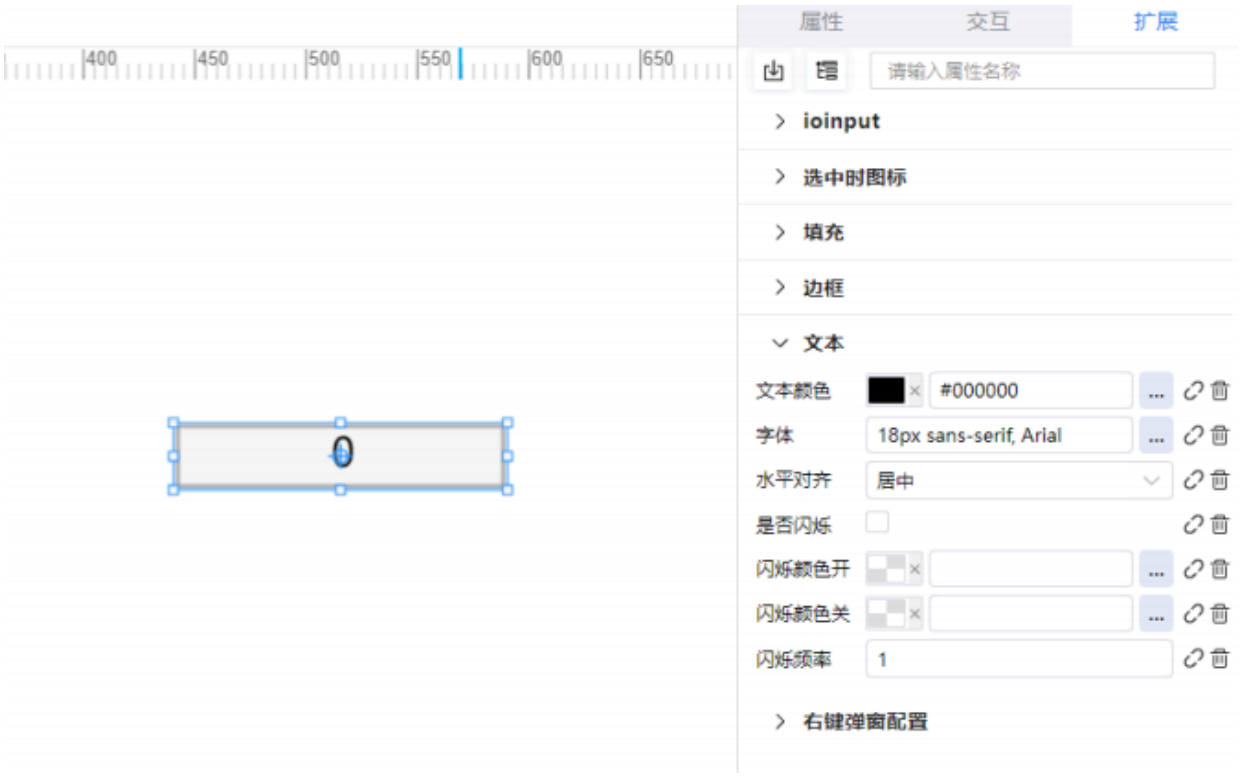


- 是否启用：勾选此项，在运行态时，io域图元若使用回车输入，焦点切换时，被选中的io域左上框将覆盖一条线段，提示当前操作的io域
- 选中时图标颜色：可配置左上框提示线段颜色

1.4 填充及边框

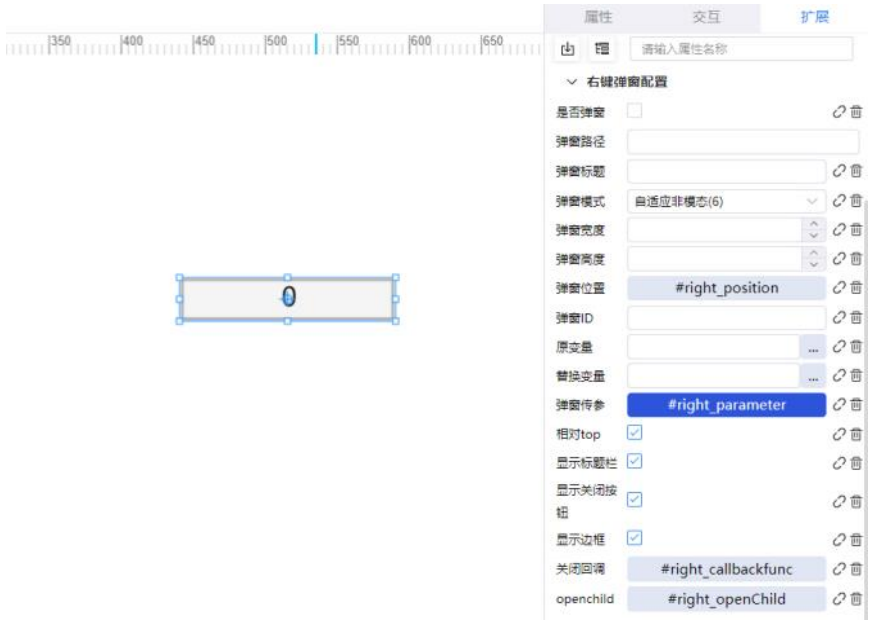
- 背景色：背景色为IO域的背景颜色，根据需要可进行调整
- 是否闪烁：可选择文本框是否闪烁
- 闪烁颜色开/关：不配置匹配规则时，默认闪烁颜色，若只配置了“闪烁颜色开”，则在闪烁条件满足时，将与元素背景色交替闪烁；若配置“闪烁颜色关”（修改默认颜色或绑点都视为配置过此项），若闪烁条件满足时，则“闪烁颜色开/关”两个颜色交替闪烁
- 闪烁频率：文本框闪烁速度
- 闪烁详细配置可参考：元素闪烁配置
- 编辑背景色：运行态编辑文本框时的背景颜色
- 边框颜色：是指IO域边框的颜色，根据需要可进行调整
- 边框宽度：调整边框的宽度，根据需要可进行调整
- 编辑边框色：运行态编辑时的边框的颜色

1.5 文本



- 文本字体：则是点值显示出来的字体，根据需要可进行调整
- 文本颜色：对应点值显示出的颜色，根据需要可进行调整
- 水平对齐：控制值显示的水平位置，分（左对齐，居中，右对齐）
- 是否闪烁：可选择文本是否闪烁
- 闪烁颜色开/关：文本不配置匹配规则时，默认闪烁颜色，若只配置了“闪烁颜色开”，则在闪烁条件满足时，将与文本默认色交替闪烁；-若配置“闪烁颜色关”（修改默认颜色或绑点都视为配置过此项），若闪烁条件满足时，则“闪烁颜色开/关”两个文本颜色交替闪烁
- 闪烁频率：文本框内文本的闪烁速度
- 编辑字体：运行态编辑时的文本字体
- 编辑文本色：运行态编辑时的文本颜色

1.6 右键弹窗配置



- 是否弹窗:勾选开启右键弹窗功能,否则不生效;
- 弹窗路径:弹窗画面的路径,编辑态选中画面后,在底部可以看到画面路径;

- 弹窗标题:弹窗标题栏内容
- 弹窗模式:有8种方式模式:
  - 页面跳转
  - 新页面跳转
  - 弹出模态画面
  - 弹出非模态画面
  - 新打开一个同级的浏览器tab页
  - 打开一个新窗口，可以指定大小6:非模态且默认新开页面原大小,此时【弹窗宽度/高度】属性不生效;7：快速弹窗，一般使用在对操作速度有要求的场景,弹窗里的所有sys的函数需要使用view去调用，例如view.sys.setAttribute;
- 弹窗宽度/高度:弹窗大小,直接填数字,单位是像素;
- 弹窗位置:默认跟随鼠标,参数值为["event.x","event.y"];也可以直接填坐标,例如[500,300];
- 弹窗ID:用于关闭画面;仅支持英文和数字,不能用中文,特殊符号,空格;关闭弹窗函数:sys.closePage("弹窗ID");
- 原变量/替换的变量:用于变量名替换;本质是绑定属性里的字符串替换;例如原来绑定变量是"*global.test*3";原变量填"3",替换的变量填"1",打开弹窗后,变量"global.test3"会被替换成"$global.test1";多变量替换,用";"隔开,原变量和替换的变量一一对应
- 弹窗传参:父页面传递给子弹窗的对象;例如{"tag1":"1","tag2":"2"}
- 相对top:弹框是否在所有画面最外层，默认为true，只对modal为2和3生效;
- 是否需要标题栏:默认勾选，有标题栏
- 是否需要关闭按钮：默认勾选，有弹窗关闭按钮
- 是否需要弹窗边框：默认勾选，可选是否需要弹窗边框
- 关闭回调:回调函数,关闭页面时触发
- openChild:传入子弹窗的函数;可以在子弹窗中调用;调用方式:window.parent.parent.frames[0].openChild();

1.7 如何使用

将图元拉取至画面中，在变量绑定对应的tag点，其他配置根据相关需求进行更改。在变量绑点之后，支持多个io域之间箭头切换。

1.8 格式化（format）说明

- 什么是格式化（format）?
  - 格式化（format）是io域图元的一个特性，可以根据输出格式、小数点位数和输出位数等条件，在运行态将绑定的变量值做特殊的显示。注意，format只会修改显示的效果，而不会更改元素实际值
- 为什么经过格式化后的显示的数据，data.a('#cvs.value')拿到的value，与显示状态不同?
  - format只会修改显示的效果，而不会更改元素实际值,因此会出现显示状态不同的情况。
- 为什么io域输出格式为字符串时，直接通过data.a('#cvs.value',value),显示状态没有修改?
  - io域输出格式为字符串时，显示为绑点中tagMap内存中的值，因此当tagMap中值没有发生改变时，data.a('#cvs.value',value)将不会修改节点的显示状态。
- 为什么文本控件和io域控件（输出格式为字符串），绑定同一个点，同时输入068时，文本显示为68，io域显示为068
  - 对于文本控件和io域控件，输入068，在组态后台逻辑中，都被处理为68记录在节点上，因此分别通过data.a('#cvs.value')获得的value都是68；但tagMap中记录的是原始推送值，而io域控件输出格式为字符串时，其显示值为tagMap中的原始值，因此io域显示为068

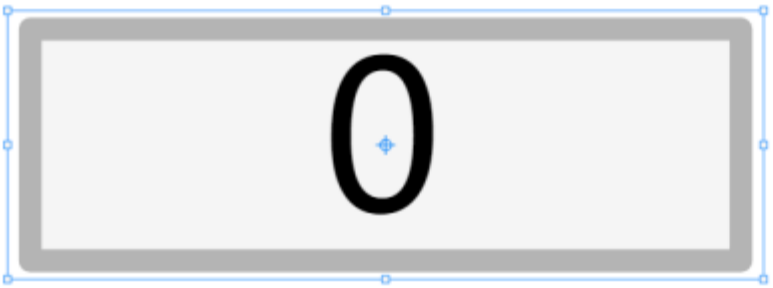
2. IO域（两变量）

2.1 概述

实现现场使用IO域控制和显示不为用一个点值

2.2 基本功能

- 首先在系统图元中找到IO域（两变量）--```路径（symbols/系统图元/文本/IO域(两变量).json），拉取到画面中，含有以下配置项



- 控制变量————是指可通过IO域编辑控制的变量
- 变量————是指显示的变量，只做显示，不做控制
- 输出格式————是指IO域输入时可以输入或者控制的变量的格式有（字符串，小数，十进制，二进制，八进制，十六进制，指数）
- 小数点位数————是指数值精确到小数点后几位
- 输出位数————是指整体位置，比如点值是12，精确到小数点后2位，输出位数设置6位，那么就会出现012.00
- 补零/补空格————是指比如上方输出位数012.00则是选择的补零，如果是补空格则会是12.00
- 上限值/下限值————是指这个变量可控制的一个范围，为全闭，比如上限设置为10，下限设置为1，那么tag点可控制的区间为1<=tag<=10，当超过或者低于这个区间时会提示
- 背景色————背景色为IO域的背景颜色，根据需要可进行调整
- 文本字体————则是点值显示出来的字体，根据需要可进行调整
- 文本颜色————对应点值显示出的颜色，根据需要可进行调整
- 水平对齐————控制值显示的水平位置，分（左对齐，居中，右对齐）
- 边框颜色————是指IO域边框的颜色，根据需要可进行调整
- 边框宽度————调整边框的宽度，根据需要可进行调整
- 禁用输入————当需要条件判断禁用输入时（例如权限不够，不能执行控制下发），在后面类似连接按钮点击绑上需要判断的点或表达式即可
- 输入时背景色/编辑时边框色————当点击IO域准备进行控制时，背景色和边框色会对应配置进行更改
- 编辑时边框色————运行态输入时，所显示边框颜色

•输入时清空————运行态输入时，是否清空目前显示值去编辑（不影响实际值），默认清空。

2.3 如何使用

•将图元拉取至画面中，在控制变量绑定需要控制下发的tag点，在变量绑定需要显示的tag点，其他配置根据相关需求进行更改

3.其他图元

3.1 文本区

•主要事件功能：

属性	功能
按键按下事件	鼠标按下后触发自定义事件
按键松开事件	鼠标松开后触发自定义事件
改变事件	文本区内容改变触发事件
其余属性	控制颜色，形状大小

3.2 文本框

•拥有通用按钮功能，此外还增加文本功能

属性	功能
文本类型	根据实际值选择
输出位数	控制显示的位数
小数点位数	显示小数时，小数点的后的位数
补充字符	两者结合使用，输出位数大于实际位数
输出位数	天剑补充字符的第一个字符内容
千分位	数据过千时，以逗号作为分割如1000显示1,000
其余文本属性	颜色、背景

3.3 输入框

属性	功能
改变事件	文本区内容改变触发事件
按键松开事件	鼠标松开后触发自定义事件
按键按下事件	鼠标按下后触发自定义事件
失焦事件	失去焦点时触发事件
其余属性	控制颜色，形状大小

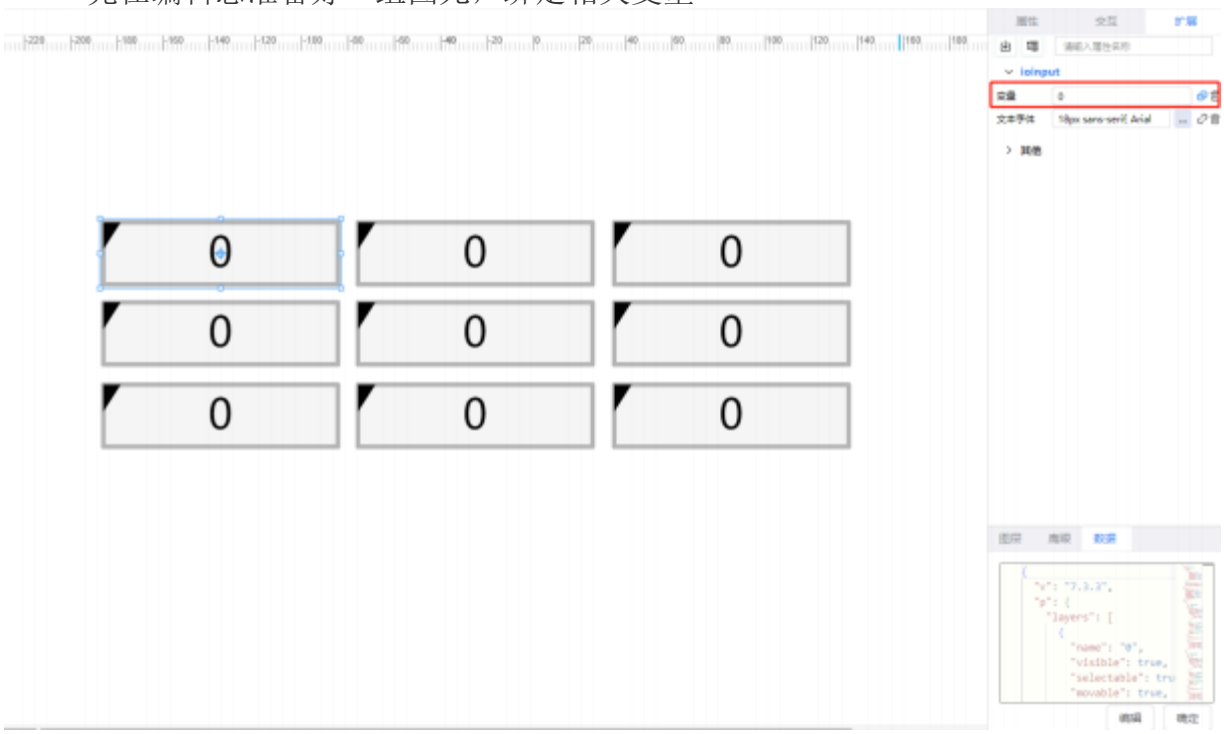
4. 可移动焦点IO域

4.1. 概述

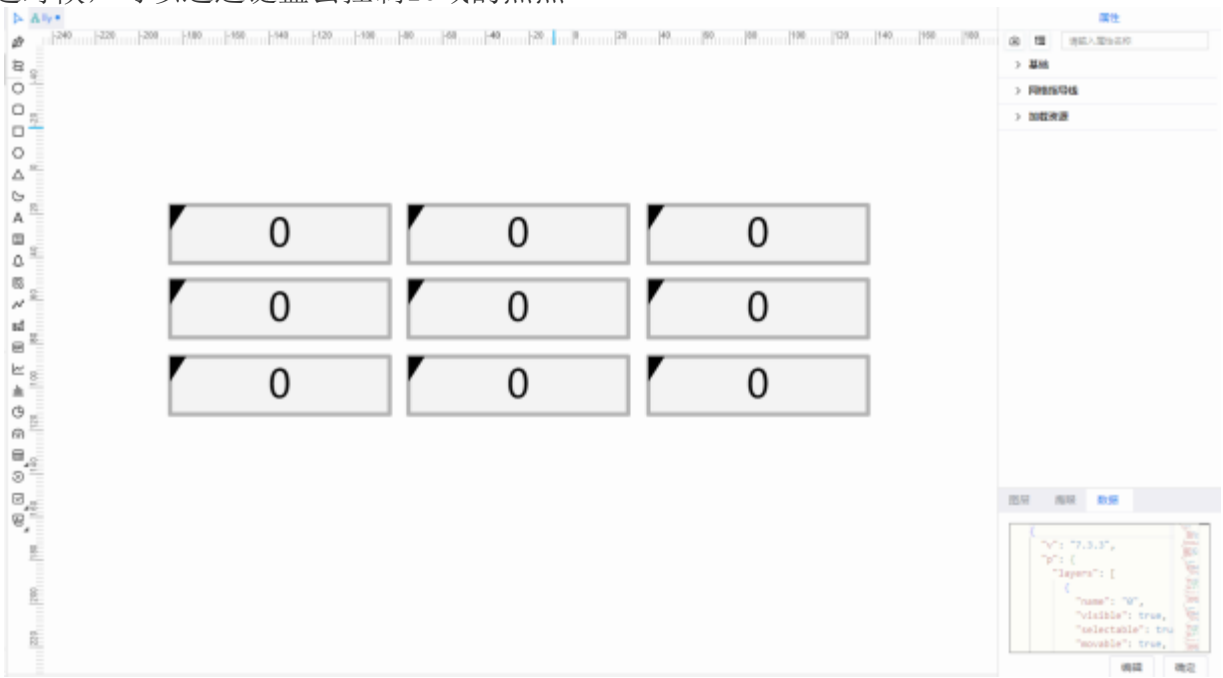
- 为了方便用户在多IO域的画面进行输入操作而制作。用户可以通过方向键去控制焦点，然后完成输入操作。

4.2. 基本功能

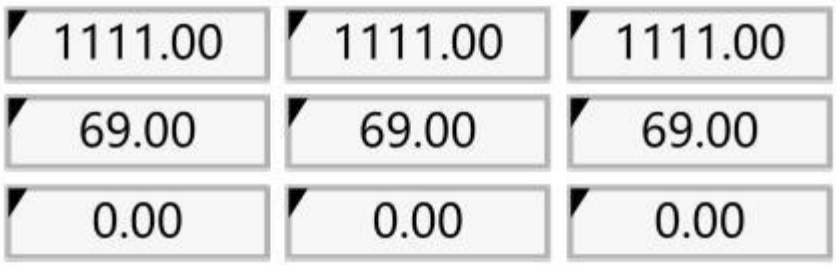
- 先在编辑态准备好一组图元，绑定相关变量



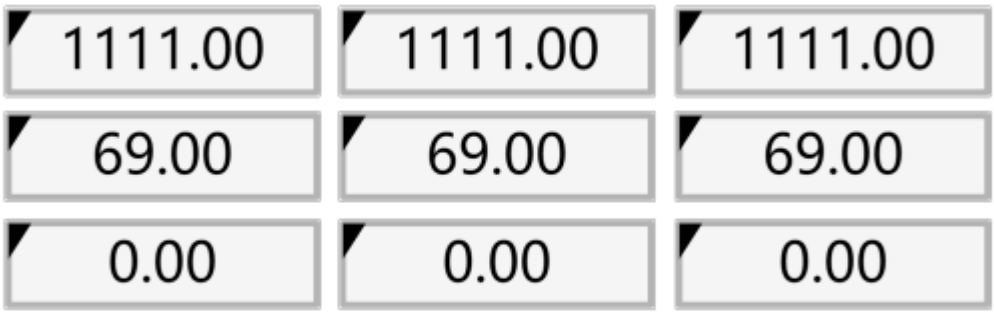
- 这时候，可以通过键盘去控制IO域的焦点



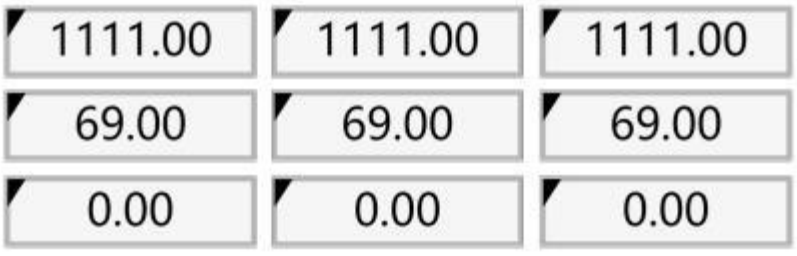
- 然后预览进入运行态，可通过键盘去控制IO域



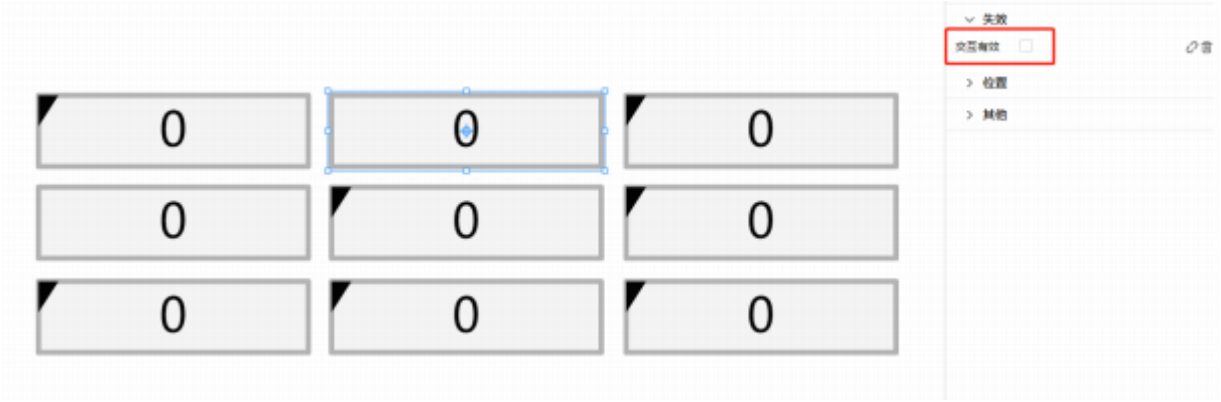
- 这里第一行绑的同一个点（A），第二行绑的同一个点（B），第三行没绑点，(注：虽然没有绑点的也能够上下左右移动，但是一般推荐先给“变量”（即"#cvs.value"）绑点后使用)



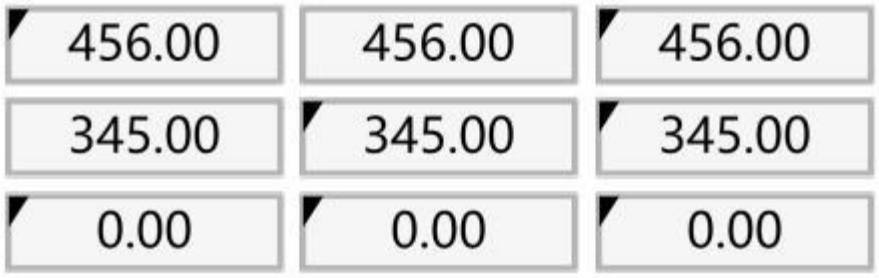
- 下发，在输入完毕后，然Enter键即可完成下发，绑了相同点的IO域的数据会同步。若没有绑点的图元则不会去进行下发。



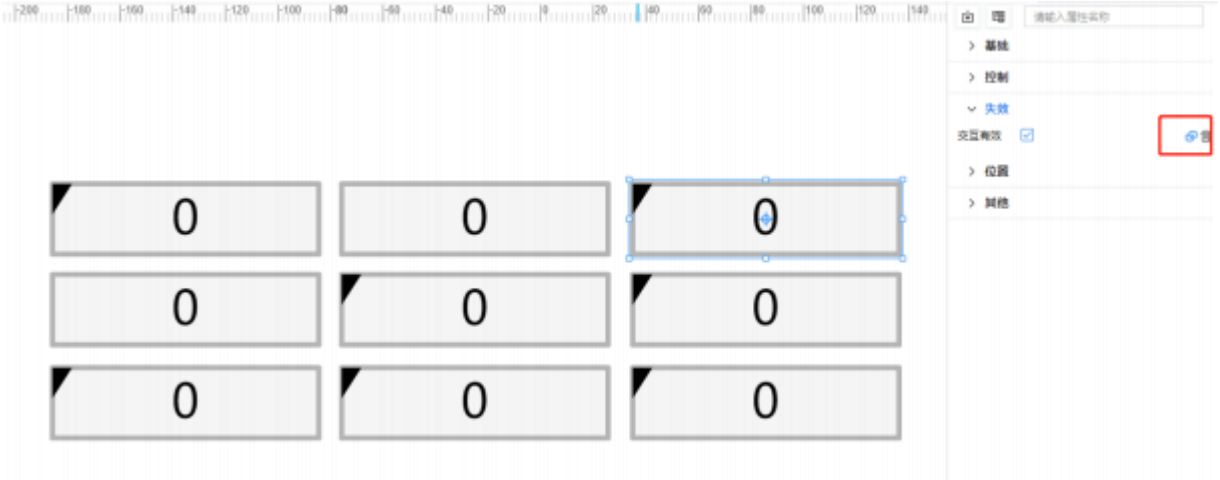
•不选中与边界值----对于不可选中的IO域，会直接进行跳过，若位于边界则是自己，例如在第一行第一列，不管往左还是往上，焦点始终在自己身上，这里将第一行第二列和第二行第一列的IO域设置为不可选中



•保存后运行



•动态不可选中---给第一行第三列的IO域的【不可选中】绑点



•然后再运行态动态的改变这个值，当值为0的时候，该图元的【不可选中】是不可选的，进行焦点移动时会跳过该图元，然后当这个IO域的值变为非0的时候，此IO域又能选中了

456.00	456.00	456.00
345.00	345.00	345.00
0.00	0.00	0.00

5.多条件文本

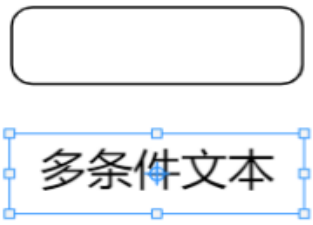
5.1.概述

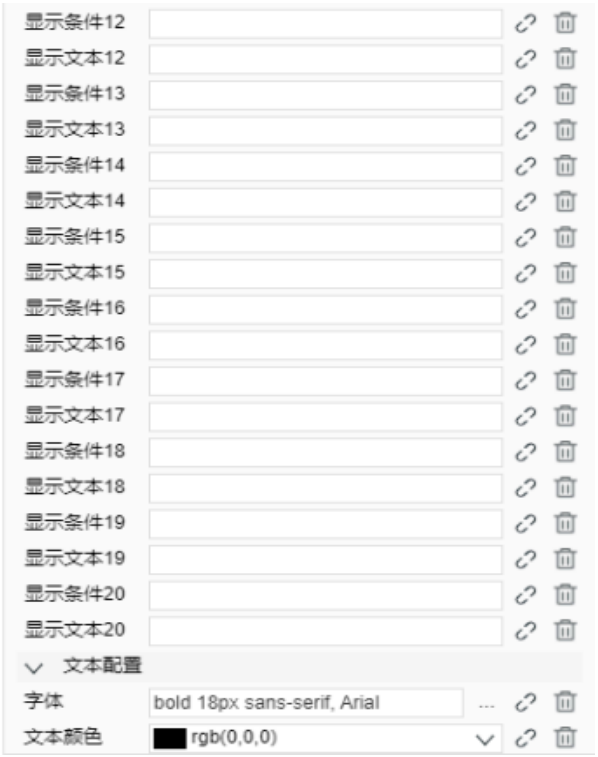
实现现场根据不同条件显示不同文本

5.2.基本功能

- 首先在系统图元中找到多条件文本→路径（symbols/系统图元/文本/多条件文本显示.json），拉取到画面中，含有以下配置项
  - 多条件文本

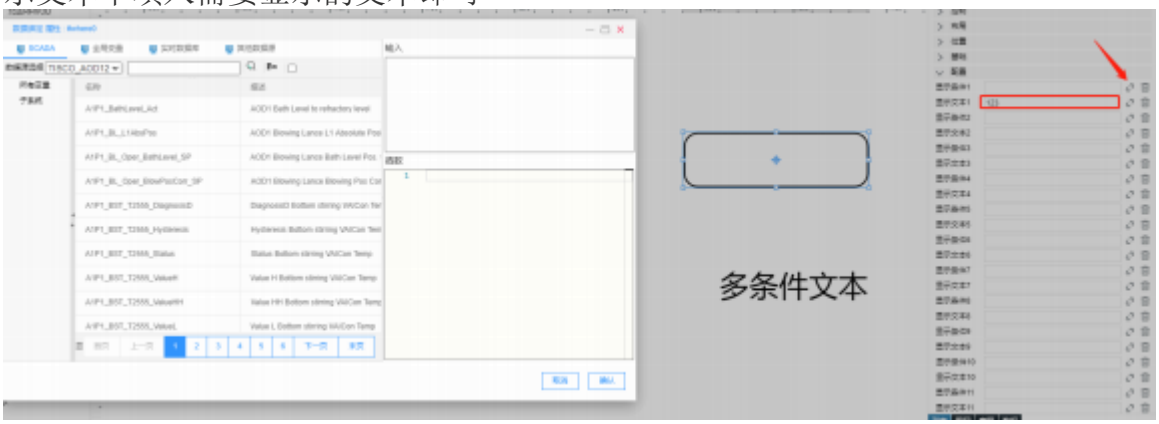
  - 多条件文本





- 显示条件1为显示文本1的条件，以此类推，多条件文本和多条件文本2区别在于一个含有边框，一个没有并且多条件文本2含有对于文本字体以及颜色的配置，根据所需使用

- 如何使用
  - 两个图元使用方法一致，在显示条件那里按点击绑定tag点或者写入表达式，在显示文本中填入需要显示的文本即可



6. 悬浮放大文本

6.1. 概述

- 实现现场需要悬浮到某个文字放大功能

6.2. 基本功能

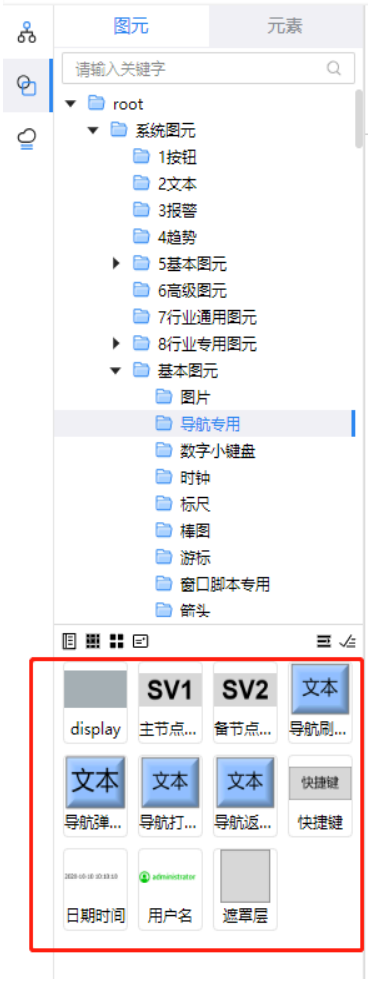
- 首先在系统图元中找到悬浮放大→路径（symbols/系统图元/文本/悬浮放大.json），拉取到画面中，含有以下配置项
  - 文本颜色，文本（可自己输入，可以绑定Tag点）

- 如何使用
  - 拉到画面中保存运行该画面，悬浮到图元上即可实现效果

基本图元

1. 导航专用

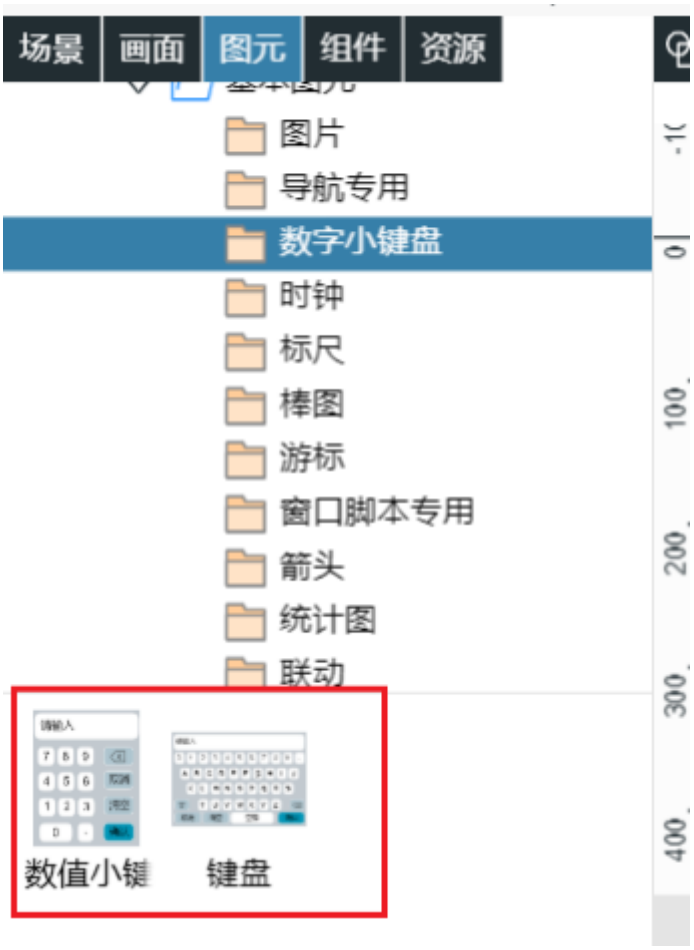
- 概述：导航图元分七种，其中导航弹窗，导航返回需要和display对应



- display
- 导航刷新
- 导航弹窗
- 导航打印
- 导航返回

- 日期时间
  - 用户名
  - 主节点状态
  - 备节点状态

2. 数字小键盘



•键盘----输入变量：键盘的输出值会绑定Tag，点击确认时会下发给这个Tag。

•数值小键盘

◦输入变量：键盘的输出值会绑定该Tag，点击确认时会下发给这个Tag

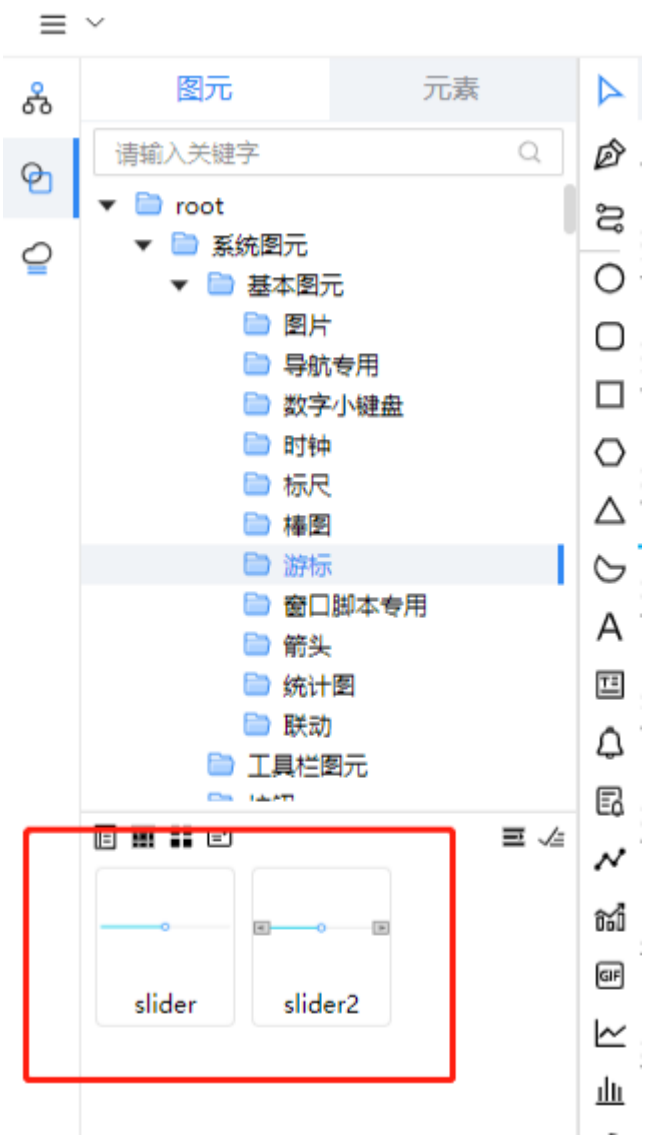
◦最小值：键盘能输入的最小值

◦最大值：键盘能输入的最大值

3. 棒图

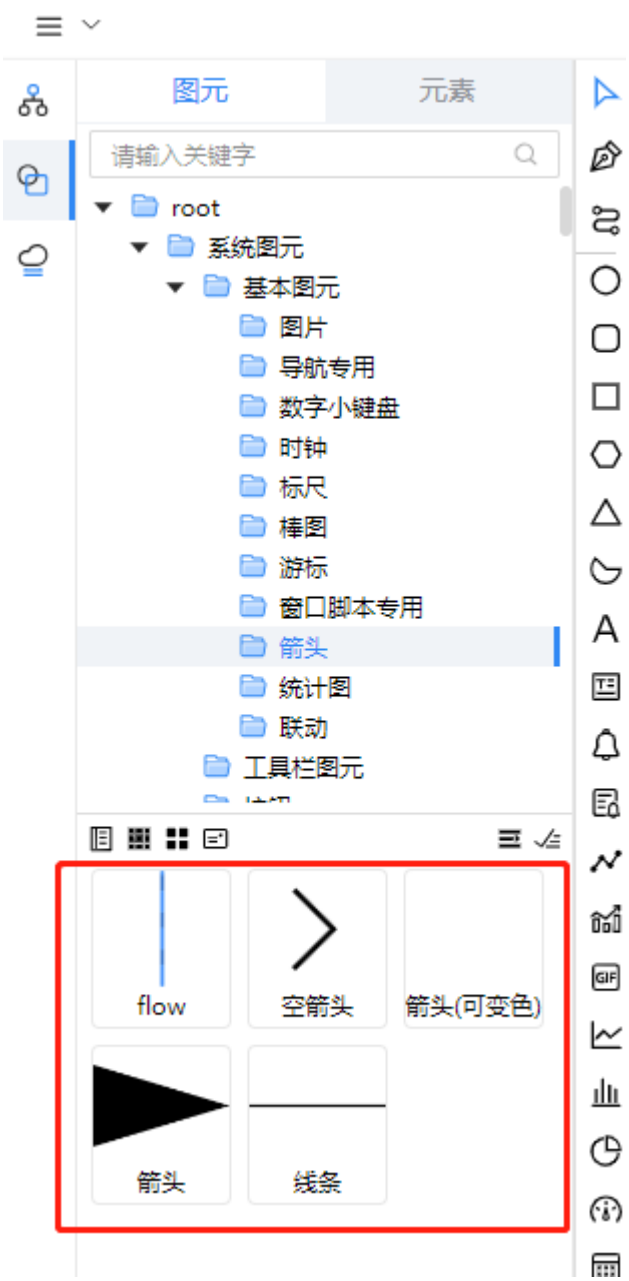
•概述：用于显示进度或者液位

4. 游标

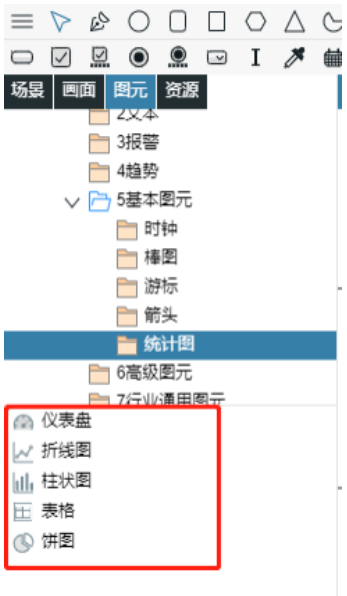

图元
元素
请输入关键字
root
系统图元
基本图元
图片
导航专用
数字小键盘
时钟
标尺
棒图
游标
窗口脚本专用
箭头
统计图
联动
工具栏图元
slider
slider2

•含有如下配置

5. 箭头


图元
元素
请输入关键字
root
系统图元
基本图元
图片
导航专用
数字小键盘
时钟
标尺
棒图
游标
窗口脚本专用
箭头
统计图
联动
工具栏图元
flow
空箭头
箭头(可变色)
箭头
线条

6. 统计图


场景
画面
图元
资源
3报警
4趋势
5基本图元
时钟
棒图
游标
箭头
统计图
6高级图元
仪表盘
折线图
柱状图
表格
饼图

表格使用方法：

7.联动图元

联动弹窗：将联动弹窗图元拖入画面，在触发源处绑定表达式，在事件中写弹窗脚本。

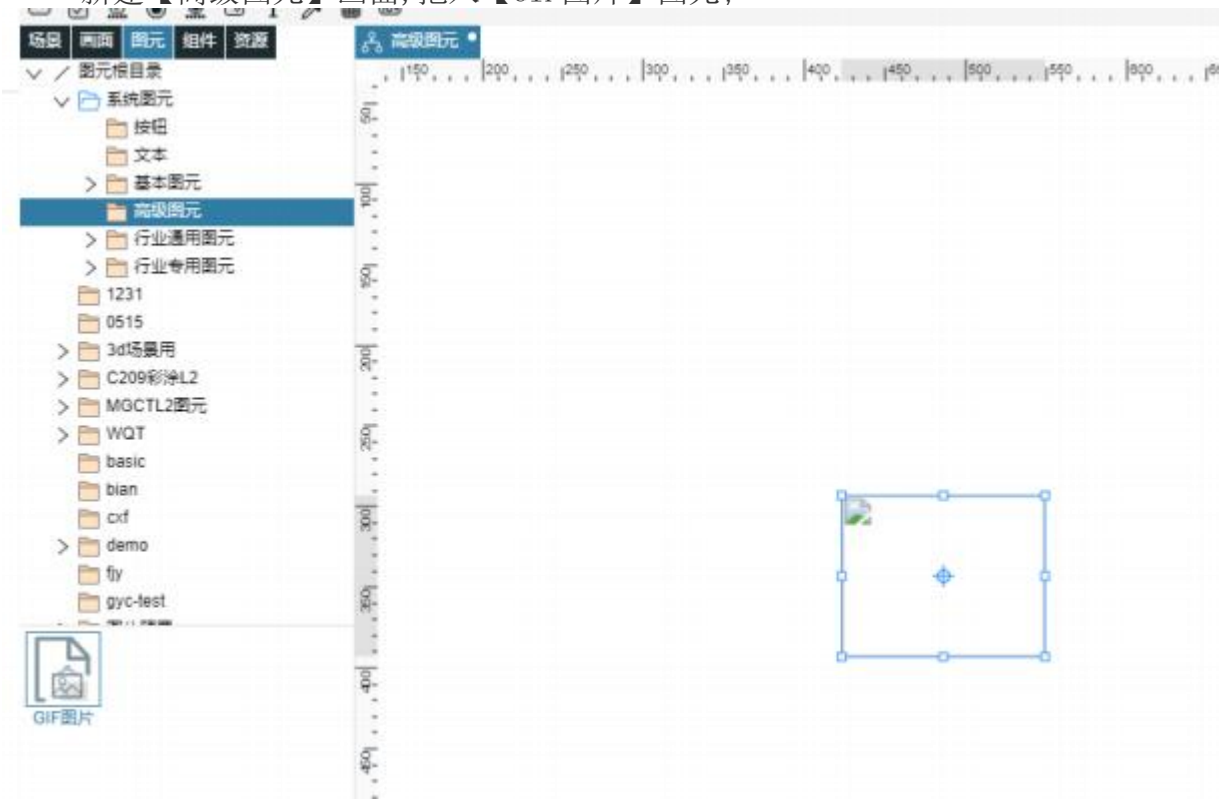
高级图元

8.概述

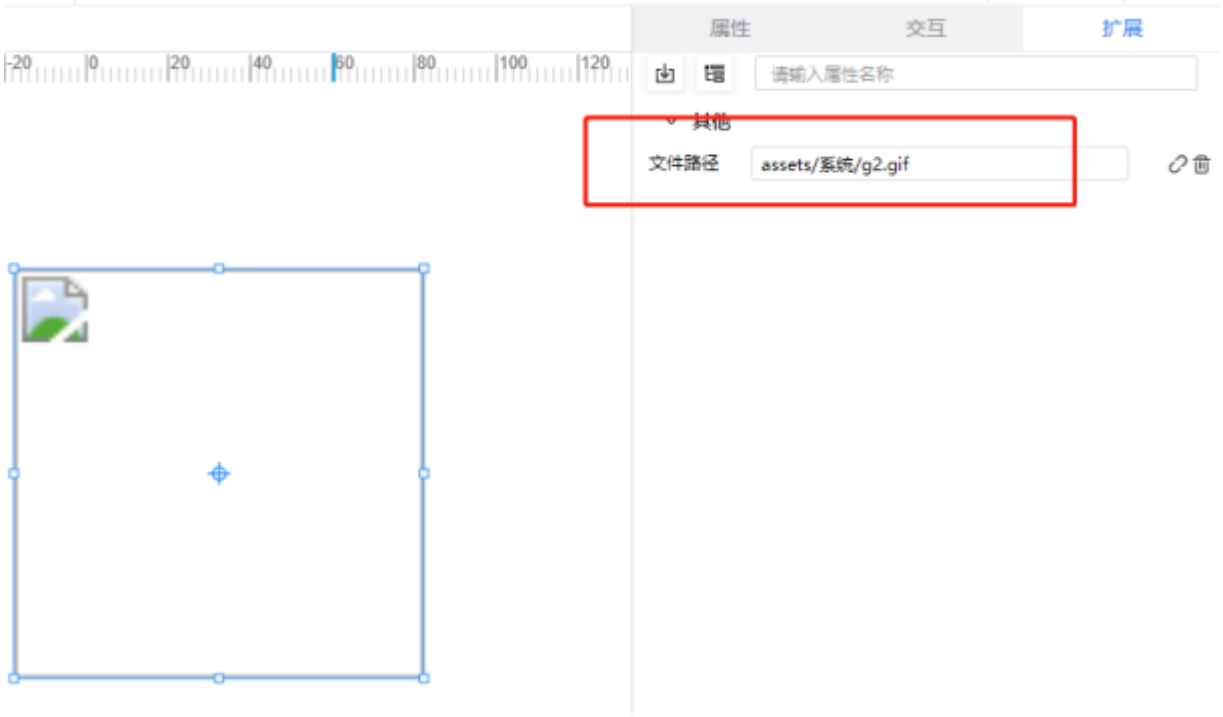
- 高级图元文件夹里的图元属于平台能力的延伸,用于实现一些特殊非基本的需求如动态图片展示;

GIF图片

- 功能
  - 在组态页面显示gif动图;
- 使用方法
  - 新建【高级图元】画面,拖入【GIF图片】图元;



- 把需要展示的gif图片(示例中为【g2】)拖入资源文件夹下,并复制底部的图片路径;每个项目建议建立独立的文件夹,方便管理;



◦选中步骤1中的图元，在【文件路径】属性里填上刚刚复制的图片路径，修改图元宽高，保存画面运行；

◦运行效果如下（可支持显隐）

行业通用图元

1. 开关

2. 手绘

3. 机房

4. 泵

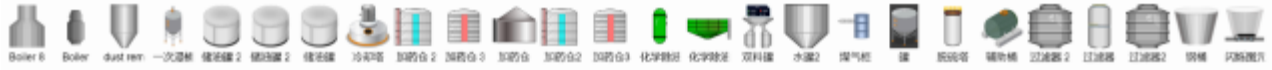
5. 炉

6. 电机

7. 符号

8. 管道

9. 罐



10. 车

11. 阀门

12. 风机

行业专用图元

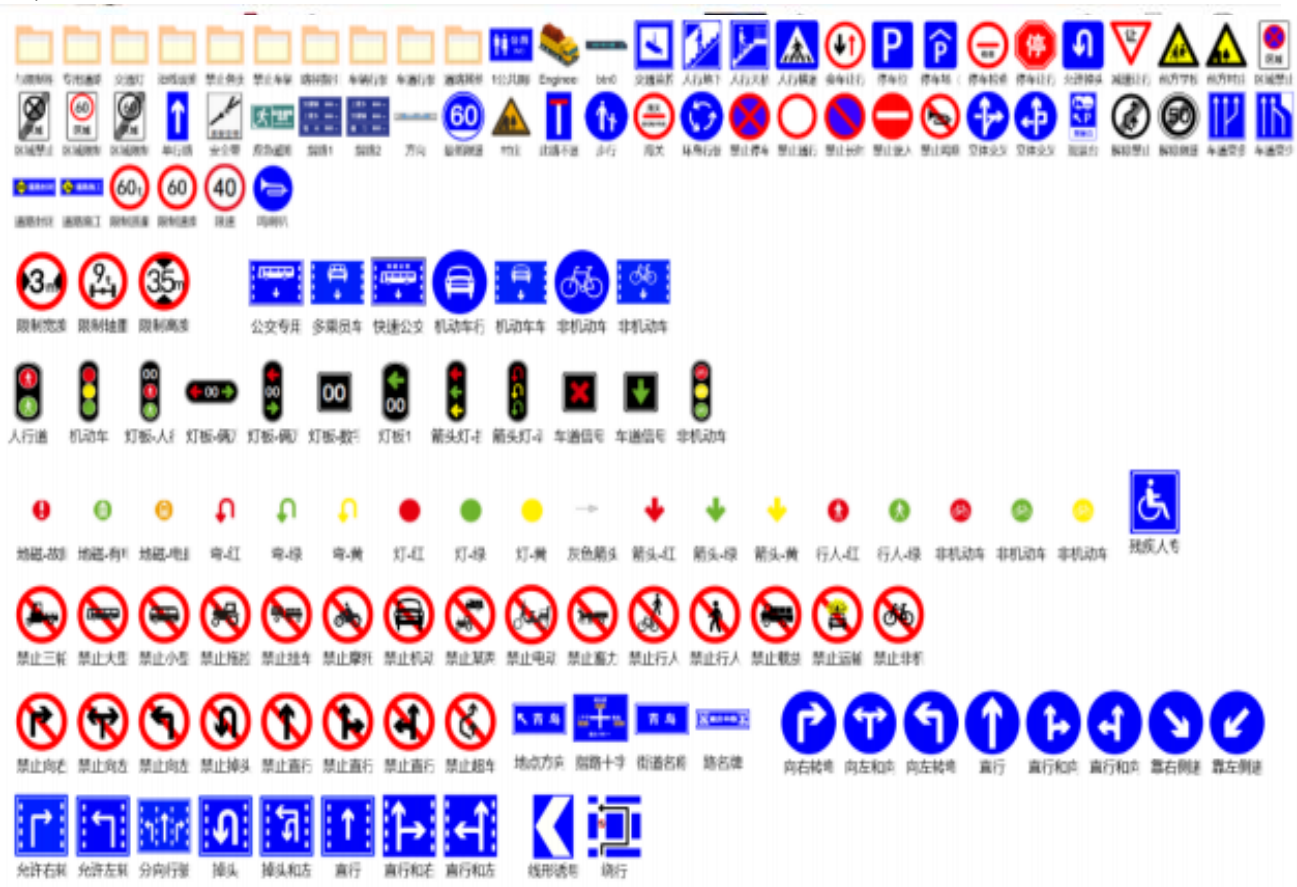
1. 化工

2. 炼钢转炉

3. 炼钢连铸

4. 电力

5. 能环



6. 车

工具栏图元

图元使用注意事项

1. 按钮

1.1. 通用按钮

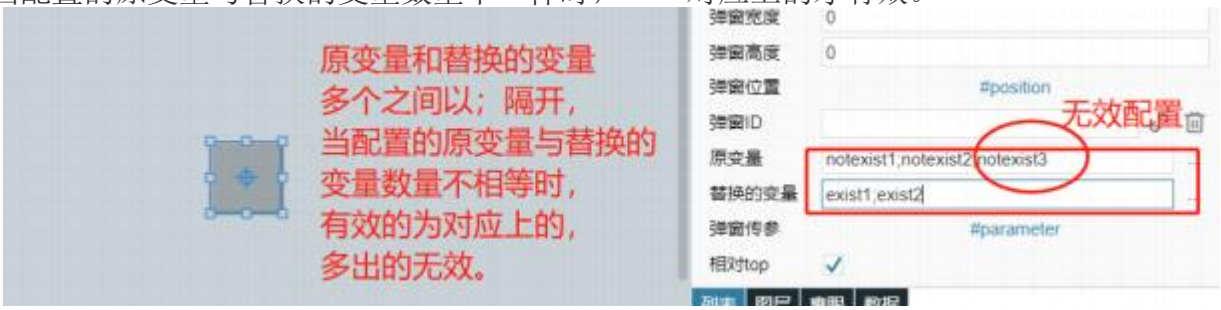
- 配置了默认颜色后，在运行态如果配置的背景颜色没有匹配到，例如：配置了点值1显示绿色，点值0显示红色，但点值异常了，这个时候如果有默认色就会显示默认色，否则显示红色。

1.2. 关闭弹窗配置：

- 遇到点击按钮后，执行按钮的功能且关闭按钮所在弹窗的场景时，可以直接勾选启用关闭弹窗配置，然后弹窗ID缺省即可。

1.3. 弹窗配置：

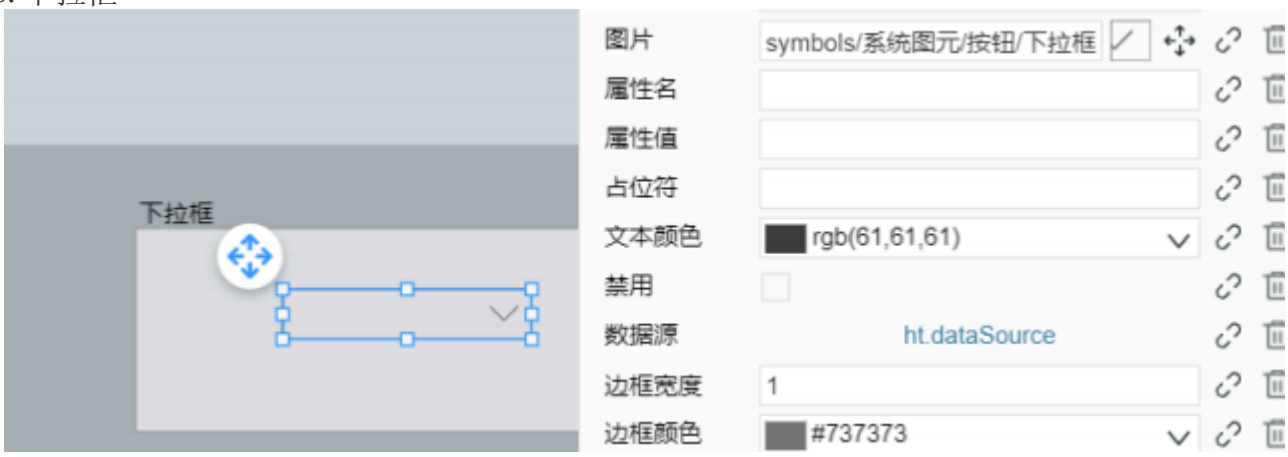
- 当配置的原变量与替换的变量数量不一样时，一一对应上的才有效。



1.4. 单选框

- 属性：
  - boxvalue是用来绑点的，适用于DI值得点
  - 状态反转：不勾选，默认1显示对号，0不显示对号；勾选后，默认0显示对号，1不显示对号
  - 注意：点击图元用于点值取反

1.5. 下拉框

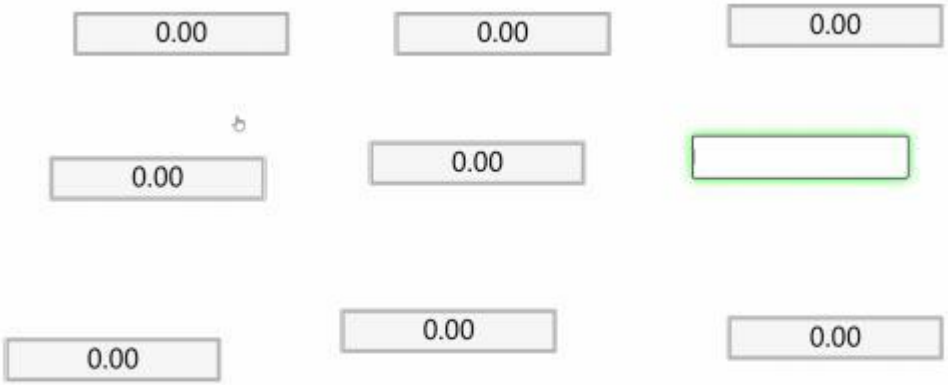



```
[{
"label": "Home Screen",
"value": "Home Screen"
},
{
"label": "Home Screen",
"value": "Home Screen"
}]
```

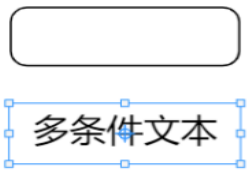
2. 文本

- io域1
  - io域的上下限会限制输入，输入的数超出上下限则无效，但不会限制输出，如果点值采上来时就已经超出上下限会直接显示，如图中配置的是0到10包含0和10，如果点值采上来就是100，会显示100。

- io域2
  - 可移动焦点IO域，使用时注意，前后左右IO域尽量对齐，横平竖直，如下图，不能出现错位程度太大的情况，否则容易移动时达不到想要的效果



- io域(两变量)
  - 此类IO域，输入和输出分别绑点不同的Tag，使用时注意区分。
- 多条件文本
  - 使用时注意，条件和文本的一一对应关系，应该得到保证，避免出现，多个条件同时成立的情况



- 文本框

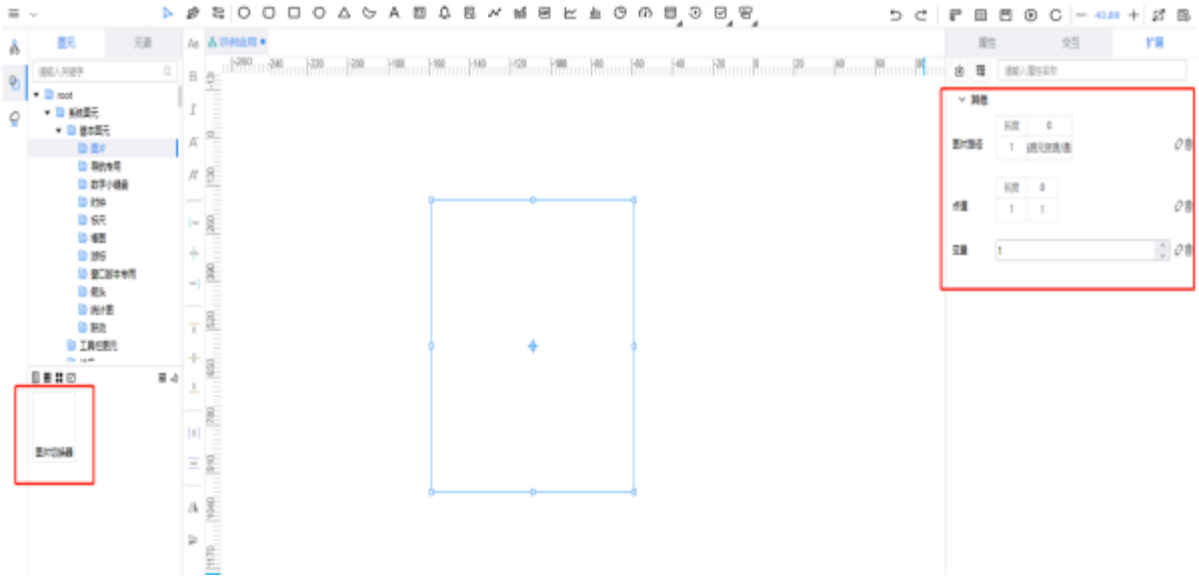
。该文本框与文本控件，区别在于，文本框图元：

. 更丰富的颜色配置规则

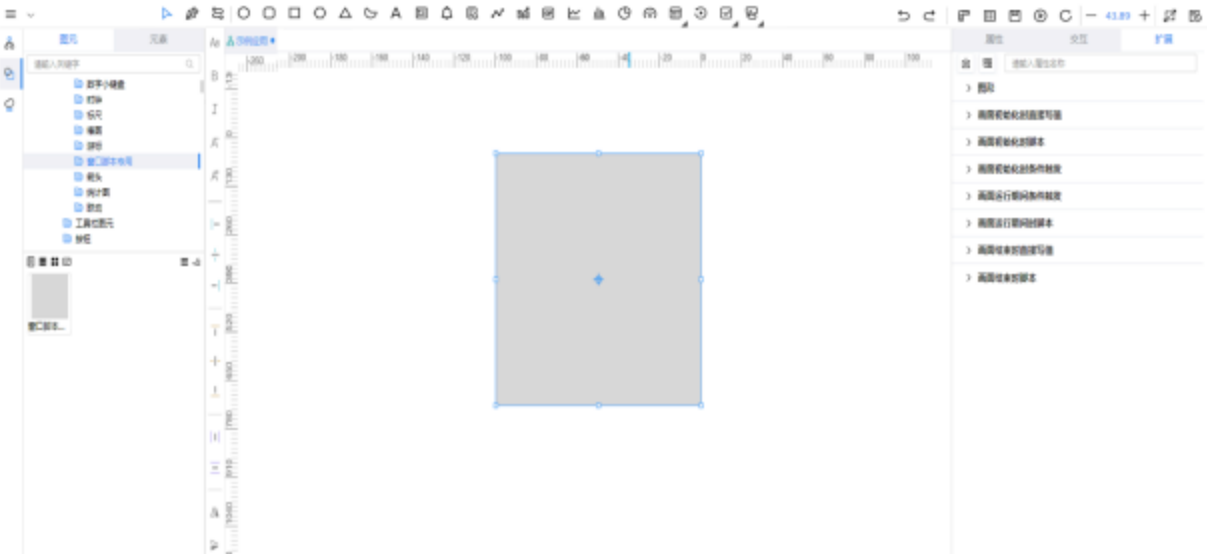
. 增加了各种触发时间，比如单击写值等
. 文本边框可配置
•文本域
◦更灵活的多行、单行文本配置，可以灵活调用、控制以达到预期效果
•多行文本
弥补了文本控件的不足，在文本换行的场景场景中使用

3. 基本图元

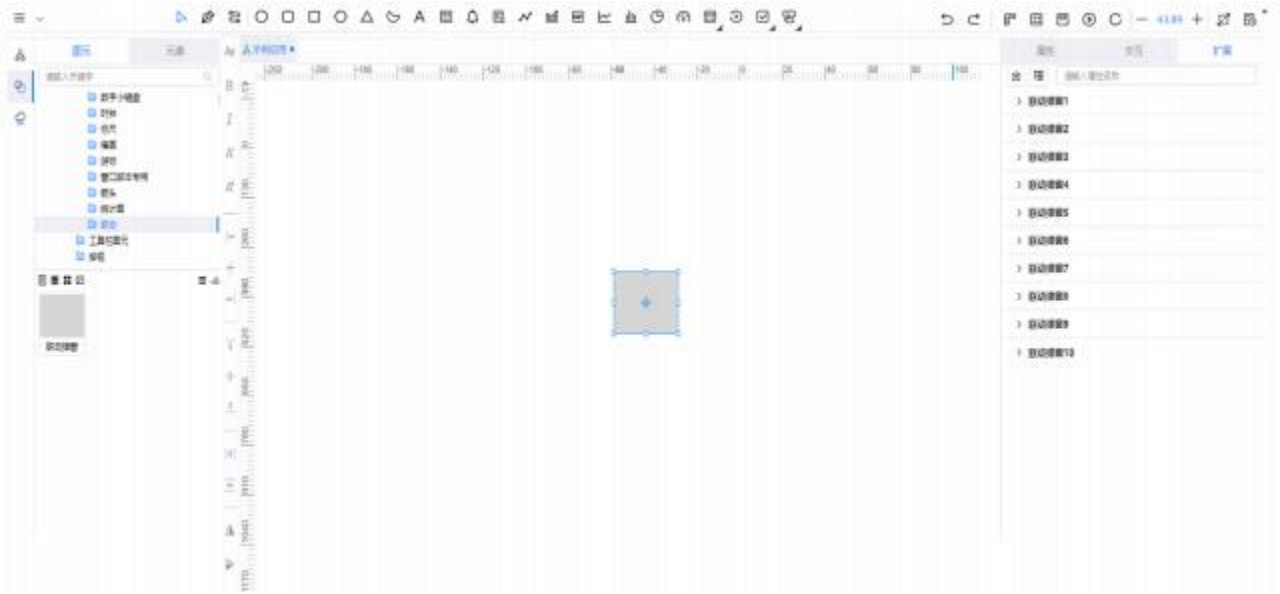
•图片
◦图片切换器：图片和点值双数组一一对应，通过控制变量点值去切换图片。



•导航专用
◦刷新按钮：功能与浏览器刷新或f5一样。
◦返回按钮：只能返回上一幅画面，如果连续点那就是在两幅画面之间切换。
•数字小键盘
◦数值小键盘
. 虚拟键盘功能，九键数值小键盘
•键盘
◦二十六键数值小键盘。
•窗口脚本专用
◦画面结束时脚本，目前只支持写值



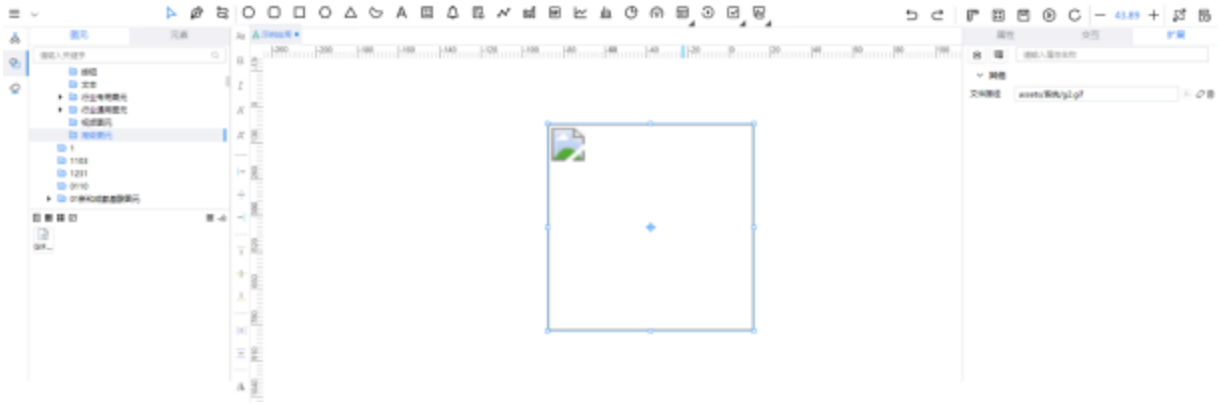
•联动



4. 高级图元

•GIF图片

。将gif图片资源导入资源文件夹后，把路径配置到文件路径属性上即可。



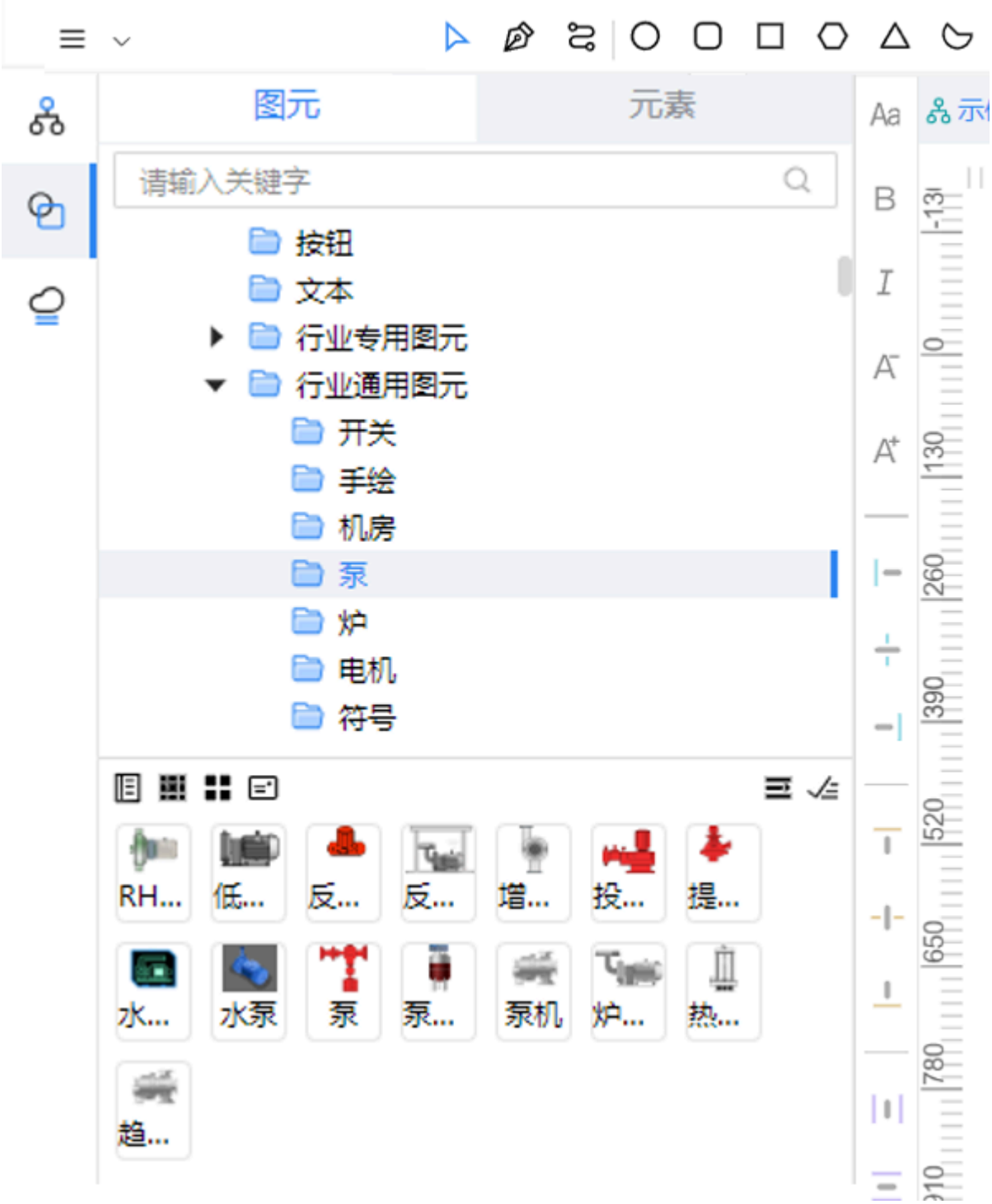
5. 行业通用图元

•行业通用图元包含：开关、手绘、机房、泵、炉、电机、符号、管道、罐、车、阀门、风机在内的总共12中元素集合。

。各行各业比较通用的元素集合用于快速找到需要的元素或者类似的元素

。这个集合内的元素基本都没有功能，如果需要增加一些逻辑功能，需要自行复制改造图元

。图元集合可以不断扩展，方式一般是，项目结束后整理，或提出相应需求



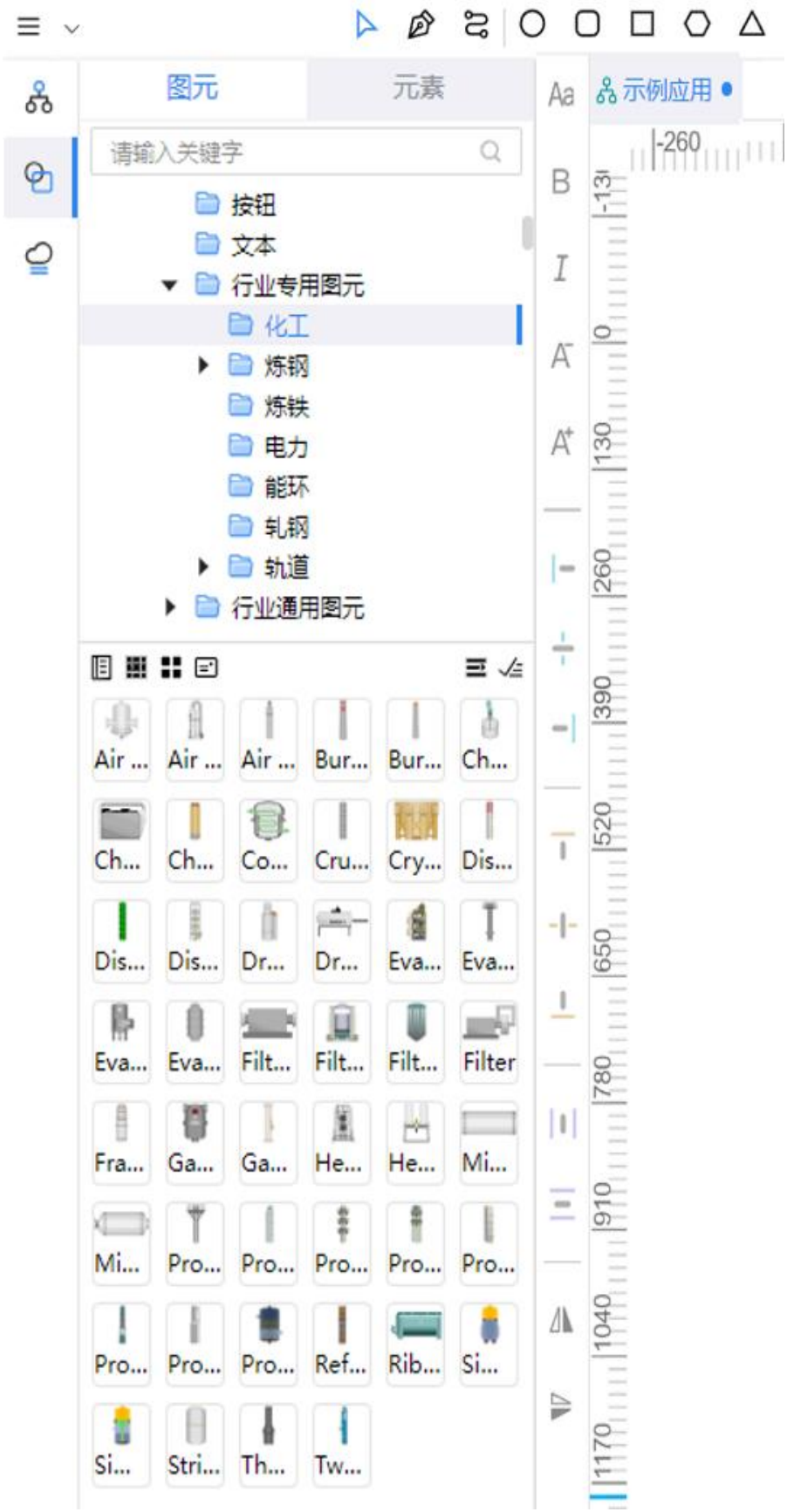
6. 行业专用图元

•行业专用图元包含：化工、炼钢、炼铁、电力、能环、轧钢、轨道在内的多个行业沉淀的项目图元。

。同一个行业内，图元的通用性极强，有很高的复用性

。相当一部分图元是含有功能和业务逻辑的，需要在业务场景下使用

。可扩展，项目结束后可以集中收集上传



5.25 图元自定义

基本介绍

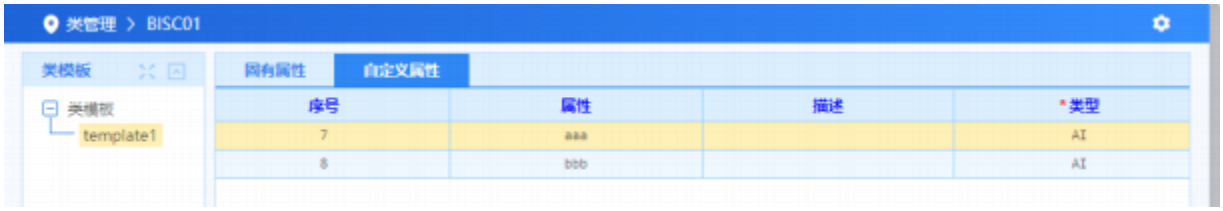
在某些应用场景下，用户需要在图元中，给对象点的值做一些逻辑判断。这种情况可以用图元的自定义属性来完成。

典型案例

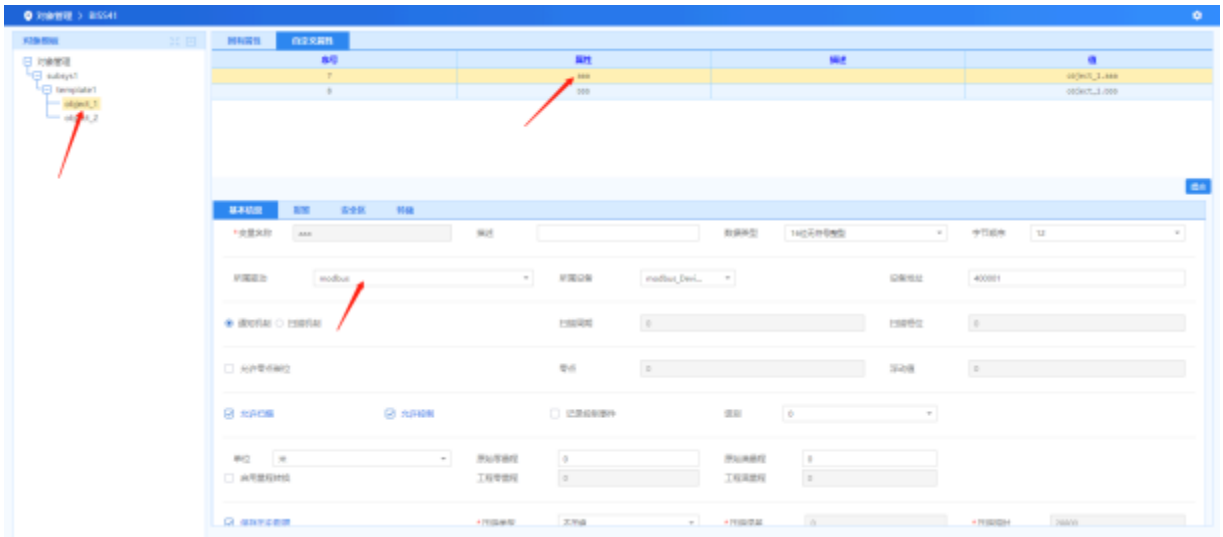
本案例中，新建了一个绑定了模板属性的自定义属性，然后给多边形的背景色绑定了一段函数，这段函数中利用自定义属性做了一些逻辑判断，从而达到我们想要的效果。

1. **准备数据**

•新建类模板template1,并定义自定义属性aaa

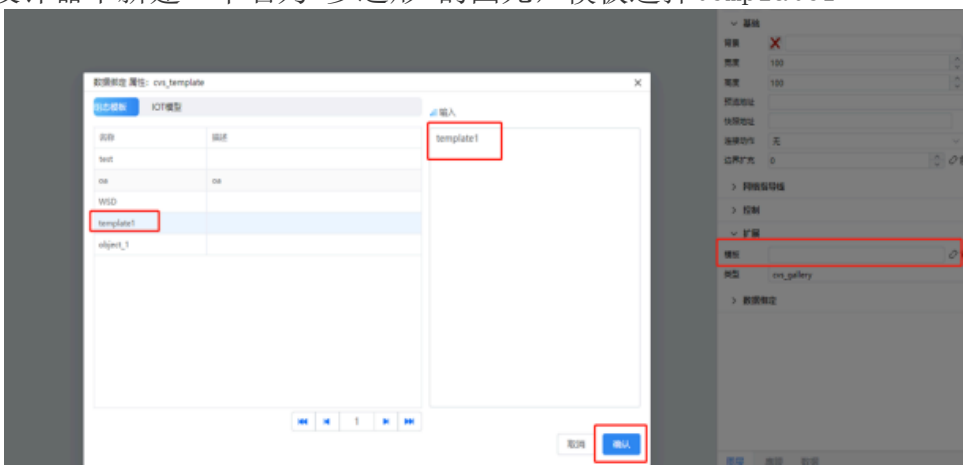


•在scada的对象管理中，实例化对象object1，并对属性aaa配置好驱动等信息



2. **图元选模板**

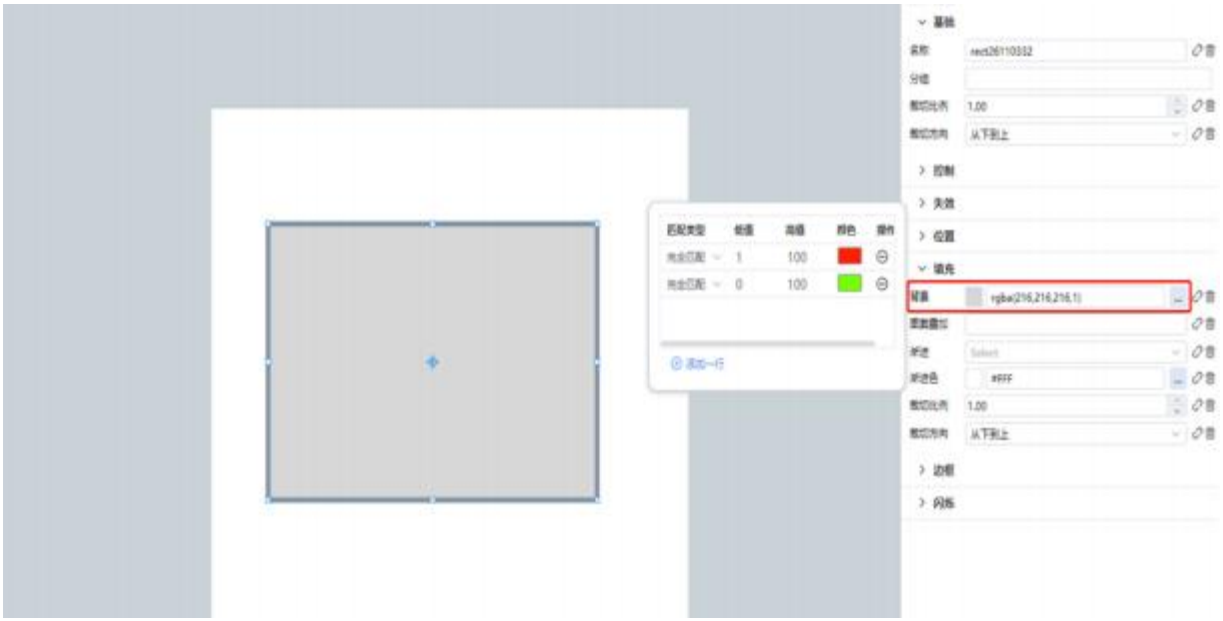
•在2D设计器中新建一个名为"多边形"的图元，模板选择template1



3. **新建自定义属性**

•新建一个名为#attr0的自定义属性，Tag类型，默认值为@template1@.aaa

4. **配置多边形背景色**
   - 图元中拖拽一个多边形，背景配置中设置背景颜色完全匹配，值为0和1时，对应红色和绿色

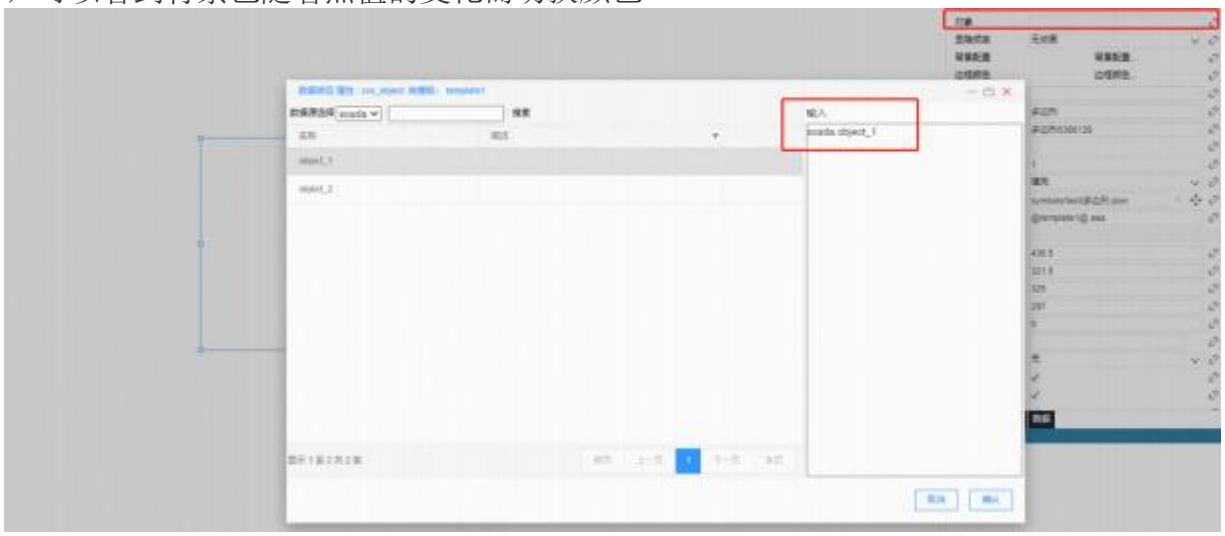


   - 点击背景配置右侧的别针，弹出选点界面，勾选"函数校验"复选框，切换至函数编辑态，输入以下函数内容，保存

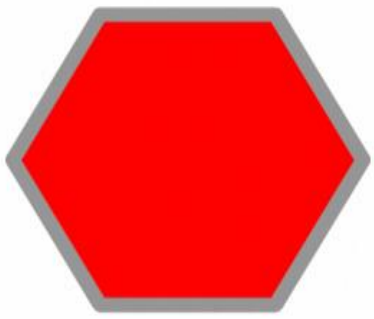
```
function(data, view){
    var attr0 = data.a("#attr0");
    var ret = attr0%2;
    return ret;
}
```

5. **画面操作**

•新建"画面1"，把图元多边形拖入画面，给多边形选择对象object1。保存画面并运行，可以看到背景色随着点值的变化而切换颜色



•效果：



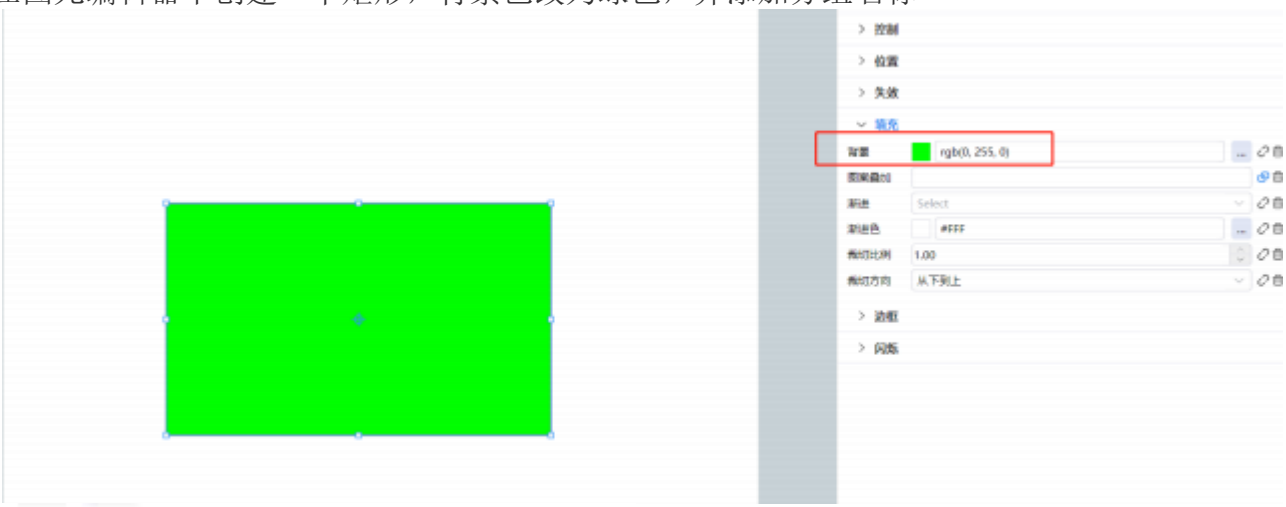
颜色规则自定义属性

颜色规则自定义

在图元中，颜色属性（背景颜色、边框颜色、闪烁颜色等）可以关联一个颜色规则类型的自定义属性，并在实例化的画面中去配置具体的规则，绑定用户指定的tag点。

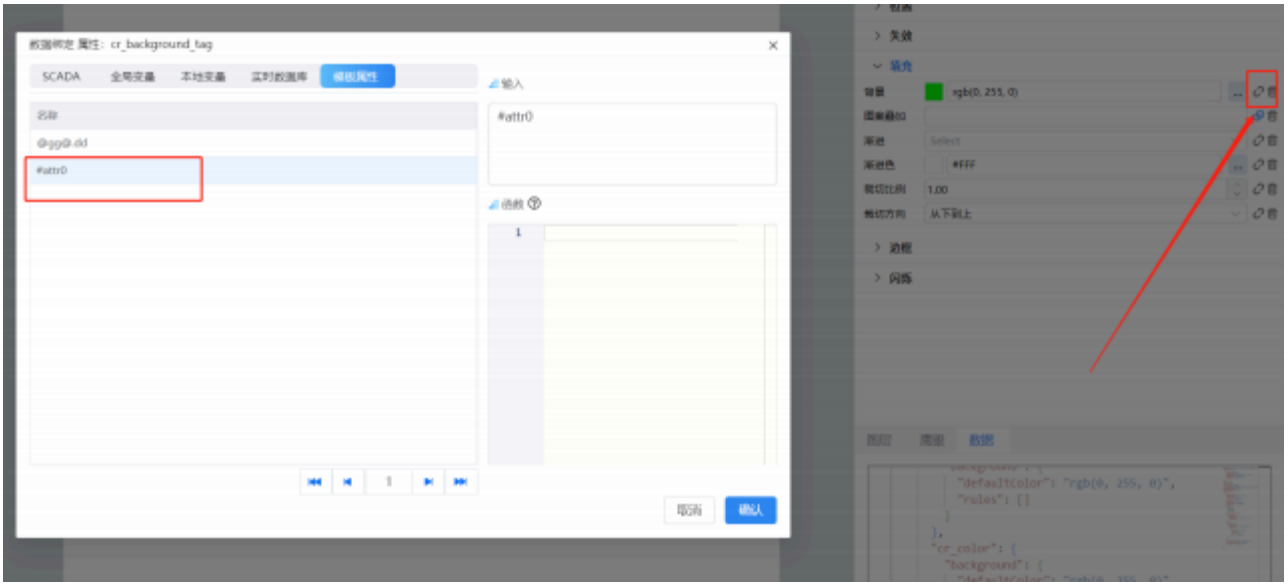
•示例

1. 在图元编辑器中创建一个矩形，背景色改为绿色，并添加分组名称



2. 新建#attr0为【颜色规则】颜类型，设置默认值为红色，并配置规则。

1. 给矩形的背景色绑定#attr0，保存并签入图元。



4. 图元拖入画面，在图元的扩展属性中，#attr0如下：

5. 给#attr0绑定tag点

6. 矩形的颜色按照#attr0规则变化

颜色自定义

- 生效规则

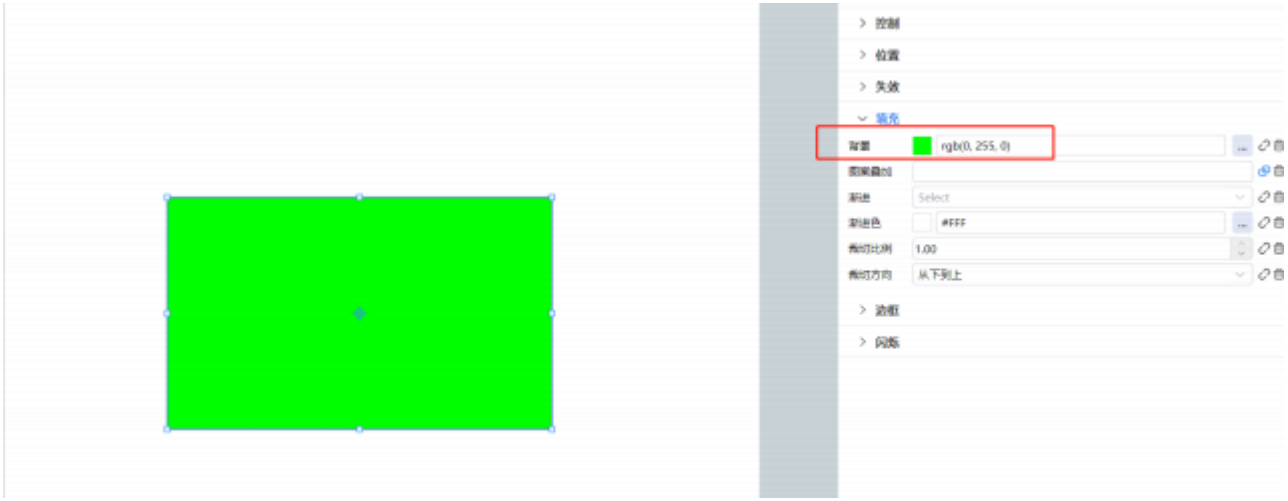
1. 画面如果绑点，以绑点值为准
2. 画面没有绑点，以画面修改后的默认色为准。
3. 画面没有绑点，且没有修改过默认色，以图元的#attr0的默认色为准。
4. 图元如果没有绑#attr0，则以图元自己的颜色为准。

综上，颜色优先级别为：

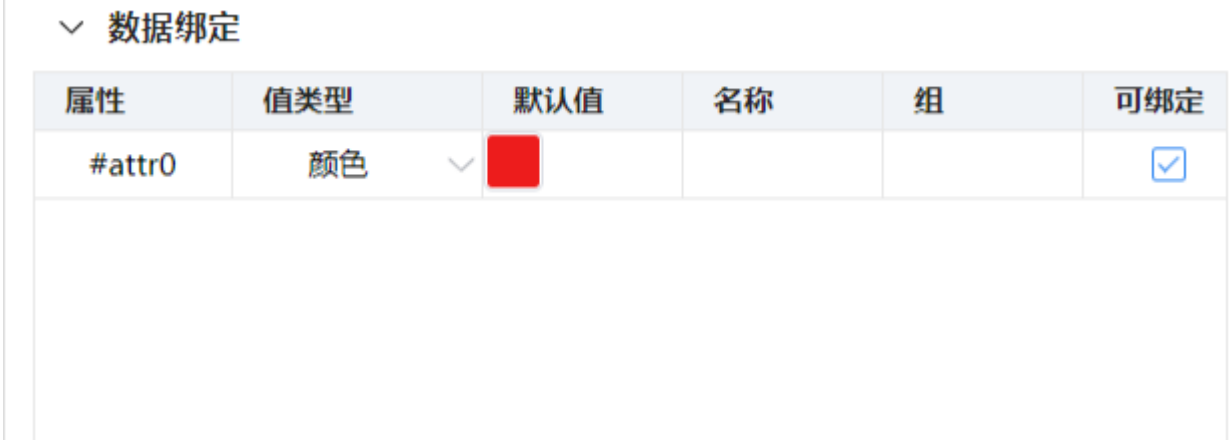
画面绑点色画面扩展属性的自定义属性默认色图元自定义属性默认色、、、图元属性色

•示例

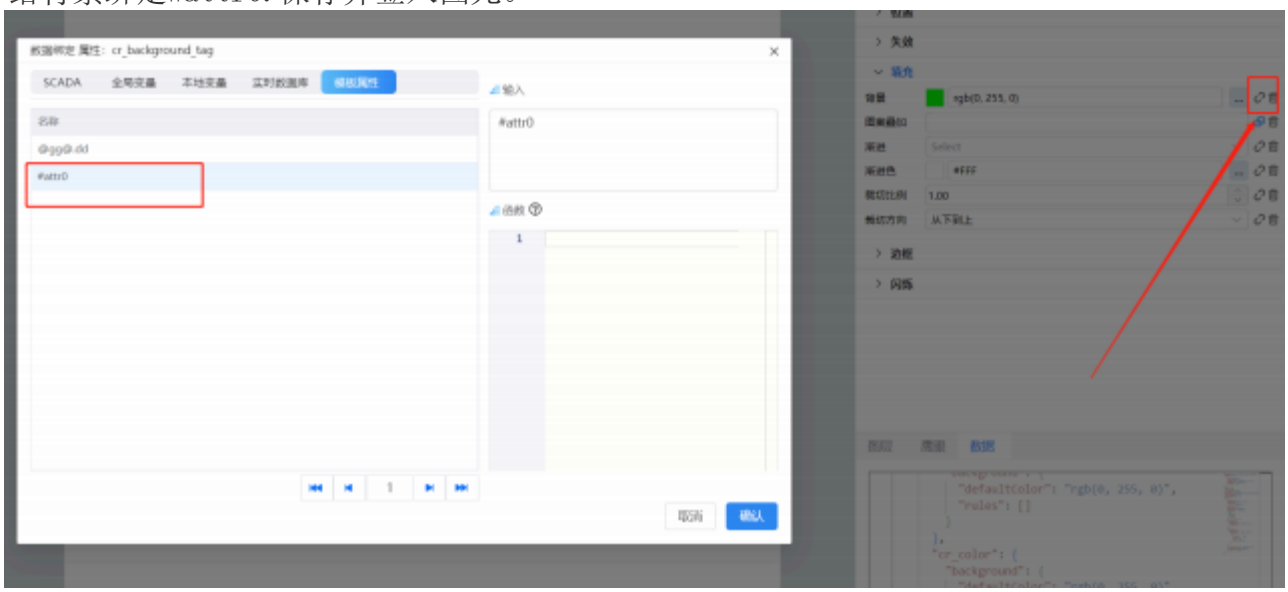
1. 在图元编辑器中创建一个矩形，背景色改为绿色



2. 新建#attr为颜色类型，设置默认值为红色



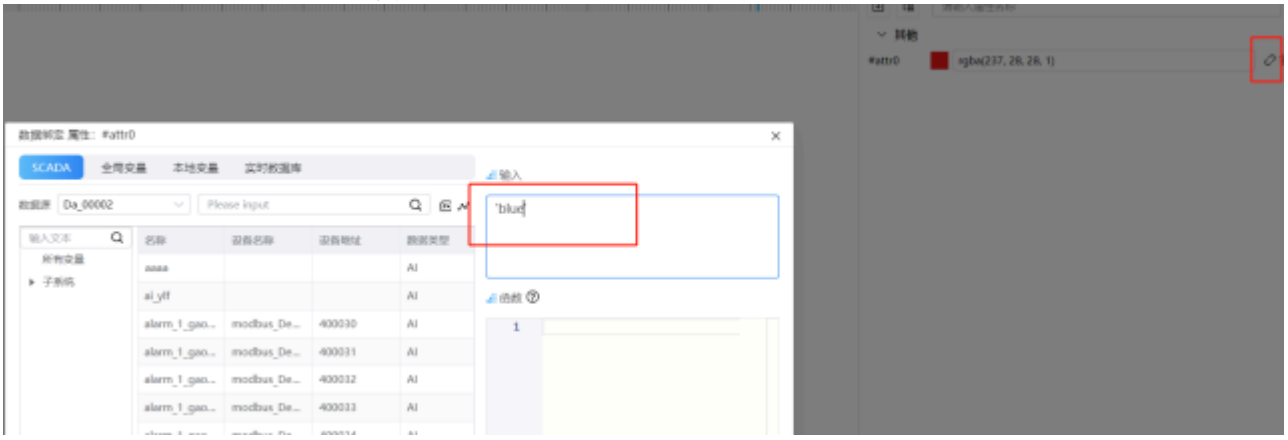
3. 给背景绑定#attr0. 保存并签入图元。



4. 画面中拖入图元，扩展属性中可以看到之前配置的#attr。运行画面，矩形颜色是红色，被#attr0的默认色替换。

5. 给画面扩展属性的#attr颜色修改为黄色。运行画面，矩形颜色是黄色。

6. 在画面中，给#attr0，绑定内容，例如绑表达式"blue"，运行画面，矩形变成蓝色。

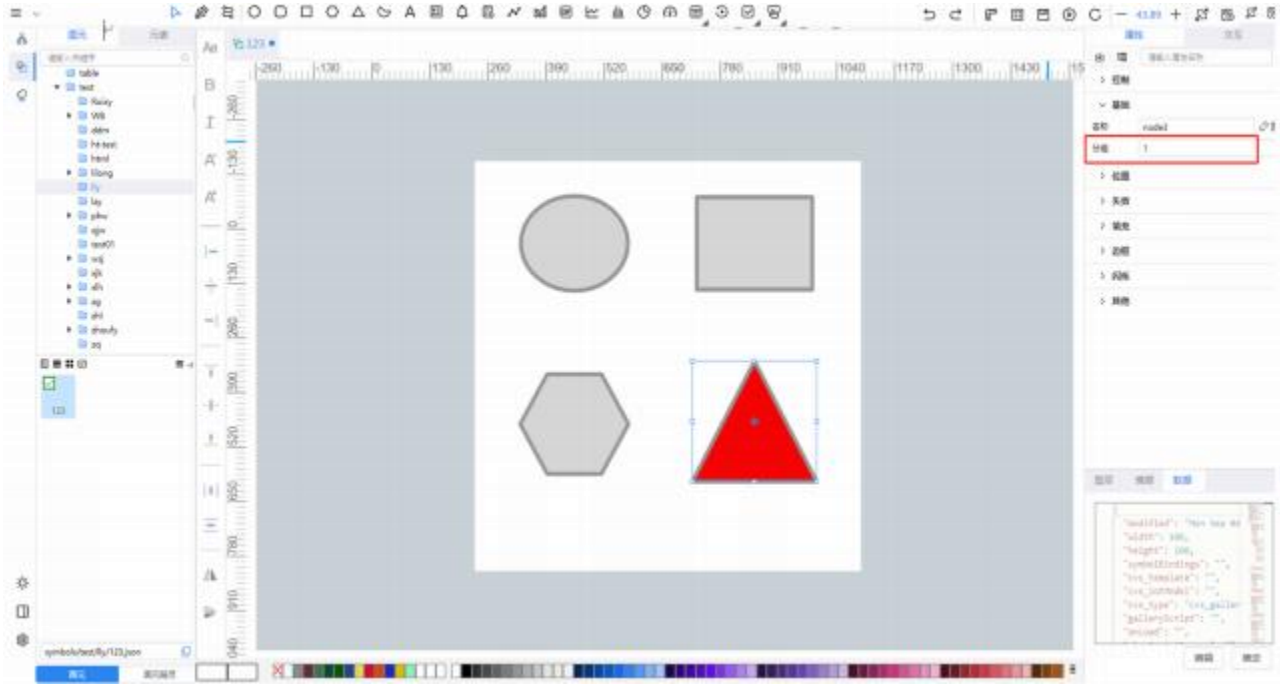


5.26 图元分组

什么是分组？

•概念：″分组″是图元中各个元素都存在的一个属性，以前的版本中叫″前缀″；

•效果：该属性主要是用来将图元中的元素进行分组，组内所有元素的属性表达均相同；



分组的作用

•使用图元元素的扩展属性，则必须给元素添加分组

•当图元内多个元素有相同的属性设置，可以将这类元素分组，分组后只需给其中一个元素进行属性设置，组内所有元素均表达相同效果。

•使用脚本直接set一个或者同组图元元素的属性时，通过前缀可以在data.a中直接访问元素属性（data.a("分组名.属性")获取，data.a("分组名.属性",值)设置），这会对所有为该前分组的元素的这个属性生效。

分组在具体场景中可以很灵活的使用，既可以用来通过设置不同分组在设置、获取属性时区分不同元素，也可以用来设置为相同分组让不同元素的属性发生一样的变化。

⚠如果不设置分组，该元素的扩展属性不会生效。在未通过data.a("分组.属性",值)设置前，data.a("分组.属性")拿到的是undefined

⚠同一分组元素的同一个属性，以解析时第一个绑点或function的元素属性为主，如果都没有绑点和function，"显隐频率"以找到的第一个非"不可见"的元素为准；"边框颜色"、"背景颜色"和"文本颜色"以第一个非默认色的元素为准

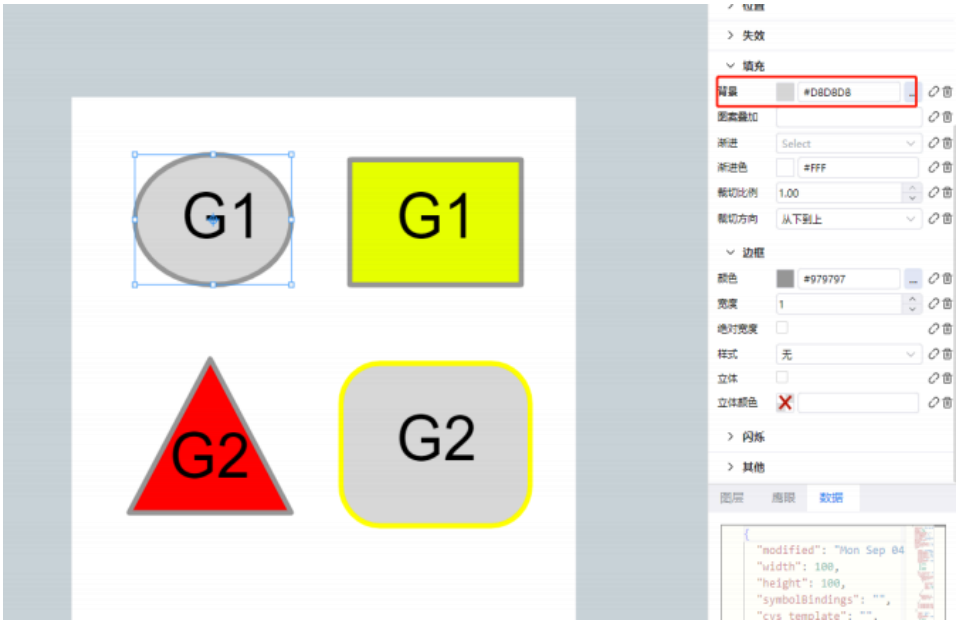
⚠图元编辑态发现相同同组的不同元素属性配置不同，保存时会提示用户进行修改

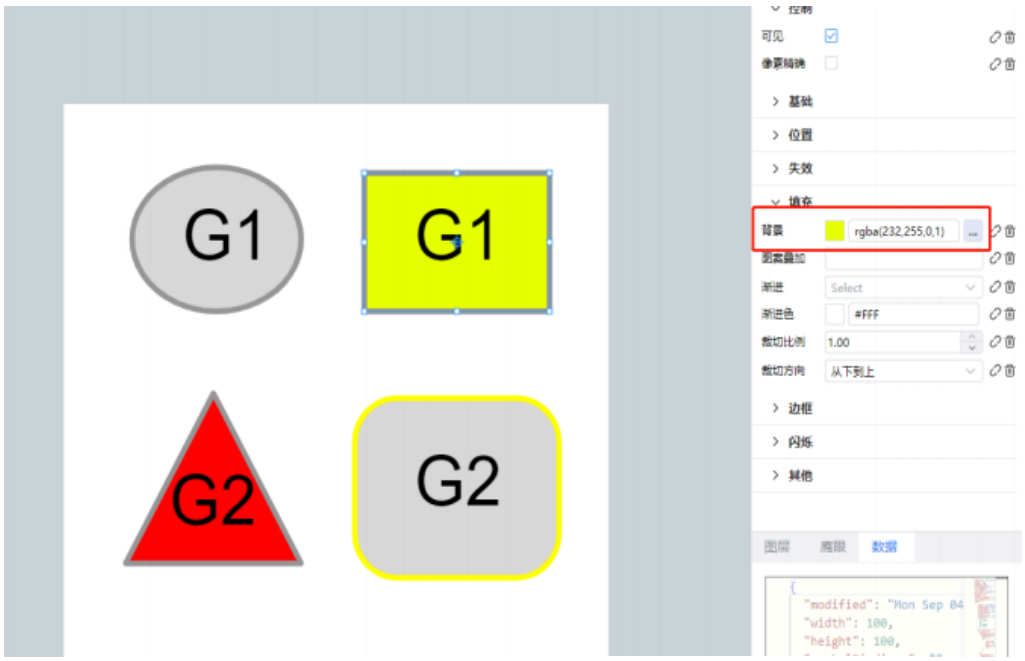
//运行态效果展示：

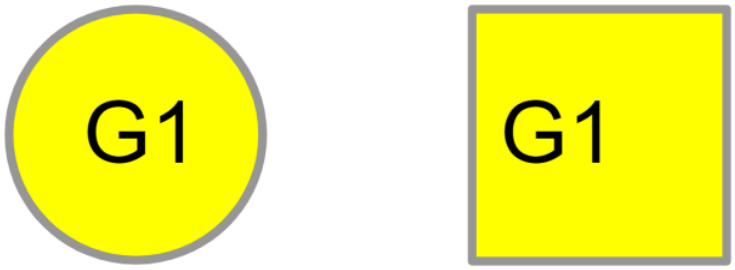
①以解析时第一个绑点或function的元素属性为主

②如果都没有绑点和function，"显隐频率"以找到的第一个非"不可见"的元素为准；"边框颜色"、"背景颜色"和"文本颜色"以第一个非默认色的元素为准

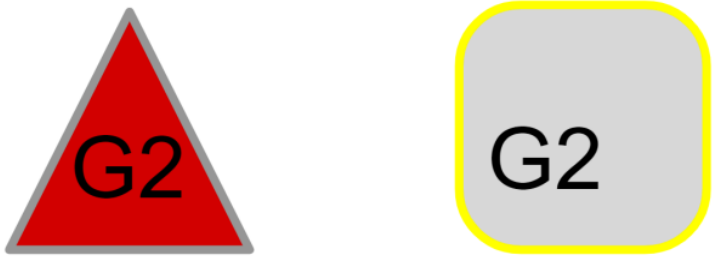
例：同组的第一个元素背景颜色为#D8D8D8（默认色），第二个元素背景颜色设置为rgb(255,255,0)（黄色），该组第一个非默认色的元素背景颜色为rgb(255,255,0)（黄色），且该组元素背景颜色无其他绑点，则运行态时该组所有元素背景颜色都为rgb(255,255,0)（黄色）。（边框颜色、文本颜色和显隐频率同理）





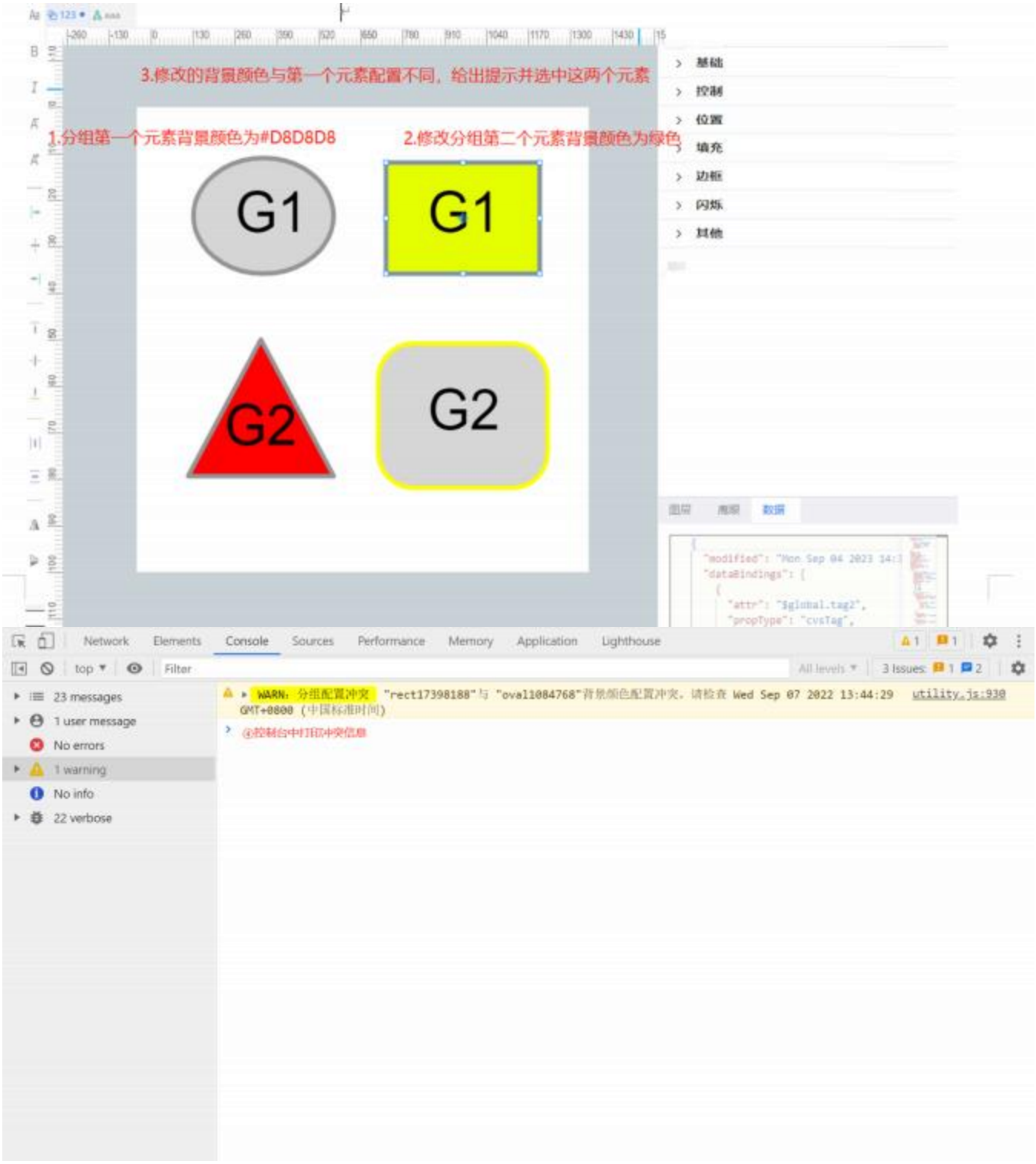


运行态时，G1的背景色均显示为黄色

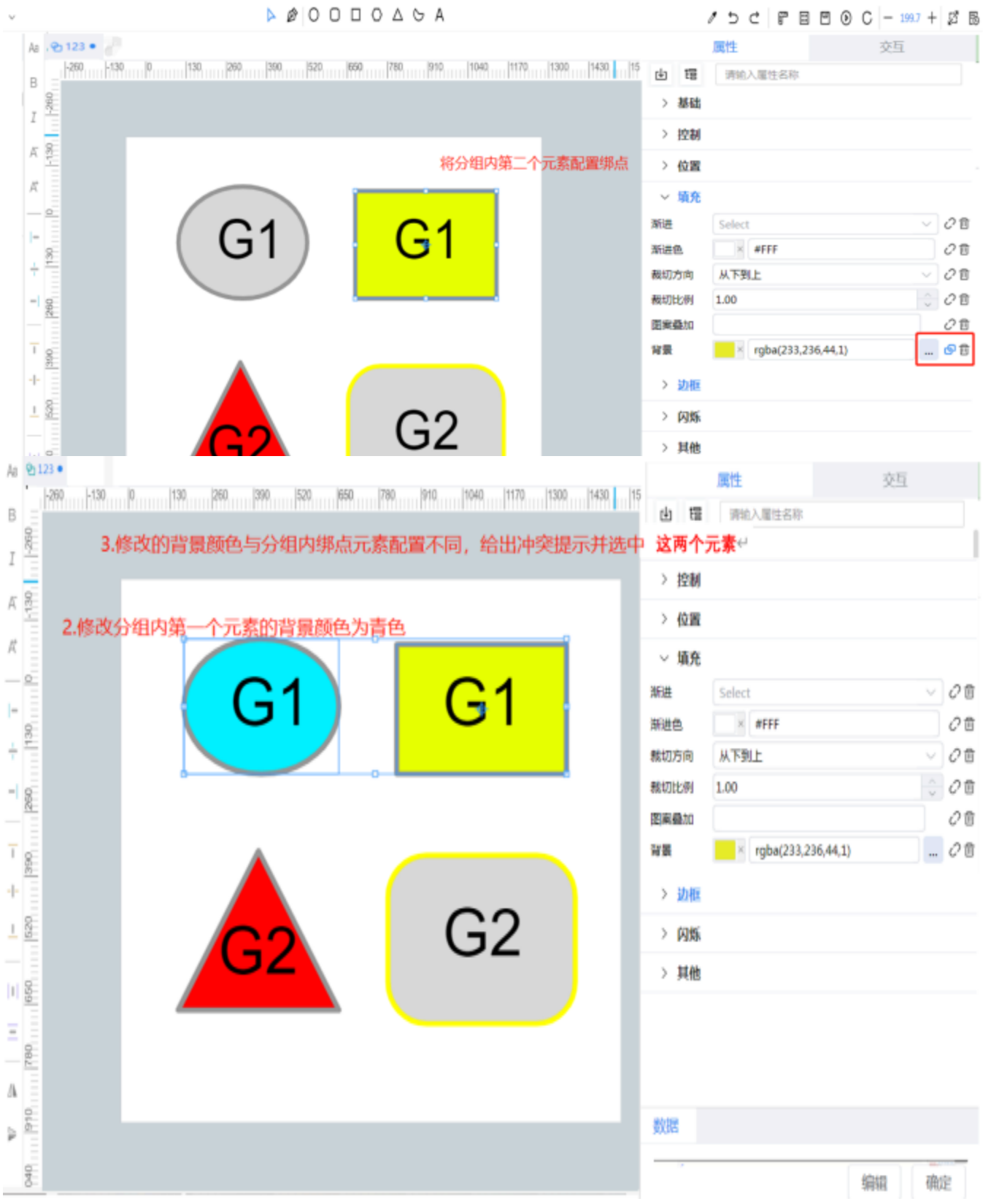


//编辑态校验效果展示：

①修改背景颜色默认颜色时（边框颜色、文本颜色和显隐频率同理）时，如果该组元素无绑点，会与该组第一个元素对比，如果配置不同，给出提示选中所有冲突的元素，并在控制台中打印

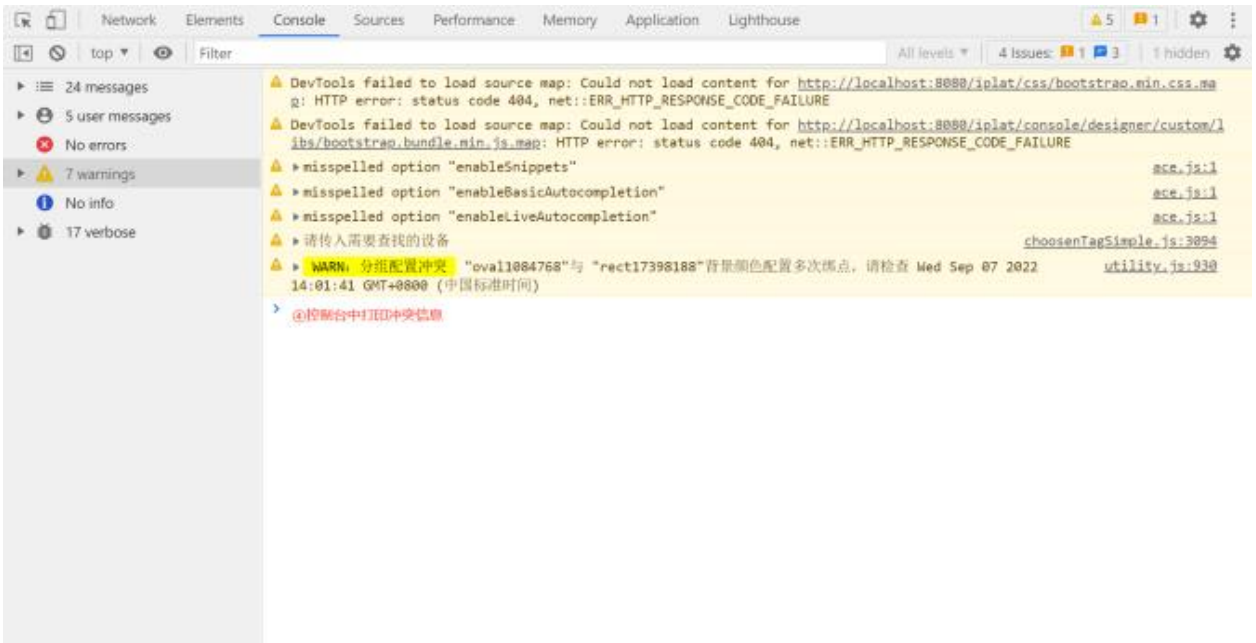


②修改背景颜色默认颜色时（边框颜色、文本颜色和显隐频率同理）时，会与该组第一个绑点元素（假设该组第一个元素无绑点，第二个元素有绑点，即为列表中的第二个元素）对比，如果配置不同，给出提示选中所有冲突的元素，并在控制台中打印

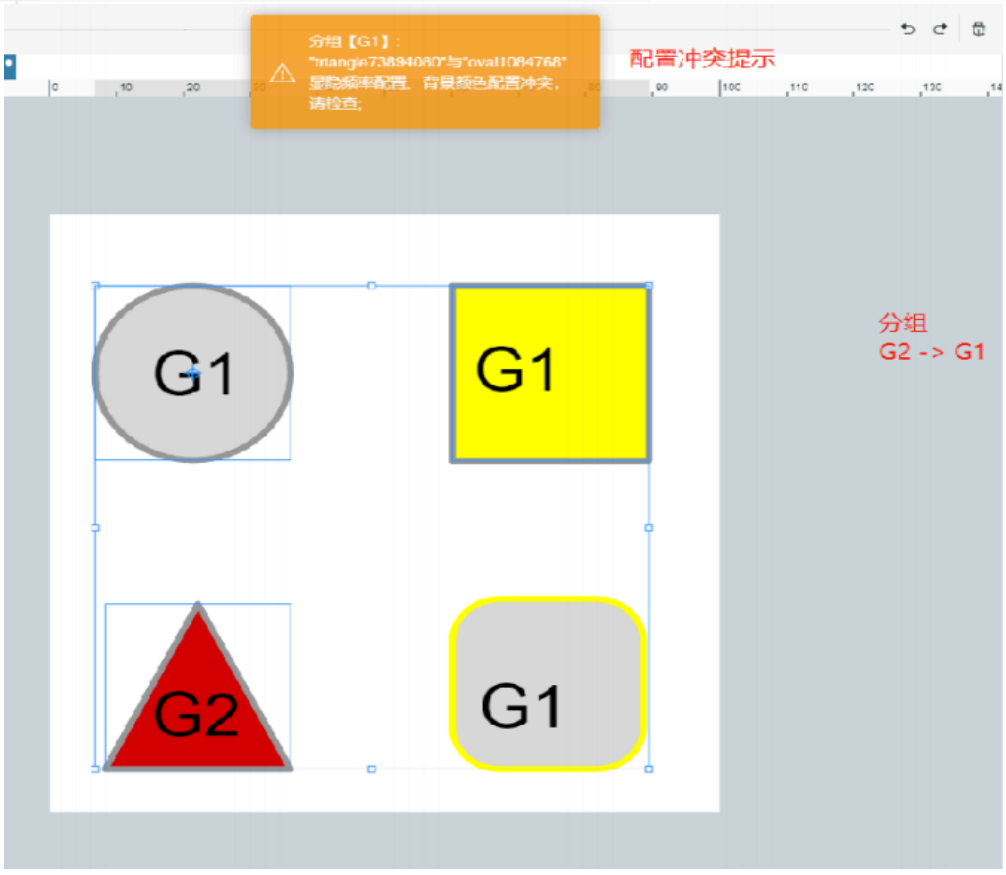

将分组内第二个元素配置绑点
G1
G1
G2
G2
属性
交互
基础
控制
位置
填充
渐进
Select
渐进色
#FFF
裁切方向
从下到上
裁切比例
1.00
图案叠加
背景
rgba(233,236,44,1)
边框
闪烁
其他
3.修改的背景颜色与分组内绑点元素配置不同，给出冲突提示并选中 这两个元素
2.修改分组内第一个元素的背景颜色为青色
数据
编辑
确定

③增加背景颜色绑点逻辑时（边框颜色、文本颜色和显隐频率同理）时，如果该组元素已存在绑点，给出多次绑点的冲突提示选中该元素和原本的主配置元素，并在控制台中打印

"oval1084768"与 "rect17398188"背景颜色配置多次绑点，请检查
③分组内同一配置多次绑点，提示多次绑点冲突并选中元素
②再对分组内另一个元素的背景配置绑点
G1
G1
G2
G2


Network
Elements
Console
Sources
Performance
Memory
Application
Lighthouse
top
Filter
4 Issues:
24 messages
5 user messages
No errors
7 warnings
No info
17 verbose
DevTools failed to load source map: Could not load content for http://localhost:8080/iplat/css/bootstrap.min.css.map: HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAILURE
DevTools failed to load source map: Could not load content for http://localhost:8080/iplat/console/designer/custom/libs/bootstrap.bundle.min.js.map: HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAILURE
misspelled option "enableSnippets"
misspelled option "enableBasicAutocompletion"
misspelled option "enableLiveAutocompletion"
请传入需要查找的设备
WARN: 分组配置冲突 "oval1084768"与 "rect17398188"背景颜色配置多次绑点，请检查 Wed Sep 07 2022 14:01:41 GMT+0800 (中国标准时间)
④控制台中打印冲突信息

④修改元素的分组（前缀）名时，需要校验边框颜色、背景颜色、文本颜色和显隐频率，且满足以下规则：
多次绑点时，显示多次绑点冲突提示；
该组元素背景颜色（边框颜色、文本颜色和显隐频率同理）无绑点时，新加入的元素如果也无绑点且配置不同，给出冲突提示；
该组元素背景颜色（边框颜色、文本颜色和显隐频率同理）原本有绑点时，新加入的元素如果无绑点但配置与绑点元素不同，给出冲突提示；
该组元素背景颜色（边框颜色、文本颜色和显隐频率同理）无绑点时，新加入的元素有绑点，给出默认值更新提示。



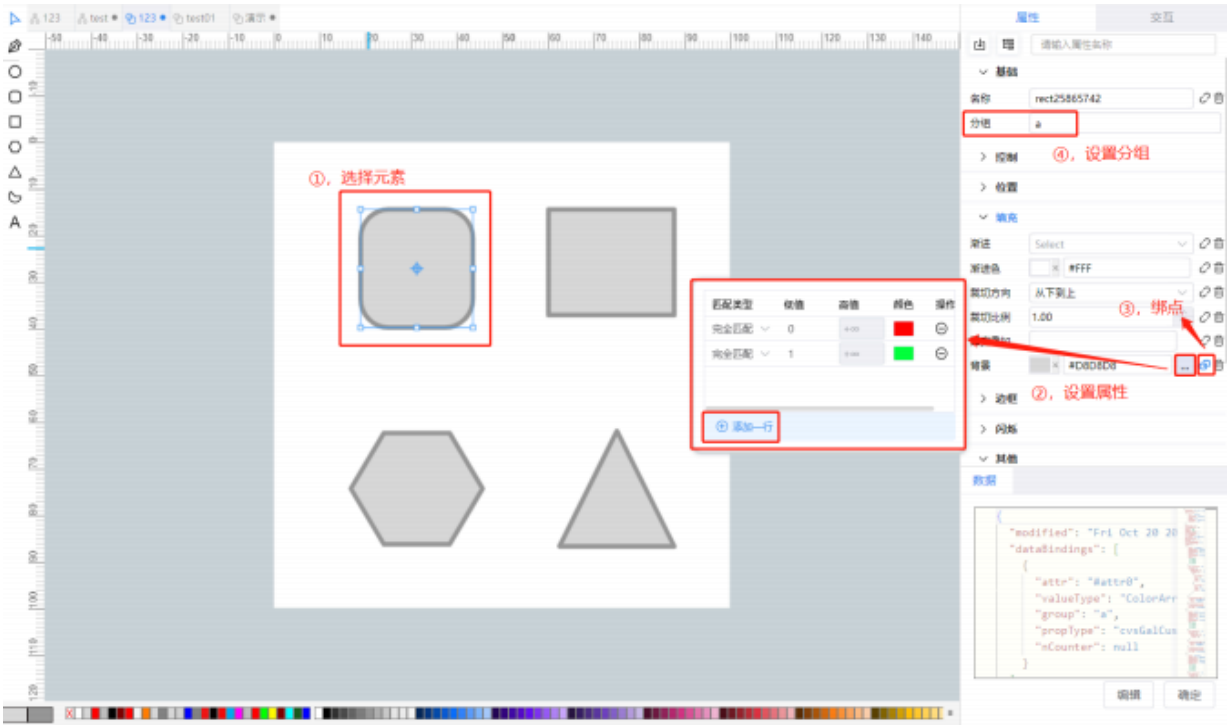
//使用场景满足两者之一就需要设置分组

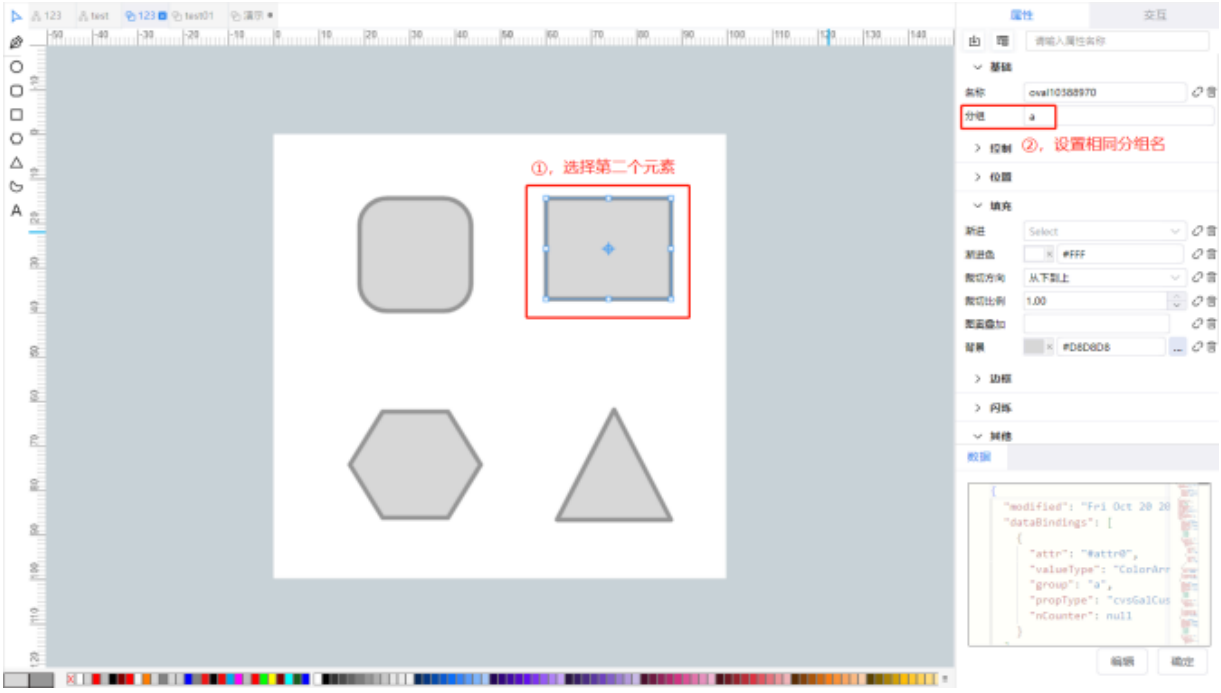
①关联性：

对于多个元素，若想要他们某个属性发生相同的变化，将他们设置相同的分组名后只需要对其中一个元素的该属性绑点或者再脚本中直接通过data.a("分组名.属性",值)设置

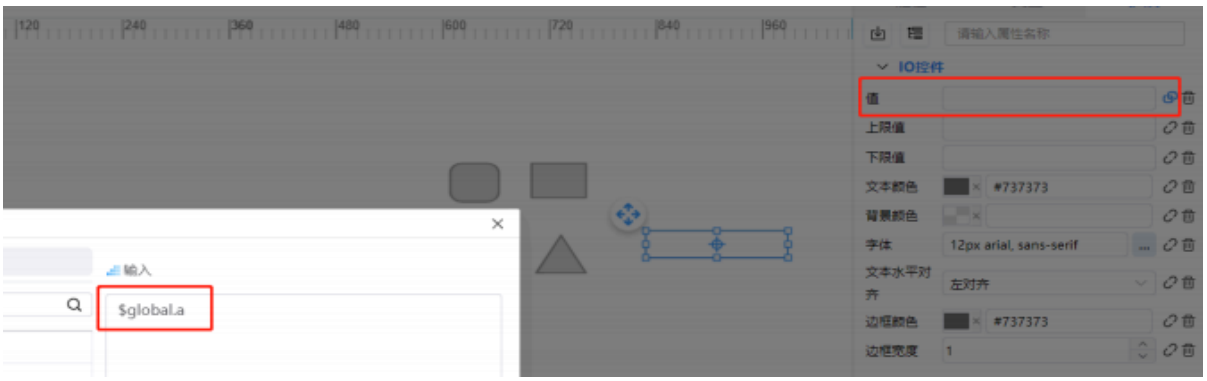
示例1：设置两个元素相同的分组，其中一个元素设置了背景变化并关联了点，而另一个点没有配置背景

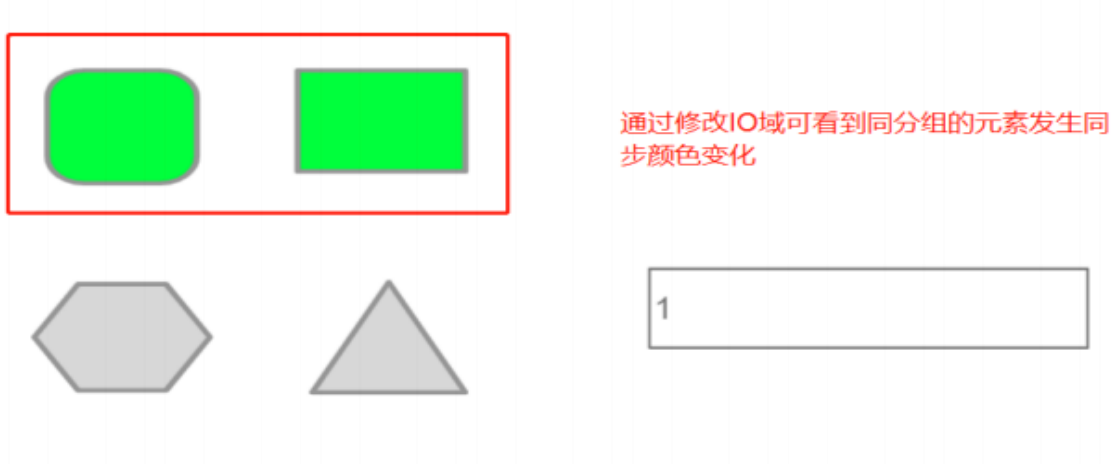
1）设置分组和属性





2）设置关联数据值;画面添加IO域,IO与值绑定矩形背景色的绑点;运行画面,修改IO域的值,可以看到矩形和正方形背景色同步变化





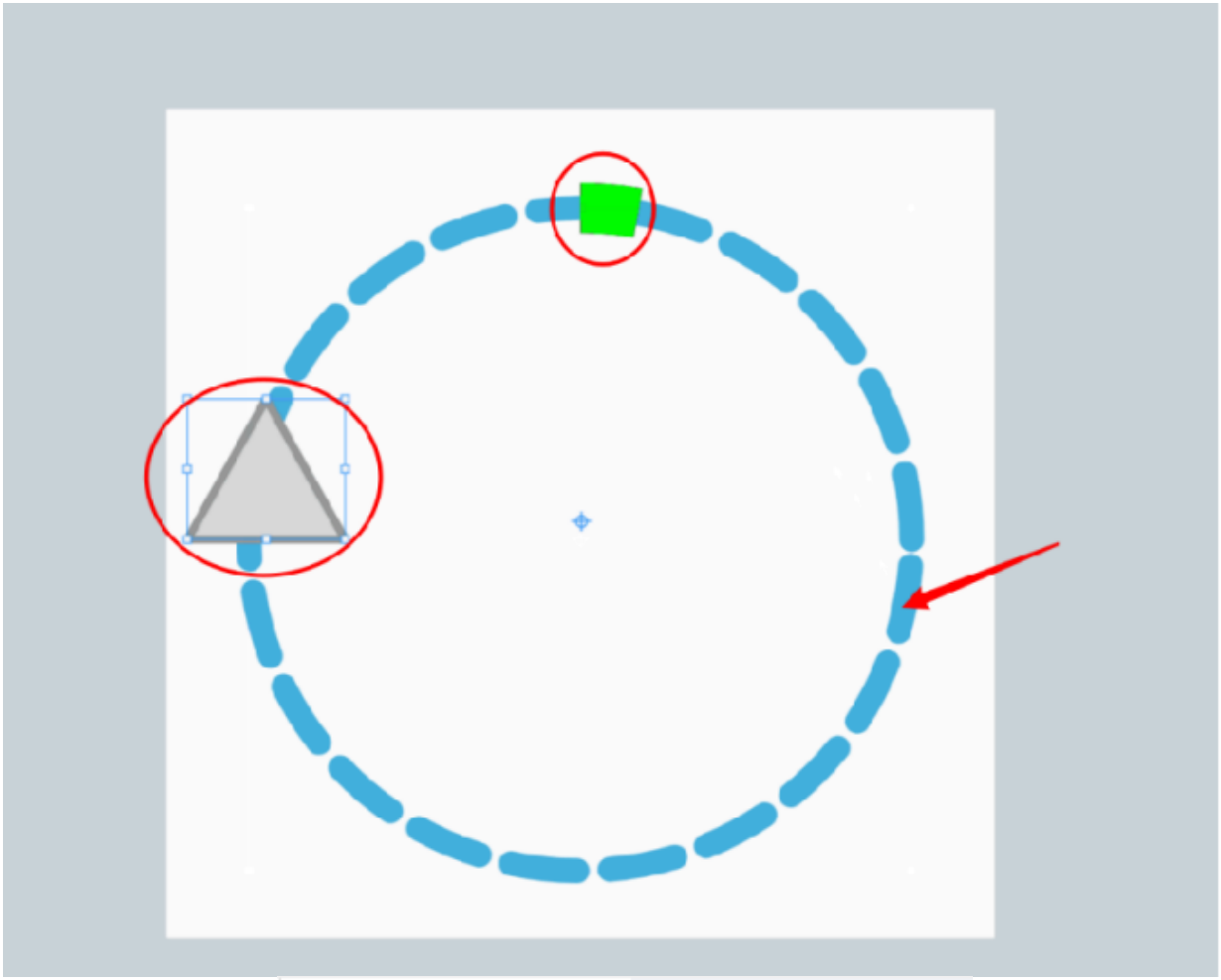
优点：通过将多个元素设置为同一分组名，只需要对一个元素设置属性并绑点，当获取点的数据时，与该元素前缀相同的其他元素与其发生相同的变化。这样就可以设置多个元素的同一属性只需要通过对其中一个元素绑点一次实现，简化了重复绑点的操作。

缺点：如果想要对其中一个元素的某个属性作单独修改，不能使用data.a("分组.属性","值")的方式，否则所有的元素都会变。这时可以单独绑定attr，然后data.a设置attr的方式

②排他性：

直接操作某个元素，需要设置唯一的分组名来区分
示例2：设置三个元素不同分组，通过脚本可以分别直接设置和获取他们的属性值

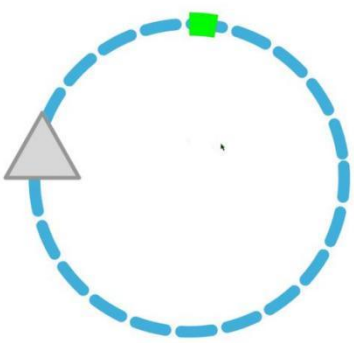
在下面的示例中A圈旋转90度触发B块旋转90度，B块旋转90度触发C三角消失;三者的前缀均不同

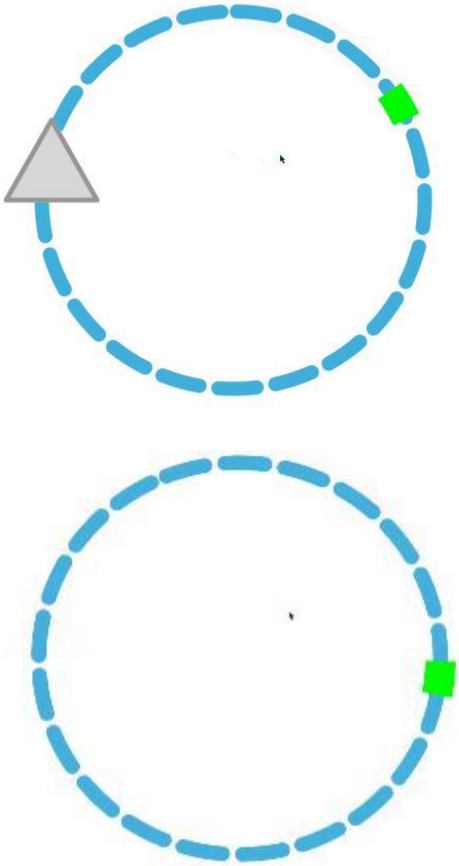


脚本：

```
    var rotation1 = - Math.PI/2;
    var rotation2 = Math.PI*2;
    var mid = 0;
    var c_visible = data.a("ttt.visible");
    // var c_visible = data.a("ttt.visible");//这种是拿不到的 还没加载进来
    undefined setInterval(function(){
        //A圈不断旋转
        rotation2 = rotation2 - Math.PI*2/(10*40);
        data.a("circle1.rotation",rotation2);
        console.log("--------"+rotation1+"======="+rotation2+"------"+c_visible)
        //圈旋了90后 mid=1 触发B块旋转
        if(rotation2 <= Math.PI*5/3 && mid==0){
            mid = 1;
        }
        if(mid == 1){
            rotation1 = rotation1 + Math.PI*2/(10*25);
            data.a("arc0.rotation",rotation1);
            if(0 <= rotation1 && rotation1 < Math.PI*3/2 && c_visible ==
            undefined){
            data.a("ttt.visible",false); //B块旋转90后 C三角块隐藏
            c_visible = 0;
        }
        if(rotation1 ```= Math.PI*3/2){
            rotation1 = - Math.PI/2;
        }
    }
    if(rotation2 <= 0){
        rotation2 = Math.PI*2;
    }
    },100);
};
```

演示效果：





优点：在脚本中直接访问任意元素的属性，可以灵活修改元素属性，实现各种动画效果

缺点：需要自己配置不同的分组名

③灵活性和便利性：

图元中有的元素可以设置相同分组名，有的元素可以分组名设置唯一，有的元素可以不设分组，结合排他性和关联性可以灵活修改元素属性，减少绑点操作和脚本代码量

优点：根据用户的配置，可以完成各种需求

不使用分组的场景

当不需要使用元素的扩展属性（如下图的显隐、边框、背景）而且不需要在脚本代码中通过data.a来直接设置或修改某一元素或部分元素的属性时，应当不设置使用分组

优点：配置简单

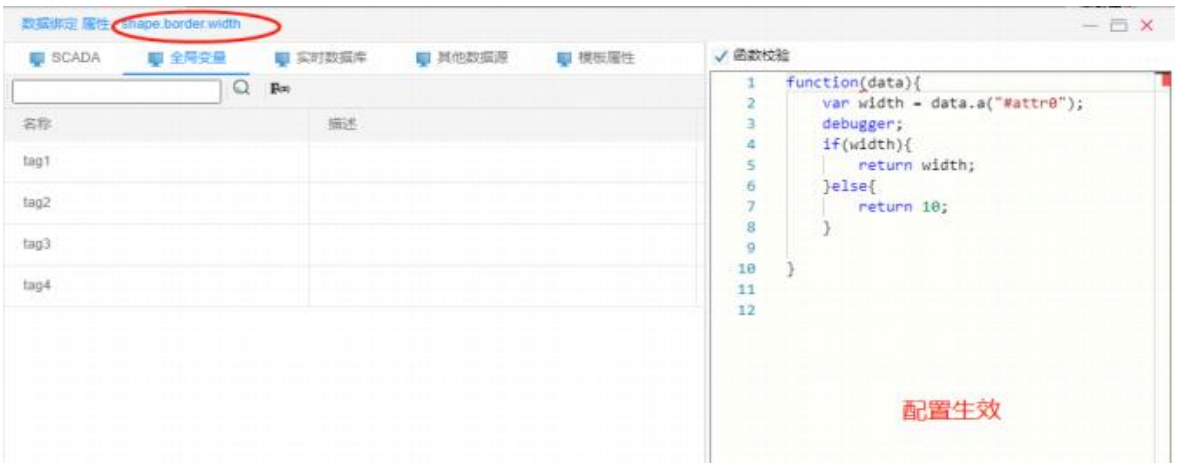
缺点：不适用于复杂场景，不够灵活

分组使用时需要注意

同一个图元不同元素配置扩展属性（显隐频率、背景配置、边框配置）后需要配置分组才能生效（因为我们自己扩展的属性需要通过data.a来设置）。但是在这种情况下配了分组的元素除了扩展属性外其他属性如果配置了**直接关联tag点**的模式，就可能引起该配置无效。

规避问题

如果配置了扩展属性关联tag点，需要配置分组。此时还想要其他的属性生效，可以通过function模式来规避,例如:新建自定义属性"#attr0",绑定需要的tag点(也可以是类模板属性);在边框宽度属性的函数输入框里,通过data.a("attr0")的方式读取tag点;



5.27 图元引用列表

概述

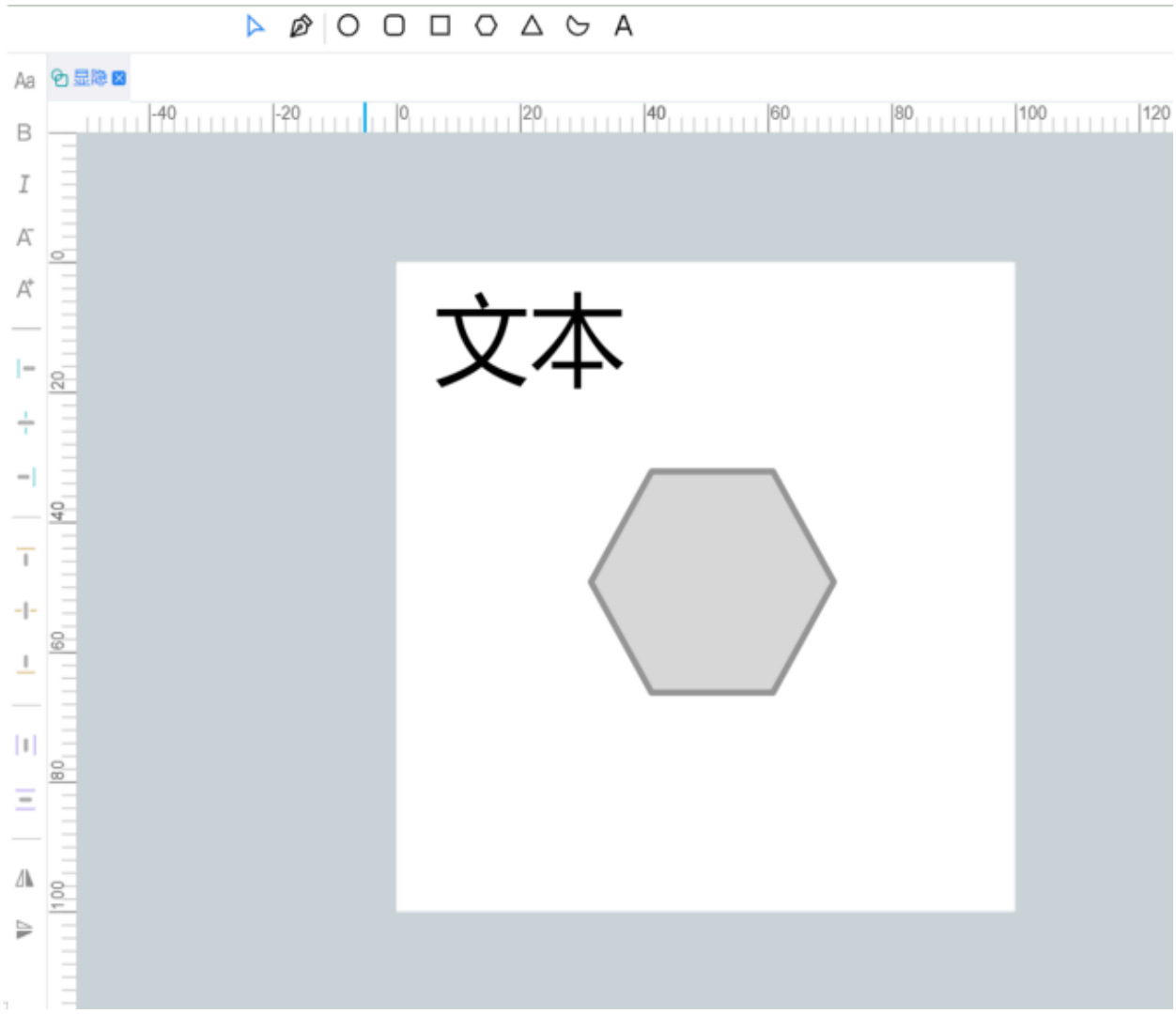
在绘制画面的时候，有时候想查看某个图元具体被哪些画面所引用。但是一个个画面去

找，费事费力，而且容易出现遗漏。所以提供一个可以查看的列表，方便画面的制作。

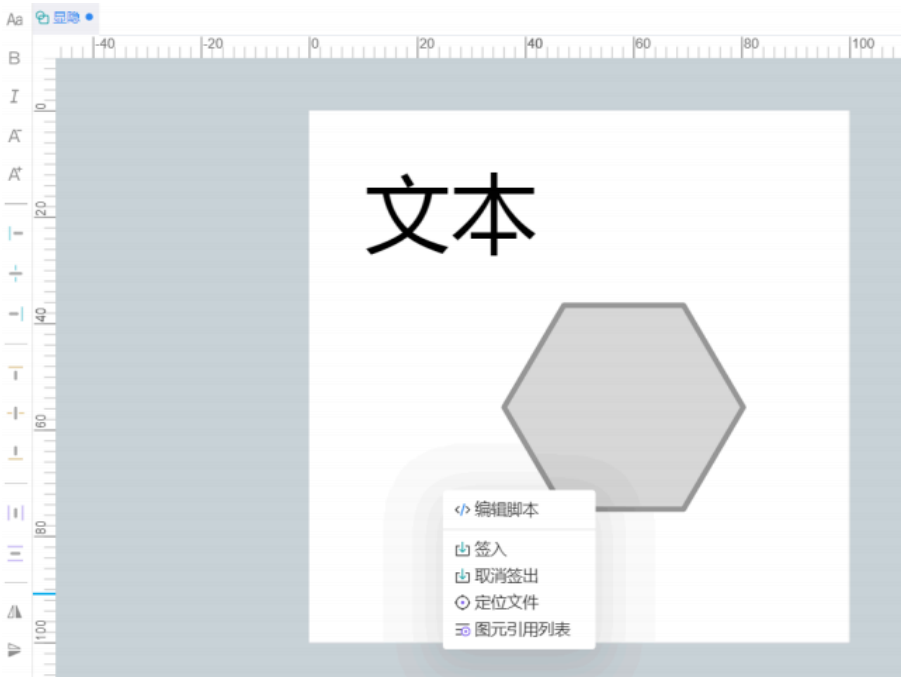
功能介绍

查看引用

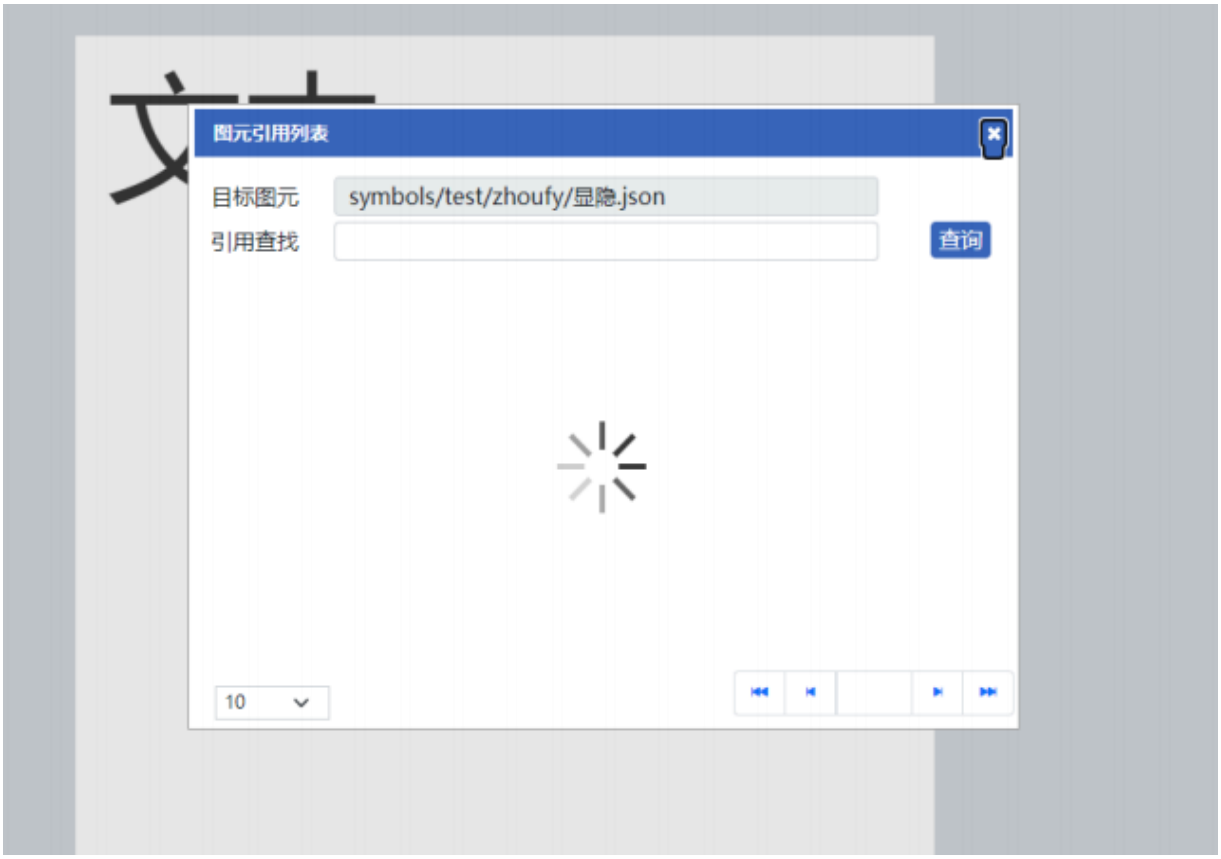
1. 打开想要查看的图元。

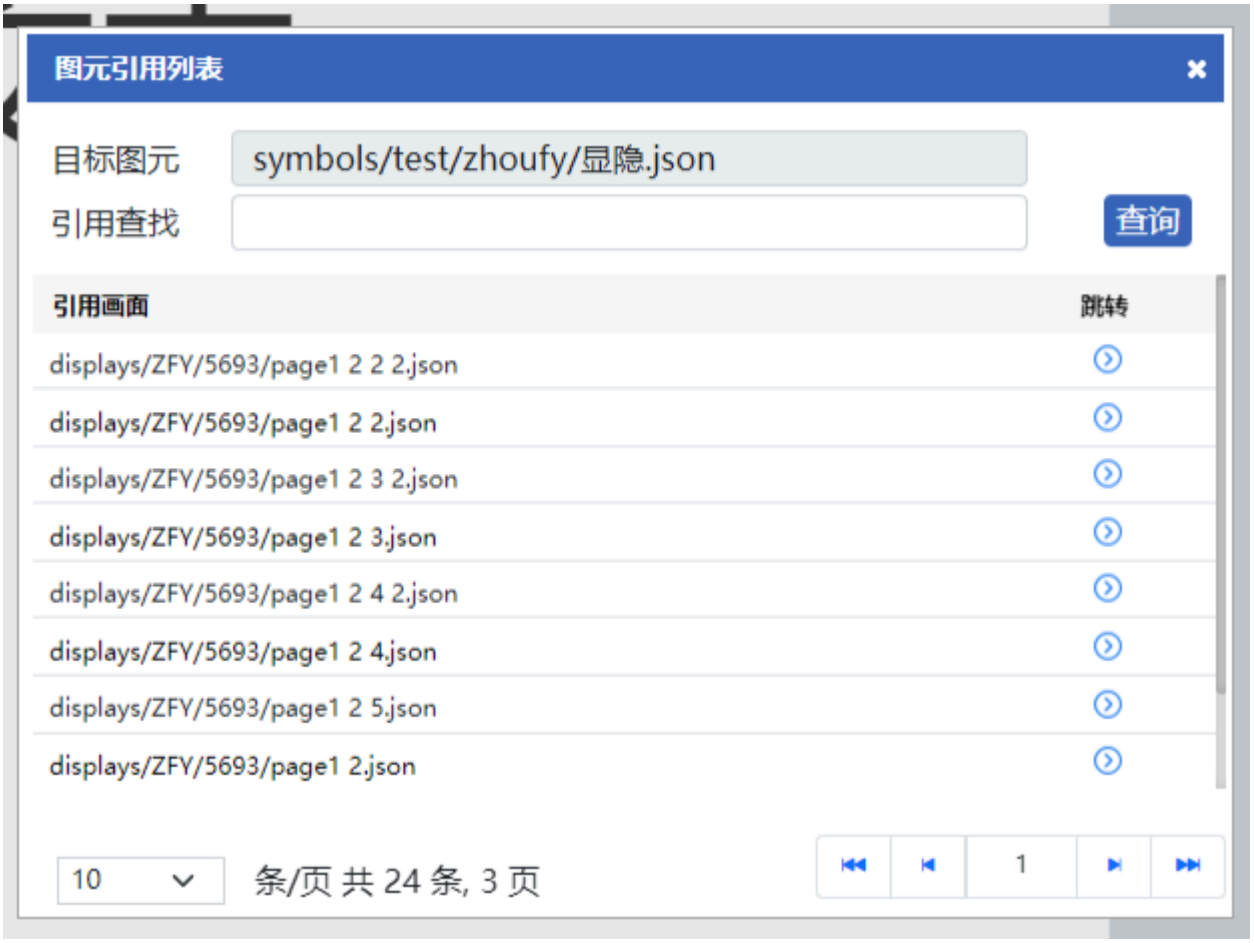


2. 右键图元空白处，然后点击【图元引用列表】


文本
编辑脚本
签入
取消签出
定位文件
图元引用列表

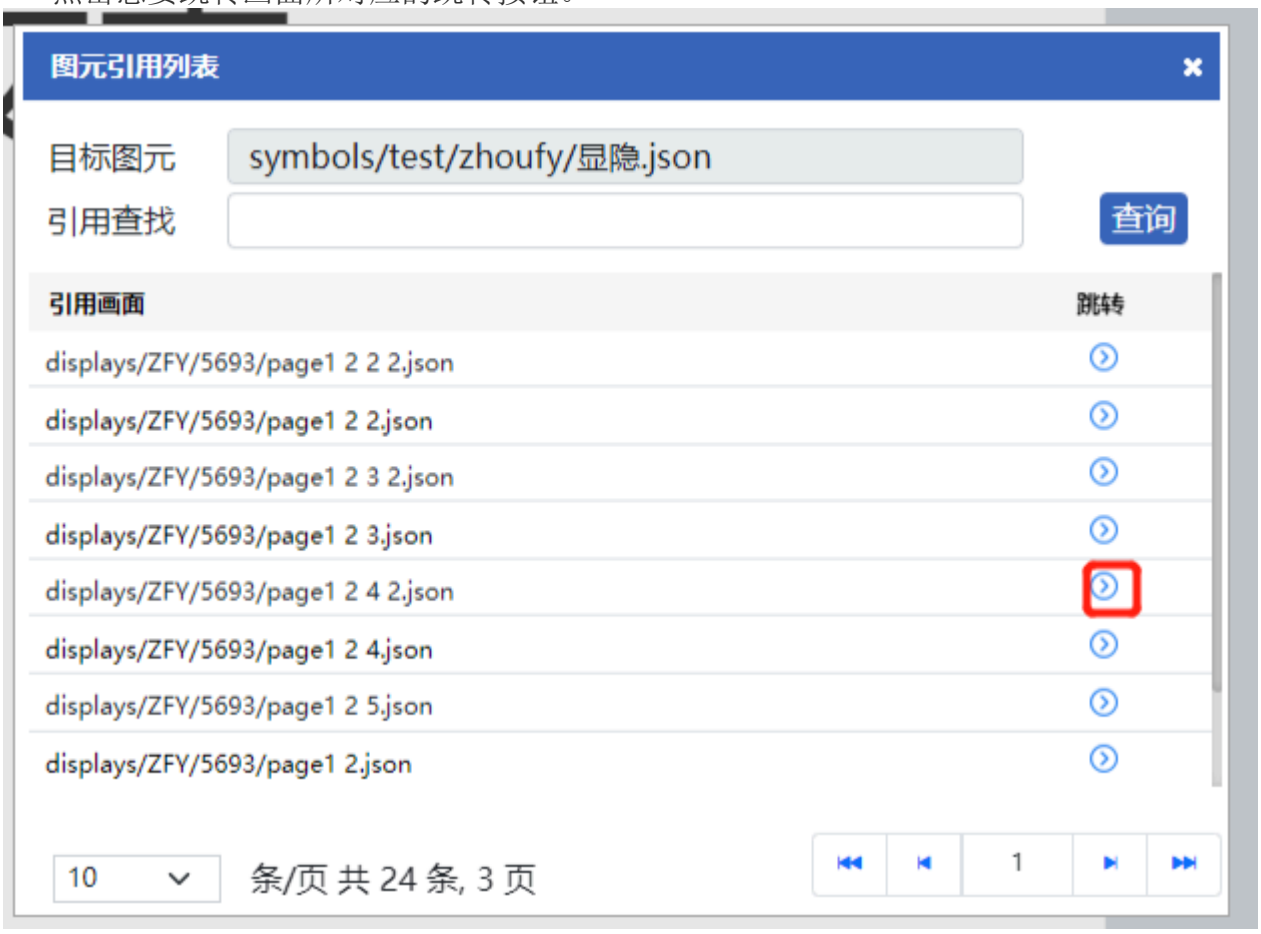
3. 等待数据加载

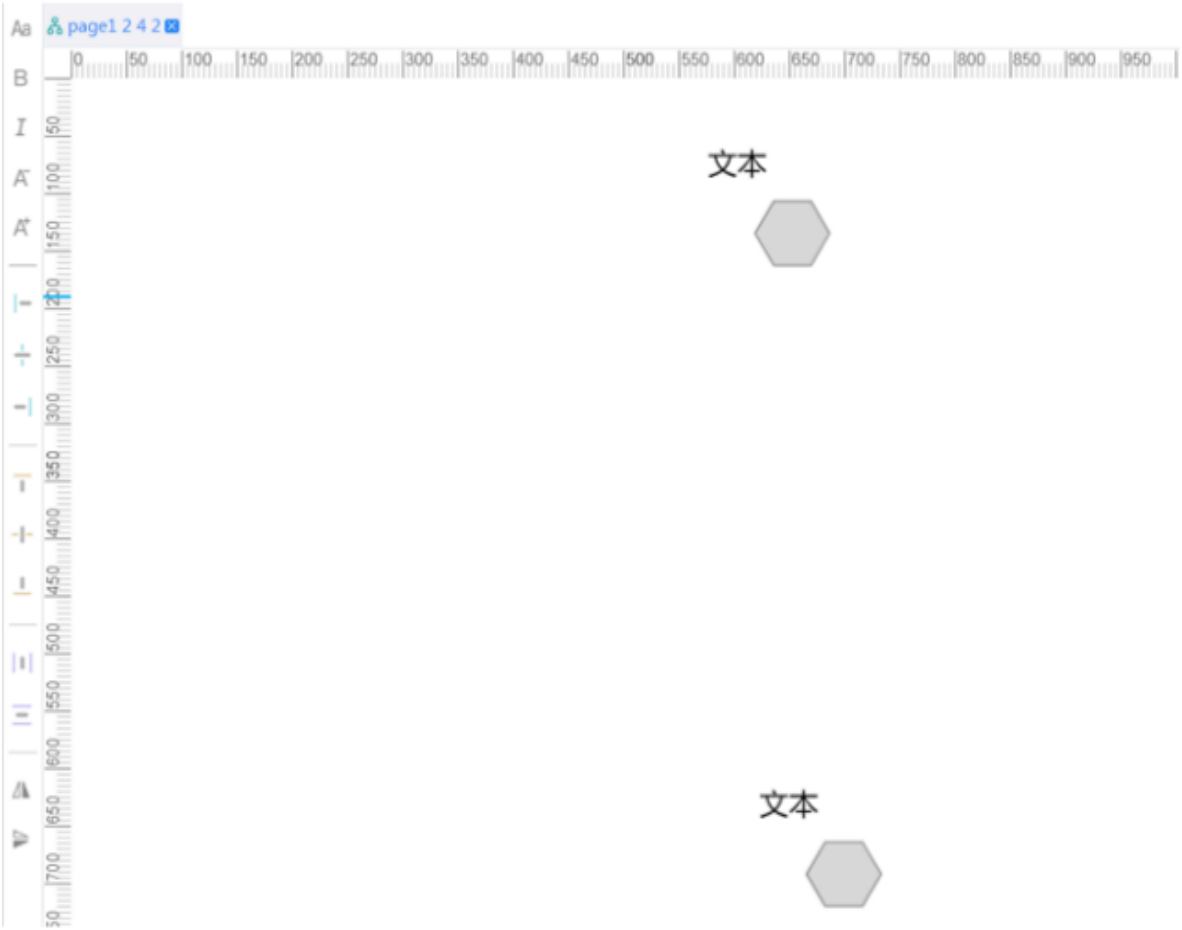

图元引用列表
目标图元
symbols/test/zhoufy/显隐.json
引用查找
查询
10



页面跳转

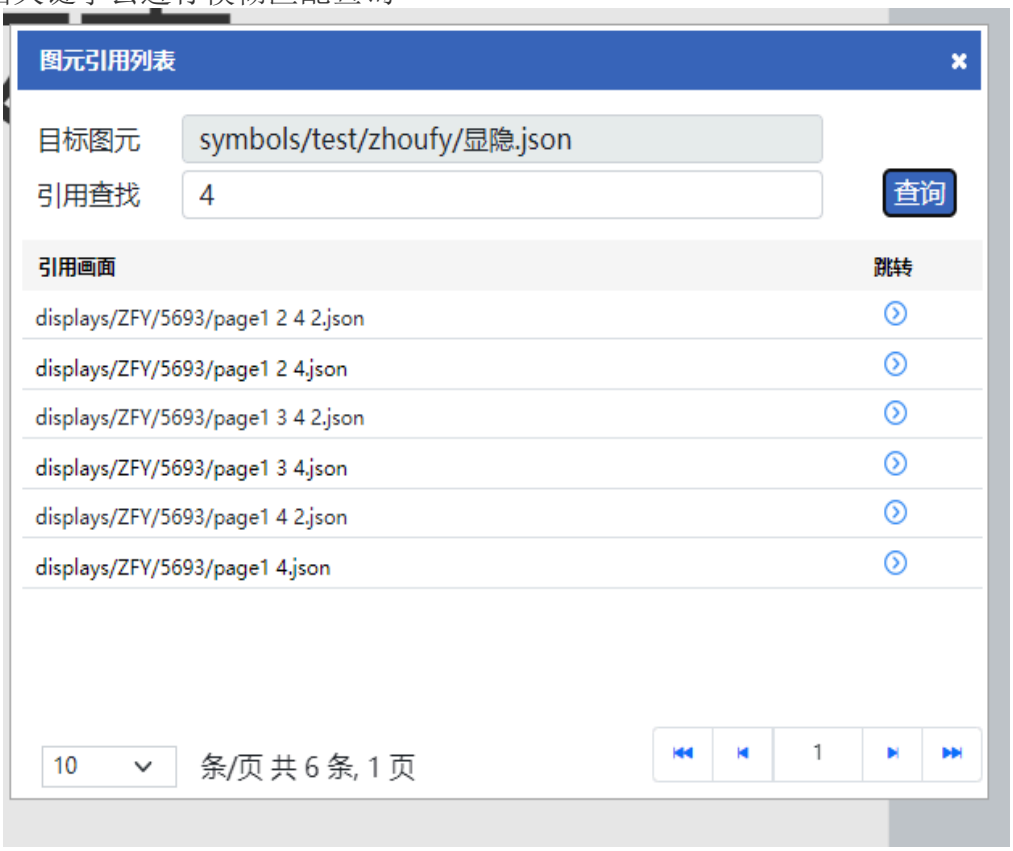
- 点击想要跳转画面所对应的跳转按钮。





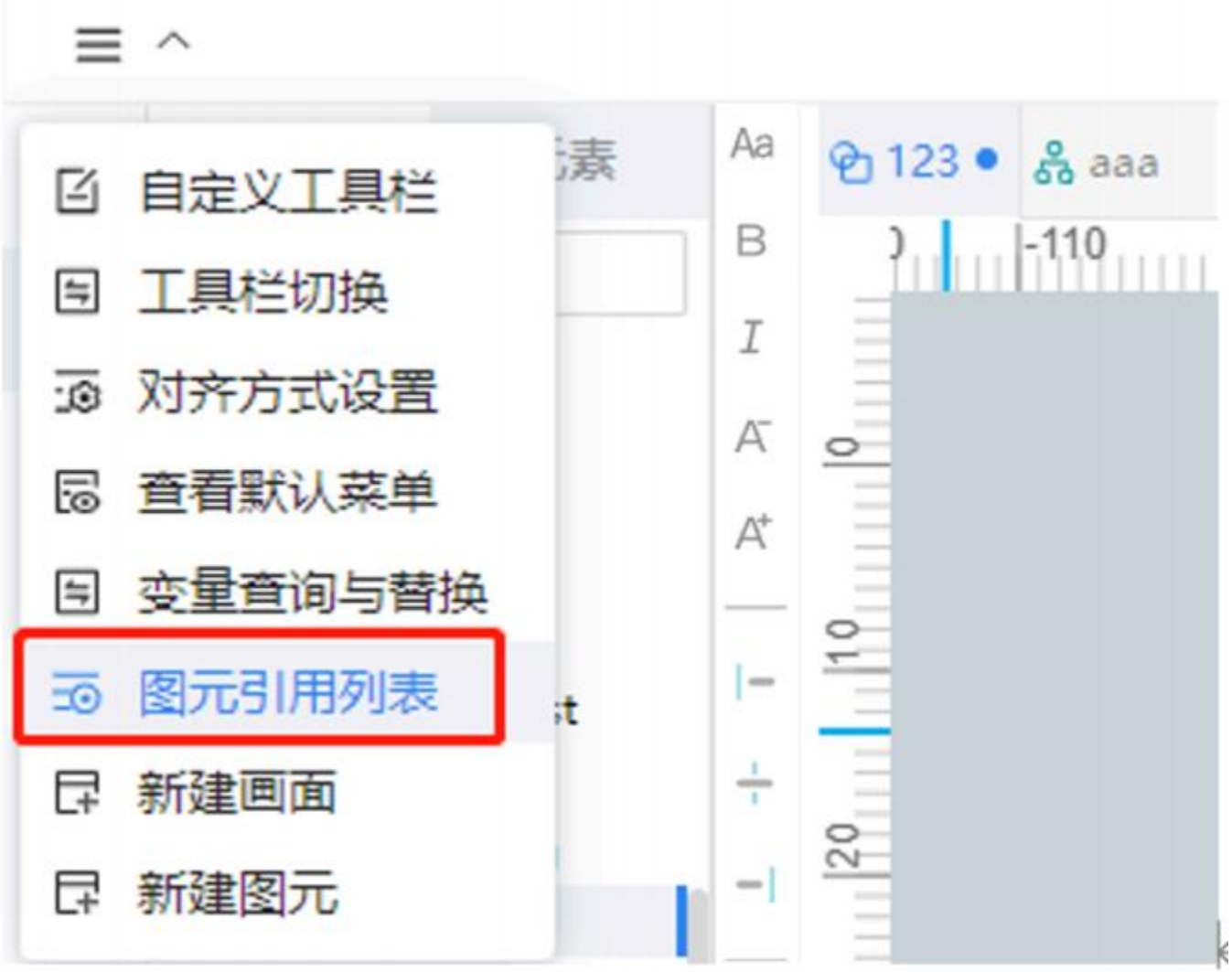
引用查找

可以根据关键字去进行模糊匹配查询

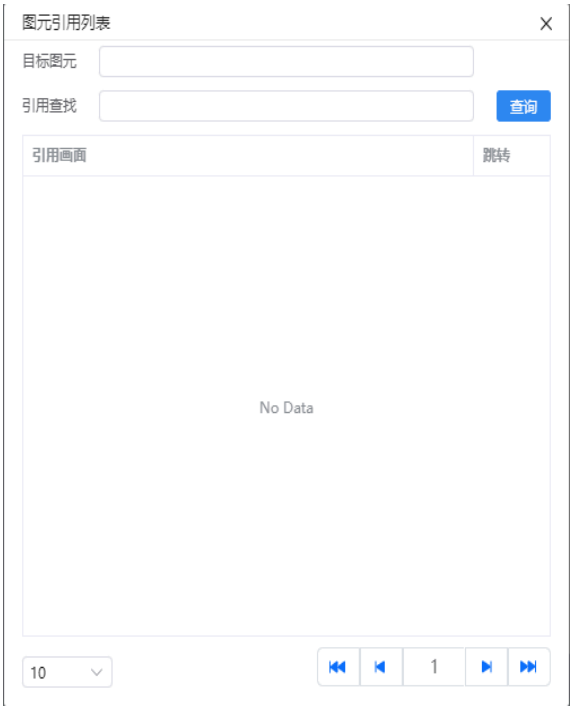


文件→图元引用列表入口

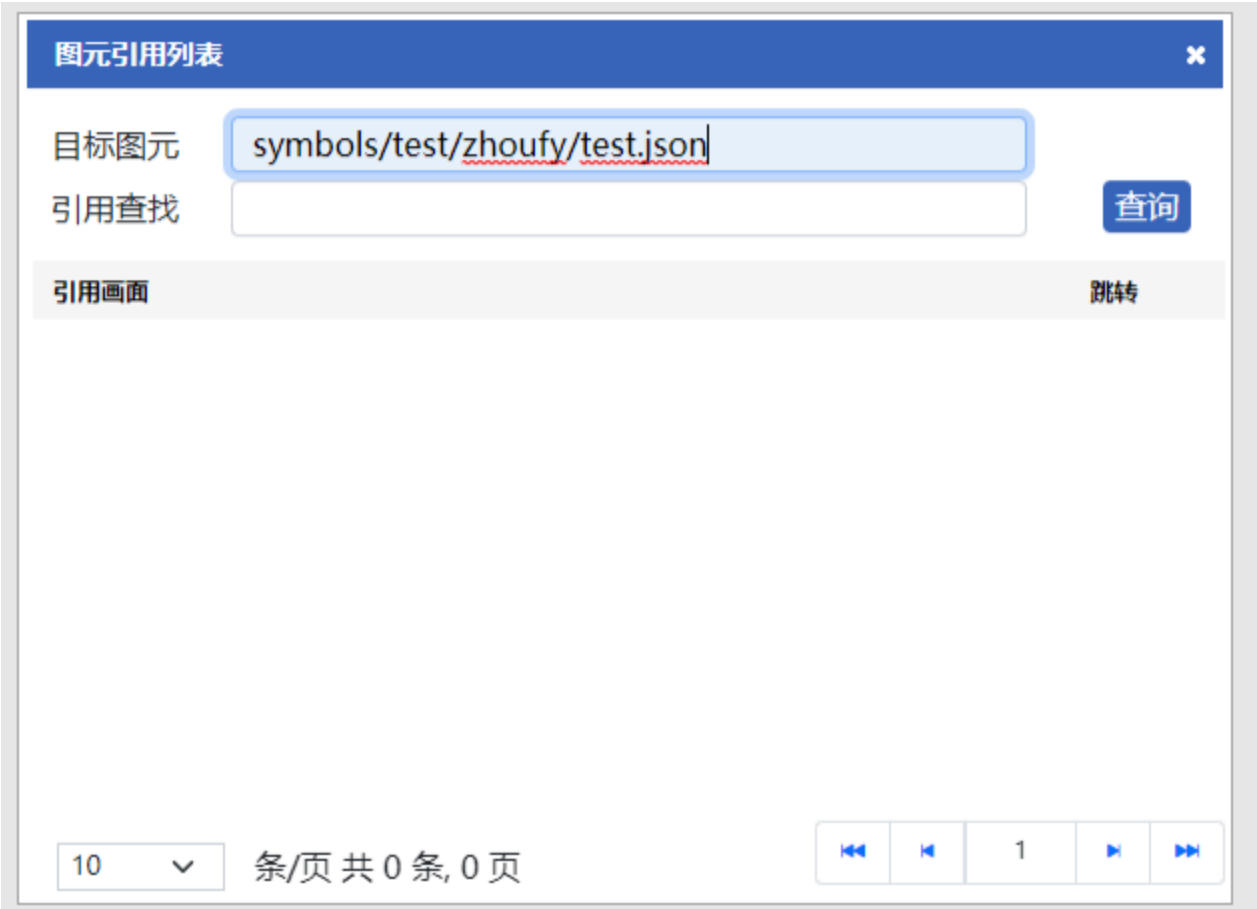
在左上角的配置中，也存在图元引用列表的入口。

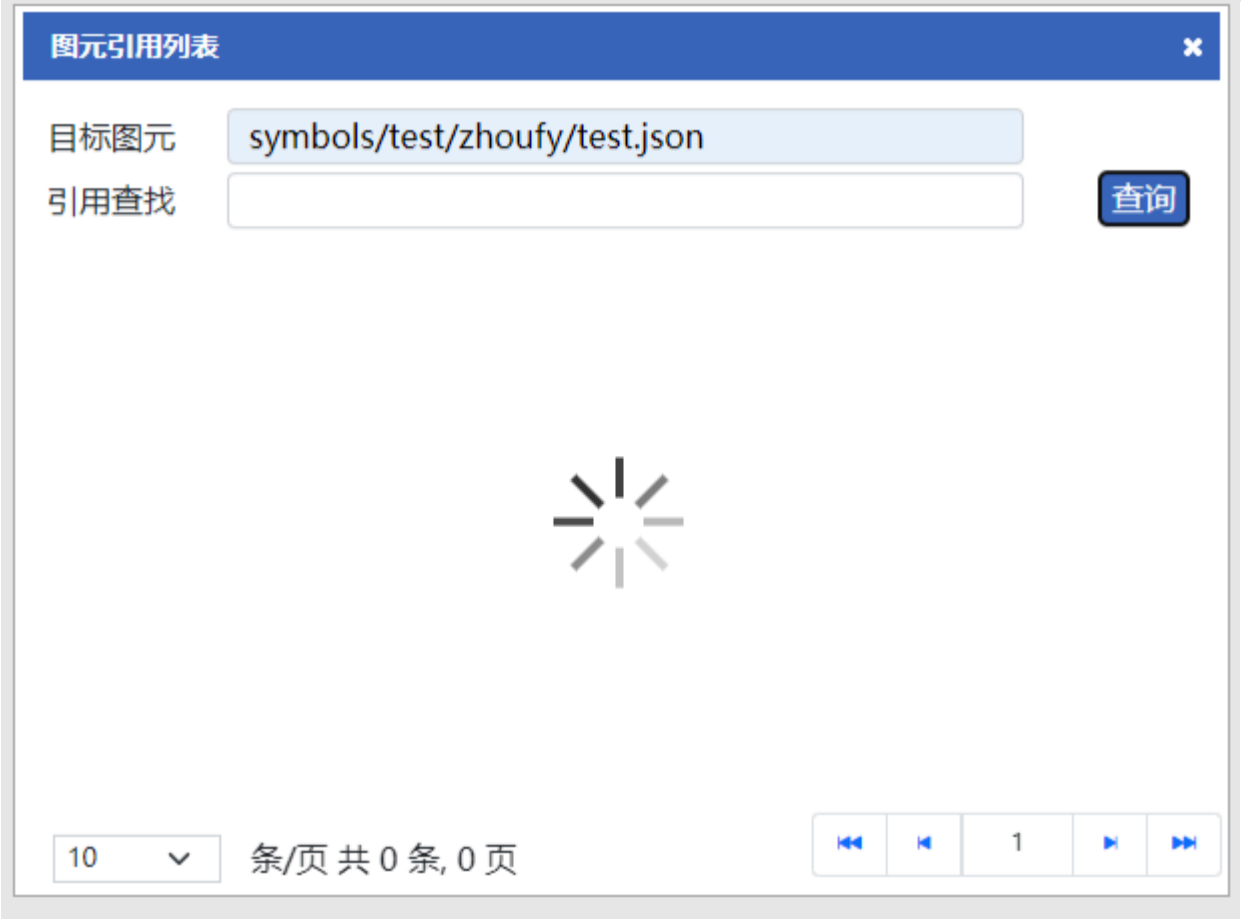


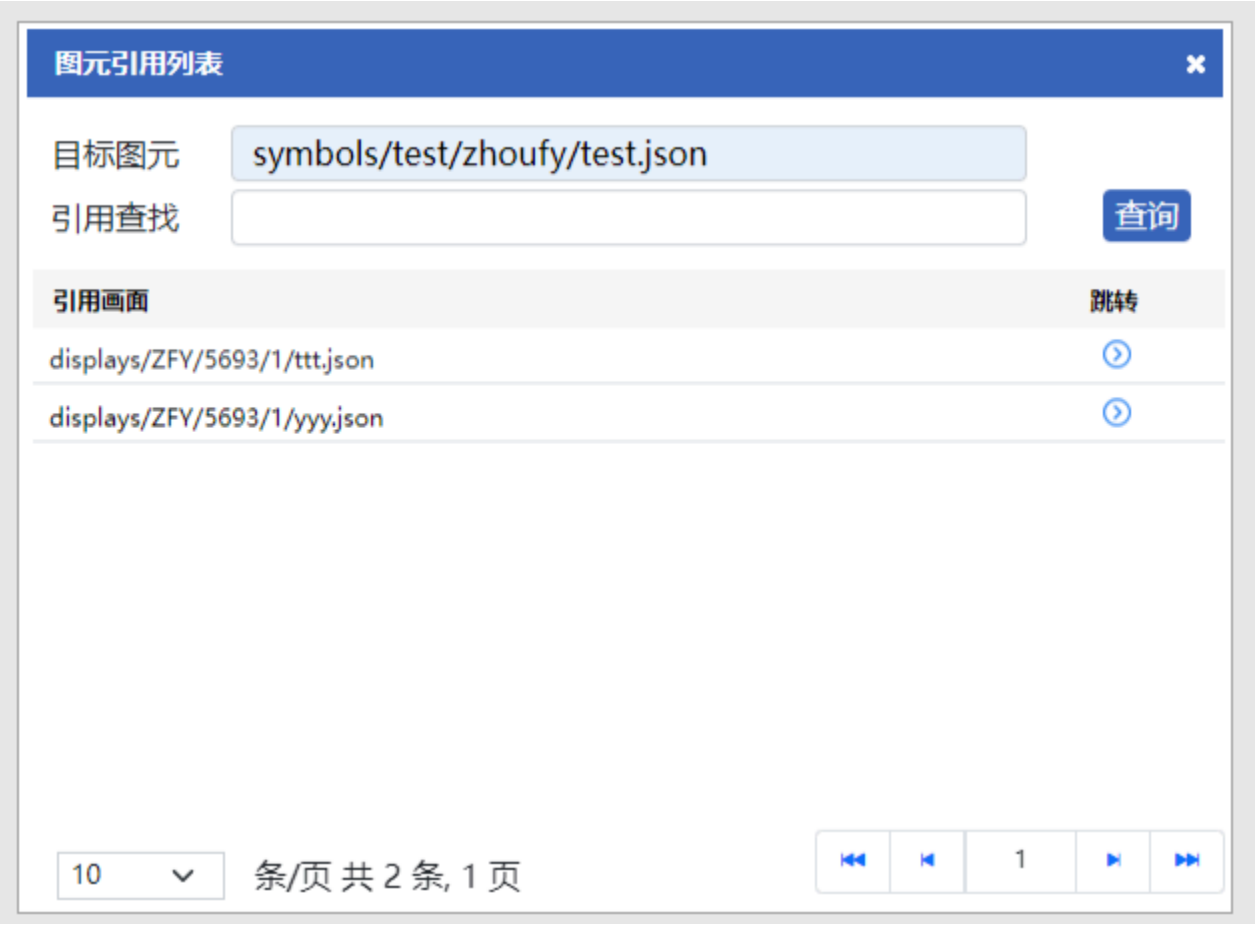
通过该入口进入的界面，其目标图元需要手动输入（【/symbols】开头到【.json】结尾的路径）



输入后，点击查询，并等待数据加载



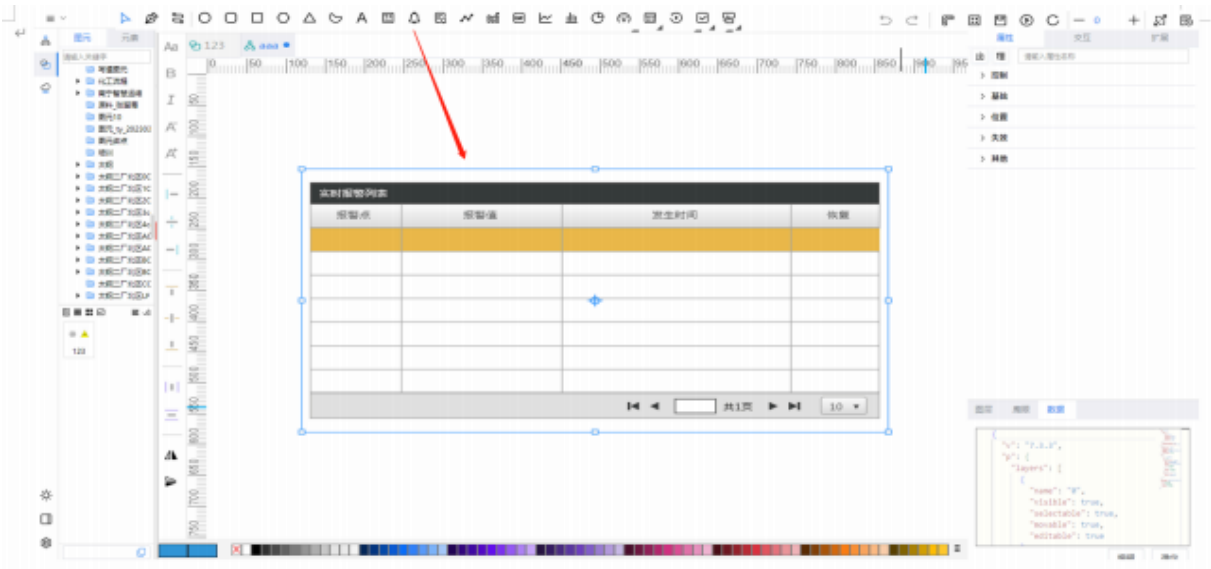




5.28 实时报警控件

使用实时报警控件

1.打开2D设计器，新建一个画面，在工具栏上点击“实时报警”控件，然后在画面上空白的地方拖拽，以便将“实时报警”控件放置在画面上。



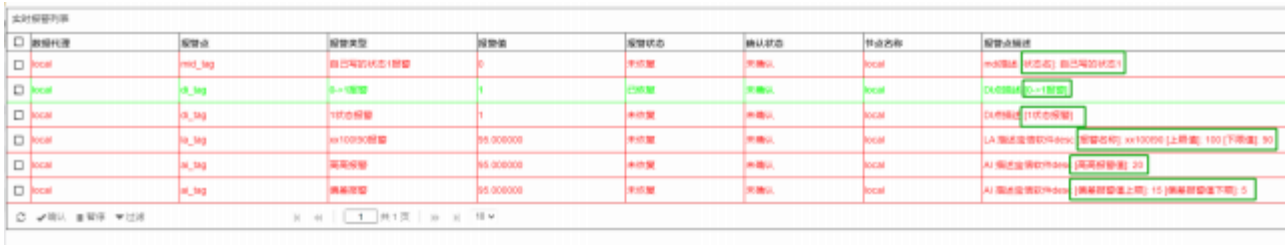
2.保存页面之后，点击“预览”，即可看到当前系统内的实时报警。（列表左侧小框可选择对应报警信息的关注/取关）

3. 鼠标悬停的背景颜色可以通过基本配置进行设置

4. 报警控件的报警描述

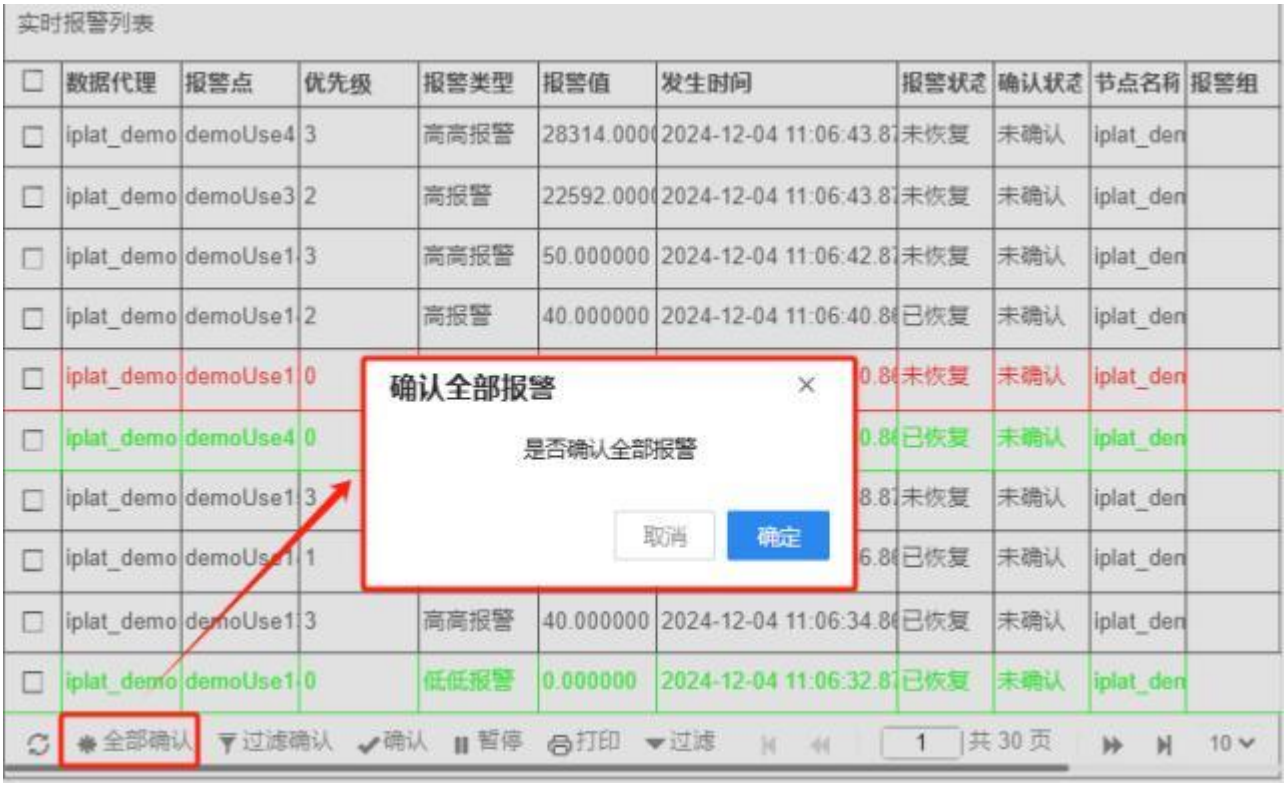
报警描述由2部分内容拼接而成：

“Tag点的描述”+报警的上下限配置信息，如图所示



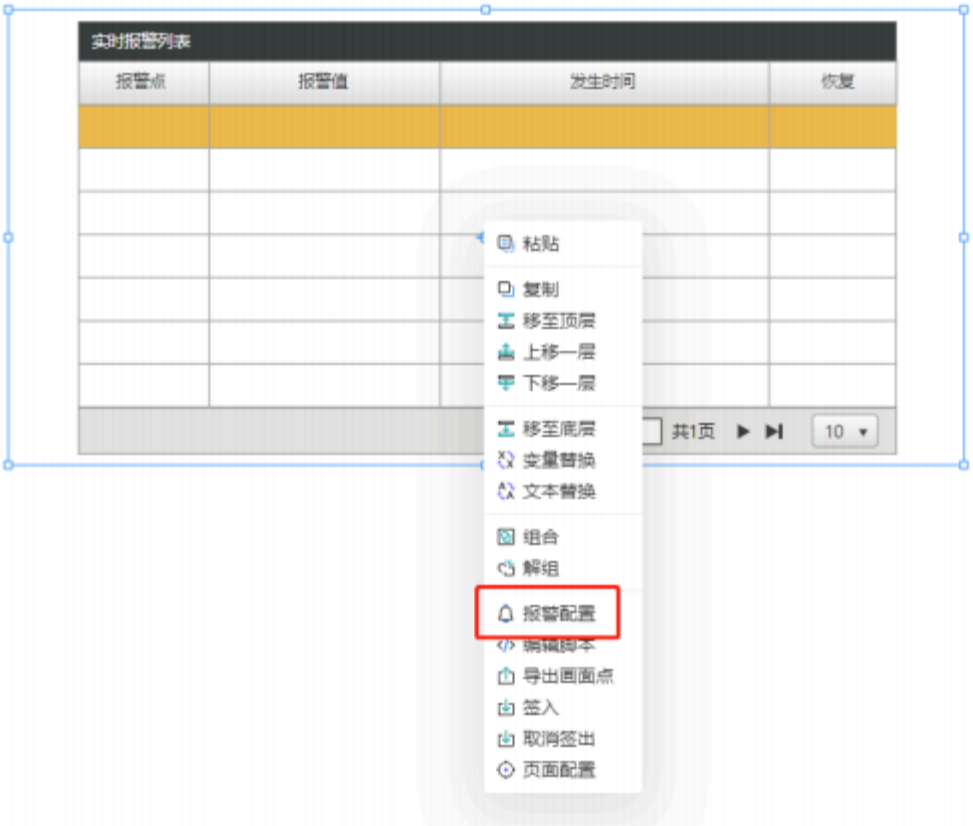
5. 使用时，可以配置过滤条件，根据不同条件进行报警信息筛选查看；

6. 支持一键确认全部报警状态，如下图所示；

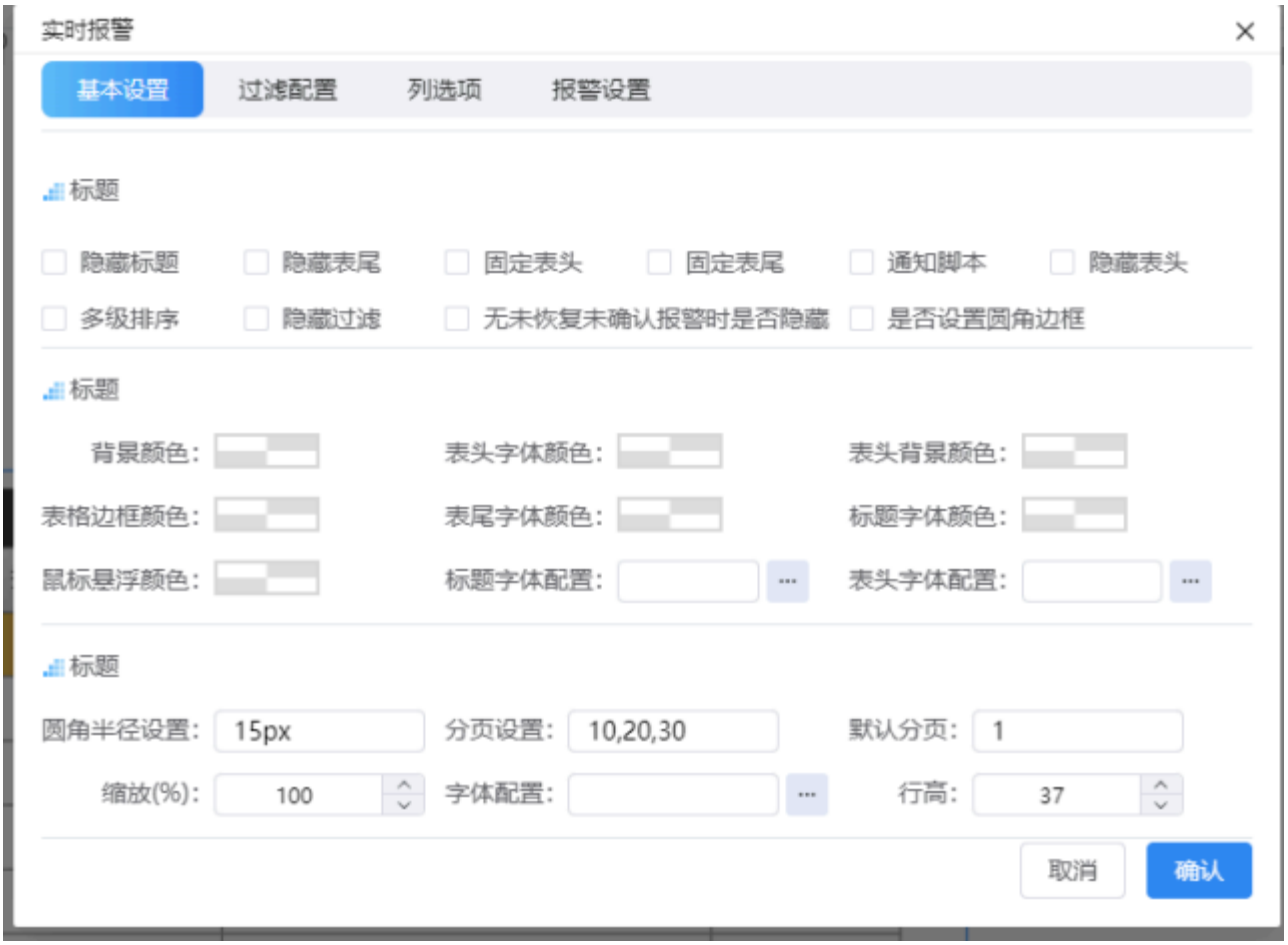


配置实时报警控件

•在“实时报警”控件上右击，点击“实时报警配置”，进入“实时报警”控件配置页面。



•基本设置。


实时报警
基本设置
过滤配置
列选项
报警设置
标题
隐藏标题
隐藏表尾
固定表头
固定表尾
通知脚本
隐藏表头
多级排序
隐藏过滤
无未恢复未确认报警时是否隐藏
是否设置圆角边框
标题
背景颜色:
表头字体颜色:
表头背景颜色:
表格边框颜色:
表尾字体颜色:
标题字体颜色:
鼠标悬浮颜色:
标题字体配置:
表头字体配置:
标题
圆角半径设置:
15px
分页设置:
10,20,30
默认分页:
1
缩放(%):
100
字体配置:
行高:
37
取消
确认

•过滤配置。

实时报警
基本设置
过滤配置
列选项
报警设置
启用报警区域过滤
启用报警状态过滤
启用报警时间过滤
启用报警优先级过滤
启用变量名称过滤
启用报警类型过滤
启用代理数据过滤
启用报警点描述过滤
启用子系统过滤
启用分组过滤
取消
确认

•列选项。

•报警设置。

实时报警控件接口

•脚本配置报警过滤

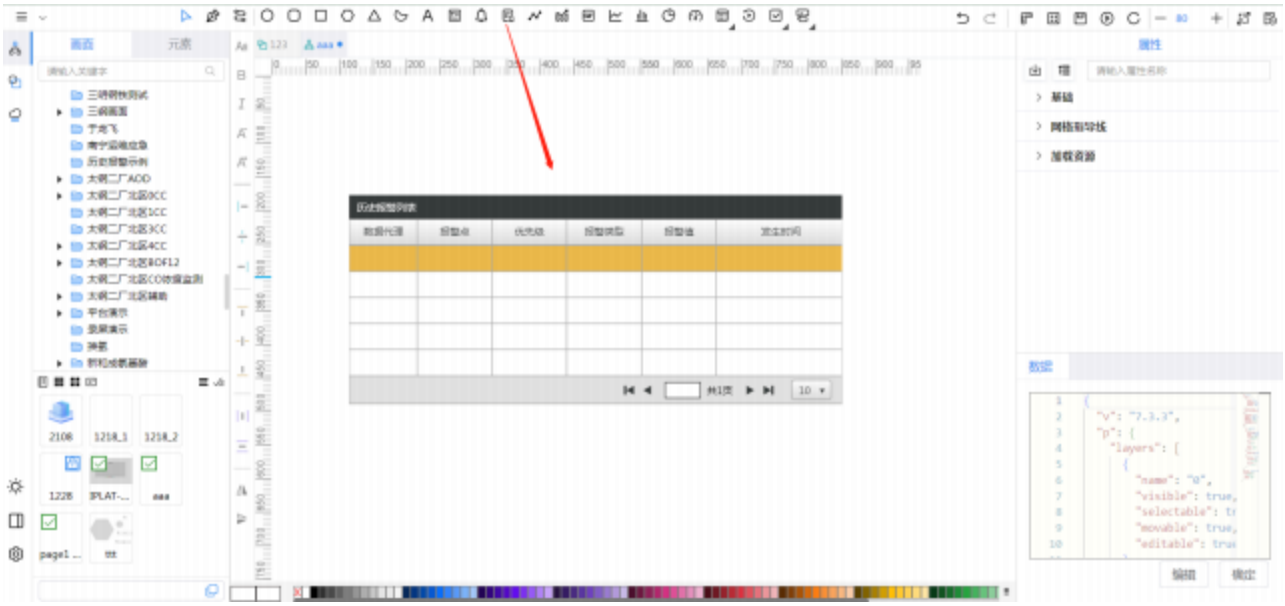
属性名	数据类型	
enableFilterArea	number	0或1，0表示
enableFilterStatus	number	0或1，0表示
enableFilterTime	number	0或1，0表示

enableFilterPriority	number	0或1，0表示
enableFilterTag	number	0或1，0表示
enableFilterType	number	0或1，0表示
enableFilterAgent	number	0或1，0表示
enableFilterDesc	number	0或1，0表示
enableFilterSubsysname	number	0或1，0表示
enableFilterGroup	number	0或1，0表示
filterAreaSetting	array	报警区域过滤
filterStatusSetting		string
filterTimeSetting	string	报警时间过滤
filterPrioritySetting	string	报警优先级过
filterTagSetting	string	变量名称过滤，
filterTypeSetting	string	报警类型过滤，
filterAgentSetting	string	数据代理过滤，
filterDescSetting	string	报警点描述过
filterSubsysnameSetting	string	子系统过滤，
filterGroupSetting string	分组过滤，输入分组路径，以/a/b/c 的形式，将会过滤出该组以及下属所有子组的报警信息。	

5.29 历史报警控件

使用历史报警控件

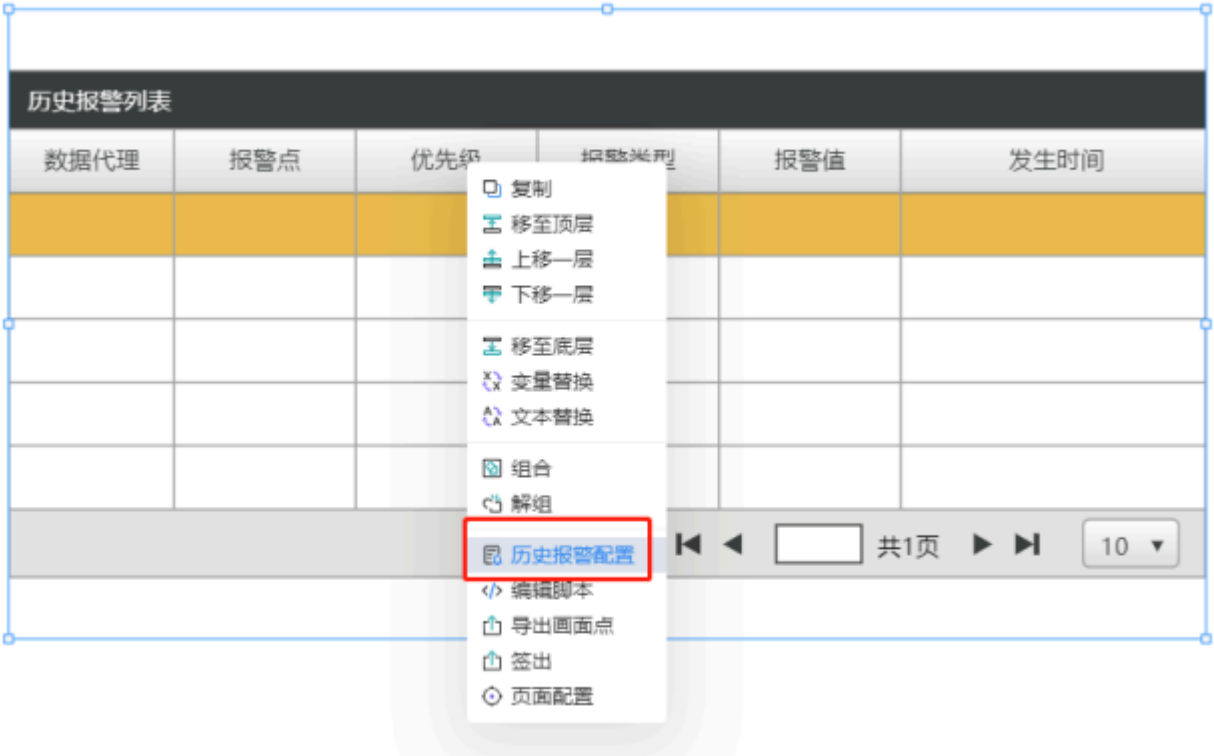
1. 打开2D设计器，新建一个画面，在工具栏上点击“历史报警”控件，然后在画面上空白的地方拖拽，以便将“历史报警”控件放置在画面上。



2. 用户通过搭建历史报警查询条件区域，在脚本中调用历史报警查询接口，实现完整的历史报警控件。

配置历史报警控件

1. 在“历史报警”控件上右击，选择“历史报警配置”，打开“历史报警”控件配置页。



2. 历史报警控件有列选项、报警设置、通用配置可供配置

•列选项

◦历史报警控件默认只显示“数据代理”、“报警点”、“优先级”、“报警类型”、“报警值”、“发生时间”、“报警状态”、“确认状态”、“节点名称”9列。

▪通过取消/勾选“列选项”页“启用”复选框从“历史报警”控件中增减列。

▪选中某一已经启用的列，点击“上移”、“下移”按钮，可以调整列的位置顺序。

▪启用某一列后，可以点击该列中“替代名称”编辑框，编辑该列的列名以替代原有列名

•报警设置

- 该设置和实时报警的报警设置类似，可以根据优先级和报警状态进行颜色配置。

•通用配置

- 可以配置历史报警列表字体
- 可以配置历史报警控件背景颜色、表头颜色、表尾颜色
- 可以设置历史报警列表上方鼠标是否可操作
- 可以设置分页

历史报警控件接口

参数	描述	是否必要	数据类型	
starttime	起始时间	否	String	
endtime	结束时间	否	String	起始时间和结束时间参数不传入的情况下，默
da	数据代理	是	String	
tagname	tag名称	否	String	模糊查询：单个模糊查询：″tagname″多个模
priority	优先级	否	String	格式为″x_x″例如″0_255″、″1_1″
amtype	报警类型	否	String	
alarmArea	报警区域	否	String	
tagDesc	tag描述	否	String	模糊查询
alarmStatus	报警状态	否	String	”-1“：全部状态；″0″:″未恢复未确认″；″1″:未恢
confirmor	确认者	否	String	精确查询
className	所在类	否	String	精确查询
subsysName	子系统	否	String	精确查询
objectName	对象名	否	String	精确查询
almGroupName报警组	否	String	精确查询	
orderRules	排序方式	否	String	发生时间降序：″fd_ocurrtime;DESC″;发生时″fd_recovertime;ASC″;确认时间降序:″fd_con″fd_value_content;DESC″;报警值升序:″fd_va

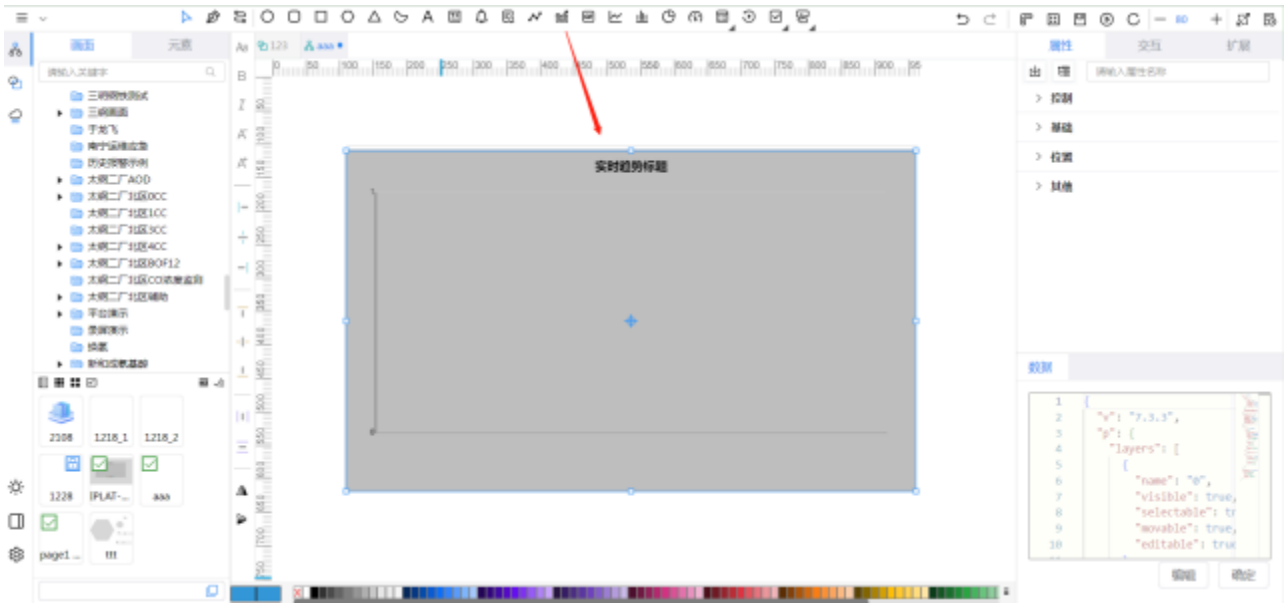
5.30 实时趋势控件

使用实时趋势控件

实时趋势最小尺寸

为了保证实时趋势图的显示效果，控件的宽高不能小于[宽*高240*180]。

1.打开2D设计器，新建一个画面，在工具栏上点击“实时趋势”控件，然后在画面上空白的地方拖拽，以便将“实时趋势”控件放置在画面上。



配置实时趋势控件

1. 实时趋势控件提供了一些配置项，可以在属性面板中进行配置。

2. 基本属性：

•标题：

◦实时趋势标题，默认值："实时趋势标题"

•基本属性：

◦时间跨度：实时趋势图展示的时间跨度，默认值5分钟，最大值30分钟

◦样本频率：实时趋势采集实时数据的频率，默认值1000ms，最小值500ms（小于500ms会自动设置为500ms

◦背景颜色：实时趋势背景颜色，可配置rgba格式，默认值：rgb(194,194,194)

•曲线配置：

◦线名：曲线名称

◦变量：通过选点配置曲线的变量表达式

◦颜色:曲线颜色，默认值：【rgb(44,159,44)rgb(228,63,63)rgb(0,0,0)rgb(12,135,135)

◦线宽：曲线线宽，默认值：1

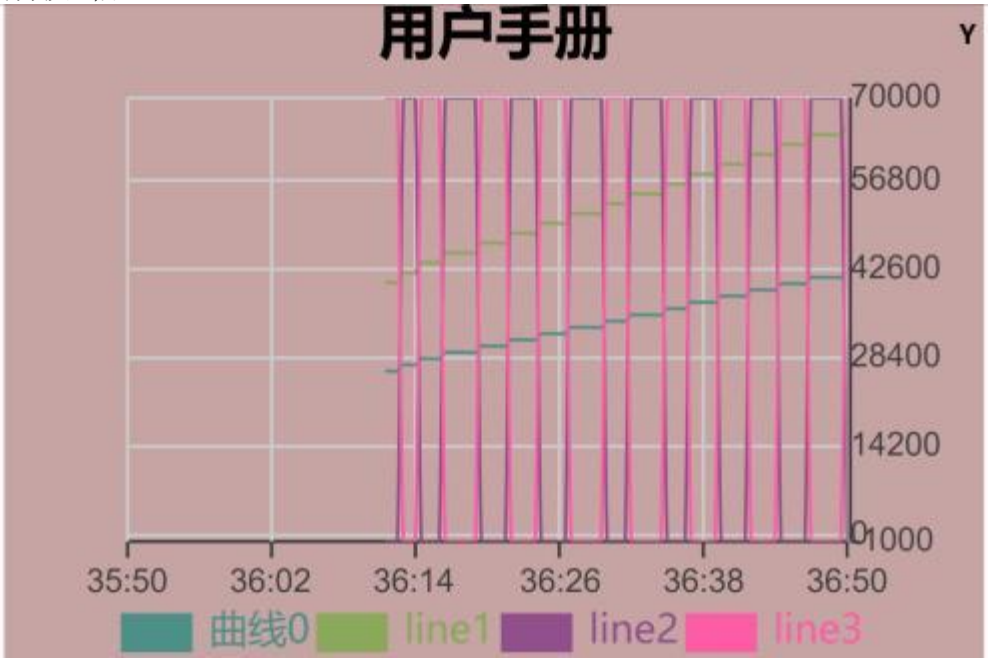
◦Y轴：曲线对应的Y轴，默认所有曲线都配置成系统的默认Y轴，用户可在Y轴配置中添加Y轴，选取需要的Y轴配置到曲线中，默认值：'Y'

•值刻度
◦是否显示Y轴标题：默认不显示
◦位置：Y轴显示在左侧还是右侧

3. Y轴配置：
    •标题
    •Y轴的量程，不配置默认自适应
    •刻度数：Y轴刻度分成n段，有n+1个刻度数，默认不配置，自适应

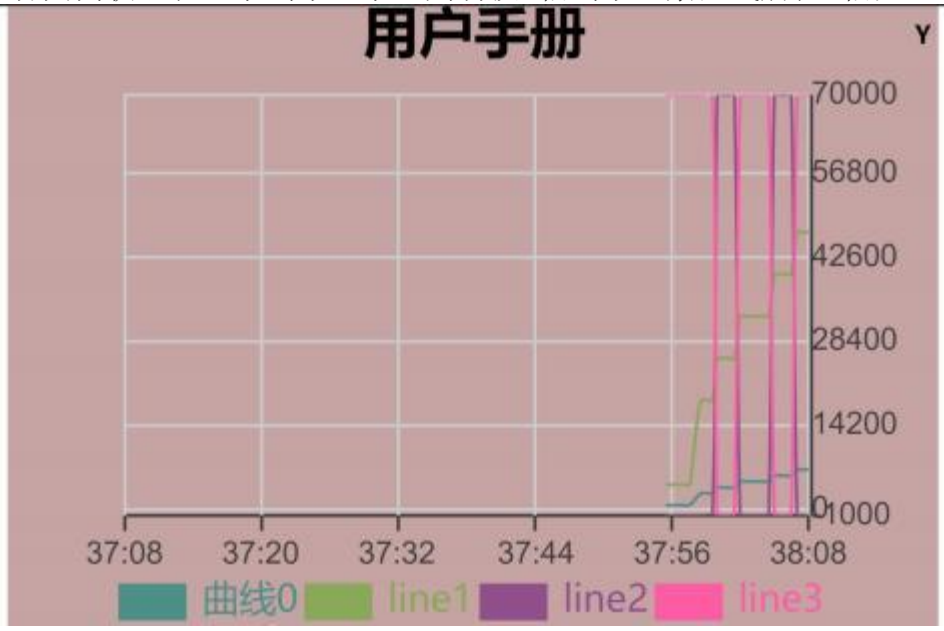
实时趋势运行态功能

1. 点击Y轴区域切换Y轴：

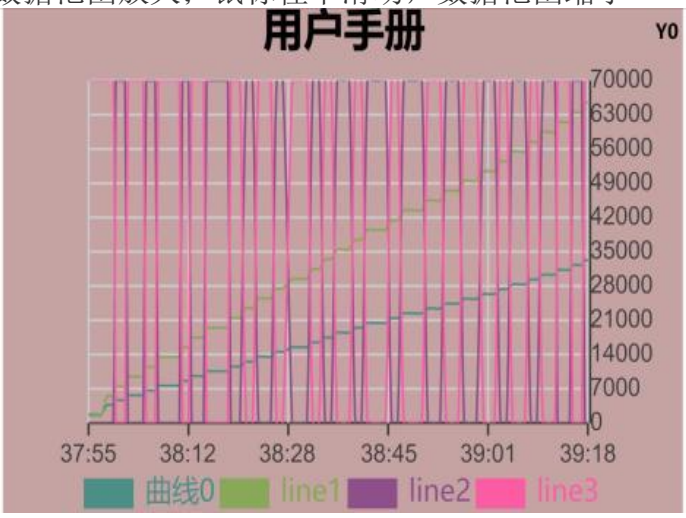


2. 点击图例切换Y轴

- 点击图例，若图例状态从不显示到显示，则切换Y轴到当前曲线所在Y轴；
- 点击图例，若图例状态从显示到不显示，则切换Y轴到下一条曲线所在Y轴；



3. 鼠标缩放功能：
   - 鼠标往上滑动，数据范围放大；鼠标往下滑动，数据范围缩小

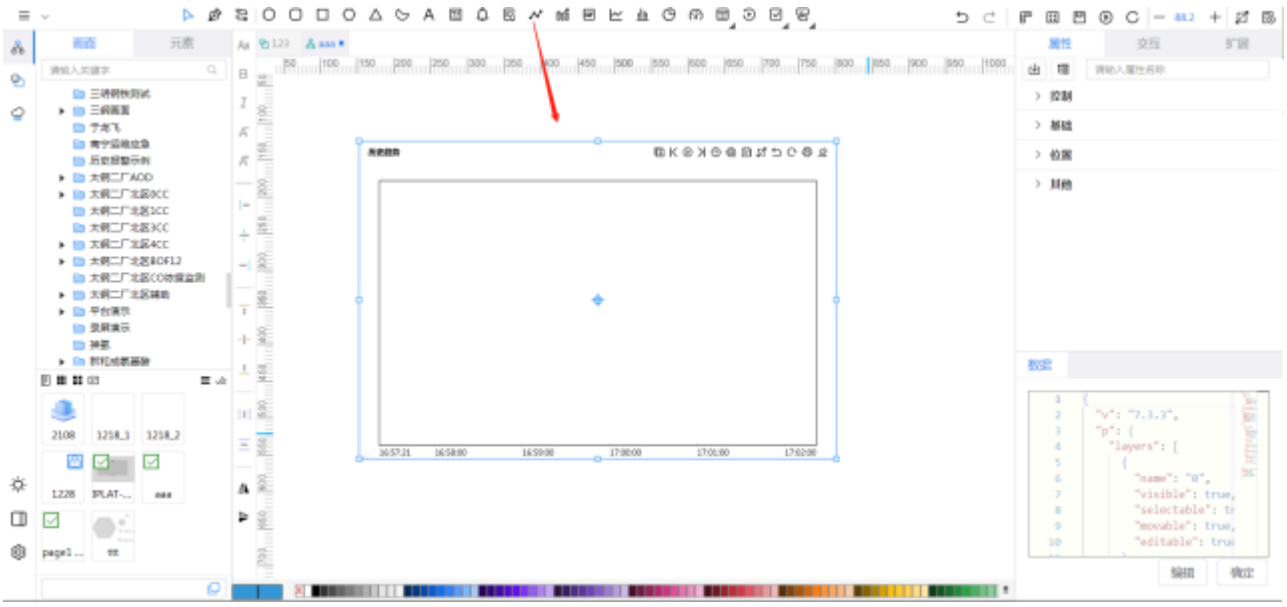


4. 使用注意事项：
   - 实时趋势时间跨度上限为30分钟
   - 实时趋势绘制内容为即时绘制，时间跨度越长绘制速度越慢，为了避免绘制速度对观感的影响，一般推荐时间跨度在5分钟以内

5.31 历史趋势控件

使用历史趋势控件

1. 打开2D设计器，新建一个画面，在工具栏上点击“历史趋势”控件，然后在画面上空白的地方拖拽，以便将“历史趋势”控件放置在画面上。

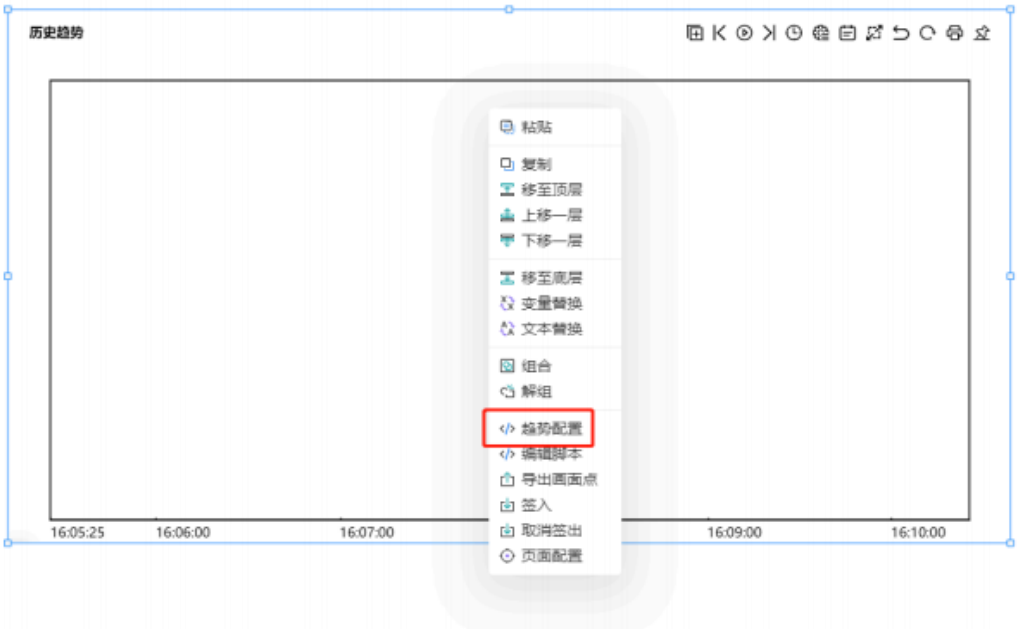


历史趋势最小尺寸

为了保证趋势图显示效果，历史趋势的尺寸不能小于[高*宽*280600].

配置历史趋势控件

1. 在“历史趋势”控件上右击，选择“趋势配置”，打开“趋势配置”控件配置页。



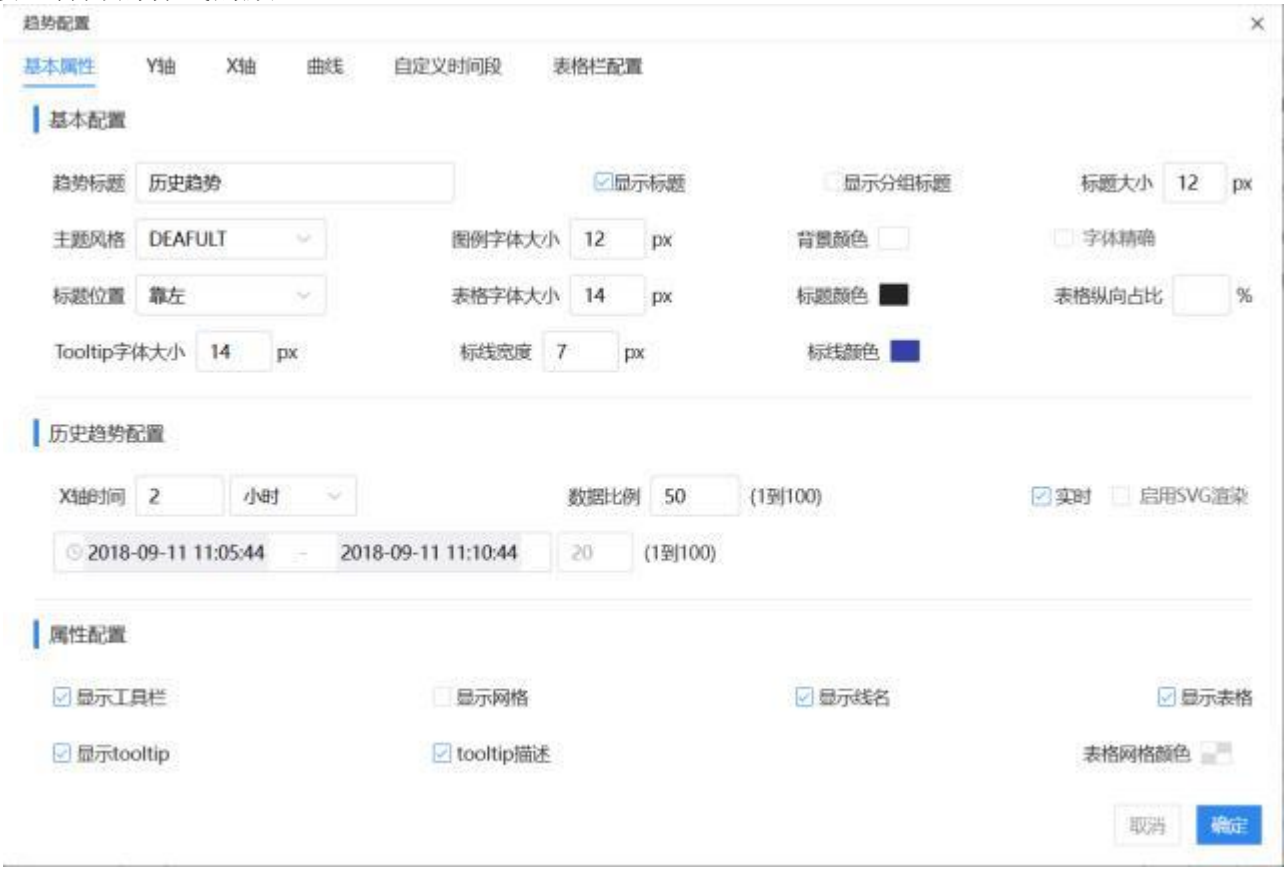
2. 基本属性：

- 可配置是否显示标题、趋势标题内容以及标题的样式（颜色，字体大小，位置）、表格字体大小、tooltip文本字体大小（复用表格字体大小）
- 可配置趋势图背景颜色和主题风格
- 可配置是否是实时趋势，选择实时趋势时，不能配置时间
- 不开启实时趋势时，可配置趋势发生时间区间
- X轴时间范围（仅对实时状态时有效）x轴的时间单位包含：分、小时和分钟。上图中选择了x轴的范围为2小时，实际运行时，x轴的时间范围显示的为2小时
- 可配置运行态是否显示工具栏
- 可配置运行态是否显示线名
- 可配置趋势图是否显示网格线以及辅助网格线
- 显示表格：可以选择是否显示趋势表格

•显示tooltip：可以配置是否显示tooltip
•tooltip描述：可以选择tooltip中是否需要显示曲线描述信息（需要先确保“显示tooltip”已被勾选）
•标线宽度：可以配置

按钮打开的-标线的宽度（可以配置的范围为1~7）
•标线颜色：可以配置

按钮打开的标线的颜色



3.Y轴配置：
a.【全局属性】可以对趋势图的Y轴做全局配置
•可配置是否显示Y轴标题【默认显示】
•可配置Y轴的位置：全部分布在右侧、全部分布在左侧、均匀分布在两侧【默认分布在两侧】
•配置多Y轴的情况下是否只显示一条Y轴：勾选后，运行的趋势控件只显示一条y轴，默认为第一条曲线配置的y轴，可通过点击y轴切换y轴，切换顺序按照配置的所有曲线配置的Y轴轮询【默认不开启】
•Y轴间隔：每两条Y轴之间的间隔（若Y轴刻度长度超过这个间隔，则超出部分用...表示，鼠标悬浮在刻度上可以展示刻度的实际值）【默认60px】

b. 【新增Y轴】

4. X轴配置：
    - x轴配置包括：
        - 标题文字配置
        - 刻度字体大小
        - X轴颜色配置
        - 线宽（可以在曲线样式预览中看到）

。X轴时间刻度格式

5. 曲线配置：

•切换到【曲线配置】tab页，可以添加趋势组，对曲线进行增删改操作，显示表格：可以选择是否显示趋势表格

•默认有一个分组'default'，也可以添加一个趋势组，点击【新增组】按钮，在【添加组】对话框中进行操作

•为group1添加曲线，必须在一个已有的组下添加曲线，点击【新增曲线】按钮，弹出【添加曲线】对话框

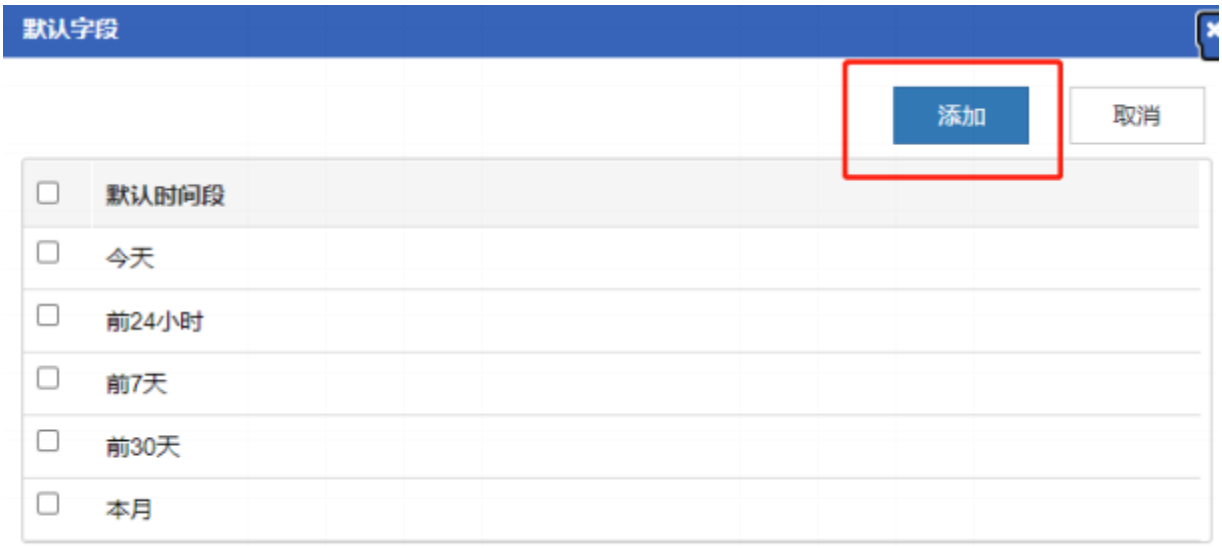
- 曲线：曲线名称
- 变量：点击变量文本框后的选点按钮，可以选择需要显示在趋势图中的曲线
- 线型颜色：选择蓝色
- 线型风格：选择0 Solid
- 使用原始值生成曲线：勾选
- 线型粗细：设置为3，通过点击上下箭头可以修改线宽
- **线形颜色、线形风格、线形粗细可以在线样式中预览**
- 显示方式：选择默认（拟合）
    - 对应y轴：选择添加的y轴
    - 值标记：选择1-正三角形()
    - 保留小数：根据实际需求自定义小数位数
    - 显示曲线值范围（限历史值）：勾选
    - 统计值显示方式：曲线方式
    - 统计值勾选了：最大值，最小值（最大值和最小值会以两条直线方式显示在趋势图上，如果选择了文本提示方式，当鼠标移动到曲线时，会以文本方式显示统计值）

- 为曲线设置量程

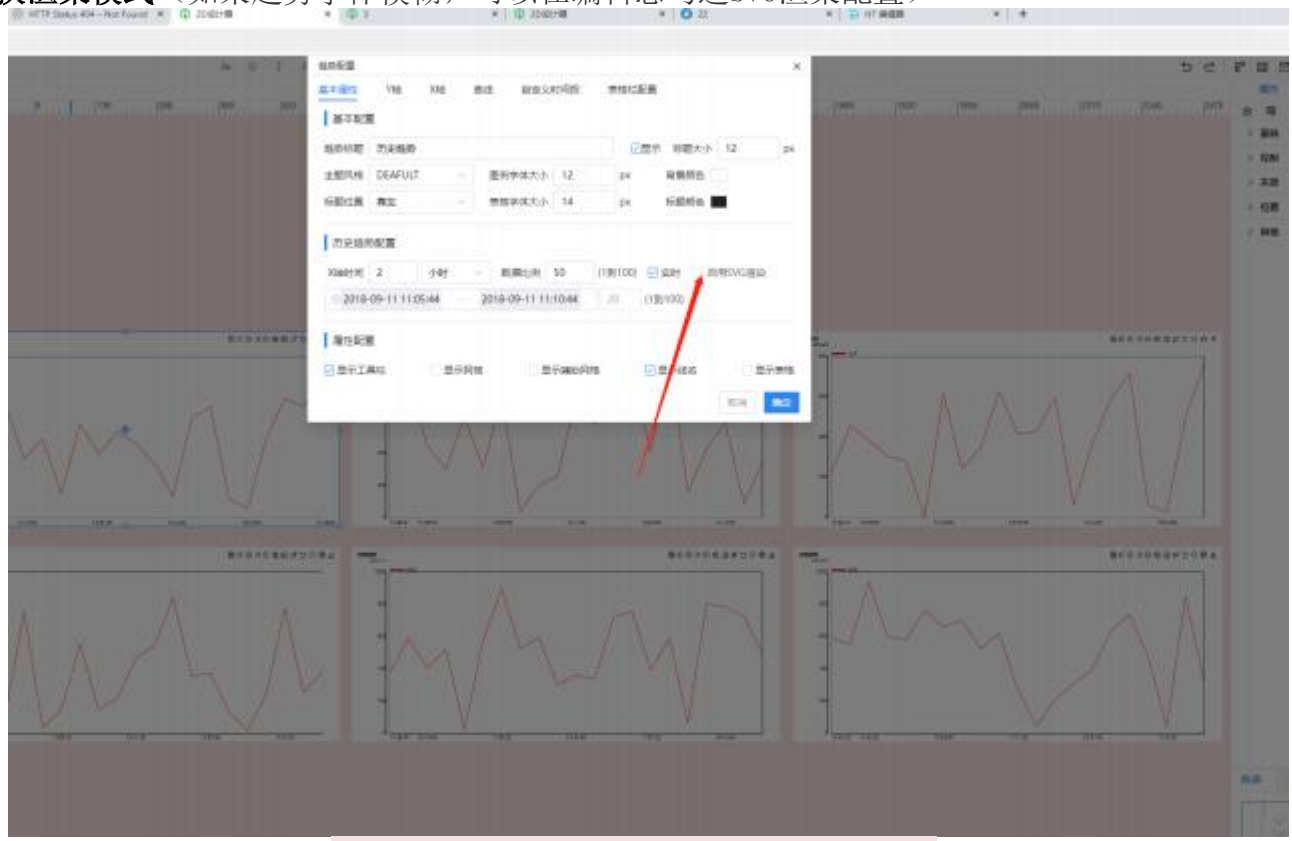
6. 自定义时间段配置：
   - 用于配置用户常用时间段，在趋势运行时，用户可以迅速切换到常用时间段的趋势图，进行查看

   - 点击【默认字段】按钮，可添加默认时间段，包含：今天、前24小时、前7天、前30天、本月；点击【+添加】按钮，是指趋势运行态的当前时间减去某一个时间，几天，几小时，几分钟或者几秒；

   - 点击添加按钮，可以将该条自定义时间段添加到列表中

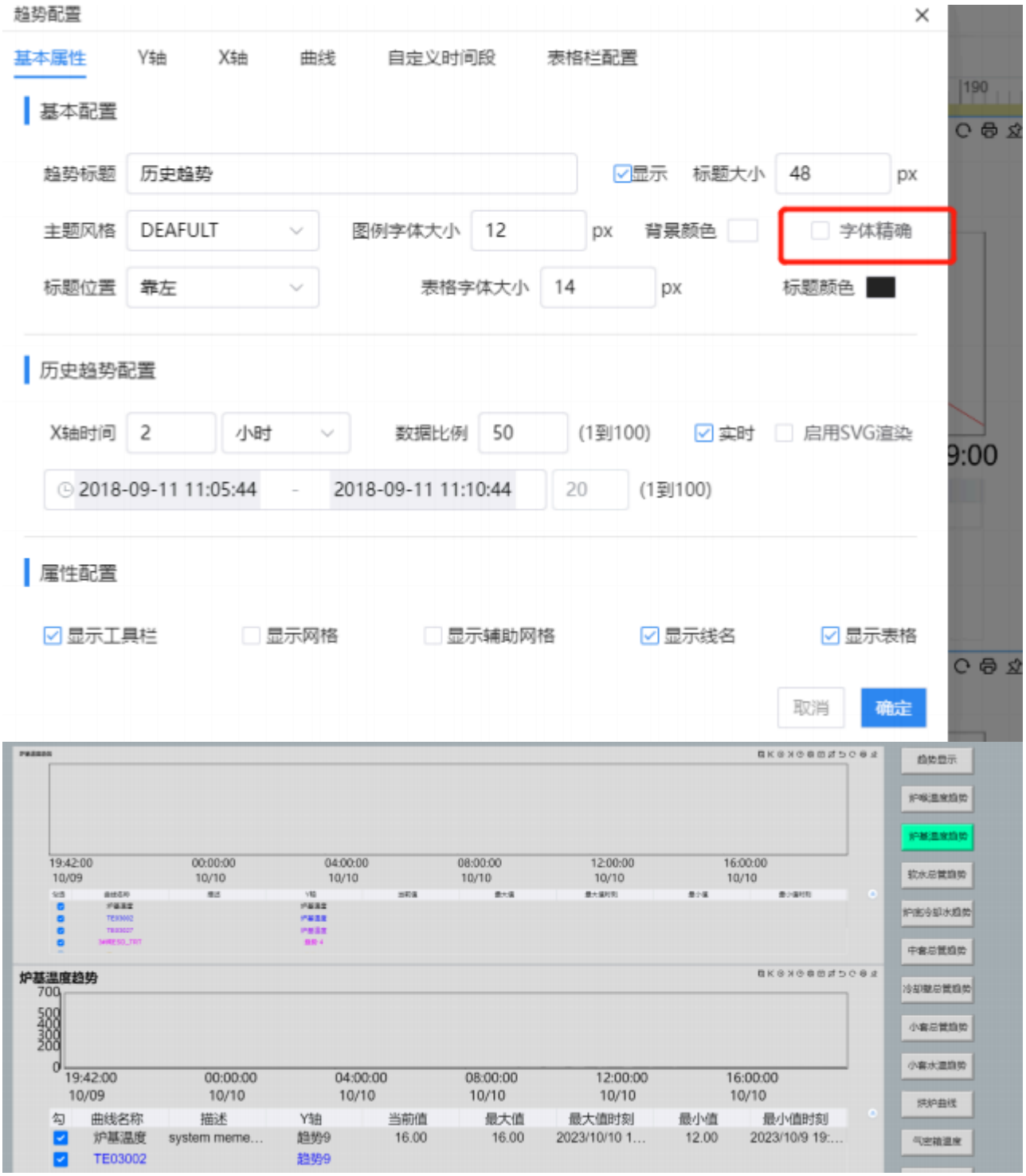


切换渲染模式（如果趋势字体模糊，可以在编辑态勾选SVG渲染配置）



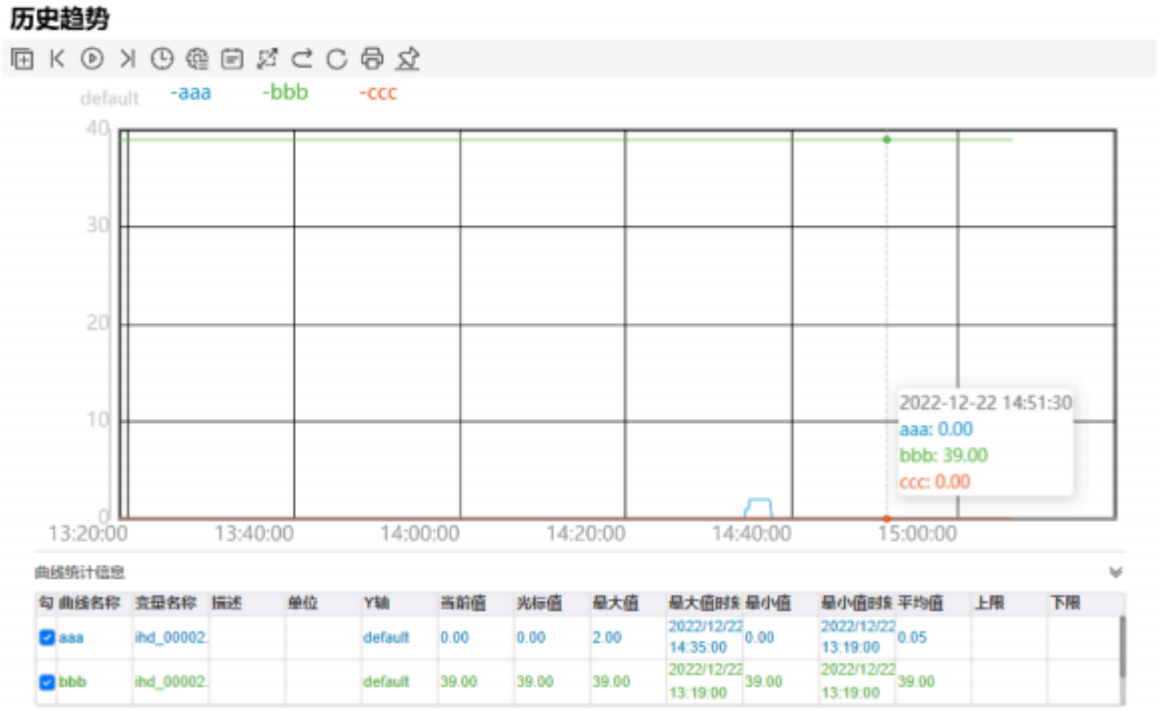
字体精确

勾选后趋势相关字体配置（表格字体大小、刻度字体大小、x轴字体大小、y轴刻度字体大小）会直接应用在趋势上，不会再随趋势控件大小自适应



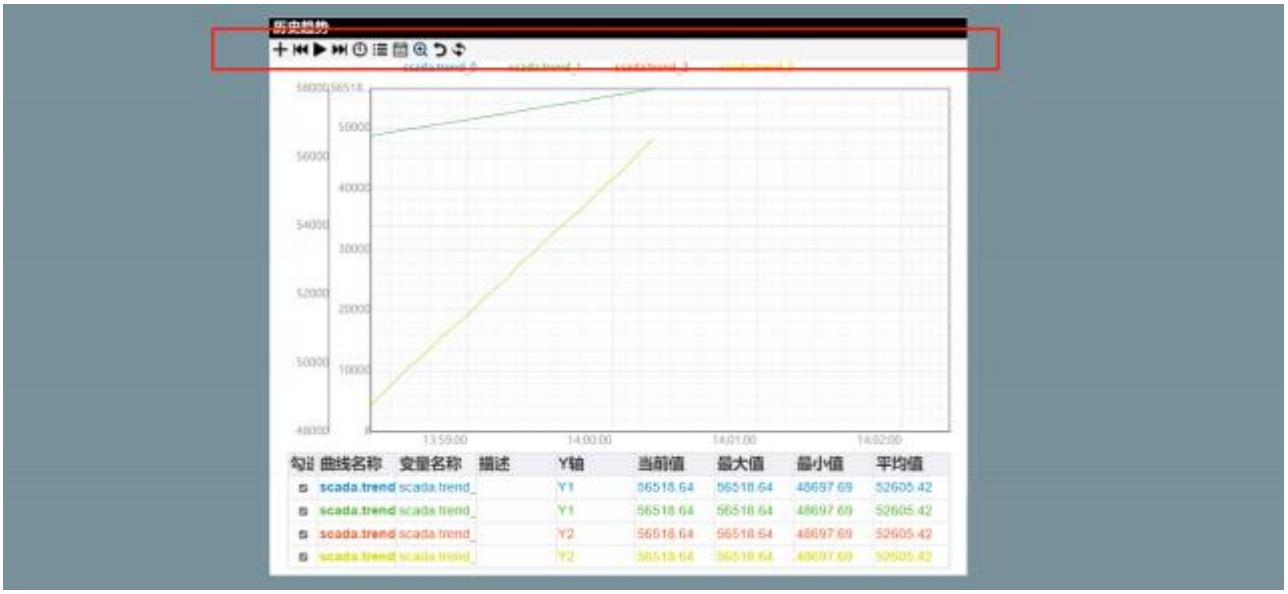
历史趋势运行态效果图

•历史趋势控件在运行态的效果图。



趋势工具栏

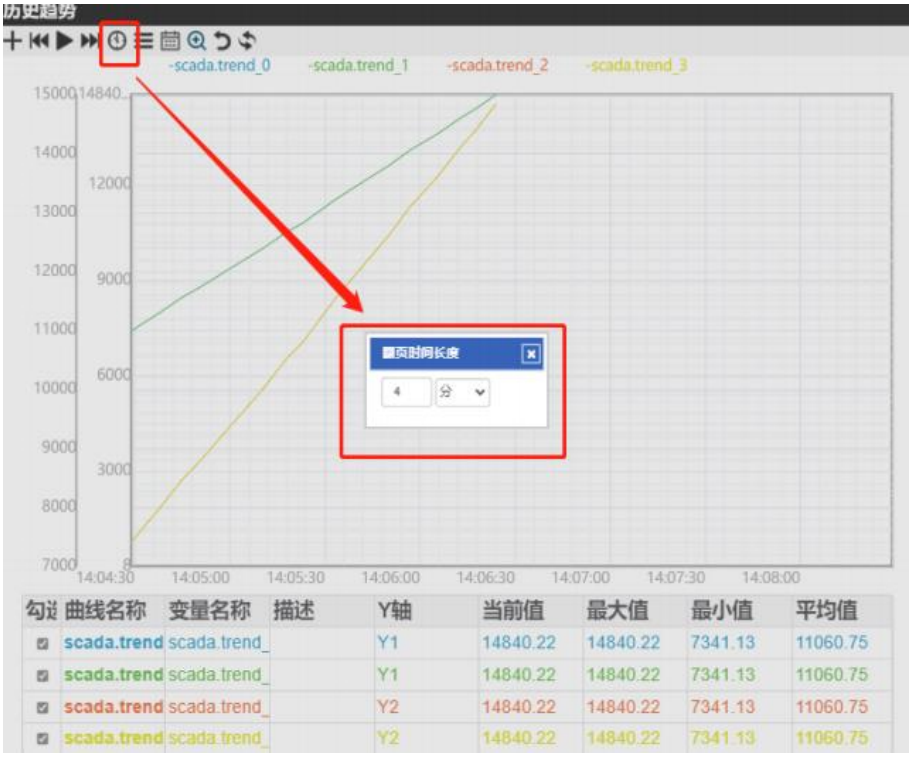
1. 趋势运行态的功能按钮集成在工具栏中，如下图所示，需要在编辑态勾选’显示工具栏’



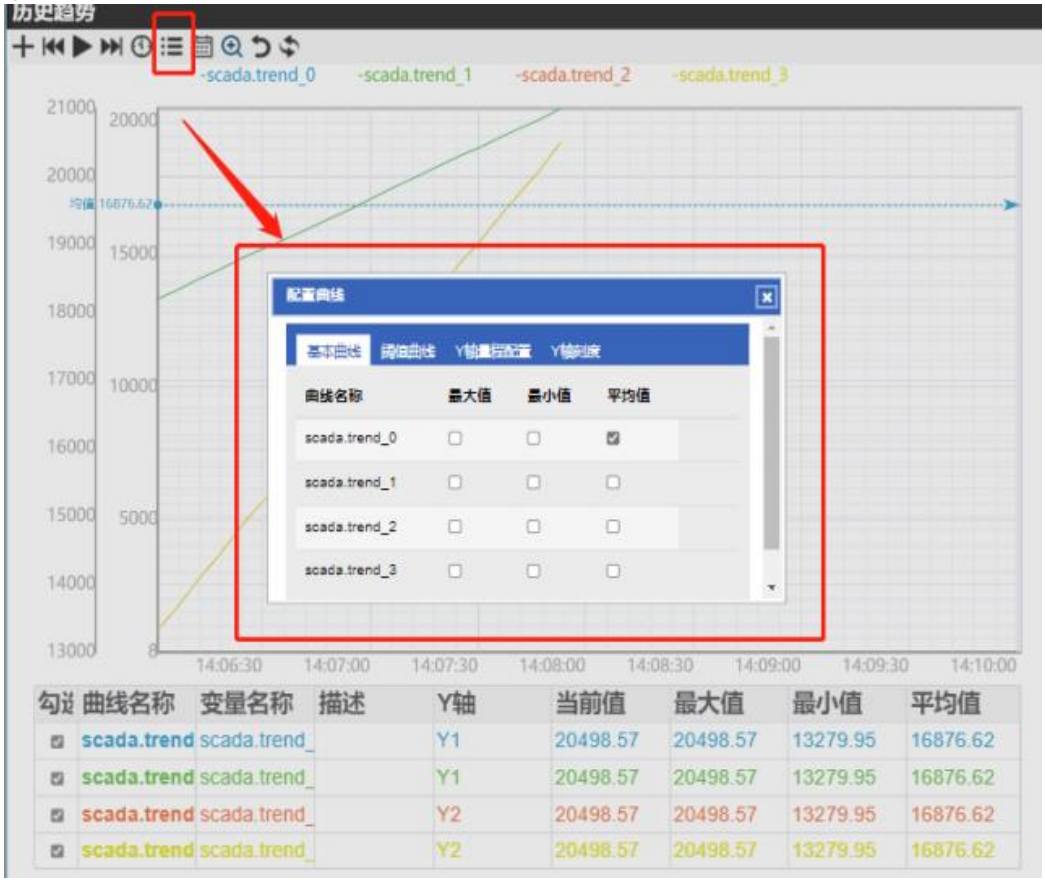
2. 此处点击新增后能够根据绑定新的点来绘制趋势线（默认是不保存在本地），如果需要保存在本地可以勾选保存在本地

历史趋势
选择分组
分组名称
分组描述
新增
修改
删除
应用
default
曲线
点
配置曲线
曲线
变量
选点
线型颜色
线型风格
0 Solid
线型粗细
1
显示方式
默认(拟合)
线样式预览
对应Y轴
default
值标记
0-无
使用原始值生成曲线
保留小数
2
统计值显示方式
曲线方式
最大值
最小值
平均值
量程
保存到本地
确定
取消

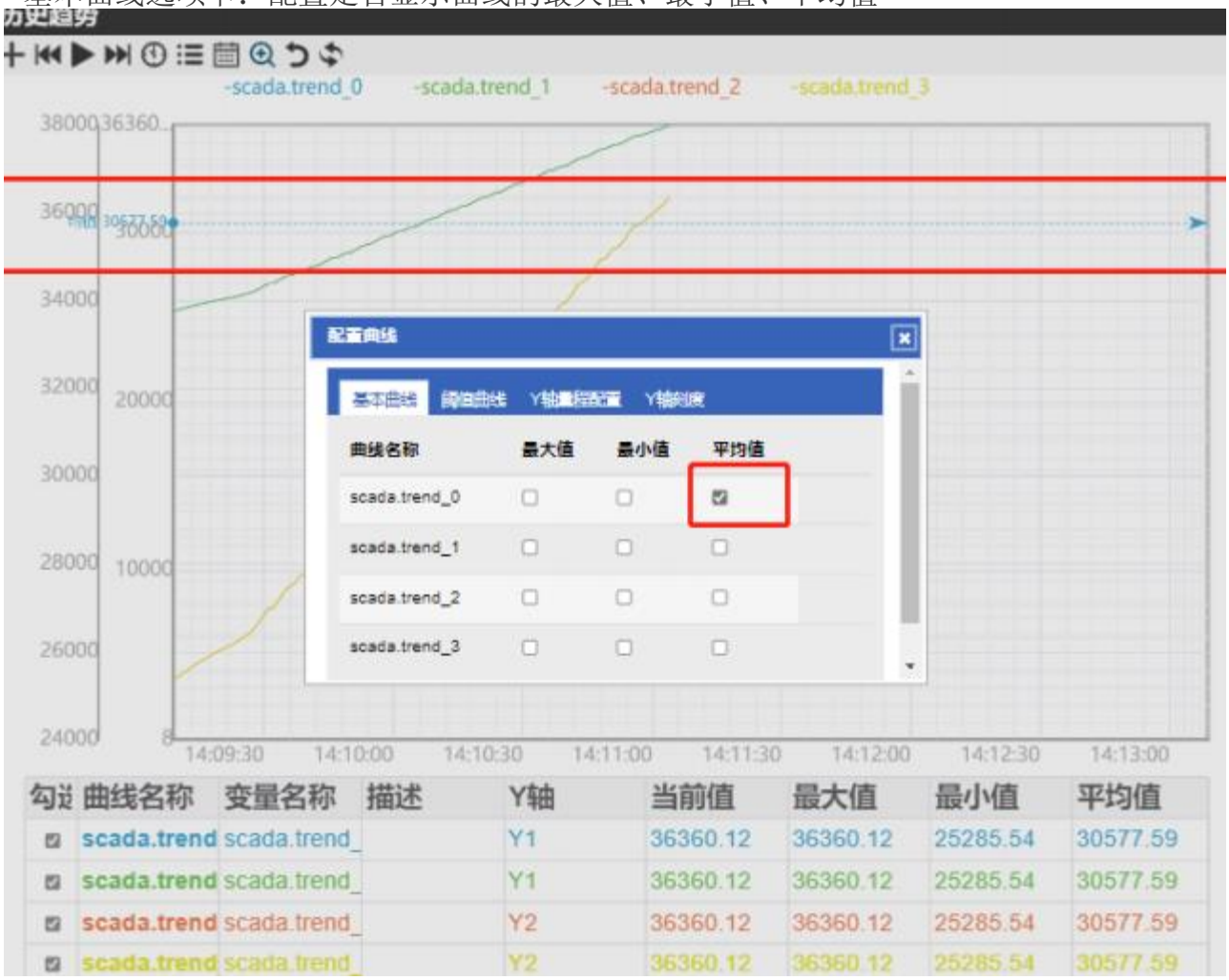
3. 翻页时间长度


历史趋势
-scada.trend_0
-scada.trend_1
-scada.trend_2
-scada.trend_3
翻页时间长度
4
分
勾选
曲线名称
变量名称
描述
Y轴
当前值
最大值
最小值
平均值
scada.trend
scada.trend_
Y1
Y2
14840.22
7341.13
11060.75

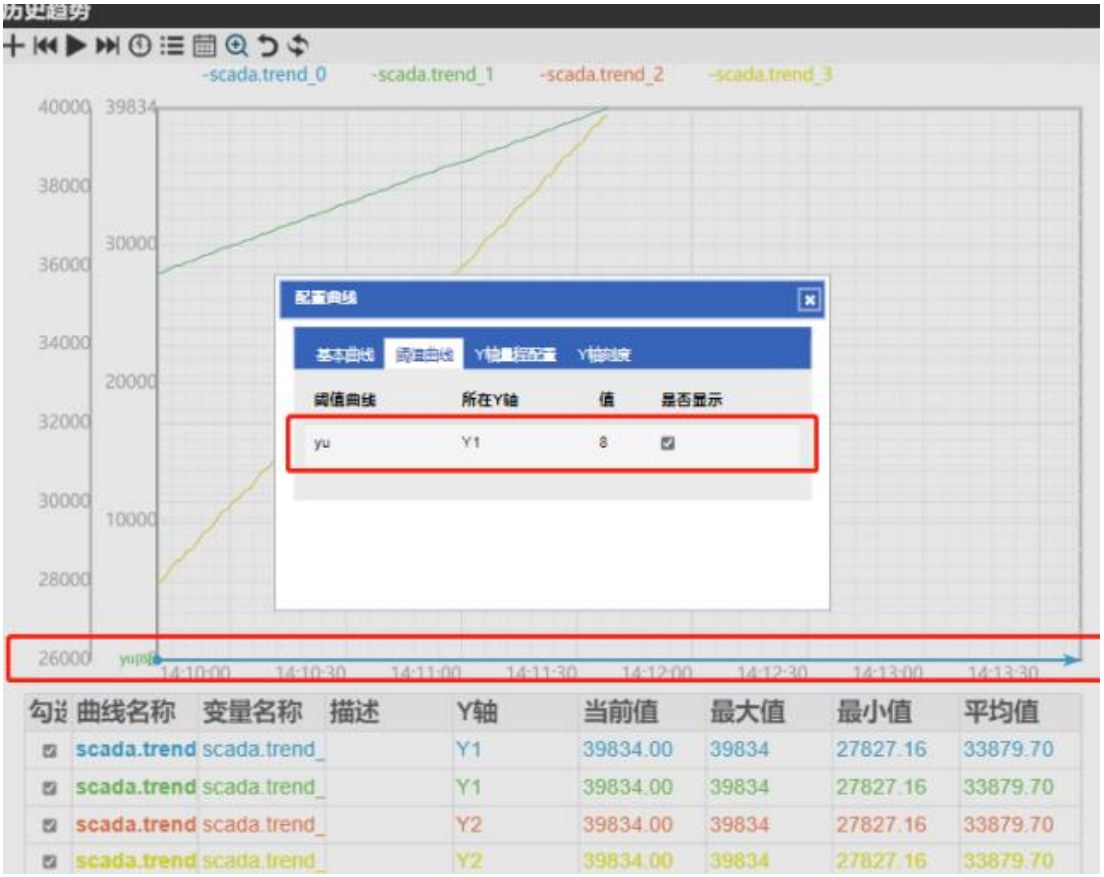
4. 阀门配置



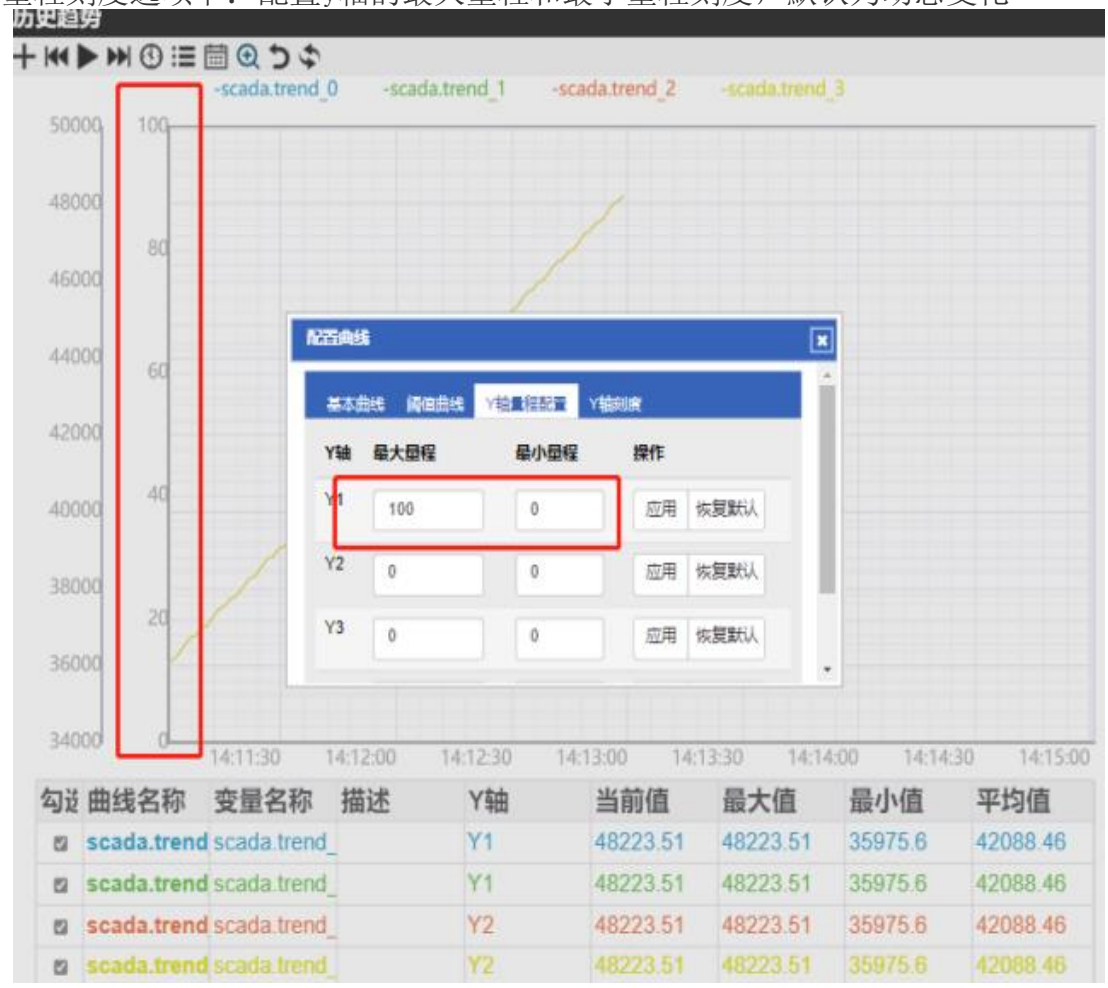
5. 基本曲线选项卡：配置是否显示曲线的最大值、最小值、平均值



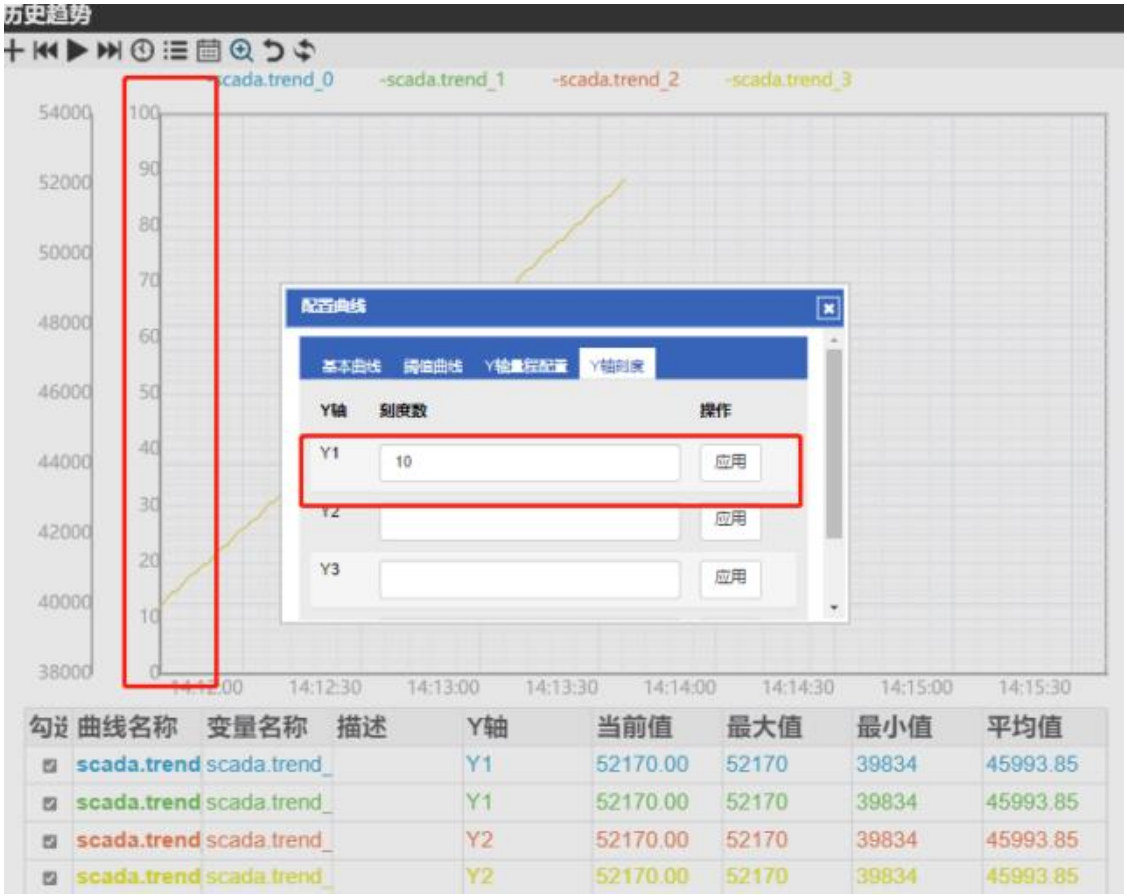
6.阈值曲线选项卡：配置运行态阈值曲线（IHD数据库中对应部分内容勾选后，阈值曲线则可以配置）



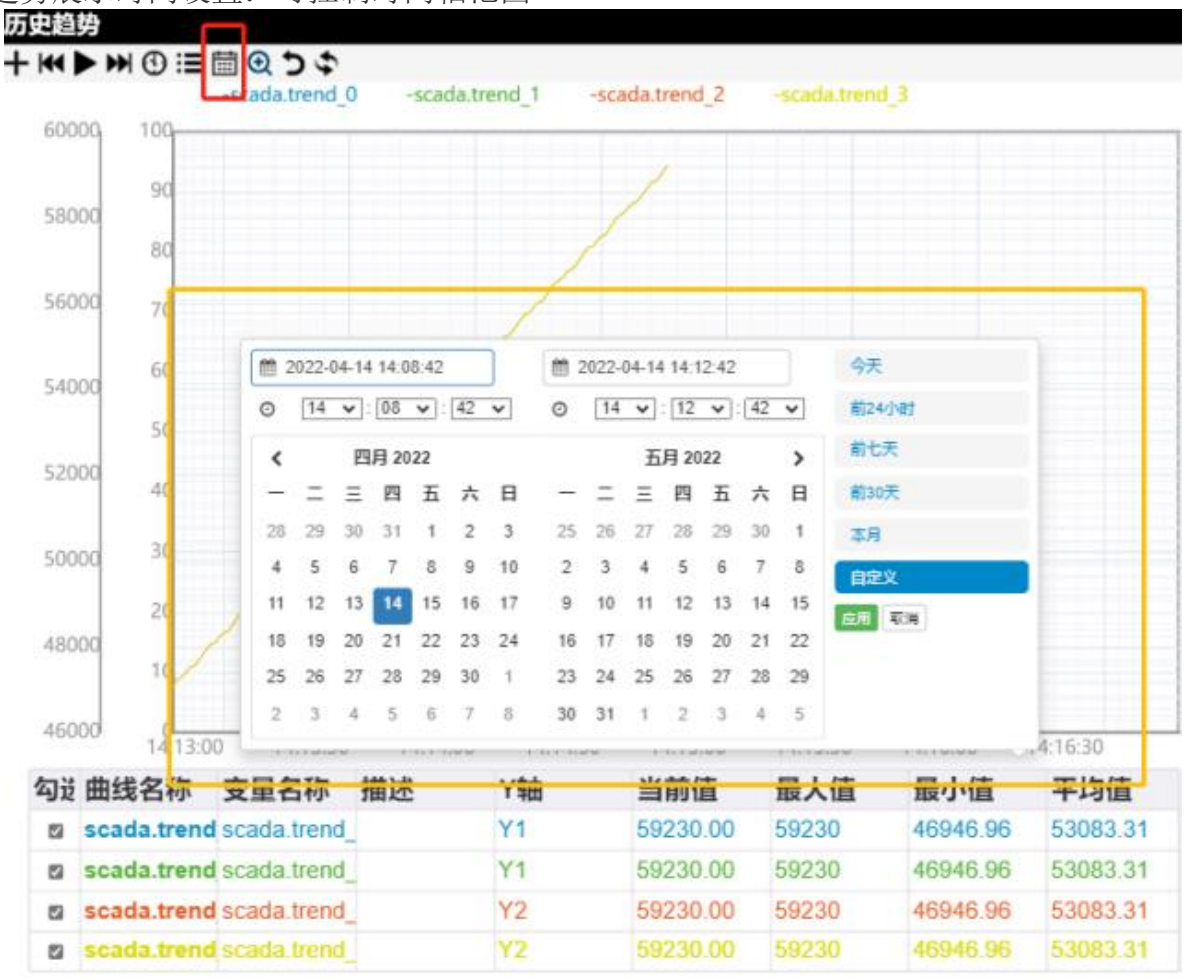
7. Y轴量程刻度选项卡：配置y轴的最大量程和最小量程刻度，默认为动态变化



8. Y轴刻度选项卡：配置y轴曲线的刻度数，只在当前画面的运行态生效



9. 趋势展示时间设置：可控制时间轴范围



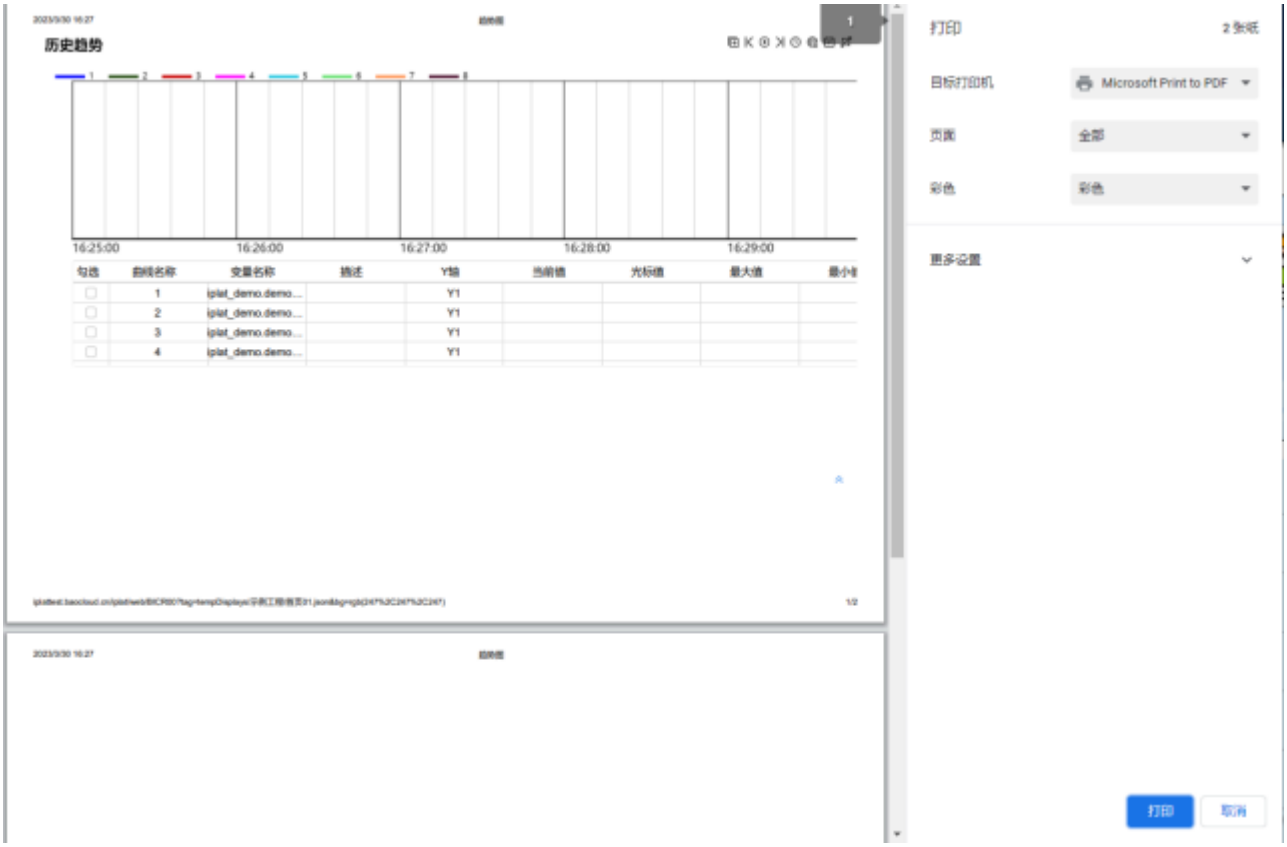
10. 区域缩放&区域缩放还原：默认开启区域缩放，按钮有蓝色的边框代表已开启；可选中一块区域使用鼠标拖拽指定区域缩放

勾选	曲线名称	变量名称	描述	Y轴	当前值	最大值	最小值	平均值
☑	scada.trend	scada.trend_		Y1	2473.59	65534	255	49080.50
☑	scada.trend	scada.trend_		Y1	2473.59	65534	255	49080.50
☑	scada.trend	scada.trend_		Y2	2473.59	65534	255	49080.50
☑	scada.trend	scada.trend_		Y2	2473.59	65534	255	49080.50

11. 缩放区域重置：无论是否缩放过区域，点击该按钮后都将把趋势图的缩放置为初始状态

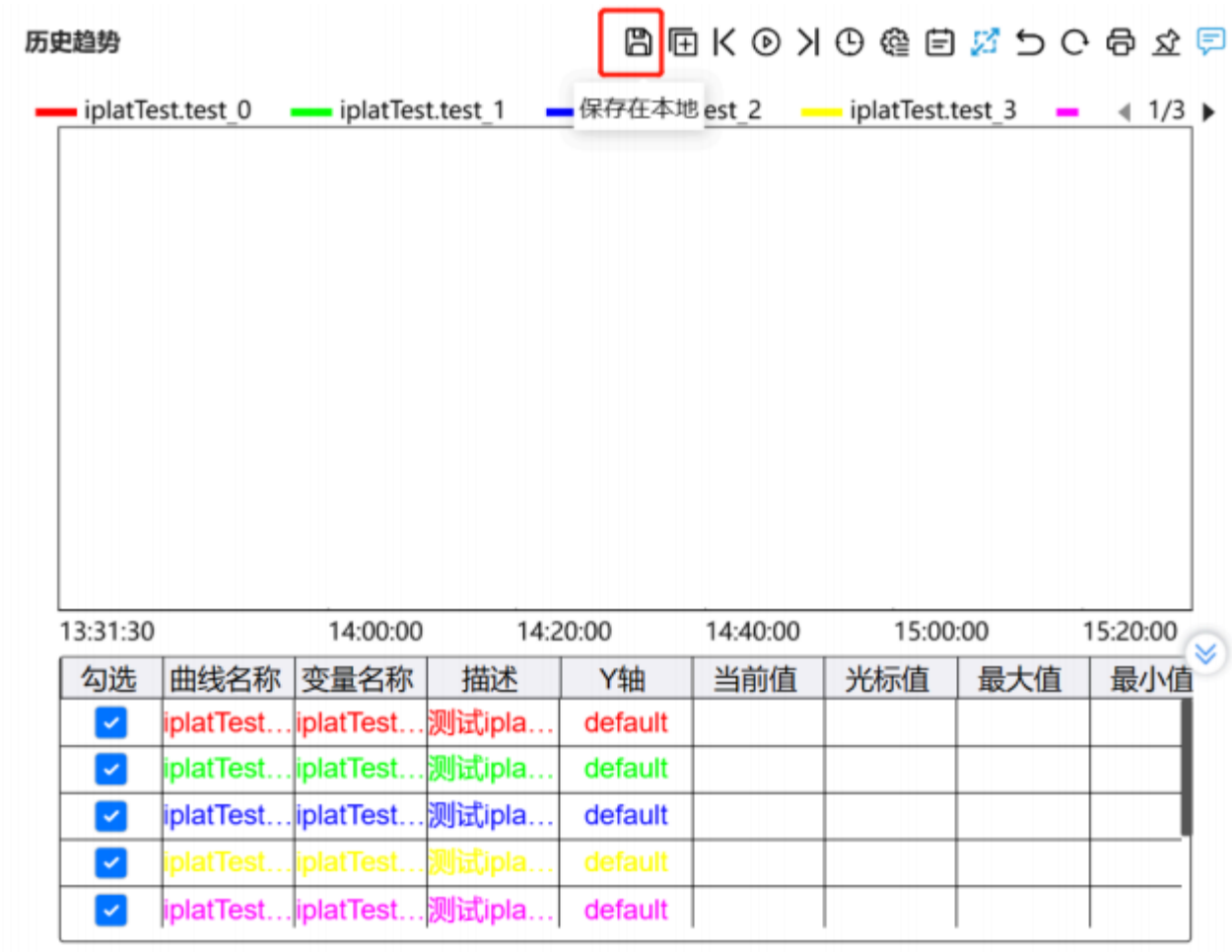
12. 趋势图打印功能：可选择打印当前趋势页

勾选	曲线名称	变量名称	描述	Y轴	当前值	光标值	最大值	最小值
☑	1	iplat_demo.demo...		Y1				
☑	2	iplat_demo.demo...		Y1				
☑	3	iplat_demo.demo...		Y1				
☑	4	iplat_demo.demo...		Y1				



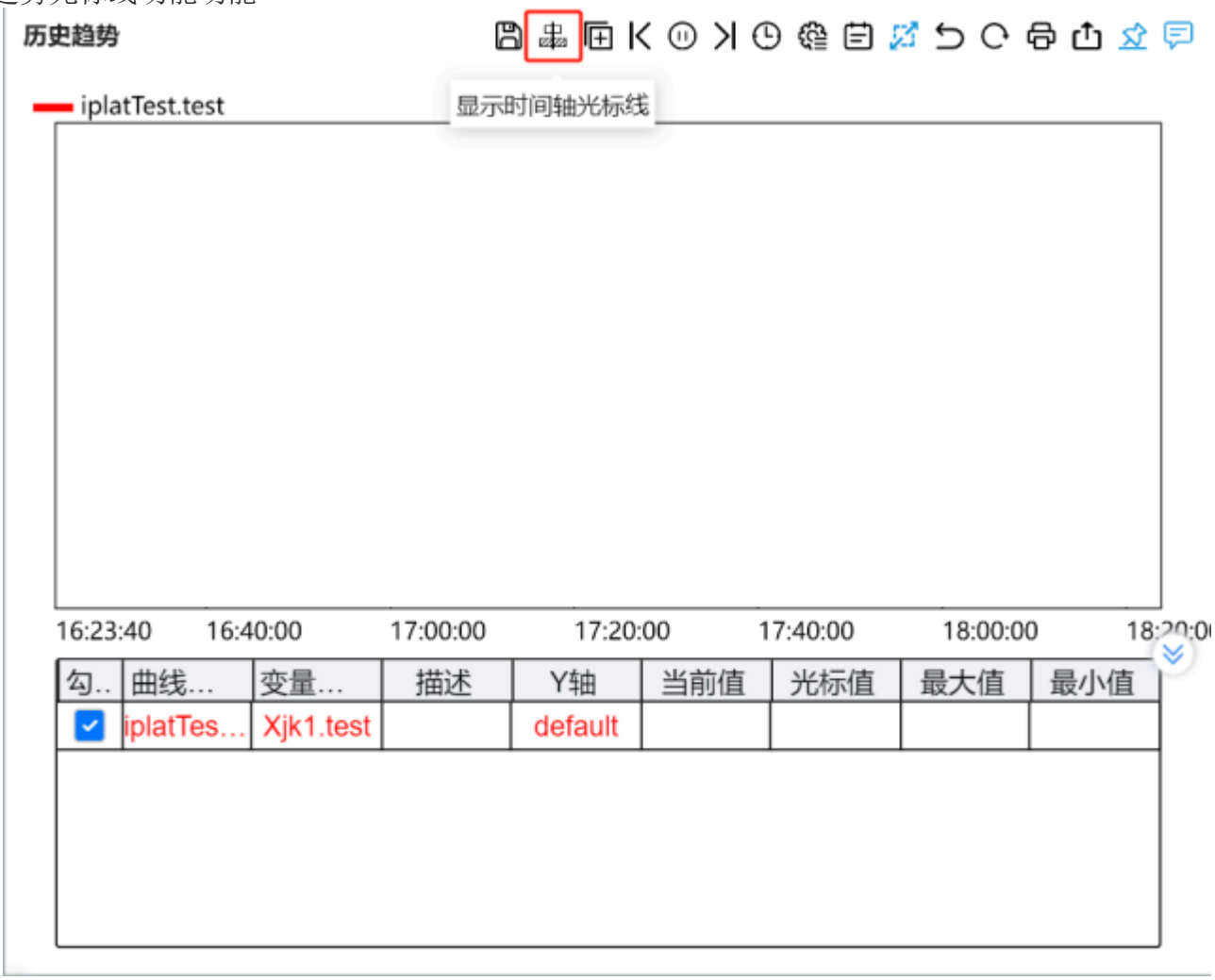
历史趋势尺寸

13. 趋势保存功能

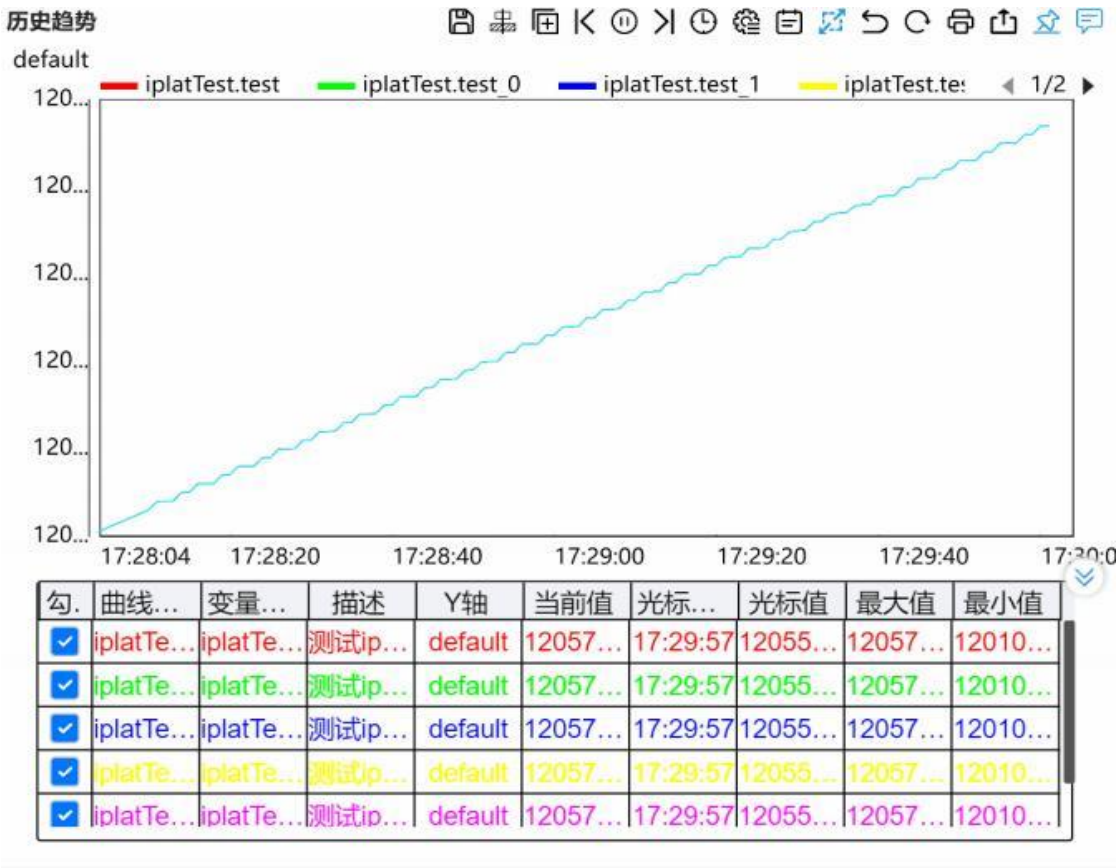


历史趋势：页面刷新后：是否显示tooltip、表格的收缩释放、趋势中的曲线（无论新增还是删除后的）、时间跨度，会被保存。
文本趋势：是否显示tooltip、表格的收缩释放、趋势中的曲线小数位数、时间跨度，会被保存
对于文本趋势：
1).曲线配置是以曲线的tag点名为依据进行区分保存的。配置读取时，也是按照曲线的tag点名称，在已经保存的配置中进行查找读取对应的配置项。
2).点击保存按钮后，会保存当前文本趋势中的所有曲线配置
3).最多保存的曲线条数
-约定数据库中，总共最多存储40条不同的曲线配置
-若保存新的一批曲线配置后，曲线条数超出了40这一上限，则会从最旧的配置开始删除（譬如已经保存了35条曲线配置，又保存了新的8条不同曲线的配置，则会删除保存的配置中，最旧的3条配置，以腾出空间）
-若同一条曲线，在已经保存了配置的情况下，再次保存了配置。则数据库中，该曲线的旧配置会被删除，并存入该曲线的新配置。（新配置的删除优先级会被延后）
4).保存的配置生效的情况
-在画面中，用脚本/点击文本元素的方式添加曲线，如果有保存过的配置，则会被读取以生效
-如果是直接在文本趋势窗口中新增的曲线，不会读取保存的配置。此时，需要用户在新增界面中，自行设定新增曲线的小数位数。

14.趋势光标线功能功能



实际作用如下：



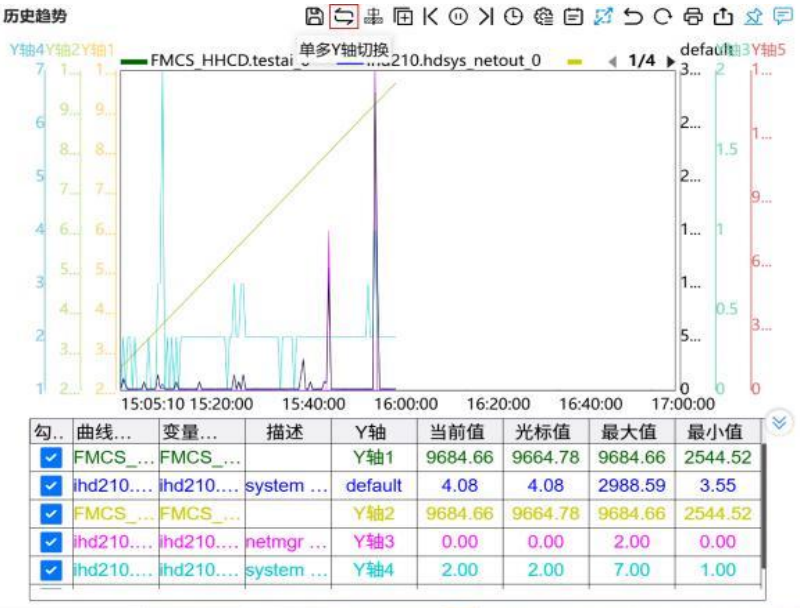
注意：

1.拖动光标轴的过程中，趋势会自动处于暂停状态。

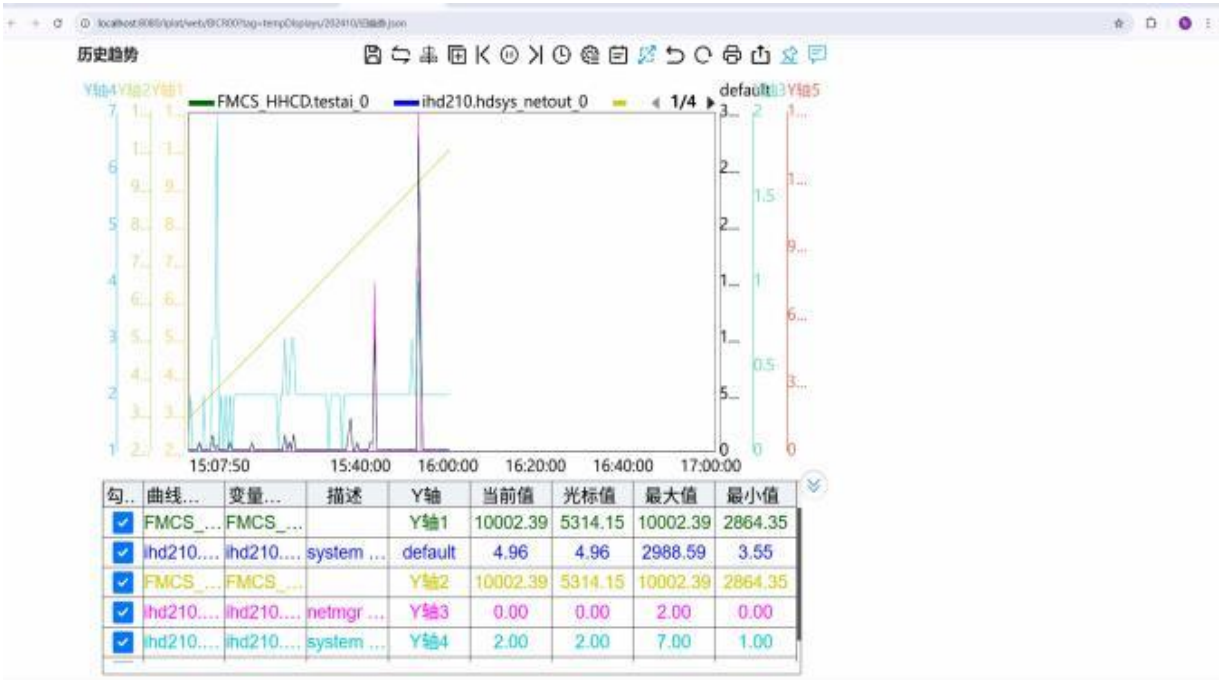
2.当处于显示光标轴的时候，tooltip文本框不会出现，且自动处于关闭状态。时间及数据会在表格中的光标时间、光标值中显示。

3.为了避免渲染问题，拖拽的过程中，只有鼠标停止100ms时，才会将次数的数据填写在表格中。

15.趋势单/多Y轴切换功能



实际作用如下：



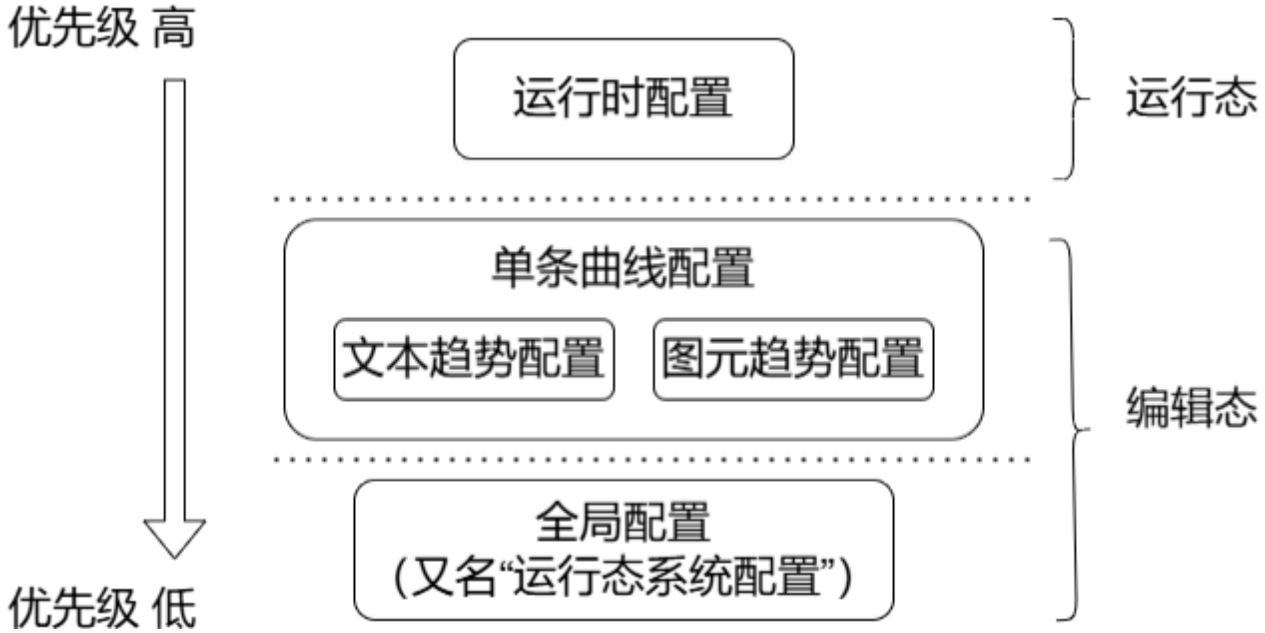
注意：由多Y轴切换至单Y轴后，可以通过点击单Y轴来切换需要显示的Y轴。

5.32 文本趋势

文本趋势，是使用文本控件通过配置实现右键文本，打开该文本绑点的历史趋势

趋势配置

有多个地方可以对趋势进行配置。它们之间的优先级顺序如下图所示（其中，运行态系统配置即为编辑态下的全局配置）



1.编辑态有全局配置和单条曲线配置之分；单条曲线配置又可能来自于文本的趋势配置或图元的趋势配置

2.编辑态下，单条曲线配置（即文本/图元趋势配置）的优先级高于全局配置ba

3.进入运行态时，趋势的初始配置状态，由编辑态的所有配置项经过优先级顺序处理后决定

4.运行态下的趋势配置具有最高优先级例：

•编辑态中的全局配置（即运行态系统配置），配置全局小数位数为2；编辑态中的文本趋势配置，配置小数位数为1并启用。且处于运行态时不进行额外配置，则运行态中，实际Y轴

•在运行态，配置小数位数为1。则无论进入运行态以前，在编辑态的全局配置/文本趋势配置/图元趋势配置进行了何种的配置，在运行态的小数位数始终为1

文本趋势配置

在文本的“趋势配置”属性中，必须配置：
1.勾选显示趋势
2.曲线颜
色注意
该文本的点，必须在设备点配置页面勾选了“保存历史数据”，才有历史趋势、文本趋势显示

在文本的“趋势配置”属性中，可配置“启用小数”和“小数位数”，表示Y轴上的数字是否格式化为指定位数的小数形式

5.33 图元趋势配置

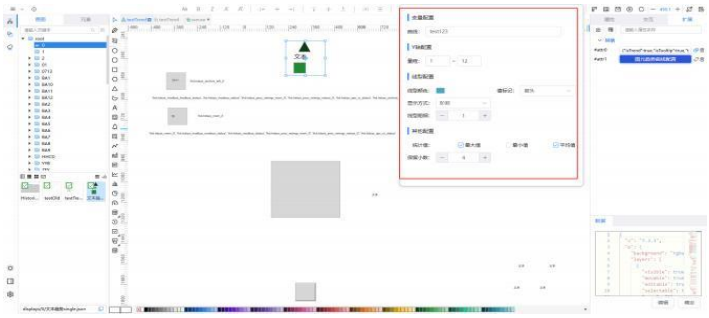
给图元添加趋势曲线配置的详细方法见：图元Tooltip配置及趋势展示以下主要就配置界面与效果进行介绍

图元趋势建立方法

2D设计器中，新建图元，在图元右侧的属性栏中，找到数据绑定，进行以下配置
•图元中通过添加customTooltip属性，实现鼠标移入时出现提示框；以及右键点击时出现属性框。在属性框中就可以打开图元趋势。同时，对customTooltip属性进行配置，即可进行tag点绑定
•图元中通过添加图元趋势曲线配置属性，实现图元趋势的配置

使用customTooltip属性时，务必确保下图属性被勾选，方可成功打开图元趋势

图元趋势配置界面



界面中提供的配置项有：

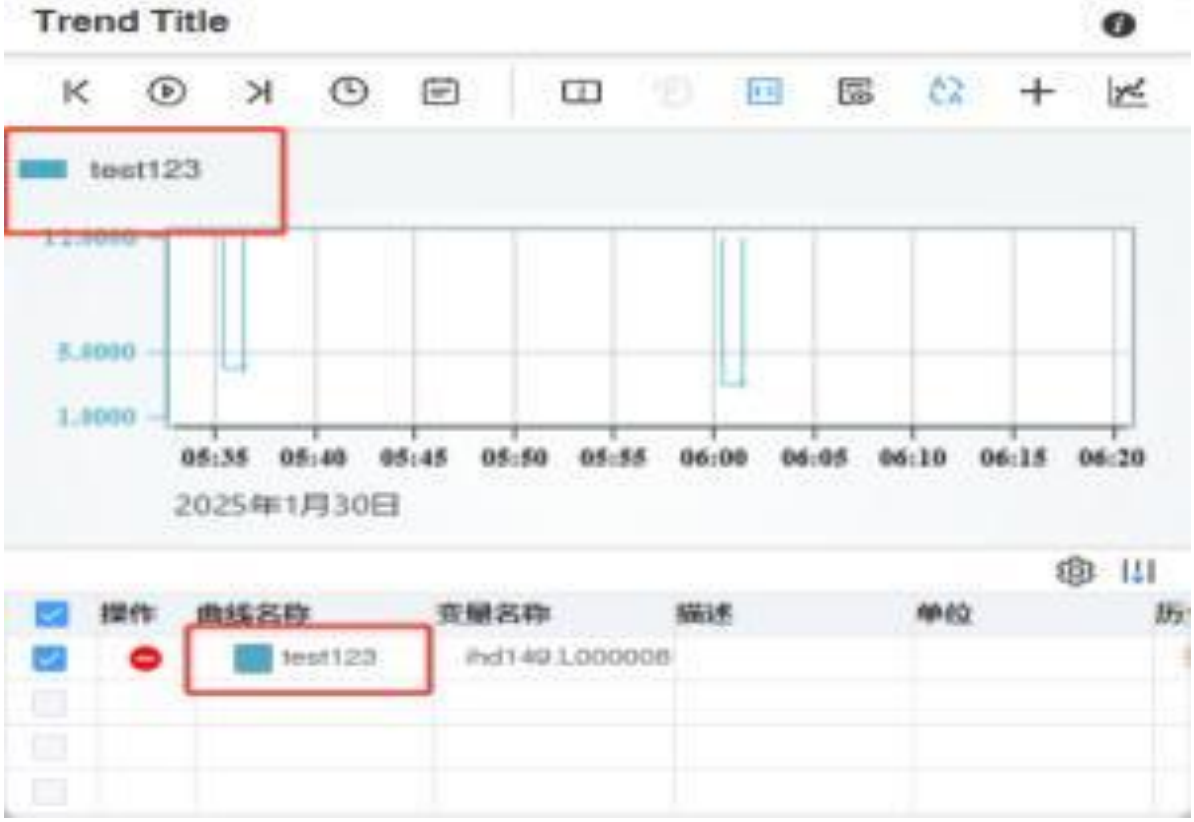
- 曲线（曲线名称）
- 量程（Y轴自定义量程）
- 线型颜色（曲线颜色）
- 值标记（曲线数据点形状）
- 显示方式（阶跃/连线）
- 线型粗细（曲线宽度）
- 统计值（开启哪些阈值线）
- 保留小数（Y轴刻度精度）

配置效果

以下图为例进行配置的效果展示

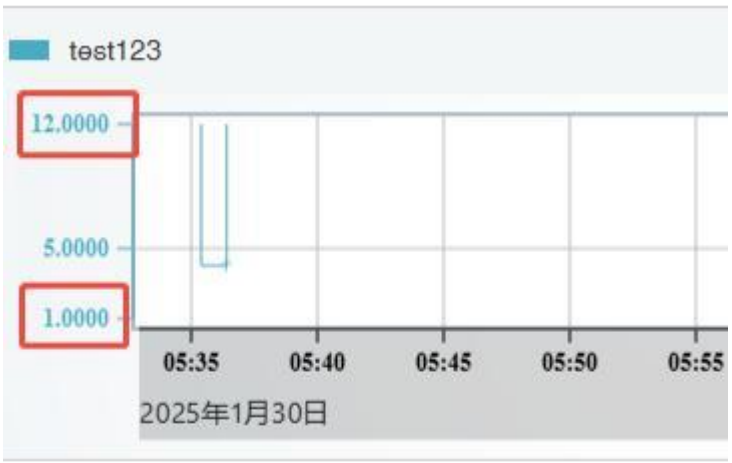
曲线（曲线名称）

修改该项后，文本趋势中，展示的曲线的名称将发生相应的变化，主要体现在图例与表格中



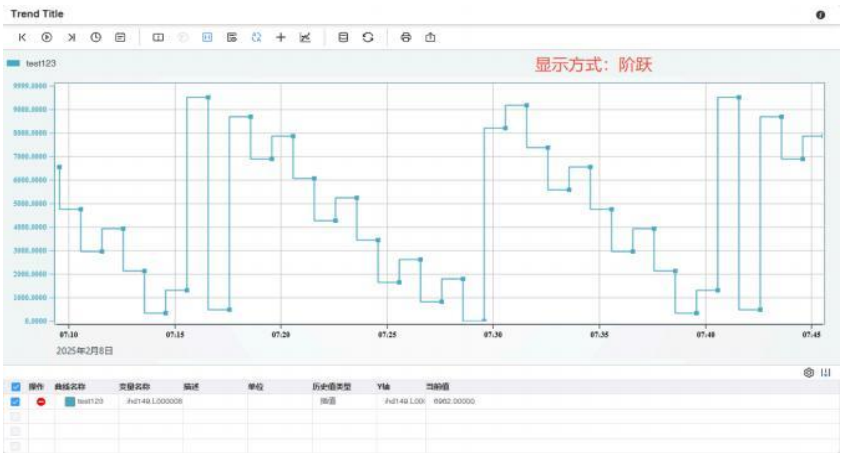
量程

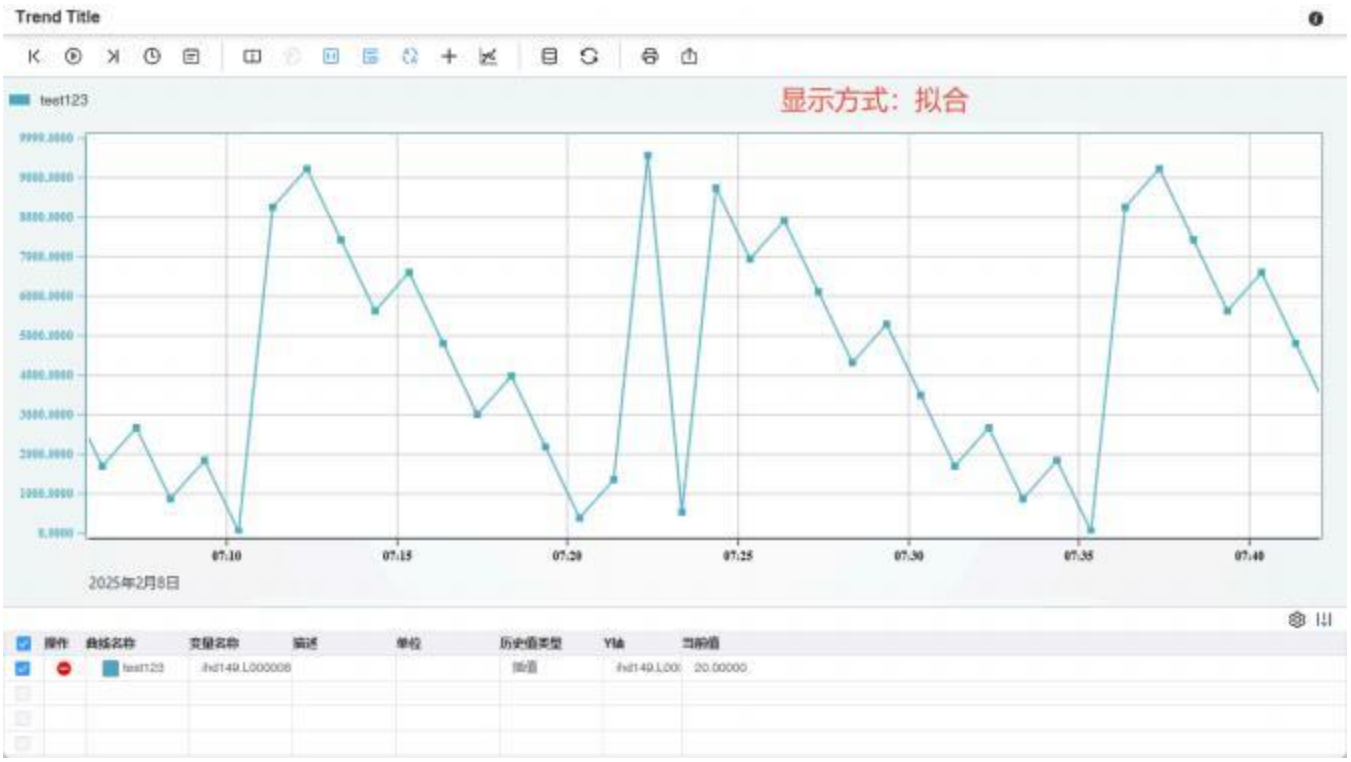
给该项输入有效值（上下限为有效数字，下限小于上限）后，打开的文本趋势将会自动开启自定义量程，并按照配置中进行量程范围设置



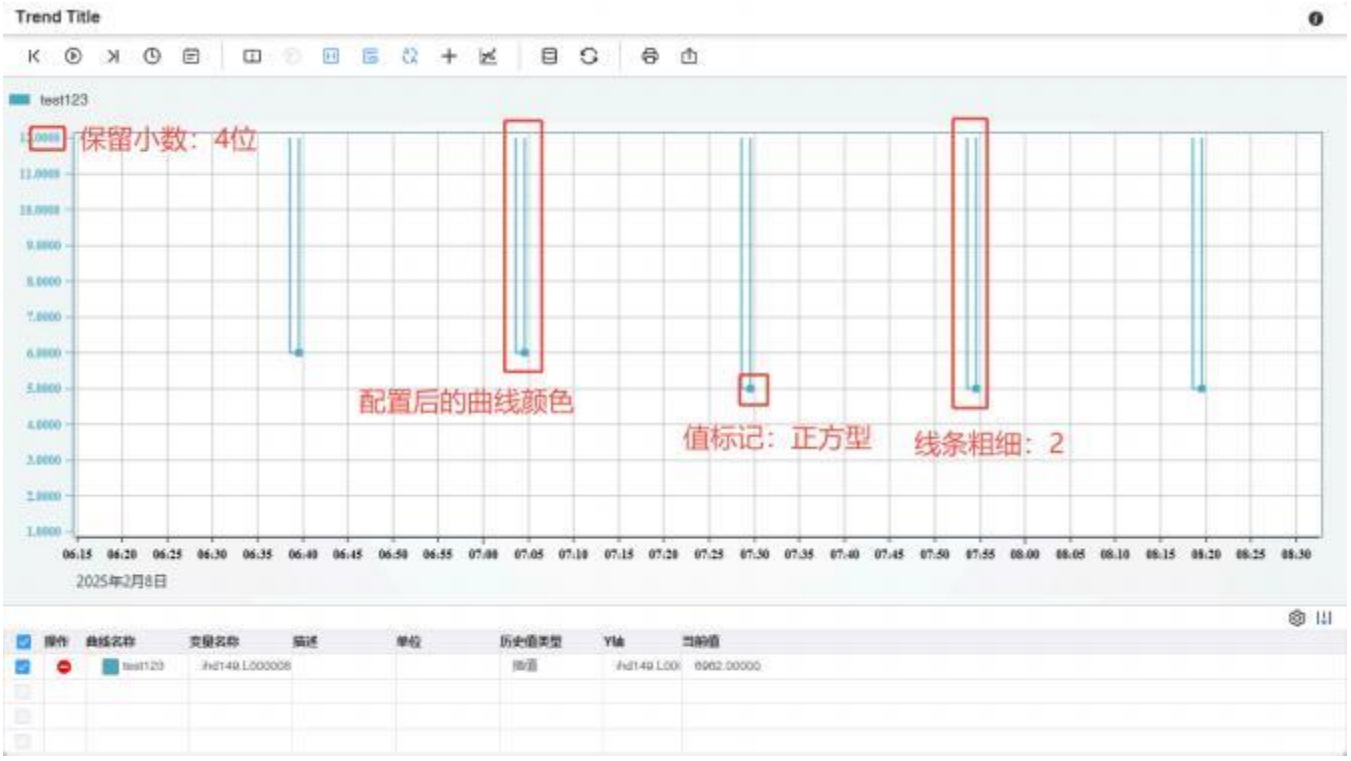
显示方式

分为阶跃曲线和拟合曲线（目前拟合曲线的表现形式，就是两个数据点之间用直线连接）





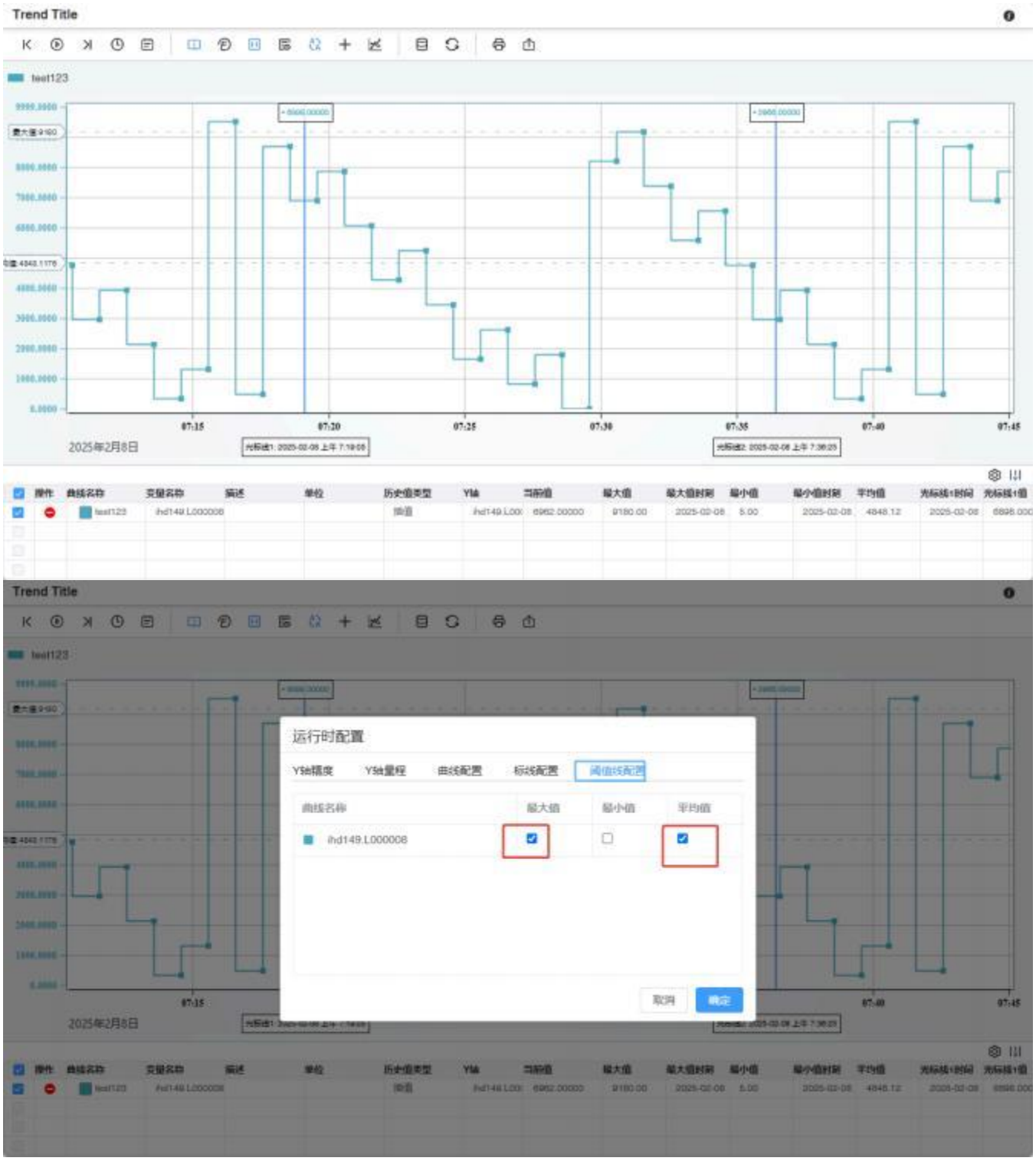
线型颜色/值标记/线型粗细/保留小数



若值标记被配置为“0-无”的情况下，打开“数据点突出显示”，则数据点形状默认为圆形（即“0-无”等价于“4-圆形）

统计值

统计值有三个勾选项：最大值，最小值，平均值。分别可以开启对应的阈值线。勾选后打开趋势，阈值线不一定第一时间可见，需要打开标线并计算统计值后，阈值线才会显现



运行态系统配置

文本趋势

基本配置

启用新文本趋势 启用

弹窗宽度 800

弹窗高度 700

全局时间格式 YYYY-MM-DD HH:mm:ss.SSS

全局小数位数 8

全局时间制 12小时制 24小时制

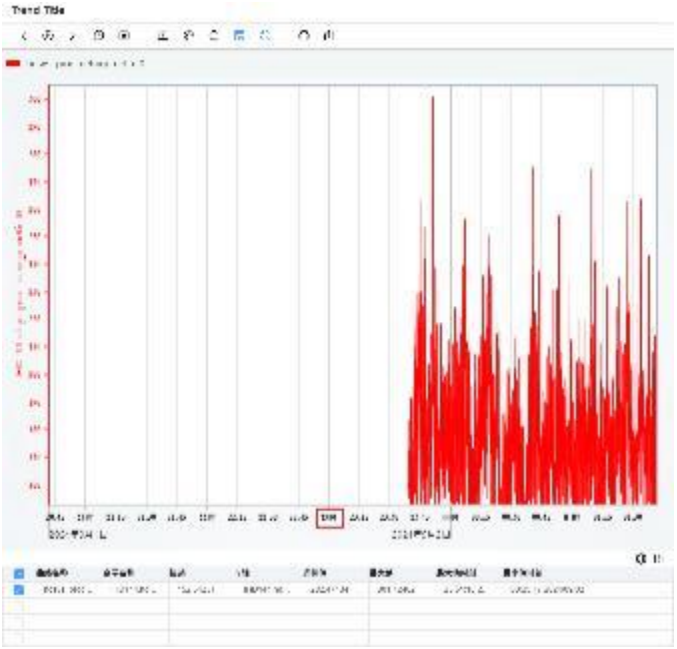
基本配置

- 弹窗宽度Y
  文本趋势的弹窗宽度（单位：px）
- 弹窗高度
  文本趋势的弹窗高度（单位：px）
- 启用新文本趋势
  - 开启新文本趋势（该配置关闭时，大部分配置项会被隐藏，因为这些配置项对旧文本趋势不生效）
  - 不会被隐藏的配置项有：弹窗宽度，弹窗高度，文本趋势打开方式
- 全局时间格式
  趋势图上所有显示时间的位置，格式按照此设置显示，默认显示是“1970-10-10 10:00:00”

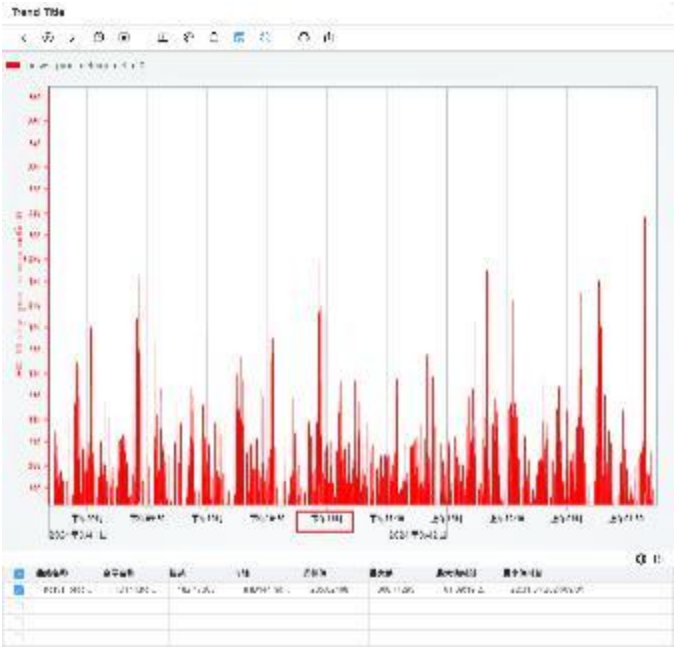
  设置包括：

  1. tooltip的显示
  2. 阈值线的显示
  3. 表格的显示
  4. 标线的显示
  5. 导出excel的显示
- 全局时间制
  设置12小时或者24小时，与全局时间同步使用

24H



12H



- 全局小数位数

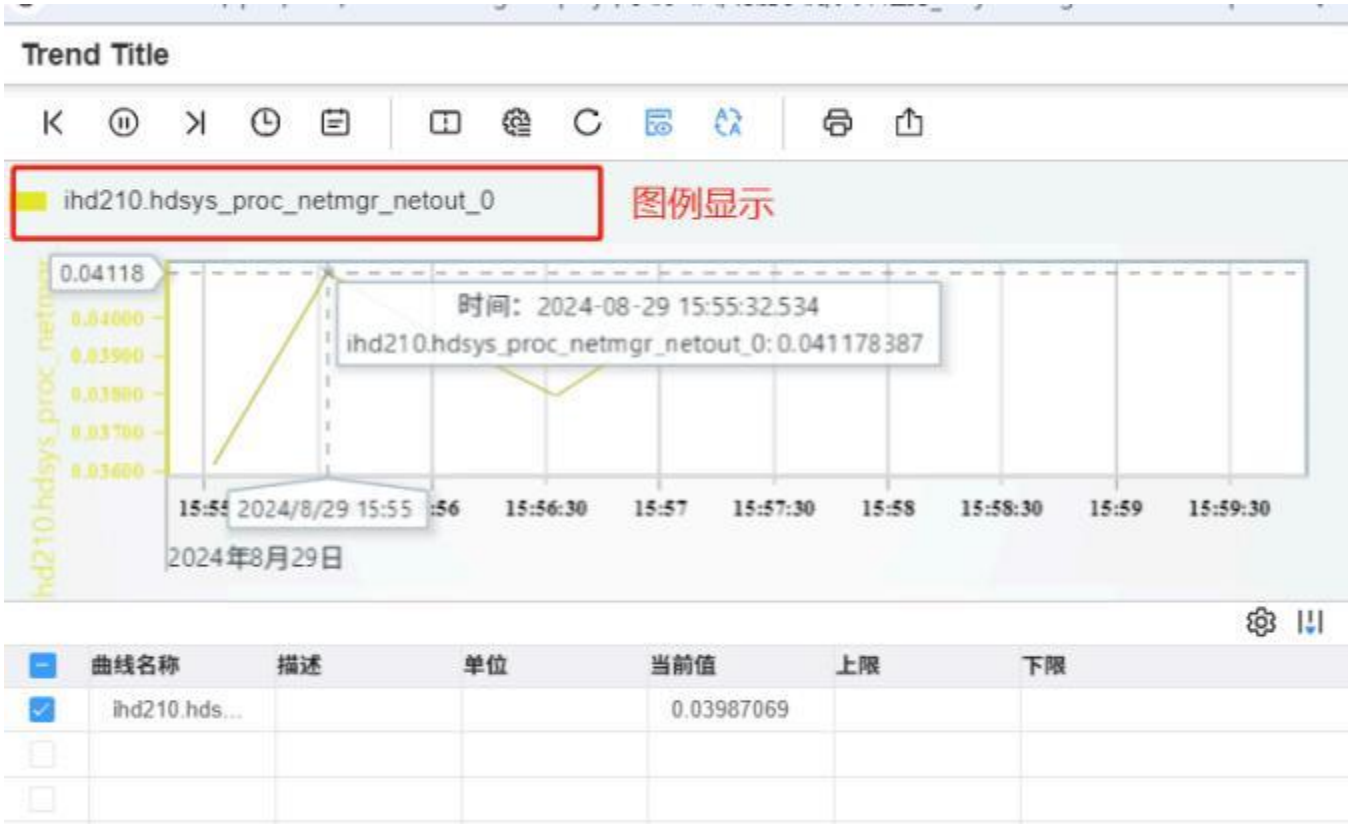
  趋势上所有位置的小数位数的设置。但是如果设置了文本属性，或者运行态的小数位数，均覆盖全局配置。

  设置包括：

  1. tooltip的显示
  2. 阈值线的显示
  3. 表格的显示
  4. 标线的显示

- 图例

显示线名
用于控制图例的显隐。

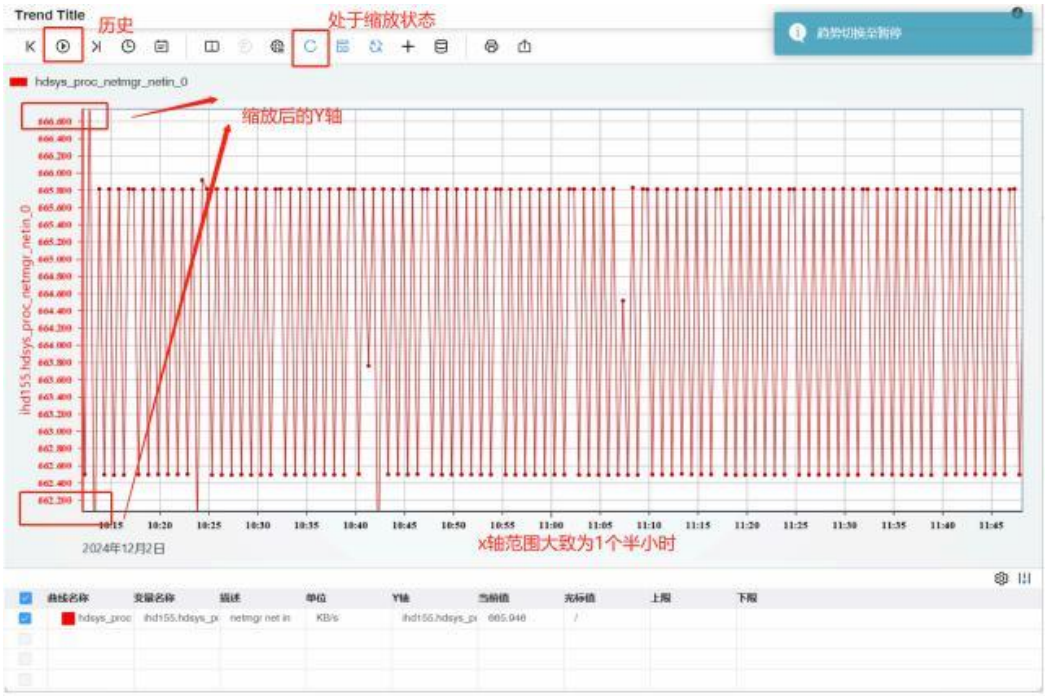


•图表

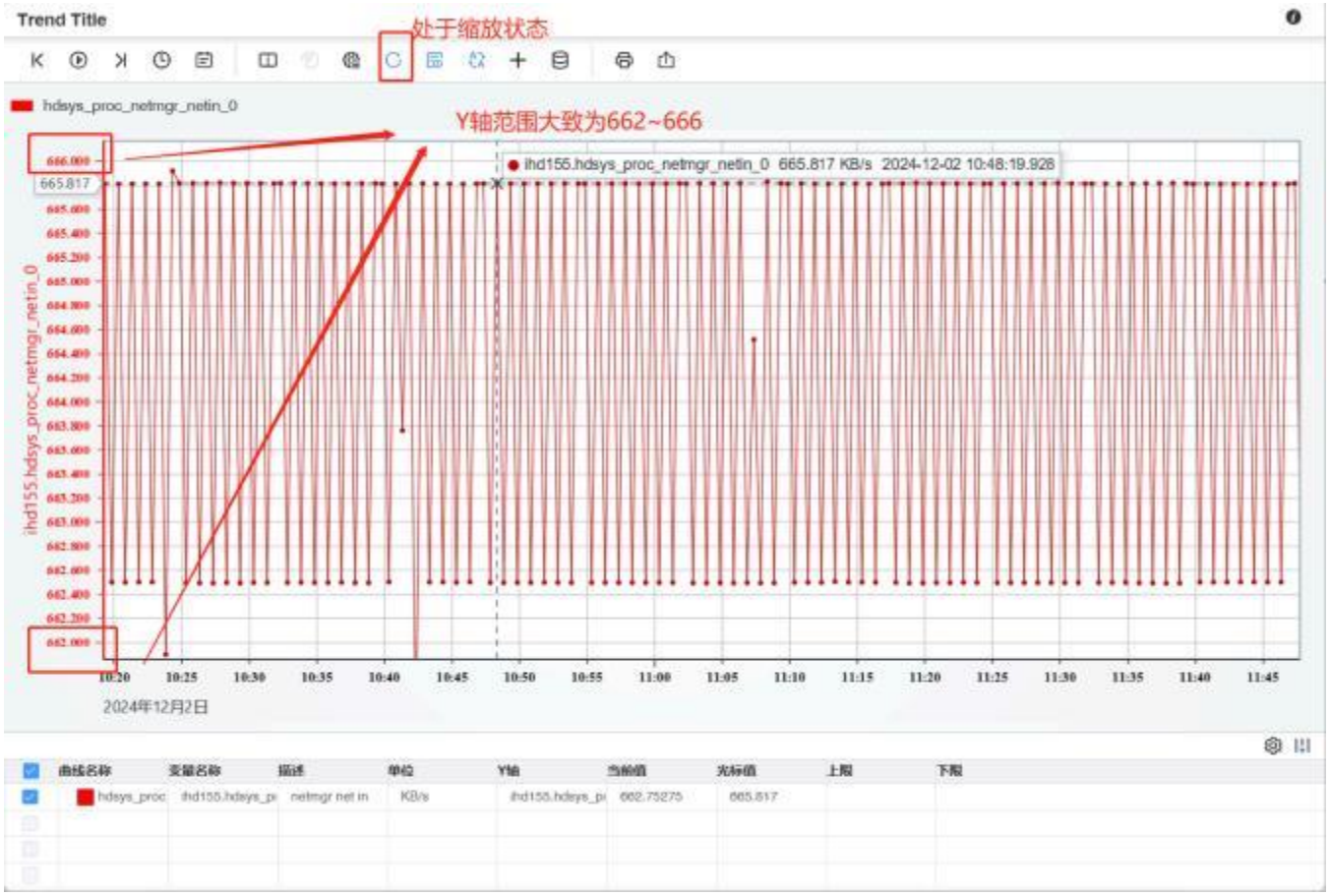
用于控制图例的显隐。

•实时保持缩放

关闭时，文本趋势从历史切换到实时，如果图表处于缩放状态，则会重置缩放（即如果切换到实时前，“区域重置缩放”按钮亮起，则切换到实时后，图表会自动执行一次区域重置缩放操作，使得“区域重置缩放”按钮不再亮起）

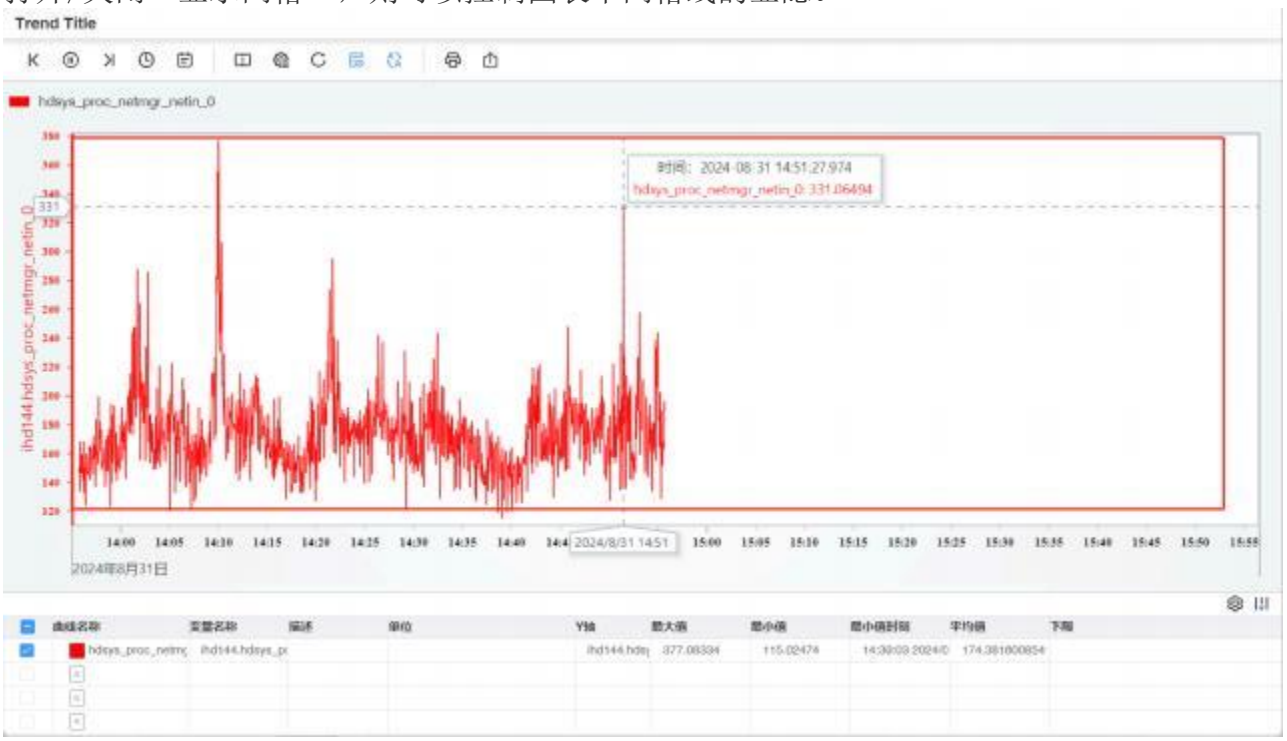


开启时，文本趋势从历史切换到实时，如果图表处于缩放状态，则X轴和Y轴的缩放状态都会被保留（譬如：切换到实时前，y轴范围为1020，x轴长度为1小时，并且“区域重置缩放”按钮亮起；切换到实时后，y轴范围还是1020，x轴长度还是1小时）



•显示网格

打开/关闭“显示网格”，则可以控制图表中网格线的显隐。

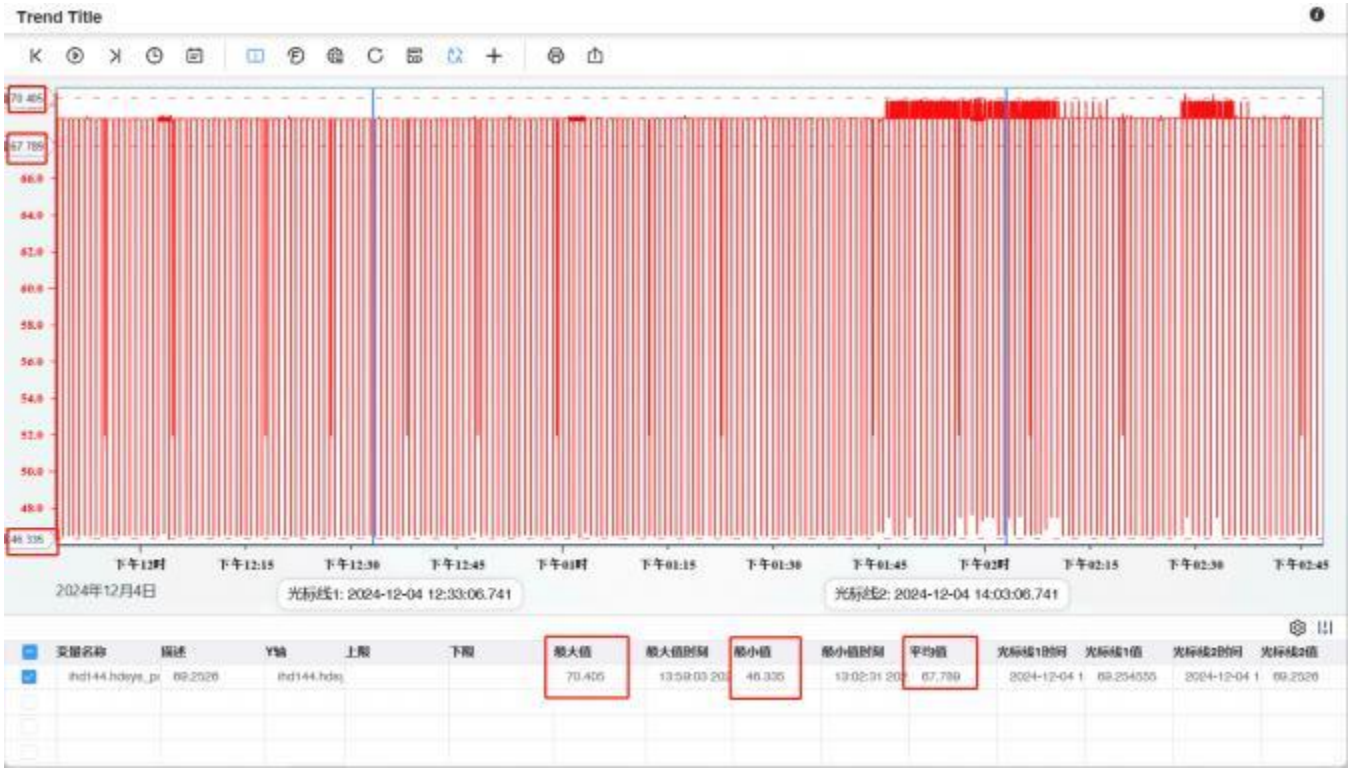


上图中，左侧为关闭网格线的效果，右侧为打开网格线的效果

•统计值位数

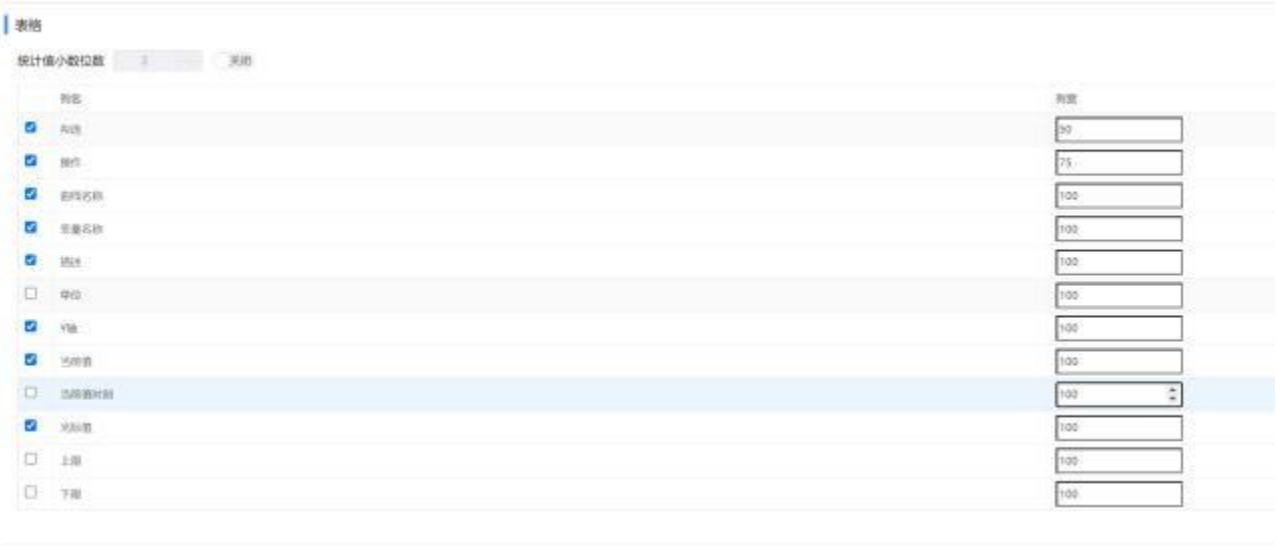
开启后，可以设定文本趋势中，计算出的统计值，需要保留到小数点后几位。

统计值位数会影响趋势表格中“最大值”、“最小值”、“平均值”这三列的位数，也会影响曲线阈值标签上的位数



上图中，为打开统计值位数（精确到小数点后两位）的效果

t•表格



表格栏
用于配置文本趋势表格中显示哪些列，以及这些列的列宽为多少

统计值小数位数
用于配置表格中统计值的小数位数显示。不影响统计值在图上的显示。

- 标线

标线宽度/标线颜色
用于修改文本趋势标线的样式（其中，需要确保“标线宽度”为正整数）

- 曲线

  - 条件限制

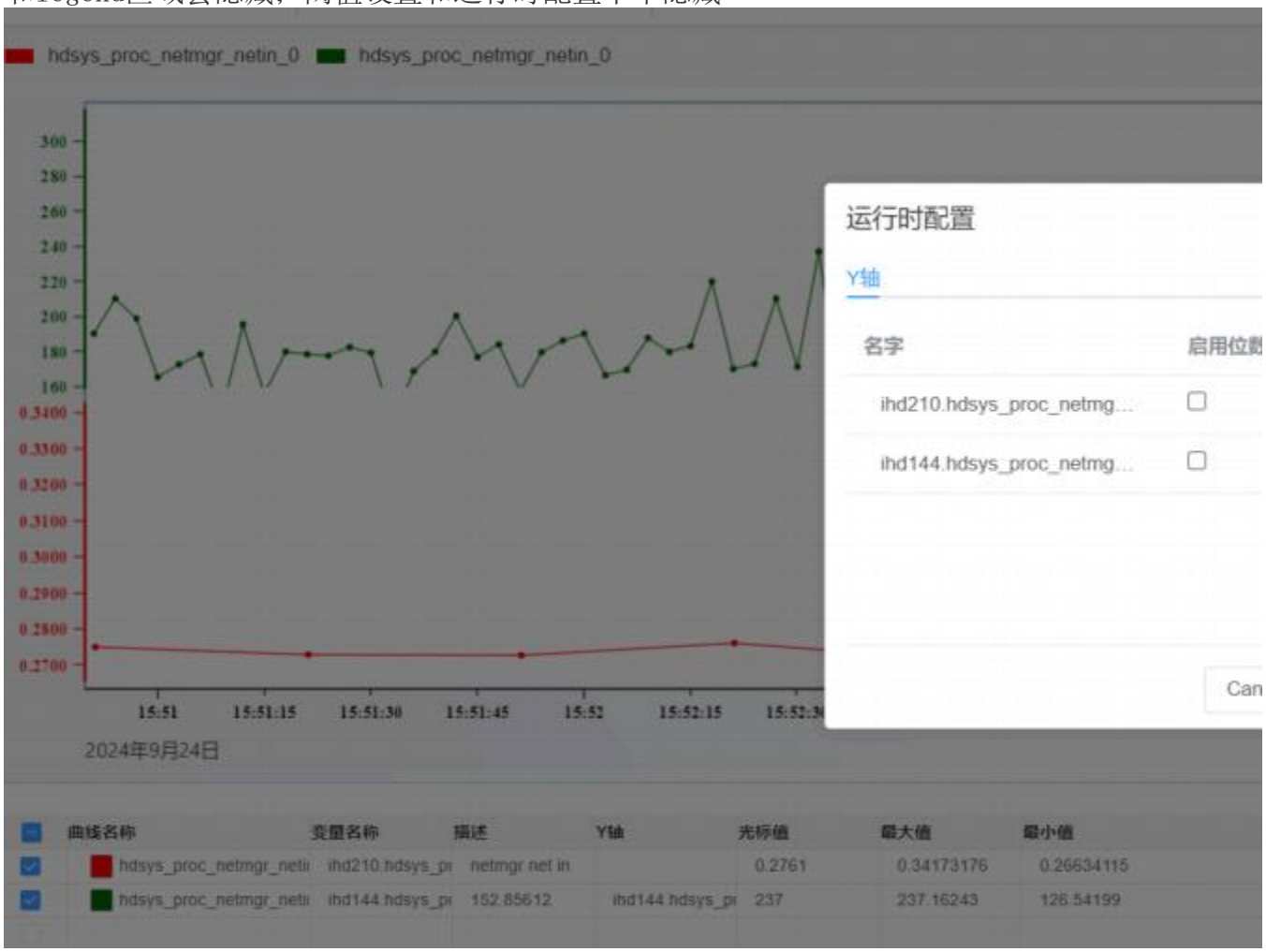
    限制单个文本趋势窗口可以最多展示的曲线条数（**可配置范围为：0~8条**）

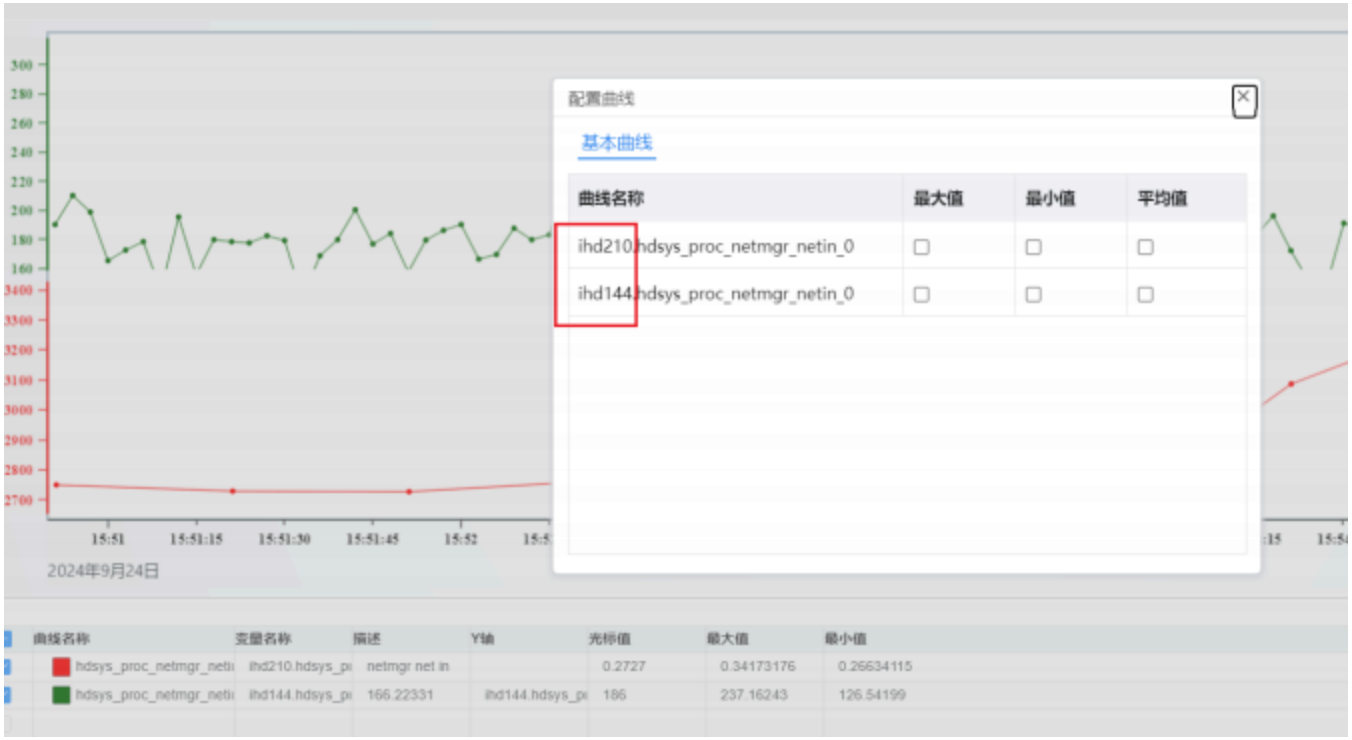
  - 默认线宽

    线宽会应用在未配置曲线宽度的地方：1.在趋势中添加曲线2.通过sys.openTextTrend打开

曲线名称隐藏节点名

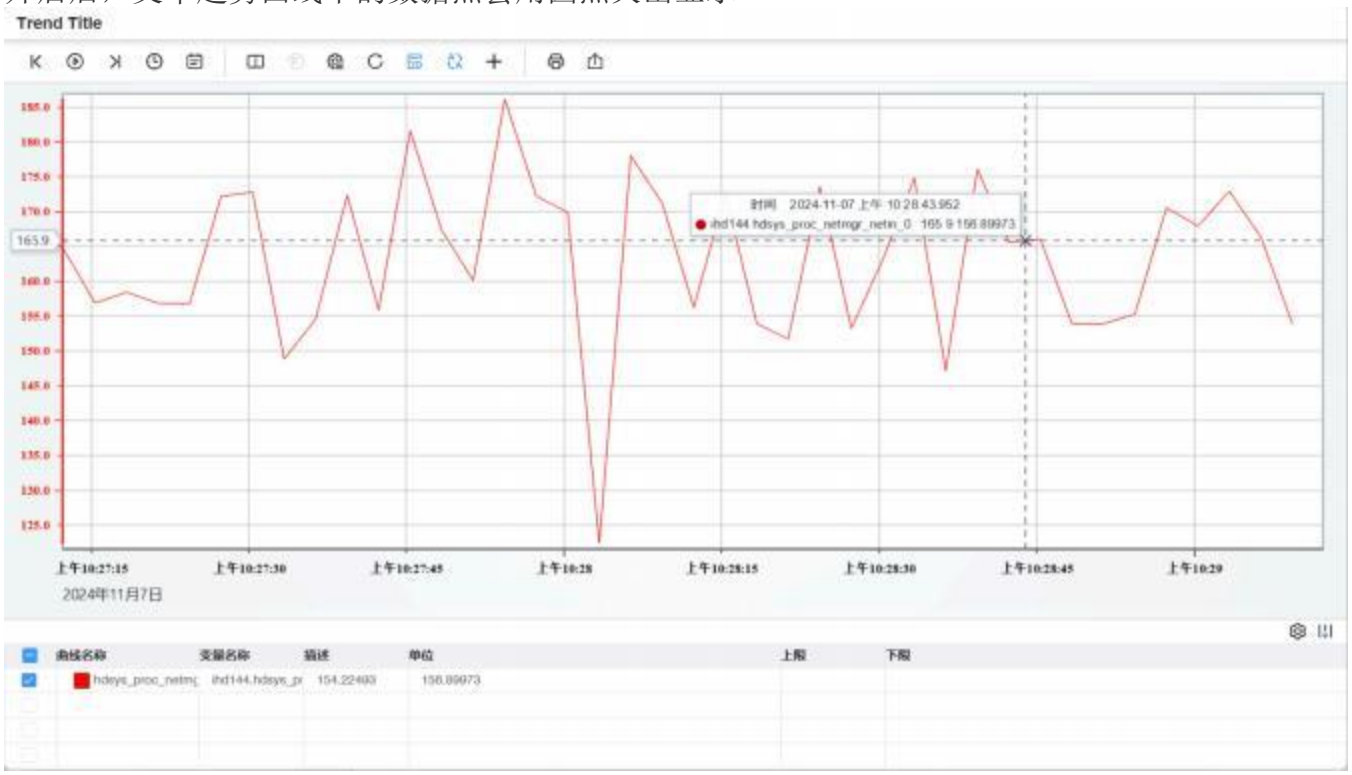
开启后tag点的节点名会被隐藏（eg：IplatTest.text_1会显示为text_1）在表格中的曲线名称和legend区域会隐藏；阈值设置和运行时配置中不隐藏

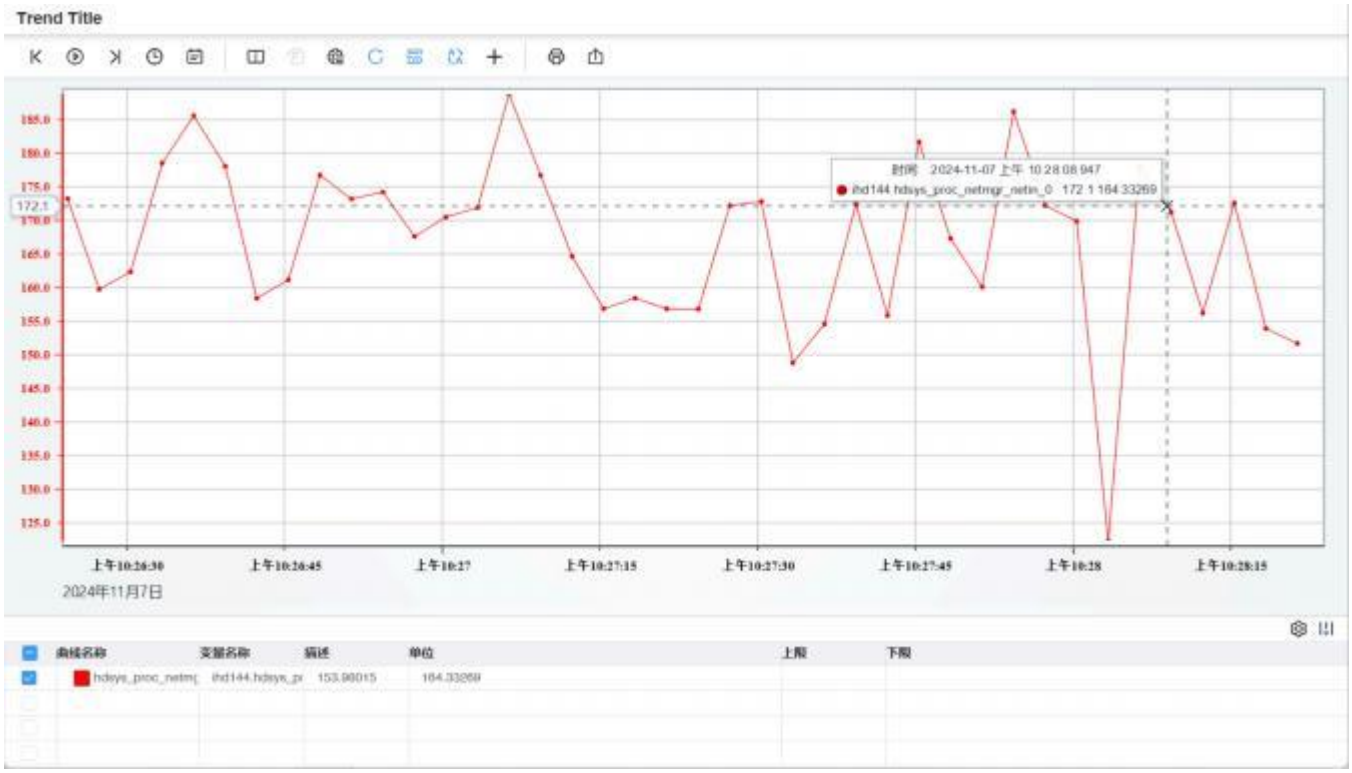




数据点突出显示

开启后，文本趋势曲线中的数据点会用圆点突出显示





上图中，左侧为关闭数据点突出显示的效果，右侧为打开数据点突出显示的效果

当曲线的数据点较为密集时，数据点会被自动隐藏

- 单条曲线插值上限
  当使用插值法绘制曲线，曲线的数据大致会在该数值左右

注：1.即便原始值只有1000个，此处配置为25000，也会用插值法获取25000左右个点2.

此处的25000上限，不一定完全精准，大致在25000左右

- Y轴

使用Y轴工程量程

开启后会将Y轴的工程量成设置为上下限，**注意：如果此处开启，则会去读取上下限，如果没有读取到，则会将上下限设置为0~10。BA5.1.2之后，使用域名F_VALUE_LOW、F_VALUE_SPAN替换原本的F_EGU_LOW、F_EGU_SPAN，能够确保上限大于下限**

- Y轴分布样式
  有横向和纵向之分

上图中，左侧为y轴横向分布，右侧为y轴纵向分布

•注意：

1. 如果开启Y轴分布样式的纵向，则使用Y轴工程量程会失效（目前纵向分布的情况的缩放，只缩放x轴，不会缩放y轴）
2. 如果开启Y轴分布样式的纵向，则显示y轴标题配置项会失效（目前纵向分布的情况下，y轴标题默认不显示）
3. 如果开启Y轴分布样式的纵向，则显示y轴位置配置项会失效（目前纵向分布的情况下，y轴位置默认全部显示在左侧）

•显示Y轴标题

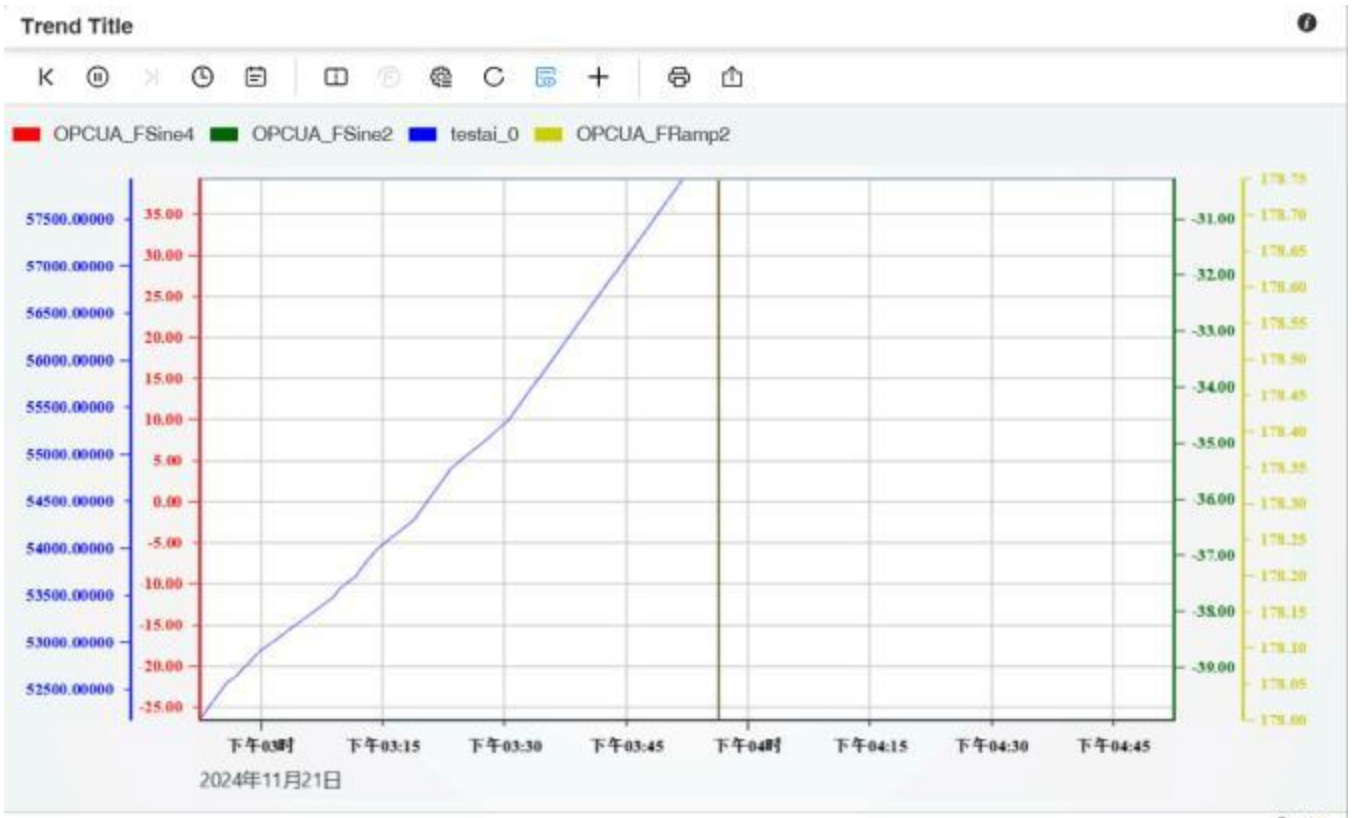
y轴标题的开关（**y轴纵向分布时无法配置，会默认关闭**）

上图中，左侧为关闭标题的效果，右侧为显示标题的效果

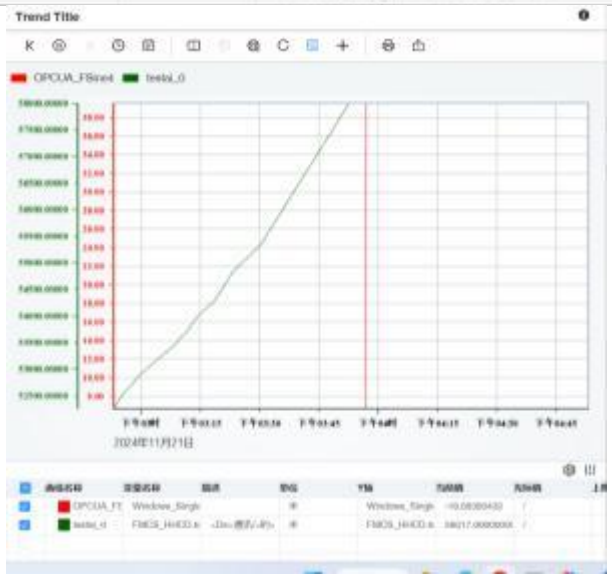
•Y轴位置

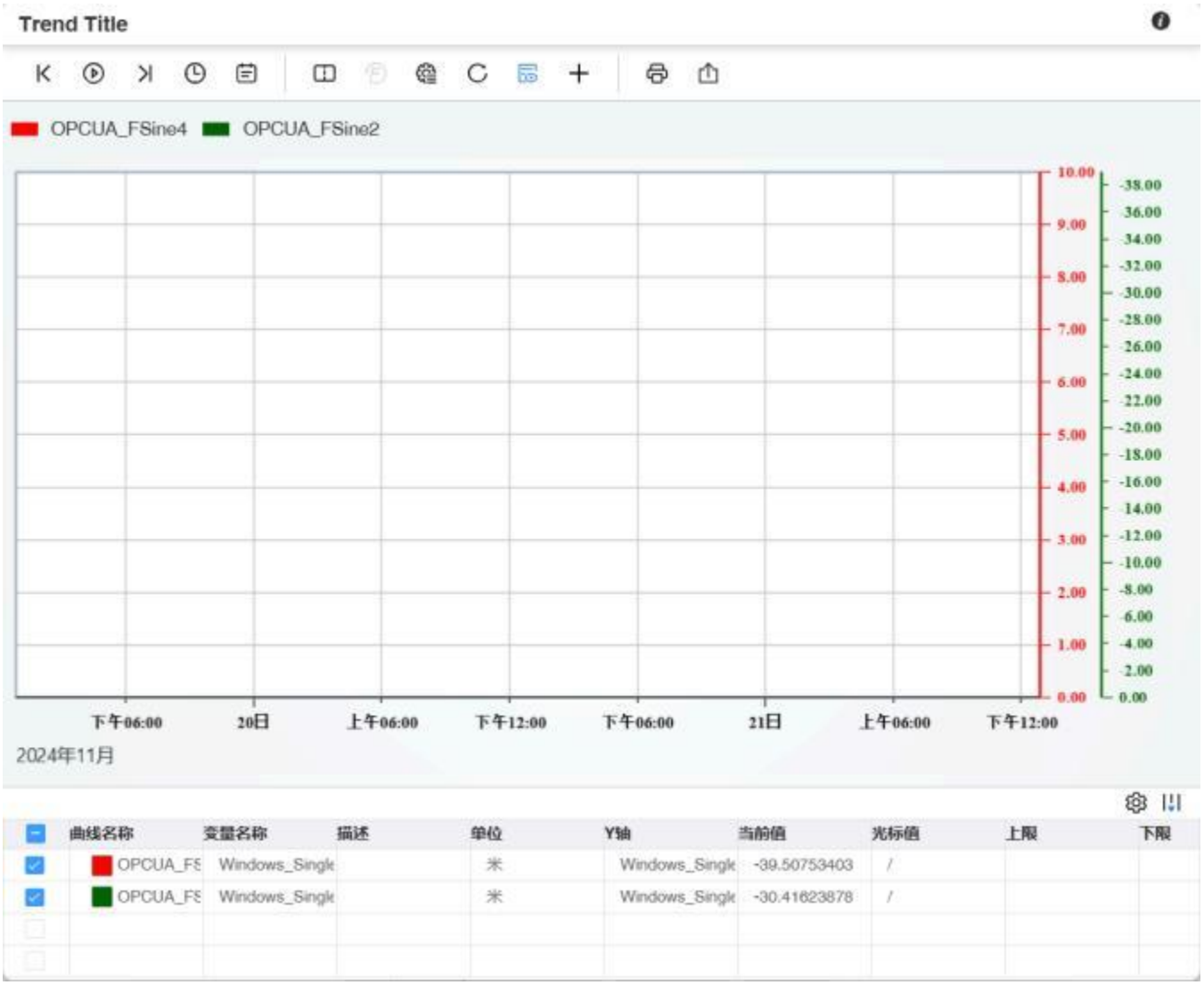
用于设定Y轴的左右位置（y轴纵向分布时无法配置，会默认Y轴全部分布在左侧）

改配置项提供了三种选项：均匀分布在两侧，全部分布在左侧，全部分布在右侧



曲线名称	变量名称	描述	单位	Y轴	当前值	光标值	上限	下限
OPCUA_FS	Windows_Single		米	Windows_Single	5.08812428	/		
OPCUA_FS	Windows_Single		米	Windows_Single	2.54924440	/		
testai_0	FMCS_HHCD.te	<De>通讯/<的>	米	FMCS_HHCD.te	57938.0000000	/		
OPCUA_FF	Windows_Single			Windows_Single	174.75000000	/		





上图为配置的效果，从左到右依次为：（1）均匀分布在两侧（第一条Y轴在左侧，第二条右侧，第三条左侧）（2）全部分布在左侧（3）全部分布在右侧•Y

轴小数位数

Y轴小数位数，可以在以下三种场景下设置，优先级依次递增：

1. 系统设置-----当前处设置
2. 文本元素右侧属性
3. 运行态

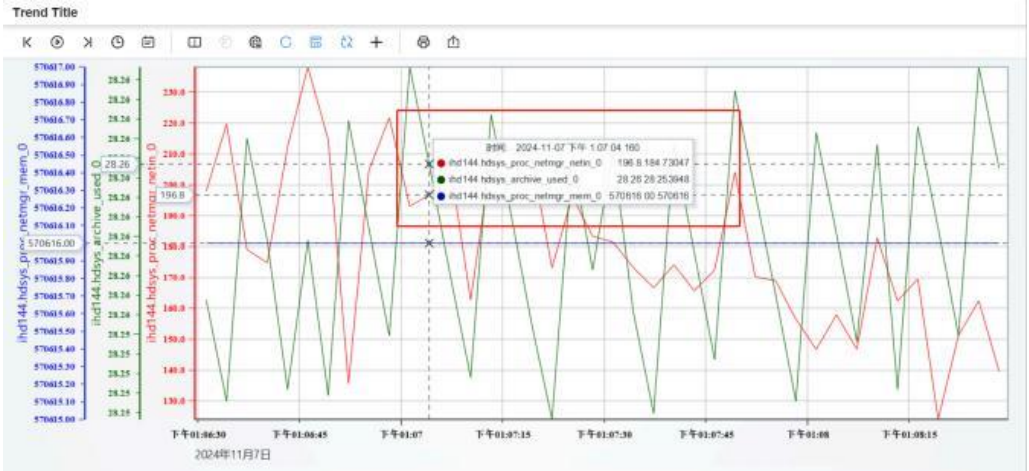
当Y轴小数位数都不启用时，采用系统设置的全局小数位数。

- Tooltip

启用插值

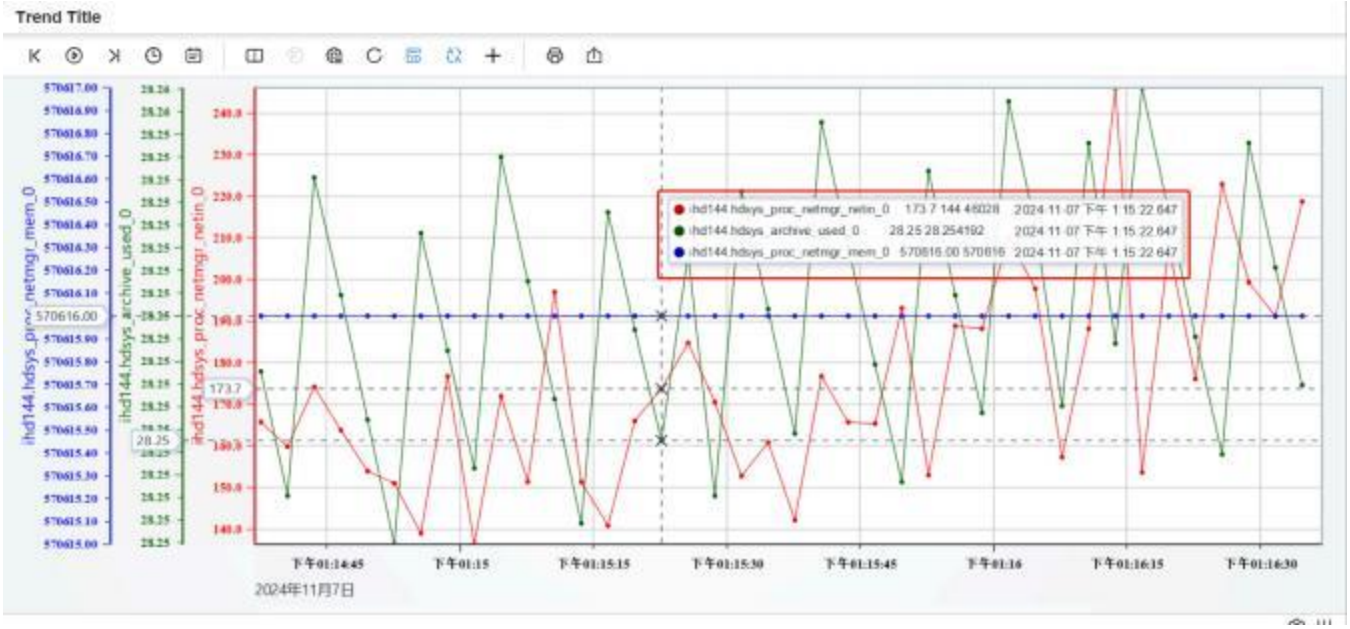
tooltip展示数据时是否会根据实际值自动生成插值的开关

1.启用插值:曲线中不会添加插值点，而是tooltip根据曲线中的实际数据点模拟出插值，来使tooltip的移动更加平滑。此时，多曲线下的tooltip效果如下图所示



从上图可以看出，因为tooltip模拟出了插值数据，所以来自不同曲线的数据点共用一行时间信息进行展示

2.关闭插值：tooltip只会吸附在曲线中的实际数据点上并进行展示。此时，多曲线下的tooltip效果如下图所示



从上图可以看出，因为tooltip只会展示曲线的实际数据，所以来自不同曲线的数据点都会具有独立的时间信息

- tooltip字体

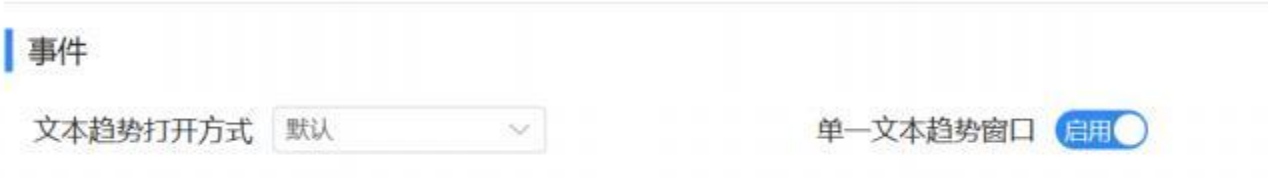
  用于配置tooltip字体大小

Tooltip字体不可以配置过小，否则会导致趋势无法正常显示！推荐tooltip字体大小配置范围为8~32

- 默认颜色组

不启用默认颜色组，则默认颜色组为系统内置的颜色组。若开启，用户可以自定义默认颜色组。这些颜色组会应用于文本趋势中未配置颜色的曲线。

•事件



文本趋势打开方式
目前文本趋势打开方式分为三种【默认】、【单击】和【双击】

如果一个文本框，配置了tag点，那么他可以查看这个tag点的文本趋势

注：当配置的是【单击】或者【双击】时，移入文本框时，文本需要有下划线，并且鼠标变成手型，提示可以点击。

且在文本上右键打开的菜单中【文本趋势】去除。

默认

单击

双击

•历史趋势配置

实时

用于设定文本趋势打开时，是处于实时状态还是暂停状态

•数据比例

用于设定实时状态时，曲线占据画面的比例

左边图片为“数据比例”设置为50的效果；右边图片为“数据比例”设置为100的效果

只在【实时】开关开启时，才可以对该配置项进行设置，允许设置的值得范围为0~100•X轴时间

用于设定X轴时间的范围，具有“时间跨度“和“时间范围”两种设定方式可供选择“时间范围”设定方式，只能在“实时开关”关闭的情况下才可以选择

该配置的使用方式有三种：

一、使趋势打开为实时状态：

【步骤】：

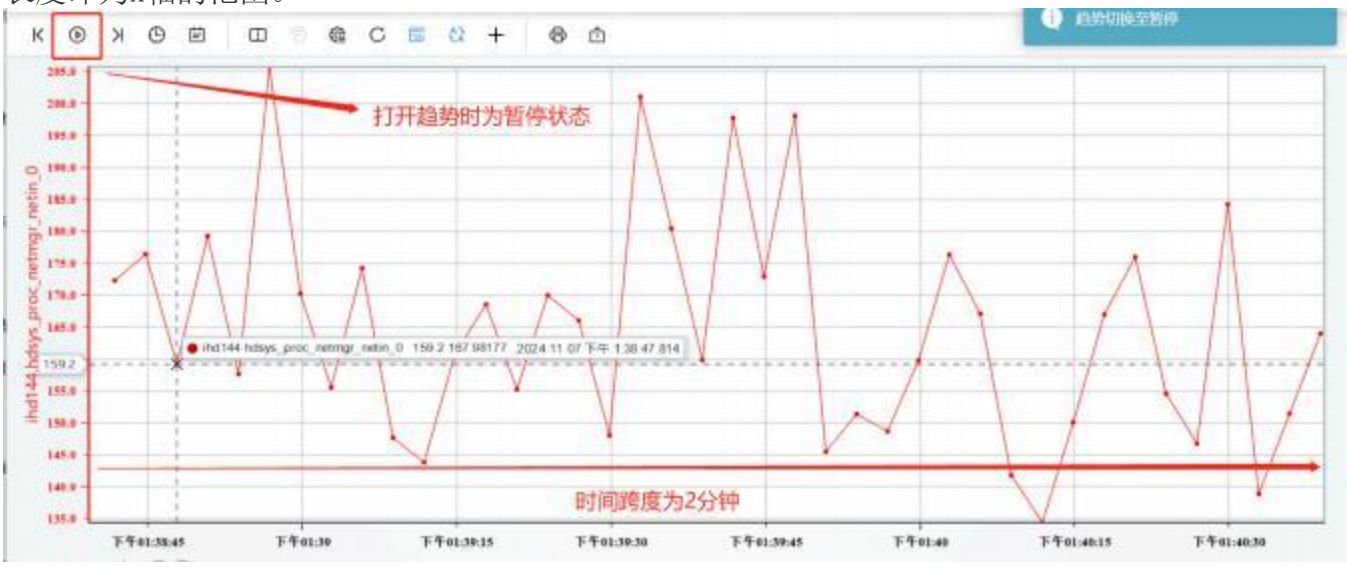
（1）开启“实时”按钮。X轴时间此时只能选择用“时间跨度”进行配置（2）调整时间跨度中的值和时间单位，就可以谁设定x轴展示的范围

【效果】：趋势打开后为实时状态。趋势展示的时间跨度（图表x轴的范围）与配置相同。

二.、趋势打开为历史趋势，展示最新一段时间的数据

【步骤】：（1）取消勾选“实时”按钮，（2）勾选“时间跨度”按钮。

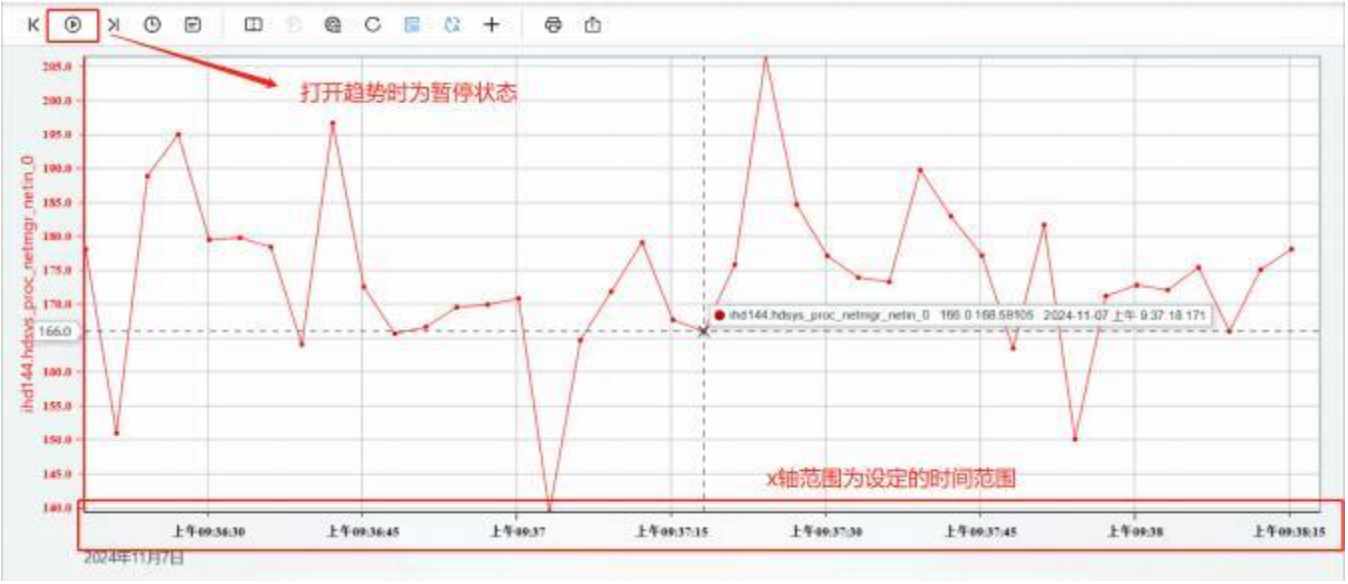
【效果】：此时打开趋势，趋势会处于暂停状态。并展示最新一段时间的历史趋势。这段时间的长度即为x轴的范围。



三、趋势打开为历史趋势，指定展示某一段时间的数据

【步骤】：（1）取消勾选“实时”按钮，（2）勾选“时间范围”，就可以对趋势需要展示的时间段进行设定

【效果】：此时打开趋势，展示指定的那段时间的历史趋势。（即展示的是上图中配置的2024-11-07 09:36:18~2024-11-07 09:38:18这段时间的数据）



【比例设定】

“时间范围”配置项，还具有一个特殊的功能：通过时间范围选择框后面的输入框，还可以进一步设定：需要展示这段时间范围末尾百分之多少的数据

例如：修改配置中的比例，譬如修改比例为20。那么，趋势会展示用户指定时间段中，末尾20%的数据。（下图时间范围为11月7日早上9点到下午14点，跨度5小时，展示末尾20%的时间的话，就是展示1小时的数据，即13点到14点的数据）

脚本接口

按钮显隐控制

【接口】sys.enableTextTrendButton(buttonInfo)

【参数】buttonInfo按钮显隐信息，类型为Object，键值对为：`{按钮名称：true/false}`接口目前可以控制的按钮列表如下

按钮名称	说明
Page-Forward	向前翻页

TBPreview	切换至实时
Page-Back	向后翻页
Page-Turning-Time-Length	翻页时间长度
midline	显隐标线
Threshold-Configuration	阈值配置
Time-Setting	时间选择
TBReload	区域缩放重置
Print-Trend	打印
Export	导出
Tooltip	显隐Tooltip

【示例】禁用打印按钮和向前翻页按钮

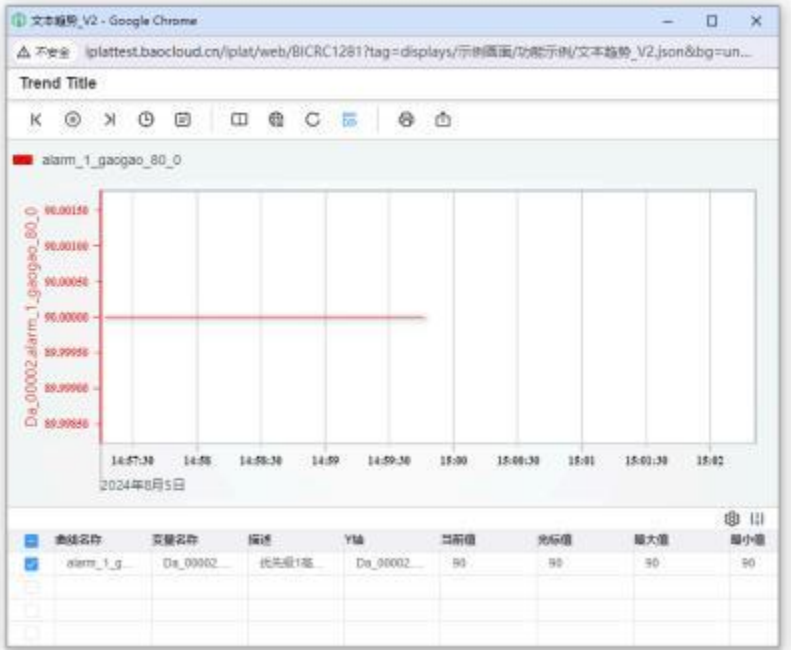
```
    sys.enableTextTrendButton({
        'Print-Trend': false,
        'Page-Forward':false,
    });
};
```

趋势界面

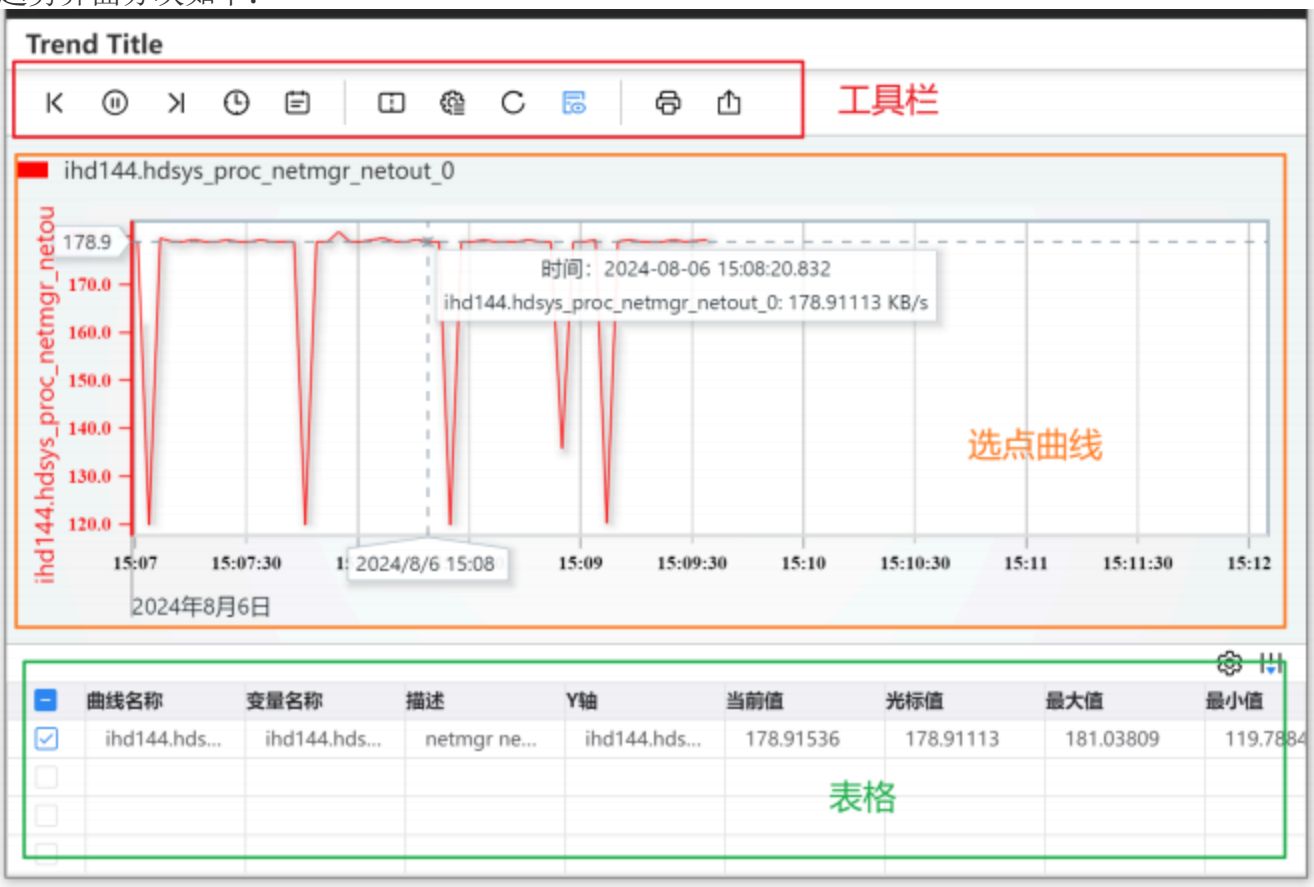
在画面预览状态，右键该文本，选择打开趋势：

打开后，历史趋势如下：

90



趋势界面分块如下：



工具栏

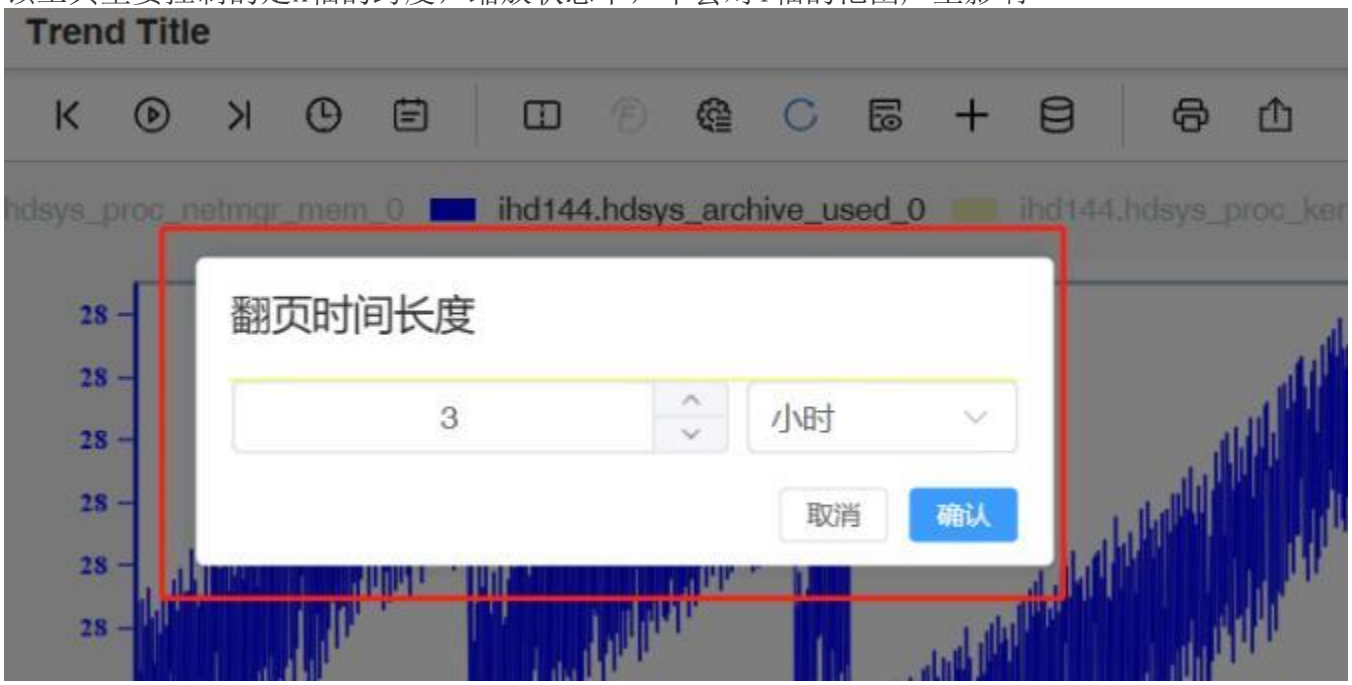
时间控制工具

- 调整趋势中的时间区域，包括：向前翻页、切换至暂停(或实时)、向后翻页、翻页时间长度、时间选择。

翻页时间长度：

控制翻页时长，用户可以更改翻页时长的数据及时间单位，点击“确认”按钮后生效，点击“取消”按钮以取消更改。

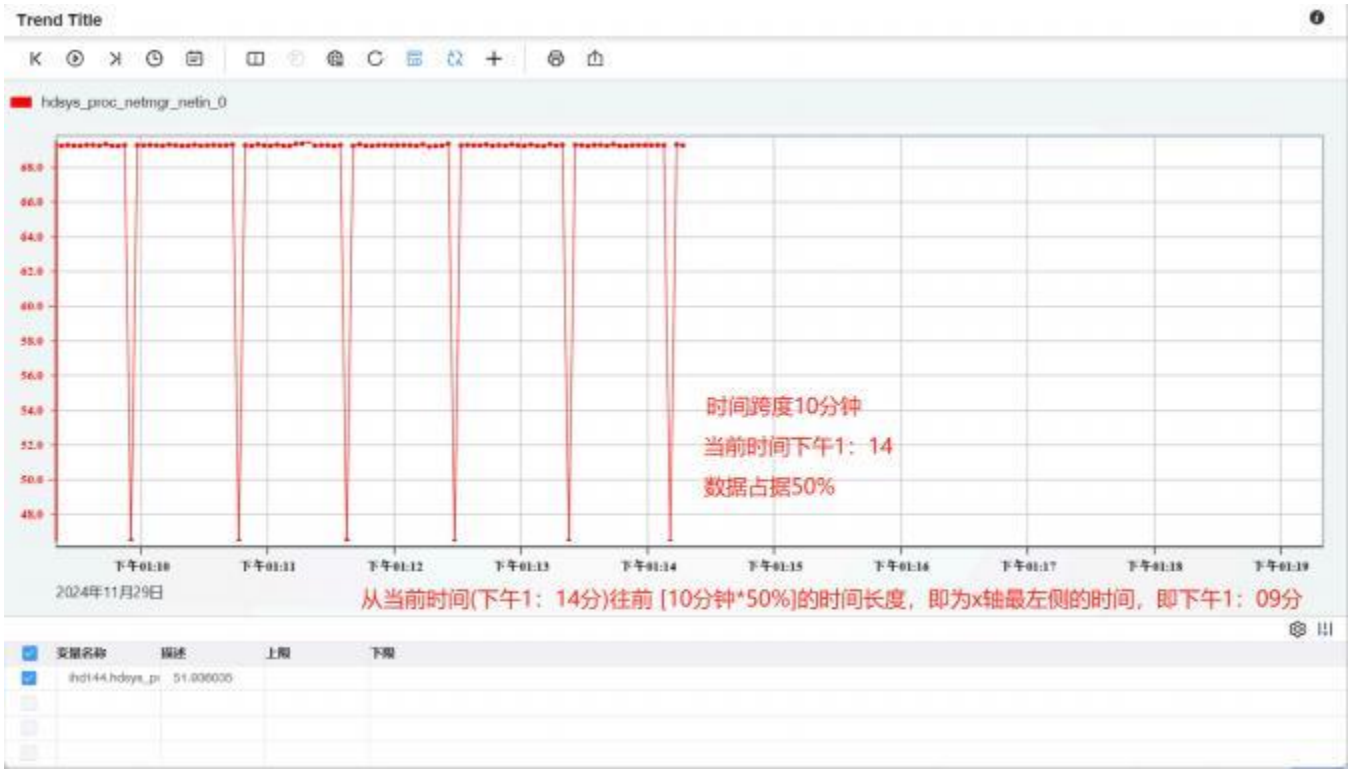
该工具主要控制的是X轴的跨度，缩放状态下，不会对Y轴的范围产生影响



以下为常见场景效果：

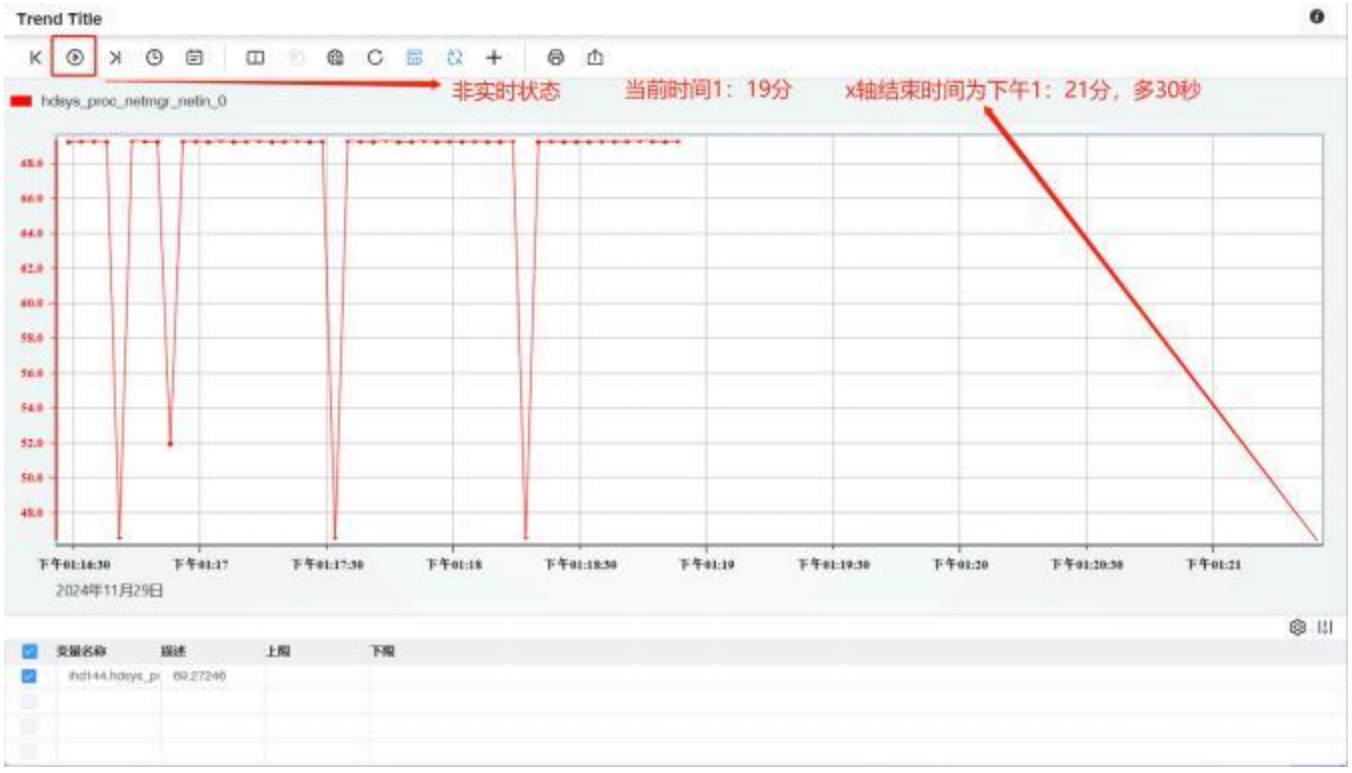
1.实时状态下切换

假设初始翻页时间长度为5 min，数据展示50%，将其更改为10 min后，x轴的时间间距更改为10min，并且数据依然展示50%

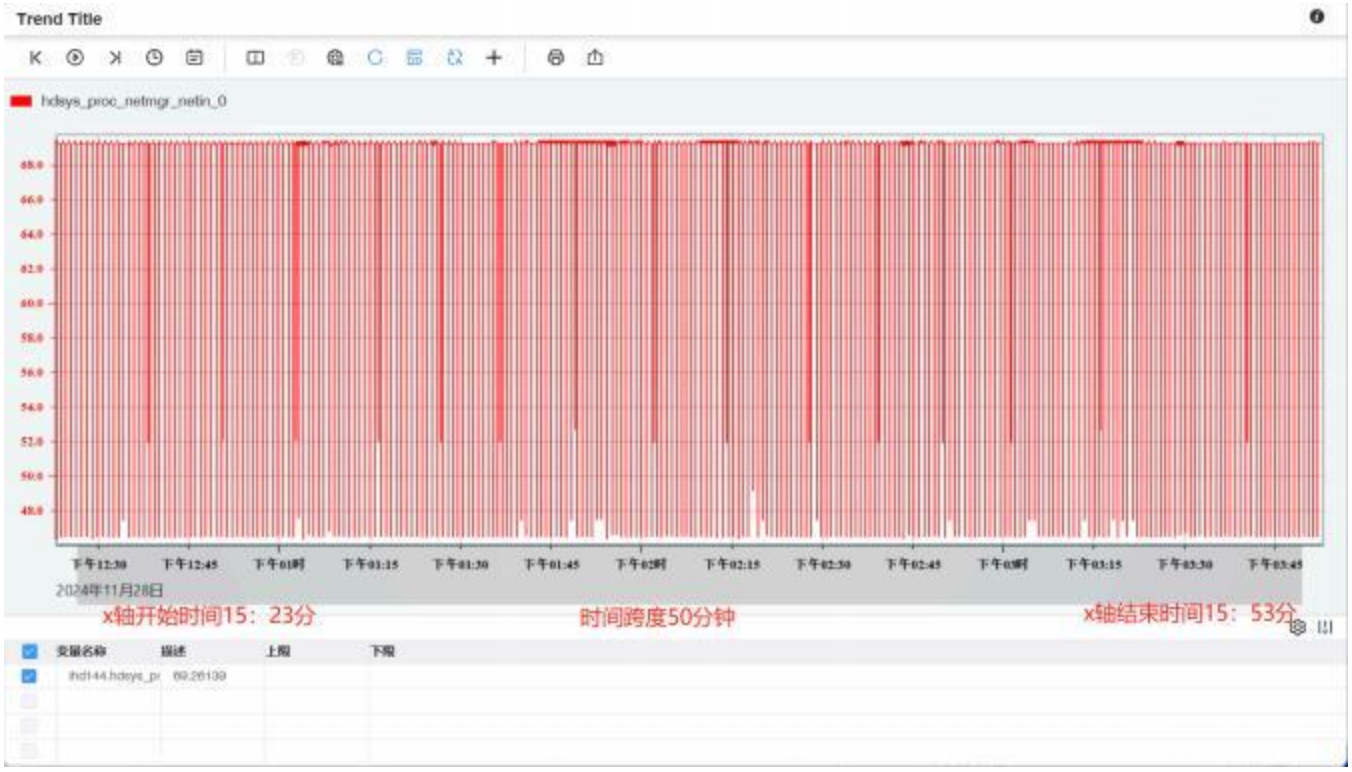


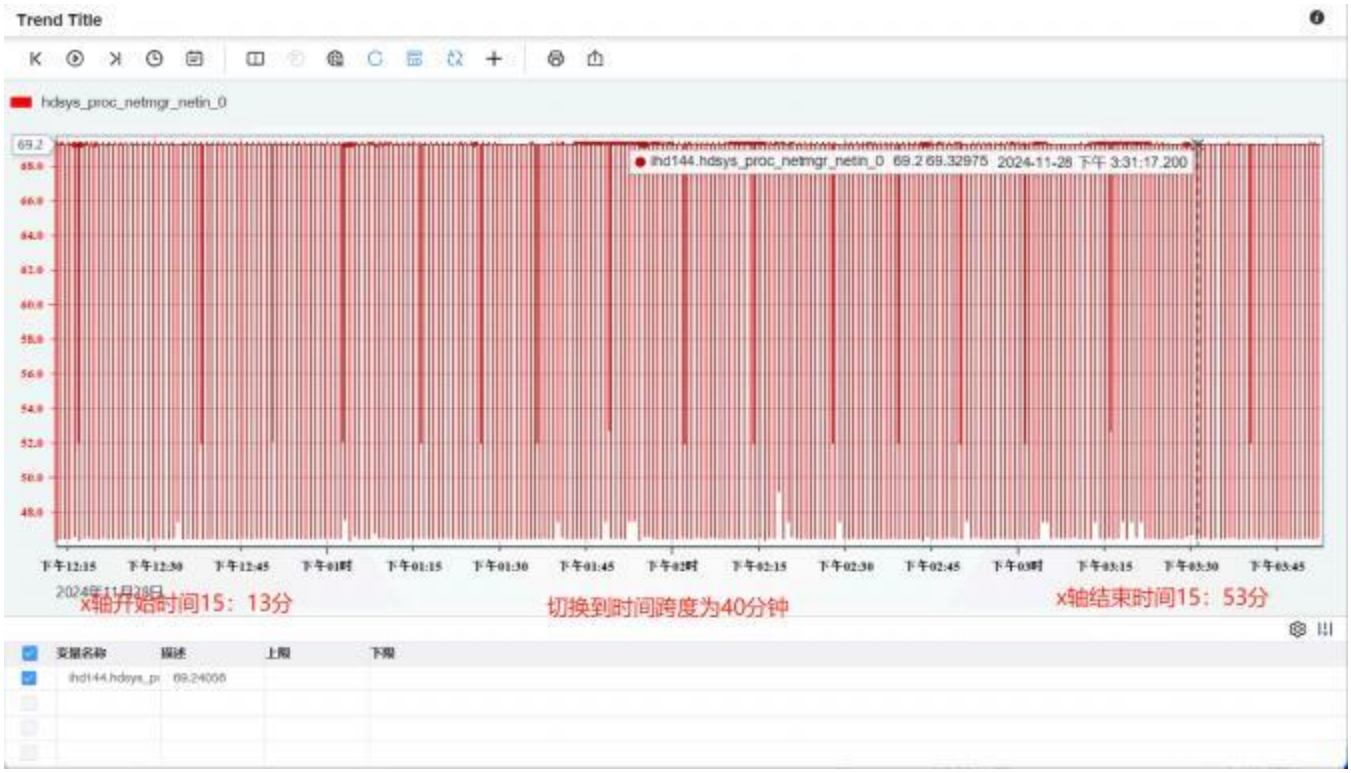
2. 非实时状态下切换

如果之前x轴的结束时间比当前时间晚，那么切换之后（假设设置为30分钟），那么切换之后的x轴的结束位置为当前时间，x轴的开始位置为当前时间减去30分钟



如果之前x轴的结束时间早于当前时间（例如往前翻页之后又切换翻页时长为40分钟），那么切换之后的x轴的结束位置为之前的结束时间（15:53），x轴的开始位置为结束时间减去翻页时长（15:13）





向前翻页：
以当前开始时间为最终时间，向前查询一个翻页时间长度的趋势数据。
场景效果如下：

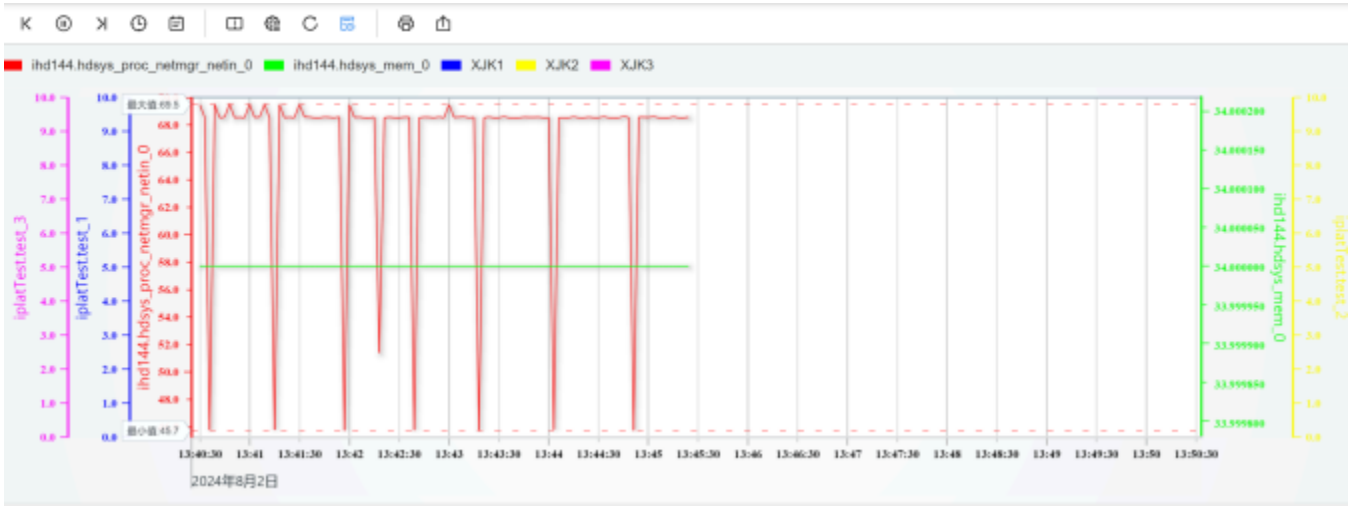
1. 如果当前是实时状态，会切换到暂停状态

之前

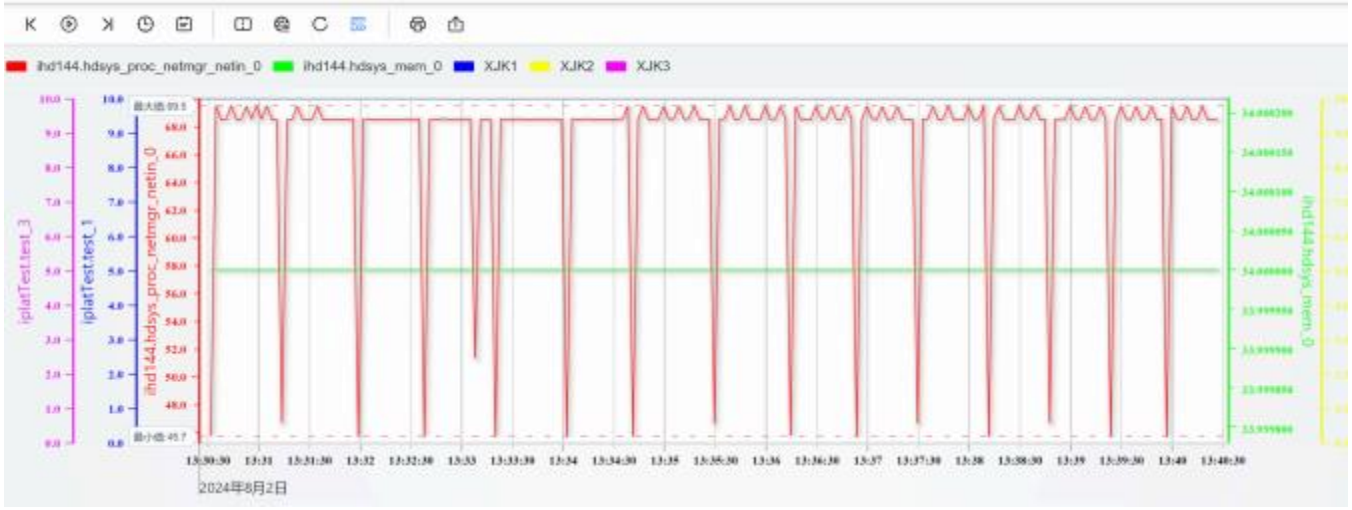
之后

2. 点击之后，x轴的新的结束之间为之前的开始时间，新的开始时间为新的结束时间减去当前的时间翻页。

例如当前翻页时间为10分钟，当前开始时间为13:40，结束时间为13:50

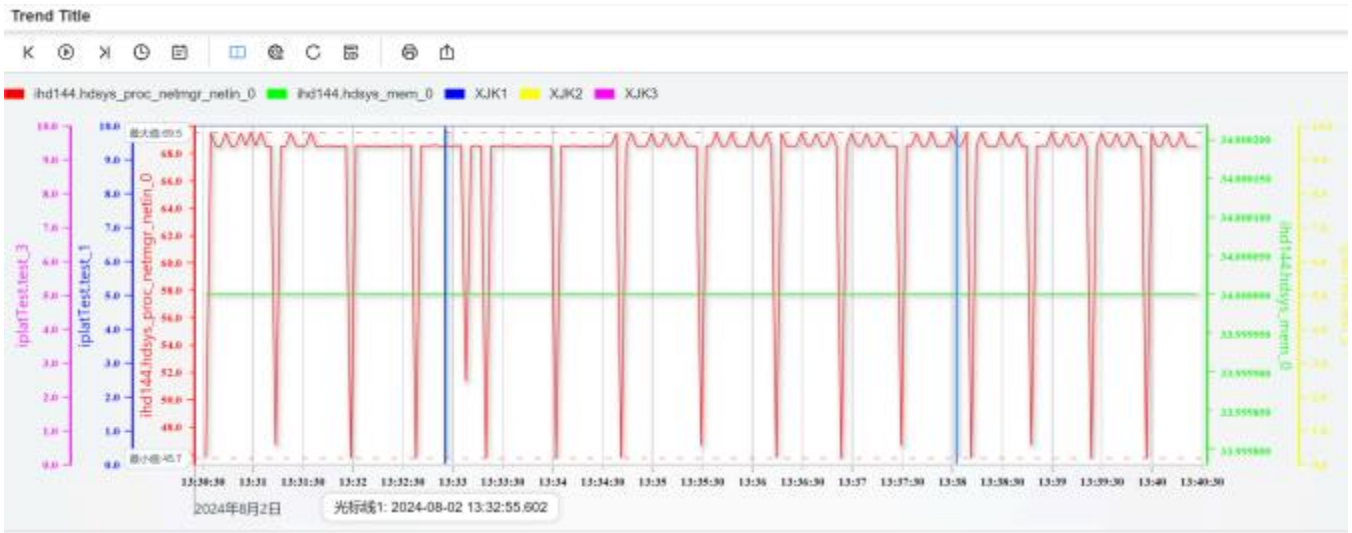


点击向前翻页之后，新的开始时间为13:30，结束时间为13:40

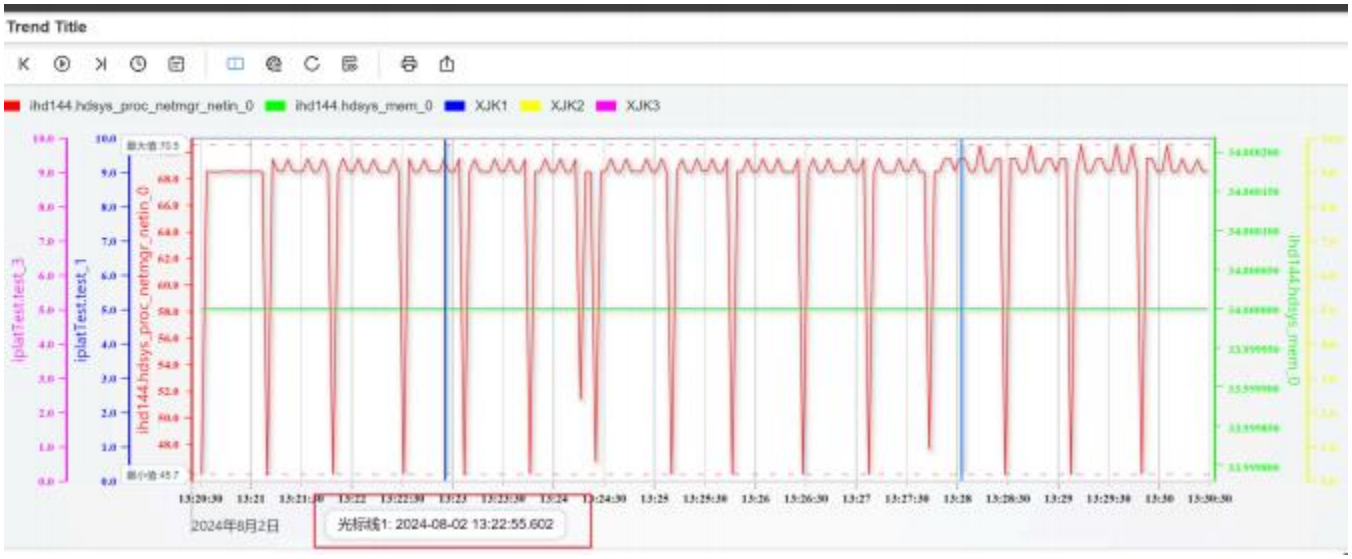


3. 在有标线的时候，向前翻页之后，标线依然存在，但是标线对应的时间也会随之调整

例如，此时光标线1对应的时间为13:32，翻页时间为10分钟



点击向前翻页之后，光标线依然存在，对应的时间变更为13:22



切换至暂停（或实时）：

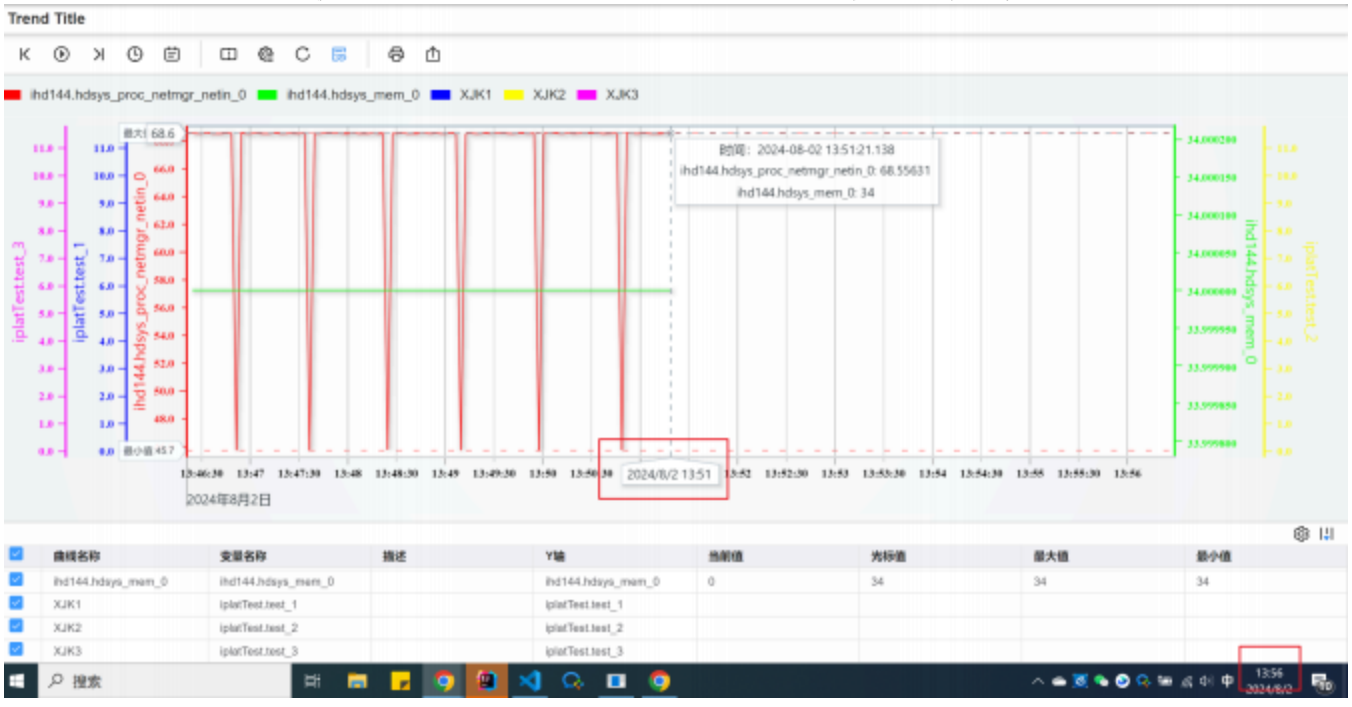
当前为实时状态时，点击后会暂停数据更新，并将曲线设置为历史状态；当前为历史状态时，点击后会切换至实时趋势，并将曲线设置为实时状态。

进入页面之后，默认是实时状态，此时该按钮为，提示为【切换至暂停】

场景效果如下：

1. 实时状态下点击，会变成暂停状态，此时按钮会变化为，提示为【切换至实时】。页面上已经绘制的曲线保持不变，随着时间增加不会再绘制新的点

例如13:51切换至暂停状态，时间推移到13:56，但是页面上最后一个点依然是13:51对应的点



2. 暂停状态下点击，会变成实时状态，此时按钮会变化为，提示为【切换至暂停】。页面上的已经绘制的曲线会重新拉去数据再绘制一遍，随后会随着时间的推移绘制新的点

例如上图的例子在13:56分变回实时状态后，随着时间的推移也会继续绘制新的点

其它影响：

该按钮的状态会受到其他页面操作的影响，也会影响其他操作

具体场景有：

1. 实时状态下，点击标线会变成暂停状态
2. 实时状态下，点击向前翻页会变成暂停状态
3. 实时状态下，时间选择弹窗中，选择某个时间会变成暂停状态
4. 实时状态下，点击向后翻页，页面没有相应；暂停状态下，点击向后翻页，如果翻页后的x轴结束时间早于当前时间，保持暂停状态，如果翻页后的x轴的结束时间晚于当前时间，变为实时状态
5. 暂停状态有标线的情况下，切换到实时状态后，标线会消失

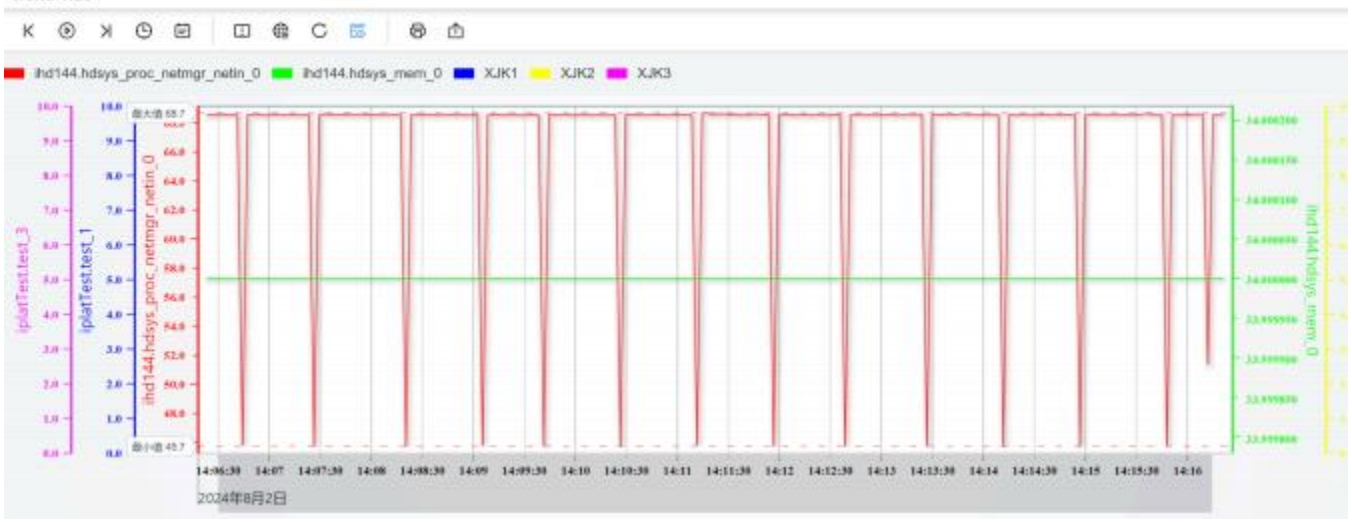
向后翻页：

场景效果如下：

1. 如果当前是实时状态，点击之后没有响应
2. 暂停状态下点击之后，x轴的新的开始时间为之前的开始时间，新的结束时间为新的开始时间加上当前的翻页时间长度

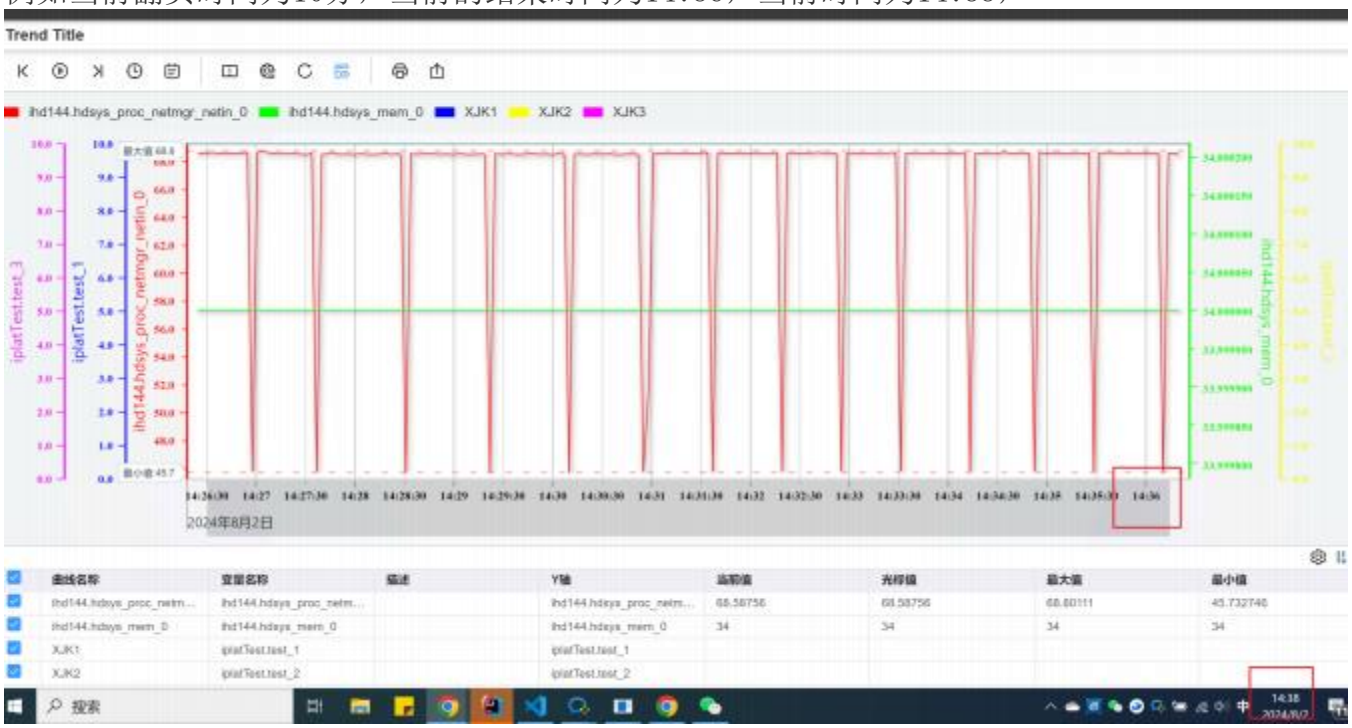
如果新的结束时间早于当前时间，那么例如当前翻页时间为10分钟，当前开始时间为13:56，结束时间为14:06

点击向后翻页之后，新的开始时间为14:06，结束时间为14:16



如果新的结束时间晚于当前时间，会切换回实时状态

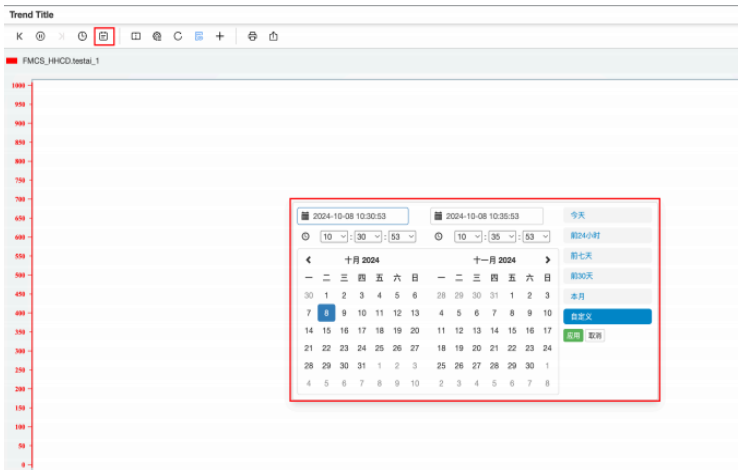
例如当前翻页时间为10分，当前的结束时间为14:36，当前时间为14:38，



点击向后翻页之后，结束时间应该变更为14:46，晚于当前时间，所以会切回实时状态

3. 在有标线的时候，向后翻页之后，标线依然存在，但是标线对应的时间也会随之调整，与向前翻页相同。

时间选择



用户可选择任意时间，点击应用；或者可以直接选择右侧列出的【今天、前24小时】等预设的时间间隔(可以在趋势配置中更改)。

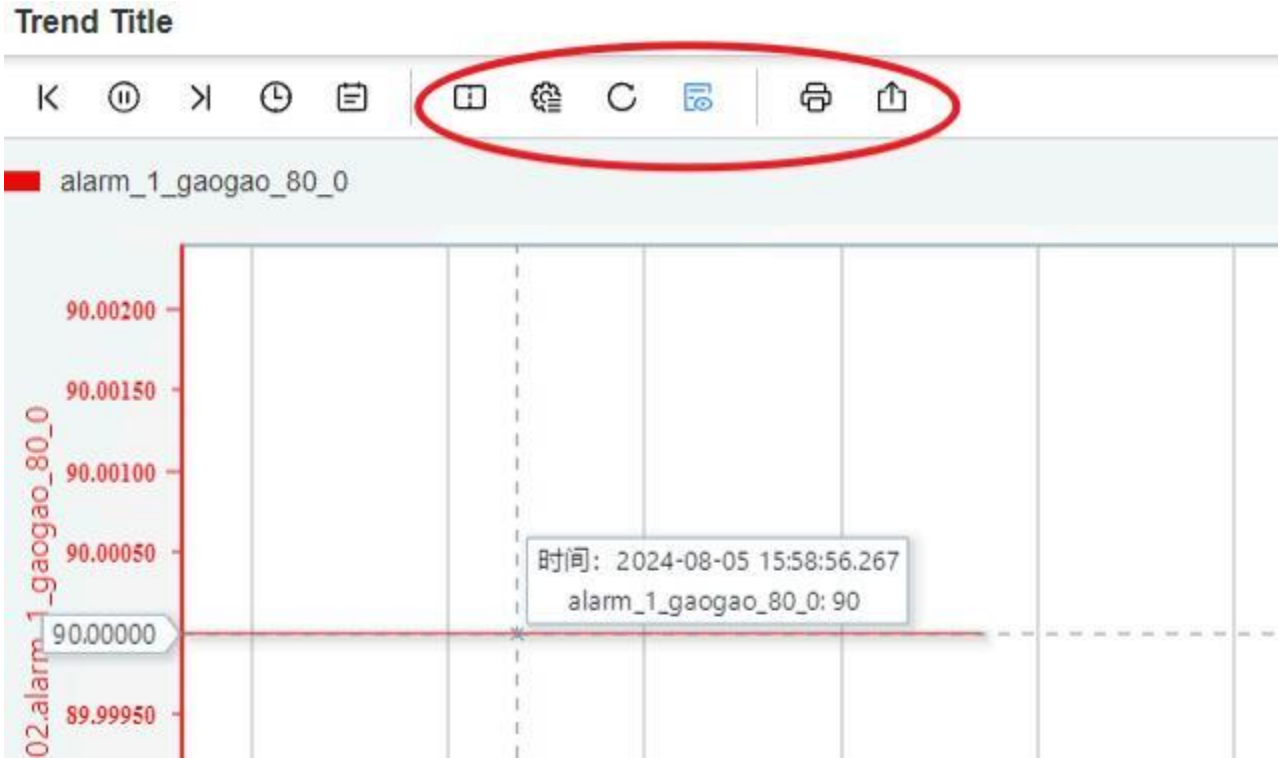
选择时间之后，

1. 时间面板会消失
2. x轴展示为显示的时间
3. 如果是实时状态会切换为暂停状态

注意：使用时间选择的方法调整x轴的范围，并不会影响翻页时间长度

历史数据时间是服务器时间,假如客户端有时差,可能导致趋势曲线显示不全甚至显示不出来

其它工具



显隐藏标线：
可以显示及隐藏竖标线，标线可以左右拖动

1. 按钮状态切换

（1）标线不显示，按钮为黑色。当标线显示时，按钮则会高亮。刚打开趋势时，默认标线不显示

（2）【显示标线】与【实时趋势】互斥。即趋势切换至实时状态时，标线会自动隐藏。在趋势处于实时状态时，打开标线功能，趋势会自动暂停

2. 标线功能

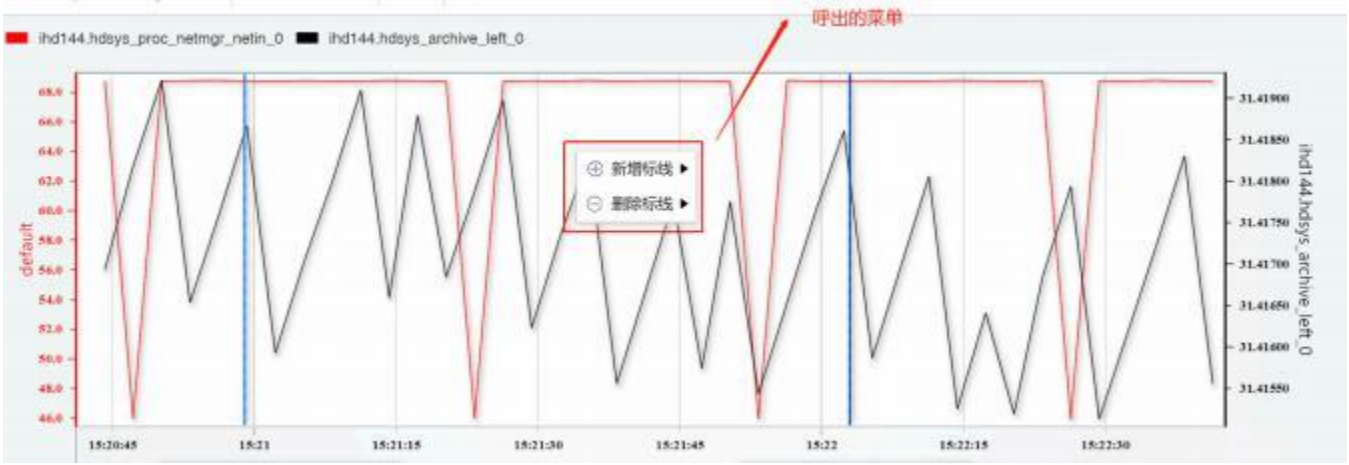
（1）点击显示按钮后，按钮高亮，趋势图表中会出现两条竖直的标线。这里称它们为光标线。如下图所示

光标线的下方会对应出现一个时间轴标签，负责显示光标线对应的时间点（即x轴数值）。同时，光标线的时间、以及图中每条曲线在光标线位置的数值会被记录在表格内。用户可以拖动这两条光标线，来查看不同时刻下，所有曲线的数值。如下图所示。

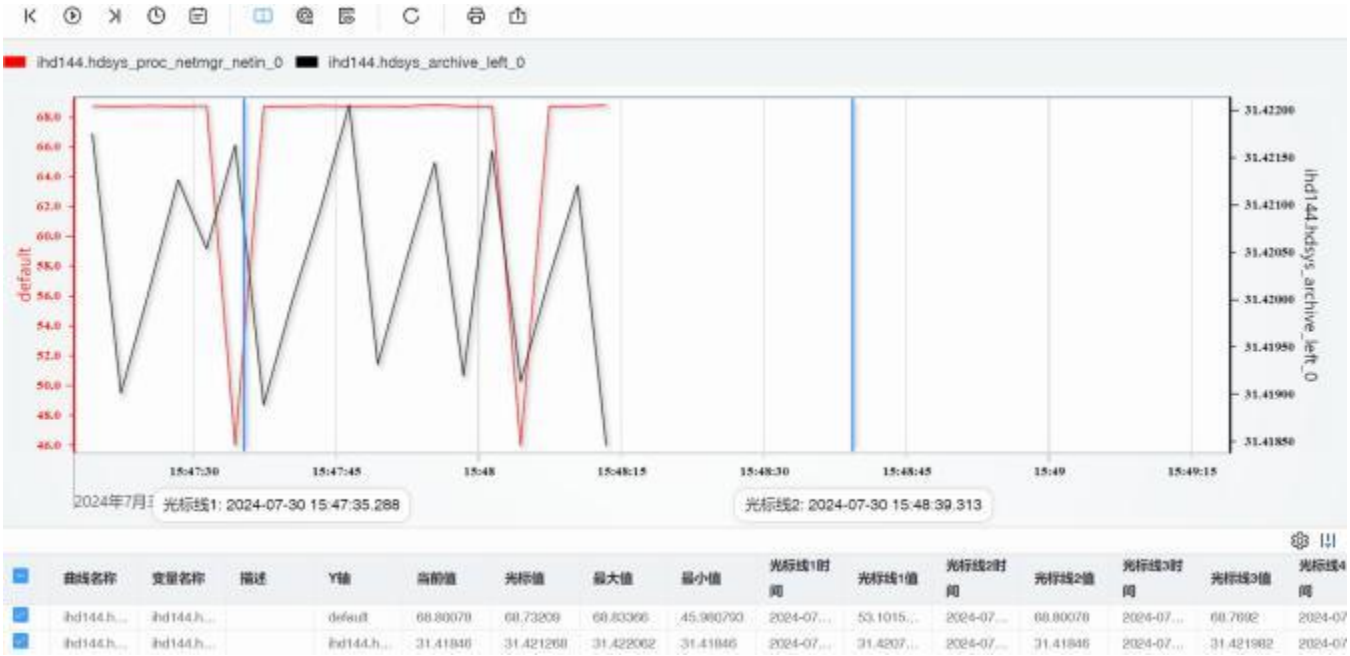
需要注意的时，只有用户的拖动行为可以改变光标线的位置，光标线不会随着放大缩小，翻页，时间跨度改变等操作，而发生位置上的变化

（2）如果，用户对某一时刻的数据感兴趣，需要记录或对比研究。那么标线还提供了另一种功能，允许用户在目标位置打上一种固定的标线，这里称其为固定标线。这种标线也会记录对应时刻的时间与曲线数据，而因为标线固定，该标线的时间与曲线数据不会发生变化

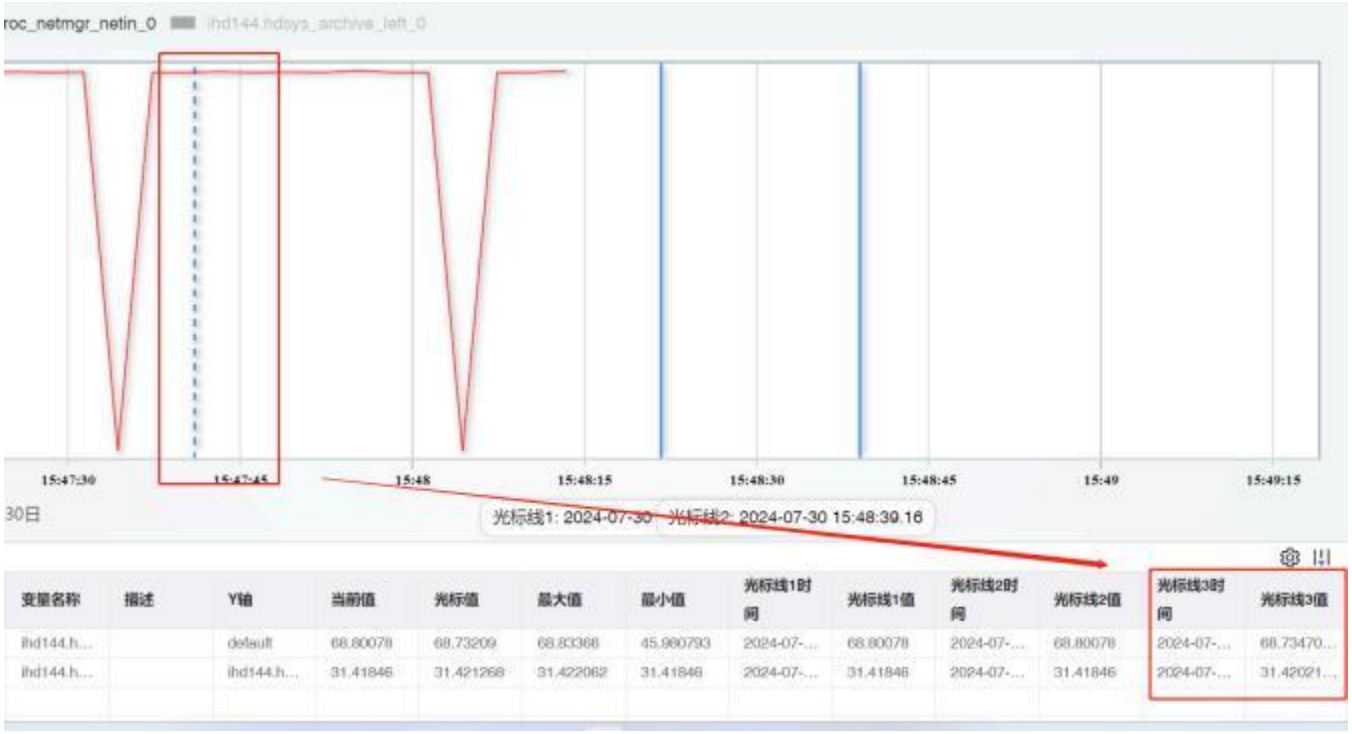
添加固定标线操作方法为：右键图表界面，呼出标线的功能菜单。可以从下图中看出，菜单具有两个功能：新增标线与删除标线。



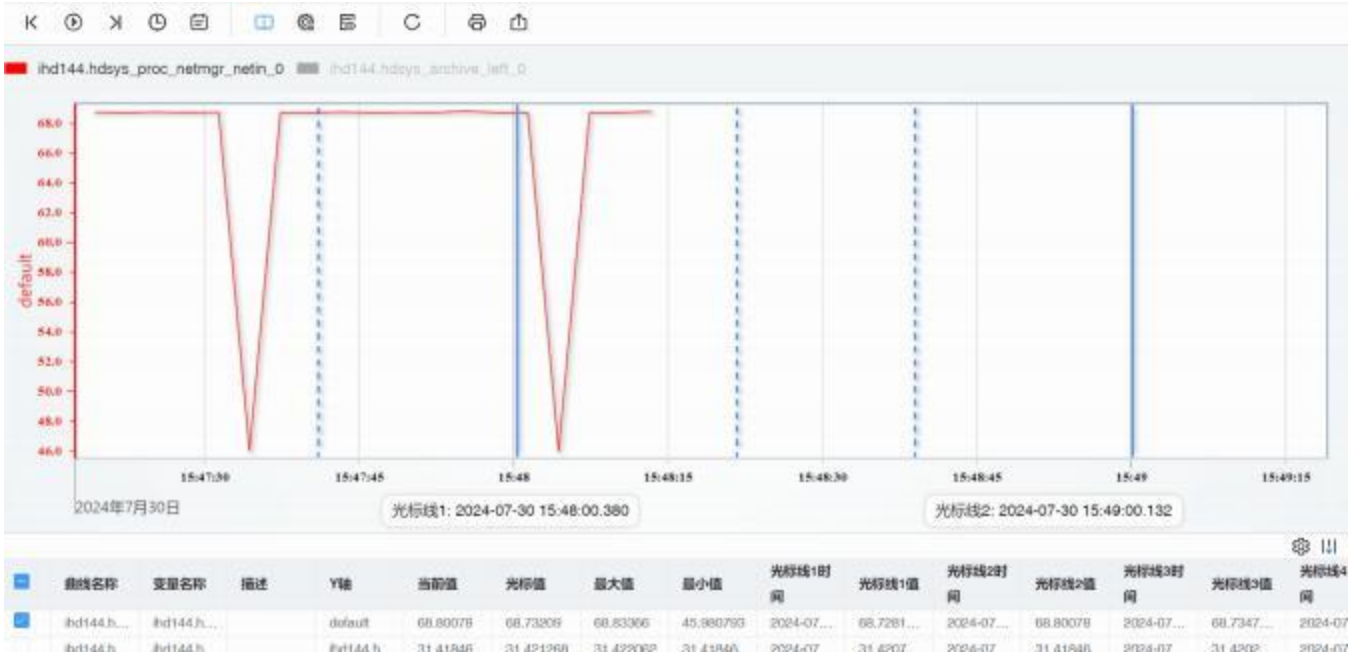
鼠标移动至【新增标线】的选项上，二级菜单就会自动弹出。【新增标线】的二级菜单固定提供两种选项：【在光标1处新增】以及【在光标2处新增】。光标线1，2即为前文提到的用户可以拖动的标线。点击二者中的其中一个选项。那么相应光标线的位置会出现一个虚的竖线。这就是固定标线。



鼠标悬浮在固定标线上，就会出现一个标签提示，内容为固定标线的名称及其对应的时刻。于此同时，表格中也会新增两列，展示曲线在固定标线位置的值与时间。如下图所示



若要删除固定标线，则只需要在右键打开的菜单中，找到【删除标线】项。鼠标悬浮在上面，就会呼出一个二级菜单。二级菜单会展示所有固定标线。点击相应的固定标线即可删除



（3）若关闭标线功能，所有的光标线、固定标线、以及标签，都会消失。表格中对应的列也会消失。

区域缩放重置：
鼠标滑选曲线可以对曲线局部放大，点击该按钮会恢复到默认状态

高亮：

进入页面时，【区域缩放重置】按钮默认不是高亮。

当用户在x轴或者y轴区域缩放，或者在整个图表区域缩放后，【区域缩放重置】按钮会变成高亮的状态。

点击启用：

当【区域缩放重置】按钮高亮时，点击后，会回退为未缩放时的状态，并且【区域缩放重置】不再是高亮

显隐ToolTip：
启用后，鼠标悬浮在曲线上会显示曲线时间、点值

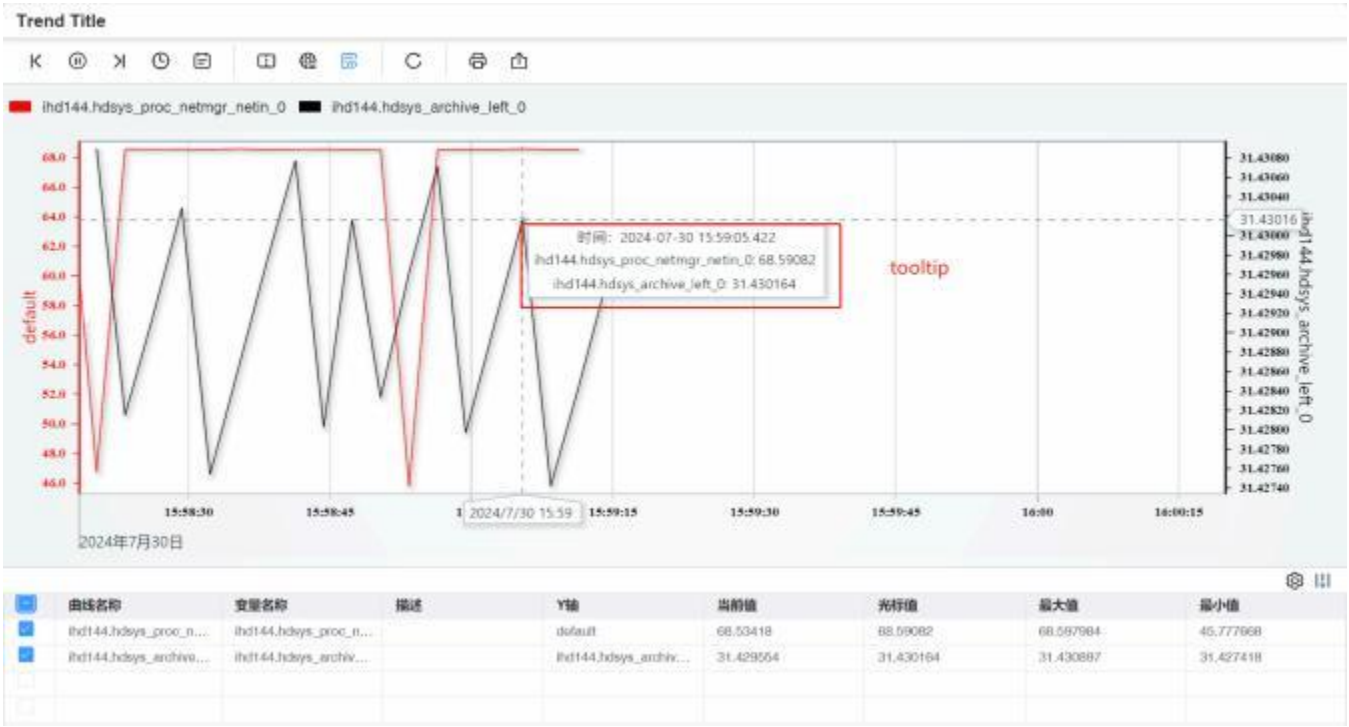
1.按钮状态切换

（1）tooltip不显示，按钮为黑色。当tooltip显示时，按钮则会高亮。刚打开趋势时，默认tooltip为打开状态，即高亮

（2）【显示标线】与【显示tooltip】互斥。即显示tooltp的状态下，点击【显示标线】，tooltp会自动关闭；在显示标线的情况下，点击【显示tooltip】，则会弹出提示来禁止这一操作。

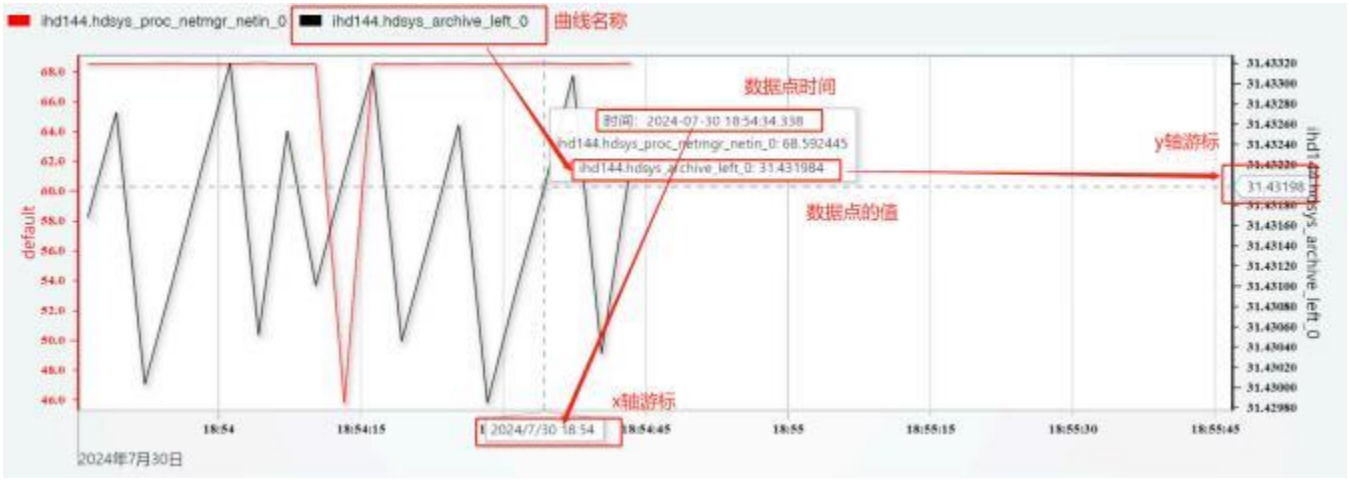
2.tooltip功能

tooltip功能展示了鼠标所处时刻所有曲线的数值
（1）在打开tooltip的情况下，鼠标移动到曲线上，就会弹出tooltip提示。



（2）一般来说，鼠标在曲线上时，或者在曲线附近时，tooltip会自动吸附到最近的一个数据点并进行数据展示。

（3）tooltip展示的内容

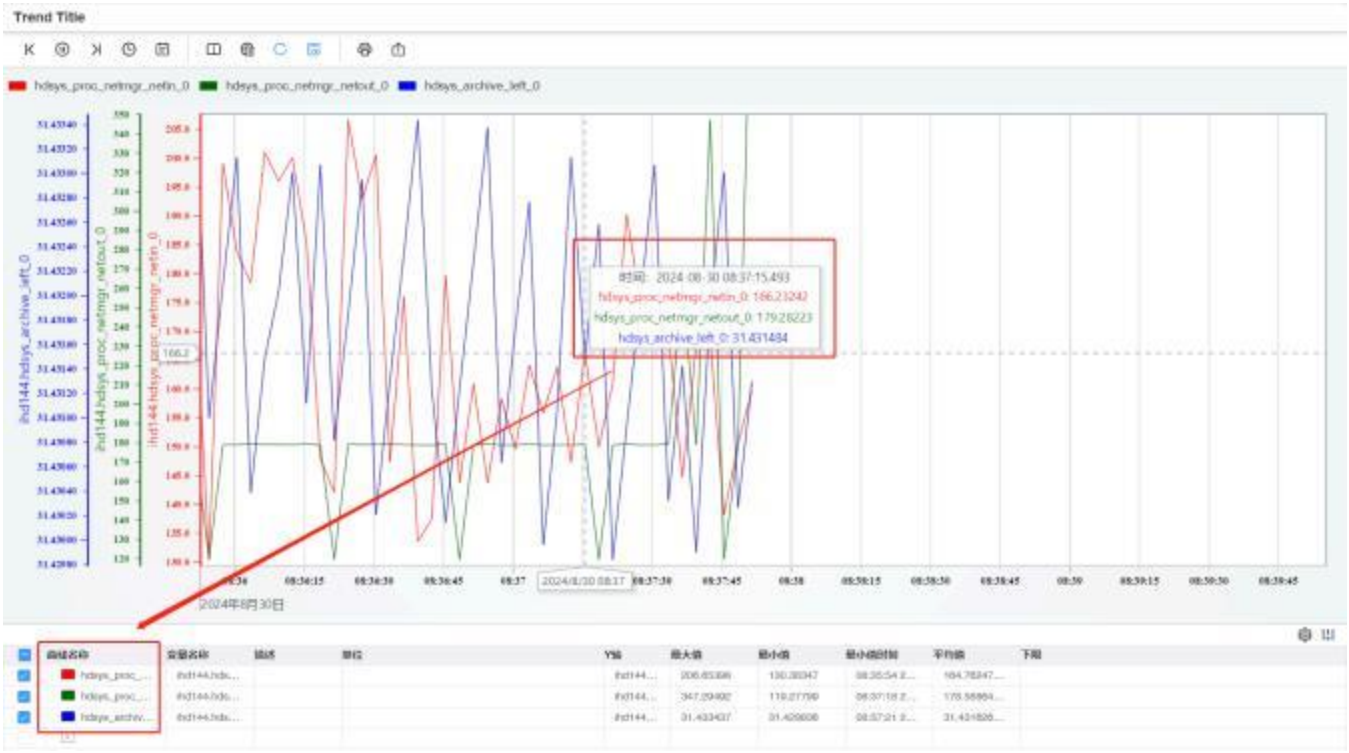


以上图为例，tooltip展示了当前数据点的时间以及数值信息。同时，x轴，以及与数据点对应的y轴上也会出现两个游标，展示数据点的时间与值。

同时，tooltip也会展示相同时刻下，其他曲线的值。

但同一时刻，并不是所有数据源都会有值。往往会出现一个数据源的点在这个时刻有值，而其他数据源都没有这个时刻的数据点的情况。故而目前（截止24/8/30）的做法是：寻找其他数据源最邻近该时间点的数据进行展示

新版本中（24/8/30），tooltip展示的文字颜色会与对应的曲线颜色相同。



强制原始值查询 :

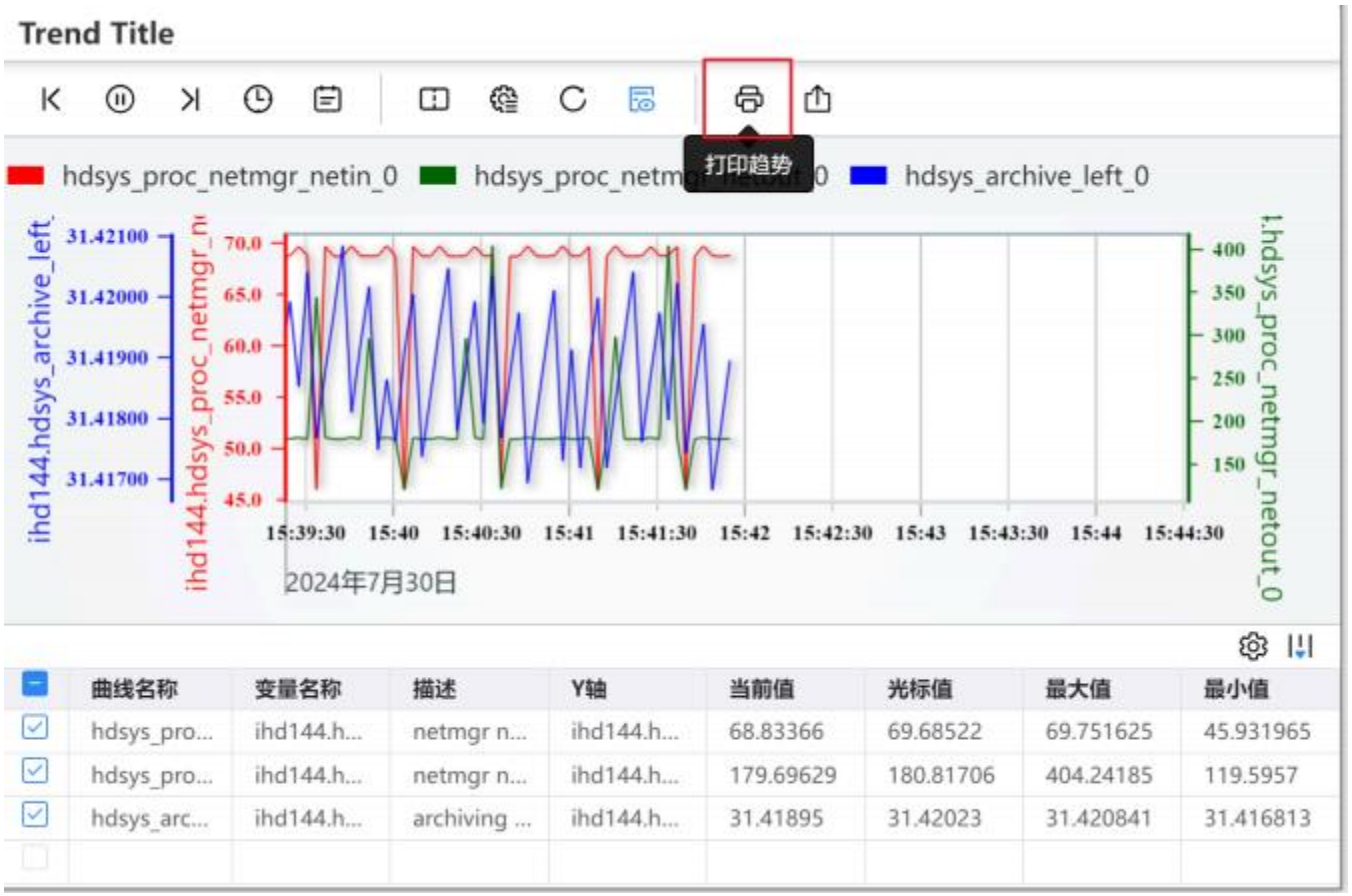
默认的查询状态会根据查询曲线的数据量的大小和时间长度决定是采用插值和原始值，具体为查询时间长度超过两周或者单条曲线查询范围内的点数超过10w（两者满足其一即可）时转换为插值点；其它情况下保持原始值。

在Toolbar工具栏上提供一个新的按钮，默认为黑色，hover提示为【强制原始值查询】；点击之后，当前时间范围内对曲线重新进行一次查询，且查询结果返回原始值数据。

强制原始值查询只会对（1）数据类型为插值的曲线（2）存在查询失败的数据片段的曲线，进行强制原始值查询

右上角数据表格中增加值类型列，用原始值和插值表明这次查询服务端返回的是原始值还是插值。

打印趋势：点击打印当前趋势图。用户点击之后，默认是横版、A4纸张进行打印。页面上的元素展示清
晰。



曲线名称	变量名称	描述	Y轴	当前值	光标值	最大值	最小值
hdsys_pro...	ihd144.h...	netmgr n...	ihd144.h...	68.83366	69.68522	69.751625	45.931965
hdsys_pro...	ihd144.h...	netmgr n...	ihd144.h...	179.69629	180.81706	404.24185	119.5957
hdsys_arc...	ihd144.h...	archiving ...	ihd144.h...	31.41895	31.42023	31.420841	31.416813

导出趋势数据:
点击导出趋势的点值与时间，Excel表格

用户点击导出之后，会生成一个excel表格，表格名称为趋势数据。
1.导出过程中会出现loading

2.最多导出80万条数据，超过这个范围会提示用户，是否继续导出？用户选择取消会停止导

出；用户选择继续导出，那么表格中会按照曲线数量将80万的份额均分给各条曲线，然后继续导出

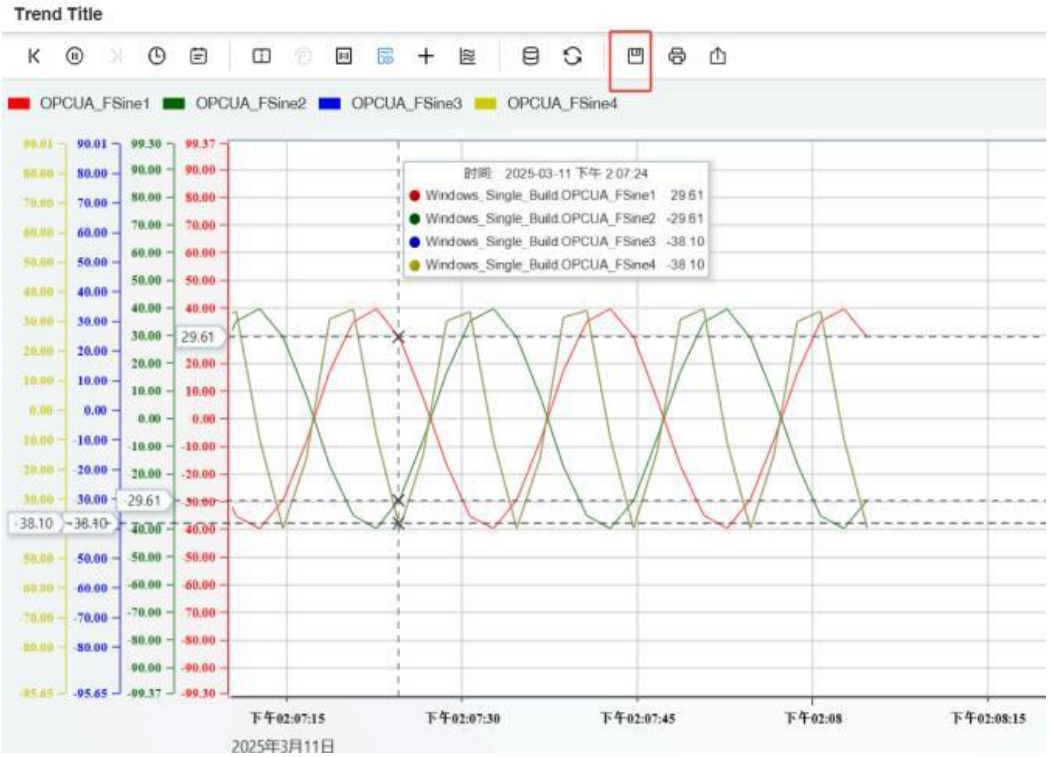
3. 表格内部的内容依次是【曲线1时间】【曲线1值】【曲线2时间】【曲线2值】...

	A	B	C	
1	hdsys_archive_used_0		hdsys_netin_0	
2	archiving disk usage		system net in	
3	时间	点值	时间	点值
4	2024-12-11 16:58:12	50.22675	2024-12-11 16:58:12	12.8776045
5	2024-12-11 16:58:15	50.22565	2024-12-11 16:58:15	12.759114
6	2024-12-11 16:58:18	50.22846	2024-12-11 16:58:18	13.817383
7	2024-12-11 16:58:21	50.22736	2024-12-11 16:58:21	13.785156
8	2024-12-11 16:58:24	50.22565	2024-12-11 16:58:24	13.489583
9	2024-12-11 16:58:27	50.22742	2024-12-11 16:58:27	13.456706
10	2024-12-11 16:58:30	50.225895	2024-12-11 16:58:30	13.104492
11	2024-12-11 16:58:33	50.22864	2024-12-11 16:58:33	13.286133
12	2024-12-11 16:58:36	50.227604	2024-12-11 16:58:36	12.483398
13	2024-12-11 16:58:39	50.226566	2024-12-11 16:58:39	12.6875
14	2024-12-11 16:58:42	50.22553	2024-12-11 16:58:42	13.409505

导出的范围为当前视图内的数据。导出的日期格式受到全局日期格式配置的控制；导出的数据始终是原始值，不受小数点的控制

保存趋势配置

点击后，会保存当前文本趋势中的曲线配置信息（P9发布）



1. 目前保存的配置

•曲线配置界面中的”启用小数“与”小数位数“列

•翻页时间长度

2. 配置的存储位置

•IndexDB

•localStorage（备用，只会在检测到浏览器不支持indexDB时使用）

3. 配置的存储限制

•曲线配置的条数上限为40条

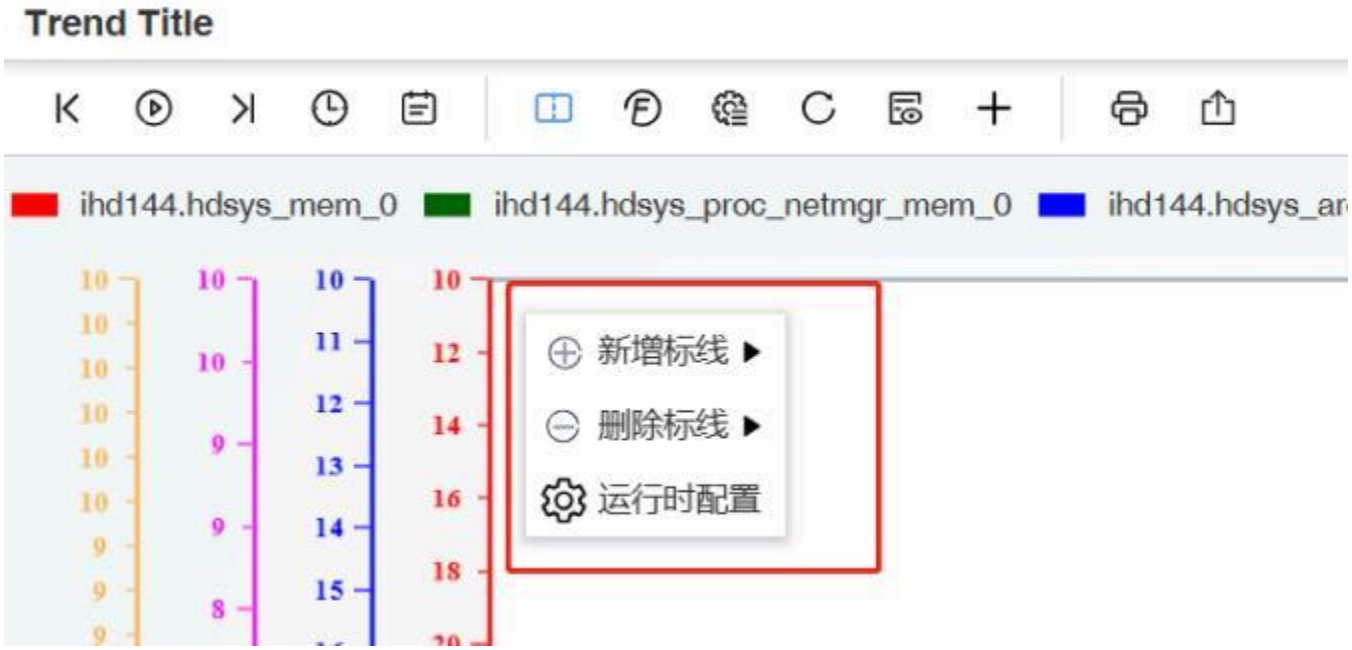
•若保存配置后，曲线条数超出上限，则只存储最新存入的40条曲线

•若同一条曲线，在已经保存了配置的情况下，再次保存了配置。则存储的该曲线的旧配置会被删除，并存入新的配置。（会被视为一个新配置，删除的优先级会被延后

•存储时的键名，固定为“LCTextTrend”，所有文本趋势共用一个键名进行存储（如下图）

趋势Chart

趋势图上右键，可以展开的所有运行态配置如下：



新增标线

•鼠标悬浮在“新增标线”上，就可以打开子菜单。子菜单中可以指定在哪根光标线的位置处新增固定标线

新增好标线后，右键菜单中还会多出一项“删除标线”，同样将鼠标悬浮在它上面，就可以选择要删除哪根固定标线

运行时配置

点击运行时配置，弹出如下对话框，对话框中包含多个tab页可供选择

•Y轴小数配置

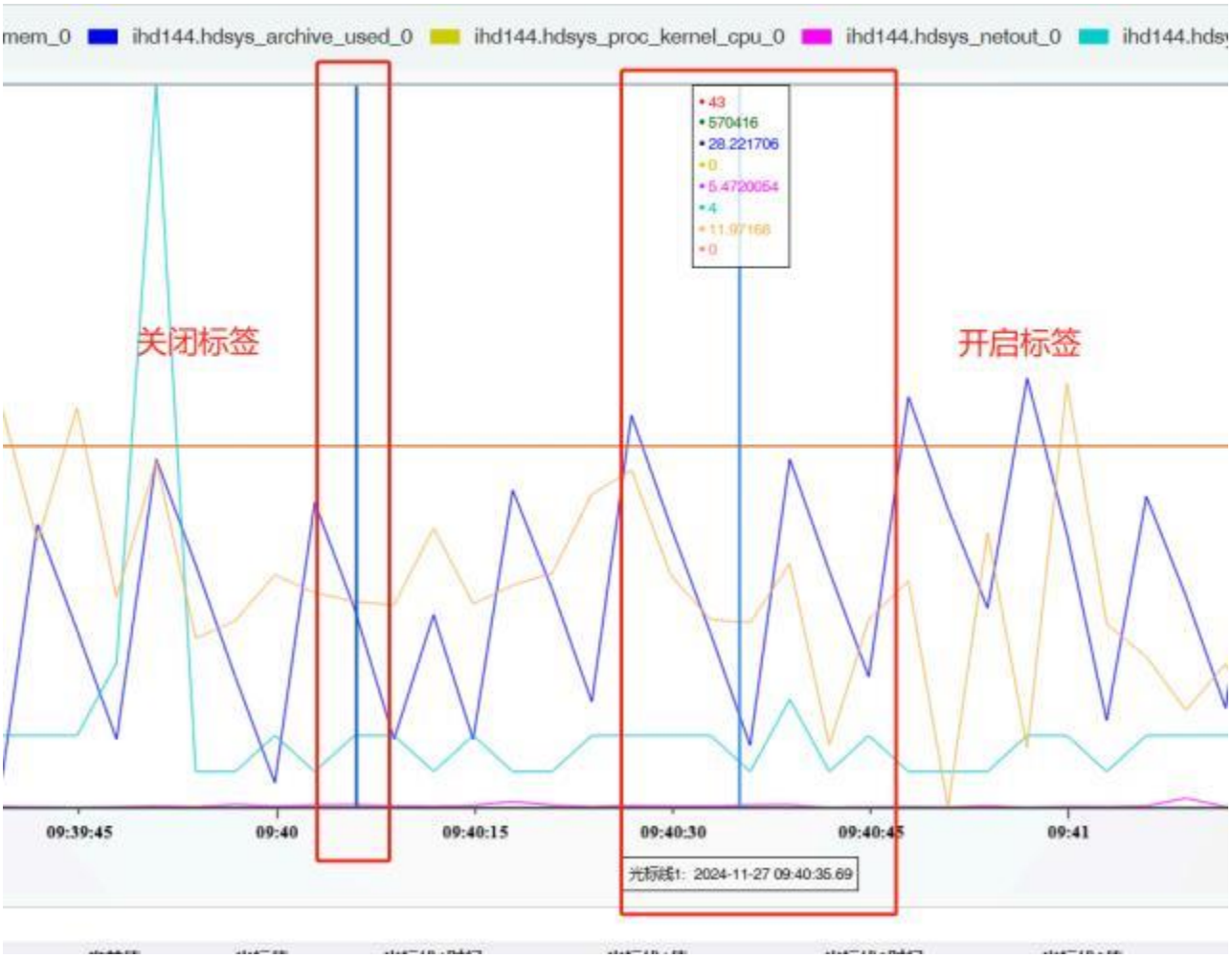
可配置每条曲线对应的Y轴上的小数位数。如果勾选启用，则该配置覆盖编辑态对趋势控件上的整体/单体配置（即具有最高优先级）。

•Y轴量程配置

可配置每条Y轴的量程，配置时，请确保“最小量程”的值小于“最大量程”的值。如果勾选启用，则该配置会覆盖工程量程/自适应量程（即具有最高优先级）

•标线配置

目前仅支持：配置每条标线标签的显示与否。



阈值线配置

显隐曲线最大值、最小值、平均值的阈值线

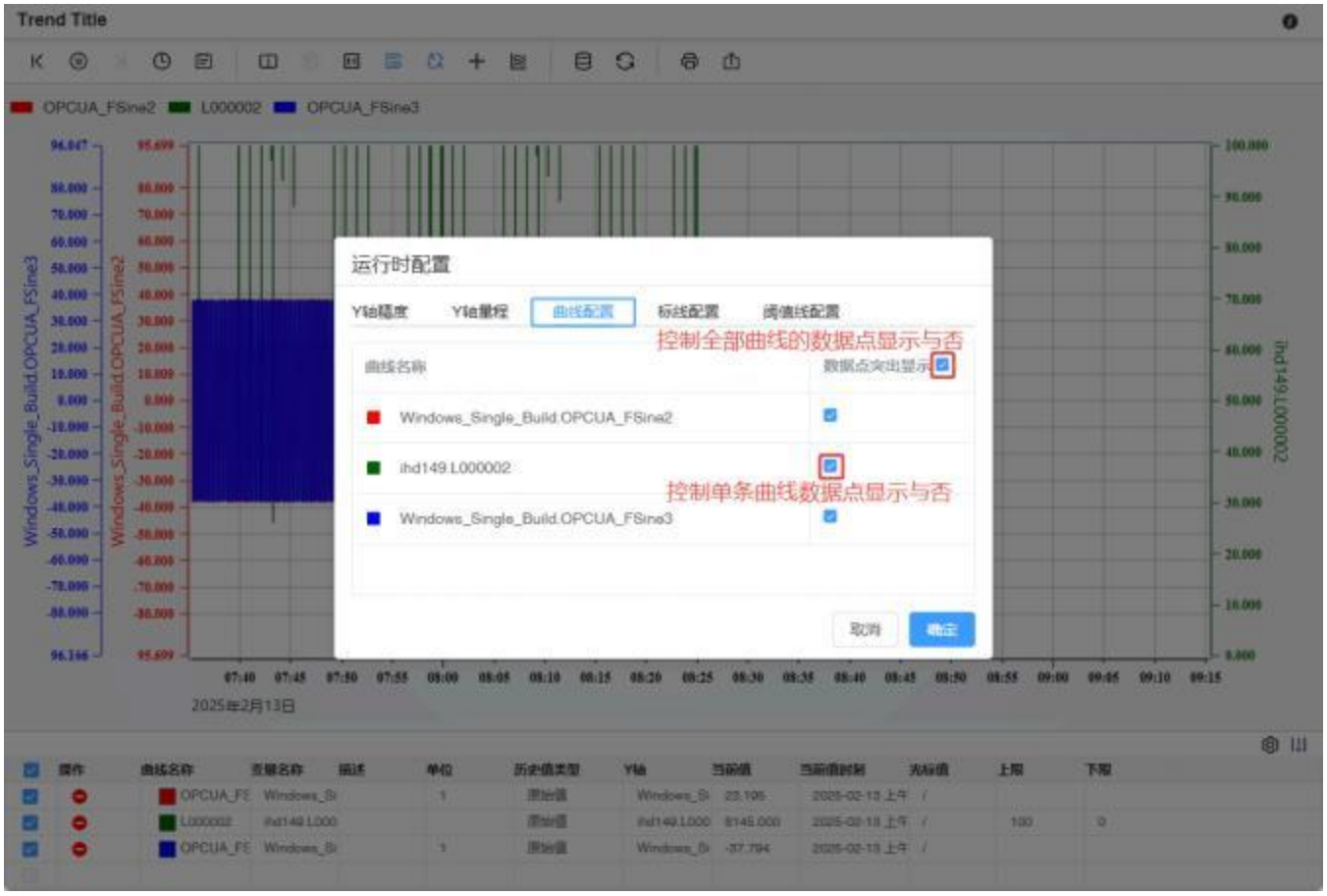
注意勾选之后并不是立刻出现。阈值线出现的前提如下：

1. 曲线本身是显示的状态
2. 表格中有对应的统计值，并且统计值不为空

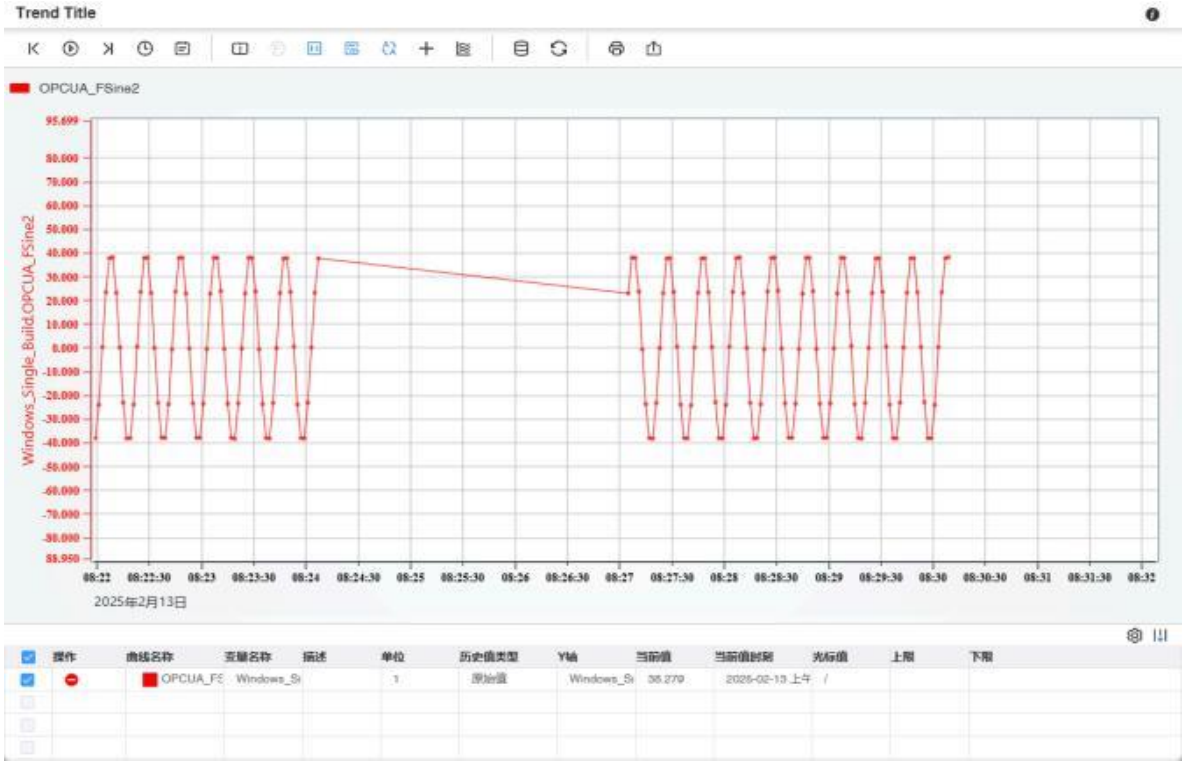
•数据点突出显示

控制曲线中的数据点是否要被标记，以突出显示。

配置界面中，通过勾选框，实现数据点突出显示与否。允许对单条曲线进行配置，也允许对全部曲线进行操作。



数据点突出显示的效果如下图。



当数据点非常密集，或以较小的缩放级别查看时。曲线会自动隐藏数据点（即使你开启了数据点突出显示）。这是LightningChart有意为之，否则曲线的视觉效果会看起来很奇怪。

趋势表格

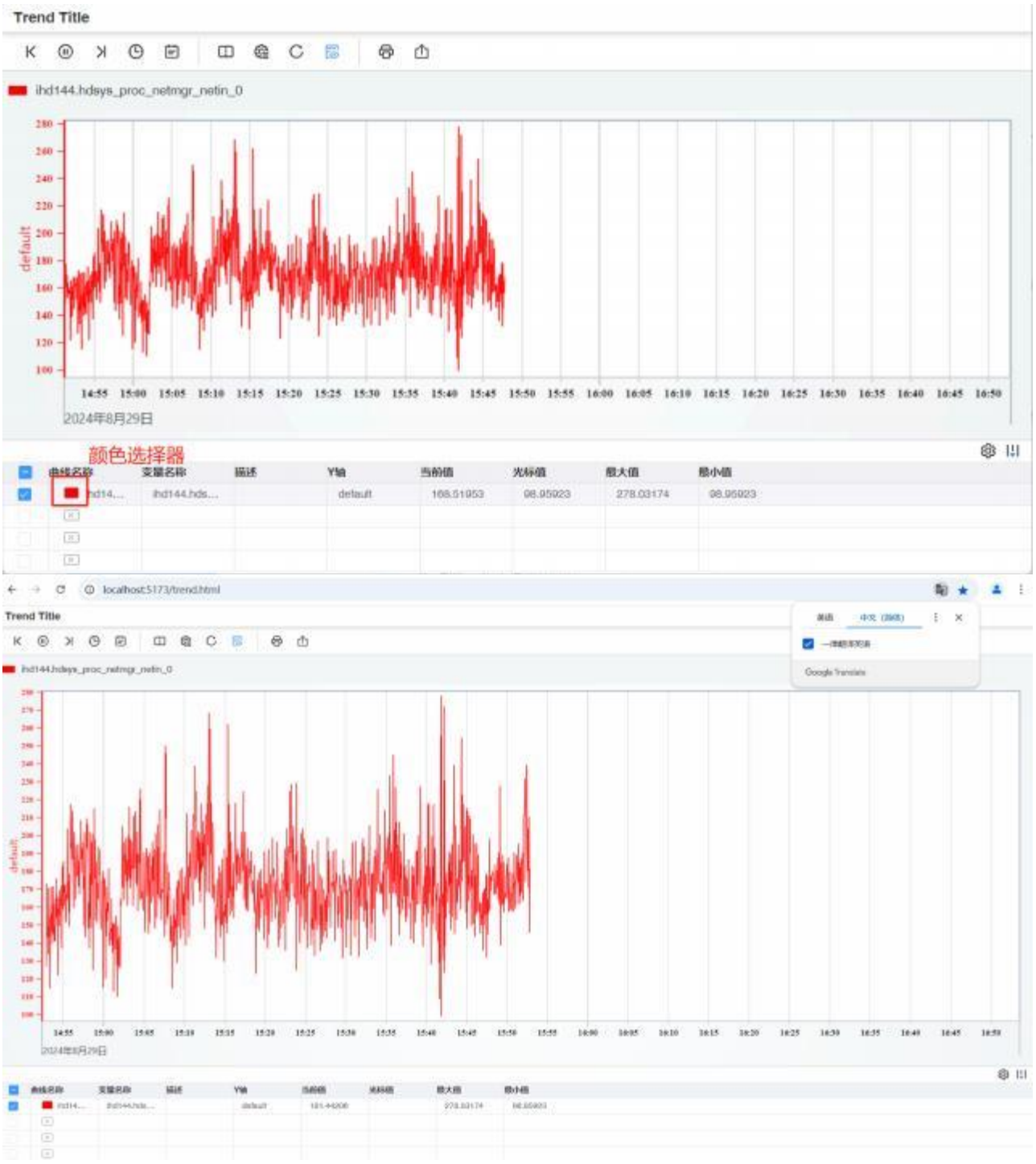
功能按钮

折叠 ：可对表格进行折叠、打开操作列值配置

 ：可选择显示那些列，并对宽度进行调整

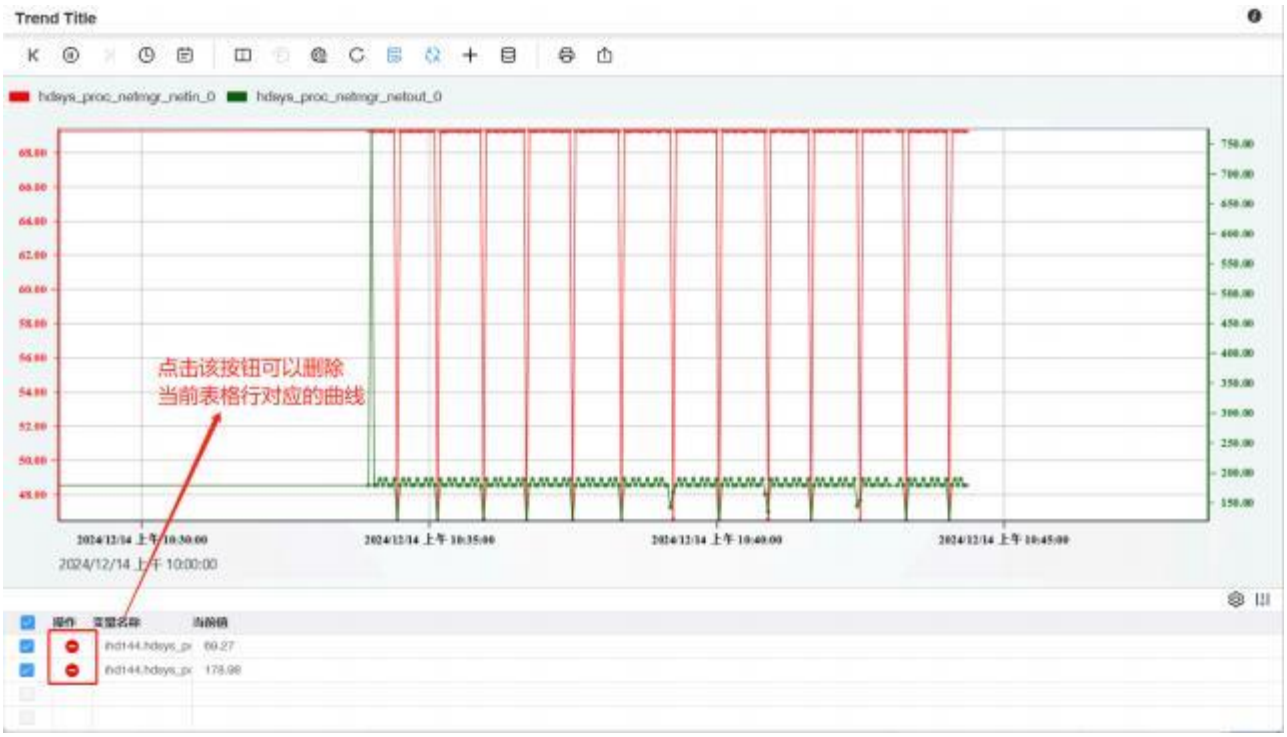
曲线颜色修改

• “曲线名称”列还提供了曲线颜色选择器，通过点击列中的颜色方块，可以选择对应曲线的颜色。



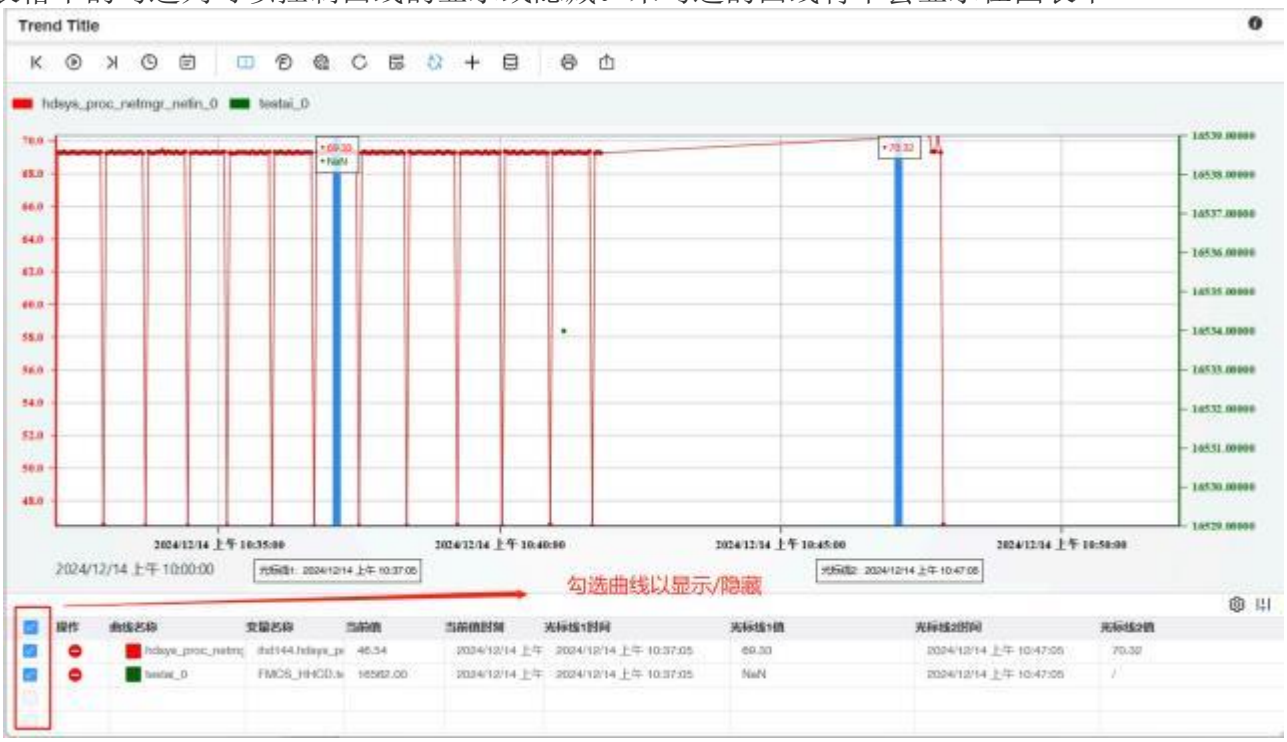
曲线删除

•在编辑态/运行态开启表格中的“操作”列以后，就可以在操作列中，通过点击红色的减号图标，来删除指定的曲线



曲线隐藏

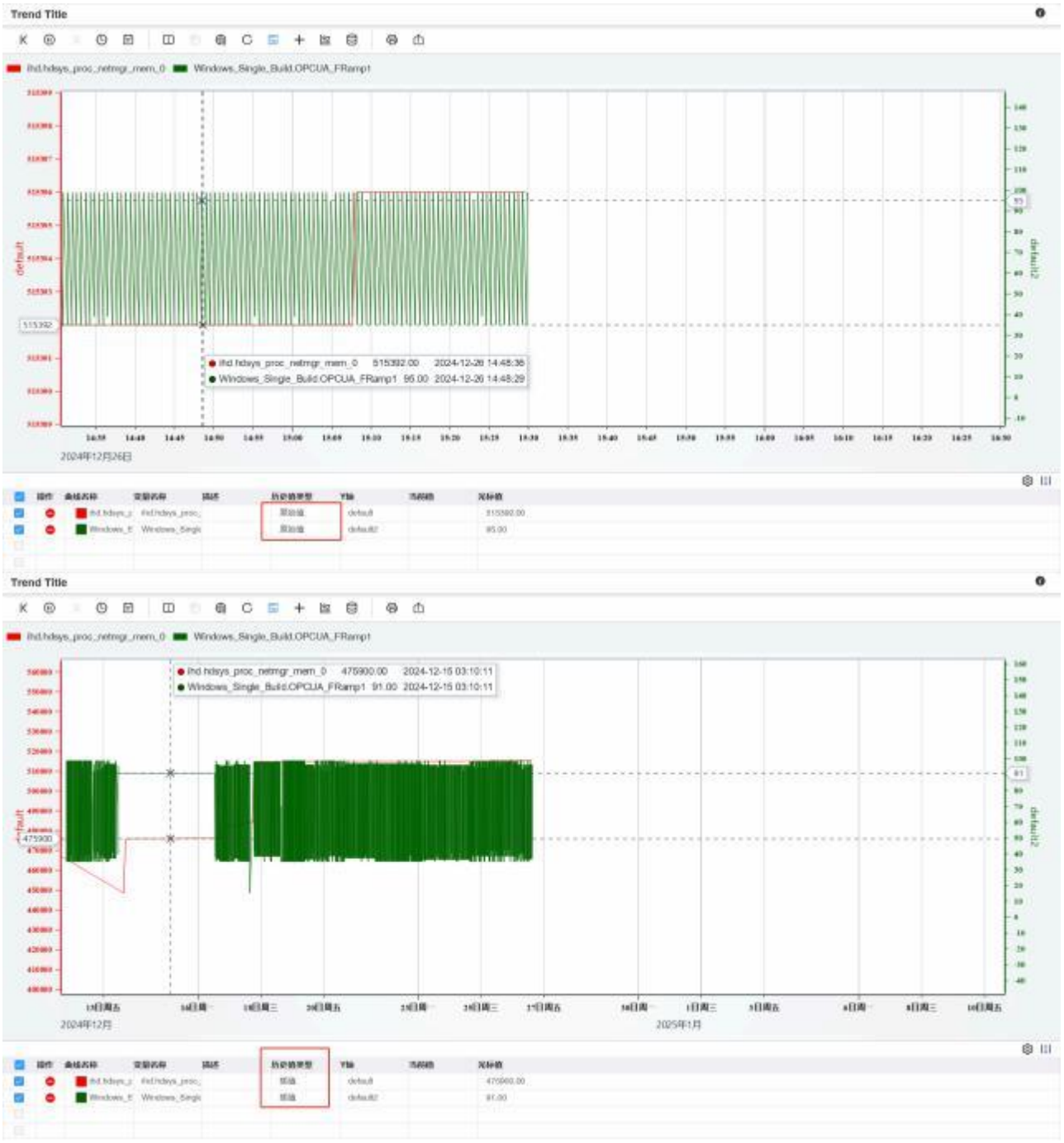
•表格中的勾选列可以控制曲线的显示或隐藏。未勾选的曲线将不会显示在图表中



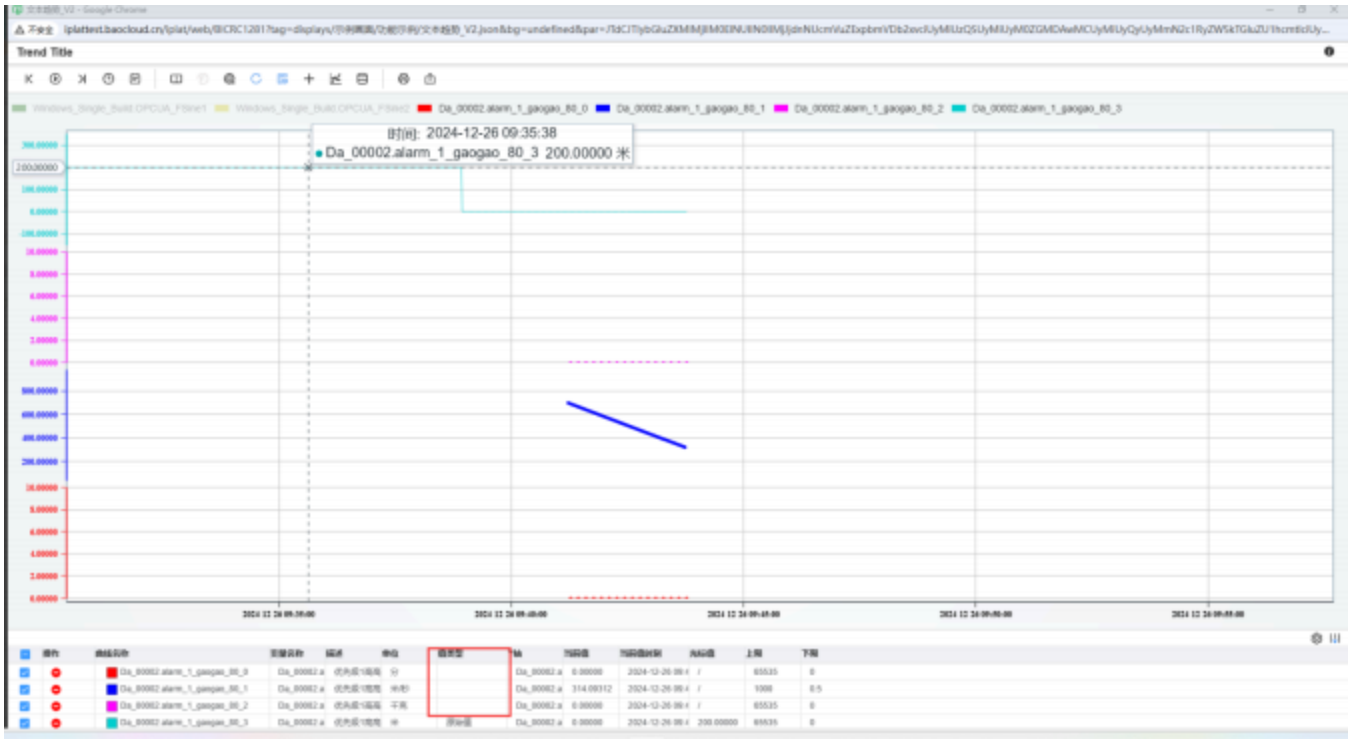
未勾选的曲线，不会在图表中画出，但该曲线的当前值、光标线值、统计值仍然会显示在表格中，并随数据推送（当前值）/光标线的移动（光标线值）/统计值的重新计算（统计值）而变化

历史值类型

•用于标识曲线数据是插值还是原始值



历史值类型列，只会在曲线存在历史值的时候，存在值
如果在实时状态下，遇到下图这种情况（曲线存在实时值，但没有历史值，那么”历史值类型”列依旧为空



只要曲线具有历史值，那么该曲线对应的当前值列就不会为空

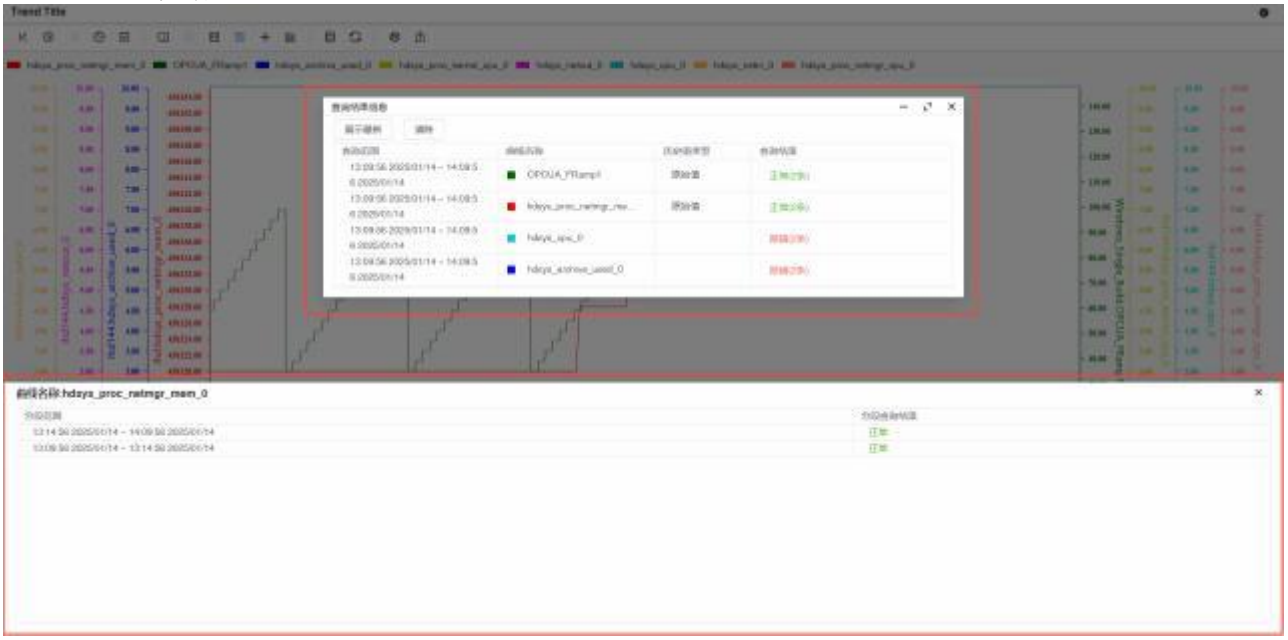
当前值时刻

•在实时状态下,当有新数据推送时当前值会更新，同时当前值时刻也会随之更新

	当前值	当前值时刻
_HHCD.te	54657.00	2024/12/12 09:18:38

查询结果信息表

•用于记录历史数据查询的成功与否



表格对查询结果记录的展示遵项以下原则：

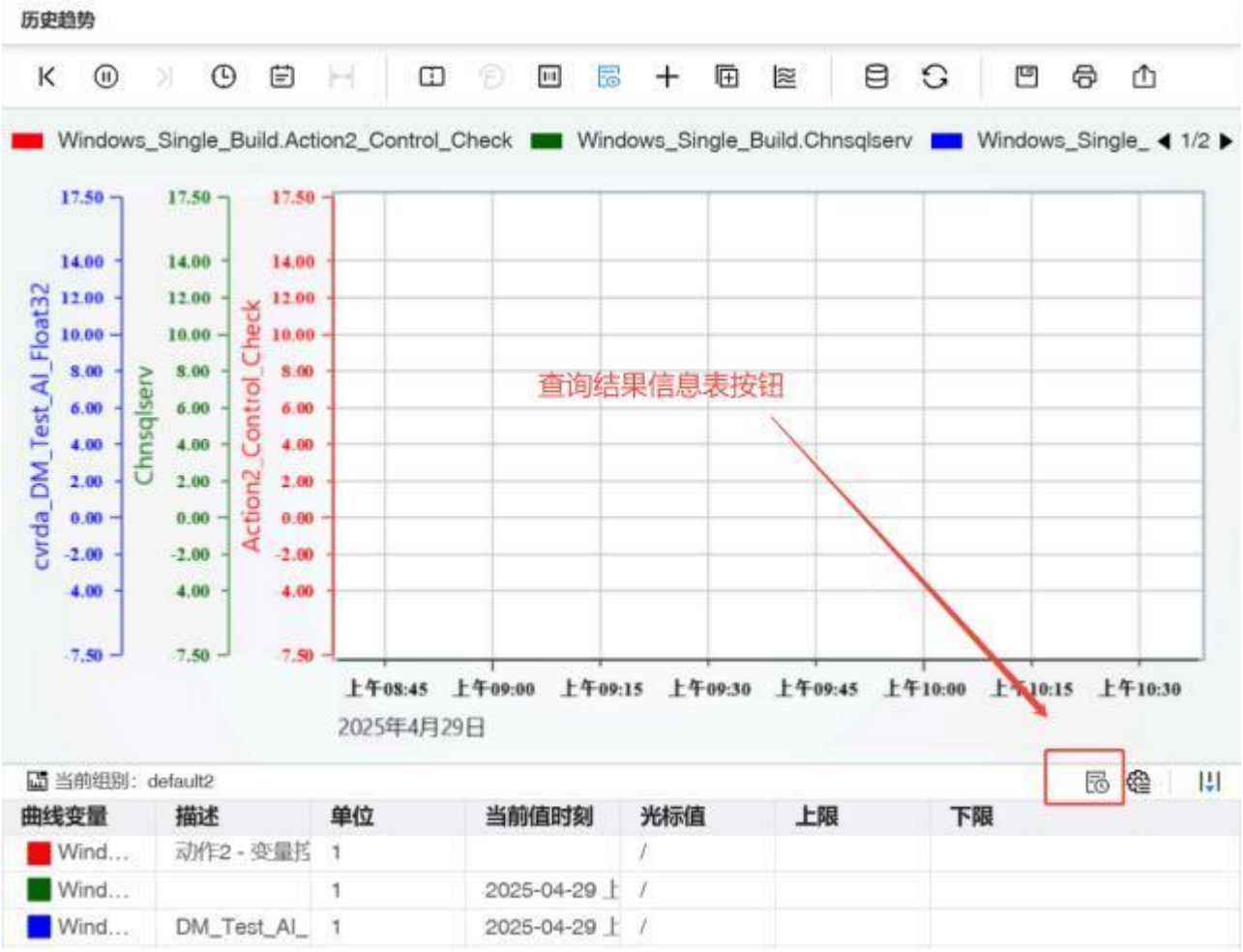
•表格刚打开时，默认展示最新的一批查询结果

•表格最多记录10批查询结果

•每条曲线的查询结果信息最多记录200条

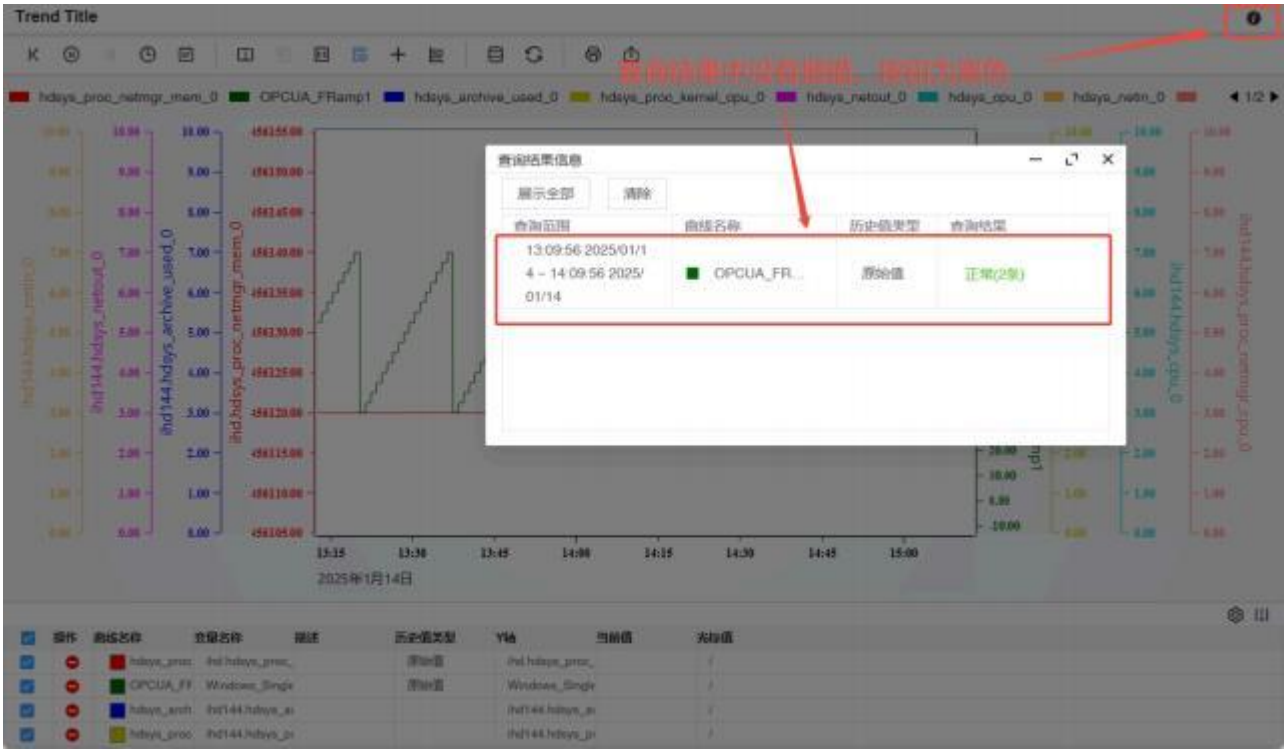
打开方式

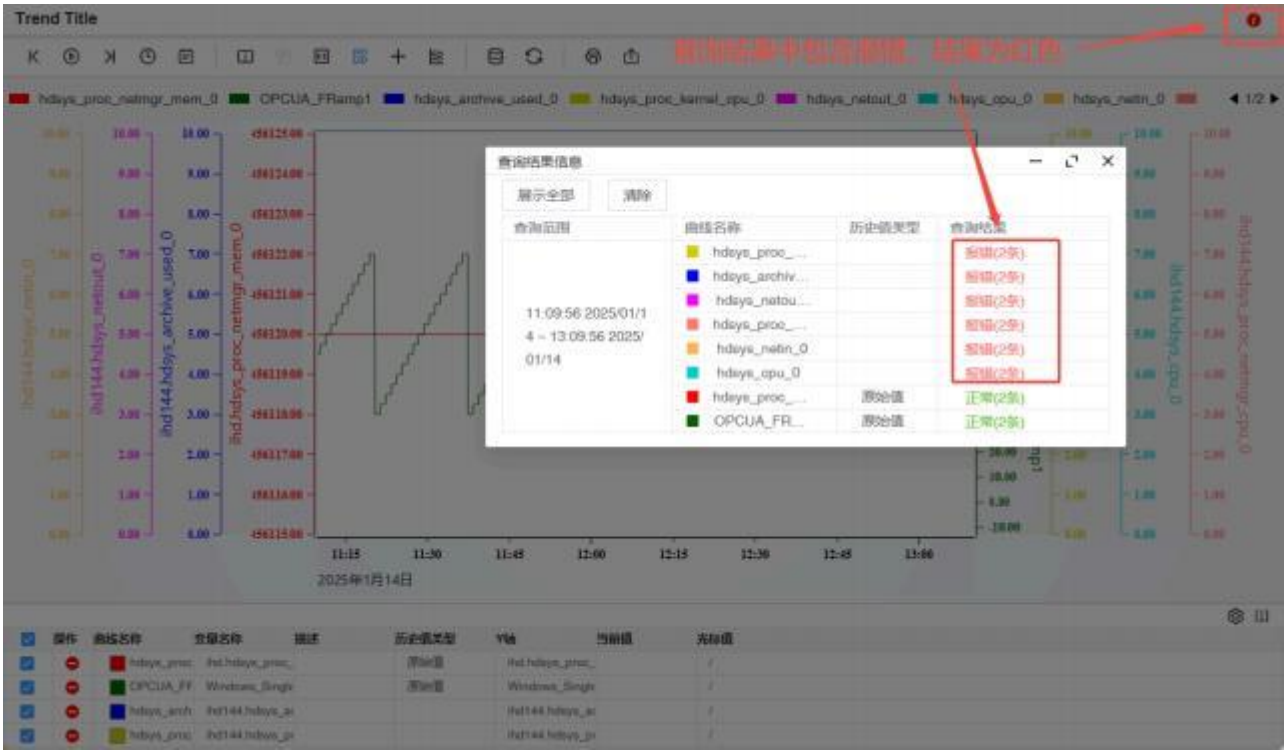
•点击表格旁的信息按钮



信息按钮

•如果最新一批查询结果包含错误信息，则按钮会变红。反之，按钮为黑色





切换展示的查询结果

•趋势控件提供了切换展示的查询结果的功能，可以在运行态切换查询结果。

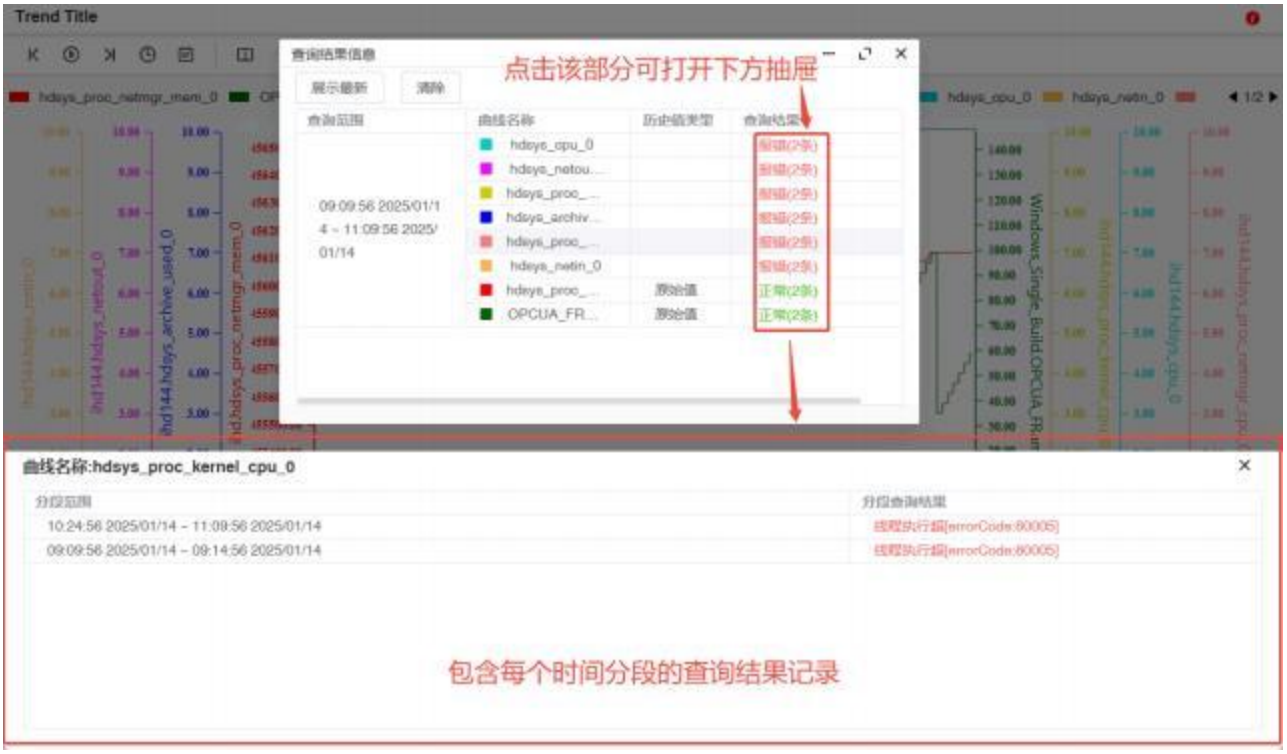
左图为展示最新一批查询结果的效果；右图为展示全部查询结果的效果全部数据最多只会包含最新的10批查询结果记录

清空查询结果记录

•查询结果信息表有两种展示模式：展示最新一批查询结果记录，和展示全部查询结果记录。可以通过点击表格左上角的按钮进行切换

查看详细的查询结果

•点击表格中，“查询结果”一栏的单元格，可以从下方打开一个抽屉，展示当前曲线每个时间分段的查询结果。

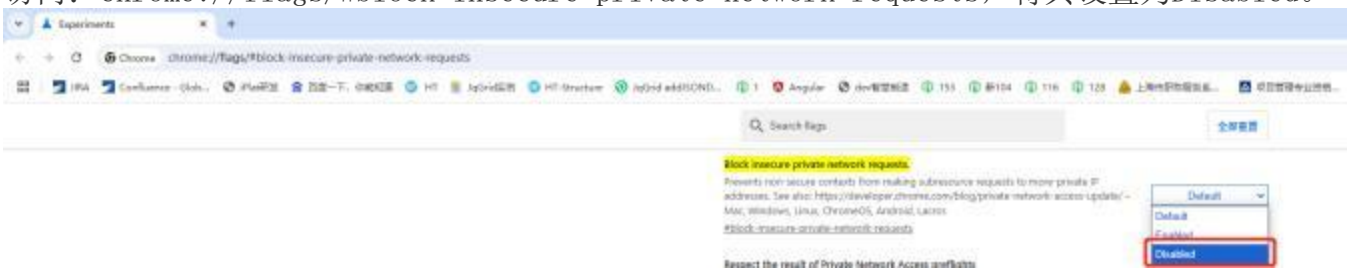


安全性问题导致历史数据查询失败

•问题表现：文本趋势打开后，历史部分稳定不绘制，直接从实时部分绘制。

原因：由于安全性问题，可能会阻止请求发送，导致历史部分的请求失败，实时部分为推送，可以正常绘制。

访问：chrome://flags/#block-insecure-private-network-requests，将其设置为Disabled。



文本趋势V1、历史趋势(旧)的queryTagDescription请求的调用修改

代码版本为iplat-cm 5.29 P10

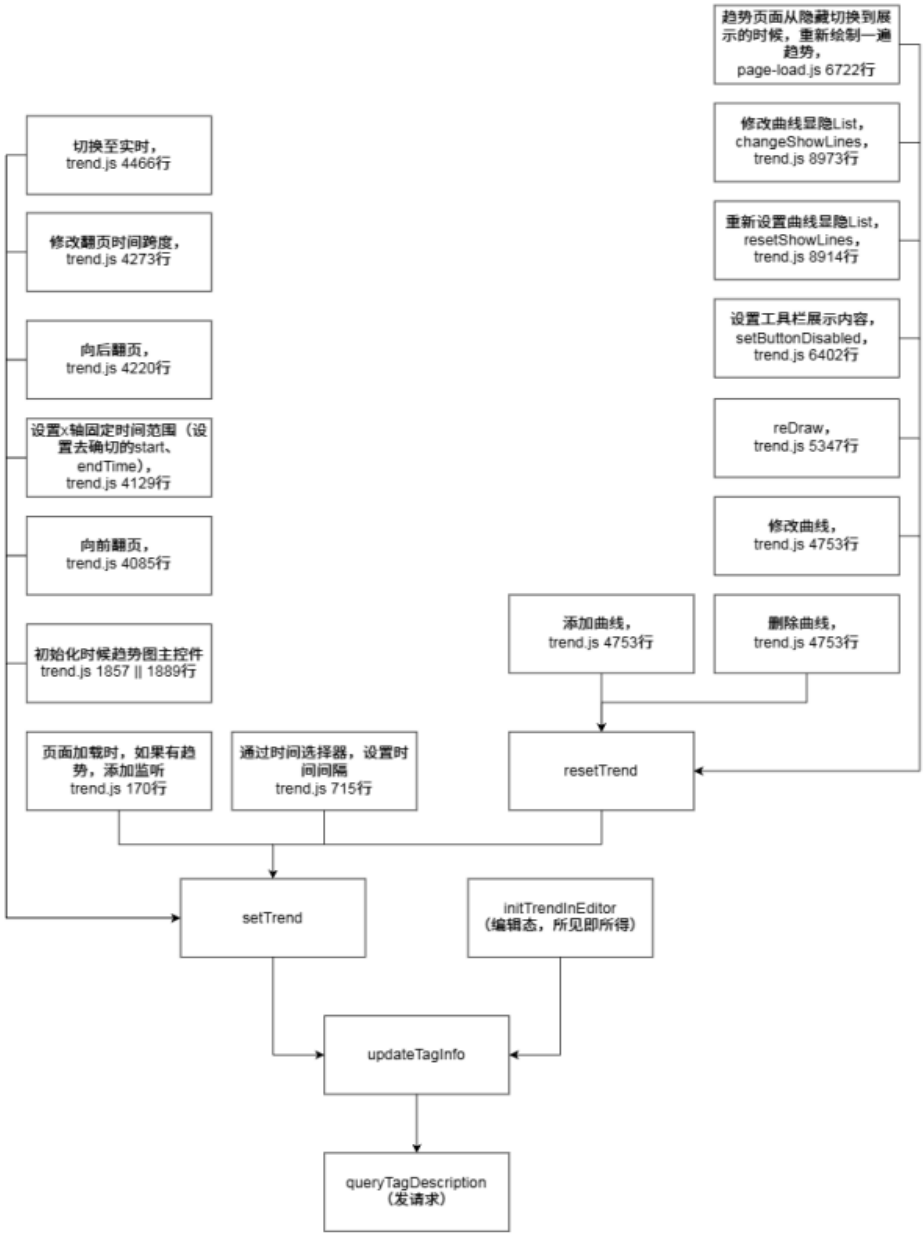
问题：queryTagDescription请求太耗时，希望减少该消耗。

思路1.减少queryTagDescription次数：

目前遇到的困难：很多queryTagDescription都是合理存在的

思路2.通过注册来代替queryTagDescription：

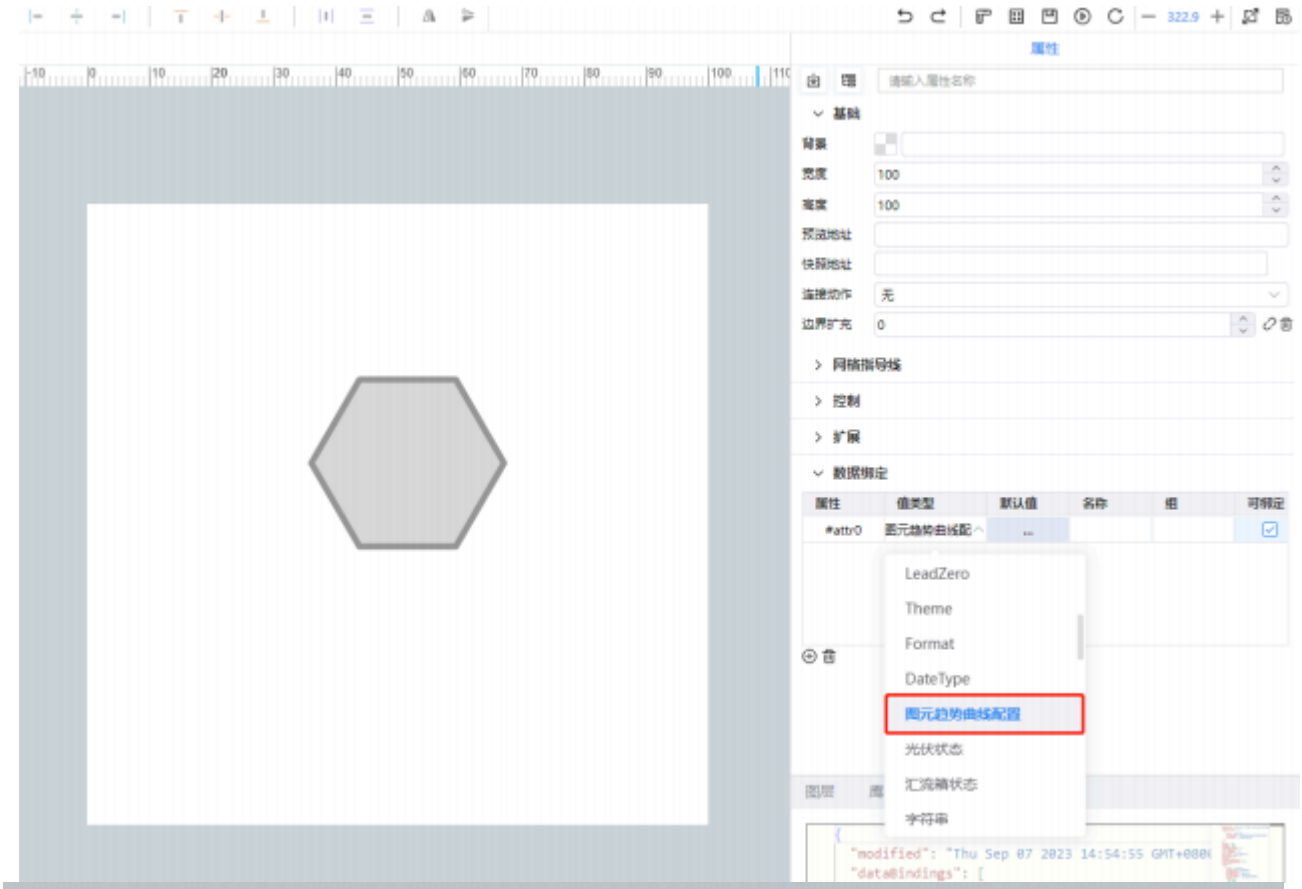
目前遇到的困难：ihd的注册数据，不正确。后来推送正确的之后，由于已经注册过，所以不会再注册，因此也不会修改

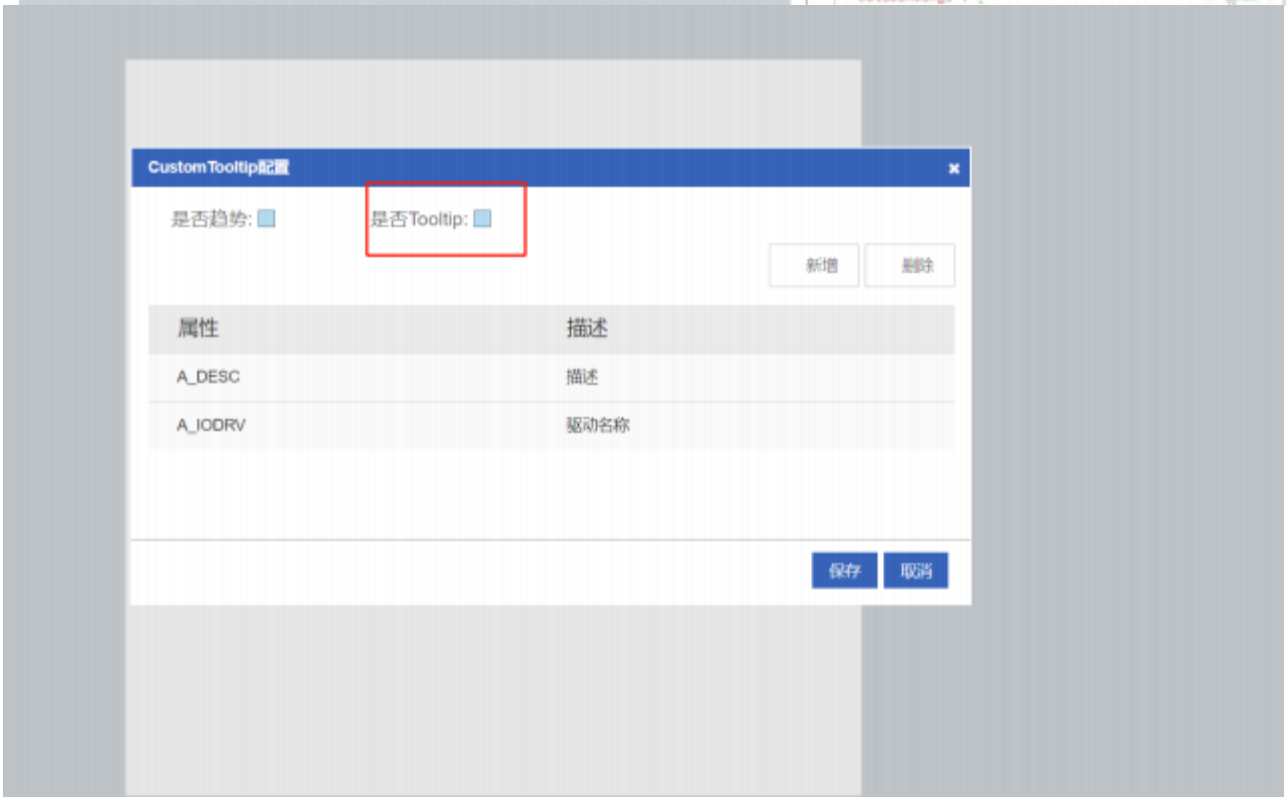


目前选择思路二进行修改。
1.将原本的queryTagDescription，替换为registTagDescription
2.registTagDescription中的点注册通过追加注册的方法appendRegister来实现（这里只追加注册，tag点的描述、单位、上限、下限）
3.追加注册后，通过延迟（2s）读取【延迟的原因：在读取tagMap的时候，注册可能没有完成】
4.基于获取的内容，去更新表格、更新趋势中的tag信息（tag对象）、更新趋势中line的信息。
【待review】IPLAT-13372【文本趋势V1】将趋势的描述的推送修改为注册的方式(!2965)・Merge requests・IPLAT/iplat-cm・GitLab

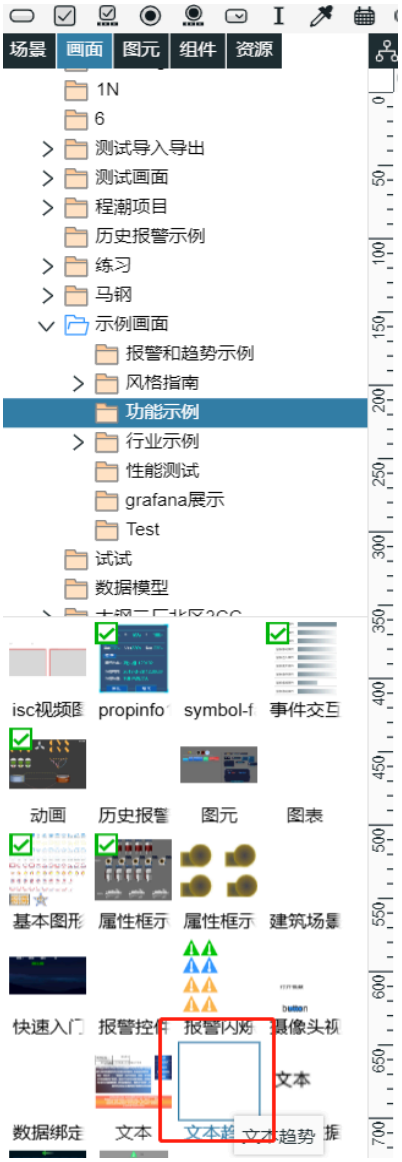
5.34 图元趋势

图元趋势，是使用图元设计器，设计带有历史趋势的图元。
1.对于图元要设置通用tooltip的变量点，勾选显示trend





2. 弹出的画面默认是示例画面/功能示例/文本趋势. json



如果已经有该画面，需要删除该画面，新建一个同名的画面，从上方拖动一个趋势到画面中3.复制tag点名称

运行态，在弹出趋势的元素上右键，会弹出复制tag点名称的选项，点击后会将绑定的tag点名称复制到剪切板

说明：

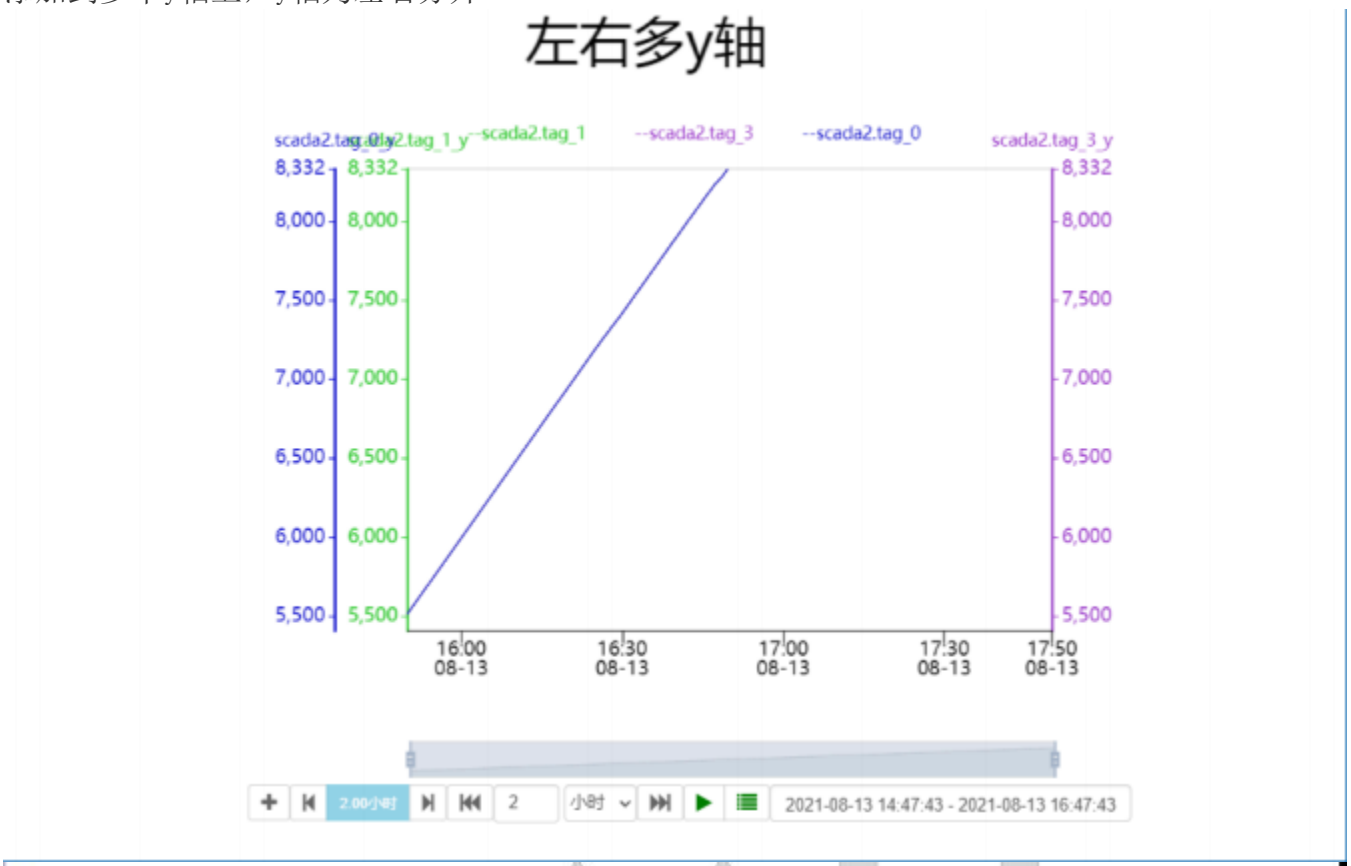
每个画面（包括以iframe形式嵌入对的子画面或切换画面）只会打开一个弹出趋势的画面，当右键图元或文本打开趋势时，曲线会添加到该趋势画面中，后续添加的曲线都会添加到指定的趋势画面中

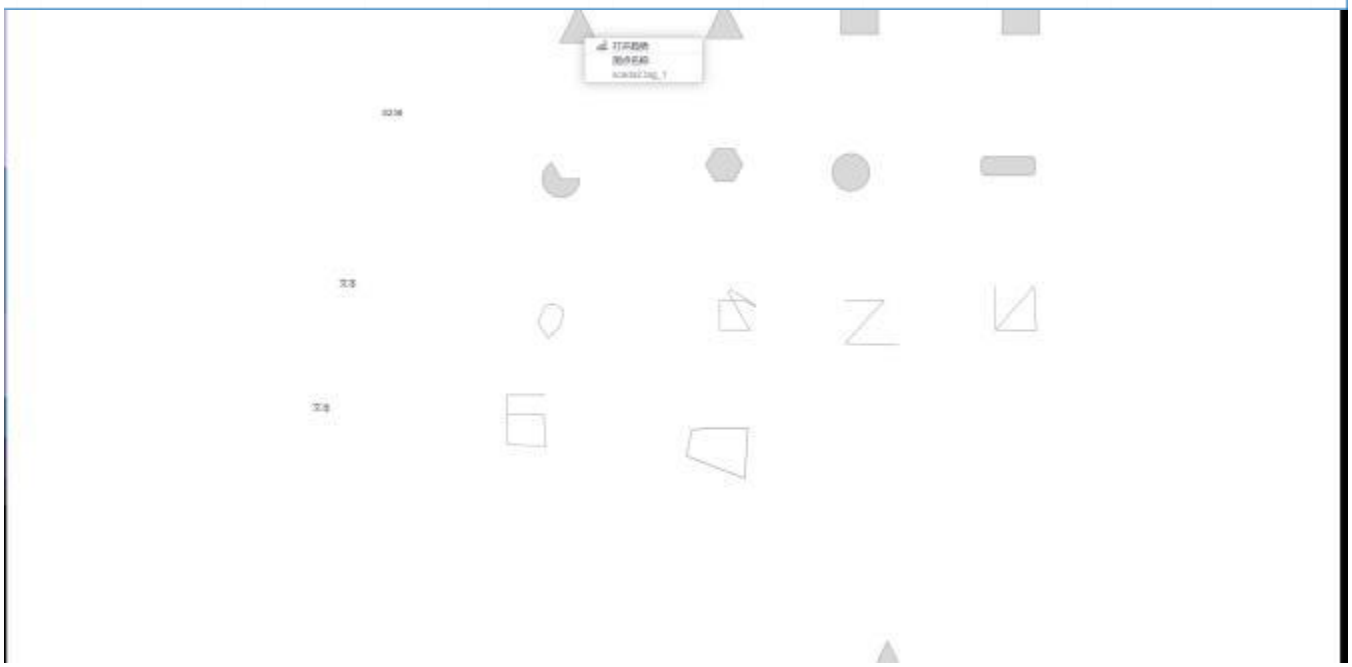
相关配置项：（脚本编辑器或者全局脚本）

示例场景

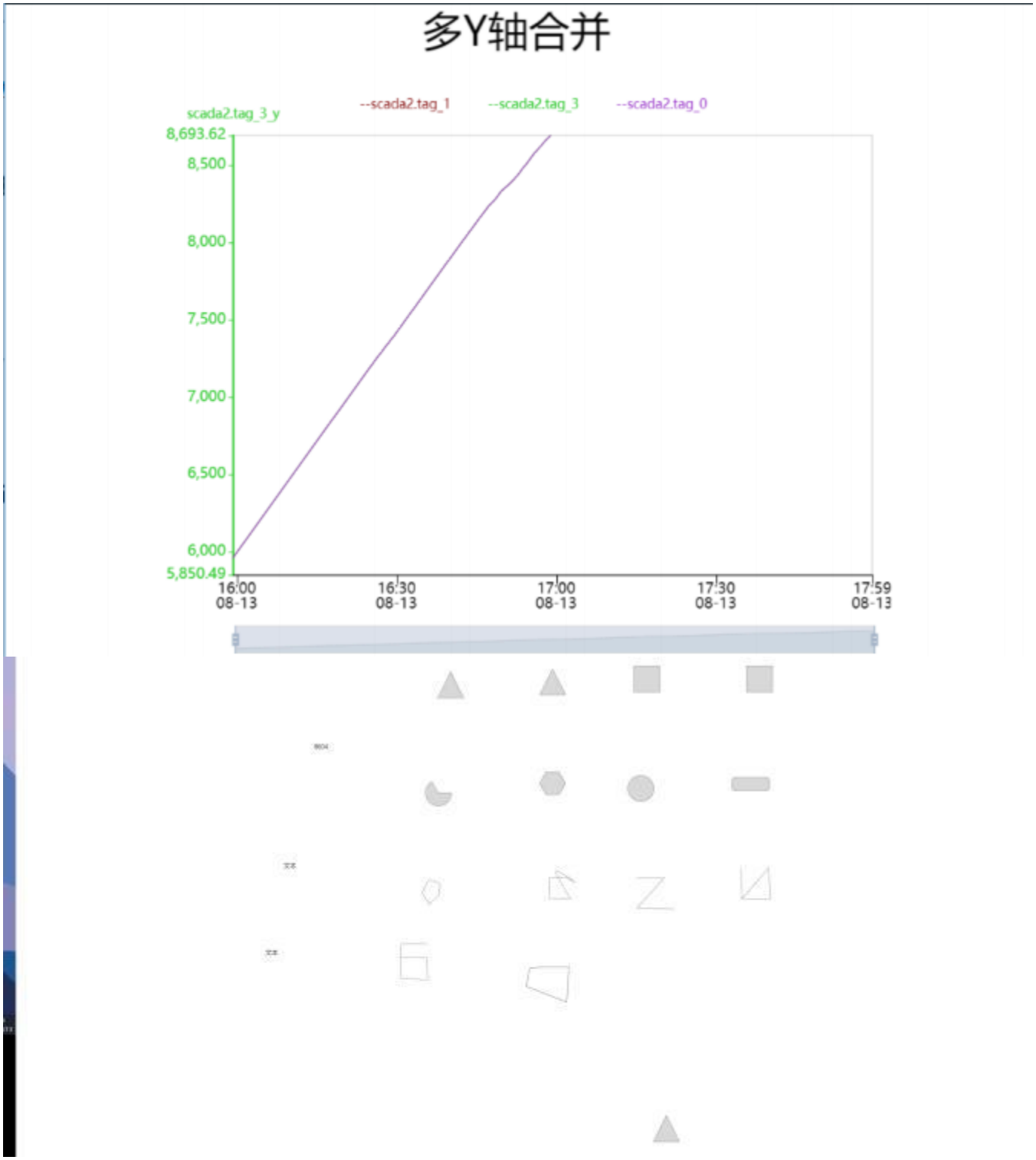
场景1：每个曲线都对应一个自己的y轴，y轴为左右的方式显示

修改文本趋势的配置如下，删除所有y轴，不勾选“多y轴只显示1条”，这时文本趋势的曲线会添加到多个y轴上，y轴为左右分开





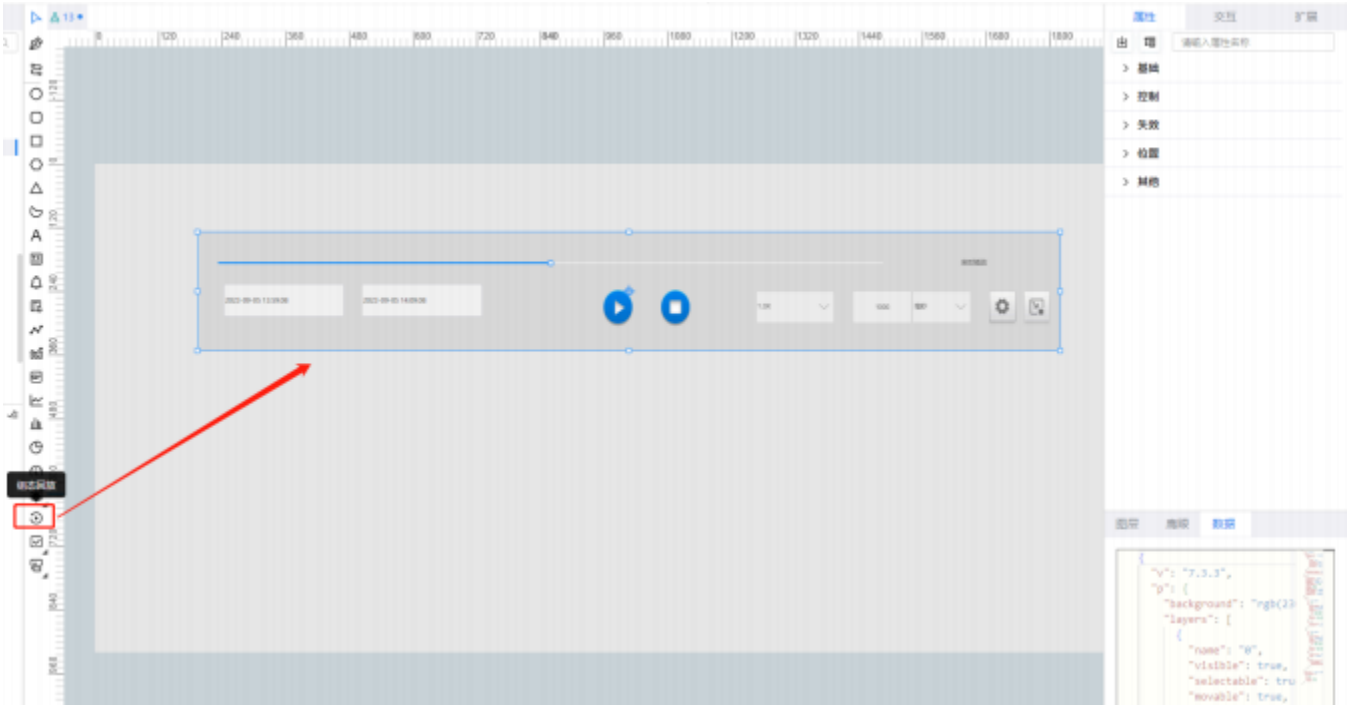
场景2：每个曲线都对应一个自己的y轴，y轴显示在同一个轴上，点击y轴会切换显示的y轴对于文本趋势的那个画面，删除所有y轴，勾选“多y轴只显示1条”



5.35 组态回放控件

组态回放控件配置

1.打开2D设计器，新建一个画面，在工具栏上点击“组态回放”控件，然后在画面上空白的地方拖拽，以便将“组态回放”控件放置在画面上。



2.选中组态回放控件后点，可以在扩展页看到回放控件特有配置信息

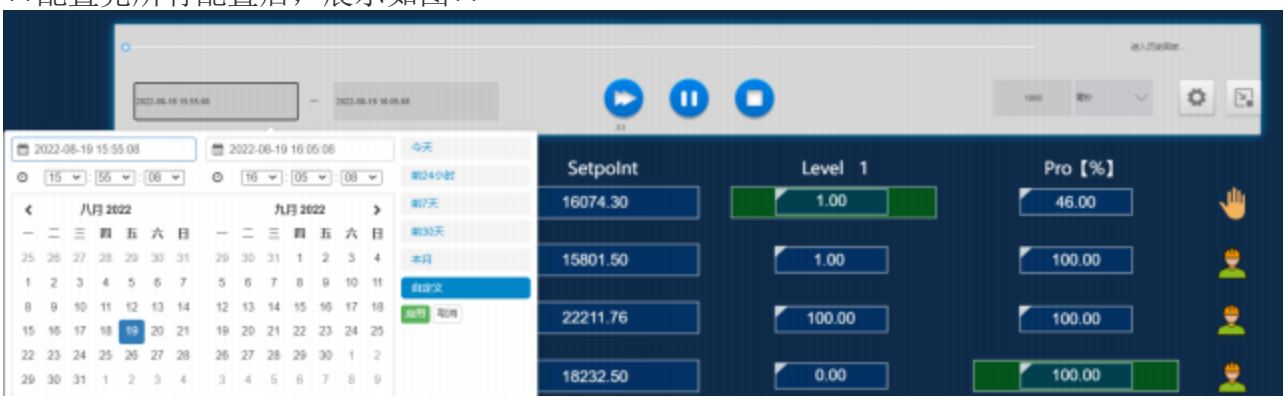
（1）回放

- 播放间隔：
- 起始中止时间：
- 起始时间：
- 中止时间：
- 格式：时间格式（年：月：日时：分：秒）
- 日期类型：
- 时间间隔单位：
- 文本颜色：
- 时间间隔背景色：
- 进度条背景色：
- 播放进度：
- 播放开始时间：
- 播放结束时间：
- 播放状态：
- 回放提示颜色：回放时的提示边框颜色

（2）其他

- 回放背景色：回放时的画面背景颜色
- 回放其他页面：
- 最大化可见：
- 展示回放颜色：
- 字体：
- 回放速率：

配置完所有配置后，展示如图



预览后可选择回放时间段。
这样，回放控件的配置就完成了。

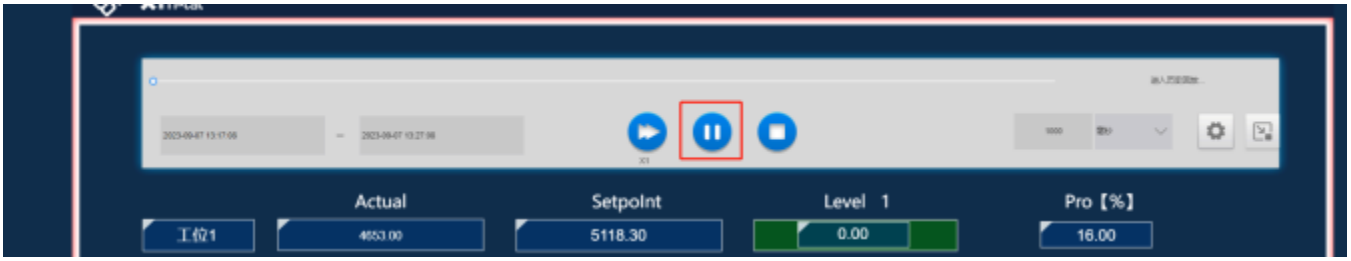
回放倍速

1. 回放的速率提供了回放倍速、时间长度和时间单位三个配置项，可在控件上选择或输入更改。

0.06477865

2. 在回放时，更改上述配置也会立即生效。

回放开始

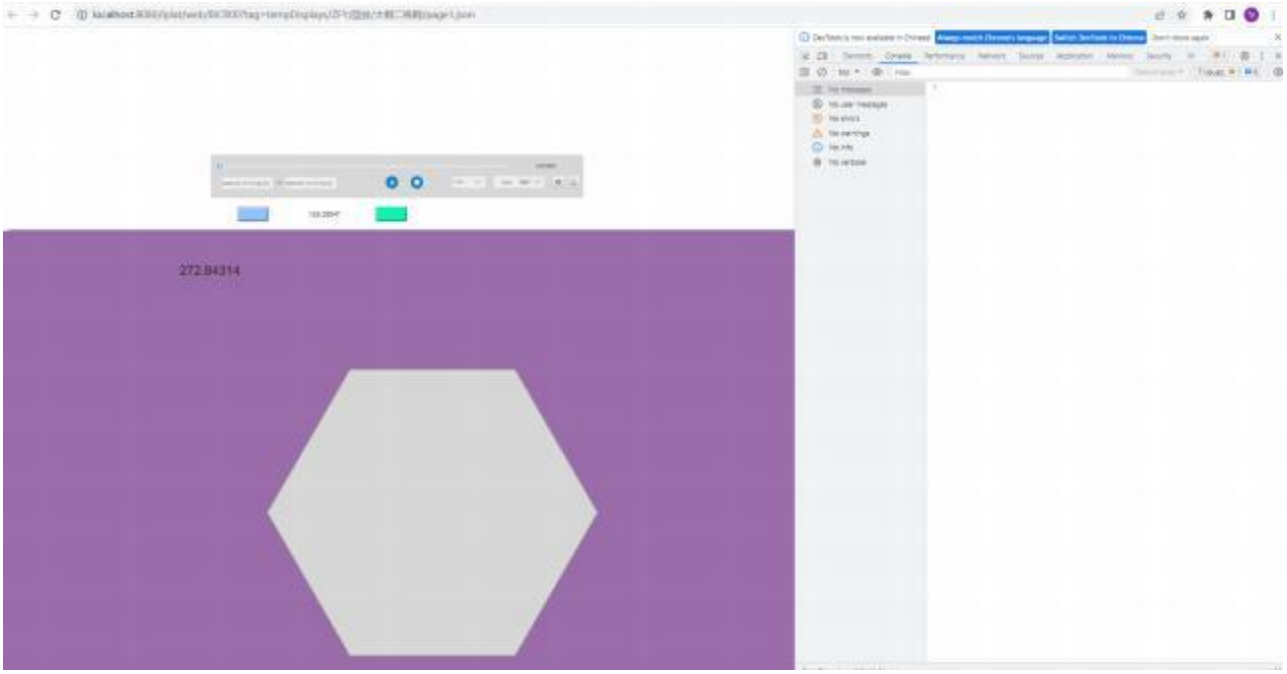


•运行态点击如上按钮，即可进入回放状态，回放状态画面边框会显示配置的边框提示颜色

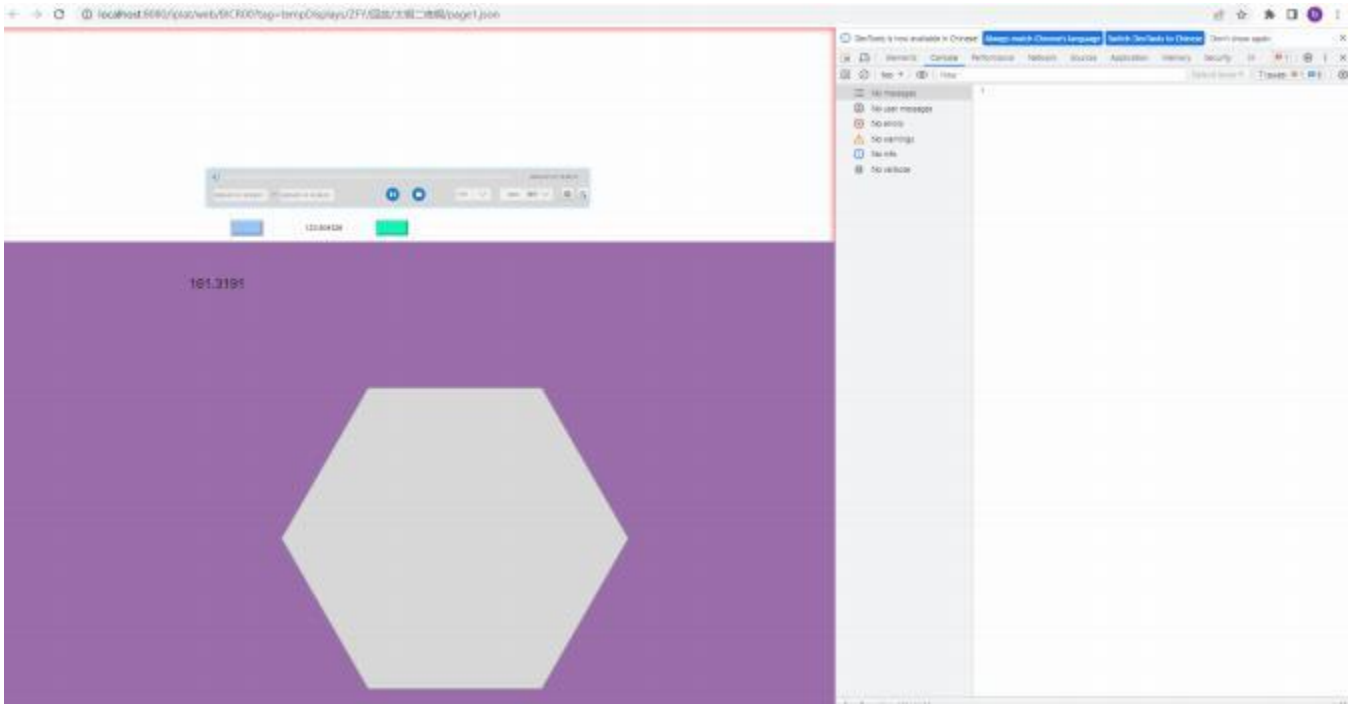
回放暂停

1. 回放暂停时现在支持点击画面，但是禁止下发操作。

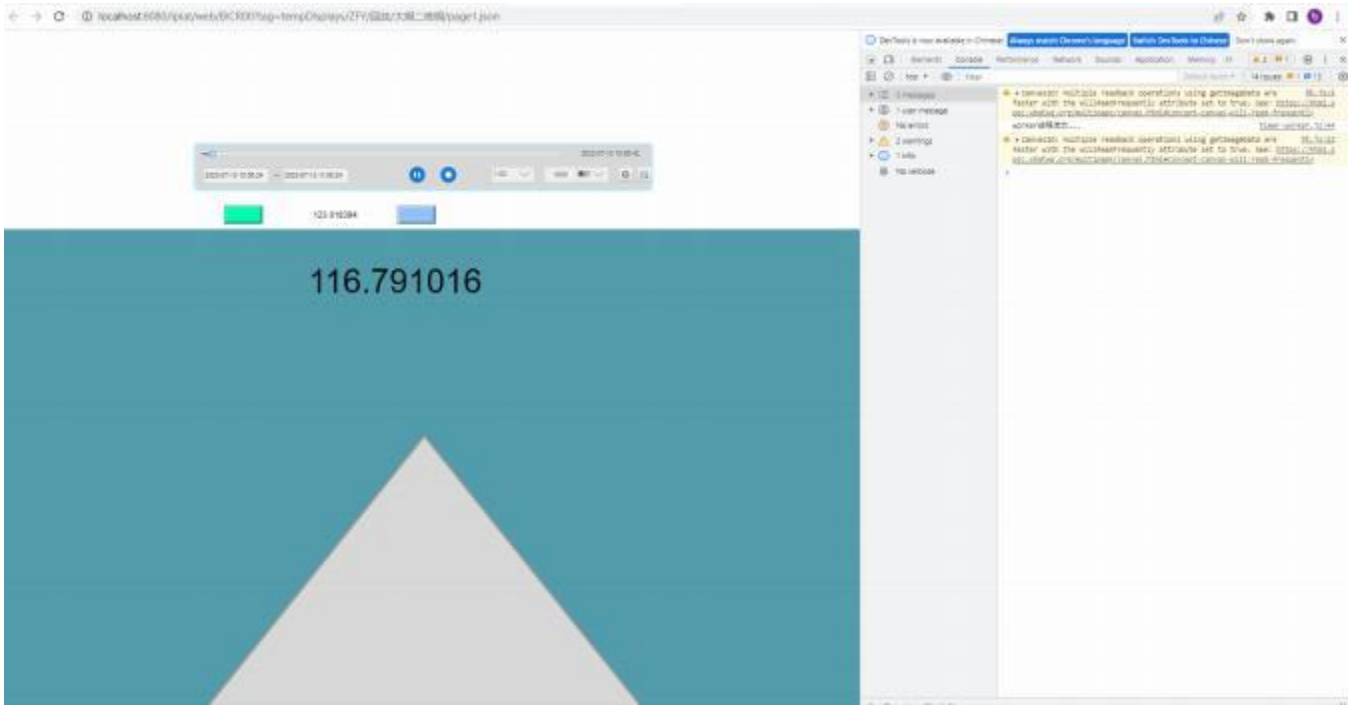
回放状态中，是无法进行点击的



进行暂停后，可以触发事件



暂停时无法下发

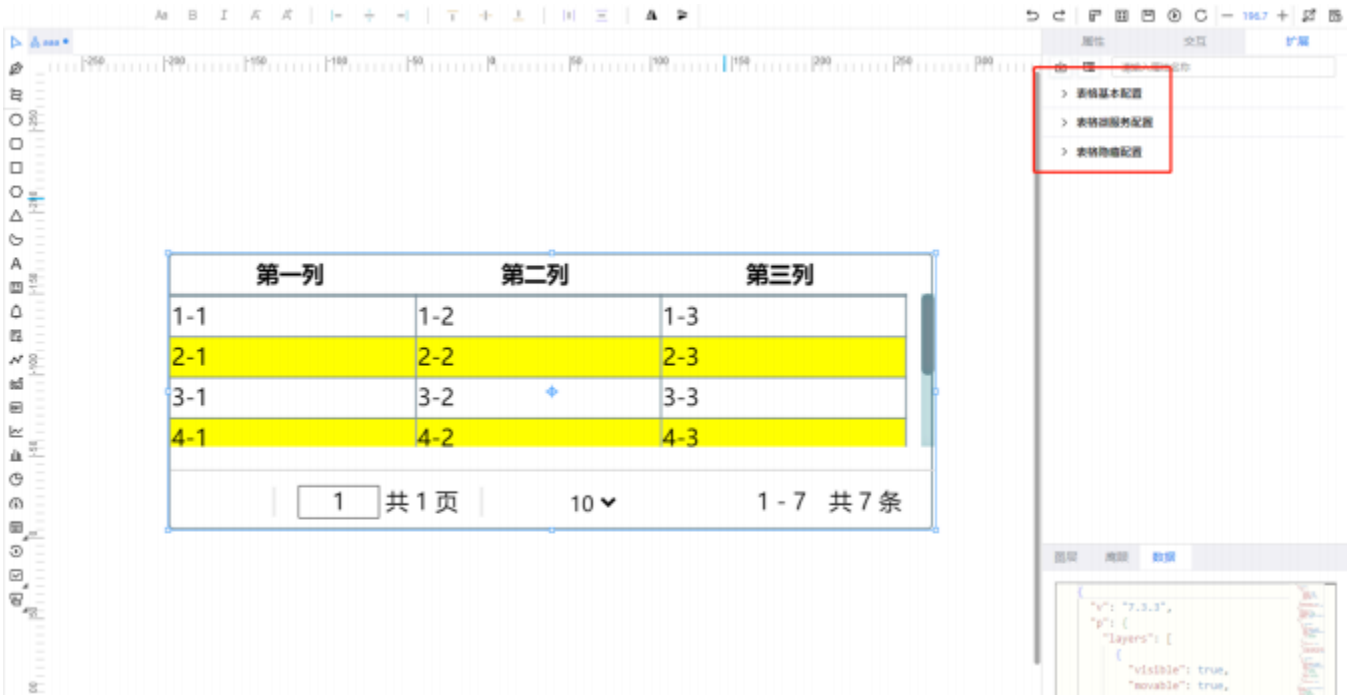


5.36表格控件

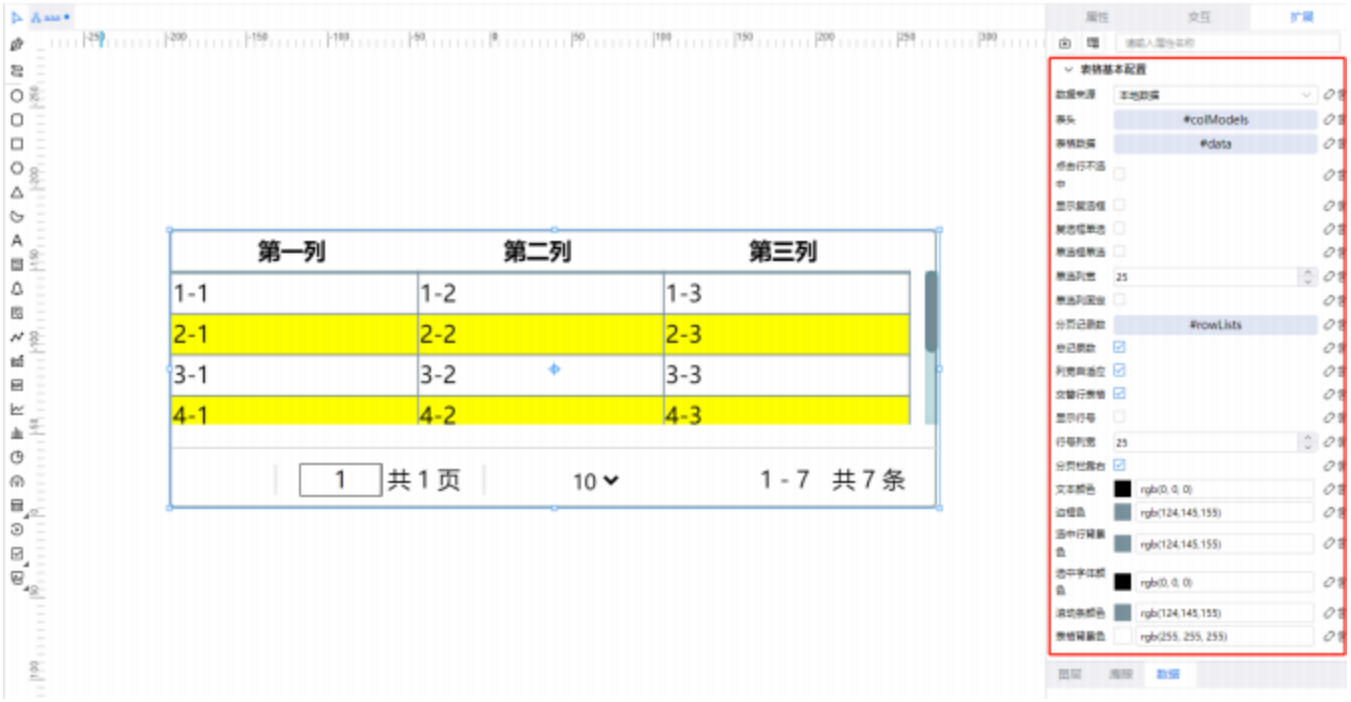
表格控件集成在工具栏中，拖拽到画布中即可使用

第一列
第二列
第三列
1-1
1-2
1-3
2-1
2-2
2-3
3-1
3-2
3-3
4-1
4-2
4-3
1
共 1 页
10
1 - 7 共 7 条

功能列表


第一列
第二列
第三列
1-1
1-2
1-3
2-1
2-2
2-3
3-1
3-2
3-3
4-1
4-2
4-3
1
共 1 页
10
1 - 7 共 7 条

基本配置



0. 数据来源

可选择本地数据、数据库和微服务三种数据来源；

本地数据的数据来源于’表格数据’配置项

1. 表头

2. 表格数据

3. 点击行不选中：只有点击复选框时选中行

4. 显示复选框：

5. 复选框单选：

6. 单选框：

7. 单选列宽：

8. 分页记录数：

9. 总记录数：

10. 列宽自适应：

11. 交替行表格：

12. 显示行号：

13. 行号列宽：

14. 文本颜色：

15. 边框色：

16. 选中行背景色：

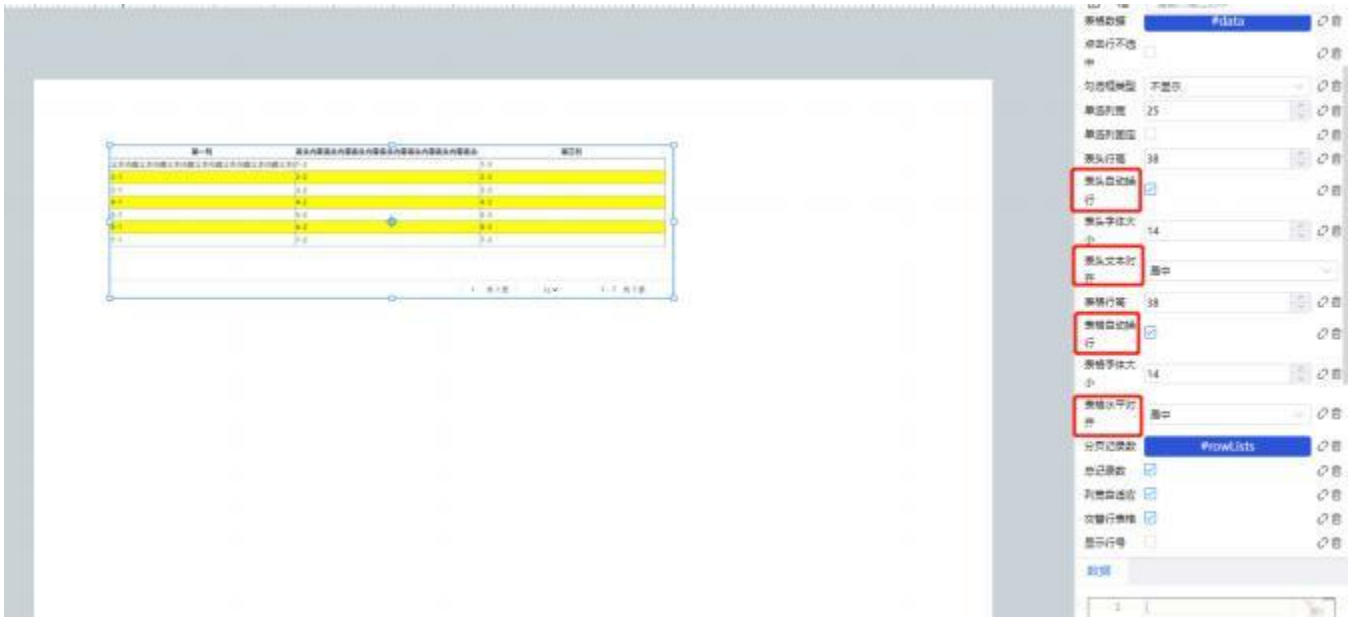
17. 选中字体颜色：

18. 滚动条颜色：

19. 表格背景色：

表格与表头的自动换行和水平对齐功能：

配置：



效果：

第一列	表头内容表头内容表头内容表头内容表头内容表头内容表头内容	第三列
文本内容	1-2	1-3
2-1	2-2	2-3
3-1	3-2	3-3
4-1	4-2	4-3

1 共1页 10 1-7 共7条

微服务配置

配置介绍

1.数据来源选择"微服务"

2.微服务配置：

微服务ID：微服务号可在EDXM01页面中查询

微服务入参：微服务的请求参数

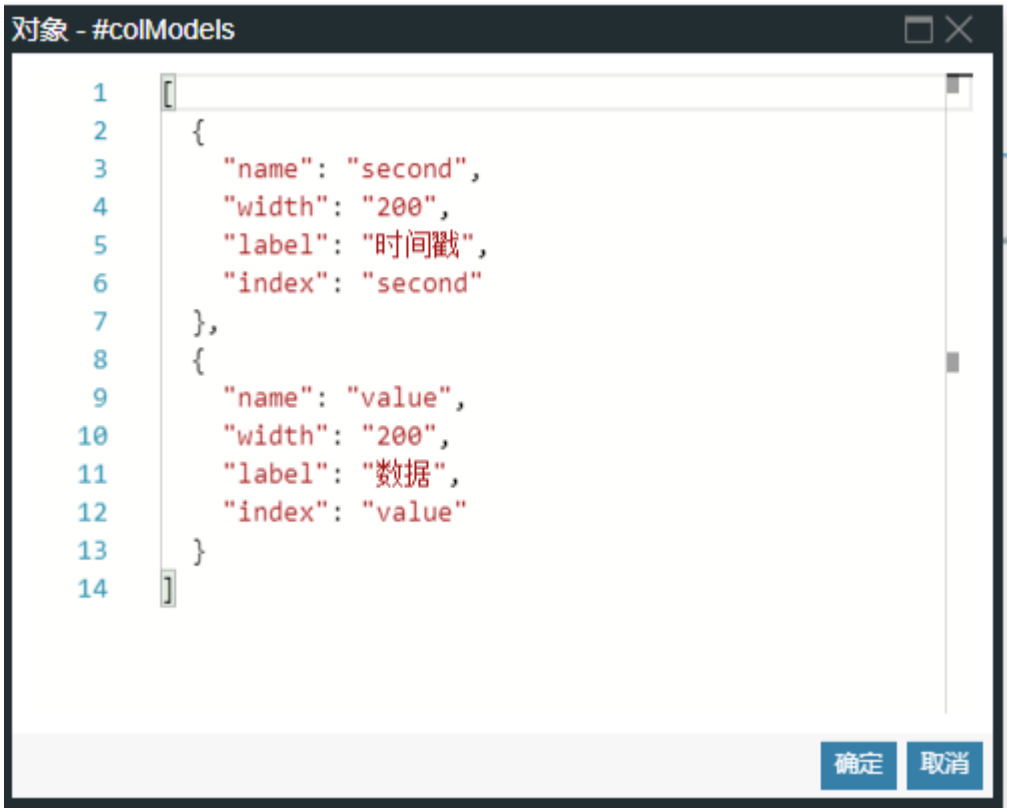
是否轮询&轮询频率：启用轮询后，会以轮询频率为时间间隔定时请求微服务，将请求返回数据渲染在表格中

数据处理：将微服务返回的数据经过处理后返回数组，这个数组会渲染在表格中

微服务请求示例

1.数据来源选择为微服务

2.配置表头



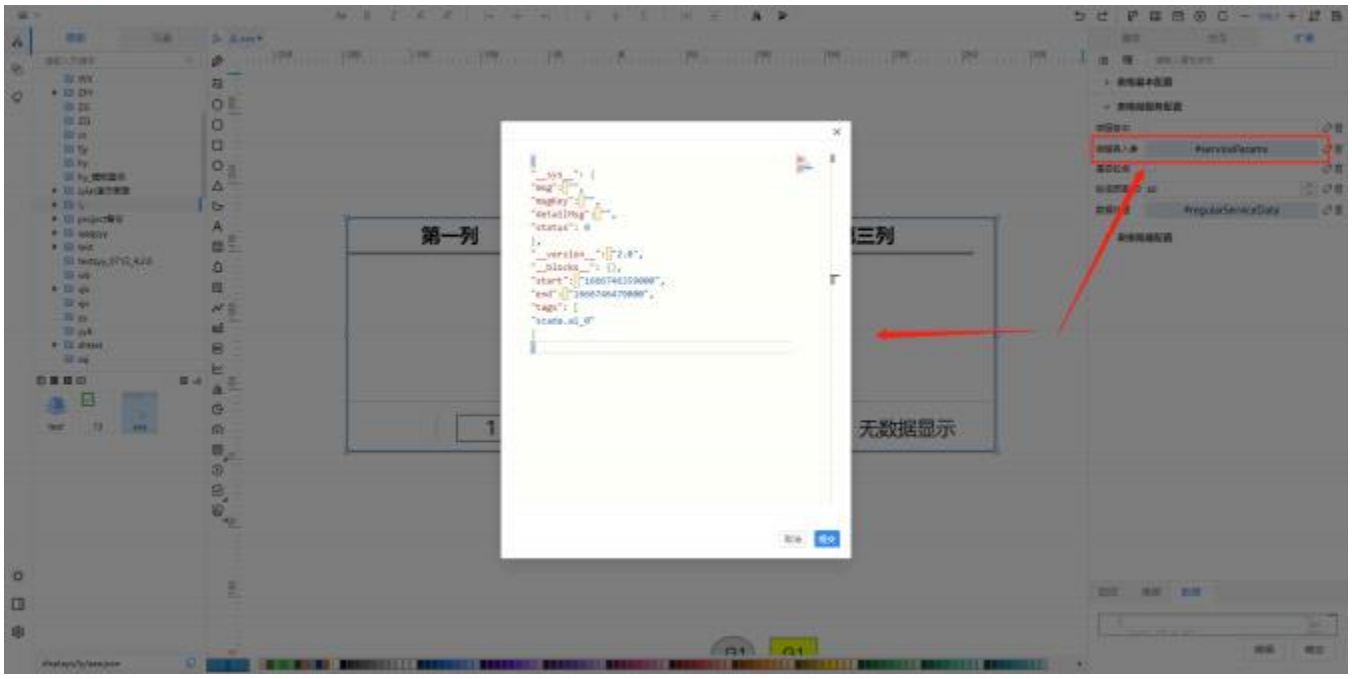
3.配置微服务相关配置

配置在EDXM01页面中查到的微服务ID

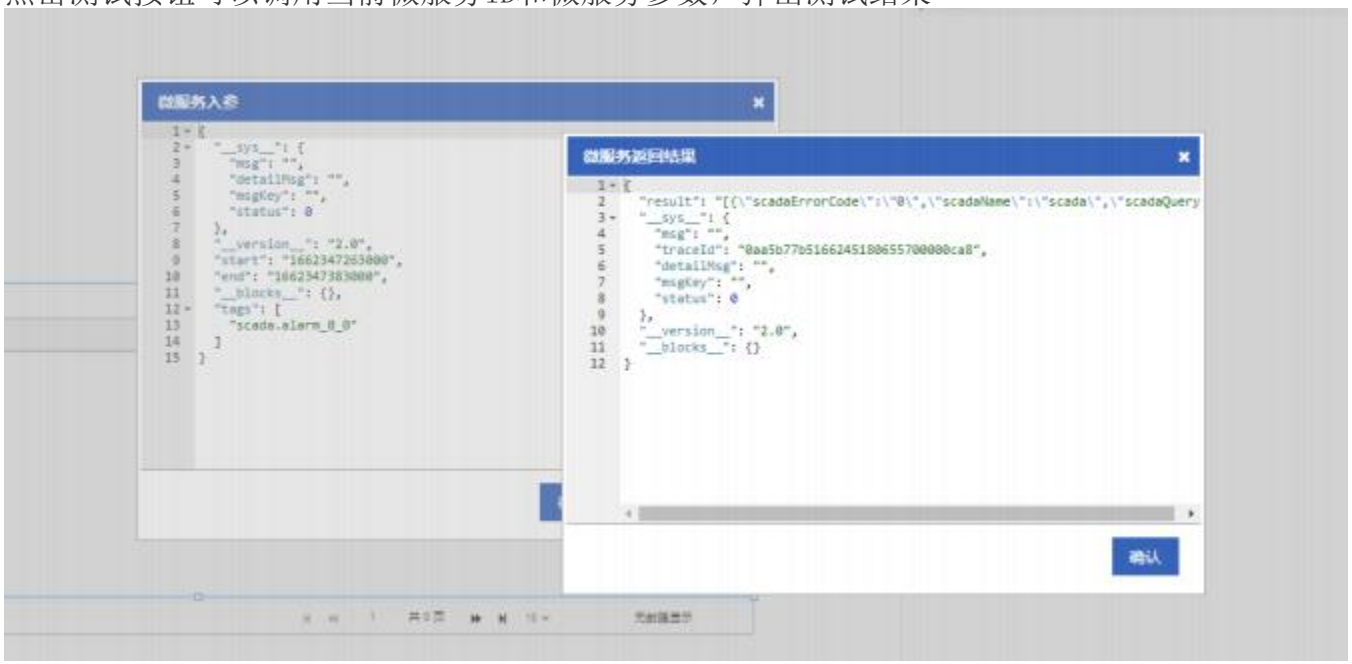
点击按钮弹出框，填入微服务入参，点击确认按钮确认

微服务入参

```
"__sys__": {
"msg": "",
"msgKey": "",
"detailMsg": "",
"status": 0
},
"__version__": "2.0",
"__blocks__": {},
"start": "1666746359000",
"end": "1666746479000",
"tags": [
"scada.ai_0"
]
}
```



点击测试按钮可以调用当前微服务ID和微服务参数，弹出测试结果



配置数据处理回调

回调函数返回一个数组，数组为对象数组，每个对象里的参数名(key)都和表头配置的name对应

函数 - #regularServiceData

```
function(response){
    //返回值为数组
    let result = [];
    //打印微服务返回数据
    console.log(response);
    //处理返回数据
    JSON.parse(response.result)[0].scadaQueryResult[0].rawList.map(v=>{
        result.push({
            second:v.second,
            value:v.value
        })
    })
    console.log(result)
    //返回结果
    return result;
}
```

确定 取消

表头结构：

表头格式

```
{
"name": "second",
"width": "200",
"label": "时间戳",
"index": "second"
},
{
"name": "value",
"width": "200",
"label": "数据",
"index": "value"
}
]
```

回调函数：

回调函数

```
    //返回值为数组
    let result = [];
    //打印微服务返回数据
    console.log(response);
    //处理返回数据
    JSON.parse(response.result)[0].scadaQueryResult[0].rawList.map(v=```{
        result.push({
            second:v.second,
            value:v.value
        })
    })
    //返回结果
    return result;
}
```

返回结果：

返回结果


```
"result":"[{\"scadaErrorCode\":\"0\",\"scadaName\":\"scada\",\"scadaQueryResult\":
                                                                    [{\"errorCode\"
"__sys__":{
"msg":"",
"traceId":"ac1f3001116669415753250000005c1",
"detailMsg":"",
"msgKey":"",
"status":0
},
"__version__":"2.0",
"__blocks__":{}
}
```


处理过后返回结果：

处理后返回结果

```
{
"second": "1666746359554",
"value": "16400.00000000"
},
{
"second": "1666746363250",
"value": "16500.00000000"
},
{
"second": "1666746363541",
"value": "16500.00000000"
},
{
"second": "1666746367652",
"value": "16600.00000000"
},
{
"second": "1666746367698",
"value": "16600.00000000"
},
{
"second": "1666746371371",
"value": "16700.00000000"
},
{
"second": "1666746371543",
"value": "16700.00000000"
},
{
"second": "1666746375065",
"value": "16800.00000000"
},
{
"second": "1666746375541",
"value": "16800.00000000"
},
{
"second": "1666746378835",
"value": "16900.00000000"
```

```
},
{
"second": "1666746379539",
"value": "16900.00000000"
},
{
"second": "1666746382631",
"value": "17000.00000000"
},
{
"second": "1666746386063",
"value": "17100.00000000"
},
{
"second": "1666746386549",
"value": "17100.00000000"
},
{
"second": "1666746389828",
"value": "17200.00000000"
},
{
"second": "1666746390534",
"value": "17200.00000000"
},
{
"second": "1666746394040",
"value": "17300.00000000"
},
{
"second": "1666746394543",
"value": "17300.00000000"
},
{
"second": "1666746397966",
"value": "17400.00000000"
},
{
"second": "1666746398539",
"value": "17400.00000000"
```

```
},
{
"second": "1666746402148",
"value": "17500.00000000"
},
{
"second": "1666746402566",
"value": "17500.00000000"
},
{
"second": "1666746406070",
"value": "17600.00000000"
},
{
"second": "1666746406537",
"value": "17600.00000000"
},
{
"second": "1666746410389",
"value": "17700.00000000"
},
{
"second": "1666746410543",
"value": "17700.00000000"
},
{
"second": "1666746414501",
"value": "17800.00000000"
},
{
"second": "1666746414549",
"value": "17800.00000000"
},
{
"second": "1666746418741",
"value": "17900.00000000"
},
{
"second": "1666746418787",
"value": "17900.00000000"
```

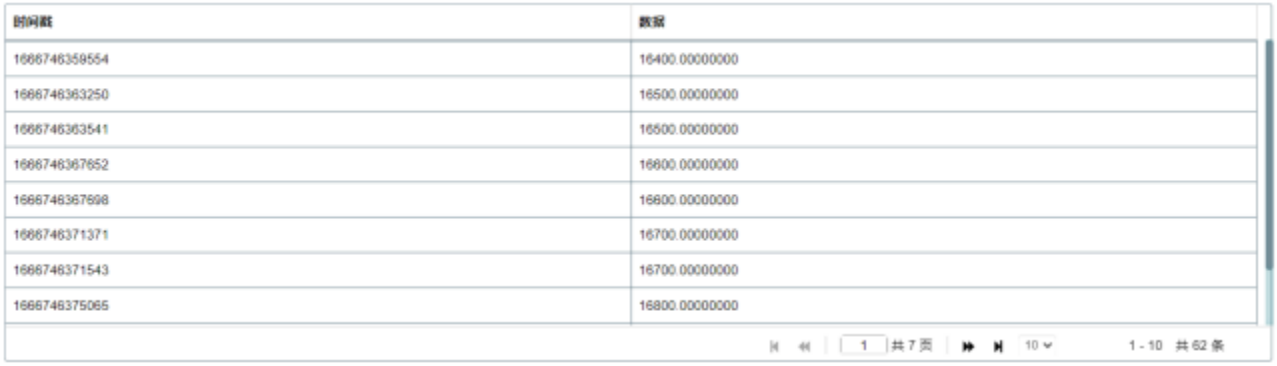
```
},
{
"second": "1666746419551",
"value": "17900.00000000"
},
{
"second": "1666746423270",
"value": "18000.00000000"
},
{
"second": "1666746423548",
"value": "18000.00000000"
},
{
"second": "1666746427158",
"value": "18100.00000000"
},
{
"second": "1666746427539",
"value": "18100.00000000"
},
{
"second": "1666746431839",
"value": "18200.00000000"
},
{
"second": "1666746431885",
"value": "18200.00000000"
},
{
"second": "1666746432540",
"value": "18200.00000000"
},
{
"second": "1666746436256",
"value": "18300.00000000"
},
{
"second": "1666746436547",
"value": "18300.00000000"
```

```
},
{
"second": "1666746440895",
"value": "18400.00000000"
},
{
"second": "1666746440940",
"value": "18400.00000000"
},
{
"second": "1666746441546",
"value": "18400.00000000"
},
{
"second": "1666746445607",
"value": "18500.00000000"
},
{
"second": "1666746445648",
"value": "18500.00000000"
},
{
"second": "1666746449801",
"value": "18600.00000000"
},
{
"second": "1666746449849",
"value": "18600.00000000"
},
{
"second": "1666746450538",
"value": "18600.00000000"
},
{
"second": "1666746453713",
"value": "18700.00000000"
},
{
"second": "1666746457799",
"value": "18800.00000000"
```

```
},
{
"second": "1666746457845",
"value": "18800.00000000"
},
{
"second": "1666746458540",
"value": "18800.00000000"
},
{
"second": "1666746461983",
"value": "18900.00000000"
},
{
"second": "1666746462543",
"value": "18900.00000000"
},
{
"second": "1666746466155",
"value": "19000.00000000"
},
{
"second": "1666746466537",
"value": "19000.00000000"
},
{
"second": "1666746470984",
"value": "19100.00000000"
},
{
"second": "1666746471033",
"value": "19100.00000000"
},
{
"second": "1666746471540",
"value": "19100.00000000"
},
{
"second": "1666746473323",
"value": "19200.00000000"
```

```
},
{
"second": "1666746473541",
"value": "19200.00000000"
},
{
"second": "1666746474539",
"value": "19200.00000000"
}
]
```


4.运行画面，微服务请求的数据经过处理后渲染在表格中



时间戳	数据
1666746359554	16400.00000000
1666746363250	16500.00000000
1666746363541	16500.00000000
1666746367652	16600.00000000
1666746367698	16600.00000000
1666746371371	16700.00000000
1666746371543	16700.00000000
1666746375065	16800.00000000

1 共7页 10 1 - 10 共62条

表格隐藏配置

- 表格隐藏配置：
- 表格内容高度：

脚本方法

sys.getIplatTable(table_name)获取表格对象

参数：

table_name字符串类型，表格控件的名称

返回：

无

示例：

```
    [参数]
    table_name 字符串类型，表格控件的名称
    [返回值]
    表格控件对象
*/
sys.getIplatTable(table_name);
```

iplatTable.addJSONData(jsonData)设置表格当前页

参数：

jsonData 表格当前页数据

数据格式：

{page:number,rows:array,total:number,records:number}
page:当前第几页
rows:当前页数据
total:总页数
records:总记录数
返回：

无

示例：

```
let iplatTable = sys.getIplatTable("table_name");
iplatTable.addJSONData({
    page:1,
    rows:[
            {key1:"v1-1",key2:"v2-1"},
            {key1:"v1-2",key2:"v2-2"},
            {key1:"v1-3",key2:"v2-3"},
            {key1:"v1-4",key2:"v2-4"}
        ],
    total:25,
    records:98
});
```

iplatTable.getDataIDs()获取所有行数据的id
参数：

无
返回：

dataIDs所有行数据的id

示例：

```
let iplatTable = sys.getIplatTable("table_name");
let dataIDs = iplatTable.getDataIDs();
```

iplatTable.getDom()获取Dom对
象参数：

无

返回：

dom表格的dom对象

示例：

```
let dom = iplatTable.getDom();
```

iplatTable.getDomSelector()获取jQuery选择器
参数：

无

返回：

domSelector表格的jQuery选择器

示例：

```
let iplatTable = sys.getIplatTable("table_name");
let domSelector = iplatTable.getDomSelector();
```

iplatTable.getGridParam()获取表格的grid参数信息
参数：

无

返回：

params表格的grid参数信息

示例：

```
let iplatTable = sys.getIplatTable("table_name");
let params = iplatTable.getGridParam();
```

iplatTable.getRowData(rowid)获取指定行的数据

参数：

rowid行号（也可通过getDataIDs获取）

返回：

row行数据

示例：

```
let iplatTable = sys.getIplatTable("table_name");
let row = iplatTable.getRowData(rowid);
```

iplatTable.hideCol(cols)隐藏指定列

参数：

cols列名（如果参数为一个列名则隐藏此列，如果给定的是数组则隐藏指定的所有列。格式：[“name1”,”name2”]）

返回：

无

示例：

```
iplatTable.hideCol("name1");
```

iplatTable.showCol(cols)显示指定列

参数：

cols列名（cols可以是数组[“name1”,”name2”],但是name1或者name2必须是colModel(列配置)中的name）

返回：

无

示例：

```
let iplatTable = sys.getIplatTable("table_name");
iplatTable.showCol("name1");
```

iplatTable.resetSelection()重置行的选择状态

参数：

无

返回：

无

示例：

```
let iplatTable = sys.getIplatTable("table_name");
iplatTable.resetSelection();
```

iplatTable.setSelectRowCallback(callback)设置选中行回调

参数：

callback回调函数

函数参数rowid

返回：

无

示例：

```
let iplatTable = sys.getIplatTable("table_name");
iplatTable.setSelectRowCallback(function (rowid) {
    //打印出该选中行数据
    console.log(iplatTable.getRowData(rowid));
});
```

iplatTable.setSelection(rowid,onselectrow)选择或反选指定行

参数：

rowid行号

onselectrow是否触发onSelectRow事件（布尔值，如果onselectrow为ture则会触发事件onSelectRow，onselectrow默认为ture）

返回：

无

示例：

```
let iplatTable = sys.getIplatTable("table_name");
iplatTable.setSelection("1",false);
```

iplatTable.setSelections(rows)批量选择或反选指定行

参数：

rows需要设置的行数组`[{rowid,onselectrow}]`

返回：

无

示例：

```
let iplatTable = sys.getIplatTable("table_name");
iplatTable.setSelections([{rowid:'1',onselectrow:true},{rowid:'4'}])
```

iplatTable.setGridHeight(height)设置表格高度

参数：

height表格高度

返回：

无

示例：

```
let iplatTable = sys.getIplatTable("table_name");
iplatTable.setGridHeight(500);
```

iplatTable.setGridWidth(width)设置表格高度

参数：

width表格宽度

返回：
无

示例：

```
let iplatTable = sys.getIplatTable("table_name");
iplatTable.setGridWidth(1000);
```

iplatTable.setRowData(rowid, data)更新行的值

参数：

rowid行号

data行数据（格式：`{name1:value1,name2:value2…}`name为colModel中名称）

返回：

无

示例：

```
let iplatTable = sys.getIplatTable("table_name");
let data = {name1:value1,name2: value2};
iplatTable.setRowData("1",data);
```

iplatTable.setServiceParams(params)设置微服务参数
参数：

params微服务参数【json对象】

返回：

无

示例：

```
let iplatTable = sys.getIplatTable("table_name");
iplatTable.setServiceParams(params);
```

5.37　多层级下拉框图元

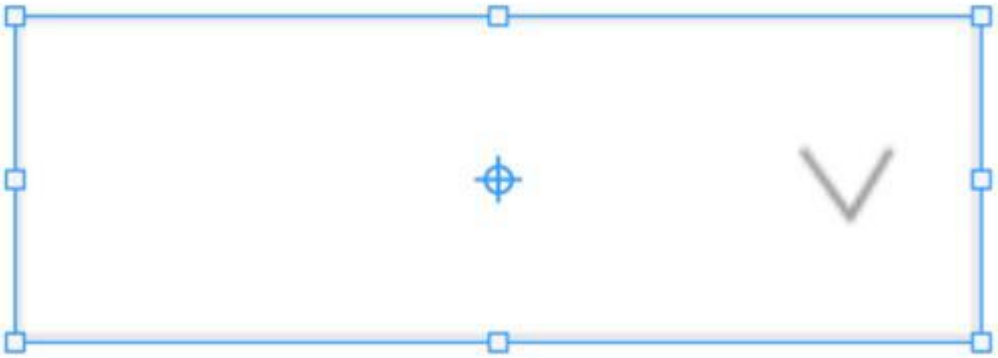
入口

选择图元，在root/系统图元/按钮/cascadeComboxBox文件中的cascadeComboxBox文件

添加

在画面上拖拽该图元可以绘制出对图案如下所示。

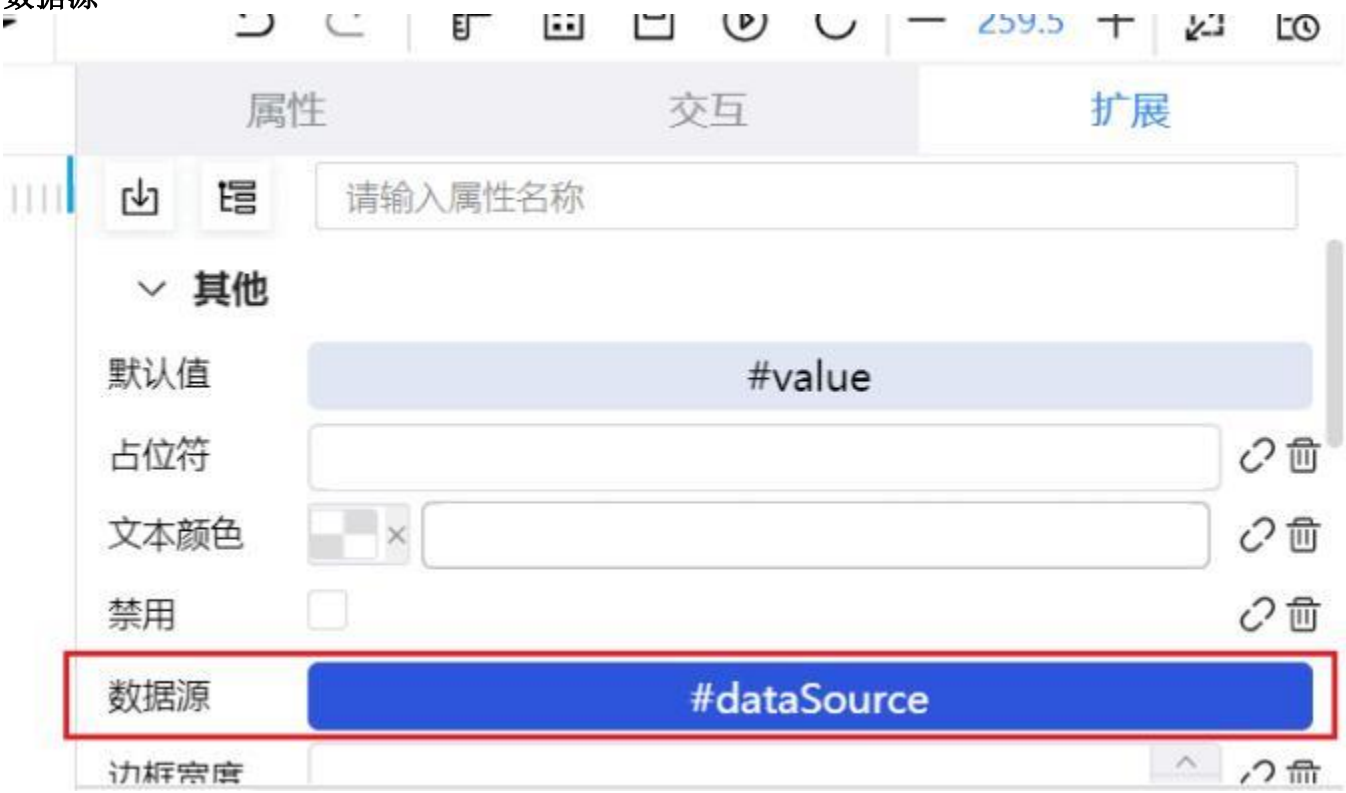
在画面中选中图元后，可自行拖拽成目标大小



配置

在画面中选中图元之后，在右侧会有一个【扩展】tab，可以在此处对多层级下拉框进行配置。具体支持以下配置

数据源



参考如下配置

```
{
"childrenTitle": "区域",
    "children": [
        {
            "label": "华南",
            "childrenTitle": "公司",
            "children": [
                {
                    "label": "厦门总公司",
                    "childrenTitle": "人员",
                    "children": [
                        {
                            "label": "A"
                        },
                        {
                            "label": "B"
                        },
                        {
                            "label": "C"
                        }
                    ]
                }
            ]
        },
        {
            "label": "华北",
            "childrenTitle": "公司",
            "children": [
                {
                    "label": "天津分公司",
                    "childrenTitle": "人员",
                    "children": [
                        {
                            "label": "D"
                        },
                        {
                            "label": "E"
                        }
                    ]
                }
            ]
        }
    ]
}
```

配置如上的数据源后，在页面上对应的展示如下：

其中

childrenTitle:这一层级的名称；

label:这个选项的名称

children:这个选项下下一层级的信息(可选)

默认值

可以设置该下拉框默认选中的值。例如上面的例子中，如果设定

```
[
{
"label": "华北"
},
{
"label": "天津分公司"
},
{
"label": "E"
}
]
```

那么打开页面时，会默认选中这个选项，如下图所示

占位符

即输入框区域的placeholder，例如占位符设置为【区域】，那么在没有选中任意一项的时候输入框区域展示为【区域】

文本颜色

即输入框区域的placeholder，例如占位符设置为【区域】，那么在没有选中任意一项的时候输入框区域展示为【区域】

文本对齐（8.29新增）

可以选择输入框处文本对齐方式，支持左对齐，居中和右对齐

![多层级下拉框图元](../../assets/BA_assets/iMage/07组态设计5.0/03图元和控件/image20250814-295-多层级下拉框图元13.5.png)

禁用

勾选之后，下拉框处于不能交互状态。

边框宽度

可更改边框的宽度

边框颜色

边框颜色

可以更改边框的颜色

边框半径

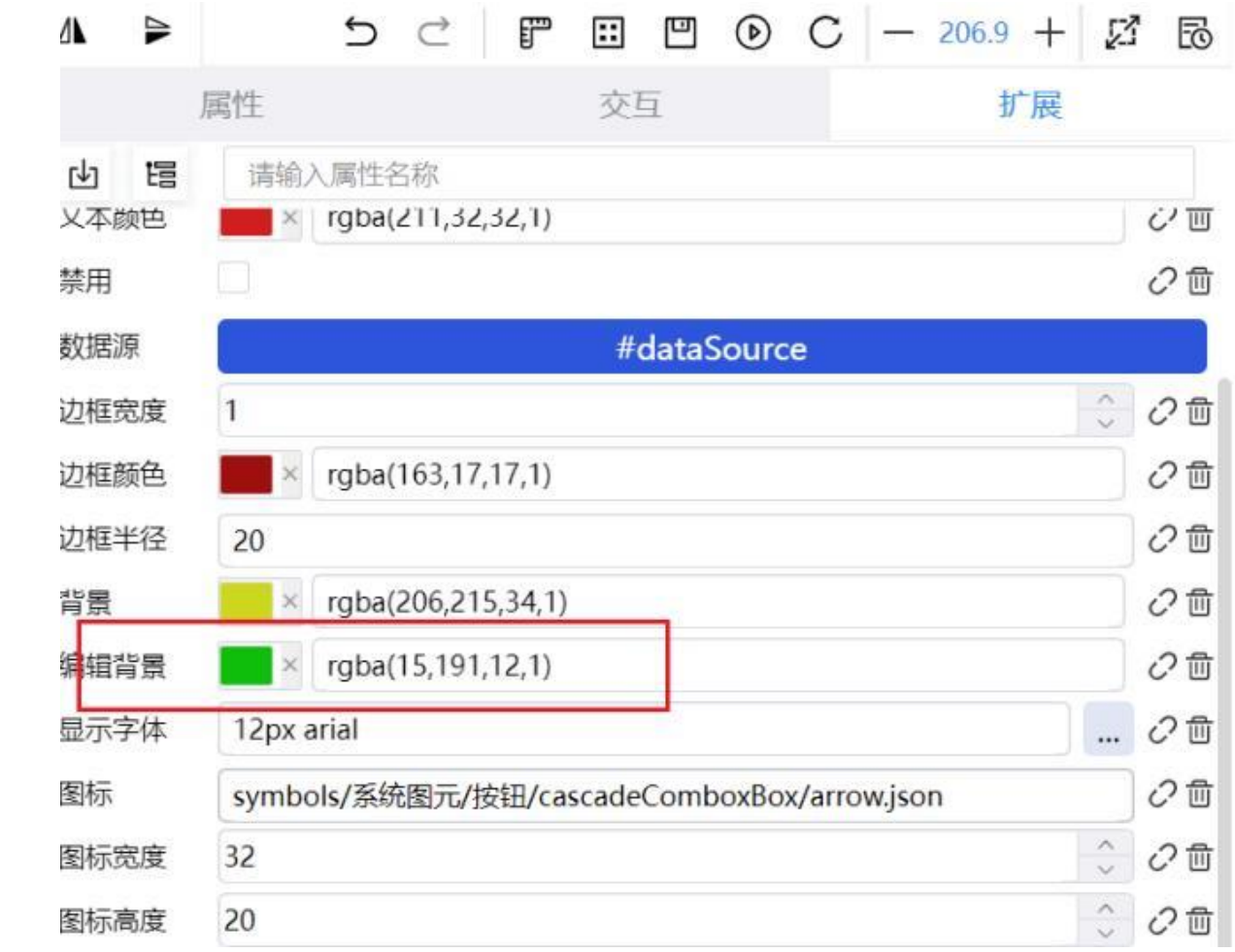
可以更改边框的圆角大小

背景

可以更改输入框背景色

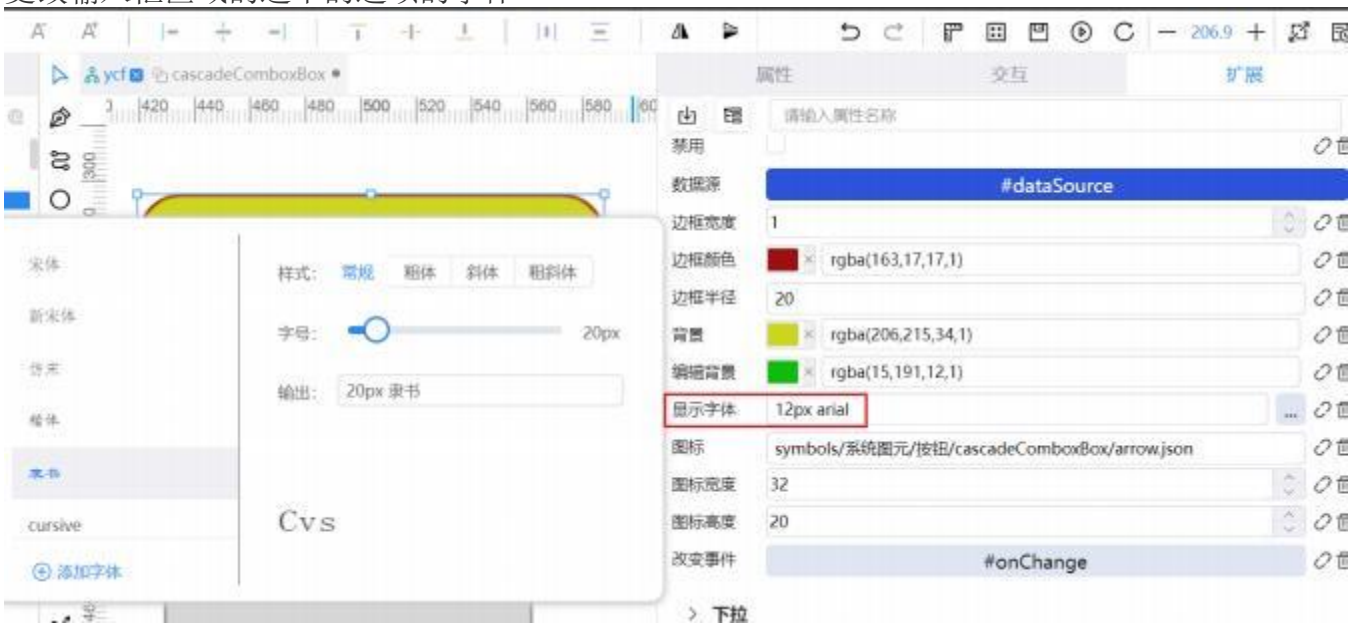
编辑背景

可以更改下拉面板展开时，也就是编辑态的背景色



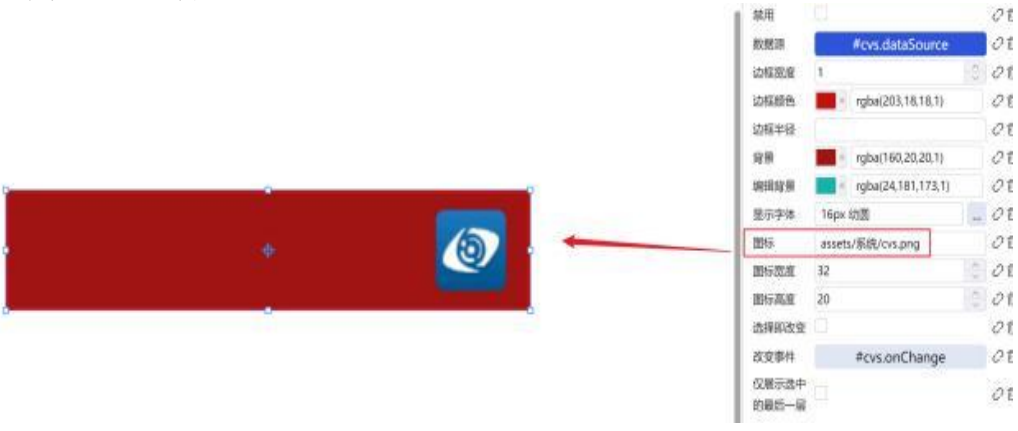
显示字体

更改输入框区域的选中的选项的字体



图标

可以指定图标的路径，可以替换对应的下拉图标

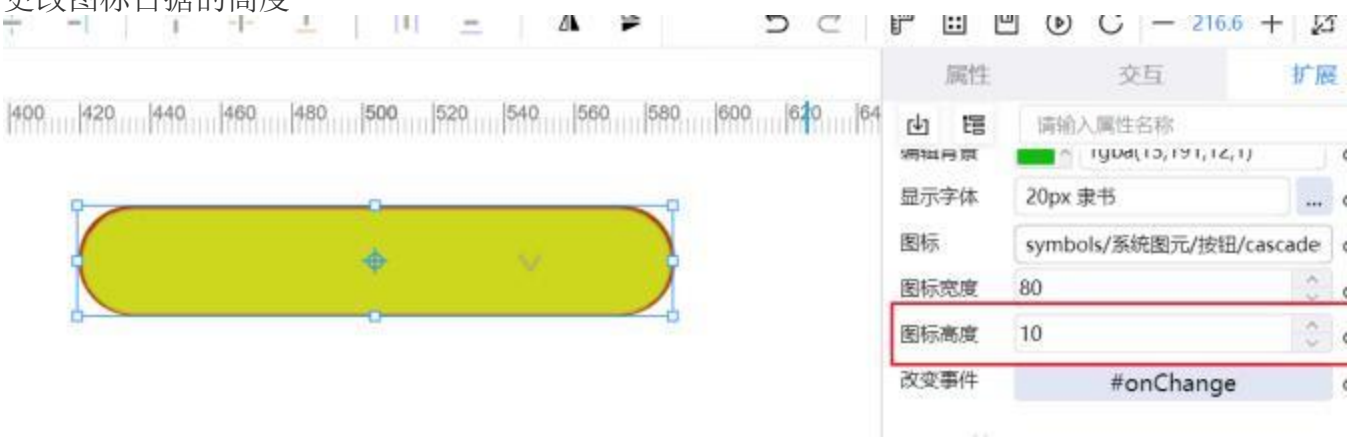


图标宽度

更改图标占据的宽度

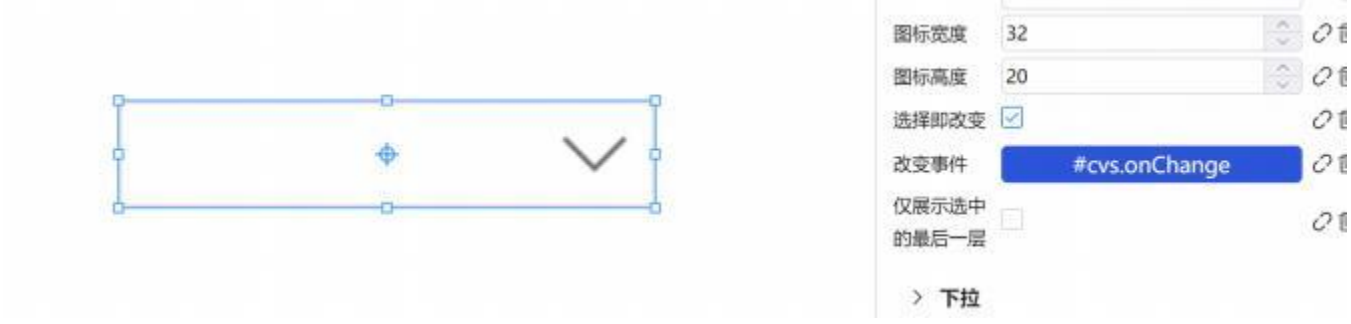
图标高度

更改图标占据的高度



选择即改变（8.29新增）

用户选择某一层之后，立刻生效，在输入框会展示当前层的选中值，



选中时VS未选中时

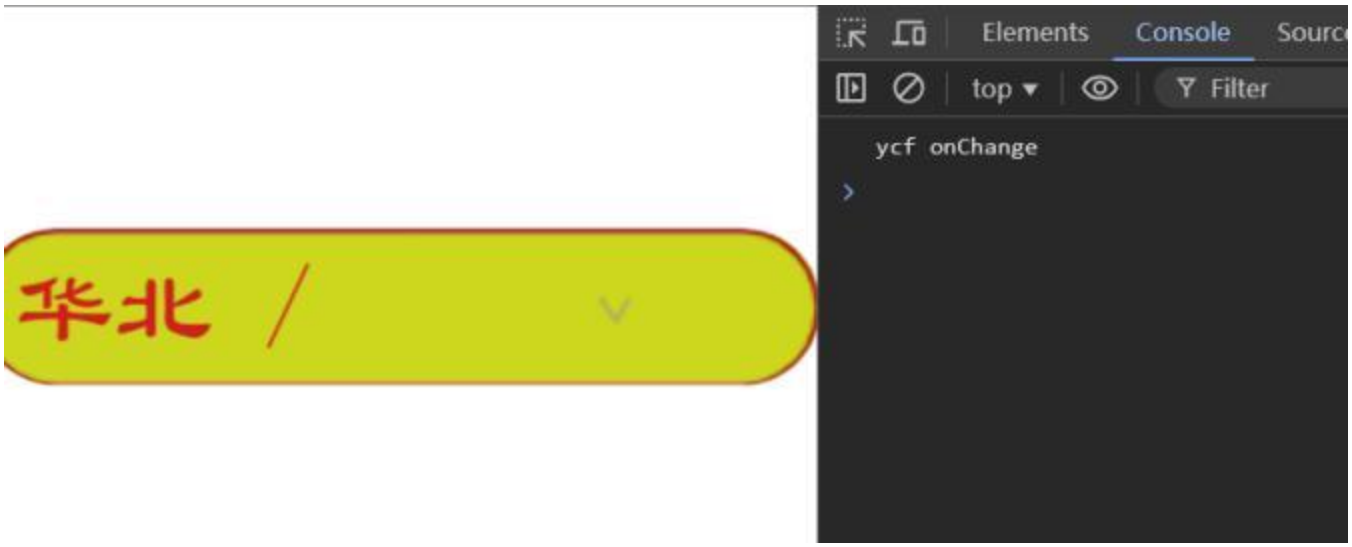
仅展示选中的最后一层（8.29新增）
用户可以控制，输入框区域是展示所选择的所有路径的值，还是仅展示最后一层的值

勾选时VS未勾选时

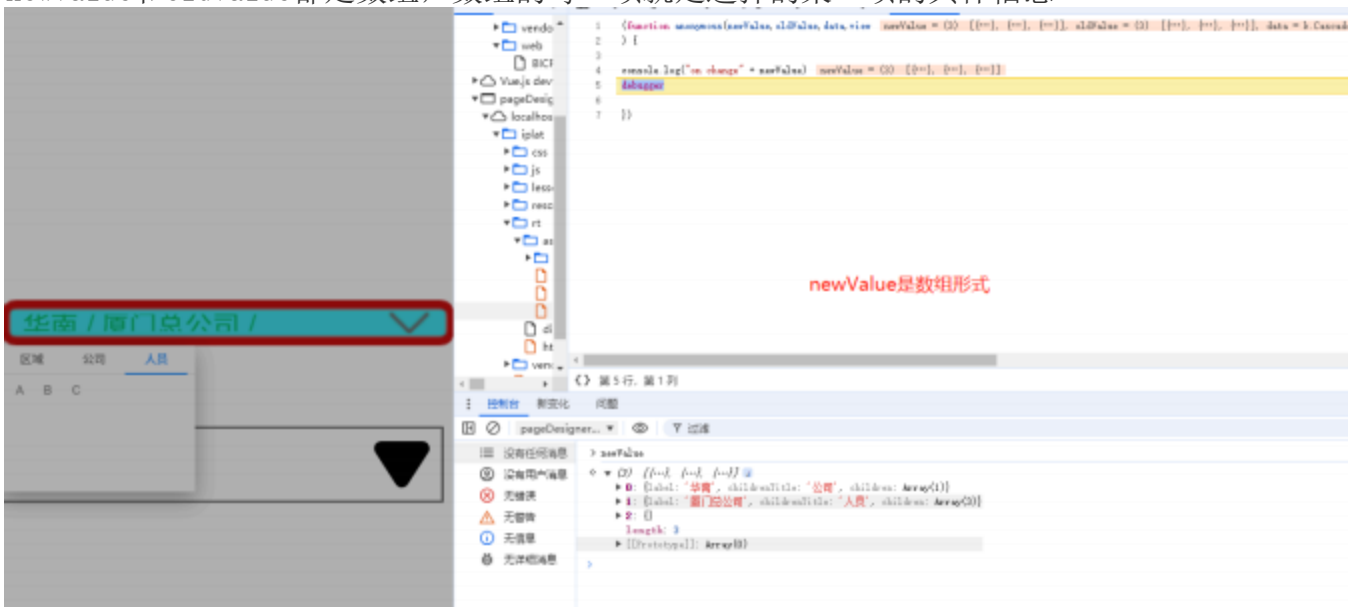
改变事件

选中的值更改时，触发的回调函数

```
function (oldValue,newValue,data,view) {
    1    console.log('ycf onChange')
```

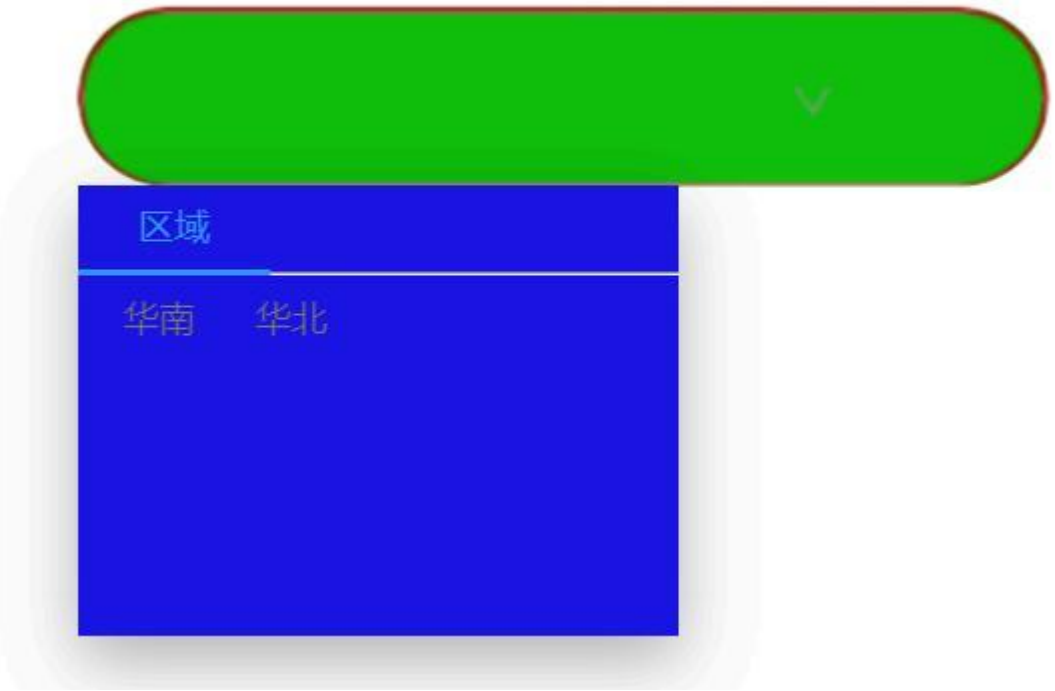


newValue和oldValue都是数组，数组的每一项就是选择的某一项的具体信息



下拉框背景

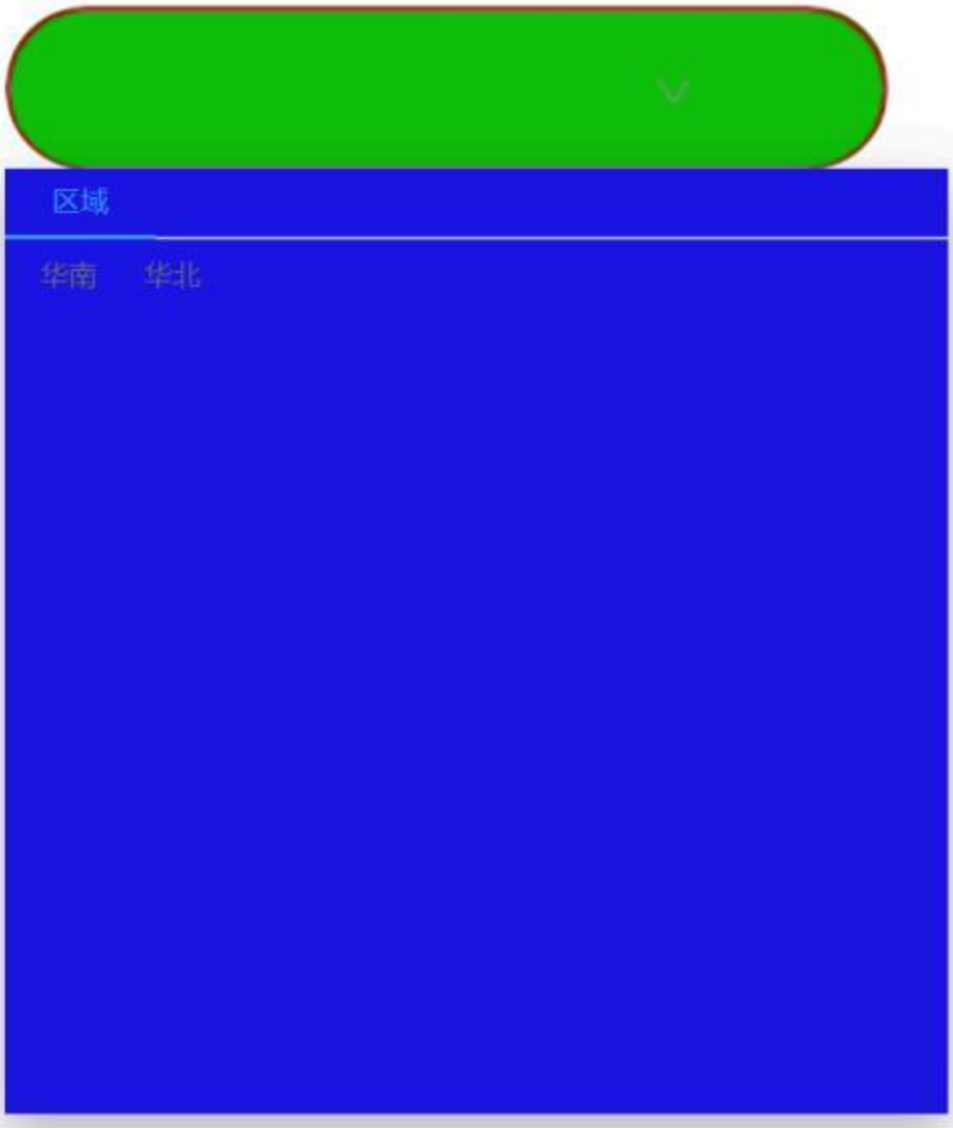
更改下拉框面板的背景色



下拉框宽度

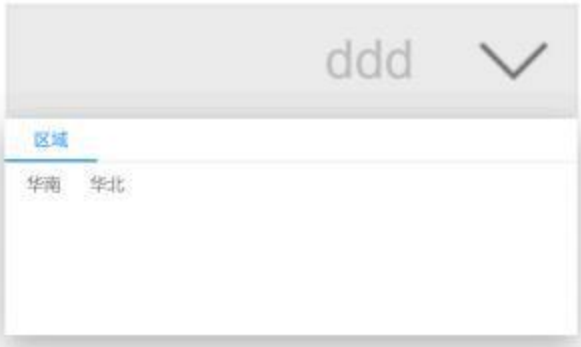
更改下拉框的宽度

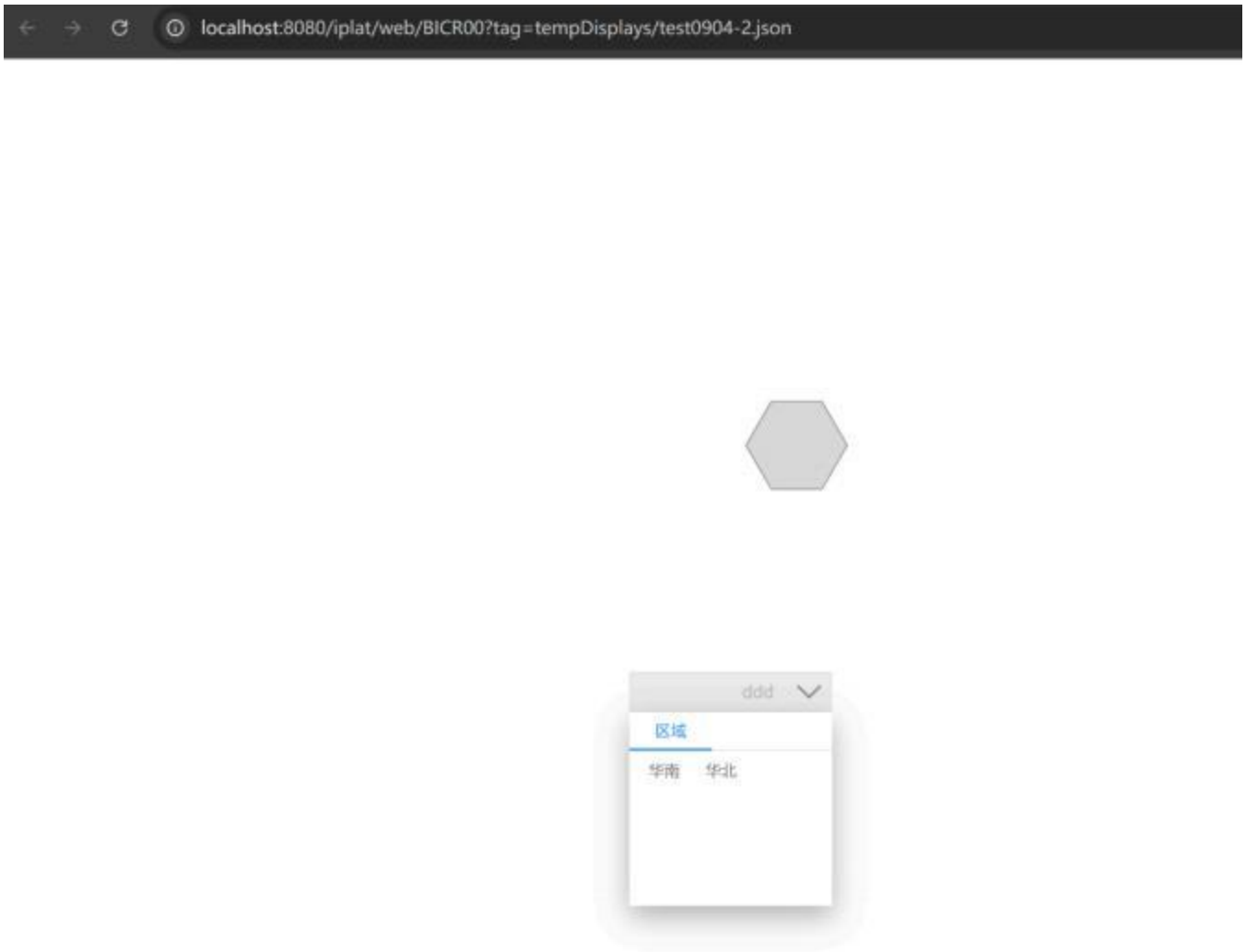
下拉框高度
更改下拉框的高度



下拉框宽度自适配（8.29新增）

选中后，下拉框设置的宽度失效。下拉宽度与上部的输入框宽度保持一致。





图元精灵配置（8.29新增）

上述配置为画面级别的配置，用户可以在画面上针对某一个特定的多层级下拉框进行配置。但是有些情况下，用户可能希望所有的多层级下拉框都有相同的配置，比如用户想统一更换一下下拉框的图标。因此本次增加一个图元精灵级别的配置。

入口

选中多层级下拉框图元之后，在右侧有一个【拓展】，拓展内可进行配置。配置项与上一节中的配置相同，但是没有【改变事件】，也不支持绑点。

优先级

画面上针对图元进行的配置优先级高于图元上针对图元精灵的配置

5.38 高性能趋势

编辑态

基本属性

该部分配置功能与文本趋势的运行态系统配置功能，基本保持一致
1.属性显隐
1.1数据点突出：
1.1.1【启用】画面中的数据点会以更加显眼的方式展示，但如果点数足够多的时候，该功能会自动关闭
1.2启动tooltip插值：
1.2.1【启用】tooltip的内容不会锁定曲线上的点。tooltip会通过插值法计算出两点之间的曲线内容。
1.2.2【不启用】tooltip会吸附到附近展示的点上。
1.3曲线名称隐藏节点名：
1.3.1【启用】曲线名称的节点名称会被隐藏
1.3.2【不启用】
2.其他配置
3.历史趋势配置
3.1数据比例，当前数据占据画面的比例（一定要确保服务端和客户端的时间是对齐的）
3.2X轴时间：时间跨度：查询的时间范围
时间范围：查询具体时间段的趋势
值得注意的点：
①显示tooltip和现实表格，可以在运行态，通过按钮动态设置。
②现实tooltip的前提，是tag点需要具有描述内容。

Y轴

1.显示Y轴标题
勾选则显示Y轴名称，否则隐藏。
2.Y轴样式分布
基于Y轴样式分布可以确定，Y轴的分布情况。
均匀分布在两侧是以左右左右的模式进行分布。
3.使用Y轴工程量程
如果tag点配置了工程量程，则会使用该量程作为Y轴量程。注意：如果一条Y轴上有多条曲线。则该Y轴的量程为多条曲线量程的并集。
4.启用上下边缘留白、Y轴自适应时上下边缘留白各占比（%）
在启用上下边缘留白开启后，Y轴自适应时上下边缘留白各占比（%）才会生效，会对Y轴的界限进行扩大（仅限于自适应的情况下）效果对比如下：左侧为不开启，右侧为开始Y轴自适应时上下边缘留白各占比30%的效果。
5.Y轴小数位数
为Y轴设置小数点位数，如上图，左侧小点位数为2，右侧小点位数为3。
6.新增Y轴&修改Y轴触发位置：
点击后都会出现如下的dialog，以修改为例进行内容展示：
7. 阈值线配置
打开方式：Y轴→新增Or修改Y轴→新增

曲线

•1.主要配置内容
曲线配置的内容对应如下图所示
•2.统计值线：最大值、最小值、平均值线
这些统计值线，需要计算统计值后才会出现。计算统计值的操作如下：
。（1）暂停
。（2）显隐藏标线
。（3）计算统计值
•3.分组

每个分组可以认为是一套单独的曲线组（可以用于切换）
新增、编辑组：
编辑态的展示效果如下：

表格栏配置

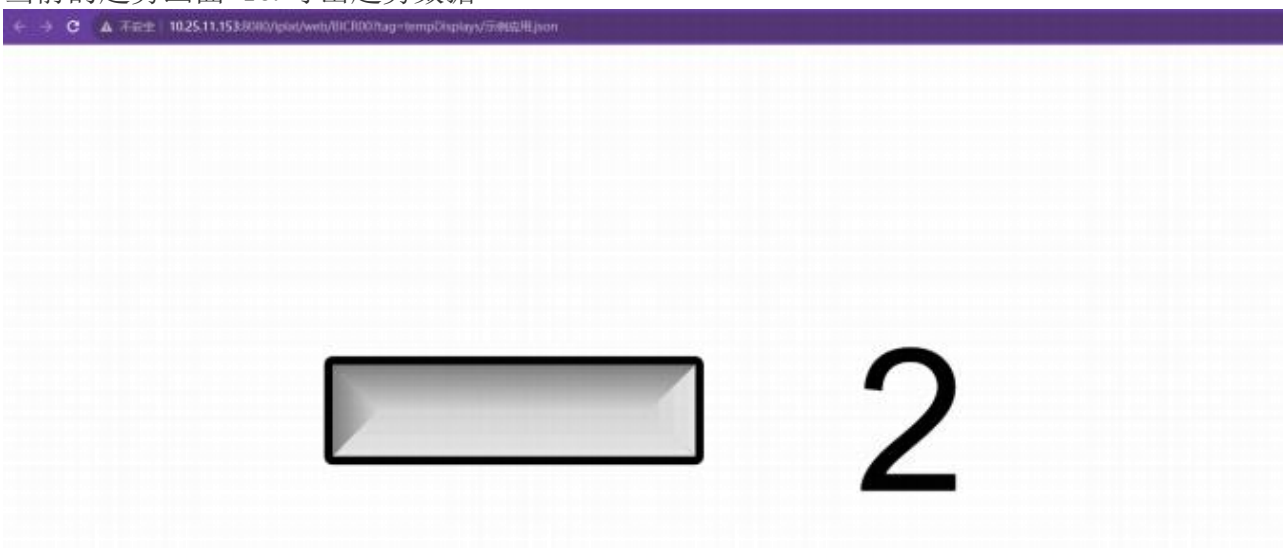
（1）选择表格中展示的列及其宽度
（2）统计值小数位数：用于设置表格中的统计值的小数点位数

运行态

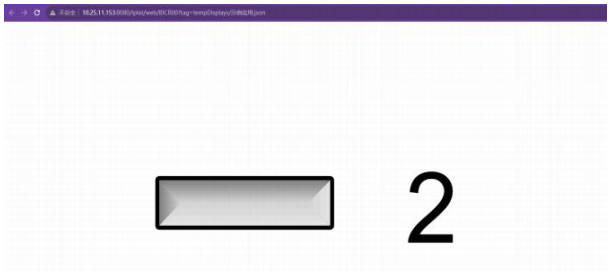
工具栏

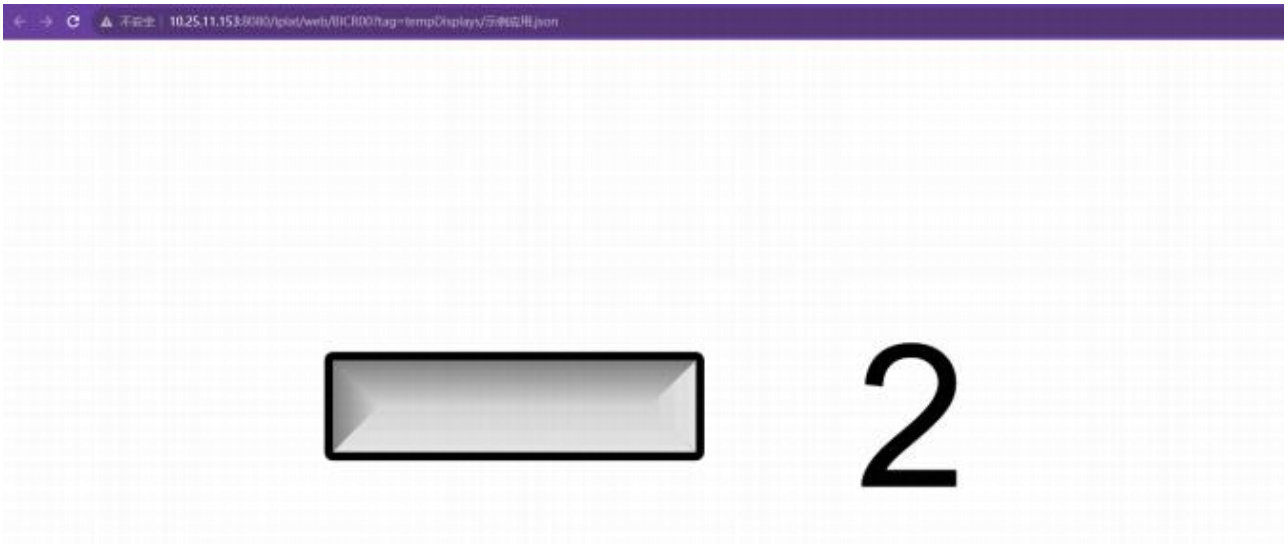
•1.向前翻页
向前翻一个X轴时间跨度•2.切换至暂停/实时
若当前状态为实时状态，点击后处于暂停状态。若当前状态处于暂停状态，点击后切换至实时状态
•3.向后翻页
向后翻一个X轴时间跨度
•4.翻页时间长度
设置X轴时间跨度（会即刻生效）
•5.历史采样间隔
设置历史数据采样的间隔，以下是5s（左）和20s（右）的采样间隔对比图•6.显隐藏标线
初始点击后会执行如下操作：
◦（1）画面切换至暂停
◦（2）产生两条实线标线
也可以通过鼠标右键，添加虚线的标线（最多6条虚标线、2条实虚线），此时的显隐藏标线能够显示隐藏所有标线
•7.计算统计值
在显示标线的情况下，可以点击计算统计值，之后便会基于编辑态的配置来展示统计值（该统计值为两条实现标线之间的数据统计）
•8.区域缩放重置
将当前的缩放状态还原为默认状态•9.显隐ToolTip
默认开启，关闭后画面中则不展示tooltip•10.添加曲线
在当前展示的趋势图中添加曲线
•11.分组功能
高性能趋势中，涉及到分组功能的区域如下图所示
•（1）分组按钮
位置在工具栏，作用是打开曲线分组配置界面，该界面如下图所示
◦（1.1）"选择分组"下拉框
▪通过配置该下拉框的内容吗，可以观察不同组别下的曲线配置
▪下拉框的选项包含组别名称和组别描述，组别描述在选项的括号中
▪支持过滤搜索，方法是在输入框中进行模糊匹配
◦（1.2）曲线配置表格
▪该表格的内容，与右键趋势打开的运行时配置中，曲线配置tab页的内容一致
▪修改该表格的内容后，点击确认按钮才可以生效
▪可以使用该表格修改所有曲线的组别配置（譬如修改了"组别a"的配置，然后下拉框切换到"组别b"，点击确定。此时趋势虽然会显示"组别b"，但其实"组别a"的配置修改，也已经被保存了）
•（2）切换分组按钮（用于快捷切换分组）
◦位置在表格左上方
◦点击后，出现一个下拉框，包含所有组别名称和组别描述（选项的括号里的是组别描述）
◦再点击下拉框中的选项，即可快捷切换分组
•（3）当前组别信息

◦在表格左下角显示当前组别的名称
◦使用工具栏的"分组"按钮修改分组配置，并保存后；以及使用表格旁的"切换分组"按钮切换了组别以后，组别的名称信息会立刻更新
•12.强制原始值
如果当前数据为插值的时候，可以通过强制原始值将插值的内容替换为原始值内容•13.重新加载数据
重新加载一次当前画面的数据（注，如果原本是插值，且使用了强制原始值，那么此次重新加载将加载插值）
•14.保存功能
功能按钮位置如下图所示。高性能趋势的保存功能与文本趋势基本相同，同样是保存曲线小数位数与翻页时间长度。但在一些细节上有所不同
配置的存储机制
•会保存所有组别的所有曲线的配置中的"启用小数"和"小数位数"列
•没有存储曲线条数上限
•每个画面中的每个趋势的配置都会独立存储（签入画面和签出的画面存储的配置目前不通用，算作两个配置）
•15.打印
打印当前的趋势画面•16.导出趋势数据

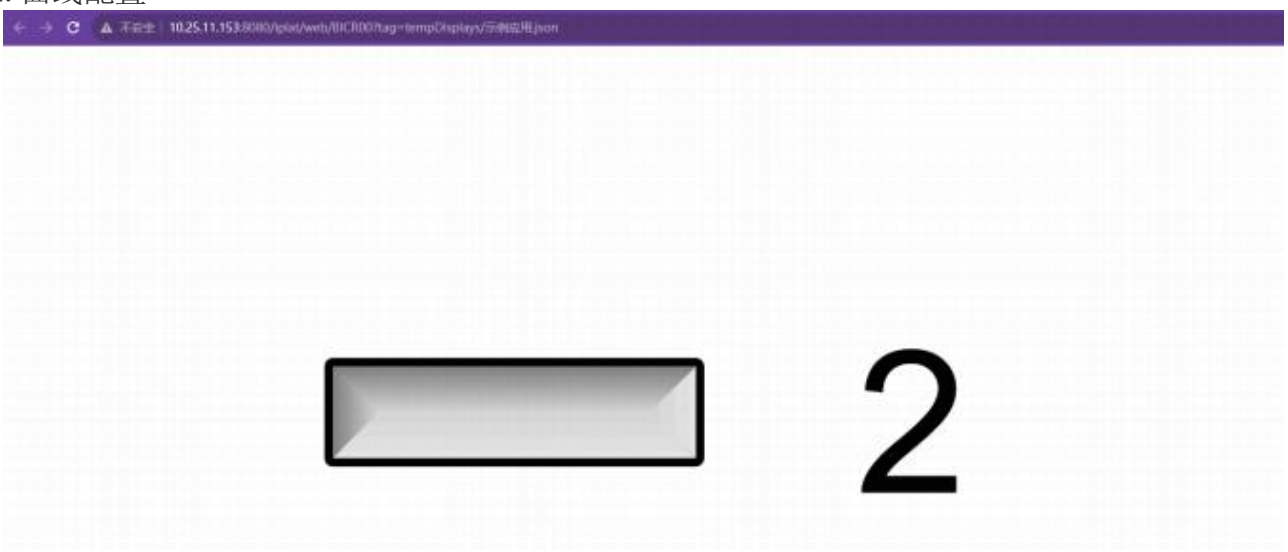


导出当前趋势图中的数据（插值则导出的是插值，原始值则导出的是原始值）
运行时配置
1.趋势图中通过鼠标右键打开运行时配置

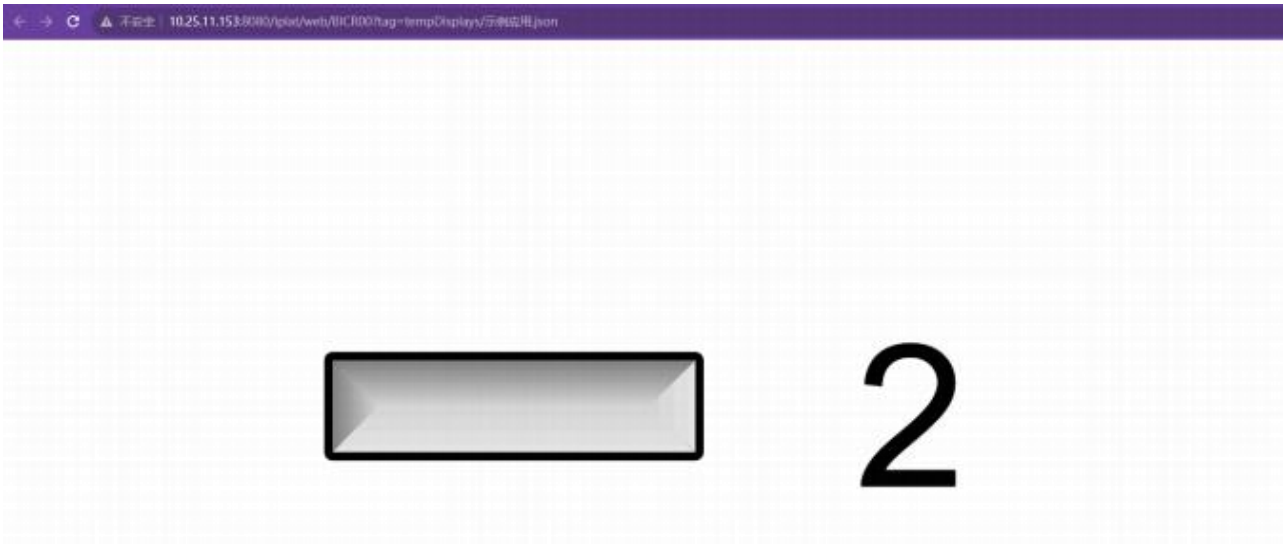




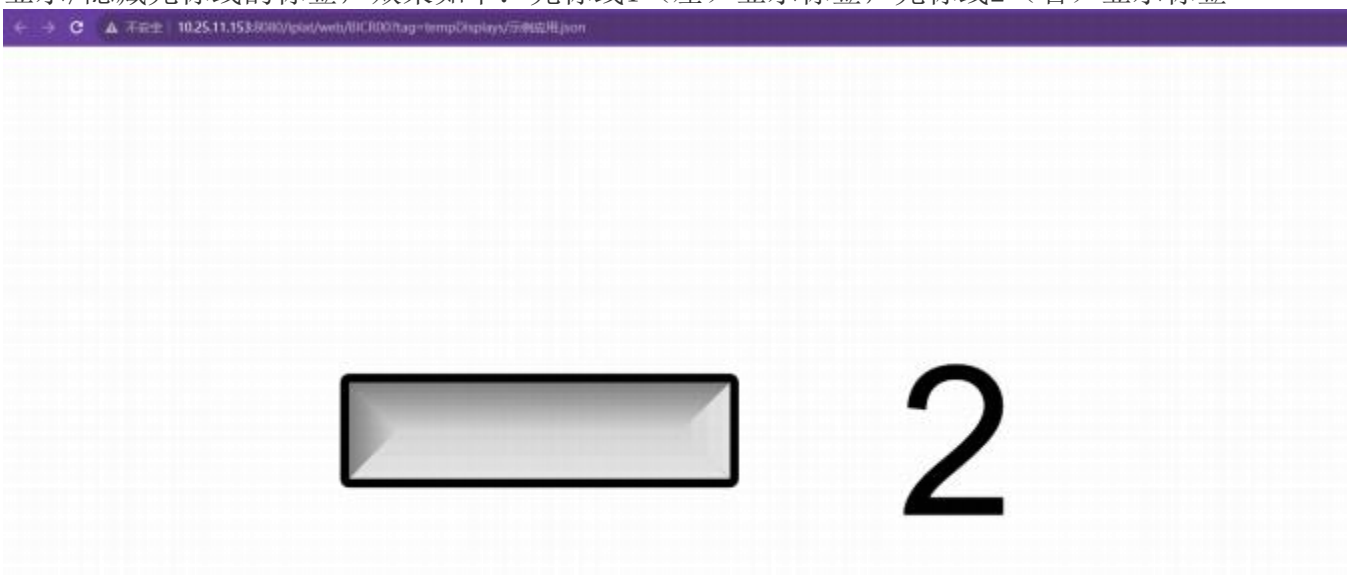
◦（1）Y轴的颜色和名称无法修改
◦（2）启用位数、小数位数：配置Y轴上刻度的小数点位数及是否启用
◦（3）启用量程、量程：配置Y轴的量程及是否使用
•2.曲线配置



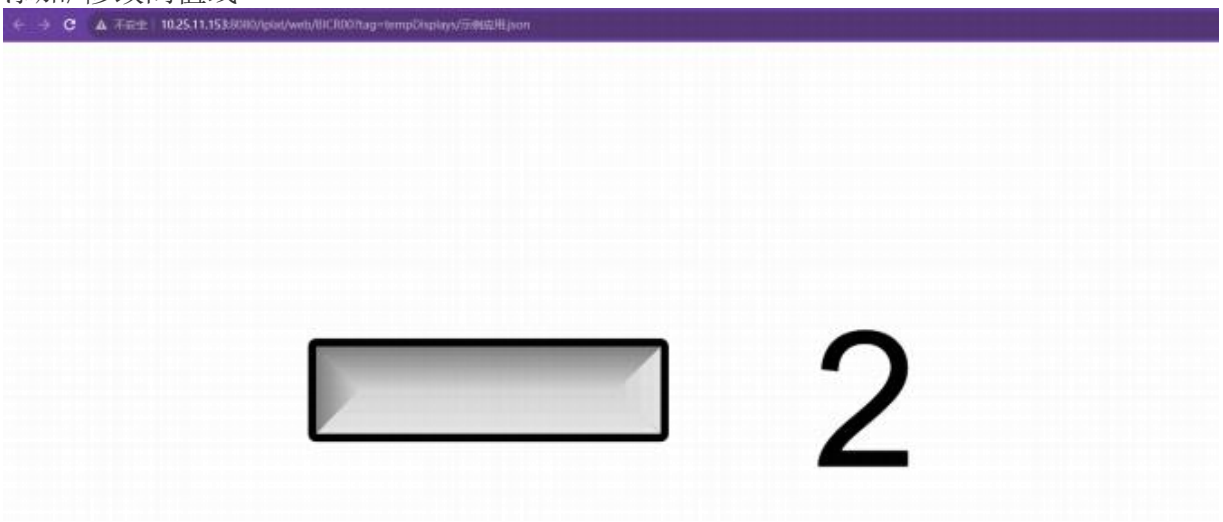
◦（1）选中：显示/隐藏曲线
◦（2）操作：删除曲线
◦（3）曲线名称：此处可以修改曲线的颜色、名称
◦（4）曲线变量：此处可以修改曲线的变量
◦（5）Y轴：此处可以切换Y轴
•3.标线配置



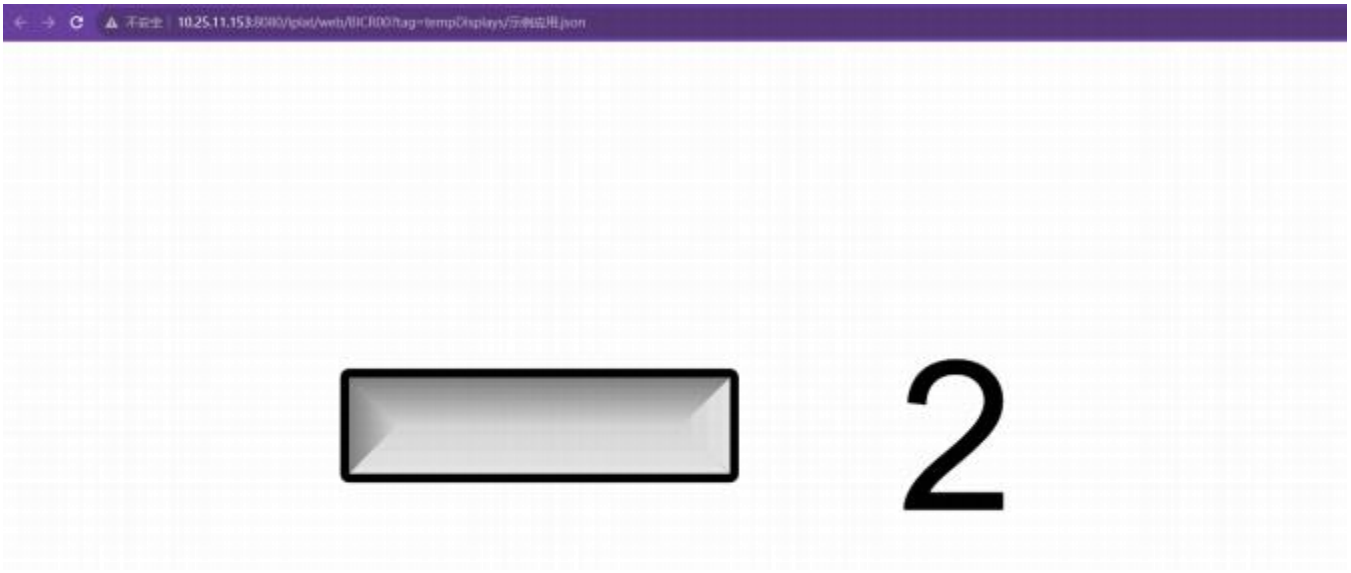
显示/隐藏光标线的标签，效果如下：光标线1（左）显示标签，光标线2（右）显示标签



- 4. 阈值线
  - 添加/修改阈值线

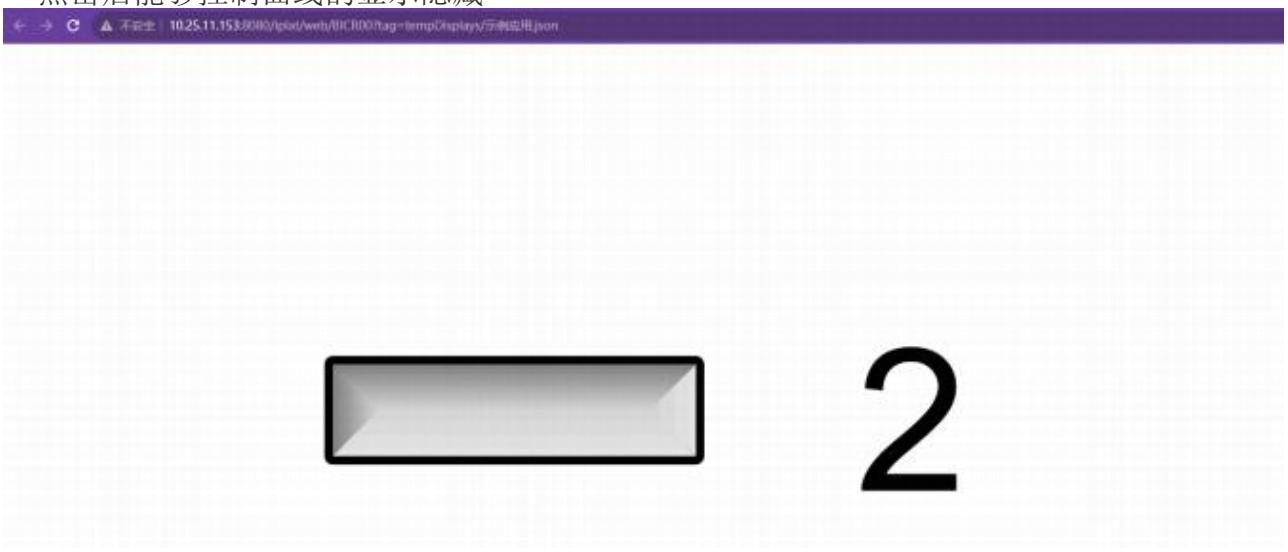


图表区域的功能



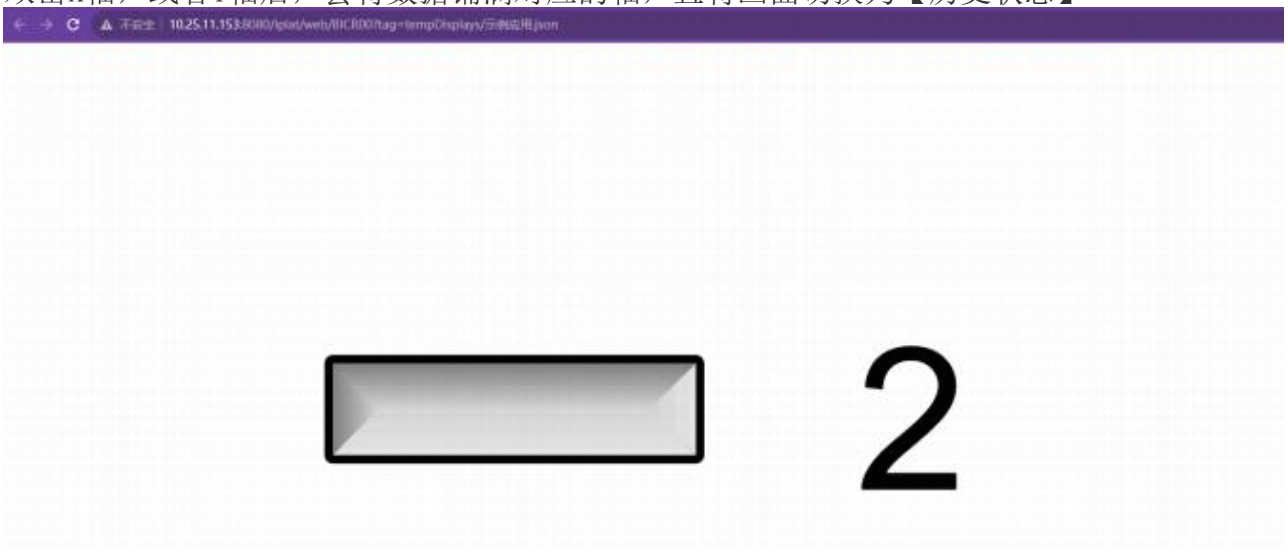
•1图标legend

点击后能够控制曲线的显示隐藏

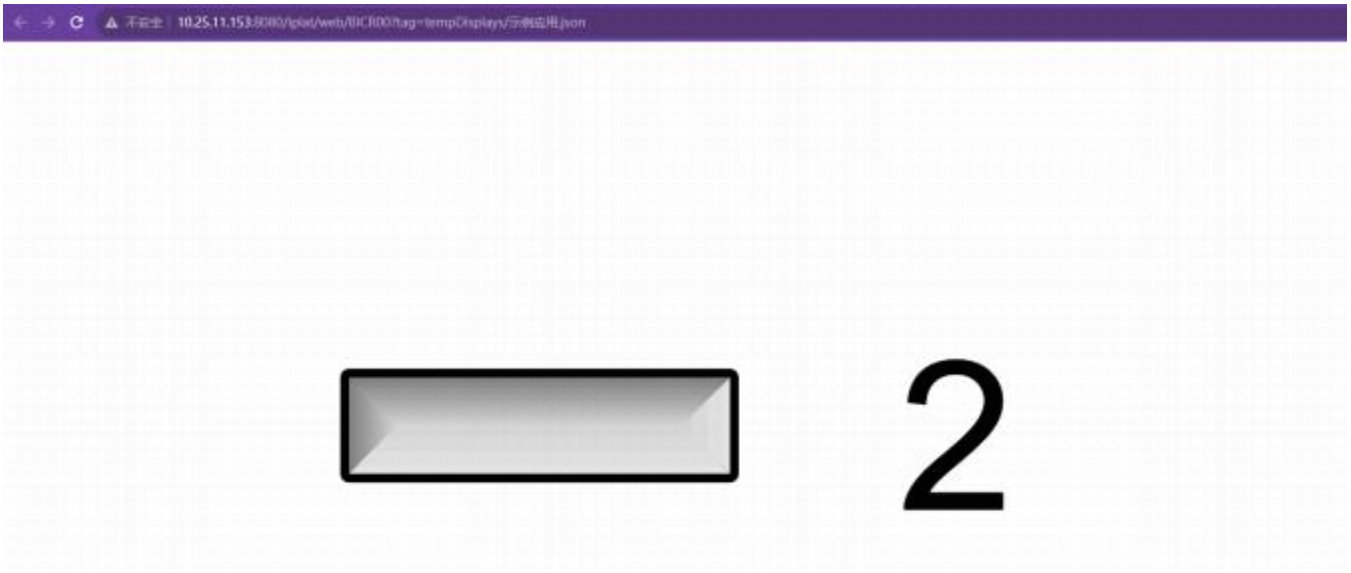


•2. 双击轴

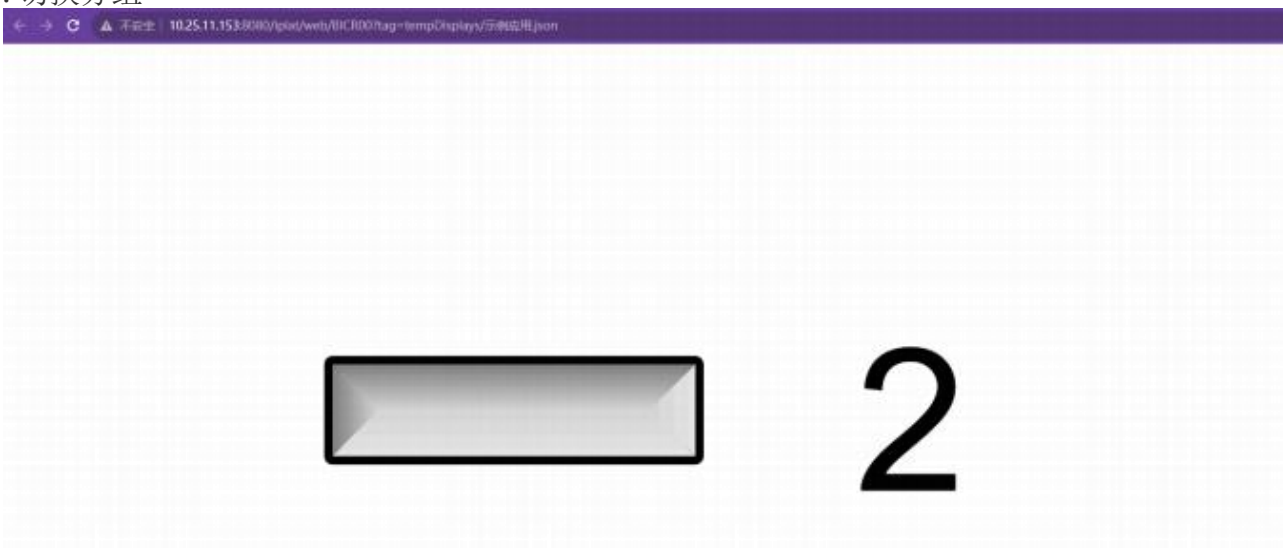
双击X轴，或者Y轴后，会将数据铺满对应的轴，且将画面切换为【历史状态】



表格区域的功能

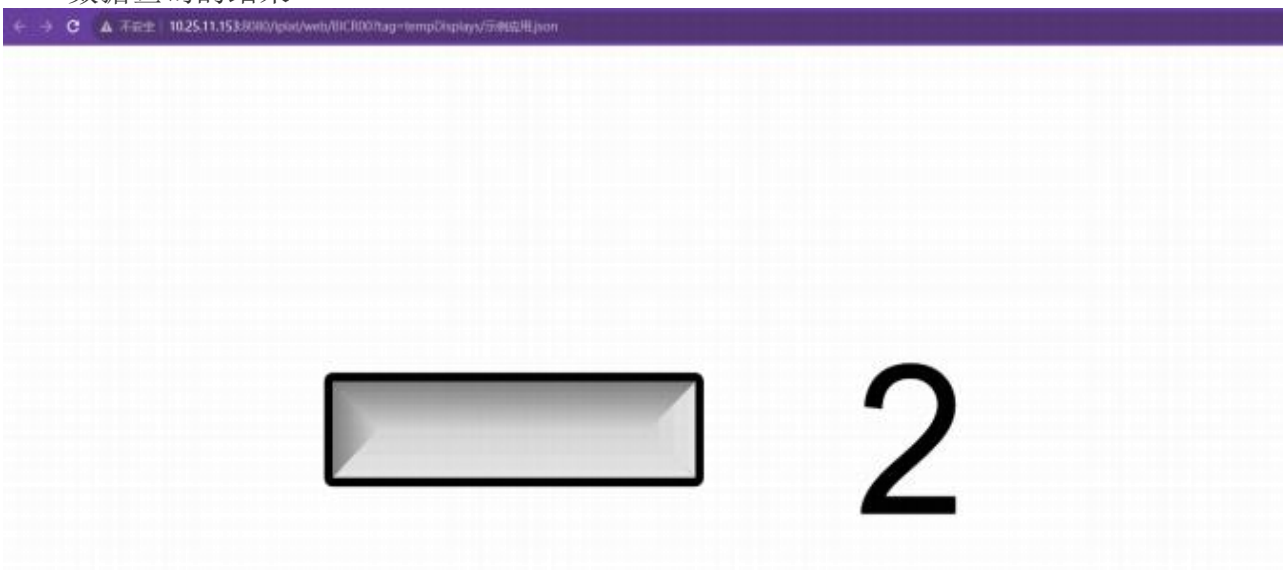


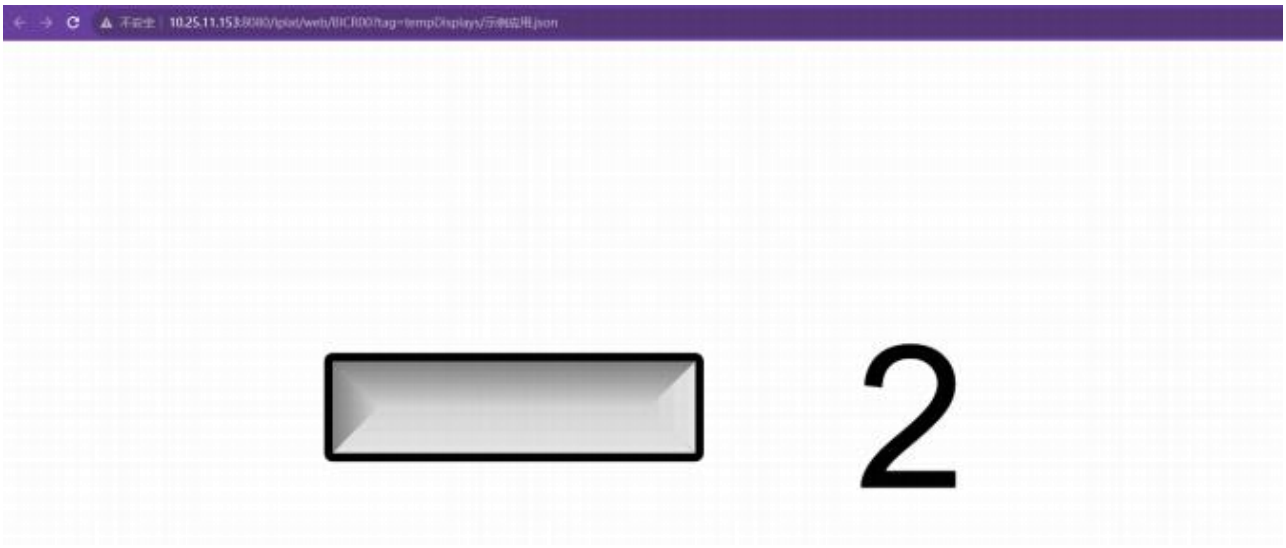
•1. 切换分组



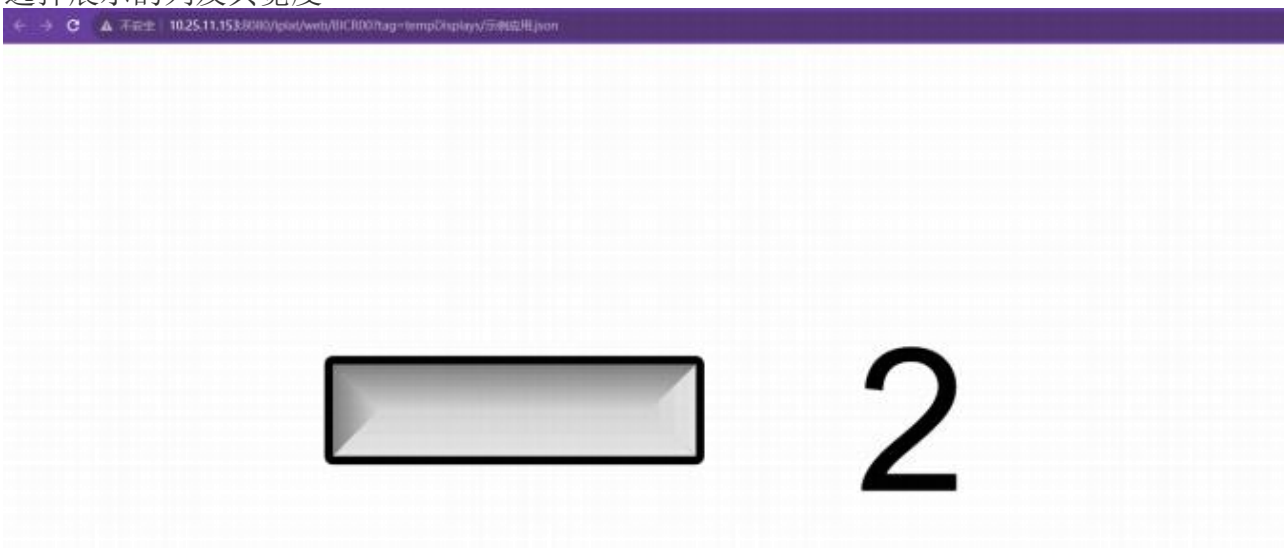
•2. 查询结果

数据查询的结果

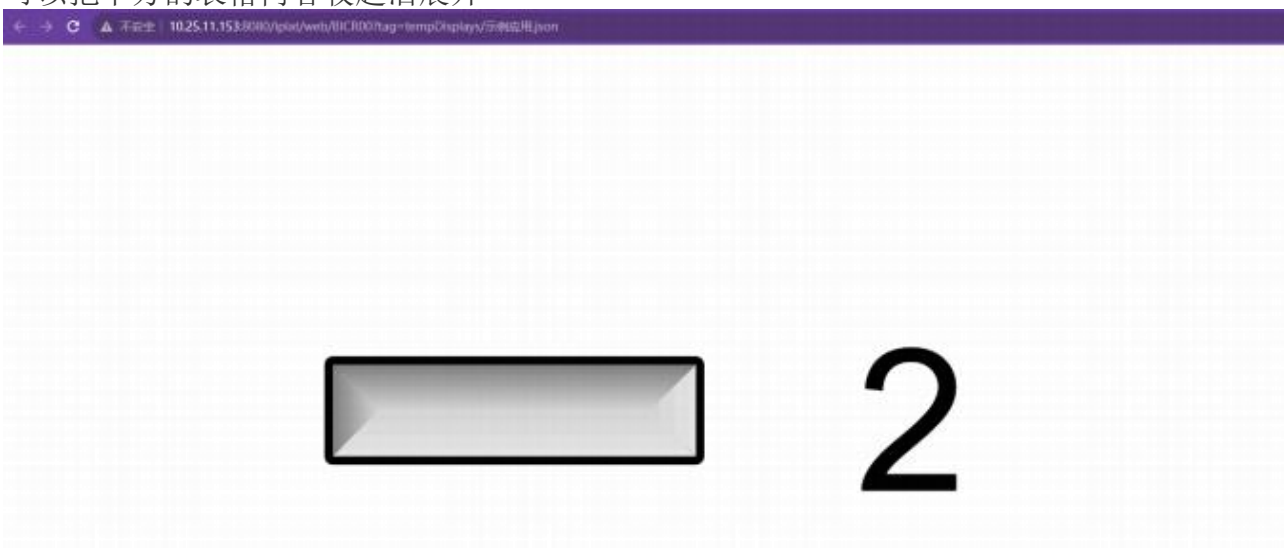




•3. 表格列配置
选择展示的列及其宽度



•4. 表格折叠
可以把下方的表格内容收起活展开



•5. 选中
显示/隐藏曲线
•6. 操作
删除曲线
•7. 曲线名称

点击颜色快，能够修改曲线的颜色

- 8. 其他

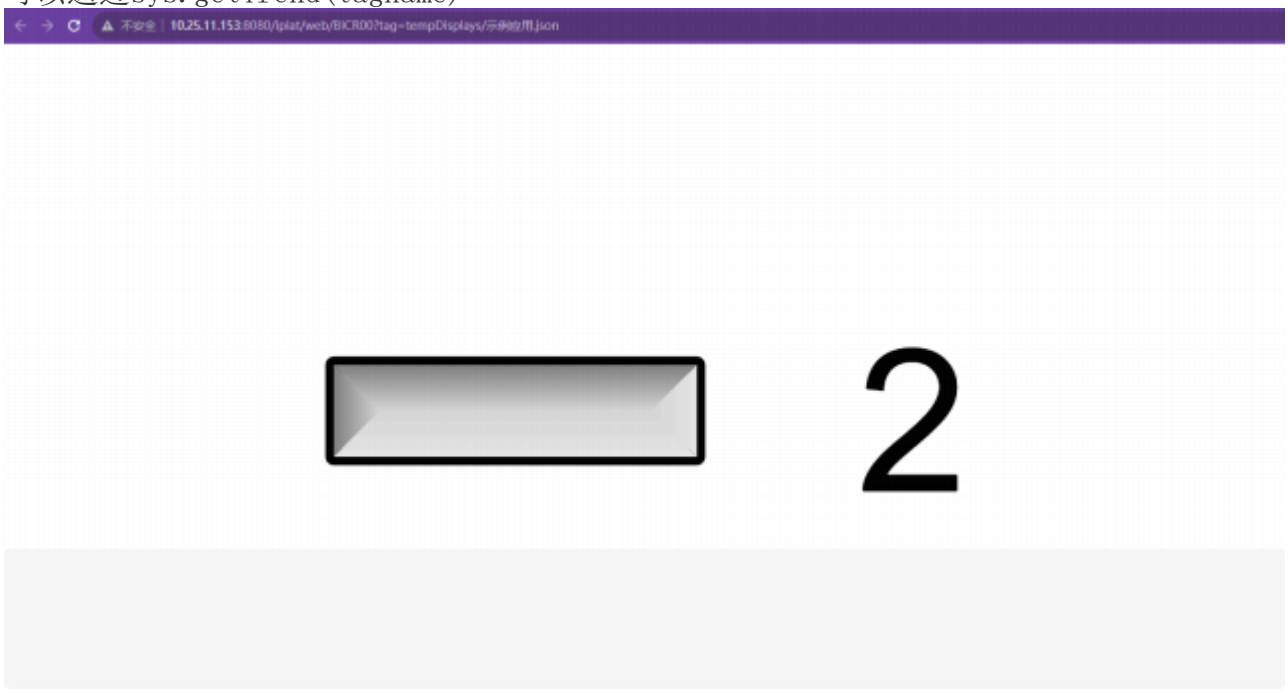
其他表格栏仅用于数据展示

系统函数

- 1. 获取高性能趋势对象

要使用高性能趋势的系统函数，首先得获取高性能趋势对象。

可以通过sys.getTrend(tagname)



注：趋势对象的获取，需要在趋势组件加载完毕之后。所以不能在page_onLoad函数中使用。推荐在page_onContentLoaded中进行使用。

- 2. 添加曲线

获取到高性能趋势的对象之后，可以通过调用添加曲线函数trend.addCurvesAsync(group,lines)，来向趋势图中添加曲线。

```
const lines = [{ name: "OPCUA_FRamp3", tag: "Windows_Single_Build.OPCUA_FRamp3"}];
const callback = ()=```{
console.log("添加曲线回调函数测试")
}
trend.addCurvesAsync("default", lines,callback)
```

参数	数据类型	备注
gropu	string[]	指定曲线所在的组
lines	Line	需要添加的曲线列表
callback	function	回调函数（非必填）

Line对象

参数	数据类型	备注
tag	string	点名
name	string	曲线名

注：高性能趋势的添加曲线系统函数为异步函数，可以在回调中处理，全部曲线添加结束后的逻辑。

•3.删除曲线

获取到高性能趋势的对象之后，可以通过调用添加曲线函数

trend.deleteCurve(group,curveName)，来删除趋势图中存在的曲线。

```
trend.deleteCurve("default", 'OPCUA_FRamp3')
```

参数	数据类型	备注
gropu	string	指定曲线所在的组
curveName	string	需要删除的曲线名称

•4.隐藏按钮

获取到高性能趋势的对象之后，可以通过调用隐藏按钮函数

trend.setButtonDisabled(buttonList)，来隐藏趋势控件中的按钮。

只有在工具栏显示的时候，该函数才生效

```
let buttonList=['myMidLineShow']
trend.setButtonDisabled(buttonList);
```

参数	数据类型	备注
buttonList	string[]	需要隐藏的按钮列表

按钮及名称

参数	数据类型
向前翻页	myToolPageForward
暂停	myToolSwitchToRealMode
向后翻页	myToolPageBack
翻页时间长度	myToolPageTurnLength
时间选择	myToolSelectTime
历史采样间隔	myToolHisTimeGap
标线	myMidiLineShow
统计值计算	myStatistic
缩放区域重置	myRestore
tooltip显隐	myShowOrHideToolTip
添加曲线	myAddSeries
分组	mySelectGroup
切换Y轴纵横排布	myChangeYDivision
查询原始值	myQueryRaw
刷新	myToolRangeConfiguration
保存在本地	mySaveToIndexedDB
打印趋势	myPrint
导出趋势数据	myExportData

•5.显示按钮

获取到高性能趋势的对象之后，可以通过调用隐藏按钮函数

trend.setButtonAvailabled(buttonList)，来显示趋势控件中的按钮。

只有在工具栏显示的时候，该函数才生效

```
let buttonList = ['myMidLineShow']
trend.setButtonAvailabled(buttonList);
```

参数	数据类型	备注
buttonList	string[]	需要隐藏的按钮列表

按钮及名称见上表

•6.切换分组

获取到高性能趋势的对象之后，可以通过调用trend.setGroup(groupName)来切换当前显示分组。

trend.setGroup("测试分组")

参数	数据类型	备注
groupName	string	需要显示的分组名称

•7.表格样式

获取到高性能趋势的对象之后，可以通过调用trend.setTrendGridStyle(options)来设置趋势表格样式。

```
trend.setTrendGridStyle({
headerBackgroundColor:"red",
bodyBackgroundColor:"rgba(1,2,3,0.4)"
});
```

参数	数据类型	备注
options	StyleObject	StyleObject对象
headerBackgroundColor	string	表头背景色
bodyBackgroundColor	string	表体背景色

•8.向前翻页

获取到高性能趋势的对象之后，可以通过调用trend.prePage()，来设置历史趋势向前翻页。

```
trend.prePage();
```

•9.向后翻页

获取到高性能趋势的对象之后，可以通过调用trend.nextPage()，来设置历史趋势向后翻页。

```
trend.nextPage();
```

•10.隐藏曲线

获取到高性能趋势的对象之后，可以通过调用trend.hideLines(lineNames)，来设置历史趋势的曲线隐藏。

trend.hideLines(['OPCUA_FRamp1','OPCUA_FRamp2']);

参数	数据类型	备注

lineNames	string[]	曲线名称列表

•11.显示曲线

获取到高性能趋势的对象之后，可以通过调用trend.showLines(lineNames)，来设置历史趋势的曲线显示。

trend.showLines(['OPCUA_FRamp1','OPCUA_FRamp2']);

参数	数据类型	备注
lineNames	string[]	曲线名称列表

•12.时间选择

获取到高性能趋势的对象之后，可以通过调用trend.selectTimeRange()，来使用历史趋势时间选择功能。

```
trend.selectTimeRange();
```

•13.翻页长度

获取到高性能趋势的对象之后，可以通过调用trend.setTimeRange(timespan)，来设置历史趋势的翻页长度。timespan={timenumber:10,timeunit:"hour"}

trend.setTimeRange(timespan);

参数	数据类型	备注
timespan	timespanObject	翻页时间长度对象

timespanObject对象

属性	类型	备注
timenumber	number	时间长度
timeunit	string	时间单位

时间单位枚举

名称	类型
秒	second
分	minute
时	hour
天	day
月	month
年	year

•14.指定显示

获取到高性能趋势的对象之后，可以通过调用trend.reSetShowLines([lineName1,lineName2])，来设置历史趋势需要显示的曲线。

trend.resetShowLines(['OPCUA_FRamp1'])

参数	数据类型	备注
lineNames	string[]	曲线名称列表

•15.导出数据

获取到高性能趋势的对象之后，可以通过调用trend.exportData()，来导出趋势数据。

trend.exportData();

5. 数据发布

5.1转储数据到ODBC数据源

iPlatda 的ODBC数据库连接、转储配置

ODBC数据库连接配置

1.在iPIat平台的【SCADA管理】中可以配置数据库连接登陆后进入主界面，点击首页左侧树节点【SCADA管理】-【某个SCADA点】-【数据库连接管理】，可以进入数据库连接管理页。

2.新增：进入数据库连接管理页面后，点击右上角【新增】按钮，弹出的配置数据源连接信息页面。

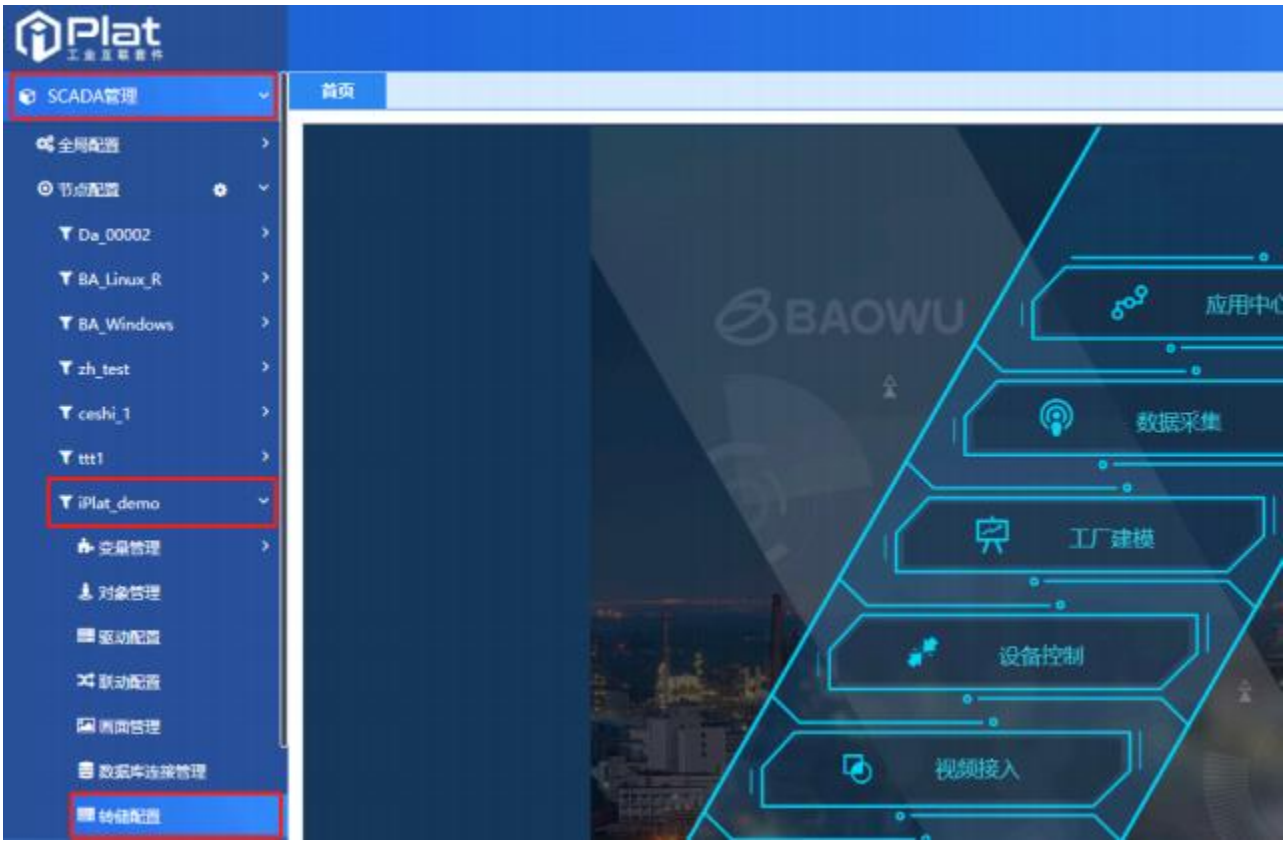
•输入如下内容：

- 。数据库类型（必填项）：此处在下拉框中选择ODBC
- 。名称（必填项）：该数据库连接的名称
- 。描述：该数据源描述
- ◦数据库名称（必填项）：在【配置前准备工作：在iplatda运行的服务器上配置ODBC Data Source】中配置的data source名
- ◦用户名（必填项）：数据库用户名
- ◦密码：数据库密码
- ◦因为odbc的连接测试是在iplatda运行的服务器上进行的，所以这里的【测试连接】是不能点击的
- ◦修改：数据连接管理页面，选中一条数据源记录，双击后可以对其进行修改操作
- ◦删除：数据连接管理页面，选中一条数据源记录，点击右上角删除按钮可以对其进行删除操作

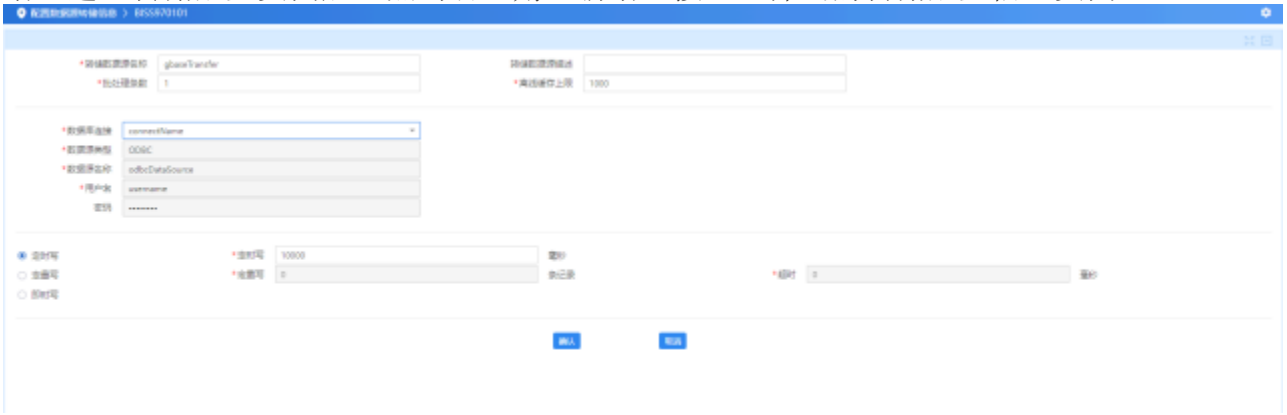
转储数据乱码问题可查看转储数据时乱码

ODBC转储配置

1. 在IPlat平台的【SCADA管理】中可以配置转储配置登录后进入主界面，点击首页左侧树节点【SCADA管理】-【某个SCADA节点】-【转储配置】，可以进入转储配置页面，选择【数据源】Tab页进行转储配置。



2. 新增：进入转储配置页面后，点击右上角【新增】按钮，弹出的转储配置信息页面。



•输入如下内容：

◦转储数据源名称：转储中配置的名称，必填，自定义字符串

◦批处理条数：单次进行转储的记录数

◦转储数据源描述：转储中的数据源描述，选填，自定义字符串

◦离线缓存上限：在数据没有被转储成功时系统最大缓存记录数

◦数据库连接：在此处选择使用的数据源。数据源连接配置参见【ODBC数据库连接配置】

◦***数据源类型、数据源名称、用户名、密码会在选择数据库连接后自动填入**

◦定时写：每隔指定时间进行一次转储。（由于定时器的原因配置的时间乘以0.8才是生效的具体时间，且这段时间缓存的数据都会转储过去，所以在配置的时候，数据量不要配置太大，会导致数据占满内存已经数据的丢失）

◦定量写：缓存到指定的记录数后进行一次转储

◦即时写：每产生一条需要转储的数据即进行转储。（搭配数据转储的扫描周期配置，可以达到定时转储的要求，比如扫描周期配置1800000毫秒，勾选即时写可以达到半个小时转储一次的效果）

◦***定时写和即时写最大的区别在于性能，定时写效率更高速度更快**

◦修改：数据连接管理页面，选中一条数据源记录，双击后可以对其进行修改操作

◦删除：数据连接管理页面，选中一条数据源记录，点击右上角删除按钮可以对其进行删除操作

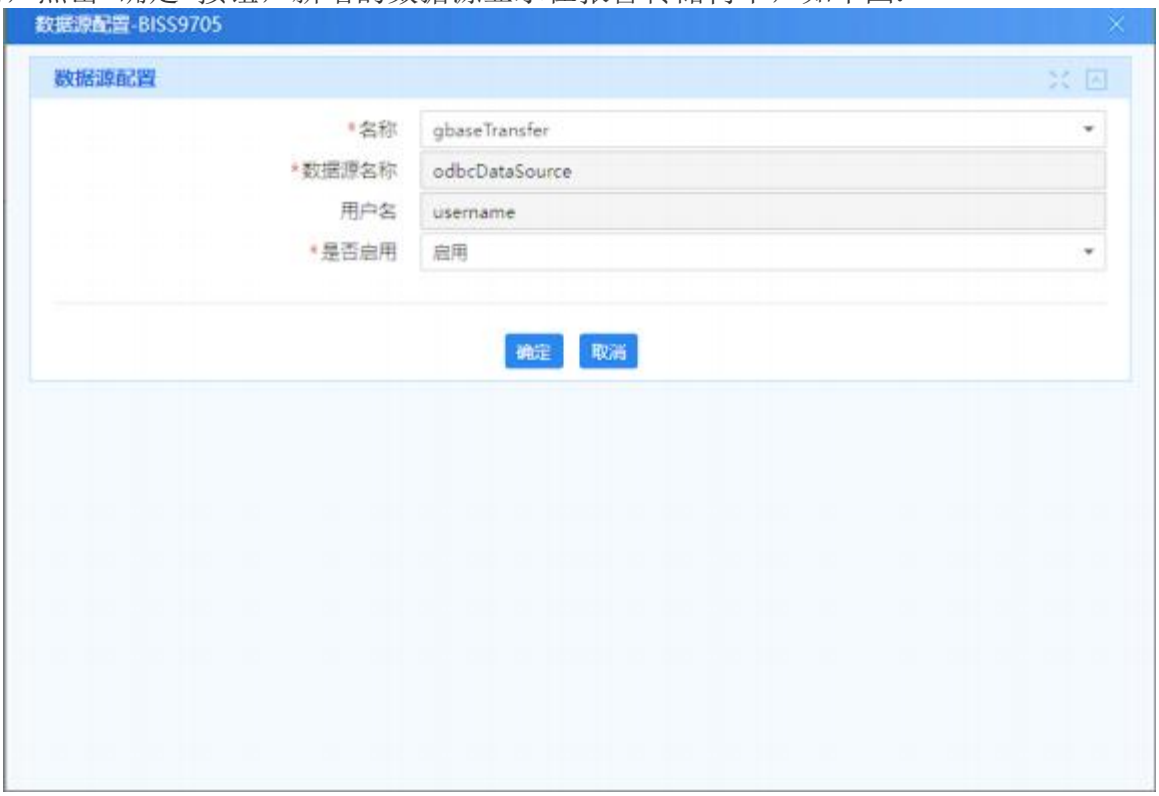
tag点的数据和报警转储到ODBC的配置

*开始配置此步骤之前，需要配置好【数据库连接配置】和【转储配置】

1. **报警转储配置**
   - 【转储配置】页面选择【报警】Tab页进行报警转储配置

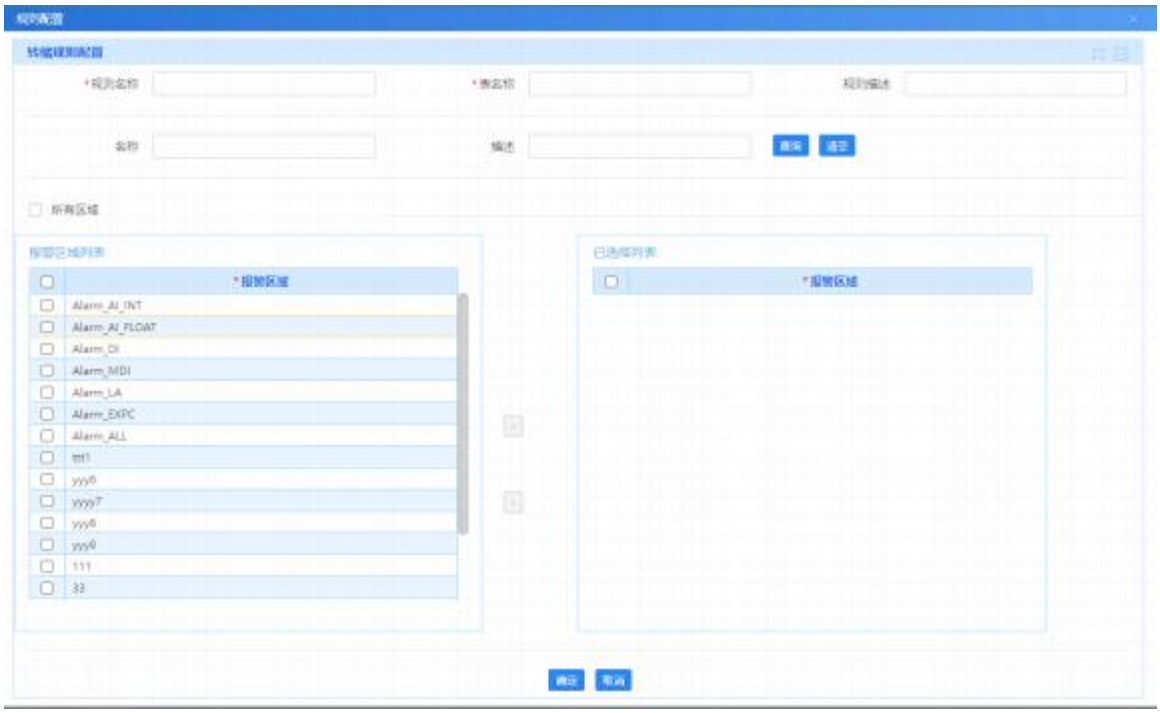
   - 新增数据源

◦右键点击根节点"报警转储"，点击"新增数据源"选项，弹出"数据源配置"界面，点击【名称】下拉框，选择需要转储到的数据源（数据源配置参见数据源配置），【是否启动】选择启用，点击"确定"按钮，新增的数据源显示在报警转储树中，如下图：



- 新增规则

◦右键点击某数据源节点，点击"新增规则"选项，弹出"转储规则配置"界面，如下图所示：



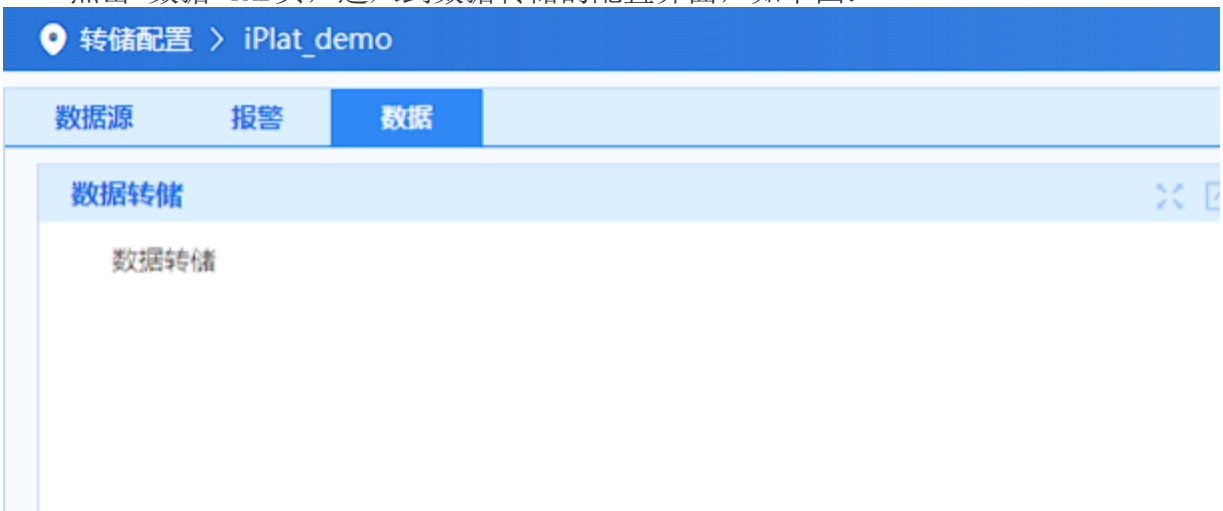
. 规则名称：自定义的规则名称
. 表名称：转储到数据库中对应的表的表名
. 规则描述：自定义的规则描述
. 所有区域：勾选所有区域后，所有的报警都会转储到数据库
. 已选择列表：若没有勾选所有区域，可以把结果列表中的一些报警区域选择到已选择列表中，已选择列表中的报警会转储到数据库
. 配置完成后，点击"确定"按钮，完成配置。注意：希望配置生效并成功转储，必须在配置完毕后进行部署操作
. 关于转储表中的fd_alarmtype中的数字和报警类型的对应关系如下表所示：

fd_alarmtype(数字)	报警类型描述
1	低低报警
2	低报警
3	高报警
4	高高报警
5	变化率报警
6	变位报警
26	0->1报警
fd_alarmtype(数字)	报警类型描述
27	1->0报警
28	user defined alarm type
29	user defined alarm type

30	user defined alarm type
31	user defined alarm type
32	user defined alarm type
33	user defined alarm type
34	user defined alarm type
35	user defined alarm type
36	user defined alarm type
37	user defined alarm type
38	user defined alarm type
39	user defined alarm type
40	user defined alarm type
41	user defined alarm type
42	user defined alarm type
43	user defined alarm type
44	=1状态报警
45	=0状态报警

2. **数据转储配置**

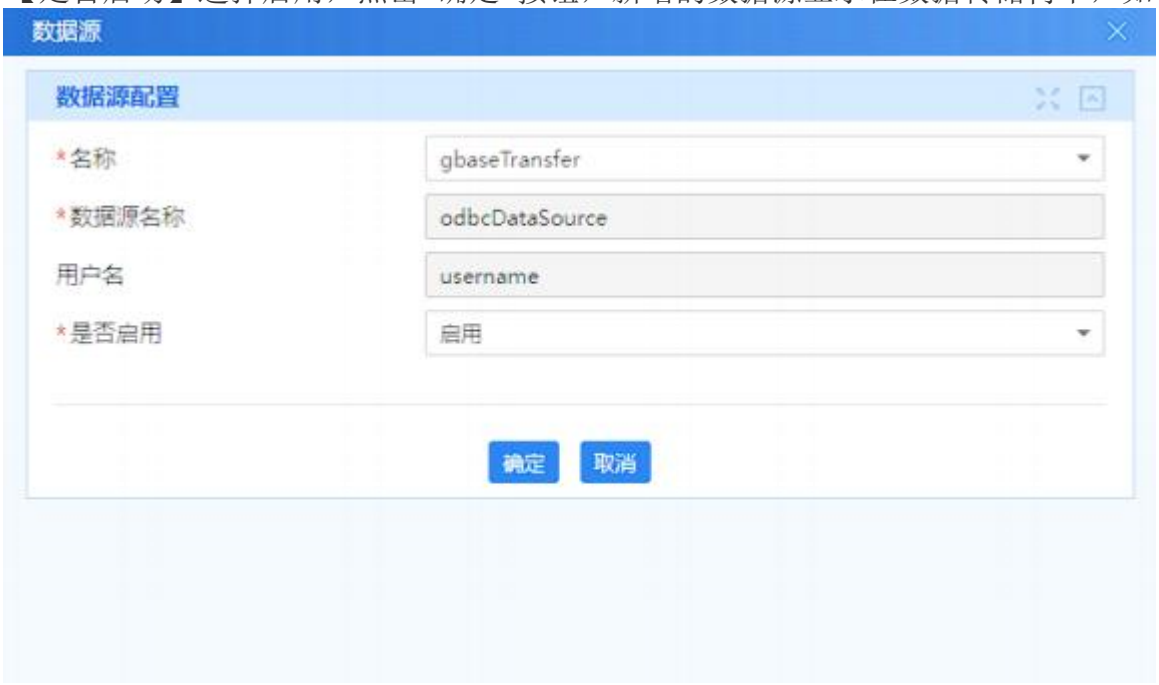
•点击"数据"TAB页，进入到数据转储的配置界面，如下图：



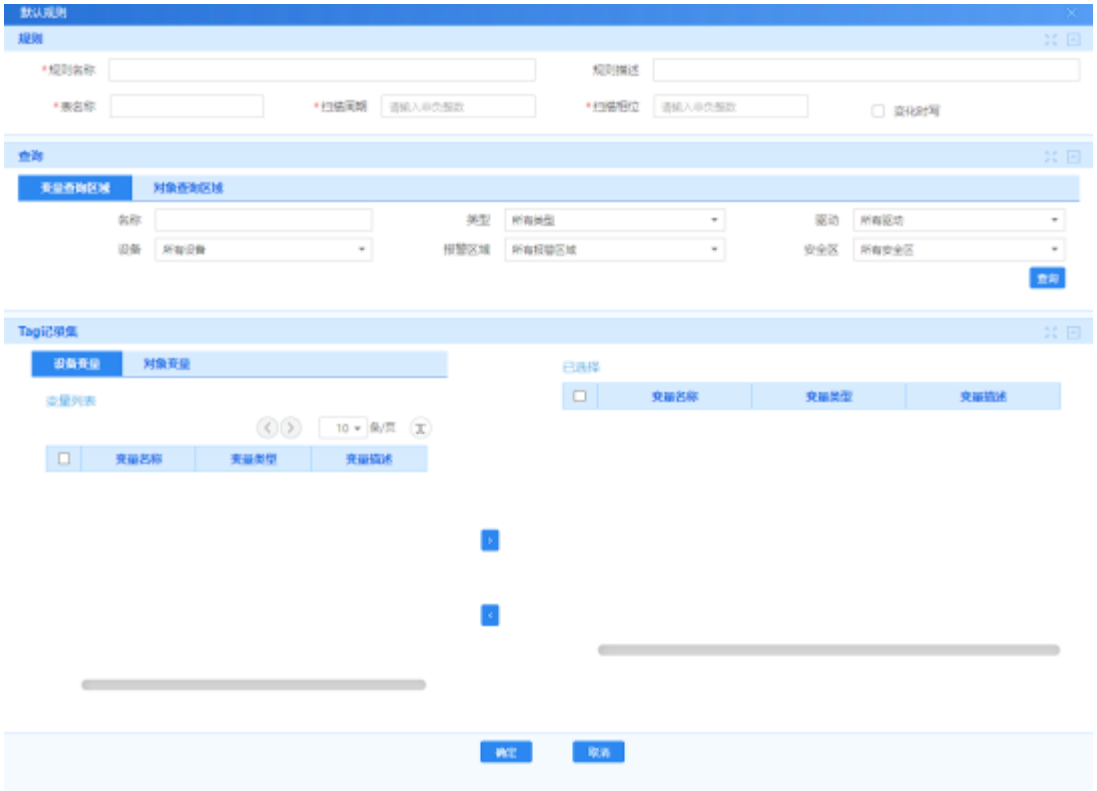
•新增数据源

◦右键点击根节点"数据转储"，点击"新增数据源"选项，弹出"数据源选择"界面，点击【名称】下拉框，选择需要转储到的数据源（数据源配置参见数据源配置），

【是否启动】选择启用，点击"确定"按钮，新增的数据源显示在数据转储树中，如下图：



- 新增规则
  - 默认规则
    - 点击某数据源节点，展开节点后，在【默认规则】上点击右键"新增默认规则"选项，弹出"默认规则配置"界面，如下图所示：



    - 规则名称：自定义的规则名称。
    - 表名称：转储到数据库中对应的表的表名。
    - 规则描述：自定义的规则描述。
    - 扫描周期：获取tag点值的时间间隔。
    - 扫描相位：扫描周期中的相位。
    - 变化时写：若勾选变化时写，则只有该变量的值变化时才进行转储。若不勾选，每个扫描周期得到的值都进行转储。

. 已选择：左侧列表中显示了当前SCADA下的tag点列表，可以选择一个或多个TAG点到右侧"已选择"列表中，被选中的TAG点将被转储。

. 配置完成后，点击"确定"按钮，完成配置。转储结果如下所示：

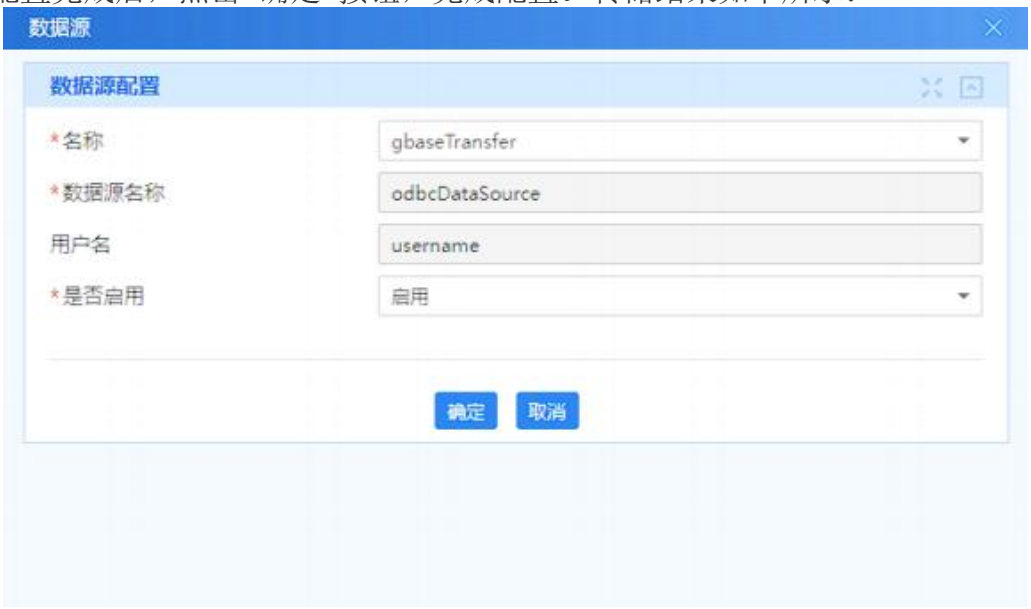
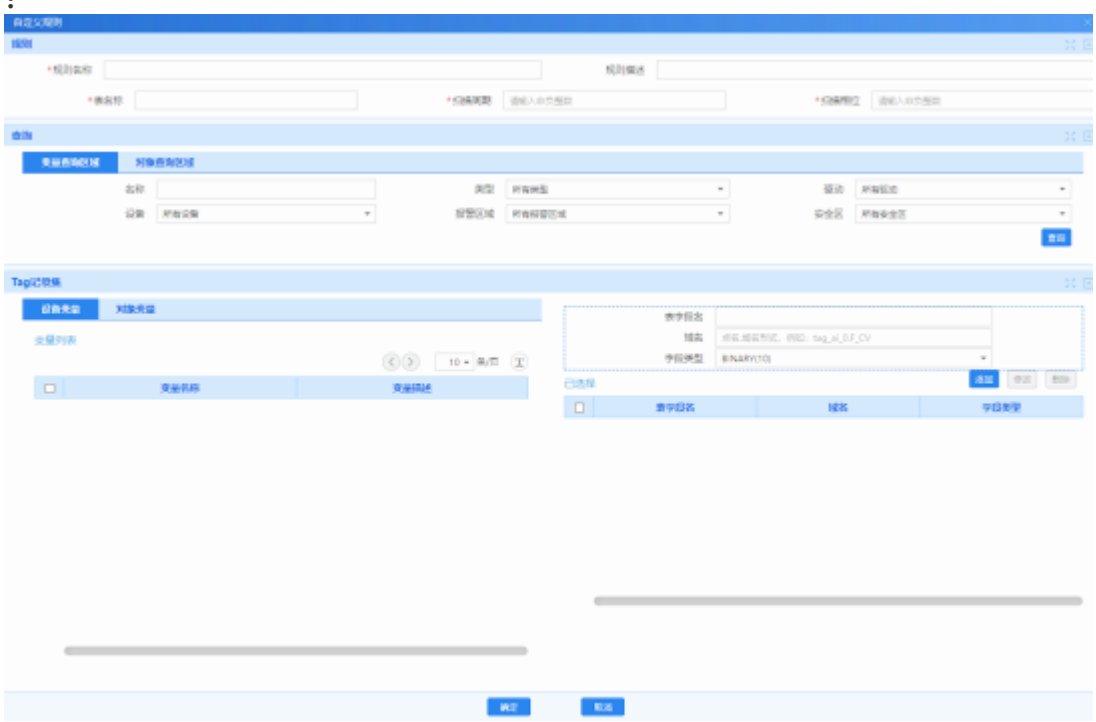


Figure 1 datatransfer_4

▪共包含六个列，分别是：转储配置中的数据源名称，tag点名称，tag点描述，时间戳，tag点值，tag点的数据质量。配置生效并成功转储，必须在配置完毕后进行部署操作。

◦自定义规则

▪右键点击某数据源节点，点击"新增自定义规则"选项，弹出"默认规则配置，如下图：



▪规则名称：自定义的规则名称。

▪表名称：转储到数据库中对应的表的表名

▪规则描述：自定义的规则描述

▪扫描周期：获取tag点值的时间间隔

▪扫描相位：扫描周期中的相位

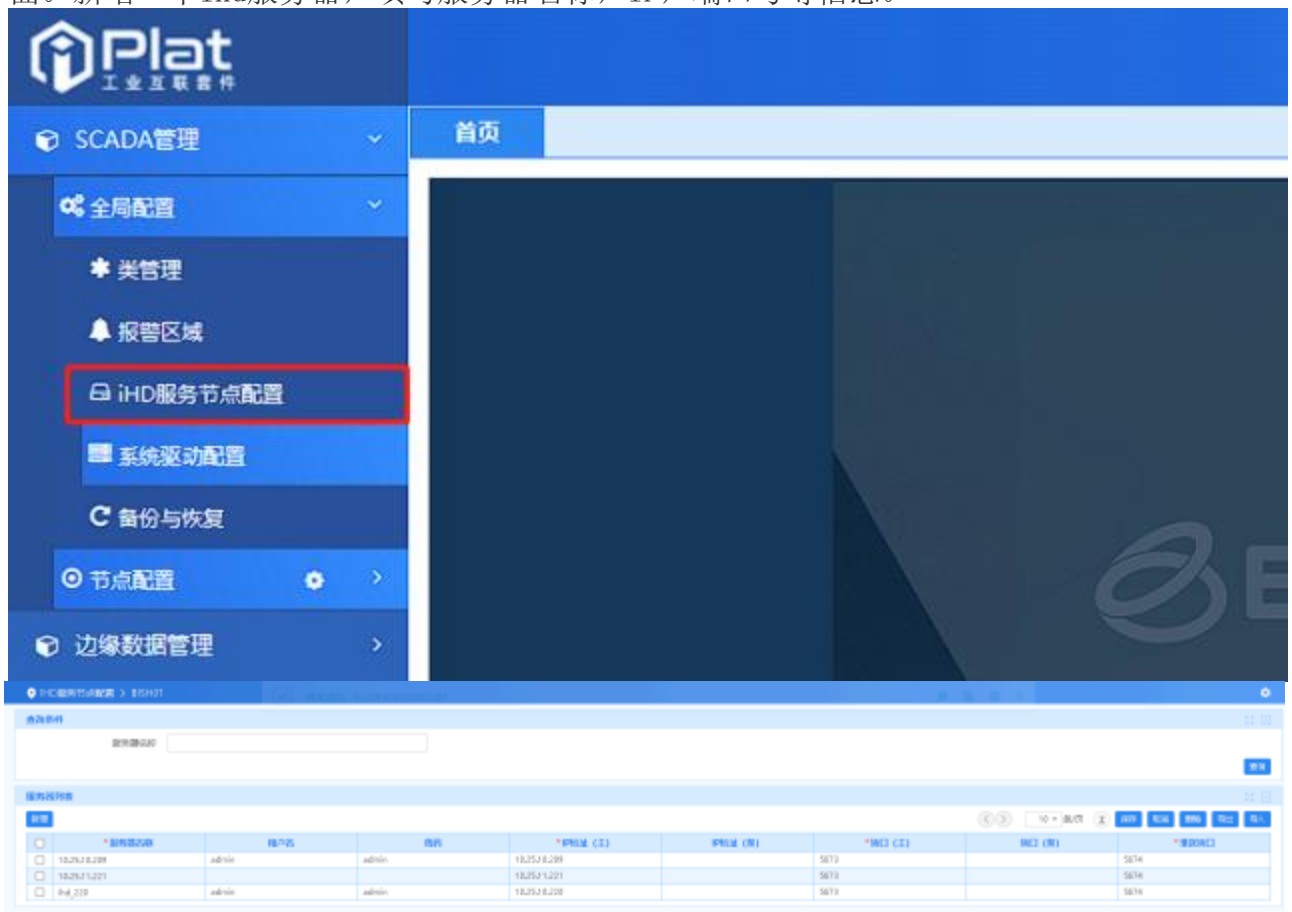
▪表字段名：对应到的数据库表中的列名

- 域名：需要转储到数据库的值域，格式为"TAG点名称.域名"。可以通过左侧变量列表选择变量，再在域名中选择需要转储的域。选择会自动生成到"域名"对应的文本框中
- 字段类型：对应到的数据库表中的该列的类型
- 添加按钮：完成"表字段名"、"域名"、"字段类型"三个字段的填写后，点击"添加"按钮，将该配置保存到下面的表字段名列表中
- 表字段名列表：转储服务会按照当前列表中的配置创建表，并将配置的对应的TAG点的域转储到对应的字段中
- 配置完成后，点击"确定"按钮。完成自定义规则的配置

转储数据到实时数据库

步骤演示

1. 配置指定的IHD，从主菜单中的"SCADA管理"-"全局配置"-"IHD服务节点配置"中进入配置页面。新增一个ihd服务器，填写服务器名称，IP，端口号等信息。



2. 在iPlat中修改点的属性，勾选"TAG点配置信息"-"转储"tab页的"IHD（v3）"选项，系统会将这个点的数据保存至指定的IHD系统中。

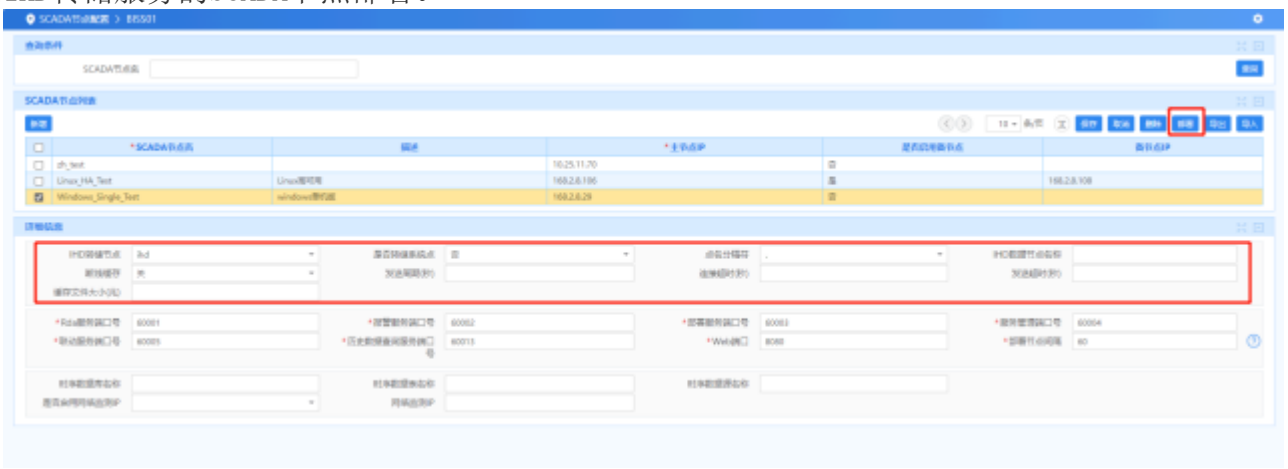
- 压缩误差：压缩误差为0时表示不压缩，压缩超时不起作用。压缩误差大于0表示开启压缩，压缩超时同时起作用。
- 扫描周期：该属性忽略。

- 压缩超时：对于于ihd的压缩超时，只有压缩误差大于0才会生效。单位为秒。
- IHD点的名称中包含scada名称：转储到ihd的名称包含scada名称作为前缀，如scada的名称为"LF"，点名为"tag1"，转发到ihd后创建的点名为"LF.tag1"。
- 变化时转储：数据变化时才转发数据到ihd，数据不变化时不转发。启动时的第一条数据总时转发。

IHD中的点名格式：

- 宝信基础自动化软件20211026之前的版本创建到IHD中的点名格式为scada_tag,从宝信基础自动化软件20211026版本开始默认创建到IHD中的点名格式为scada.tag以支持组态回放,为了兼容使用老版本的项目升级，修改iplatda\projects\defaultproject\config\ihd3transfer.xml即可将IHD点名格式修改为scada_tag

3.在"SCADA节点列表"页面，填写下列跟ihd转储相关配置项。之后再点击部署按钮，即可完成包括IHD转储服务的SCADA节点部署。

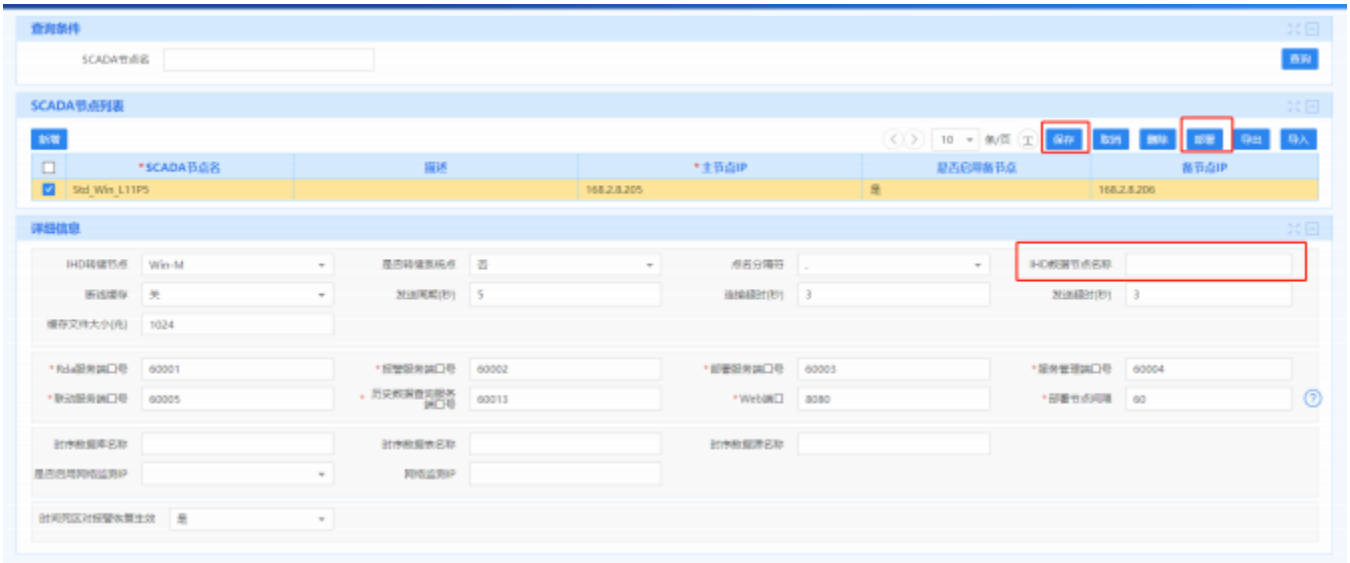


- **IHD转储节点：点击下拉框，选择目标IHD服务对应的节点。**
- **点名分隔符：scada节点和tag点中间的连接符号；如果是从宝信基础自动化软件20211026版本升级的页面，必须使用下划线作为分隔符；如果需要使用组态回放功能，必须使用"."分隔符。**
- **断线缓存：开启此功能后，ihd服务断开连接时，ihdtransfer会将tag点数据缓存在本地文件，等ihd服务重启后，继续向ihd服务推送这段时间内的tag点数据。**
- **发送周期(秒)：transfer收到的数据包很小时，不会立即转发，而是隔一段时间定期向ihd服务端发送数据，该字段就是定期向ihd发送数据的周期时长。**
- **连接超时(秒)：连接至ihd服务端的超时时长。**
- **发送超时(秒)：向ihd服务端发送数据的超时时长。**
- **缓存文件大小(兆)：断线缓存功能中本地文件的大小。**

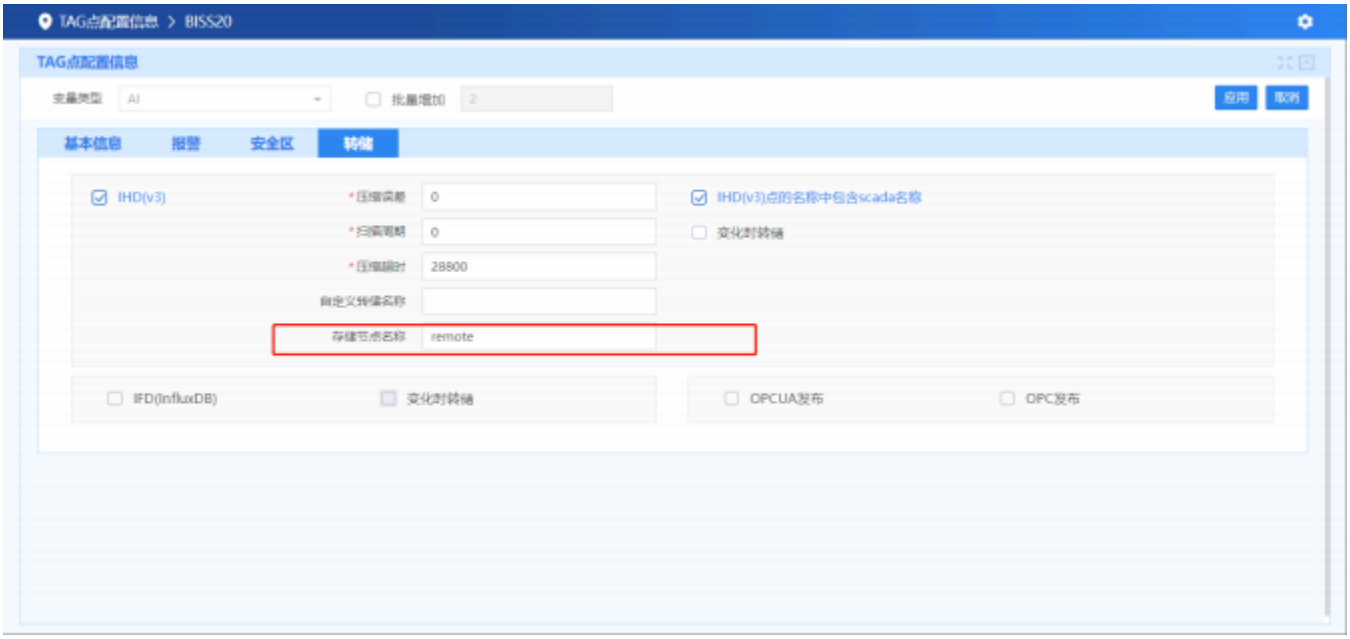
数据节点配置

目前数据节点配置分为scada全局配置和点配置，当两处都配置了时，该点最终以点配置为准，都不配置时将根据内部策略自动分配。

1.scada全局配置，在上述第三步配置完成后可以填写IHD数据节点名称，点击保存后部署



2. 点配置



存储空间计算案例

计算样例（假设）：

点数为1万点，模拟量和数字量的比例为**1:4**，（在iHD内部，数字量占用字节为9，单精度浮点型占用12字节），

则每记录平均字节数为，**(1*12+4*9)/5=10B**

iHD存储时默认采用了无损压缩机制，数据压缩率一般可达**50~60%**，取60%，则每记录平均字节数6B；

有损压缩的压缩率可取业界实时数据库产品推荐值即5%左右，iHD有3级压缩机制，如果全部都开启的，压缩率应该可以更高。

现场的数据轮询率：**点/5秒**

则每月（无损压缩）存储大小，计算公式为：

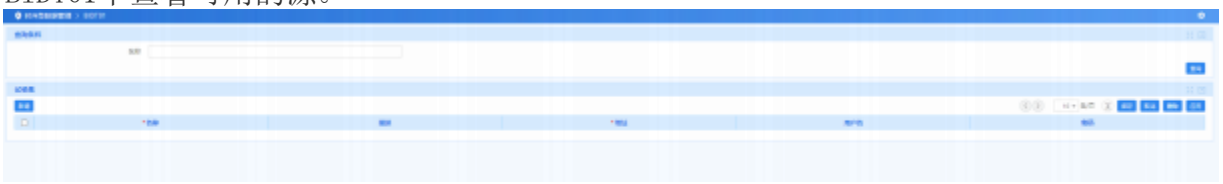
10000*(1/1)*(1/5)*6302*4*3600/1024/1024/1024=30 GB/每月

转储数据到时序数据库

步骤演示

1. 配置时序数据库源、库、表

- 时序数据库源:可用的时序数据库源可以在主页>数据源管理>时序型数据管理页面BIDT01中查看可用的源。

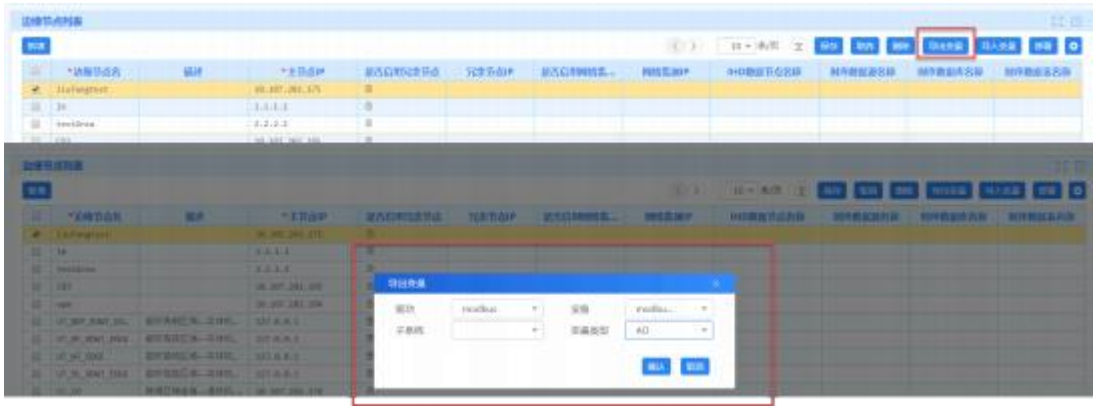


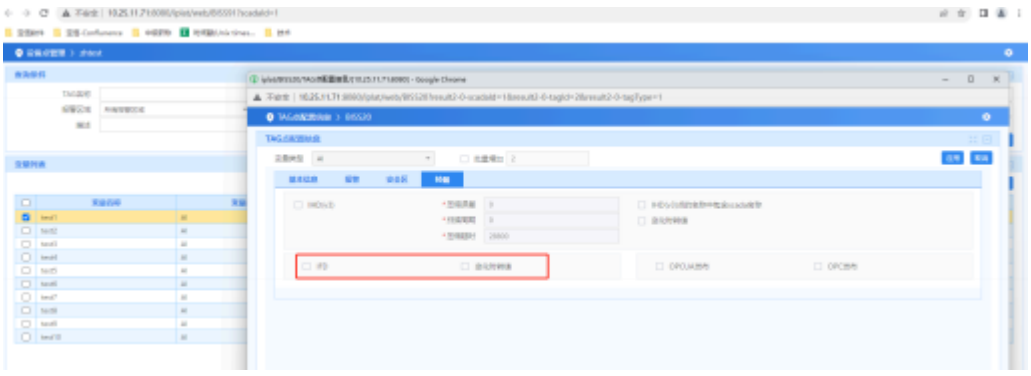
- 时序数据库:用户名密码默认为：edge/edge123
- 地址格式为：http://10.25.11.215:8086形式

2. 在BISS01页面配置转储的时序数据源信息
3. 配置变量转储参数

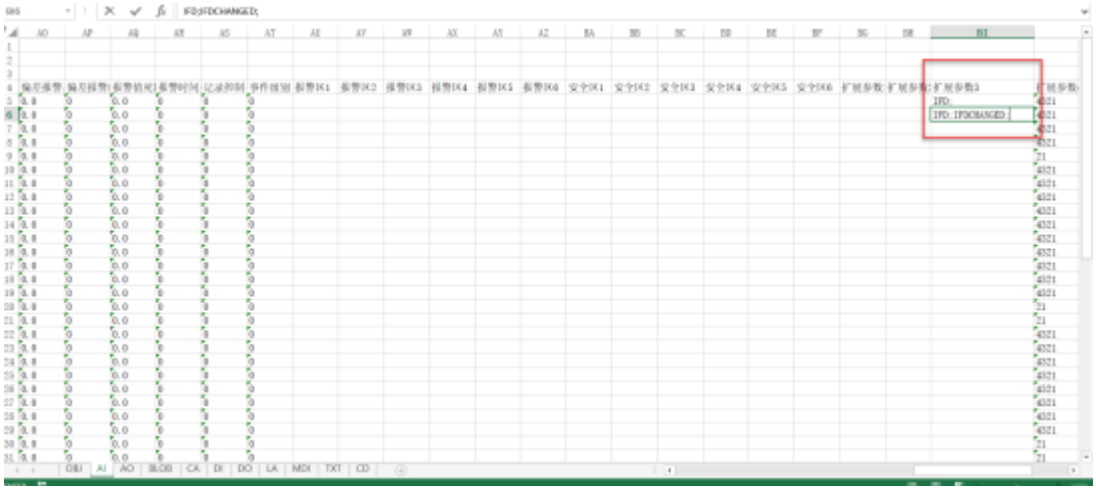
- **时序数据库转储支持两个参数配置**
  - IFD;启用influxdb转储
  - IFDCHANGED;值变化时转储
    a. 支持在页面上进行参数配置
  - b. 在scada的变量导出的excel表格中配置.步骤如下:

a. 导出变量

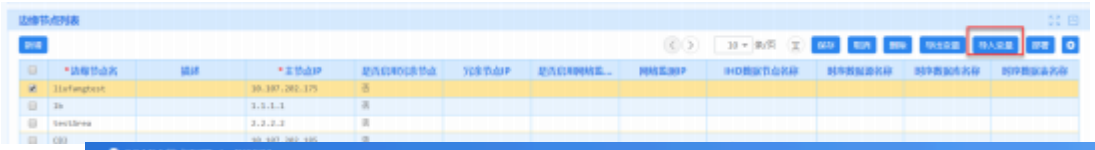




b. 在表格中相应点表的扩展参数3列填入参数,若启用转储,则填入参数IFD;若采用值变化时转储,则填入IFD;IFDCHANGED



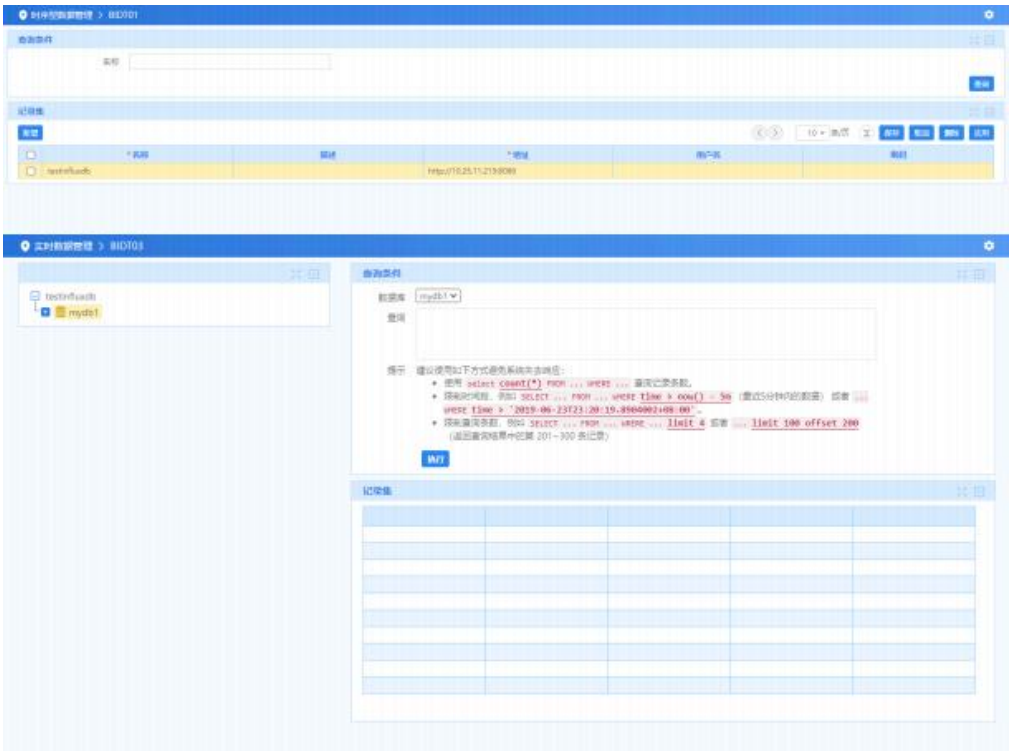
c. 导入变量



4. 配置完后需要在BISS01页面点击"部署"按钮才会生效

5. 查看转储的数据

- 在页面BIDT01,双击数据源所在行,弹出时序数据查询页面



•在查询窗口,单击数据库名,可以得到数据库下所有的表,根据表,可以查询需要的数据,如，查询test数据库中zh_test表中值TagName为test13的10条数据,SELECT*FROM zh_test where TagName='test13'limit 10

•可以看到数据查询的字段如下表

字段	数据类型	备注
TagName	string	索引
Quality	string	索引
IntVal	int	
FloatVal	float	
StringVal	string	

BoolVal	bool	
ValueType	string	
time	timestamp	时间戳

5.2 modbus发布

概述

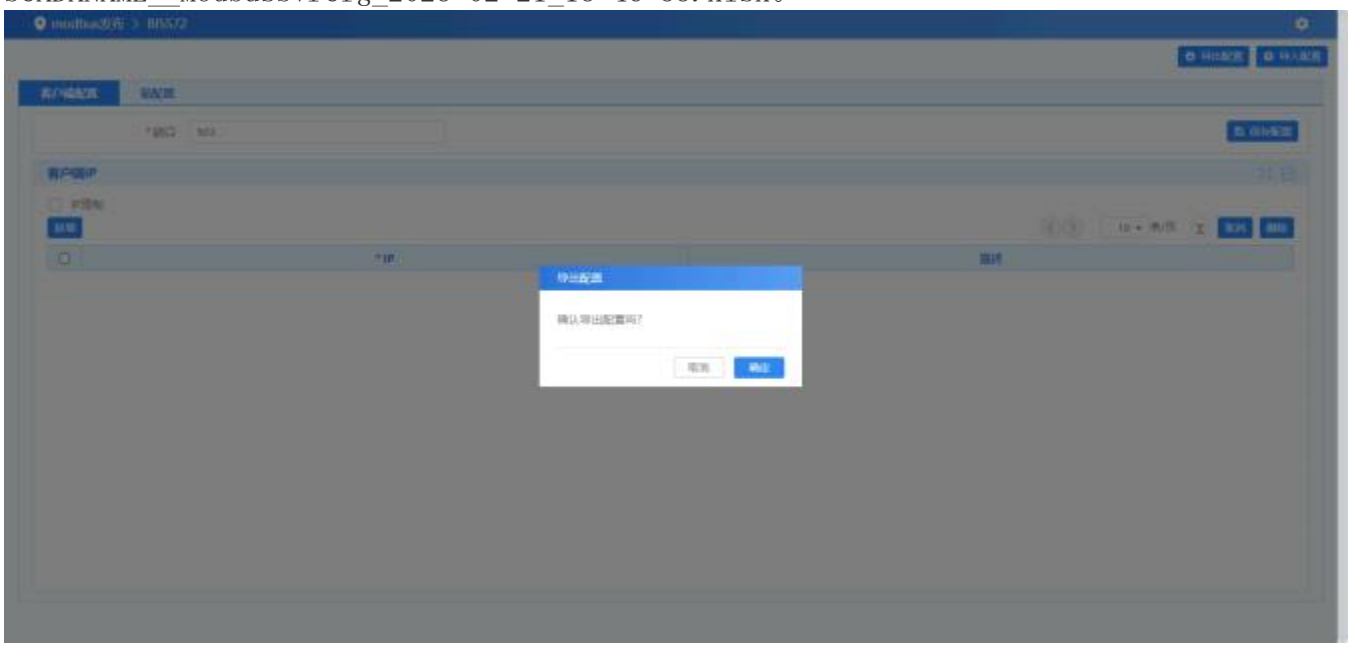
modbus数据发布服务是将iplat-da的实时数据通过modbus Tcp标准协议的方式发布出去。
第三方应用可通过modbus协议直接读取iplat-da中的实时数据，或通过此服务向设备发送控制命令（双向协议）。
实施人员可配置需要发布的Tag点，以及对应的Modbus寄存器地址。

入口

在scada节点配置页面BISS01中新增SCADA节点，在SCADA节点下的菜单中生成modbus发布子菜单。点击这个子菜单，进入modbus发布配置页面。

导出配置

点击导出按钮，确认导出后，导出SCADANAME__ModbusSvrCfg_2023-02-21_13-46-53.xlsx。

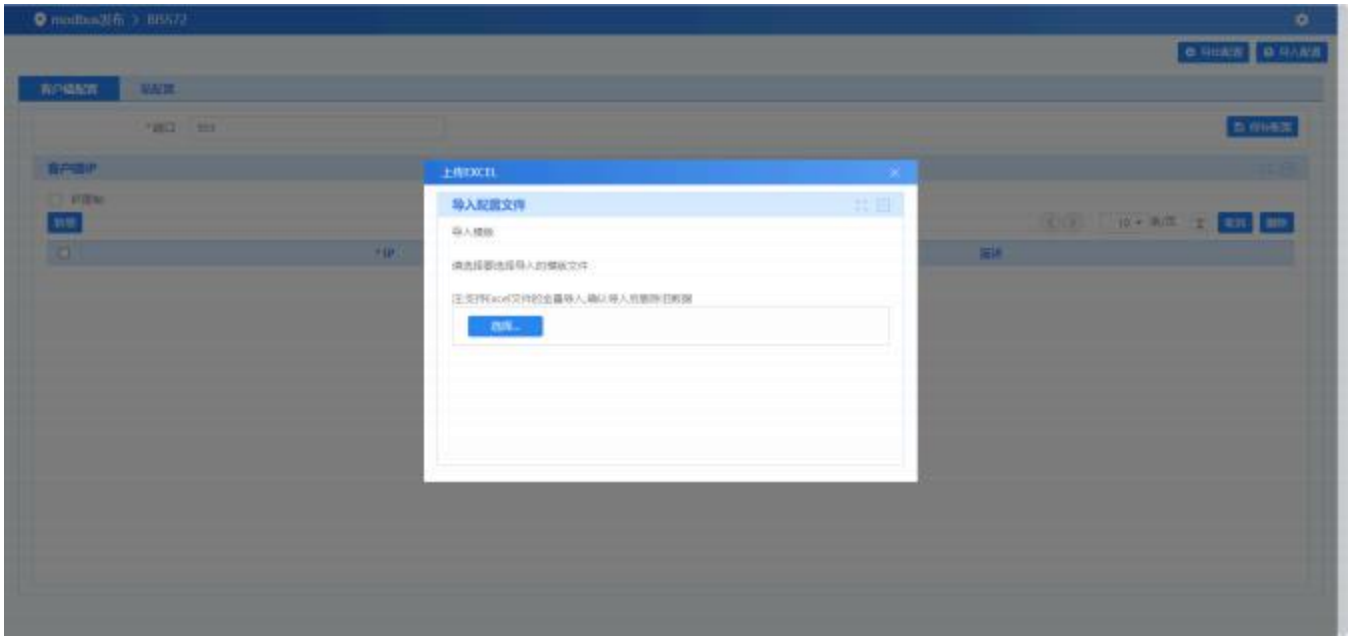


execl包括4个sheet页。

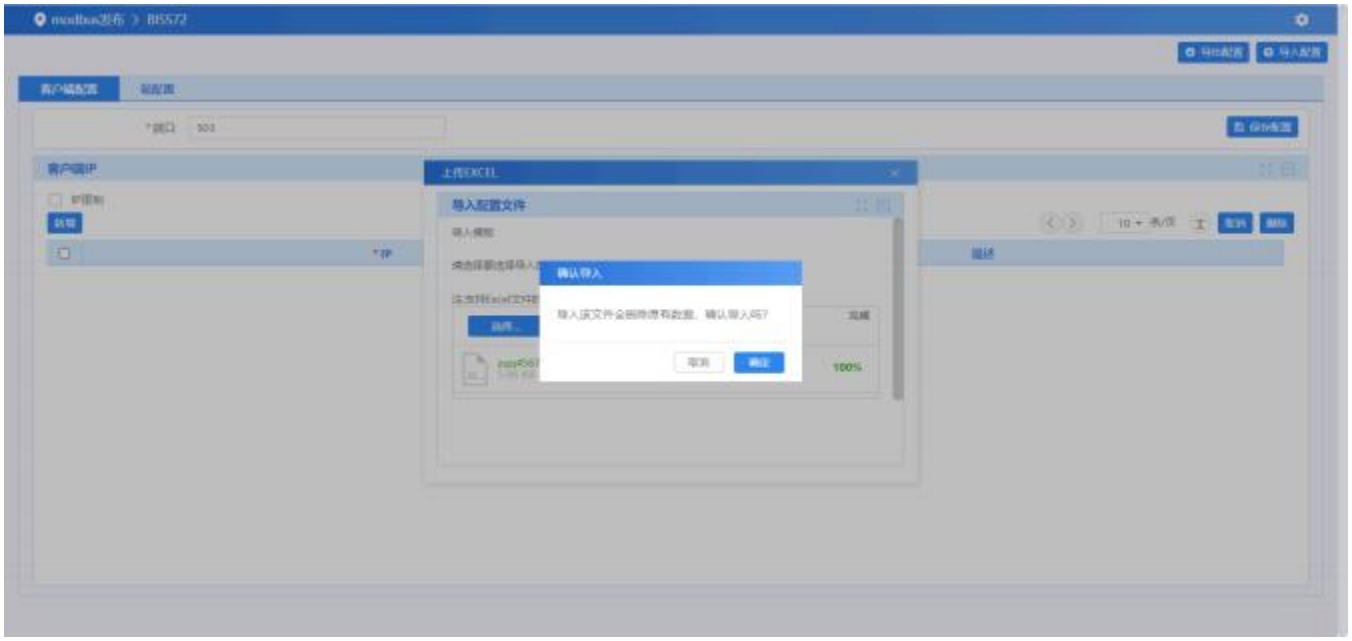
- Basic Config
- IP Config
- Station Config
- Tag Config

导入配置

点击导入配置按钮，弹出上传文件弹窗



目前只支持上传xlsx,xls类型文件



文件上传成功后，确认导入后导入该execl数据

[注意：请严格按照导出模板进行数据的导入！！！！！]格式若跟导出的模板不一致，会导致导入数据丢失或错乱。

•Basic Config:

◦该sheet页中，端口字段没有数据，导入时默认导入502。IP限制字段没有数据，导入时默认导入0;

◦该sheet页只取第一行数据，如有多行数据导入，也只取第一行数据，界面导入后有提示信息。

•IP Config

◦该sheet页中，IP该字段值如有为空的行数据，不进行导入，页面有提示信息;

◦该sheet页中，描述可为空;

◦该sheet页种IP如有重复数据，则重复行不进行导入，页面有提示信息。

•Station Config

◦该sheet页中，每行数据都不能有空字段，否则该行数据不进行导入，页面有提示信息;

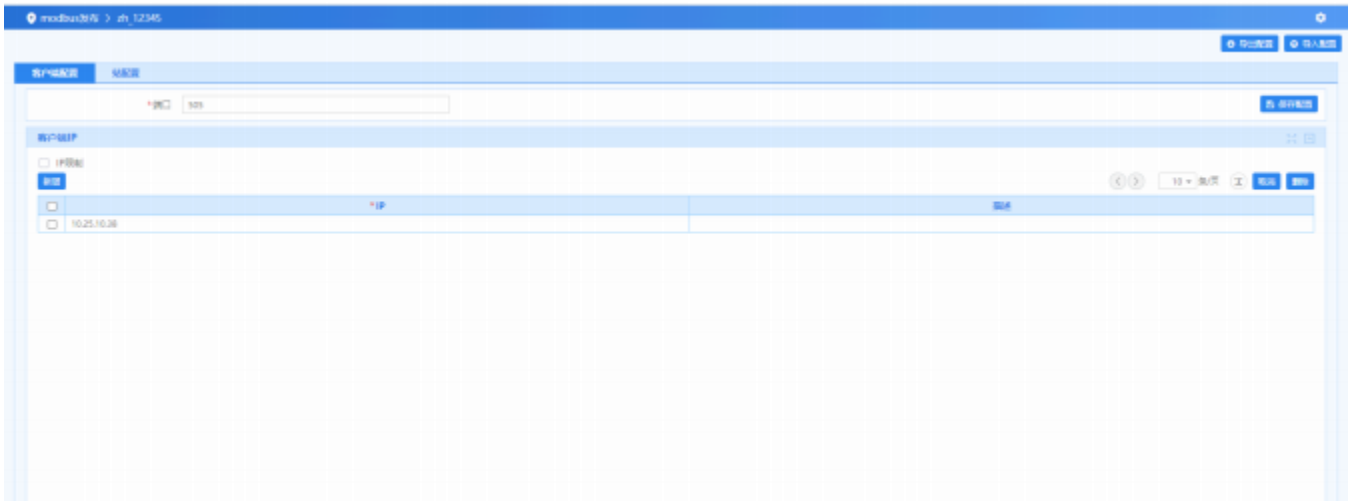
◦该sheet页中，站号不能重复，有重复的站号该行数据不进行导入，页面有提示信息;

◦该sheet页中，站号只允许在1-247范围内，站号超过这个范围的行则不进行导入，页面有提示信息。

•Tag Config
◦该sheet页中，每行数据都不能有空字段，否则该行数据不进行导入，页面有提示信息；
◦该sheet页中，站号+变量名称不能重复，有重复站号+变量名称的该行数据不进行导入，页面有提示信息；

客户端配置

客户端配置页面：



客户端配置项：

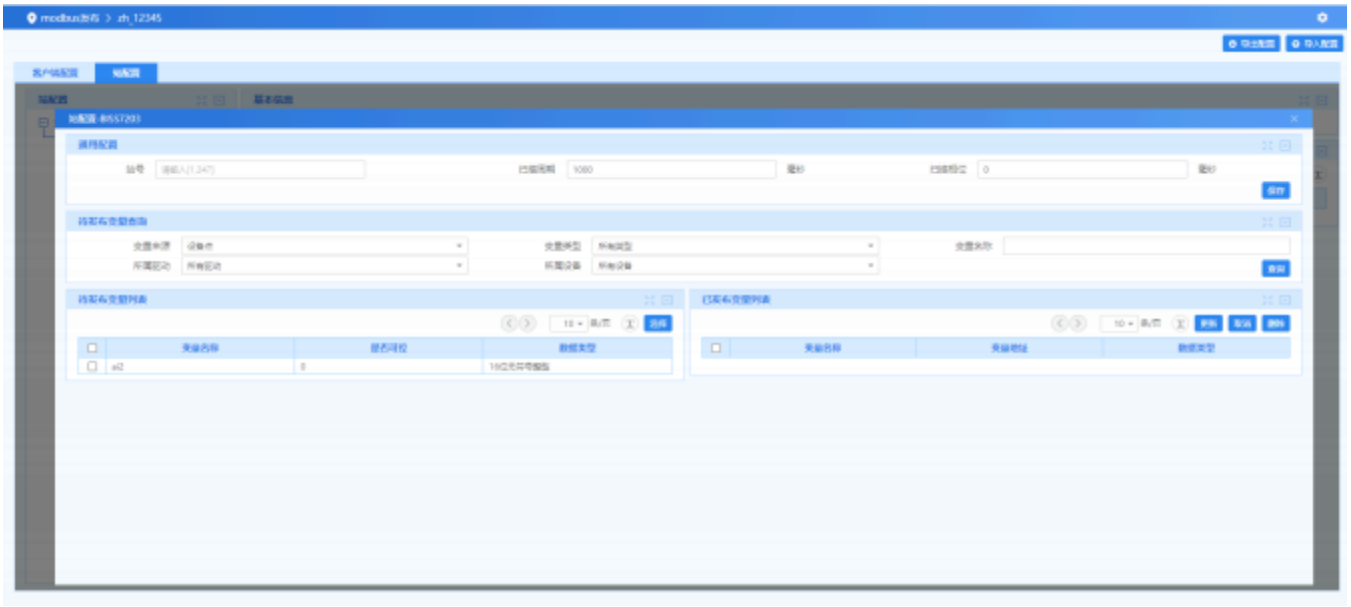
- •端口：Modbus服务监听端口,默认为502；
- •IP限制：勾选之后，只有列表中的IP地址才能访问；否则所有IP都能访问；
- •IP：可以访问Modbus tcp服务的IP地址；
- •描述：填写IP的描述信息；
- •新增：新增客户端IP；
- •取消：取消对客户端IP的编辑；
- •删除：选择要删除的IP地址，点击删除，即删除可访问的IP地址；
- •保存配置：端口和客户端IP修改之后，点击保存配置，存储到数据库。

站配置

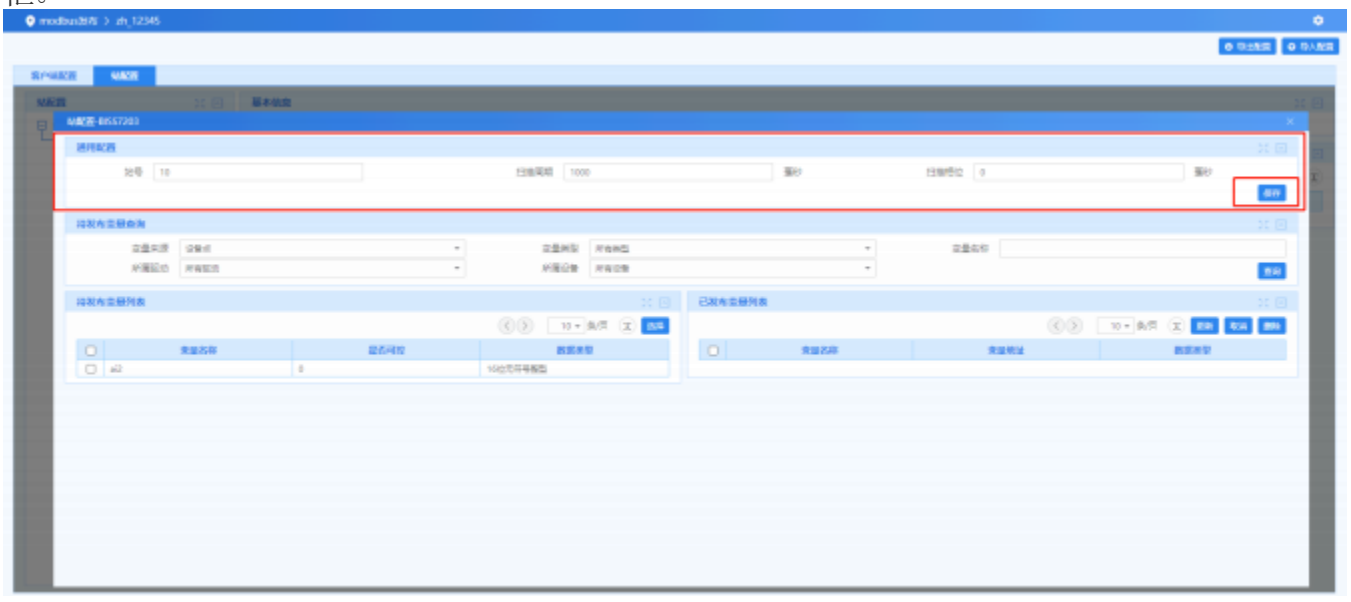
新增站

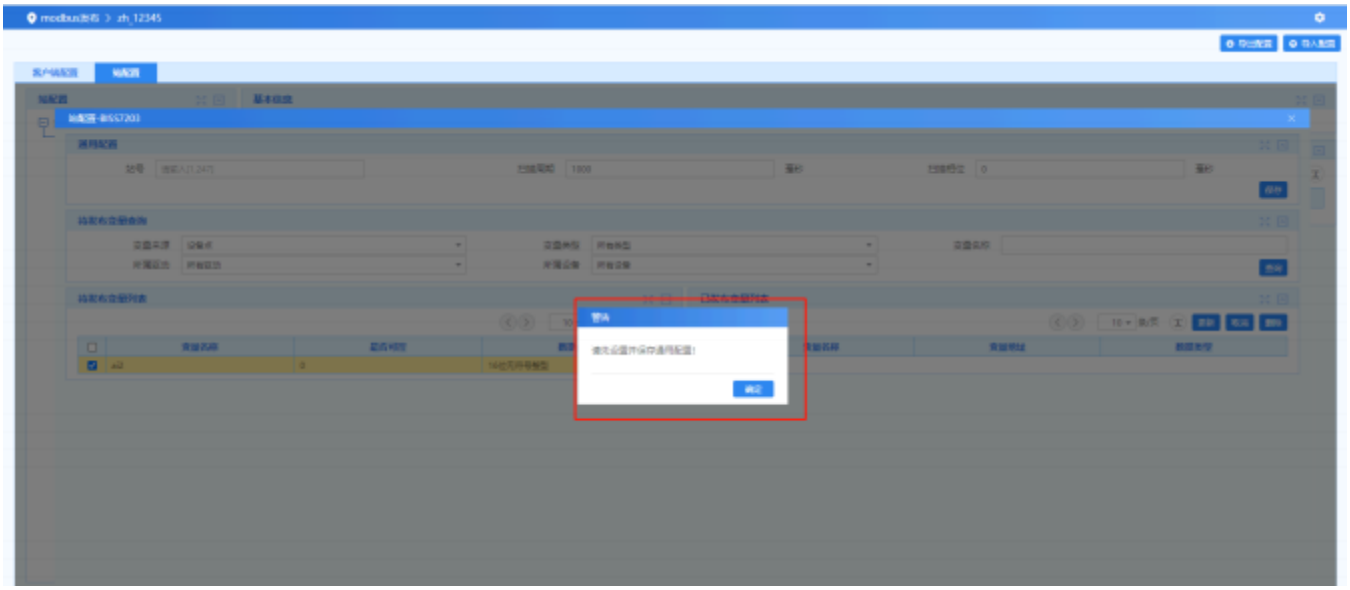
鼠标右键点击站列表，弹出新增站菜单。

单击新增站菜单，弹出站配置窗口。



新增站时，首先必须编辑并保存通用配置。否则从待发布变量列表中选点时，会弹出警告提示框。

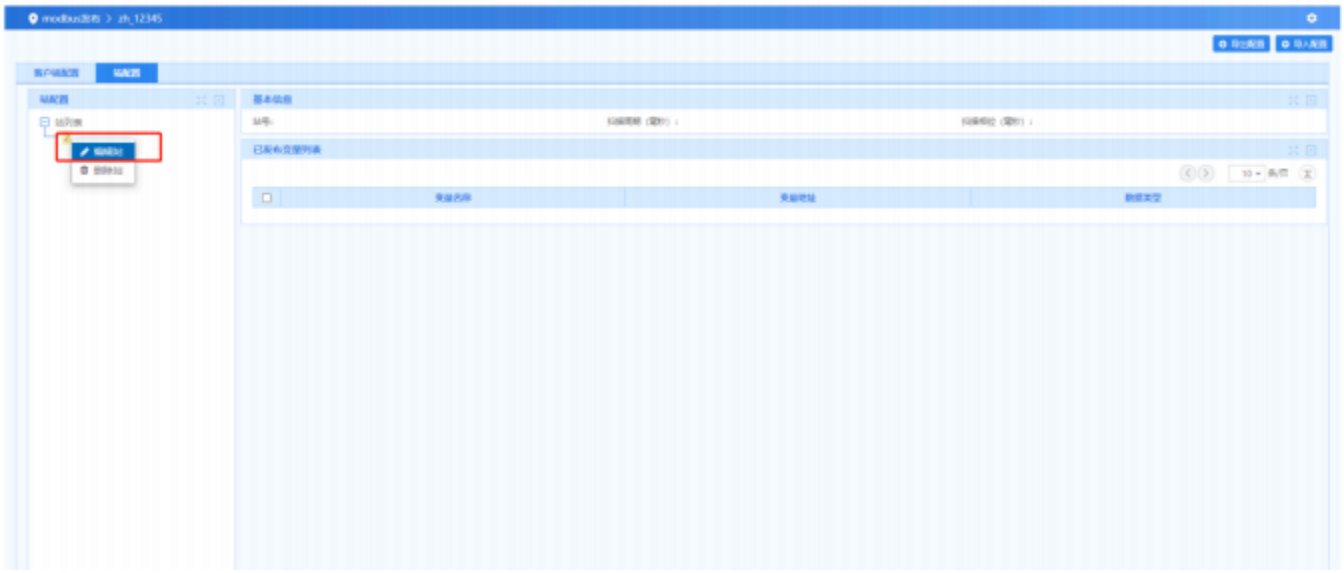




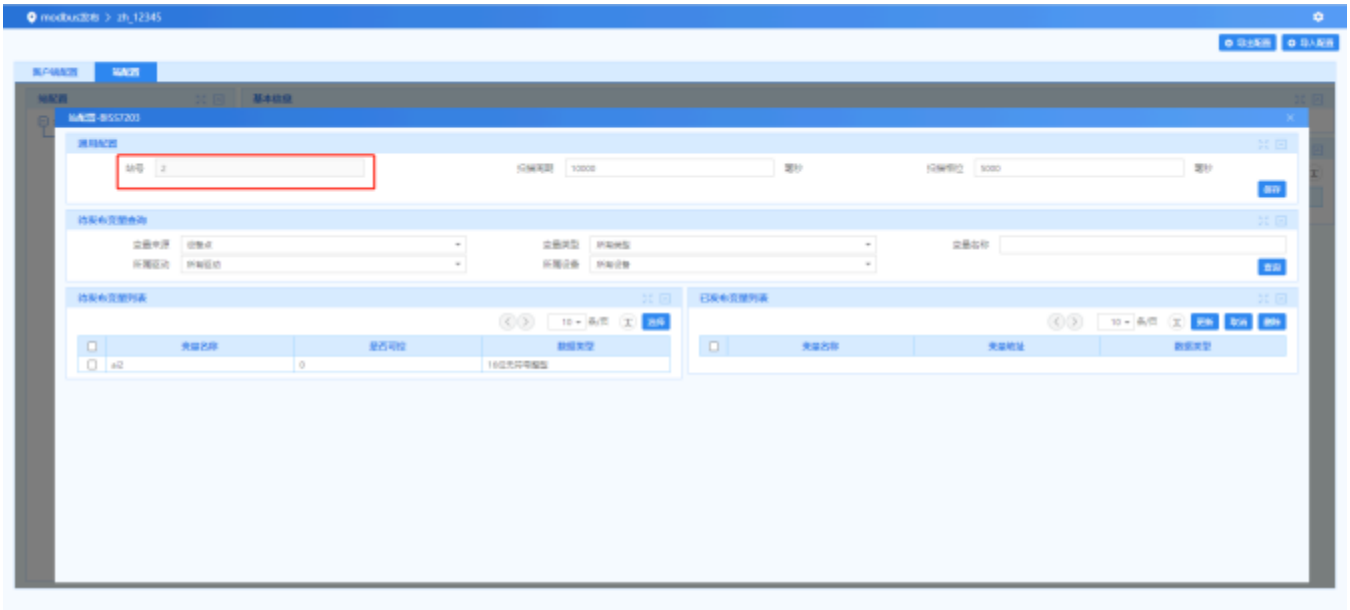
- 通用配置一旦保存，站号就不能再修改。
- 站号缺省默认为1，建议不要修改。如果配其余号码，或添加多个站，在Modbus Tcp客户端数据访问时，需要根据服务器的站号配置站号地址，并且信号点的寄存器地址格式将变为"站号@寄存器地址"。而使用站号"1"，客户端访问则无需进行站号配置。
- 建议不同数据类型的变量（Tag），分别归属不同站号管理，这样分类编辑在批量分配地址时，会降低Excel操作的复杂度。
- 扫描周期设置最小为100毫秒，并且必须是100毫秒的整数倍，周期最大86400000（24小时）。
- 扫描相位必须是50毫秒的倍数，且必须小于扫描周期。

修改站

右键点击站列表下的已有站点。点击弹出菜单中的编辑站。弹出站配置页面。



这个页面与新增站时弹出测站配置页面相同。只是站号不能修改。



删除站

右键点击站列表下的已有站点。点击弹出菜单中的删除站。删除站点。

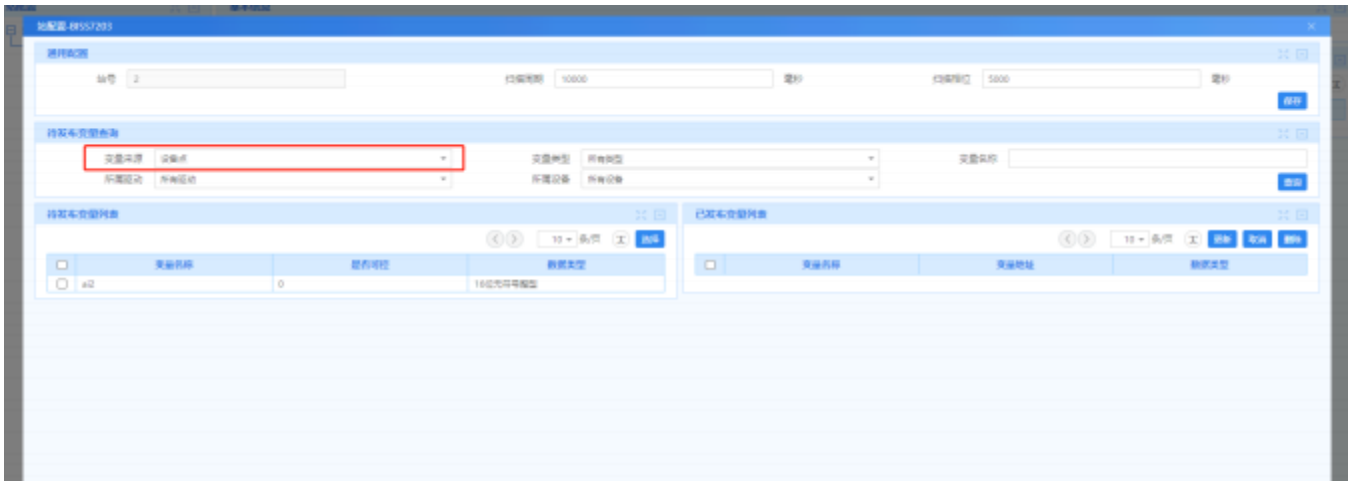
站中选点

- 可发布的变量类型：AI DI MDI TXT BLOB CA CD；不可发布的变量类型：AO,DO,LA。
- 可通过选择待发布变量列表中的点，将服务器内的变量选择为本站发布的信号。

设备点选点

进入选点页面，变量来源默认是"设备点"，此时待发布变量列表中列出的是设备点。

设备变量查询：
分别可以通过变量类型、变量名称、所属驱动、所属设备进行变量过滤。

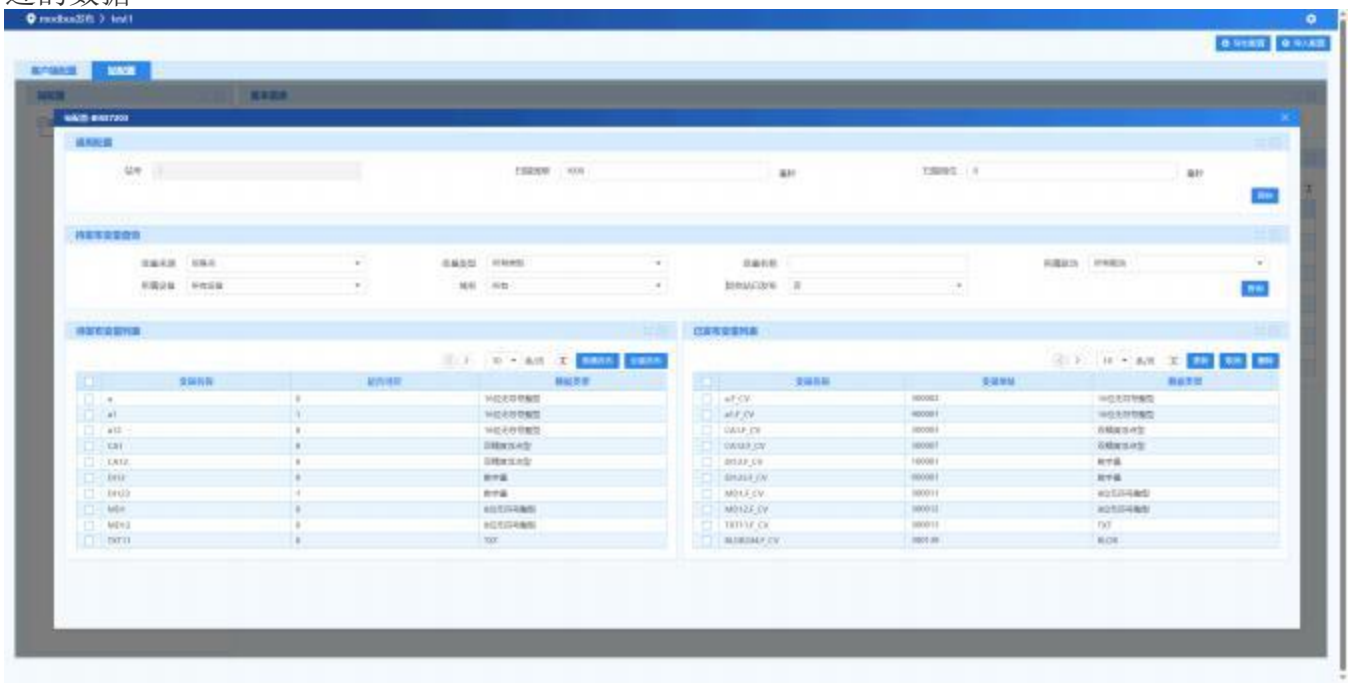


sm 5.16.0版本更新为

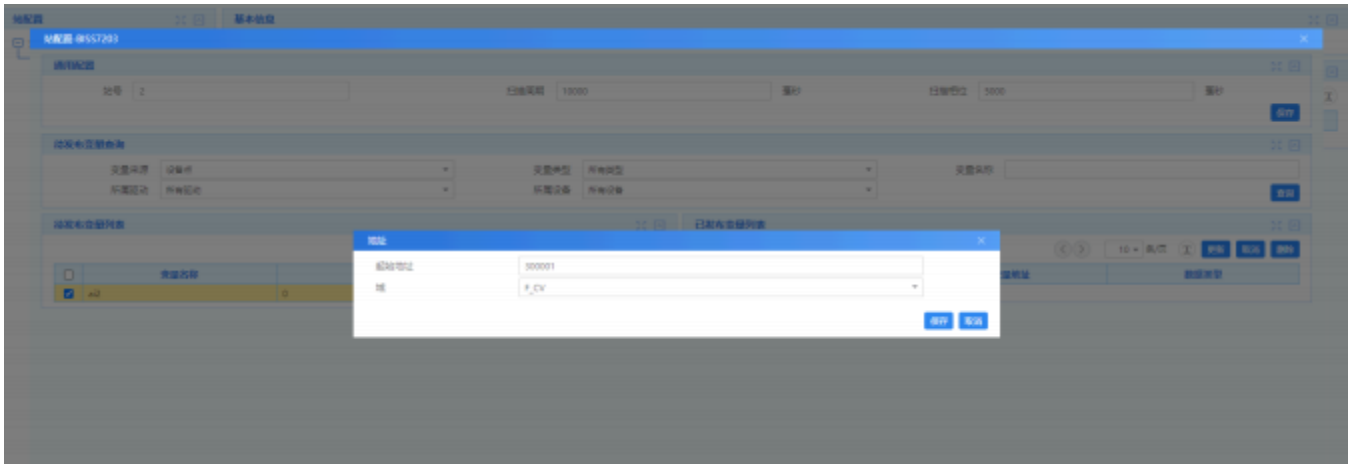
新增查询变量

1.域名：选择域名，过滤当前站下没有在该域名发布过的变量

2.其他站已发布：默认否，过滤掉在其他站已经发布过的数据。选择是，查询出在其他站发布过的数据



先选中待发布变量列中中的变量，再点击选择按钮，弹出发布起始地址和发布域配置页面。设置好起始地址和域，点击保存，选择的点保存进已发布，并在已发布变量列表中显示。



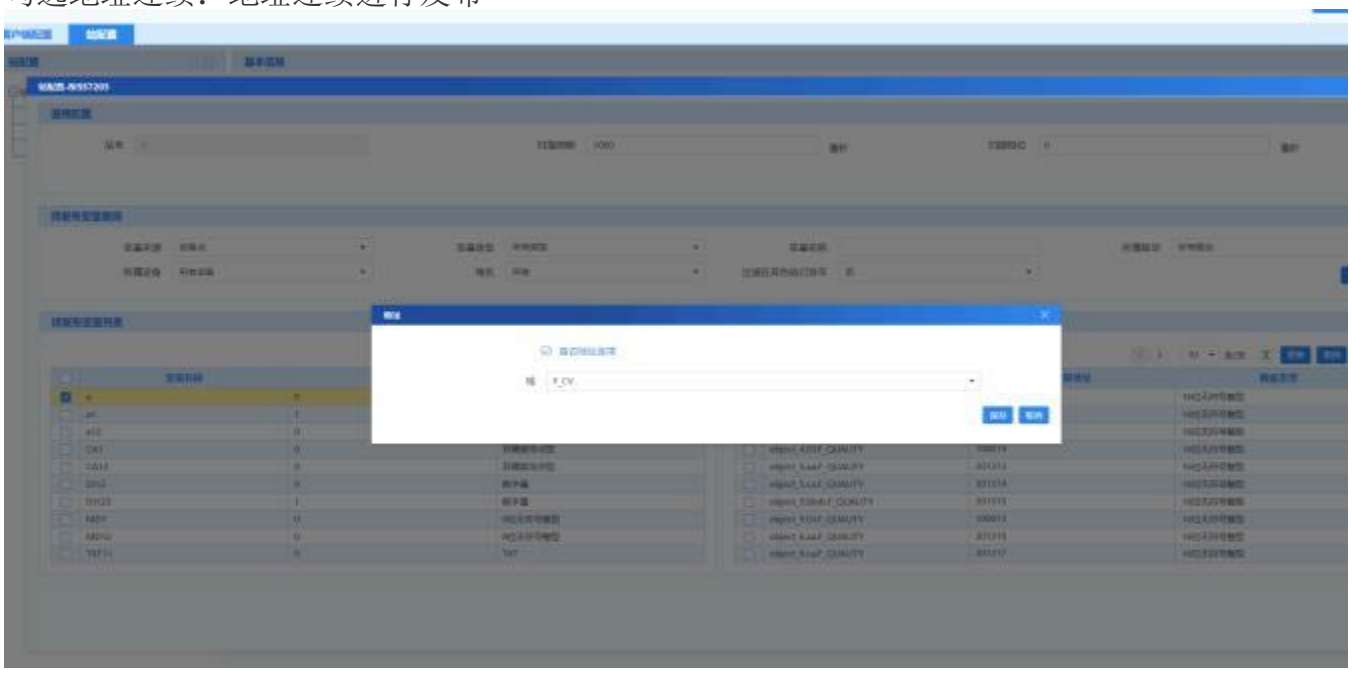
•起始地址：设置这批选中的变量的Modbus寄存器地址的首地址。除首个被选中变量，其他变量地址偏移，系统会根据数据类型自动添加。

sm 5.16.0版本更新为

不需要用户输入起始地址

默认为没有使用过的符合的最小地址

勾选地址连续：地址连续进行发布

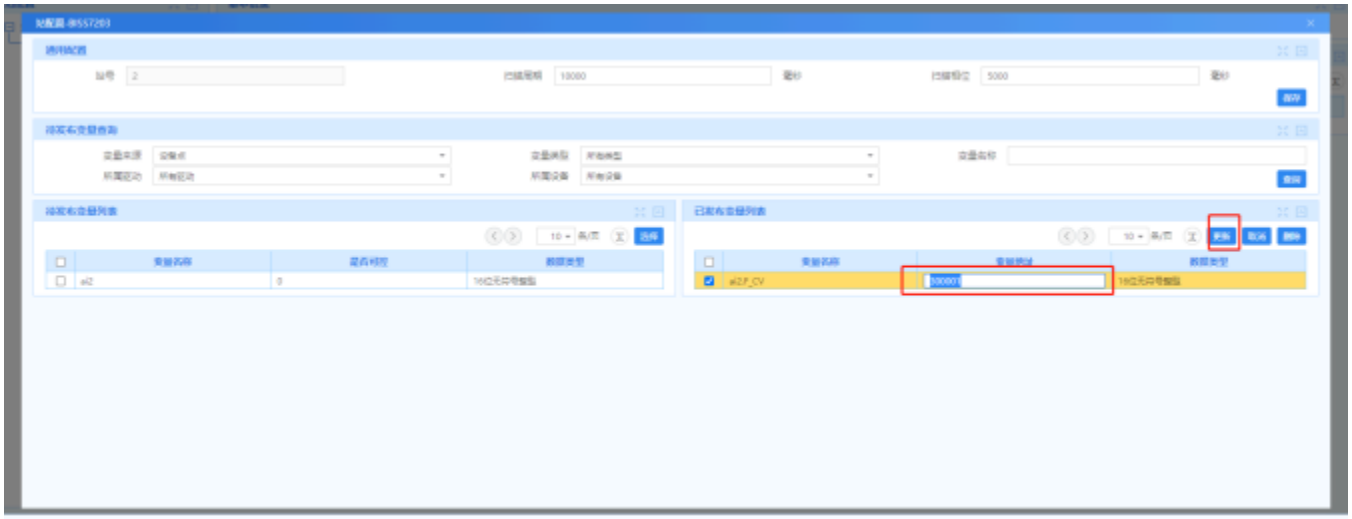


•修改地址：选中已发布变量，可修改变量地址。编辑完成后点击更新会保存修改。

sm 5.16.0版本更新为

1.已经发布过的地址不允许被重复发布

2.只支持当个修改



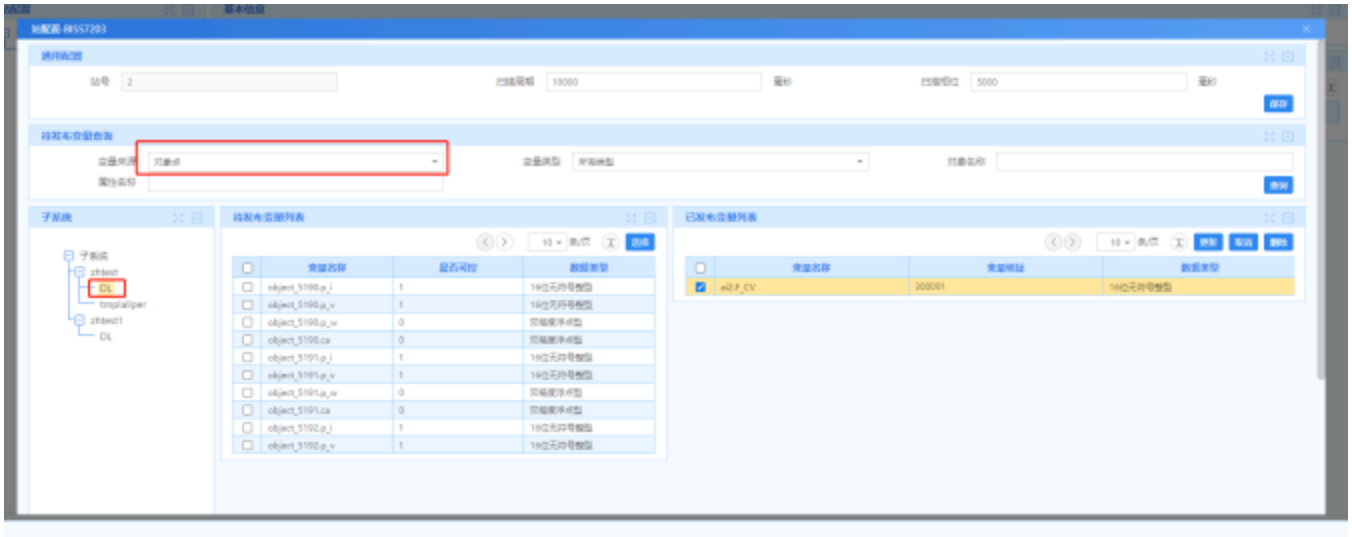
对象点选点

变量来源选择"对象点"，此时子系统区域列出对象点树，第一层是子系统，第二次是类。

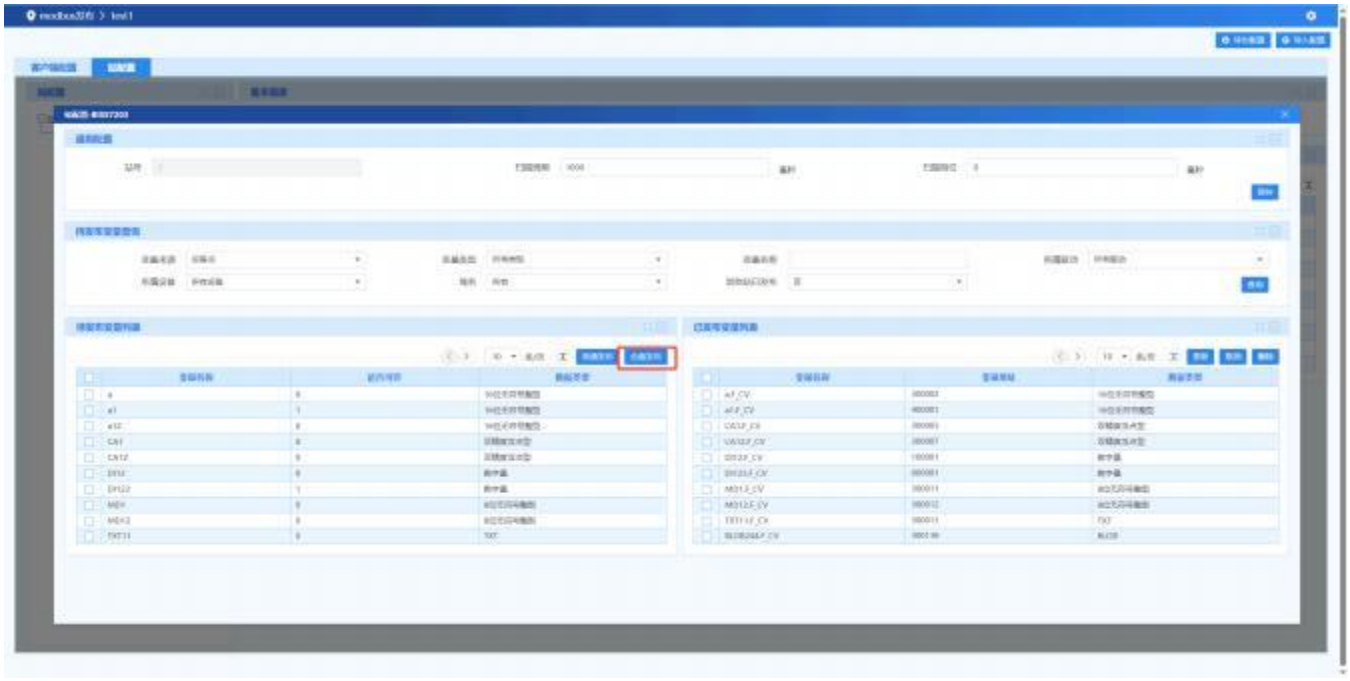
注意：
选中子系统和类，待发布变量列表中不会有对象点列出。

必须要选中类下挂的对象，待发布变量列表中才会列出此对象所有的变量。

对象点名由对象名称和属性名称组成（对象名称.属性名称），因此查询条件中将对象名称和属性分别查询。



sm 5.16.0版本新增一键发布全量发布



选择域名，和是否地址连续发布
1.发布当前查询条件下的所有没有被发布过的点

2.当没有勾选地址连续，类型地址不够时，多余的变量不进行发布

3.当勾选地址连续时，类型连续地址不够时。取消此次发布

4.已经发布在当前域名下的变量不进行重复发布

地址规则

起始地址默认值规则

弹出框中有一个默认地址值，这个默认值是通过首个选中变量的变量类型和是否可控来确定的，规则如下表所示：

变量类型	是否可控	初始地址默认值
DI,CD	0	100001
DI,CD	1	1
AI,MDI,TXT,BLOB,CA	0	300001
AI,MDI,TXT,BLOB,CA	1	400001

用户可以手动更改这个默认地址值。

地址范围

地址的可设置范围为：

[000001,065535][100001,165535][300001,365535][400001,465535]

这4个值区间对应Modbus的4个类型的信号；Modbus地址是个6位数，由1位信号类型及5位寄存器地址构成。

DO类型信号地址为：0XXXXX；（不可控）

DI类型信号地址为：1XXXXX；（可控）

AI类型信号地址为：3XXXXX；（可控）
AO类型信号地址为：4XXXXX；（不可控）

后5位数XXXXX为10进制数表达，根据modbus协议，每种信号类型我们可以使用的寄存器是65535个，即十六进制表达：FFFF个。1个寄存器是16bit，因此对于一个32位的变量，其长度为4个字节，占用2个寄存器地址。

地址自增规则

选中多个变量进行发布时，地址配置窗口中配置的起始地址是首个被选中变量的起始地址，其他变量地址偏移，系统会根据数据类型自动添加。

规则如下表：

da中的变量类型码	变量类型英文	变量类型中文	变量长度（字节）	地址自增长度
4	DT_BIT	数字量	1	1
9	DT_BLOB	BLOB	250	125
0	DT_ASCII	TXT	250	125
11	DT_UCHAR	8位无符号整型	1	1
10	DT_CHAR	8位有符号整型	1	1
2	DT_UINT16	16位无符号整型	2	1
1	DT_SINT16	16位有符号整型	2	1
6	DT_ULONG	32位无符号整型	4	2
7	DT_SLONG	32位有符号整型	4	2
3	DT_FLT	单精度浮点型	4	2
8	DT_DBL	双精度浮点型	8	4

SM中的变量类型和发布变量类型对应

sm中变量类型	SM中数据类型	发布的变量类型
BLOB	*	BLOB
TXT	*	TXT

AI	8位有符号整型	8位有符号整型
AI	8位无符号整型	8位无符号整型
AI	16位有符号整型	16位有符号整型
AI	16位无符号整型	16位无符号整型
AI	32位有符号整型	32位有符号整型
AI	32位无符号整型	32位无符号整型
AI	单精度浮点型	单精度浮点型
AI	双精度浮点型	双精度浮点型
DI	数字量	数字量
MDI	多态量	多态量
CA	累计量	累计量
CD	控制数字量	控制数字量

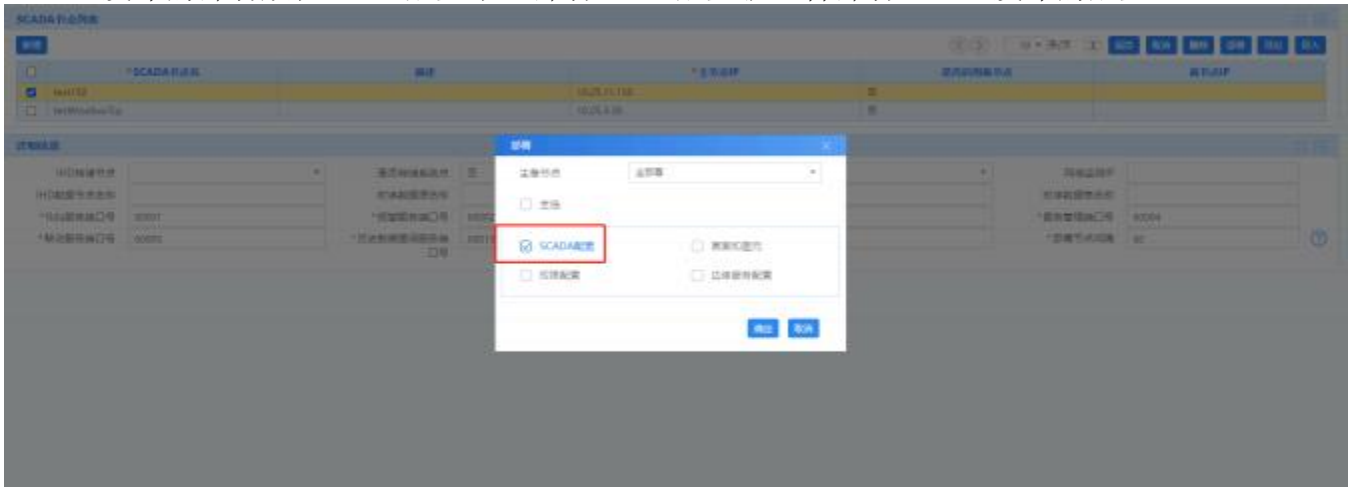
域

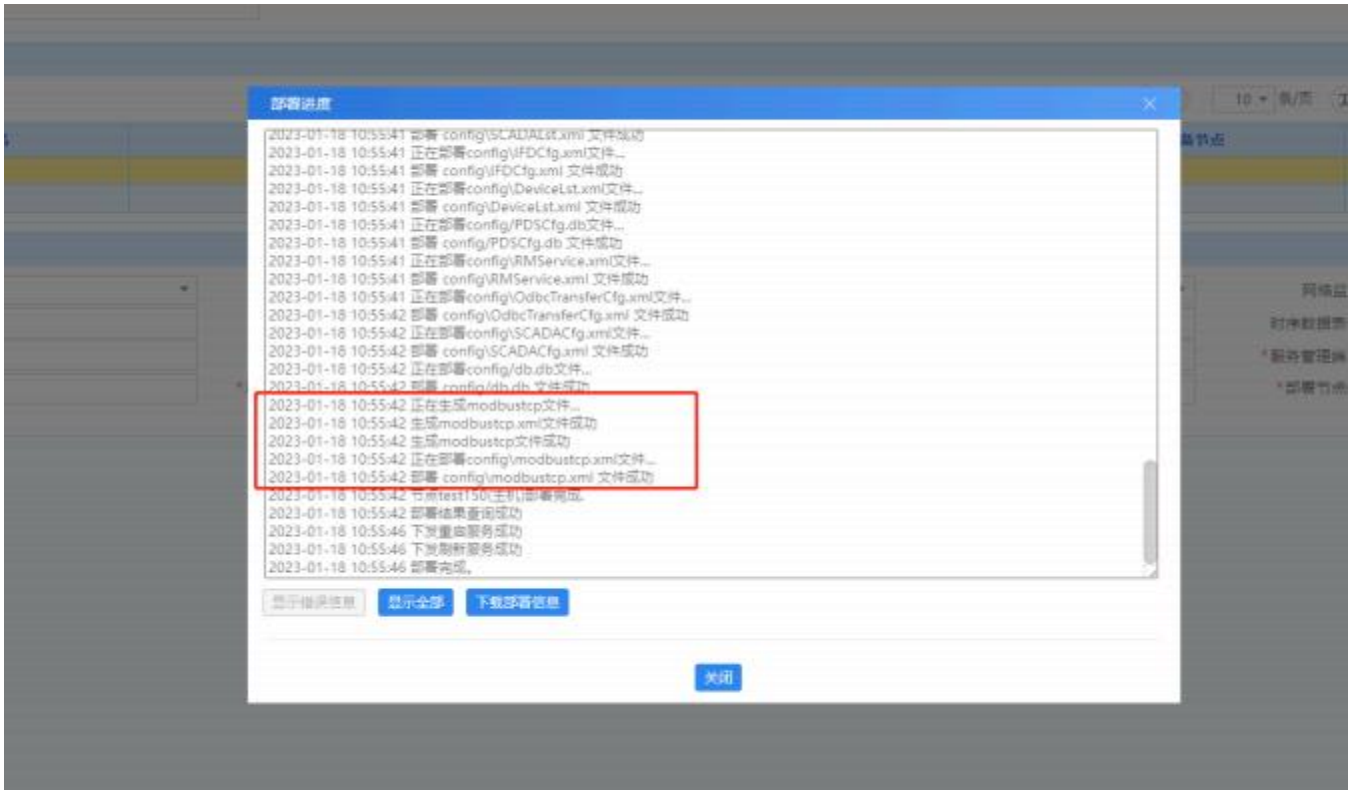
•域：有三种选项可以选择：

域名	意义	类型
F_CV	数据值	与数据类型相关
F_QUALITY	数据质量	16位无符号整型
F_TIMESTAMP	数据时间戳	32位无符号整型

部署

modbus发布的部署放在SCADA配置中。部署SCADA配置就包含部署modbus发布的配置。




部署进度
2023-01-18 10:55:41 正在部署config\IFDCfg.xml文件...
2023-01-18 10:55:41 部署 config\IFDCfg.xml 文件成功
2023-01-18 10:55:41 正在部署config\DeviceLst.xml文件...
2023-01-18 10:55:41 部署 config\DeviceLst.xml 文件成功
2023-01-18 10:55:41 正在部署config/PDSCfg.db文件...
2023-01-18 10:55:41 部署 config/PDSCfg.db 文件成功
2023-01-18 10:55:41 正在部署config\RMService.xml文件...
2023-01-18 10:55:41 部署 config\RMService.xml 文件成功
2023-01-18 10:55:41 正在部署config\OdbcTransferCfg.xml文件...
2023-01-18 10:55:42 部署 config\OdbcTransferCfg.xml 文件成功
2023-01-18 10:55:42 正在部署config\SCADACfg.xml文件...
2023-01-18 10:55:42 部署 config\SCADACfg.xml 文件成功
2023-01-18 10:55:42 正在部署config/db.db文件...
2023-01-18 10:55:42 正在生成modbustcp文件...
2023-01-18 10:55:42 生成modbustcp.xml文件成功
2023-01-18 10:55:42 生成modbustcp文件成功
2023-01-18 10:55:42 正在部署config\modbustcp.xml文件...
2023-01-18 10:55:42 部署 config\modbustcp.xml 文件成功
2023-01-18 10:55:42 部署结果查询成功
2023-01-18 10:55:46 下发重启服务成功
2023-01-18 10:55:46 下发刷新服务成功
2023-01-18 10:55:46 部署完成。
显示全部
下载部署信息
关闭

部署文件示例：

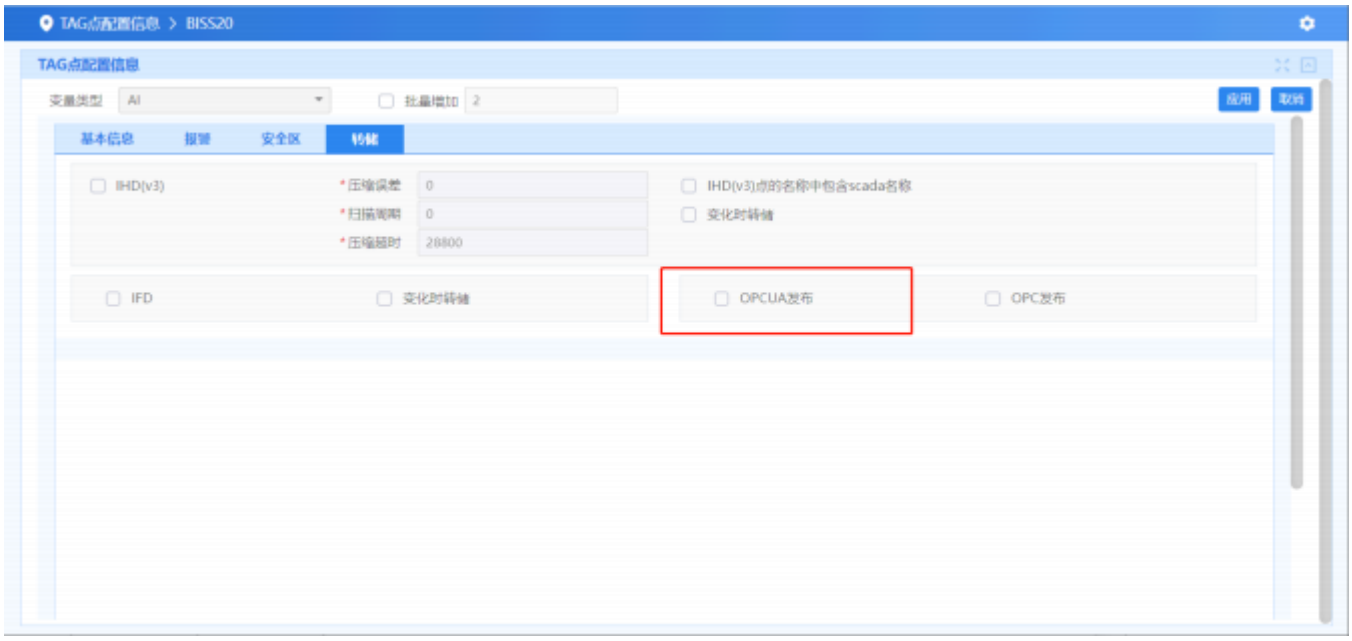
```xml
<?xml version="1.0" encoding="UTF-8"?>

<modbustcp port="502">
  <boundip iplimit="1"/>
  <stations>
    <station no="1" scanperiod="1000" phase="0">
      <ai>
        <tag name="blob.F_CV" address="300001" length="250" type="9"/>
        <tag name="ca.F_CV" address="300126" length="8" type="8"/>
        <tag name="di.F_CV" address="300130" length="1" type="4"/>
        <tag name="mdi.F_CV" address="300131" length="1" type="11"/>
        <tag name="ai_8u.F_TIMESTAMP" address="301001" length="4" type="6"/>
        <tag name="ai_8.F_TIMESTAMP" address="301003" length="4" type="6"/>
        <tag name="ai_f.F_TIMESTAMP" address="301005" length="4" type="6"/>
        <tag name="ai_d.F_TIMESTAMP" address="301007" length="4" type="6"/>
      </ai>
      <ao>
        <tag name="txt.F_CV" address="400001" length="250" type="0"/>
        <tag name="cd.F_CV" address="400126" length="1" type="4"/>
      </ao>
      <di>
        <tag name="ai_8u.F_CV" address="100001" length="1" type="11"/>
        <tag name="ai_8.F_CV" address="100002" length="1" type="10"/>
        <tag name="ai_f.F_CV" address="100003" length="4" type="3"/>
        <tag name="ai_d.F_CV" address="100005" length="8" type="8"/>
      </di>
      <do>
        <tag name="ai_16u.F_CV" address="000001" length="2" type="2"/>
        <tag name="ai_16.F_CV" address="000002" length="2" type="2"/>
        <tag name="ai_32u.F_CV" address="000003" length="4" type="6"/>
        <tag name="ai_32.F_CV" address="000005" length="4" type="7"/>
        <tag name="ai_16u.F_QUALITY" address="000020" length="2" type="2"/>
        <tag name="ai_16.F_QUALITY" address="000021" length="2" type="2"/>
        <tag name="ai_32u.F_QUALITY" address="000022" length="2" type="2"/>
        <tag name="ai_32.F_QUALITY" address="000023" length="2" type="2"/>
      </do>
    </station>
  </stations>
</modbustcp>
```

5.3 opcua发布

opcuaserver支持da中tag点变量通过opcua发布，例如各种驱动采集的数据可通过opcuaserver转发出来。

opcuaserver配置

在TAG点配置信息中的转储标签页下勾选OPCUA发布即可

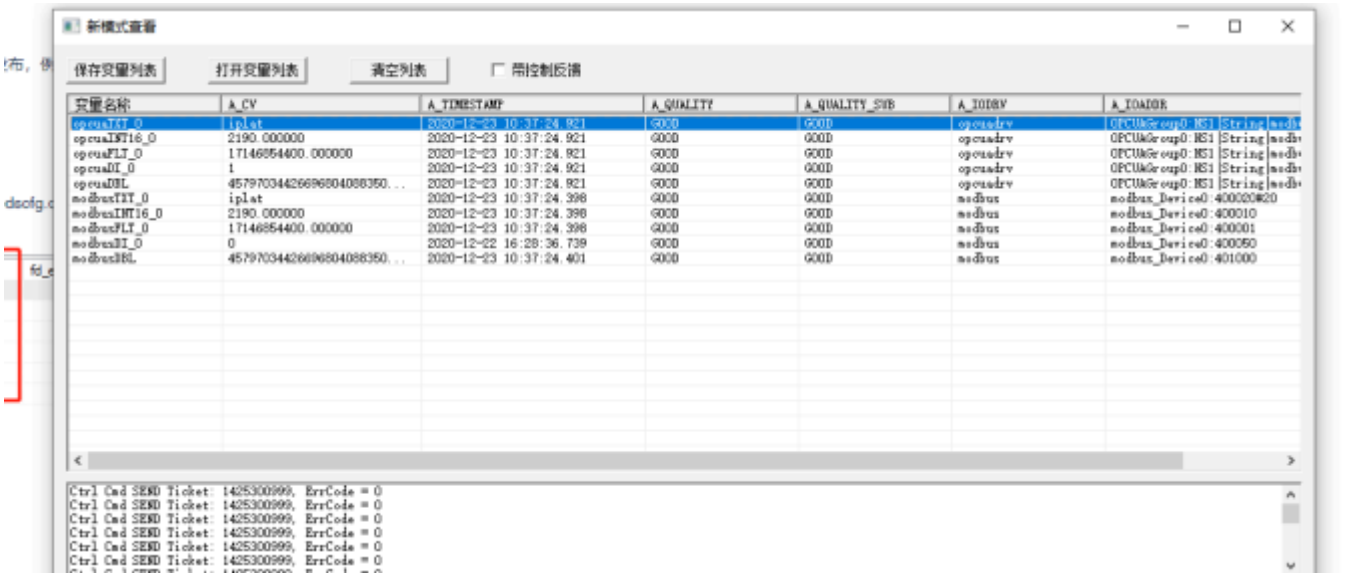


测试环境搭建

scada节点中配置opcua驱动，modbus驱动，将modbus采集的数据通过opcuaserver发布，然后再通过opcua驱动采集回来。

modbus配置ai点，di点，txt点变量，opcuadrv配置相同个数的变量，地址格式为NS1|String|modbusXXX，opcuadrv变量配置详见OPCUA变量配置

通过RDAMonitor观察配置的变量，可以看到modbus变量成功通过opcuaserver发布并再次从opcuadriver采集。也可通过opcuadrv改写opcuaserver发布的点的值，如下图



性能测试

硬件配置：

CPU:i5-7500U

内存：DDR4 16G

软件配置：

iplatdadriver配置：

10000点AI点，每个点的扫描周期配置为100ms，扩展参数2中配置"OPCUA",通过opcuaserver发布出来

opcuadrv配置：

10000点AI点，每个点扫描周期配置为100ms，每1000个点分一个设备，分配一个独立的任务号，即10个不同的设备连接同一个opcuaserver，用来提升性能opcuadrv.xml配置文件如下：

```
 <driver name="opcuadrv" type="item">
<device
connparam="opc.tcp://127.0.0.1:4840" recvtimeout="1000" param3="" name="OPCUAGroup0" c
onnt
<device
connparam="opc.tcp://127.0.0.1:4840" recvtimeout="1000" param3="" name="OPCUAGroup1" c
onnt
<device
connparam="opc.tcp://127.0.0.1:4840" recvtimeout="1000" param3="" name="OPCUAGroup2" c
onnt
<device
connparam="opc.tcp://127.0.0.1:4840" recvtimeout="1000" param3="" name="OPCUAGroup3" c
onnt
<device
connparam="opc.tcp://127.0.0.1:4840" recvtimeout="1000" param3="" name="OPCUAGroup4" c
onnt
<device
connparam="opc.tcp://127.0.0.1:4840" recvtimeout="1000" param3="" name="OPCUAGroup5" c
onnt
<device
connparam="opc.tcp://127.0.0.1:4840" recvtimeout="1000" param3="" name="OPCUAGroup6" c
onnt
<device
connparam="opc.tcp://127.0.0.1:4840" recvtimeout="1000" param3="" name="OPCUAGroup7" c
onnt
<device
connparam="opc.tcp://127.0.0.1:4840" recvtimeout="1000" param3="" name="OPCUAGroup8" c
onnt
<device
connparam="opc.tcp://127.0.0.1:4840" recvtimeout="1000" param3="" name="OPCUAGroup9" c
onnt
 </driver>
```

点表文件如下：

设备点表.xlsx

测试场景：

10.25.11.158

10.25.11.157（虚拟机）：

CPU:E5-2680 v2 2.8GHz

内存:8.00GB

本机

使用iplatdadriver模拟数据源，通过opcuaserver将模拟数据发布，再通过opcuadrv将数据采集回来

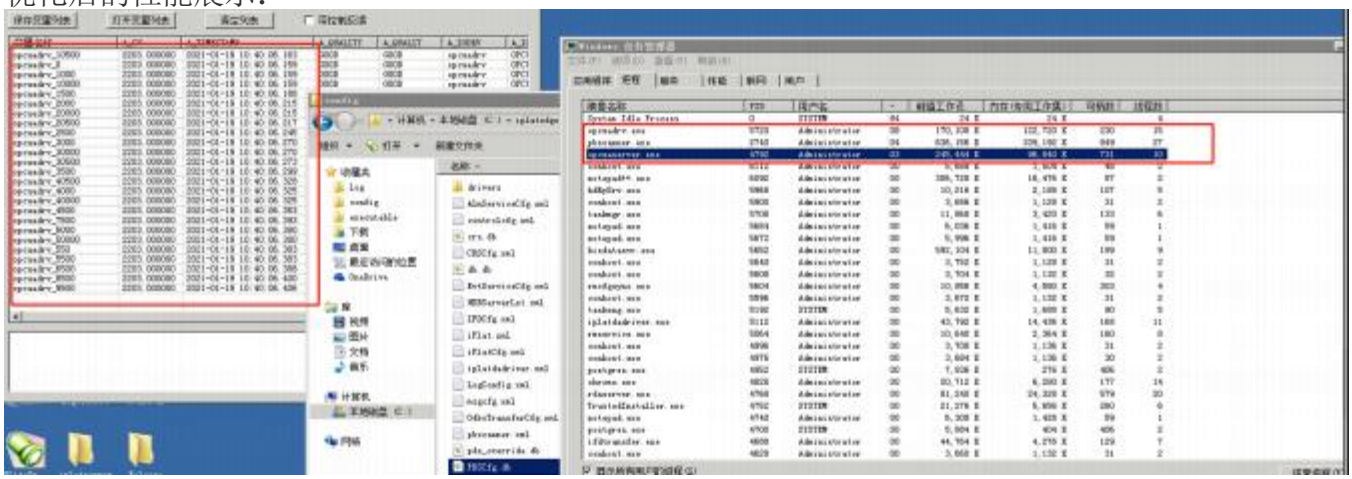
opcuaserver发布Tag点数量	opcuaserverTag点扫描周期(ms)	opcuadrvGroup分组	opcuad
50000	1000	50	500
50000	1000	20+20	500

50000	1000	20+10	500
10000	1000	1	500
10000	1000	10	1000
10000	1000	10	100
10000	100	10	1000
10000	100	10	500
10000	100	10	100

优化驱动框架性能，新增批量更新数据接口，在同一台scada上同时发布5w/10w点，同步采集5w/10w点，再次测试

opcuaserver发布Tag点数量	opcuaserverTag点扫描周期(ms)	opcuadrvGroup分组	opcuad
50000	1000	50	500
50000	1000	50	500

优化后的性能展示：



结论：

opcuaserver发布性能满足需求，但是通过opcuadrv采集回来时配置不妥可能会导致数据采集延迟和丢点情况，opcuaserver发布10000点，在opcuadrv配置10个任务号的情况下，最多支持500ms的发布周期，再往下就会出现丢点现象。

opcuaserver发布5w点时，可通过分布式部署多台scada多opcuadrv分别采集数据，单台scada的opcuadrv性能瓶颈较为明显，2w点以上会出现丢点，连接不上server的情况。

驱动框架性能优化后，单台scada同时发布10w点，同步采集10w点，不存在性能瓶颈。经过计算，采集的性能提升明显，5w点采集可在400ms以内完成，10w采集可在1000ms内完成。

OPC UA常见故障信息代码

下表包含了状态接线端返回的代码的说明。
info code描述
好0x00000000操作成功。
不确定0x40000000值不确定，但原因不明。
坏0x80000000值为坏，但原因不明。
Bad_UnexpectedError 0x80010000发生非预期错误。
Bad_InternalError 0x80020000编程或配置错误时发生内部错误。
Bad_OutOfMemory 0x80030000完成操作所需内存不足。
Bad_ResourceUnavailable 0x80040000操作系统资源不可用。
Bad_CommunicationError 0x80050000发生底层通信错误。
Bad_EncodingError 0x80060000正在序列化的对象中有非法数据，编码暂停。
Bad_DecodingError 0x80070000数据流中有非法数据，解码暂停。
Bad_EncodingLimitsExceeded 0x80080000栈设置的消息编码和解码限制已超出。
Bad_RequestTooLarge 0x80B80000请求消息大小超出了服务器设置的限制。
Bad_ResponseTooLarge 0x80B90000响应消息大小超出了客户端设置的限制。
Bad_UnknownResponse 0x80090000从服务器接收到的响应无法识别。
Bad_Timeout 0x800A0000操作超时。
Bad_ServiceUnsupported 0x800B0000服务器不支持请求的服务。
Bad_Shutdown 0x800C0000应用程序正在关闭，所以操作取消。
Bad_ServerNotConnected 0x800D0000由于客户端未连接至服务器，操作无法完成。
Bad_ServerHalted 0x800E0000服务器已停止，无法处理任何请求。
Bad_NothingToDo 0x800F0000客户端传入的操作列表中无任何元素，因此不会执行任何操作。
Bad_TooManyOperations 0x80100000请求指定的操作过多，无法进行处理。
Bad_TooManyMonitoredItems 0x80DB0000由于订阅中的监视项过多，无法处理请求。
Bad_DataTypeIdUnknown 0x80110000由于数据类型ID无法识别，扩展对象无法序列化或解序列化。
Bad_CertificateInvalid 0x80120000提供的证书不是有效参数。
Bad_SecurityChecksFailed 0x80130000验证安全时发生错误。
Bad_CertificateTimeInvalid 0x80140000证书已过期或尚未生效。
Bad_CertificateIssuerTimeInvalid 0x80150000发布方证书已过期或尚未生效。
Bad_CertificateHostNameInvalid 0x80160000用于连接服务器的主机名与证书中的主机名不匹配。
Bad_CertificateUriInvalid 0x80170000应用程序说明中指定的URI与证书中的URI不匹配。
Bad_CertificateUseNotAllowed 0x80180000证书可能无法用于请求的操作。
Bad_CertificateIssuerUseNotAllowed 0x80190000发布方证书可能无法用于请求的操作。
Bad_CertificateUntrusted 0x801A0000证书未被信任。
Bad_CertificateRevocationUnknown 0x801B0000无法确定证书是否已被吊销。
Bad_CertificateIssuerRevocationUnknown 0x801C0000无法确定发布方证书是否已被吊销。
Bad_CertificateRevoked 0x801D0000证书已被吊销。
Bad_CertificateIssuerRevoked 0x801E0000发布方证书已被吊销。
Bad_UserAccessDenied 0x801F0000用户无权限执行请求的操作。
Bad_IdentityTokenInvalid 0x80200000用户身份令牌无效。
Bad_IdentityTokenRejected 0x80210000用户身份令牌有效，但服务器拒绝该令牌。
Bad_SecureChannelIdInvalid 0x80220000指定的安全通道不再有效。
Bad_InvalidTimestamp 0x80230000时间标识超出了服务器允许的范围。
Bad_NonceInvalid 0x80240000 Nonce不是一个随机值，或长度不符合要求。
Bad_SessionIdInvalid 0x80250000会话ID无效。
Bad_SessionClosed 0x80260000会话已被客户端关闭。

Bad_SessionNotActivated 0x80270000由于未调用ActivateSession，无法使用会话。
Bad_SubscriptionIdInvalid 0x80280000订阅ID无效。
Bad_RequestHeaderInvalid 0x802A0000请求的标头无效或丢失。
Bad_TimestampsToReturnInvalid 0x802B0000要返回参数的时间标识无效。
Bad_RequestCancelledByClient 0x802C0000请求已被客户端取消。
Good_SubscriptionTransferred 0x002D0000订阅被转移至另一个会话。
Good_CompletesAsynchronously 0x002E0000处理过程将异步完成。
Good_Overload 0x002F0000由于资源限制，采样速度已降低。
Good_Clamped 0x00300000写入的值已被接受，但未限定范围。
Bad_NoCommunication 0x80310000已定义与数据源之间的通信，但尚未建立通信，无可用的最近已知值。
Bad_WaitingForInitialData 0x80320000等待服务器从底层数据源获取值。
Bad_NodeIdInvalid 0x80330000节点ID的语法无效。
Bad_NodeIdUnknown 0x80340000节点ID指向服务器地址空间中不存在的节点。
Bad_AttributeIdInvalid 0x80350000指定的节点不支持该属性。
Bad_IndexRangeInvalid 0x80360000索引范围参数的语法无效。
Bad_IndexRangeNoData 0x80370000指定的索引范围内无数据存在。
Bad_DataEncodingInvalid 0x80380000数据编码无效。
Bad_DataEncodingUnsupported 0x80390000服务器不支持为该节点请求的数据编码。
Bad_NotReadable 0x803A0000访问等级不允许读取或订阅该节点。
Bad_NotWritable 0x803B0000访问等级不允许写入节点。
Bad_OutOfRange 0x803C0000值超出范围。
Bad_NotSupported 0x803D0000不支持请求的操作。
Bad_NotFound 0x803E0000请求的项未找到，或搜索操作已结束但无结果。
Bad_ObjectDeleted 0x803F0000对象已被删除，无法使用。
Bad_NotImplemented 0x80400000求取的操作未执行。
Bad_MonitoringModeInvalid 0x80410000监视模式无效。
Bad_MonitoredItemIdInvalid 0x80420000监视ID未指向有效的监视项。
Bad_MonitoredItemFilterInvalid 0x80430000监视项的过滤参数无效。
Bad_MonitoredItemFilterUnsupported 0x80440000服务器不支持请求的监视项筛选器。
Bad_FilterNotAllowed 0x80450000监视筛选器无法与指定的属性配合使用。
Bad_StructureMissing 0x80460000强制性架构的参数丢失或为空。
Bad_EventFilterInvalid 0x80470000事件过滤器无效。
Bad_ContentFilterInvalid 0x80480000内容过滤器无效。
Bad_FilterOperatorInvalid 0x80C10000过滤器中提供了无法识别的运算符。
Bad_FilterOperatorUnsupported 0x80C20000提供的运算符有效，但服务器不支持该过滤器运算符。
Bad_FilterOperandCountMismatch 0x80C30000为过滤器运算符提供的操作数少于期望的操作数。
Bad_FilterOperandInvalid 0x80490000内容过滤器中使用的操作数无效。
Bad_FilterElementInvalid 0x80C40000引用的元素不是内容过滤器中的有效元素。
Bad_FilterLiteralInvalid 0x80C50000引用的字面值不是有效值。
Bad_ContinuationPointInvalid 0x804A0000连续点不再有效。
Bad_NoContinuationPoints 0x804B0000由于所有的连续点已被分配，无法处理该操作。
Bad_ReferenceTypeIdInvalid 0x804C0000引用类型ID无效。
Bad_BrowseDirectionInvalid 0x804D0000浏览方向无效。
Bad_NodeNotInView 0x804E0000节点不是视图的一部分。
Bad_ServerUriInvalid 0x804F0000 ServerUri不是有效的URI。

Bad_ServerNameMissing 0x80500000未指定ServerName。
Bad_DiscoveryUrlMissing 0x80510000未指定DiscoveryUrl。
Bad_SempahoreFileMissing 0x80520000客户端指定的信号量文件无效。
Bad_RequestTypeInvalid 0x80530000安全令牌请求类型无效。
Bad_SecurityModeRejected 0x80540000安全模式不满足服务器设定的要求。
Bad_SecurityPolicyRejected 0x80550000安全策略不满足服务器设定的要求。
Bad_TooManySessions 0x80560000服务器已达到最大会话数量。
Bad_UserSignatureInvalid 0x80570000用户令牌签名丢失或无效。
Bad_ApplicationSignatureInvalid 0x80580000使用客户端证书生成的签名丢失或无效。
Bad_NoValidCertificates 0x80590000客户端未向服务器提供至少一个有效且满足性能要求的软件证书。
Bad_IdentityChangeNotSupported 0x80C60000服务器不支持更改为会话分配的用户身份。
Bad_RequestCancelledByRequest 0x805A0000客户端通过Cancel服务取消了请求。
Bad_ParentNodeIdInvalid 0x805B0000父节点ID未指向有效的节点。
Bad_ReferenceNotAllowed 0x805C0000由于引用违反了数据模型设定的约束条件，无法创建引用。
Bad_NodeIdRejected 0x805D0000由于节点ID无效或服务器不允许客户端指定节点ID，请求的节点ID被拒绝。
Bad_NodeIdExists 0x805E0000请求的节点ID已被另一个节点使用。
Bad_NodeClassInvalid 0x805F0000节点类无效。
Bad_BrowseNameInvalid 0x80600000浏览器名称无效。
Bad_BrowseNameDuplicated 0x80610000浏览名称不是共享相同父节点关系的节点中的唯一名称。
Bad_NodeAttributesInvalid 0x80620000节点类的节点属性无效。
Bad_TypeDefinitionInvalid 0x80630000类型定义节点ID未引用正确的类型节点。
Bad_SourceNodeIdInvalid 0x80640000源节点ID未引用有效的节点。
Bad_TargetNodeIdInvalid 0x80650000目标节点ID未引用有效的节点。
Bad_DuplicateReferenceNotAllowed 0x80660000节点间的引用类型已被定义。
Bad_InvalidSelfReference 0x80670000服务器在该节点上不允许此类自引用。
Bad_ReferenceLocalOnly 0x80680000远程服务器引用的引用类型无效。
Bad_NoDeleteRights 0x80690000服务器不允许删除节点。
Uncertain_ReferenceNotDeleted 0x40BC0000服务器无法删除所有目标引用。
Bad_ServerIndexInvalid 0x806A0000服务器索引无效。
Bad_ViewIdUnknown 0x806B0000视图ID未指向有效的视图节点。
Bad_ViewTimestampInvalid 0x80C90000视图时间标识不可用或不受支持。
Bad_ViewParameterMismatch 0x80CA0000视图参数彼此不一致。
Bad_ViewVersionInvalid 0x80CB0000视图版本不可用或不受支持。
Uncertain_NotAllNodesAvailable 0x40C00000由于底层系统不可用，引用列表可能不完整。
Good_ResultsMayBeIncomplete 0x00BA0000服务器应跟踪远程服务器的节点引用，但并未跟踪。结果集可能不完整。
Bad_NotTypeDefinition 0x80C80000提供的节点ID不是自定义类型节点ID。
Uncertain_ReferenceOutOfServer 0x406C0000相对路径中要跟踪的引用之一指向了另一服务器上地址空间中的节点。
Bad_TooManyMatches 0x806D0000请求操作的返回匹配数量过多。
Bad_QueryTooComplex 0x806E0000请求操作需要的服务器资源过多。
Bad_NoMatch 0x806F0000请求的操作无匹配可返回。
Bad_MaxAgeInvalid 0x80700000最大年龄参数无效。
Bad_HistoryOperationInvalid 0x80710000历史详情参数无效。

Bad_HistoryOperationUnsupported 0x80720000服务器不支持请求的操作。
Bad_InvalidTimestampArgument 0x80BD0000定义的返回的时间标识无效。
Bad_WriteNotSupported 0x80730000服务器不支持写入所提供的值、状态和时间标识的组合。
Bad_TypeMismatch 0x80740000为属性提供的值与属性值的类型不匹配。
Bad_MethodInvalid 0x80750000方法ID未指向指定对象的方法。
Bad_ArgumentsMissing 0x80760000客户端未指定该方法的全部输入参数。
Bad_TooManySubscriptions 0x80770000服务器已达到最大订阅数量。
Bad_TooManyPublishRequests 0x80780000服务器已达到最大发布请求队列数量。
Bad_NoSubscription 0x80790000无订阅可用于该会话。
Bad_SequenceNumberUnknown 0x807A0000序列号对于服务器未知。
Bad_MessageNotAvailable 0x807B0000请求的通知消息不再可用。
Bad_InsufficientClientProfile 0x807C0000当前会话的客户端不支持订阅所需的一个或多个性能。
Bad_StateNotActive 0x80BF0000子状态机当前不是活动状态。
Bad_TcpServerTooBusy 0x807D0000服务器过于忙碌，无法处理请求。
Bad_TcpMessageTypeInvalid 0x807E0000标头中指定的消息类型无效。
Bad_TcpSecureChannelUnknown 0x807F0000 SecureChannelId和/或TokenId当前未被使用。
Bad_TcpMessageTooLarge 0x80800000标头中指定的消息大小过大。
Bad_TcpNotEnoughResources 0x80810000无足够的资源处理请求。
Bad_TcpInternalError 0x80820000发生内部错误。
Bad_TcpEndpointUrlInvalid 0x80830000服务器未识别指定的QueryString。
Bad_RequestInterrupted 0x80840000由于网络中断，无法发送请求。
Bad_RequestTimeout 0x80850000处理请求时发生超时。
Bad_SecureChannelClosed 0x80860000安全通道已关闭。
Bad_SecureChannelTokenUnknown 0x80870000令牌已过期或未被识别。
Bad_SequenceNumberInvalid 0x80880000序列号无效。
Bad_ProtocolVersionUnsupported 0x80BE0000应用程序没有兼容的协议版本。
Bad_ConfigurationError 0x80890000配置存在问题，影响了值的有效性。
Bad_NotConnected 0x808A0000变量应从另一个变量中接收值，但未如此配置。
Bad_DeviceFailure 0x808B0000生成值的设备/数据源存在故障，对值造成了影响。
Bad_SensorFailure 0x808C0000设备/数据源获取值的传感器存在故障。
Bad_OutOfService 0x808D0000数据源未正常运行。
Bad_DeadbandFilterInvalid 0x808E0000死区过滤器无效。
Uncertain_NoCommunicationLastUsableValue 0x408F0000与数据源通信失败。变量值为最后一个质量为Good的值。
Uncertain_LastUsableValue 0x40900000正在更新该值的操作已停止更新。
Uncertain_SubstituteValue 0x40910000该值为手动改写的运行值。
Uncertain_InitialValue 0x40920000该值为变量的初始值，变量通常从另一个变量接收值。
Uncertain_SensorNotAccurate 0x40930000该值位于一个传感器的限值。
Uncertain_EngineeringUnitsExceeded 0x40940000值超出了为该参数定义的值范围。
Uncertain_SubNormal 0x40950000该值从多个源中获取，包含的Good源数量小于所需数量。
Good_LocalOverride 0x00960000值已被覆盖。
Bad_RefreshInProgress 0x80970000条件刷新失败，或已有条件刷新操作正在运行。
Bad_ConditionAlreadyDisabled 0x80980000该条件已被禁用。
Bad_ConditionAlreadyEnabled 0x80CC0000该条件已被启用。
Bad_ConditionDisabled 0x80990000属性不可用，或该条件被禁用。
Bad_EventIdUnknown 0x809A0000指定的事件ID未被识别。
Bad_EventNotAcknowledgeable 0x80BB0000事件无法知悉。
Bad_DialogNotActive 0x80CD0000对话框条件未处于活动状态。
Bad_DialogResponseInvalid 0x80CE0000响应对于对话框无效。

Bad_ConditionBranchAlreadyAcked 0x80CF0000条件分支已被知悉。
Bad_ConditionBranchAlreadyConfirmed 0x80D00000条件分支已被确认。
Bad_ConditionAlreadyShelved 0x80D10000条件已被搁置。
Bad_ConditionNotShelved 0x80D20000条件当前未被搁置。
Bad_ShelvingTimeOutOfRange 0x80D30000搁置时间不在允许范围内。
Bad_NoData 0x809B0000请求的时间范围内或事件筛选条件下不存在数据。
Bad_BoundNotFound 0x80D70000未找到提供上/下边界值的数据。
Bad_BoundNotSupported 0x80D80000服务器无法检索到变量边界。
Bad_DataLost 0x809D0000由于采集开始/停止/丢失，数据已丢失。
Bad_DataUnavailable 0x809E0000由于存在未安装的卷、离线存档或磁带，或由于暂时不可用等类似原因，期望的数据在请求的时间范围内不可用。
Bad_EntryExists 0x809F0000由于存在匹配条目，数据或事件未成功插入。
Bad_NoEntryExists 0x80A00000由于不存在匹配条目，数据或事件未成功更新。
Bad_TimestampNotSupported 0x80A10000客户端请求历史数据所使用时间标识格式不受服务器支持。换言之，客户端请求了ServerTimestamp，而服务器仅支持SourceTimestamp。
Good_EntryInserted 0x00A20000数据或事件已成功插入历史数据库。
Good_EntryReplaced 0x00A30000历史数据库中的数据或事件已被成功替换。
Uncertain_DataSubNormal 0x40A40000该值从多个值中获取，包含的Good值数量小于所需数量。
Good_NoData 0x00A50000请求的时间范围内或事件筛选条件下不存在数据。
Good_MoreData 0x00A60000请求的时间范围内或事件筛选条件下存在更多数据。
Bad_AggregateListMismatch 0x80D40000请求的集合数量与请求的节点ID数量不匹配。
Bad_AggregateNotSupported 0x80D50000服务器不支持请求的集合。
Bad_AggregateInvalidInputs 0x80D60000由于输入数据无效，无法获取集合值。
Bad_AggregateConfigurationRejected 0x80DA0000指定节点的集合配置无效。
Good_DataIgnored 0x00D90000请求所指定的字段对于EventType无效，或无法被历史记录保存。
Good_CommunicationEvent 0x00A70000通信层已发起事件。
Good_ShutdownEvent 0x00A80000系统正在关闭。
Good_CallAgain 0x00A90000操作未完成，需重新调用。
Good_NonCriticalTimeout 0x00AA0000发生非关键超时。
Bad_InvalidArgument 0x80AB0000一个或多个参数无效。
Bad_ConnectionRejected 0x80AC0000无法与远程服务器建立网络连接。
Bad_Disconnect 0x80AD0000服务器已从客户端断开连接。
Bad_ConnectionClosed 0x80AE0000网络连接已关闭。
Bad_InvalidState 0x80AF0000由于对象被关闭、未初始化或处于其他无效状态，操作无法完成。
Bad_EndOfStream 0x80B00000无法移动至数据流结束之后。
Bad_NoDataAvailable 0x80B10000非阻隔数据流中当前没有数据可读取。
Bad_WaitingForResponse 0x80B20000异步操作正在等待响应。
Bad_OperationAbandoned 0x80B30000异步操作已被调用方放弃。
Bad_ExpectedStreamToBlock 0x80B40000数据流未返回所有的请求数据（可能由于数据流为非阻隔性数据流）。
Bad_WouldBlock 0x80B50000需要非阻隔性操作，但该操作会导致阻隔。
Bad_SyntaxError 0x80B60000值语法无需。
Bad_MaxConnectionsReached 0x80B70000由于所有变量连接正在使用，操作无法完成。

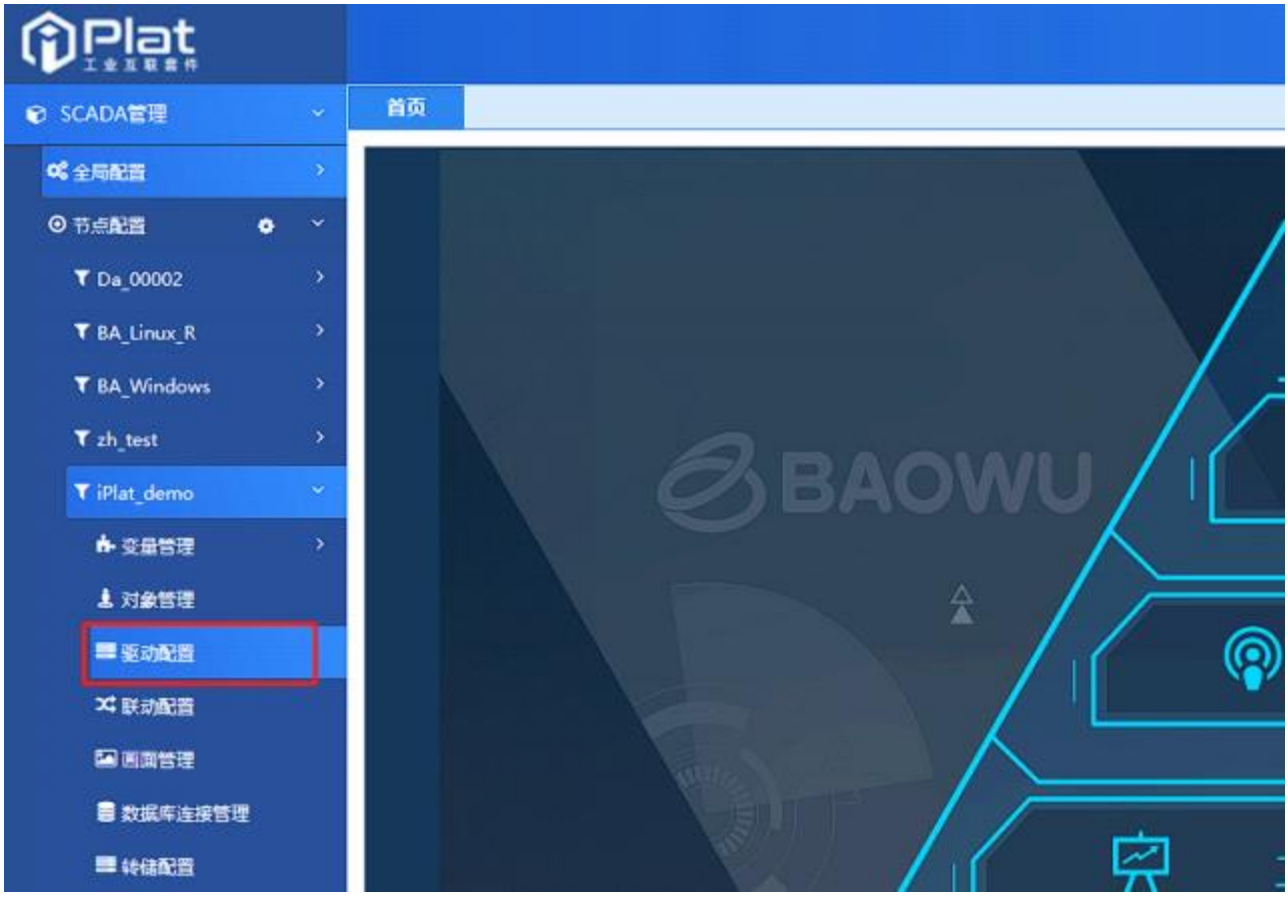
5.4 opc发布

iplatscada发布的OPCServer的名字为Baosight.iplatda配置的变量部署后可以在OPCServer中发布。
可以使用其他OPCClient软件来采集OPCServer发布的变量。
可以用scada自带的OPC驱动来采集OPCServer的变量。

需要注意的是：
AO/DO点是控制点而不是采集点，因此不支持OPC发布。
BLOB点不能进行OPC发布
AI/CA/LA点在OPCServer发布时，为double类型DI点在OPCServer发布时，为BOOL类型
TXT点在OPCServer发布时，为String类型MDI点在OPCServer发布时，为Byte类型

6. 联动介绍

1. 联动配置是边缘服务器的配置功能，包含预案配置和触发源配置两大部分。预案配置决定了一系列动作、执行顺序以及执行逻辑。触发源配置决定了触发条件。

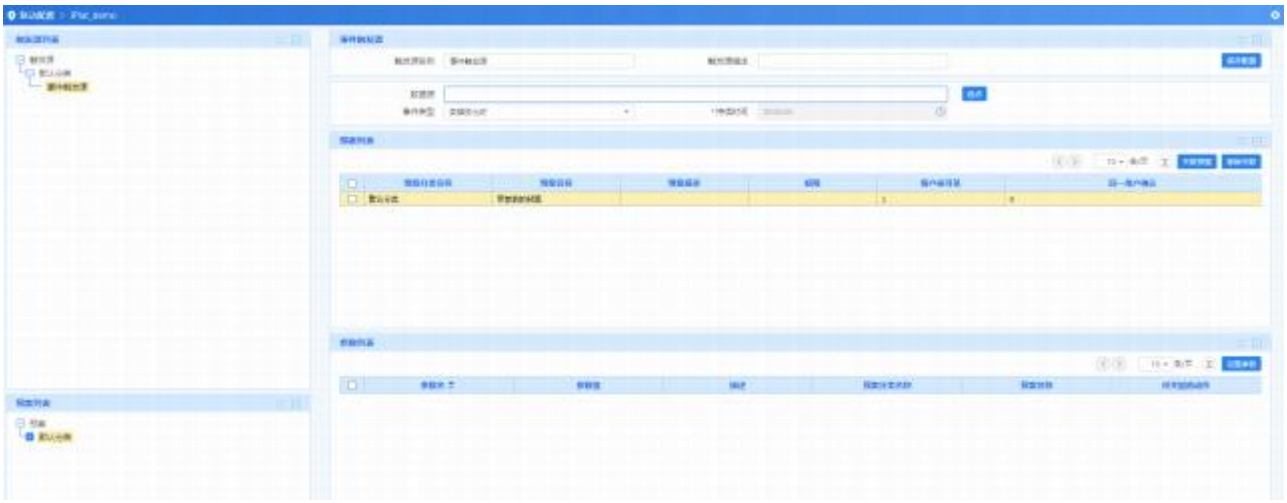


2. 展开边缘节点，点击"联动配置"即可弹出对应节点的配置界面。配置界面如下所示。

注意

触发源与预案配置界面相对独立，切换 tab 页不丢失当前的编辑。若两边加载的触发源和预案存在关联，则保存动作项后会自动更新参数列表。

联动配置将随着**部署**发往节点。

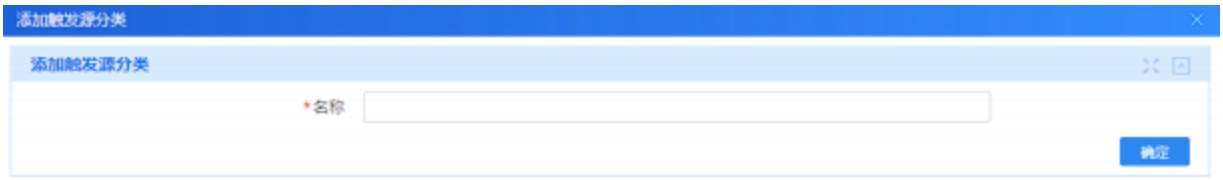


7.1触发源介绍

•触发源是指能够影响条件判断结果，导致预案执行的来源。触发源分类包括三类：**时间触发源**、**事件触发源**和**自定义触发源**。
•相关业务规则：
◦每个触发源可以同时对应到多个预案。
◦触发源满足触发条件时，相应的预案开始被执行。

7.2 触发源分类配置

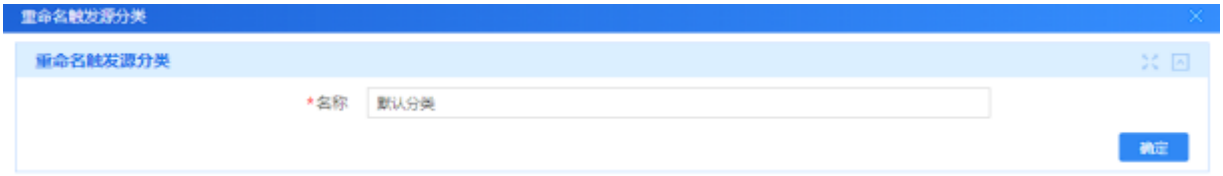
1. 新增触发源分类
•点击触发源，右键点击"添加触发源分类"，在弹出的对话框中输入分类名称，点击确定。分类名称不能和已有分类重名或者为空。分类名称不能和已有分类重名或者为空。



2. 删除触发源分类
•选中需要删除的分类，右键点击"删除触发源分类"按钮，弹出确认提示框。删除分类将会同时删除该分类下的所有触发源，以及这些触发源与预案的关联关系、参数列表，因此需要在确认对话框中输入分类的名称，点击确定将执行删除操作。

3. 重命名触发源分类

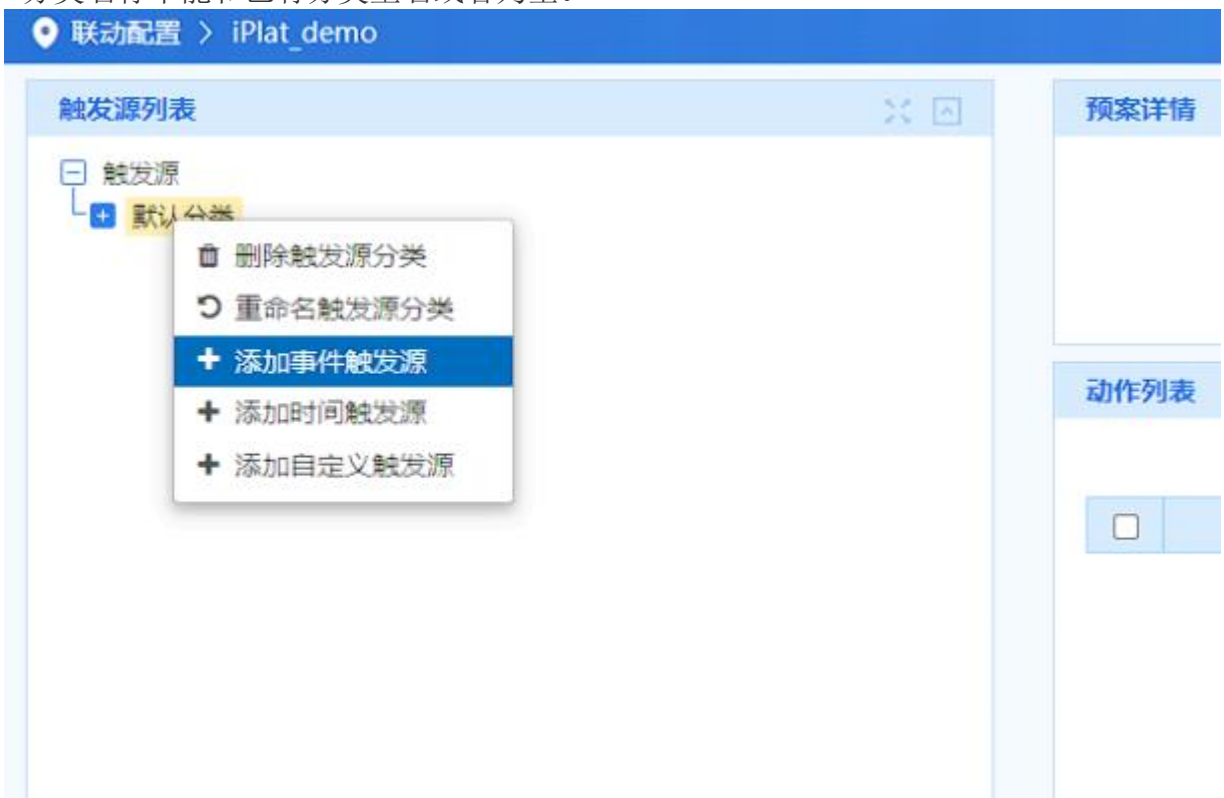
•选中需要重命名的分类，右键点击"重命名触发源分类"按钮，弹出名称编辑框。输入新的分类名称，点击确定。



7.3 触发源配置

1. 新增触发源

•点击左侧"默认分类"，右键添加触发源，可中选择一种触发源，弹出对应的配置界面。分类名称不能和已有分类重名或者为空。



。**触发源类型如下：**

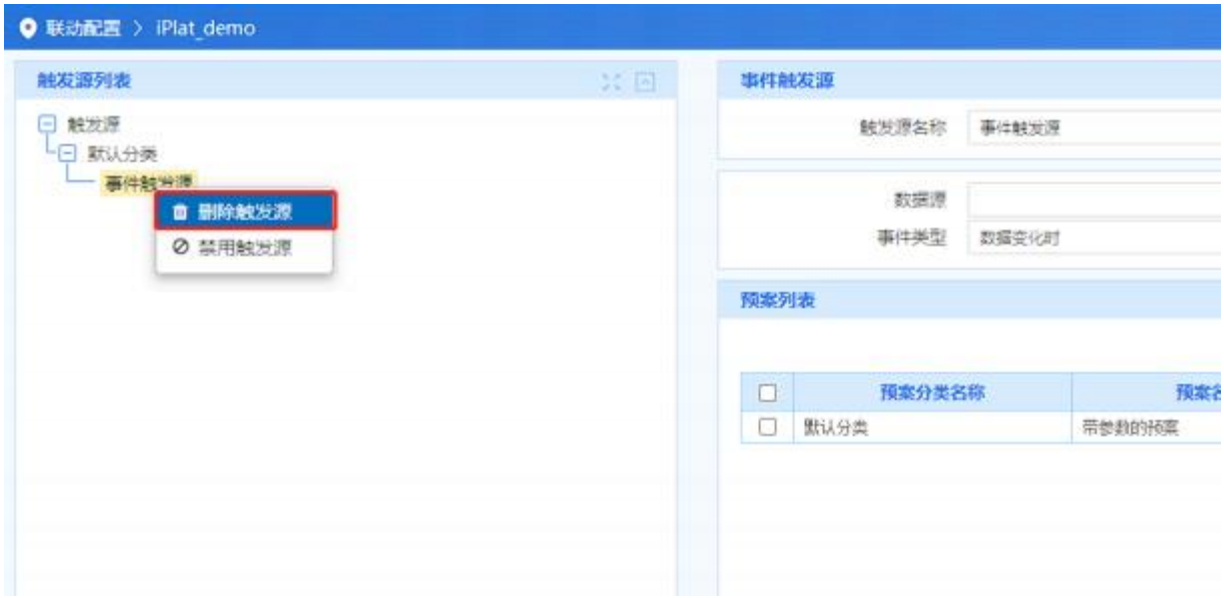
. 事件触发源

. 时间触发源

. 自定义触发源

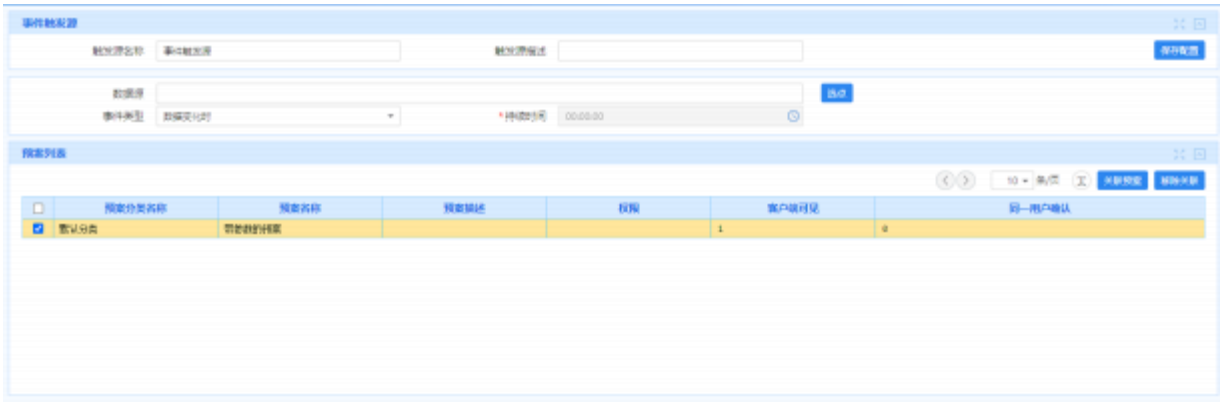
2. 删除触发源

•在左侧树节点中选择某个想要删除的触发源，右键点击删除触发源，点击确定按钮删除该触发源。



3. 修改触发源

•在左侧树节点中选择某个想要修改的触发源，触发源详情所提供的编辑框表示当前正在进行编辑，在修改完成后点击"保存配置"按钮，保存成功或者失败都会在页面顶端会有状态提示。



4. 禁用、启用触发源

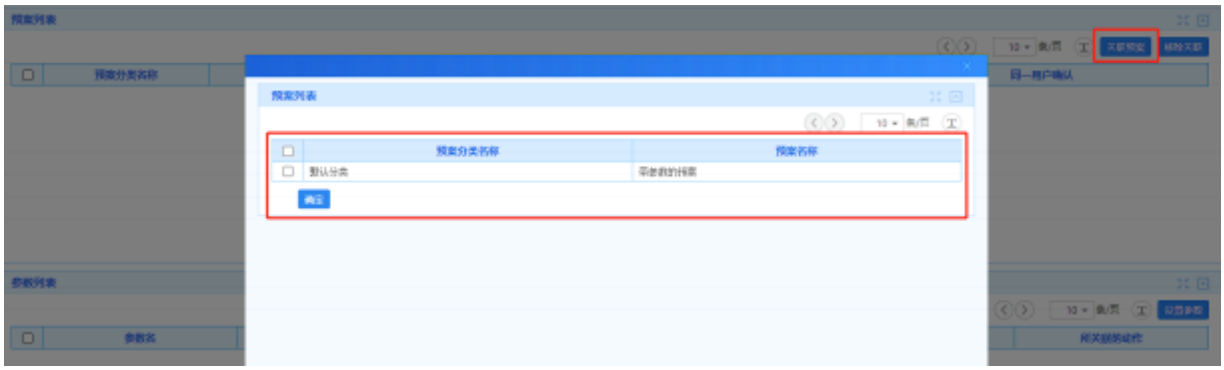
•对于配置的触发源，如果想要保留配置但不影响最终运行（即满足触发条件时不被执行），可以通过右键点击"禁用触发源"按钮禁用掉。对于已禁用的触发源，该按钮则显示为"启用触发源"。

5. 关联预案

•配置好预案后，点击"关联预案"会弹出所有未关联的预案。预案使用"分类/预案名"的形式显示。一个触发源可以关联一个或者多个预案。

•当某一个触发源被触发时，其对应的多个预案项会同时被执行。

•勾选需要关联的预案，点击确定后，已关联的预案列表以及参数列表将会被刷新。



•在已关联的列表中选中某条预案，点击移除按钮，则该条预案和相关联的参数都会被移除。本操作没有删除确认的提示框。

6. 配置动作参数

•用户在配置预案动作时，允许用参数名代替动作信息。这些参数在关联不同的触发源时可以独立配置。
•参数有两种格式：
-#{paraname}常量替换，即使用提供的参数替换该变量。
-${paraname}变量的值域替换，例如配置的是scada.tag1.F_CV，则预案被执行时，该参数会使用scada.tag1.F_CV当前的值替换。**注意：**值域替换不支持变量表达式，如果需要复杂的配置，可以关联CA变量或者在动作配置的时候编写。

7.4 事件触发源

1.事件一词指的是表达式值的状态。当这种状态满足一定条件的时候，执行预案的触发源叫做事件触发源。事件触发源支持的规则如下表：

类型	
数据发生变化时	触发源只能配置tag点域（例如scada_name.tag_name.F_CV）
表达式由假变为真	触发源可以配置表达式，当表达式的布尔值发生变化时，关联预案
表达式由真变为假	
表达式由假变为真并持续一段时间	触发源可以配置表达式，当表达式的的布尔值发生改变并持续一段
表达式由真变为假并持续一段时间	
表达式恒为真	触发源可以配置表达式，当表达式的布尔值在设定的时间内一直为
表达式恒为假	

2.表达式支持函数，详细参见变量表达式一览。例如：

- scada1.tagname_ai_0.F_CV>10
- scada2.tagname_ai_2.F_CV+scada1.tagname_ai_1.F_CV<200
- INT(scada1.tagname_ai_1.F_CV)>3 AND INT(scada1.tagname_ai_1.F_CV)<=10

3.新增触发源的步骤：
a.选中某个分类，点击新增触发源按钮后选择事件触发源。

b. 弹出配置对话框，如下图所示。

c. 修改名称，填入触发源名称（必填），选择事件类型、数据源选点。日期时刻按照事件类型和项目需要填写。
d. 点击"确定"按钮保存。

7.5 时间触发源

与时间相关的触发源，当时间满足条件时触发执行预案。
基本子类型包括单次触发源和连续触发源，扩展子类型包括每周、每月以及特殊日。详细说明如下：

类型	说明
单次	只在某一特定时刻到达时触发一次
连续	需要给定其实时间和间隔时间（即触发周期），从其实时间开始按照周期触发，可以设定结束
每周	可以分别设置每周中的哪几天有效。可以设定为单次触发，也可以设置为周期触发。如果设置

每月	可以设定每月中哪些日子有效，与每周触发源相似。如果设置为周期性触发，则同样不能跨天
特殊日	可以设定为触发源在特定的某一天有效，在特殊日当天，其他类型的时间触发源均无效。

7.6 自定义触发源

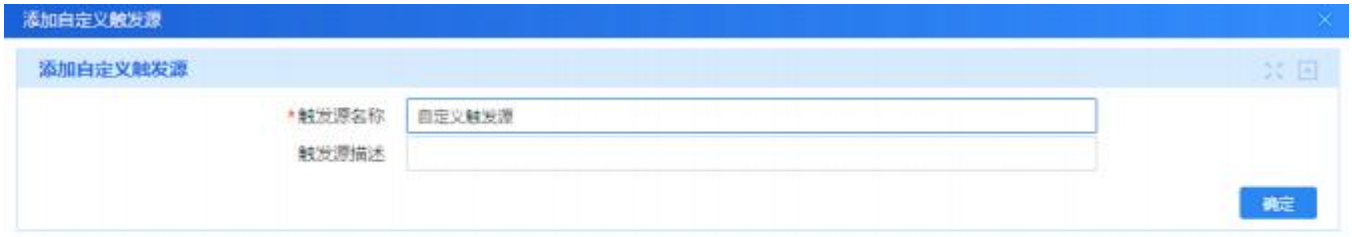
自定义触发源需要配置触发源的名称和描述，然后关联相关的预案触发条件不需要配置，何时触发由用户自己定义。

自定义触发源主要内容：

1. 配置自定义触发源
2. 自定义触发源的触发

1. 配置自定义触发源

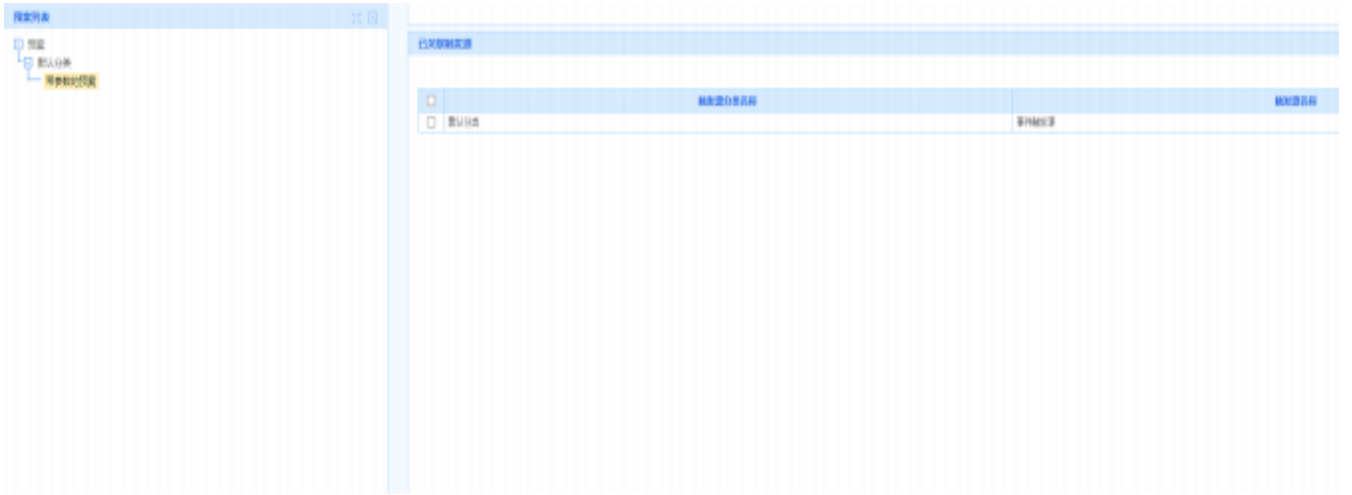
1.1. 用户在左边触发源管理树中选中一触发源分类，右键点击【增加自定义触发源】，弹出自定义触发源配置界面，如下图：



1.2. 用户在【触发源名称】中输入触发源的名称，点击确定按钮。

1.3. 关联预案，只要配置好预案，才可关联触发源，可右键点击添加预案。配置好预案后，点击【关联预案】按钮，弹出【触发源关联预案】对话框进行关联预案的操作，详情见触发源管

理中的关联预案。自定义触发源的出发一般取决于所关联的预案的相关动作条件。如下图：



自定义触发源的触发

•自定义触发源除了可以手动触发外，还可以使用组态画面中的脚本来触发。

7.7 预案介绍

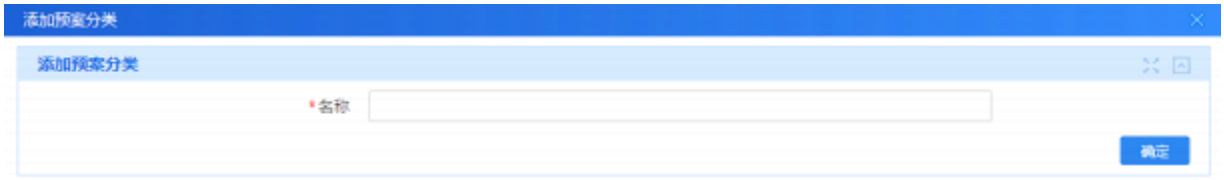
预案是指一组预先定义的动作。当触发源满足条件时，执行相应预案，预案所包含的动作将被**依次**执行。

业务名词	解释
优先级	仅用于指示该预案的重要程度，不构成执行顺序上的影响。
有效性	即预案是否被禁用，未被禁用的预案才会被触发源调用。
客户端可见	（暂未实现）若预案被设置为客户端可见，则所有运行态联动客户端都可以看到预案的执
单一确认	（暂未实现）若预案被设置为单一确认，则该预案中需要确认的预案项须由同一客户端确

7.8 预案分类配置

1. **添加预案分类**

•点击预案，右键点击添加"添加预案分类"，在弹出的对话框中输入分类名称，点击确定。分类名称不能和已有分类重名或者为空。

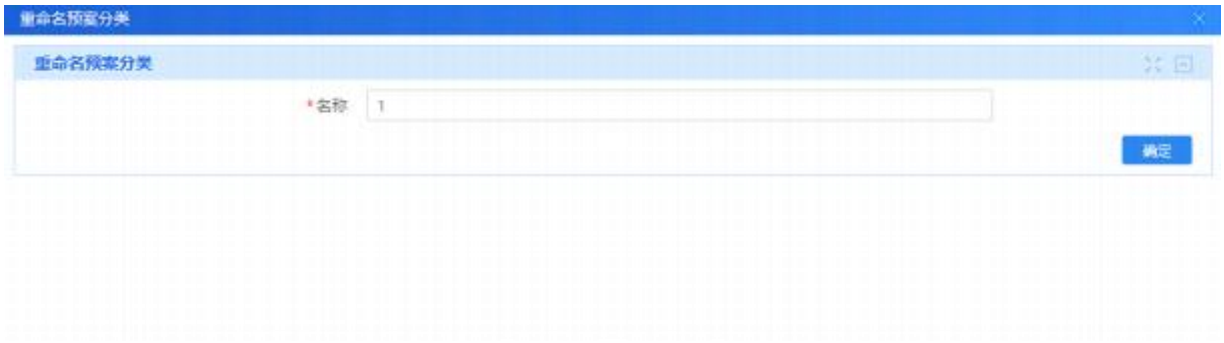


2. **删除预案分类**

•选中需要删除的分类，右键点击"删除预案分类"，弹出确认提示框。删除分类将会同时删除该分类下的所有预案、预案项，以及这些预案与触发源的关联关系、参数列表，因此需要在确认对话框中输入分类的名称，点击确定将执行删除操作。

3. **重命名预案分类**

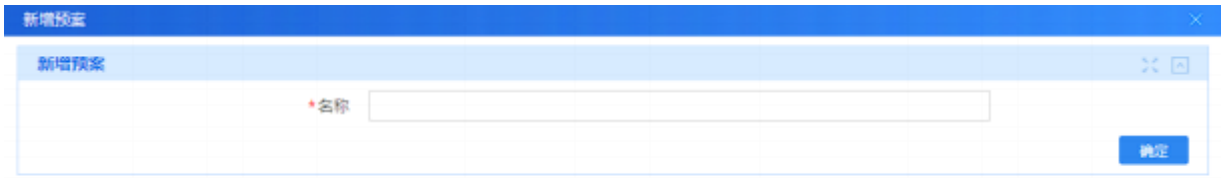
•选中需要重命名的分类，右键点击"重命名预案分类"按钮，弹出名称编辑框。输入新的分类名称，点击确定。



预案配置

1. **新增预案**

•左侧树节点选中某个分类，右键点击"添加预案"按钮，将在该分类下新增默认方式命令的预案。



2. **删除预案**

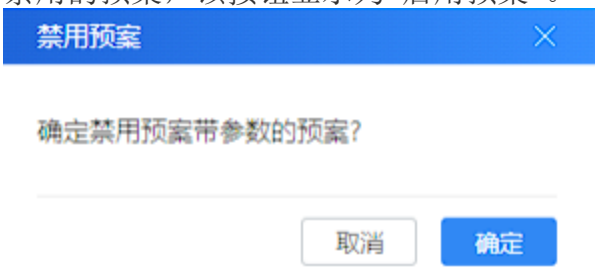
•选中需要删除的预案，右键点击"删除预案"，弹出确认对话框。和删除预案分类一样，相关联的数据也会被移除。

3. **修改预案**

•预案详情提供了编辑修改预案的功能，点击保存按钮即可更新预案名称、描述、权限、客户端可见、同一用户确认以及优先级。添加、删除、修改动作无需点击保存。

4. **禁用预案**

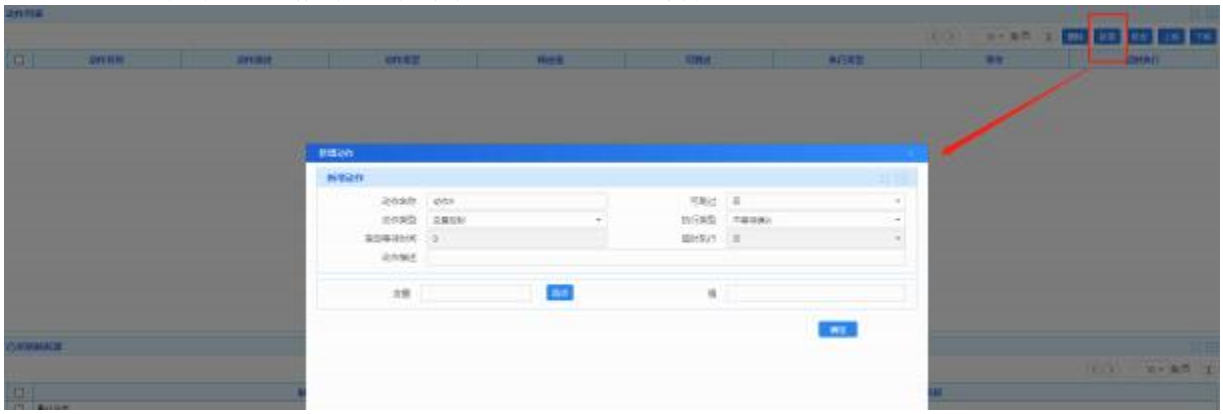
•选中某个预案，右键点击"禁用预案"按钮，则该条预案在部署后并不会被执行。对于已禁用的预案，该按钮显示为"启用预案"。



7.9 动作配置

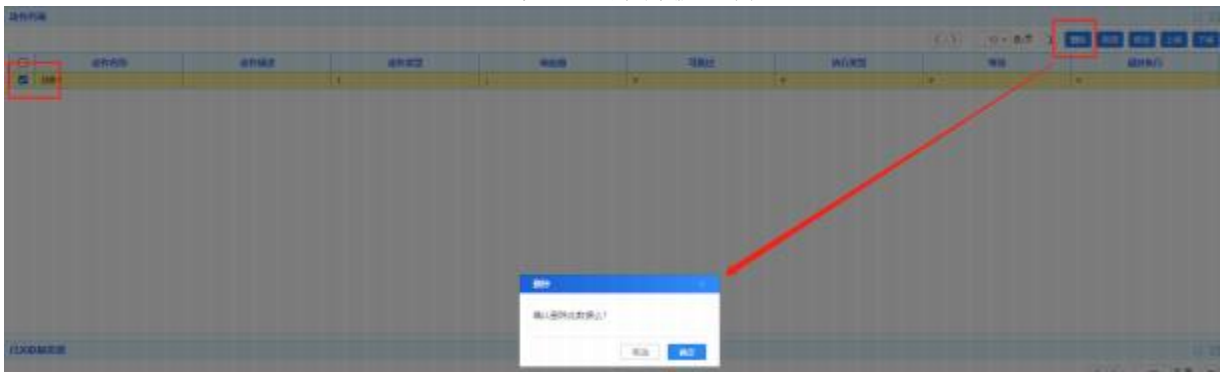
1. 新增动作

•在动作列表中，右键点击"新增"按钮，将在当前动作列表的结尾追加一个使用默认值的动作项，其类型为变量控制。表格中的数据值在配置界面会有对应的配置项。



2. **删除动作**

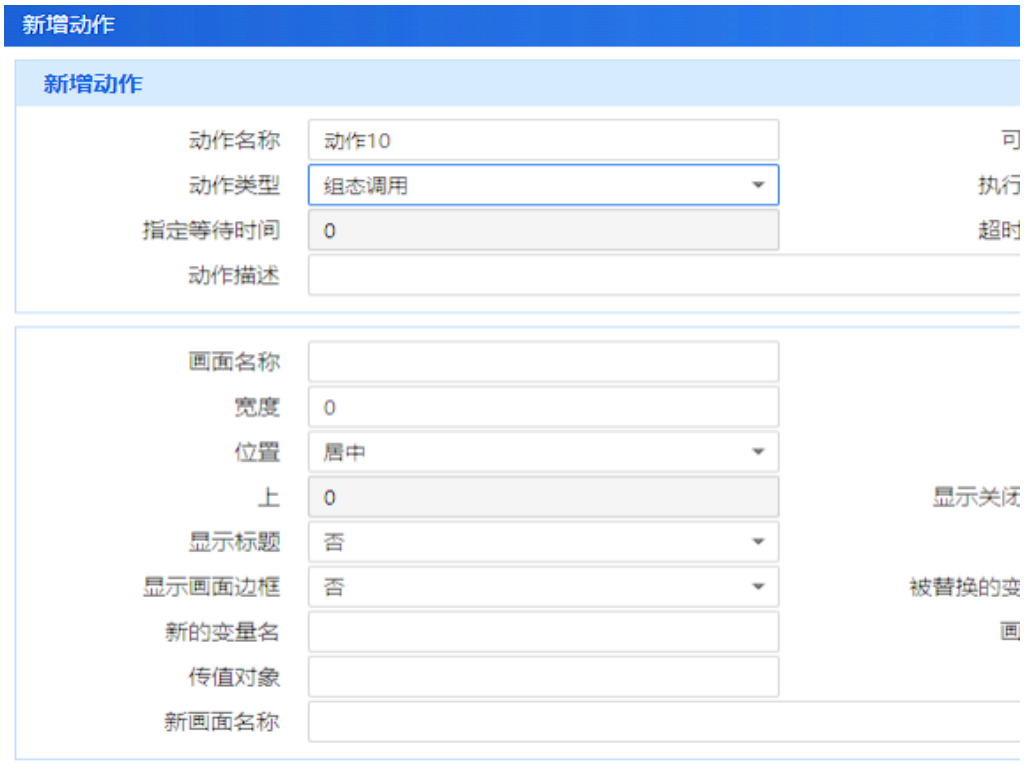
•选中需要删除的动作，点击"删除"按钮，将会弹出确认删除对话框。若该动作包含参数，则所有与该预案关联的触发源对应的该动作参数配置都将被删除。



3. **修改动作**

•选中需要的动作，点击"修改"按钮，将弹出修改动作的对话框。各种动作公共的配置项如下：

配置项	
动作名称	在集中配置以及监视客户端（暂未实现）下显示该动作。
动作描述	更多的描述说明。
可跳过	选择该选项则预案执行错误或是没有选择超时执行的时候，可跳过该动作继续执行
执行类型	包含不等待、等待指定时间、一直等待三个类型。指预案是否等待用户手动执行以
	1.等待时间为10秒，选择**超时执行**则等待10秒后执行，不选择**超时执行**则等待10秒
	2.如果第二个动作需要在第一个动作执行后等待5秒再执行，可在第二个动作上配置
动作类型	包括**变量控制：**
	组态调用：


新增动作
新增动作
动作名称
动作10
动作类型
组态调用
指定等待时间
0
动作描述
画面名称
宽度
0
位置
居中
上
0
显示标题
否
显示画面边框
否
新的变量名
传值对象
新画面名称

延迟：

新增动作
新增动作
动作名称
动作10
动作类型
延时
指定等待时间
0
动作描述
延时时间

SQL语句执行：

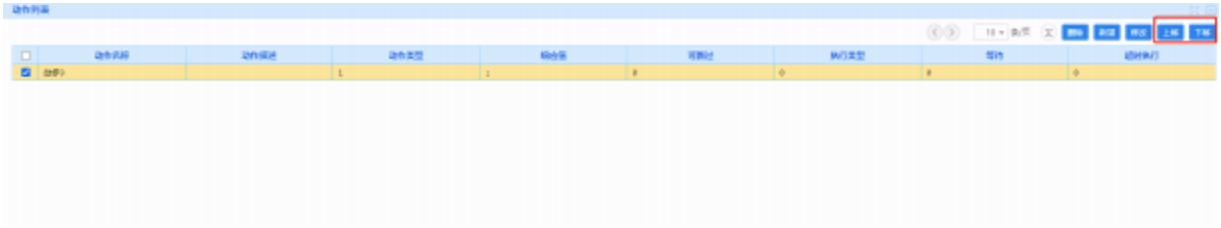
服务端可执行程序：

组态URL调用：

指定等待时间	默认指定
超时执行	默认指定
变量	可选择需要点进行绑定
值	配置值的大小

4. **移动动作**

- 已配置好的动作可以调整执行的先后顺序，选中需要调整的动作，点击上移或者下移按钮。

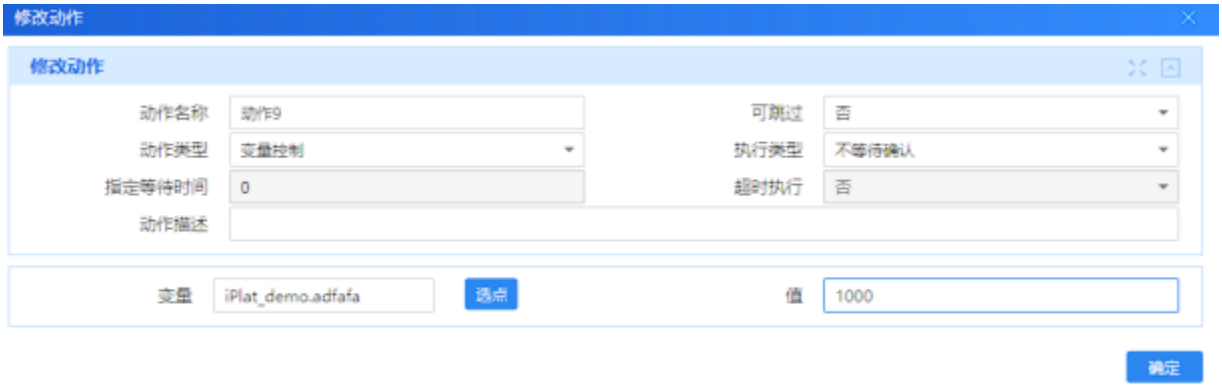


7.10 变量控制

变量控制是一种在边缘服务器执行的动作，用于给变量点赋值。注意：如果下发指令到BLOB或者TXT变量，则下发内容不能超过512字节。

1. **基础配置**

•在变量里填入需要控制的变量（使用三段式的需要具体到值域），并在值一栏中输入值，如下图所示：



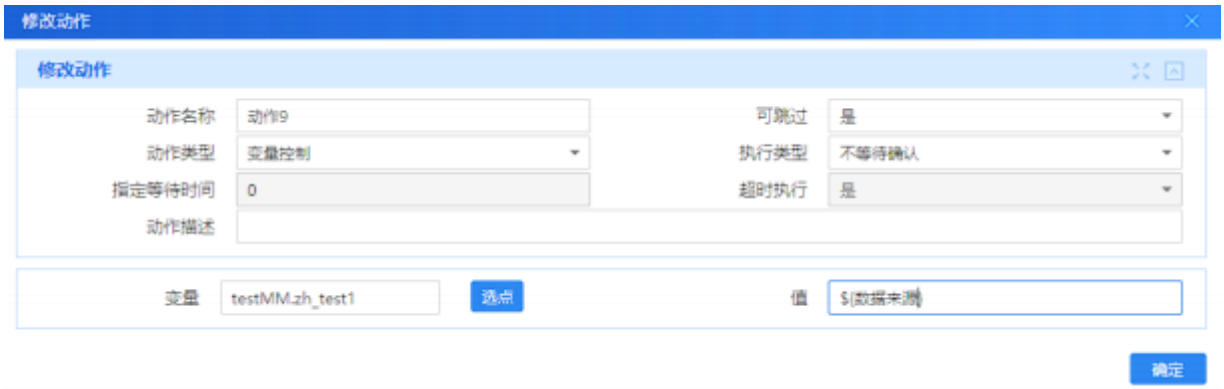
2. **设备变量的值为特定的字符串**

•对于TXT变量，可以在不同的触发源关联中结合实际给同一个动作配置不同的字符串。例如动作配置中给出：

•上图的示例可以用于和时间触发源关联，例如结合本年度的国家法定假日安排，在组态中显示特定的日期提醒。例如参数配置：

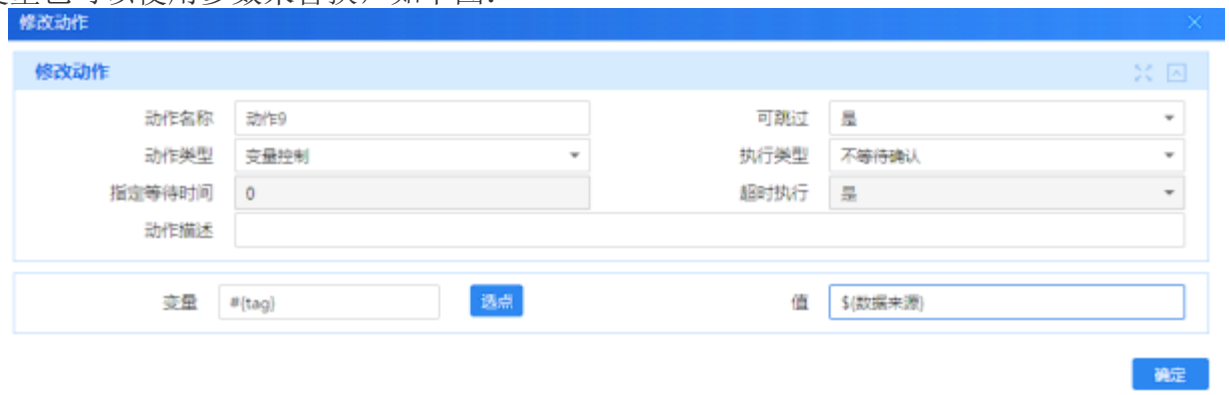
3. **使用变量的值控制另一个变量**

•在触发源配置里设置参数的时候，参数值可以填写另一个变量点，例如testMM.zh_test1。



4. **被控制的tag点位变量**

•变量也可以使用参数来替换，如下图：

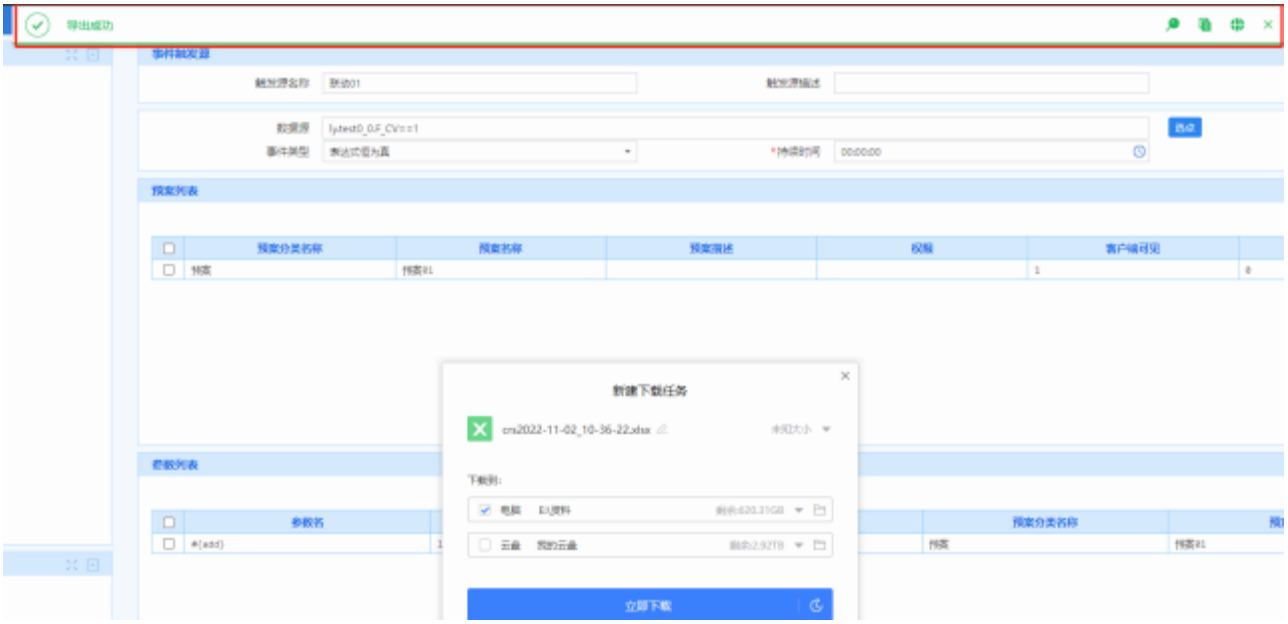


联动配置导出导入

联动配置的导出

1. 在"联动配置"页面，点击右上角的齿轮按钮，点击"导出"按钮。

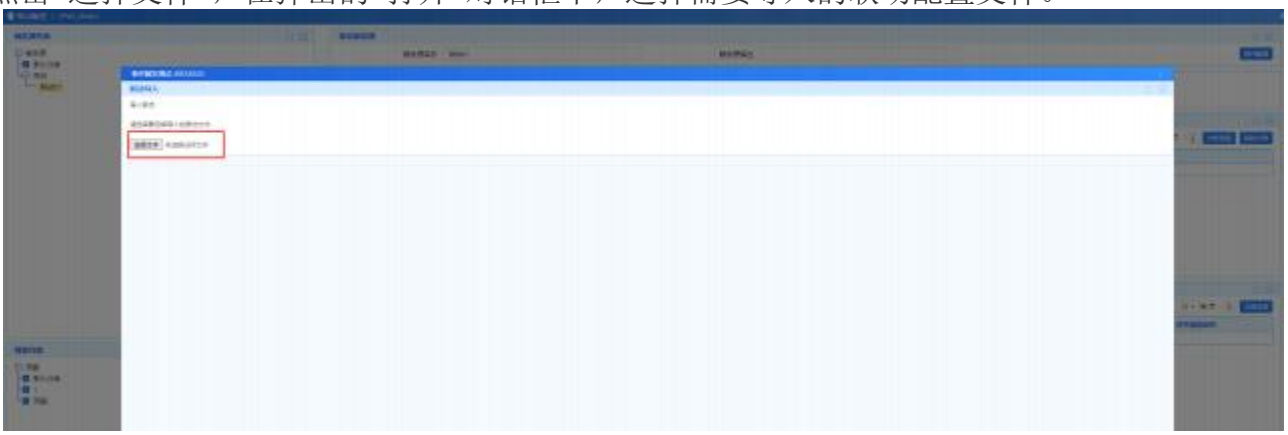
2. 页面上方显示导出成功，则联动配置文件成功导出。



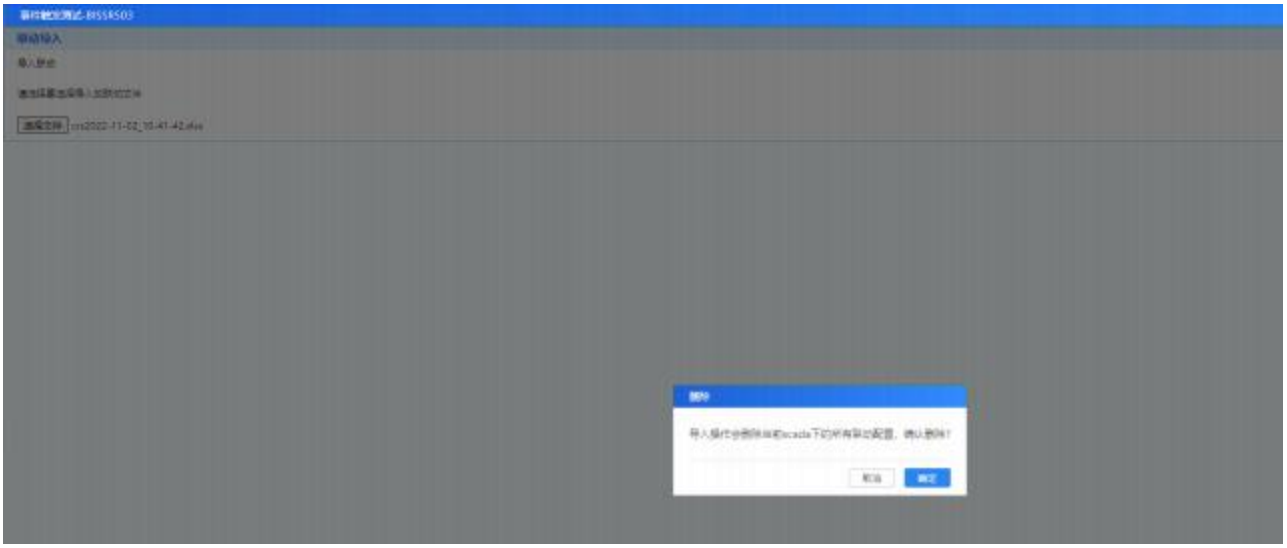
联动配置的导入

1. 在"联动配置"页面，点击右上角的齿轮按钮，点击"导入"按钮。

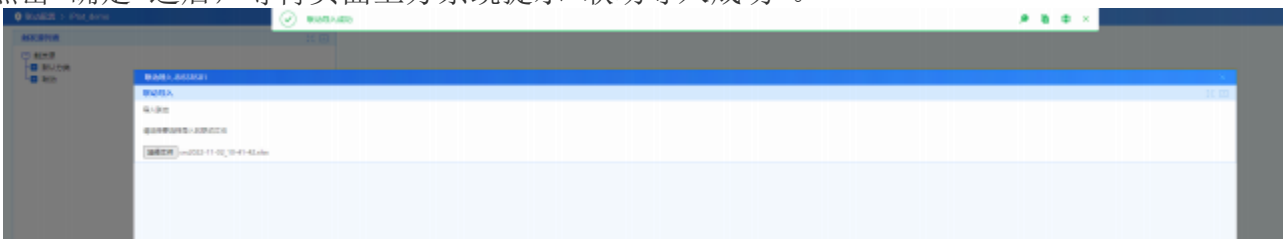
2. 点击"选择文件"，在弹出的"打开"对话框中，选择需要导入的联动配置文件。



3. 在弹出的"打开"对话框中，选择需要导入的联动配置文件，点击"打开"，弹出"删除"确认对话框；点击"确定"，执行删除原有的联动配置，并导入新的联动配置；



4. 点击"确定"之后，等待页面上方系统提示"联动导入成功"。



4.1刷新当前页面，在触发源列表中，可以看到已经导入的触发源，点击导入的触发源，可以看

4.2到触发源信息和详细配置；点击"预案"，在预案列表中，可以看到已经导入的预案，点击导

4.3入的预案，可以看到预案信息和详细配置。

联动配置导出文件说明

联动配置导出Excel文件，共有7个sheet页：

•动作配置："预案"中添加的动作。

◦预案名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；

◦动作顺序：依次顺序添加；

◦可跳过：0为可跳过该动作，执行下一动作；1为不可跳过该动作，必须执行完当前动作，才可以执行下一动作；

◦执行类型：0为"不等待确认"；1为"等待指定确认时间"；2为"一直等待确认"；

◦等待确认时间：整数，单位为秒(s)，仅在执行类型配置为1"等待指定确认时间"时可配置；对应动作配置界面的"指定等待时间"；

◦超时执行：0为否，1为是；仅在执行类型配置为1"等待指定确认时间"时可配置；

◦动作类型与响应值：

动作类型	动作类型ID	响应值格式	响应值举例
变量控制	1	变量;值	`'Scada.Tag.F_CV;1'`或者`'{#T`
组态调用	2	/page:,组态画面名称;参数	
延时	3	延时时间	10
服务端可执行程序	11	可执行程序路径	cmd.exe

组态URL调用	12	/url:	

◦动作名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；
◦动作类型执行的HMI列表：若需所有HMI则设为all；若不需要HMI则不需设置；若需指定的HMI则设成该HMI的名字；若需多个指定的HMI则在多个指定的HMI之间添加","；
◦动作默认执行时间：只有动作类型为"延时"的时候，该列设置为实际时间，单位秒(s)，其余都设为0；
◦动作描述：名称自定义，长度不得超出64位字符，32位中文；
◦动作是否有效：0为无效，1为有效；
◦预案分类名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"预案分类"中的"预案分类名称"一致；

•预案分类
◦预案分类名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"动作配置"中的"预案分类名称"一致。

•预案配置
◦预案名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"动作配置"中的"预案名称"一致；
◦预案描述：名称自定义，长度不得超出64位字符，32位中文；
◦预案分类名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"预案分类"和"动作配置"中的"预案分类名称"一致；
◦可以手动触发：0为不允许；1为允许；
◦权限：可以设置系统中所存在的群组；
。预案优先级ID：0为低；1为中；2为高；
。执行过程可见：0为不可见；1为可见；
。执行过程可终止：0为不可终止；1为可终止；
。必须同一用户确认：0为否；1为是；
。是否启用：0为不启用；1为启用；

•触发源分类
。触发源分类名称：名称自定义，长度不得超出64位字符，32位中文；

•触发源配置
- 。触发源类型ID：1为时间触发源；2为事件触发源；3为自定义触发源；
- 。触发源名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；
- 。触发源描述：名称自定义，长度不得超出64位字符，32位中文；
- 。响应值：下表；
- 。触发源分类名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"触发源分类"中的"触发源分类名称"一致；
- 。是否启用：0为不启用；1为启用；
- 。响应值：

触发源类型	触发源类型ID	触发类型	触发类型ID	
时间触发	1	一次	0	触发类

时间触发	1	连续	1	触发类
时间触发	1	每周	2	触发类
时间触发	1	每月	3	触发类
时间触发	1	特殊日	4	触发类
事件触发	2	数据变化时	0	触发类
事件触发	2	表达式由假变为真；	1	触发类
事件触发	2	表达式由真变为假；	2	触发类
事件触发	2	表达式由假变为真并持续一段时间；	3	触发类
事件触发	2	表达式由真变为假并持续一段时间；	4	触发类
事件触发	2	表达式恒为真；	5	触发类
事件触发	2	表达式恒为假；	6	触发类
自定义触发	3	无	无	不需设

触发源和预案关联关系

•触发源名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"触发源配置"中的"触发源名称"一致；

•预案名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"动作配置"和"预案配置"中的"预案名称"一致；

•预案分类名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"动作配置"、"预案分类"和"预案配置"中的"预案分类名称"一致；

•触发源分类名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"触发源分类"和"触发源配置"中的"触发源分类名称"一致；

参数配置

•触发源名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"触发源配置"和"触发源和预案关联关系"中的"触发源名称"一致；

•参数名：若"动作配置"中的"响应值"配置的是类似于含有"#{Tag}"的字符串，则参数名则为"#{Tag}"；

•参数值：为根据实际情况所设定的值；

•参数描述：名称自定义，长度不得超出64位字符，32位中文；

•预案名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"动作配置"、"预案配置"和"触发源和预案关联关系"中的"预案名称"一致；

•动作名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"动作配置"中的"动作名称"一致；
•预案分类名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"动作配置"、"预案分类"、"预案配置"和"触发源和预案关联关系"中的"预案分类名称"一致；
•触发源分类名称：名称自定义，长度不得超出64位字符，32位中文；只能包含数字、中文、字母和下划线，不能以数字和下划线开头；与"触发源分类"、"触发源配置"和"触发源和预案关联关系"中的"触发源分类名称"一致。

•注：若"动作配置"中的"响应值"配置的是实际的数据源，则该sheet页无需配置。

7.11联动运行

•联动的运行分为两部分：

◦**运行客户端：**指在组态画面响应来自边缘节点的组态调用指令，例如打开画面、打开URL。

.启动运行客户端

.联动运行客户端默认是不开启的。需要在画面中调用 `sys.startResponseClient()`来启动。

.同一个画面中只会运行一个联动运行客户端。

.停止运行客户端

.调用`sys.stopResponseClient()`来停止。该接口没有返回值。

◦**监视客户端：**指在组态画面监视边缘节点的预案执行状态，特别是对于某些配置为需要人工介入的动作。（暂未实现）

7.12 变量表达式一览

1.函数列表

函数	语法	说明
INT	INT(a)	对变量a取整
EXP	EXP(a)	自然常数e(有时也叫欧拉数，Euler's number）的a次幂
ABS	ABS(a)	a的绝对值
SIN	SIN(a)	正弦函数
COS	COS(a)	余弦函数
TAN	TAN(a)	正切函数
ASIN	ASIN(a)	反正弦函数
ACOS	ACOS(a)	反余弦函数
ATAN	ATAN(a)	反正切函数

SQRT	SQRT(a)	a的平方根
LOG	LOG(a)	a的自然对数
LOG10	LOG10(a)	LOG(a)/LOG(10)

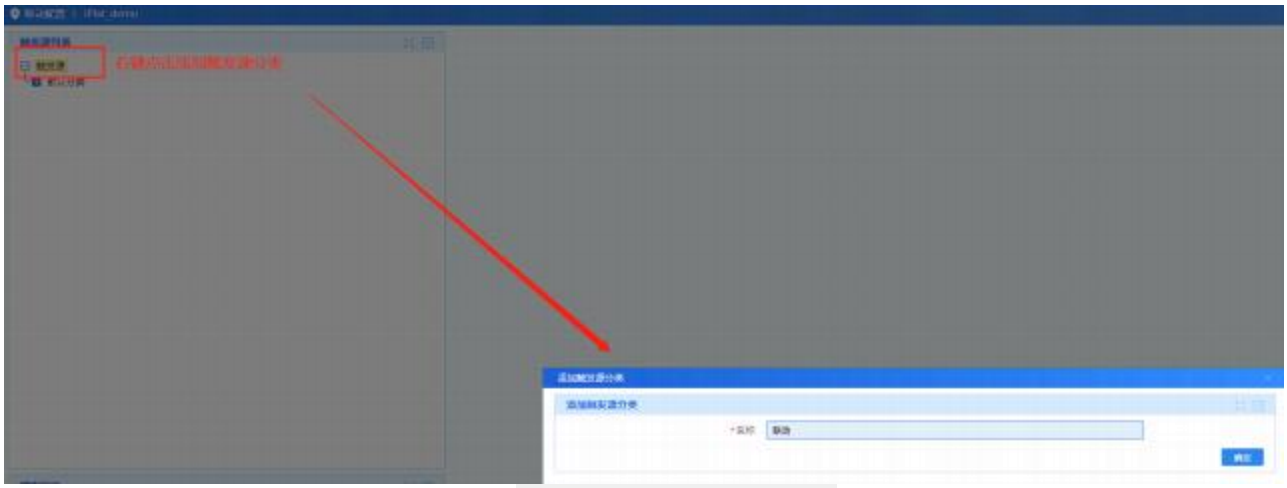
2. **运算符**

运算符	语法	说明
+	a+b	加法
-	a-b	减法
*	a*b	乘法
/	a/b	除法
AND	a AND b	逻辑与
OR	a OR b	逻辑或
XOR	a XOR b	逻辑异或
NOT	NOT a	逻辑非
==	a == b	是否相等
<>	a <> b	是否不等
<	a<b	是否小于
>	a>b	是否大于
<=	a <= b	是否小于或等于
>=	a >= b	是否大于等于

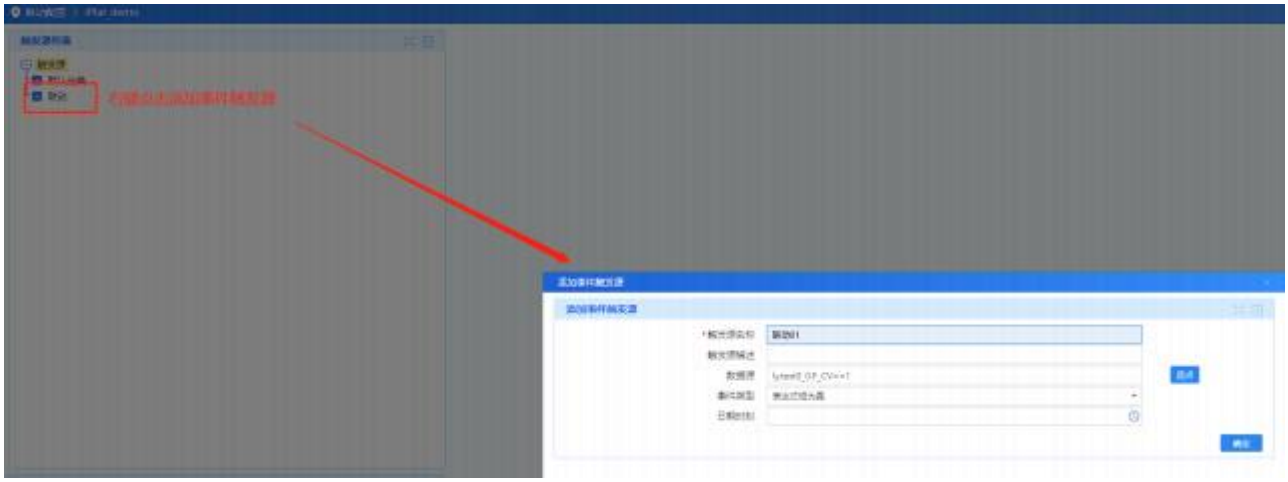
示例介绍--实现脉冲

1. 打开联动配置页面。

2.添加分类"联动"。



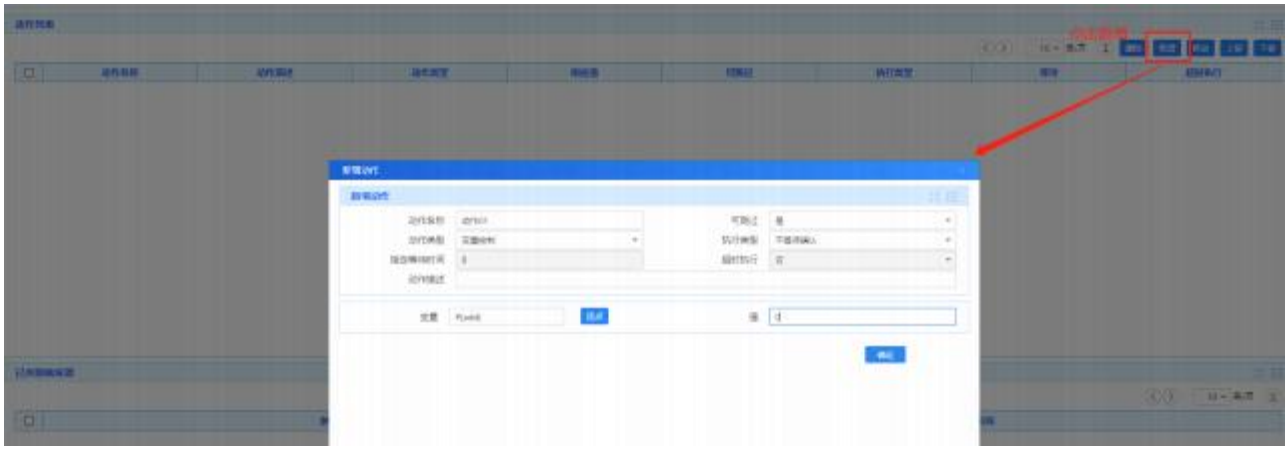
3.新增事件触发源"联动01"，数据源填"ly.test0_0.F_CV==1"，点名可以通过选点方式添加；事件类型选择"表达式恒为真"，确认数据源没问题后,点击"确定";



4. 添加"预案"分类.

5. 添加预案"预案01"； 添加动作， 点击修改动作， 最后保存预案；

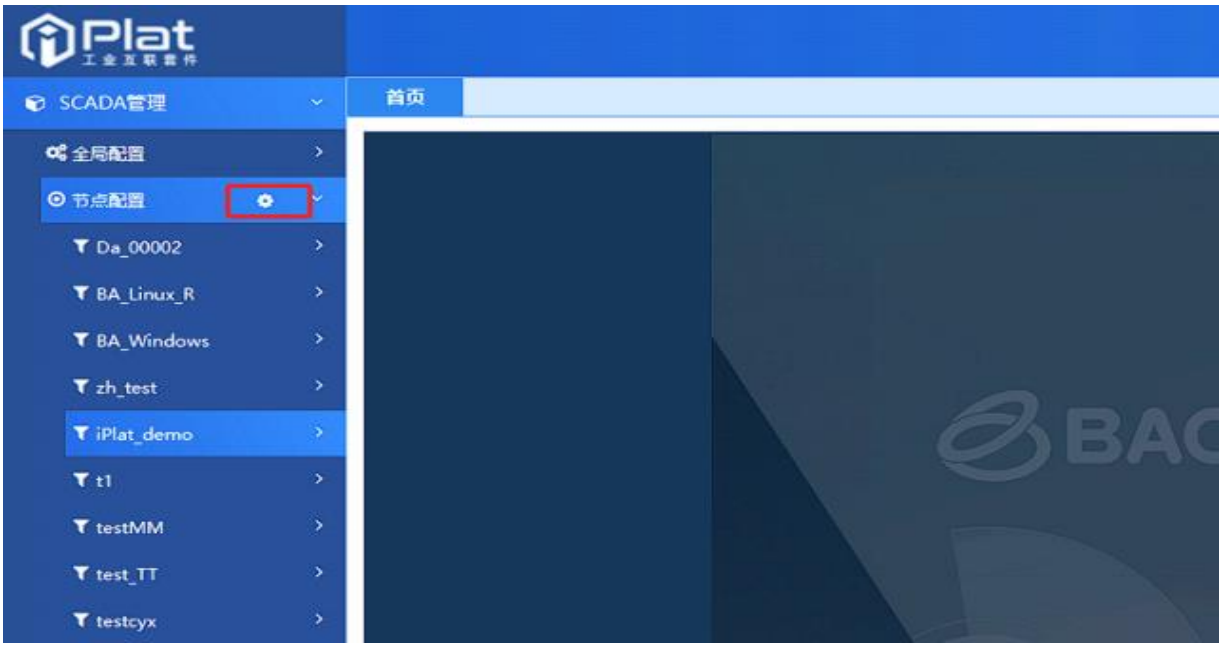
6. 点击"新增"按钮添加动作， 最后点击确定；



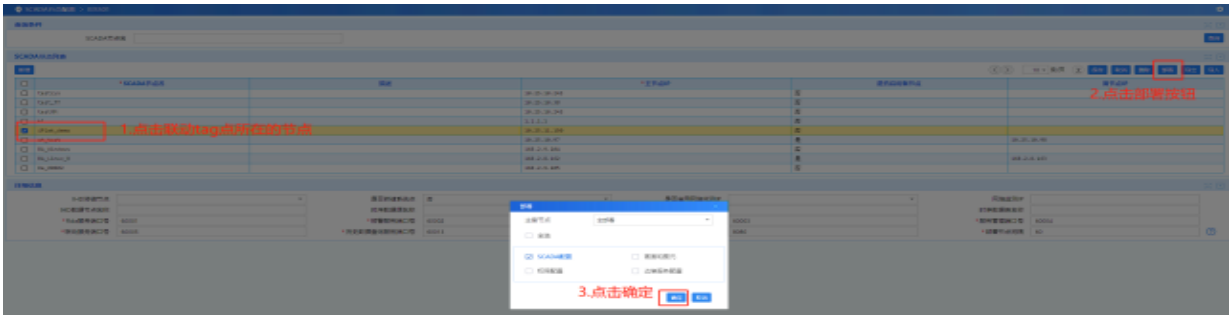
7. 返回到"联动01"触发源;点击"关联预案",关联"预案01";

8. 点击"设置参数",设置#{add}参数值为"ly.test0_0.F_CV";

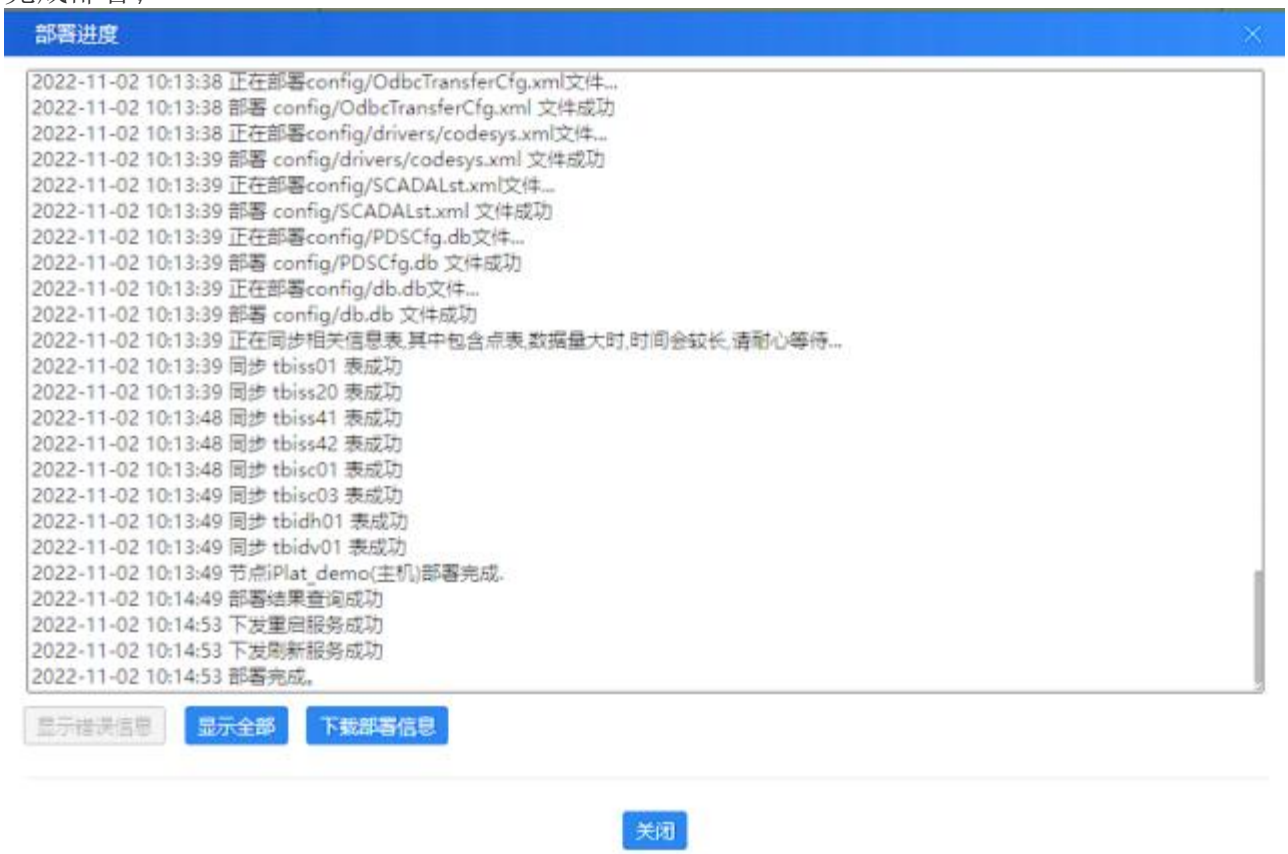
9. 回到iplat首页,点击节点列表配置后的齿轮图标,进入边缘节点配置界面;



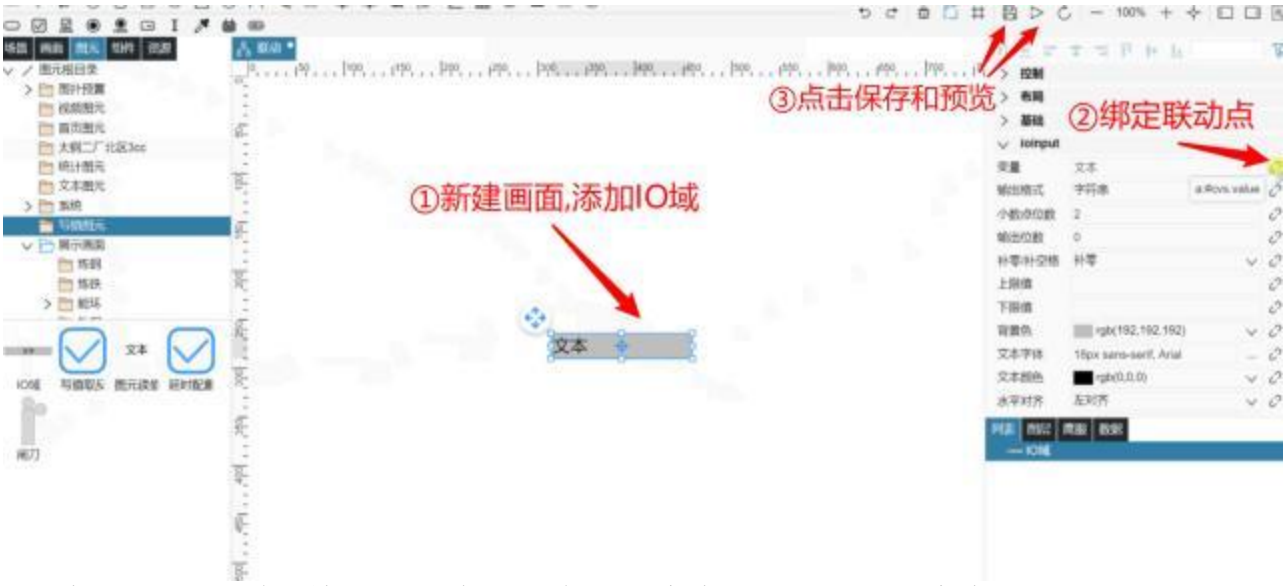
10. 勾选节点，点击部署；



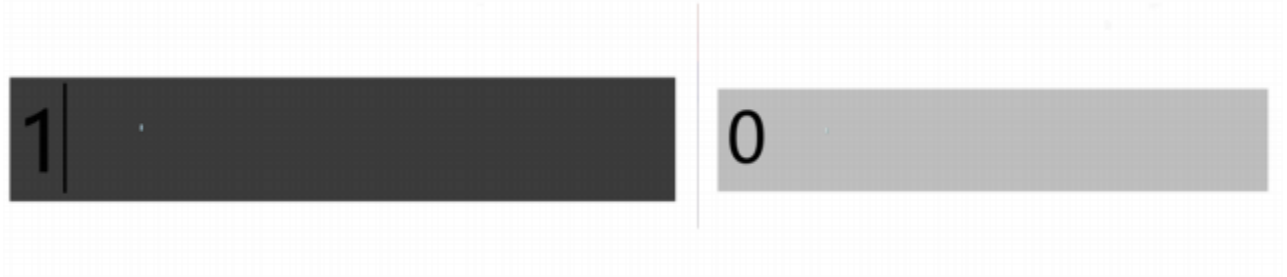
11. 等部署进度弹窗出现"部署完成"提示，切关闭按钮恢复可点击状态(蓝色)时，点击关闭，完成部署；



12. 新建联动测试画面，拖入IO域图元，绑定联动点"ly.test0_0"；保存并预览画面；



13.运行效果：点击IO域，输入1并回车，IO域值从0变为1；1s后值自动变为0；



7. 权限与安全

7.13 用户权限配置

iPlat系统中所有用户页面，按钮等等资源权限均由admin最高权限账户授予，以下为配置流程

1.搜索框授权页面BIPG01

2.相应的用户下选择新增资源类型

首先在左侧"用户组管理"根目录下选择相对应的用户组，然后在右边选择业务所需授权项；

对组态页面，可以在授权页面中选择可见可控或可见不可控权限

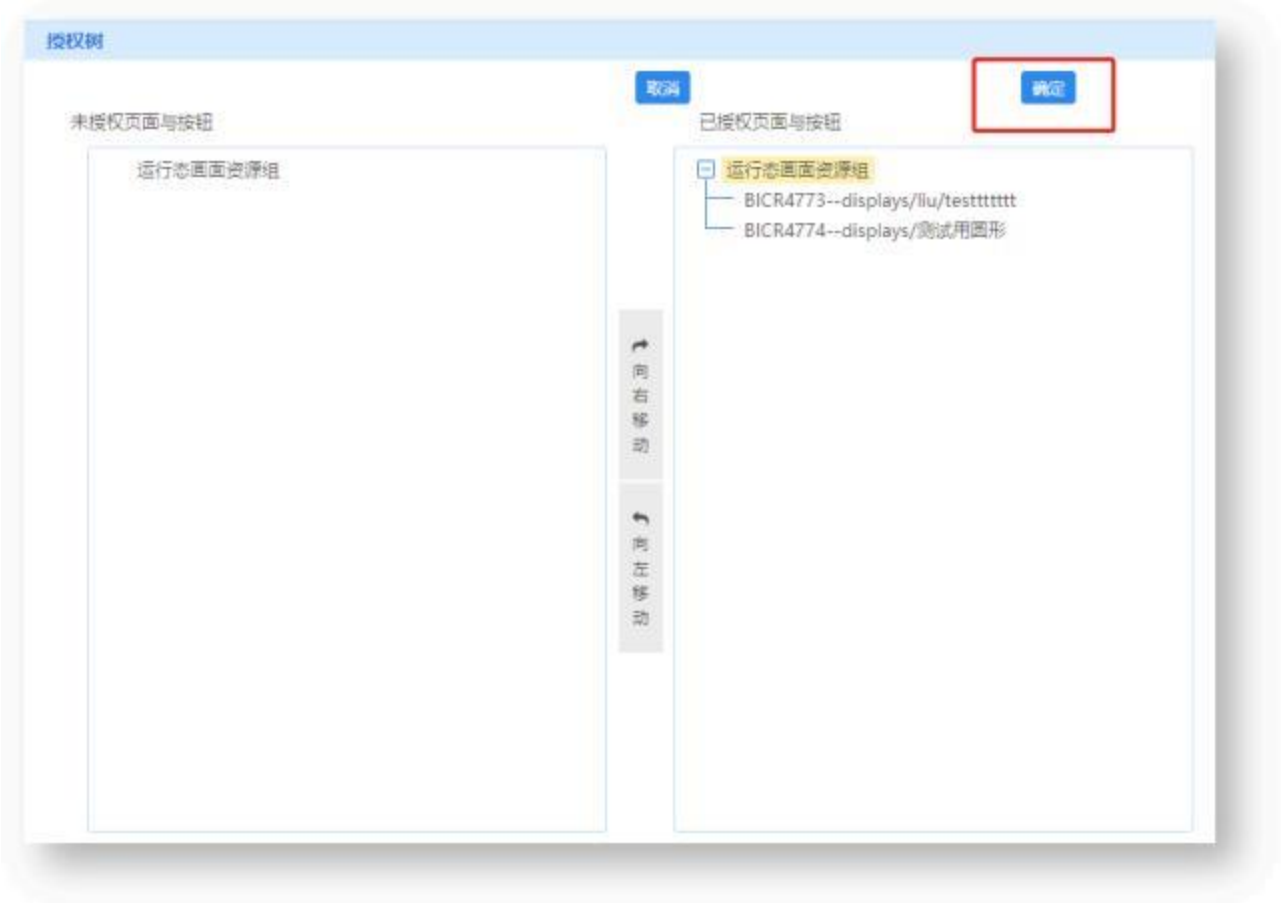
可见可控：群组用户对该页面可以查看可以进行控制操作

可见不可控：群组用户对该页面只有查看功能

3. 授权资源

如上图，左侧为"未授权页面与按钮"，右边为"已授权页面与按钮"，选择好相应资源后，点击中间"向右/向左移动"；

4. 确定授权



点击"确定"按钮，资源授权完成。

5.登录相应授权账号测试相应资源

7.14 管理端页面授权

授权管理页面可从四个视角进行授权，操作人员可根据不同需求选择最佳视角进行授权操作。具体参考：授权信息管理配置。

授权信息管理

概述

授权信息管理页面用于统一管理平台的用户或用户组的授权信息。

路径

【系统平台】-【安全管理】-【资源权限管理】-【授权信息管理】

操作步骤

授权管理页面可从四个视角进行授权，操作人员可根据不同需求选择最佳视角进行授权操作。

菜单视角授权

概述

菜单视角授权用于实现多维度授权，从菜单资源维度进行授权。

路径

【系统平台】-【安全管理】-【资源权限管理】-【授权信息管理】—菜单视角授权tab页

操作步骤

界面视图如下：

•授权信息查询

i. 在查询条件区域输入查询条件，点击【查询】按钮。

ii. 记录集区域显示符合查询条件的授权信息记录。

注意

授权主体名称与授权客体名称为模糊查询。

点击左侧的系统菜单树下的按钮资源、页面资源，则会自动查询出被点击节点的授权信息，并显示在数据集中。

•授权信息新增

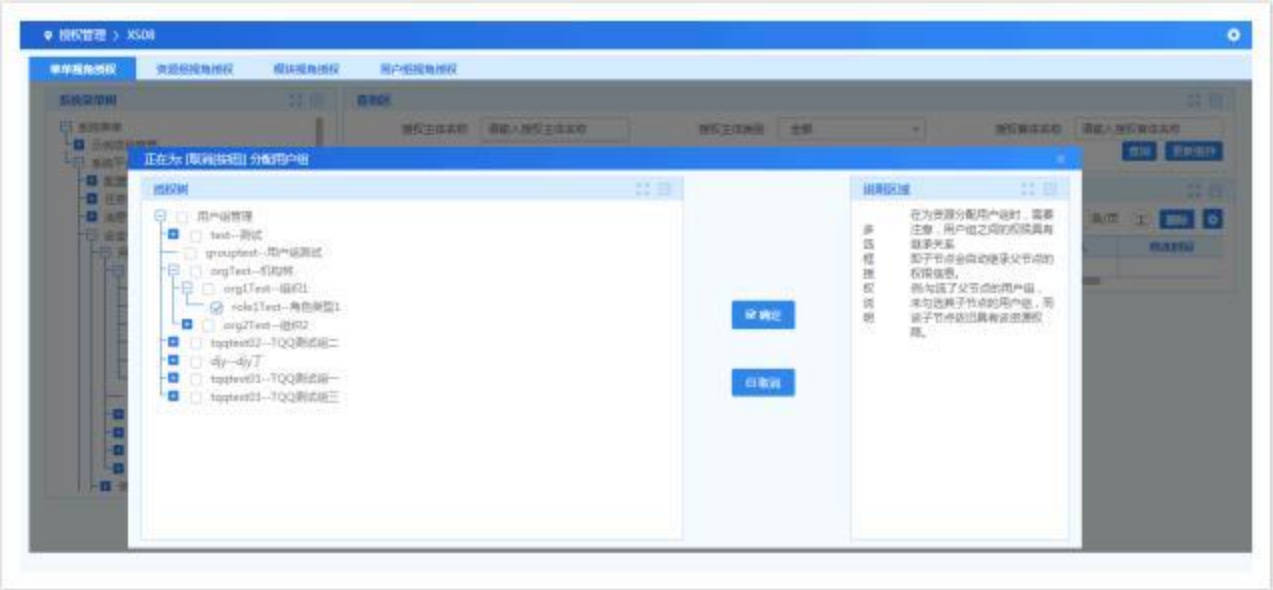
i. 点击左侧系统菜单树下的节点。

ii. 点击右侧记录集区域左上角的【授权】按钮。

iii. 在弹出的页面中，勾选需要授权的用户组前面的复选框，点击【确定】按钮，授权完成。

注意

与正在授权的资源已经是授权关系的用户组，前面的复选框会显示勾选状态，点击勾选框取消勾选，再点击【确定】按钮，可以解除用户组与资源的授权关系。



注意

只能为页面和按钮进行授权。

在为资源分配用户组时，需要注意，用户组之间的权限具有继承关系，即子组会自动继承父组的权限。

•授权信息修改

i.勾选记录集中需要修改的授权信息记录。

ii.编辑信息，点击【保存】按钮。

注意

授权主体名称、类别和授权客体类别名称、类别不可修改。

•授权信息删除

i.勾选记录集中需要删除的授权信息。

ii.点击【删除】按钮，弹出提示信息，选择是否删除此数据。

资源组视角授权

概述

以资源组的维度进行授权，适用于固定不变的权限资源组合。

路径

【系统平台】-【安全管理】-【资源权限管理】-【授权信息管理】—资源组视角授权tab页

操作步骤

界面视图如下：

•授权信息查询

i.在查询条件区域输入查询条件，点击【查询】按钮。

ii.记录集区域显示符合查询条件的授权信息记录。

注意

授权主体名称与授权客体名称为模糊查询。

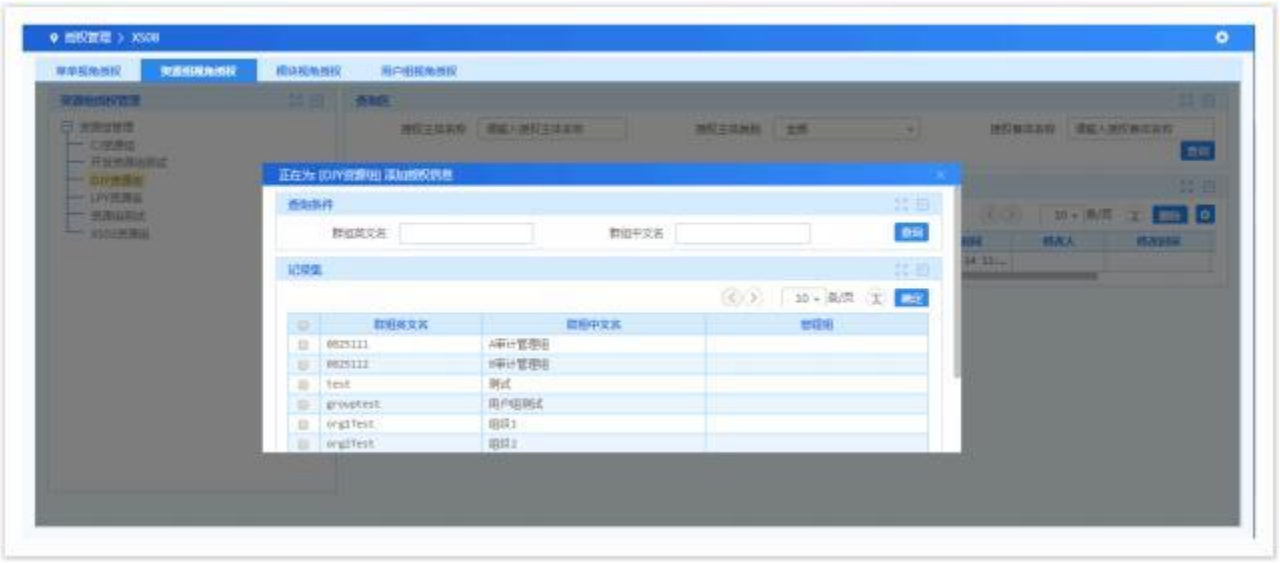
点击左侧的资源组信息树，则会自动查询出被点击的节点下的授权信息，并显示在右侧数据集中。

•授权信息新增

i.点击左侧的资源组树的节点。

ii.点击记录集区域左上角的【授权】按钮。

iii.在弹出的页面中，勾选需要授权的用户组，点击【确定】按钮，授权完成。



•授权信息修改

i.勾选记录集区域中需要修改的授权信息记录。

ii.编辑信息，点击【保存】按钮。

注意

授权主体名称、类别和授权客体名称、类别不可修改。

•授权信息删除

i.勾选记录集区域中需要删除的授权信息记录。

ii.点击【删除】按钮，弹出提示信息，选择是否要删除此数据。

模块视角授权

概述

以模块的维度进行授权。

路径

【系统平台】-【安全管理】-【资源权限管理】-【授权信息管理】—模块视角授权tab页

操作步骤

界面视图如下：

•授权信息查询

i. 在查询条件区域输入查询条件，点击【查询】按钮。

ii. 记录集区域显示符合查询条件的授权信息记录。注意

授权主体名称与授权客体名称为模糊查询。

点击左侧的资源组信息树，则会自动查询出被点击的节点下的授权信息，并显示在右侧数据集中。

•授权信息新增

i. 点击左侧的模块列表树的节点。

ii. 点击右侧记录集区域左上角的【授权】按钮。

iii. 在弹出的页面中，勾选需要授权的用户组，点击【确定】按钮，授权完成。

注意

只能为页面和按钮进行授权。

•授权信息修改

i. 勾选记录集区域中需要修改的授权信息记录。

ii. 编辑信息，点击【保存】按钮。

注意

授权主体名称、类别和授权客体名称、类别不可修改。

•授权信息删除

i. 勾选记录集区域中需要删除的授权信息记录。

ii. 点击【删除】按钮，弹出提示信息，选择是否要删除此数据。

注意
模块信息可在信息管理页面(EDFA00)查看。

用户组视角授权

概述

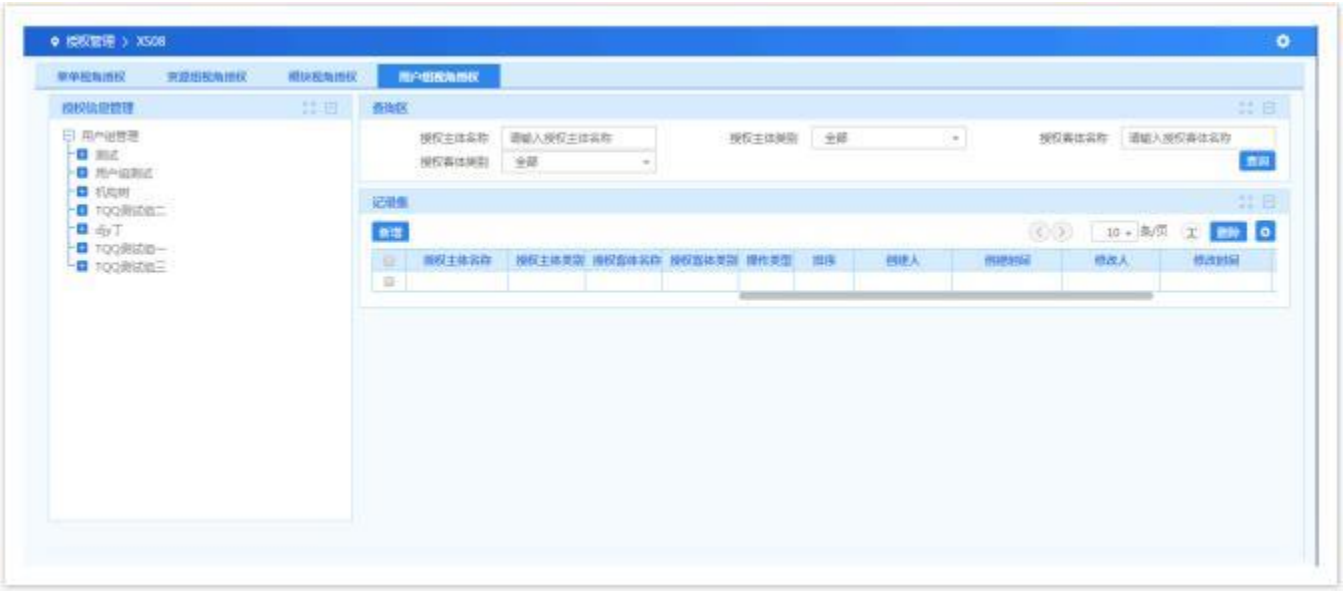
用户组视角授权页面用于统一管理平台的用户组的授权信息。

路径

【系统平台】-【安全管理】-【资源权限管理】-【授权信息管理】—用户组视角授权tab页

操作步骤

界面视图如下：



•授权信息查询

i.在查询条件区域输入查询条件，点击【查询】按钮。

ii.记录集区域显示符合查询条件的授权信息记录。

注意

授权主体名称与授权客体名称为模糊查询。

点击左侧的群组成员信息树，则会自动查询出被点击的节点下的授权信息，并显示在右侧数据集中。

•授权信息新增

i.点击左侧的用户组树的节点。

ii.点击记录集区域左上角的【新增】按钮。

iii.在弹出的页面中，在左边的未授权的页面与按钮资源框中选中资源的节点后，点击【向右移动】按钮或者双击资源将资源加到右边的已授权页面与按钮框中，再点击【确定】按钮，授权完成。

iv.在已授权页面与按钮框中，选中需要解除授权关系的资源节点，点击【向左移动】按钮或者双击将资源移动到未授权页面与按钮框中，再点击【确定】按钮，即可解除用户组
与此资源的授权关系。

注意

【用户组管理】节点无需授权，叶子结点和非叶子节点都可进行授权操作。

•授权信息修改

i.勾选记录集区域中需要修改的授权信息记录。

ii.编辑信息，点击【保存】按钮。

注意

授权主体名称、类别和授权客体名称、类别不可修改。

•授权信息删除

i.勾选记录集区域中需要删除的授权信息记录。

ii.点击【删除】按钮，弹出提示信息，选择是否要删除此数据。

超级用户组授权

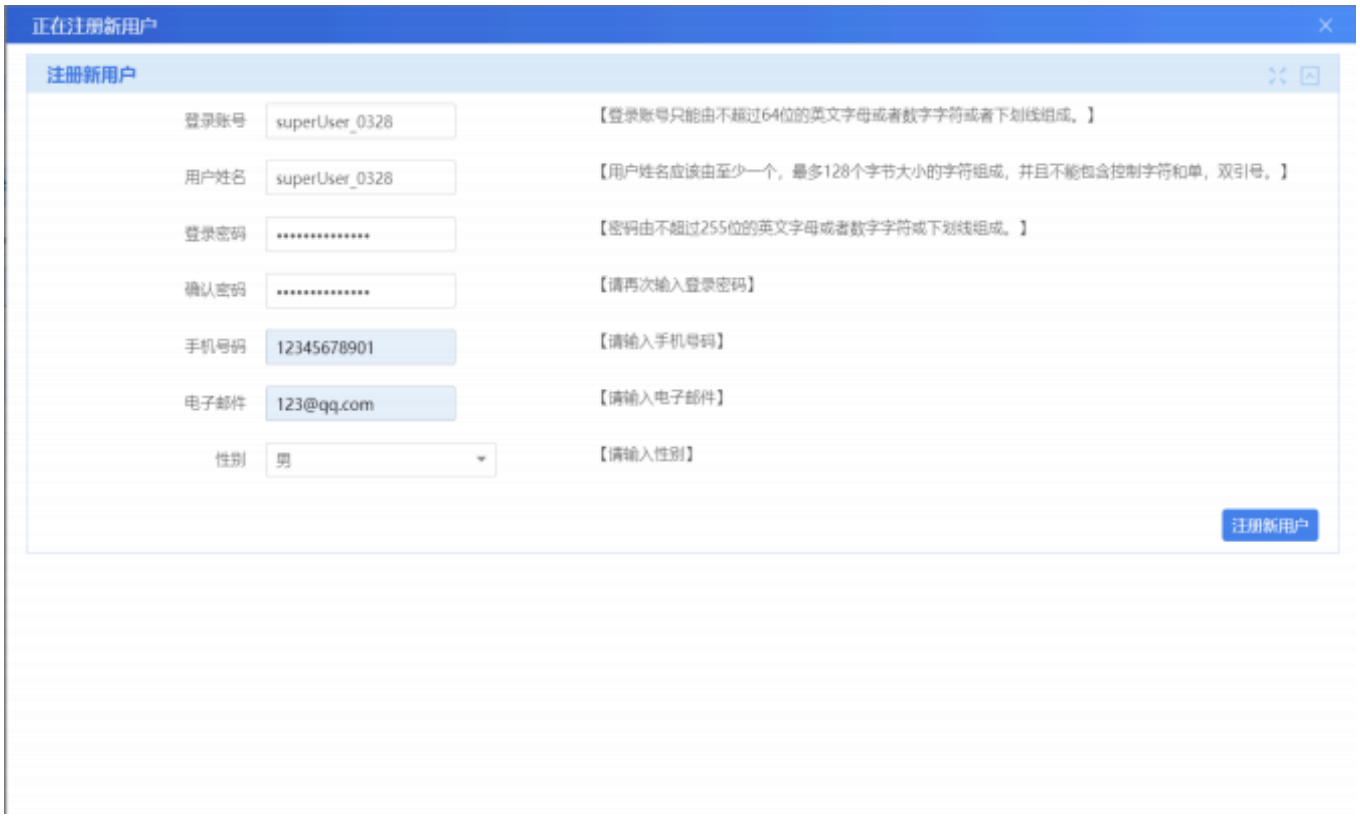
概述

超级用户组：超级用户组是指【ADMIN】用户组，该组下的成员，具有所有的权限。

创建超级用户组成员

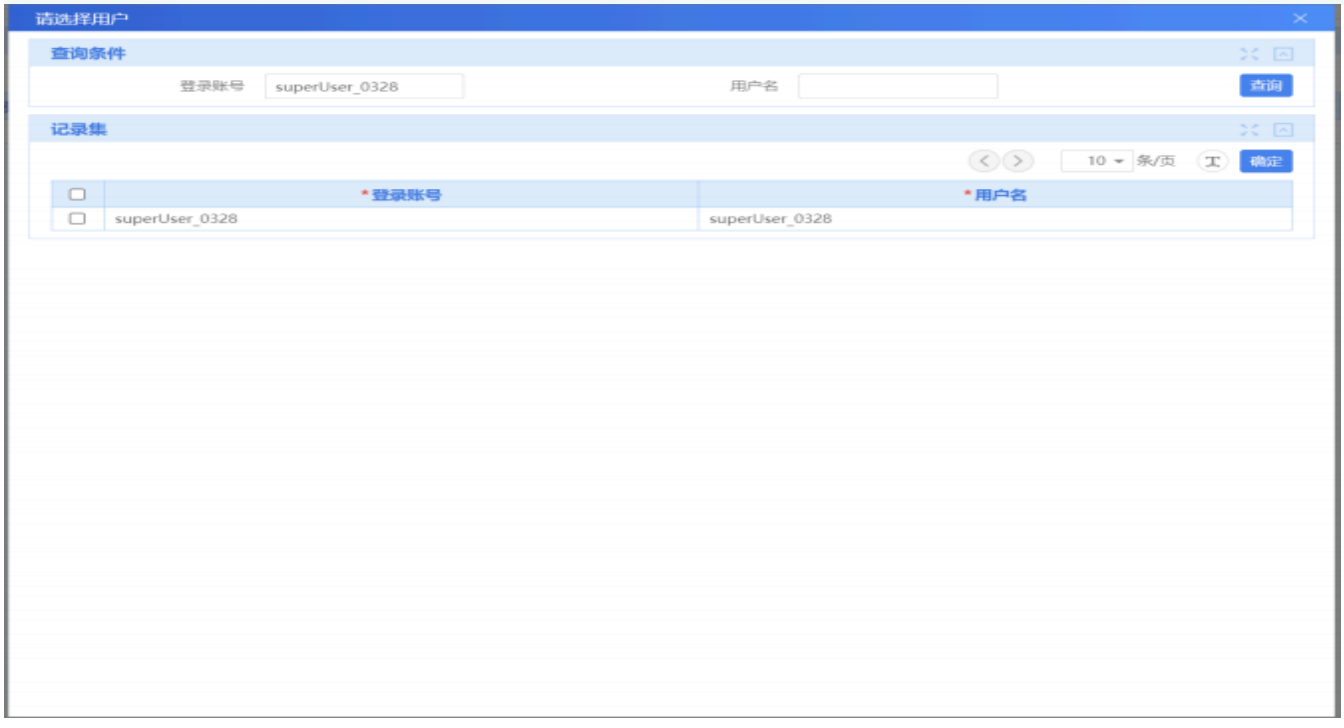
1.注册用户

前往【XS01】页面新建用户

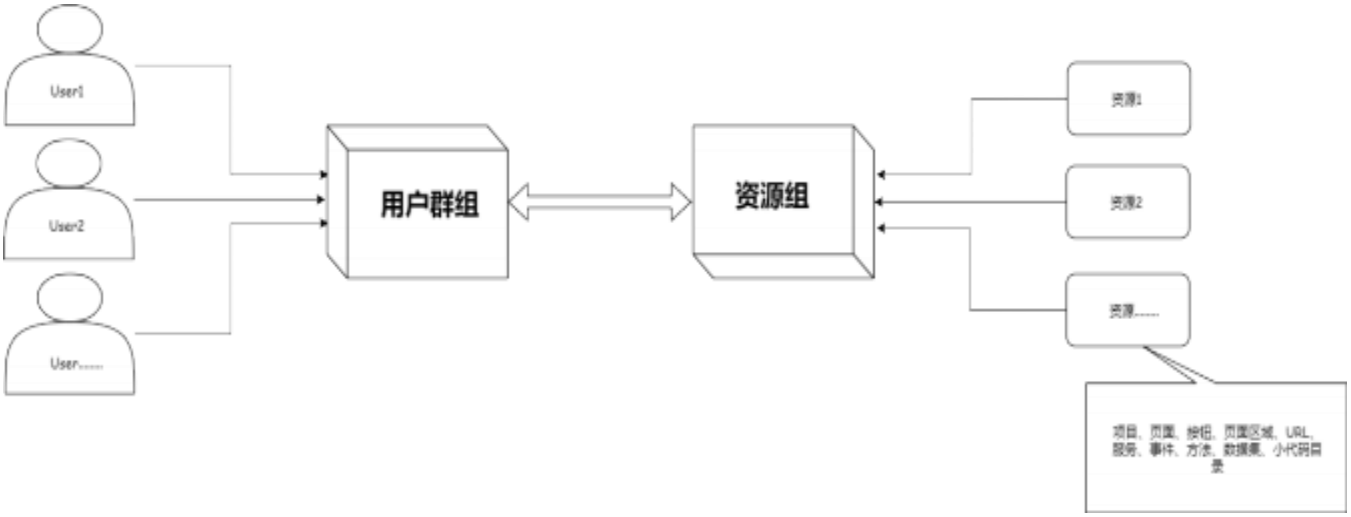


2. 进行授权

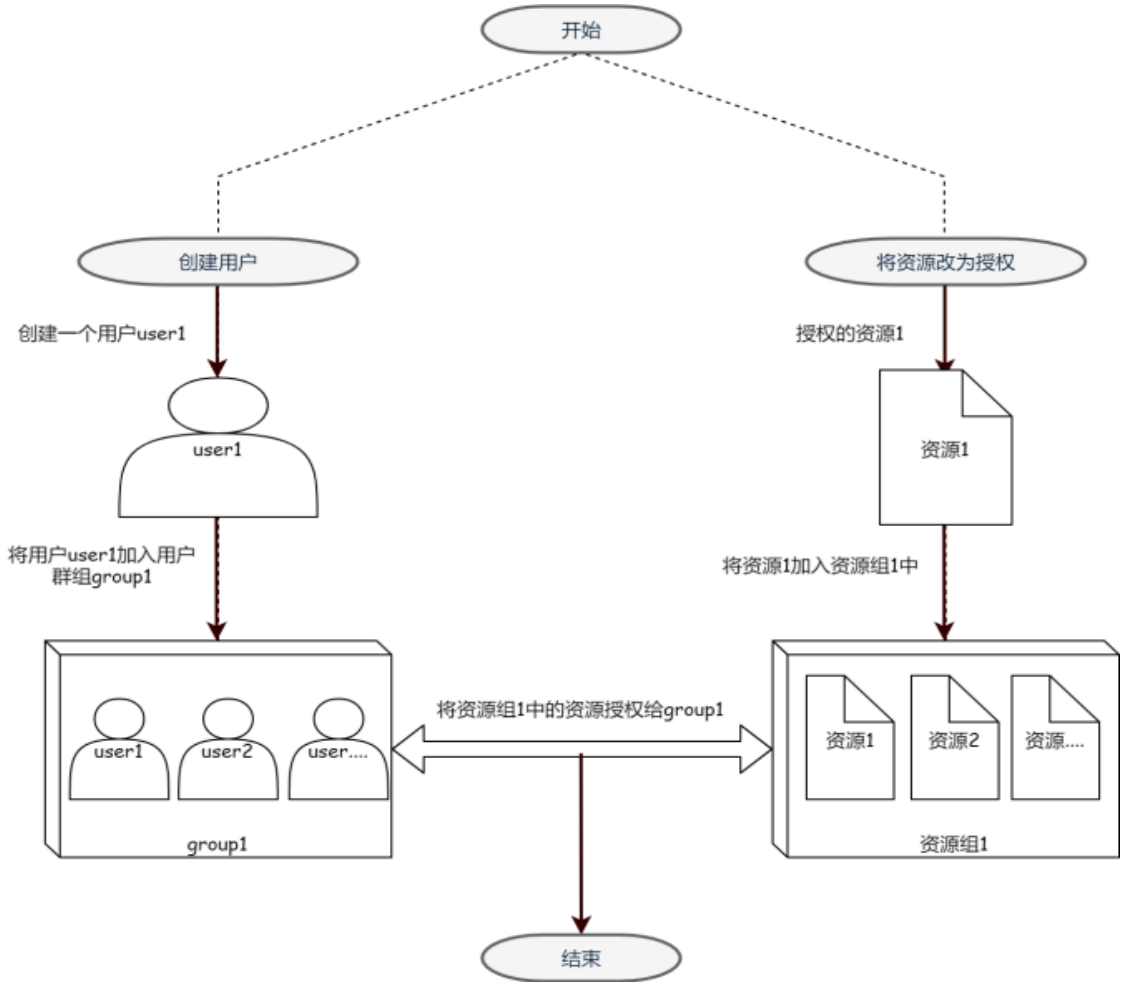
前往【XS17】页面进行授权操作



iplat对画面中的元素提供了有效的权限控制方式，只需要通过简单的配置，用户就可以实现同一画面在不同用户下操作不同。元素的权限只在画面运行态有效。
下面通过具体的操作实例实现对元素权限的配置。



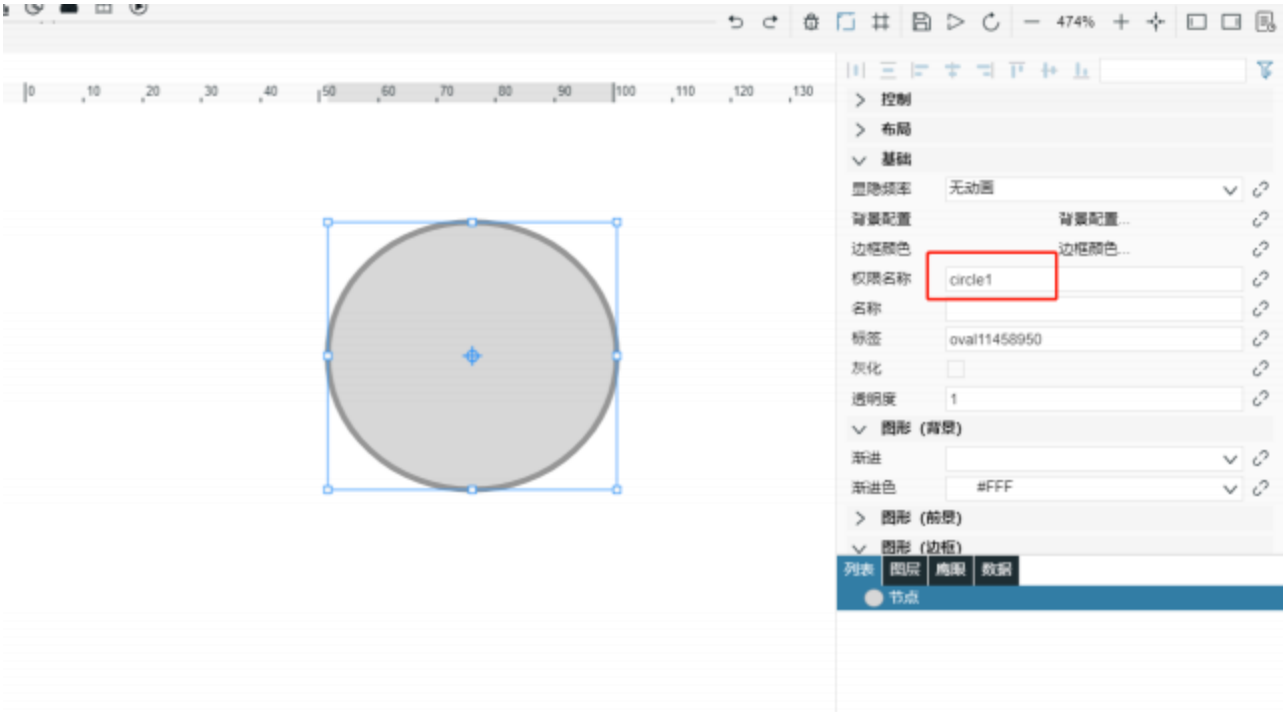
授权流程



7.15 配置过程

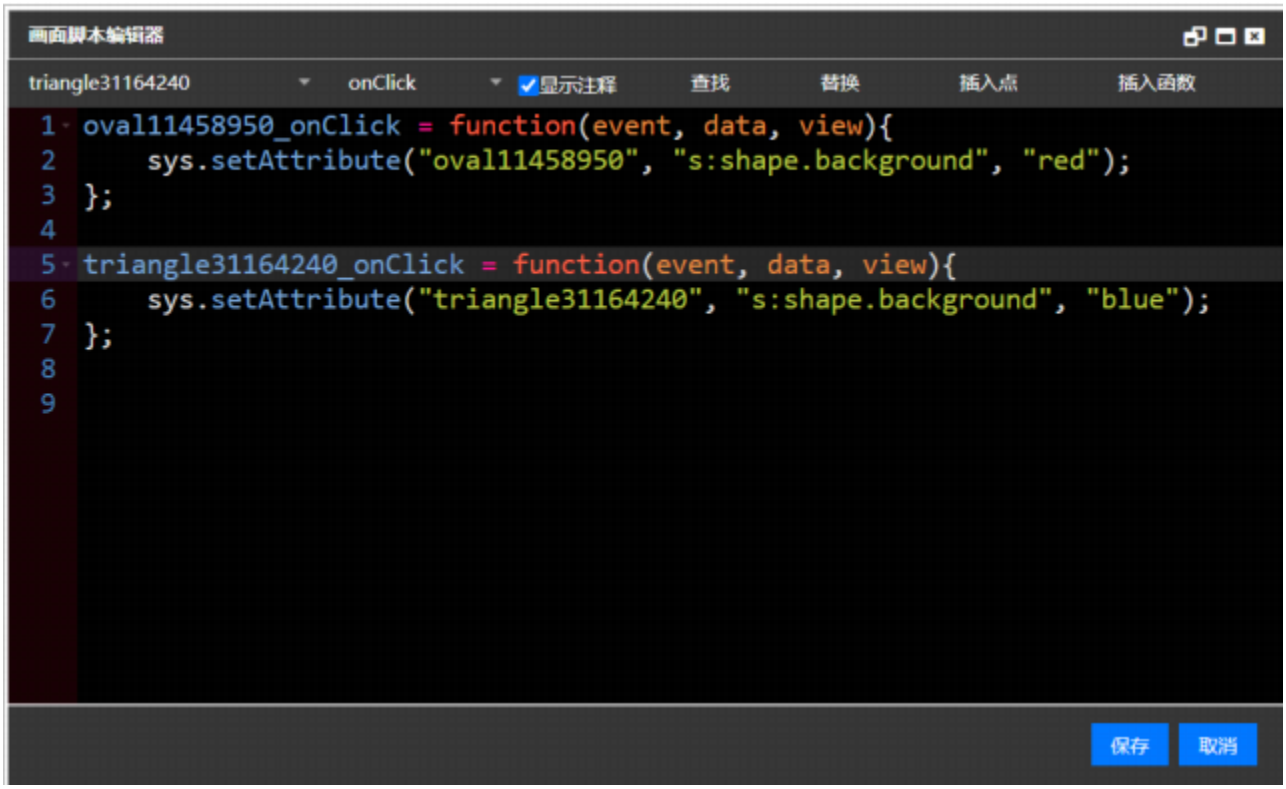
给元素配置权限名称

1.打开2D设计器，创建画面名称为"test_acess",在画面中拖入一个圆，选中之后，给改元素配置权限名称，在设计器的右侧列表中，给属性"权限名称"输入值"circle1",如下图：



2.按照上面步骤，在拖入一个元素三角形，同样配置权限名称为triangle1.

3.编写脚本代码，点击圆变成红色，点击三角形变成蓝色



4.配置好后保存画面。

给元素授权

1.首先查看该元素所在的画面的英文名称，在2D设计器的中打开该画面，查看右下角的数据tab页，找到属性"CRFileEname"，本示例的名称如下：

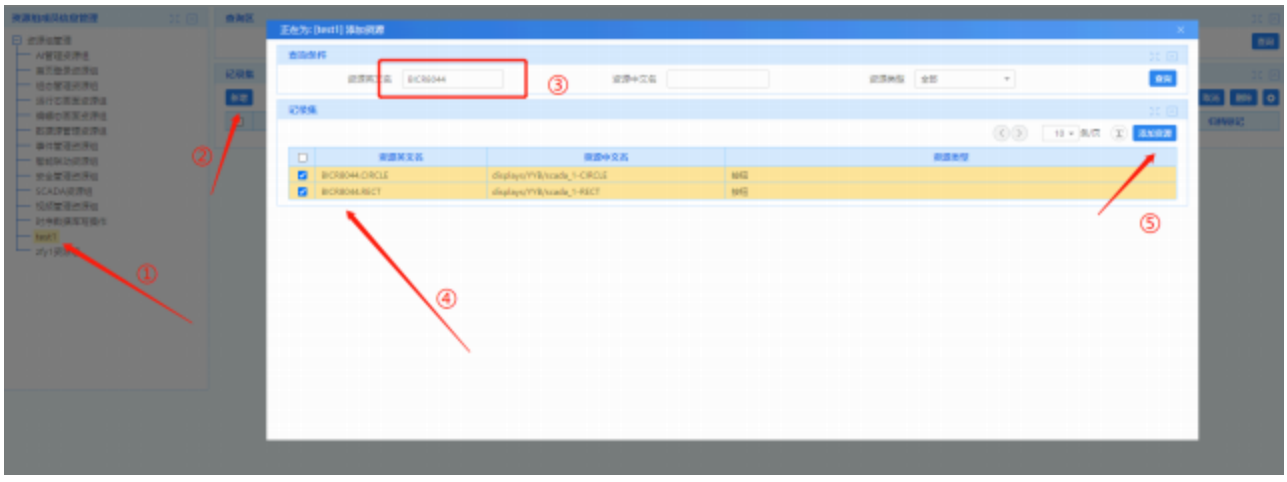
2.在系统中打开EDFA00页面，输入画面名称，点击查询，如下图：

3.点击BICR8044所在行的操作那列上的"按钮信息"，弹出"按钮信息配置"对话框：

在图中，我们可以看到我们在2D设计器中配置的2个权限rect和circle,都出现在列表中，修改他们的是否授权属性，修改为授权。注意，先要勾选，才能修改。

将授权元素(资源)加入资源组

1.在XS06-资源群组成员信息管理页面，点击左侧资源组树中的资源组

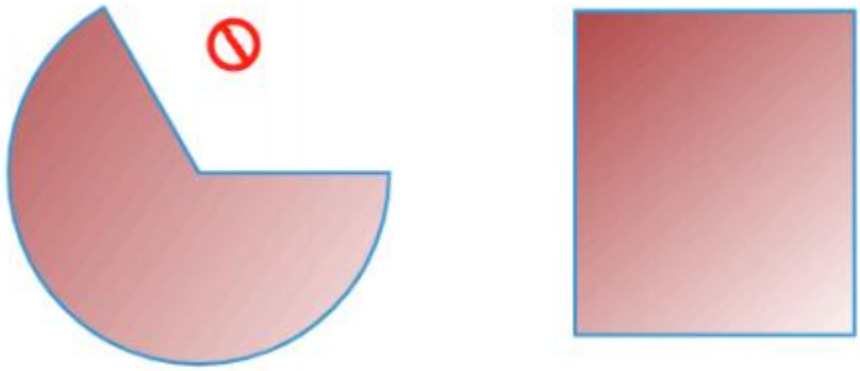


将资源组中资源授权给用户群组

1. 在BIPG02-群组授权管理页面，将资源组添加到用户

2. 运行态！

授权的用户群组用户可以正常操作元素，点击元素便是，未授权的用户群组打开画面不能操作这两个元素，点击元素没有效果。



设计器中画面可以独立对系统中的资源进行授权。目前支持的资源类型有：页面区域、URL、服务、方法、事件、数据集和项目。

推荐在画面中做授权时候使用URL类型

系统权限配置

案例中会建三个用户user,user1和user2。其中user拥有最高权限，用来做画面授权。user1拥有RESOURCE1,user2没有RESOURCE1权限。

在编辑态用user对画面进行授权。user1和user2登陆运行画面，拥有权限的可以查看运行态，没有权限的user2则显示404.

1. **配置资源**

在安全管理→资源权限管理→资源信息管理中可以配置需要的资源项。在画面中使用的资源建议使用**URL类型**。新建资源名称为：画面1，如下图：

2. **配置资源组**

在安全管理→资源权限管理→资源组信息管理中可以配置资源所属的资源组。新建资源组名称为：RESOURCE1. 如下图：

3. **资源组配置资源**

在安全管理→资源权限管理→资源组成员信息管理中可以配置资源所属的资源组。将画面1的资源给资源组1. 如下图：

4. **用户注册**

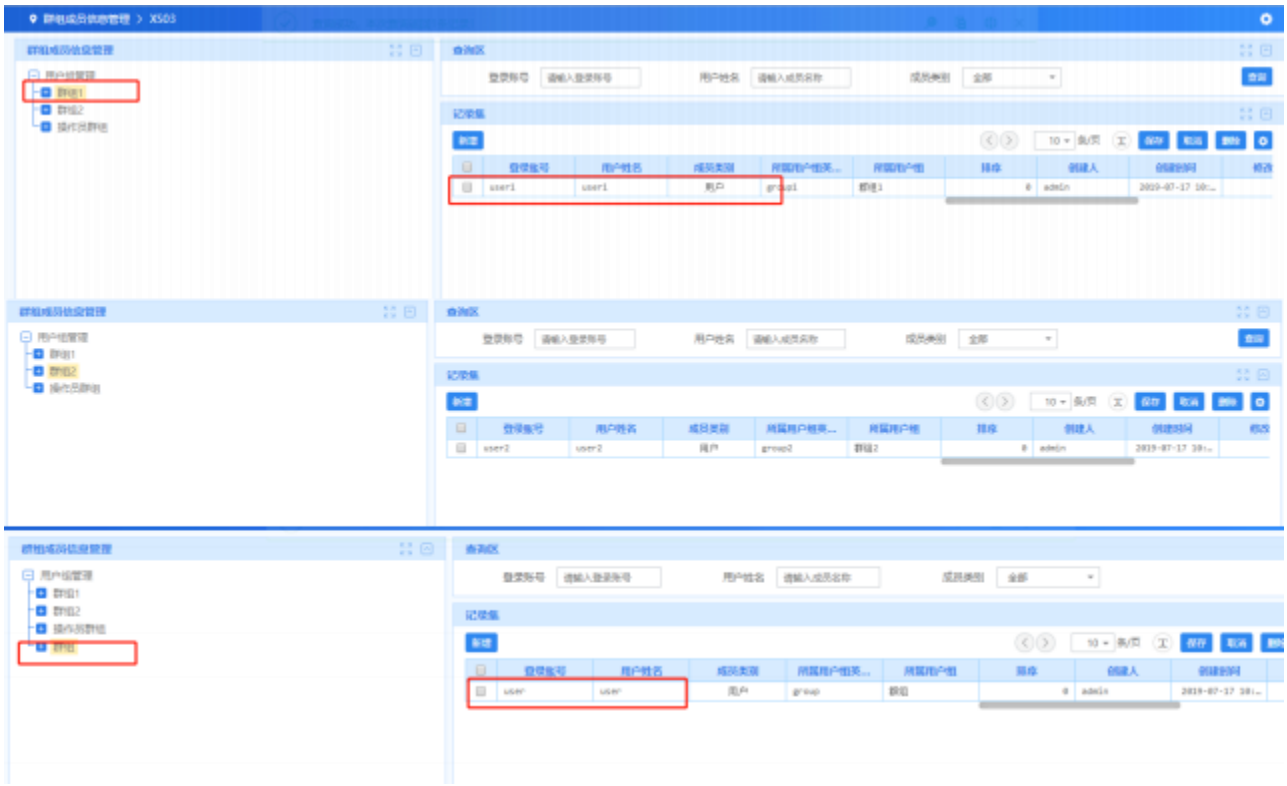
在安全管理→用户信息管理→用户信息管理中新增用户。注册用户user1和user2（两个用户用来对比拥有权限和没有权限），以及用户user拥有所有资源权限

5. **用户创建不同的用户组**

在安全管理→用户信息管理→用户群组信息管理中创建两个不同的用户组group1和group2，如以及user所在的群组group。如下图：

6. **配置用户属于不同的用户组**

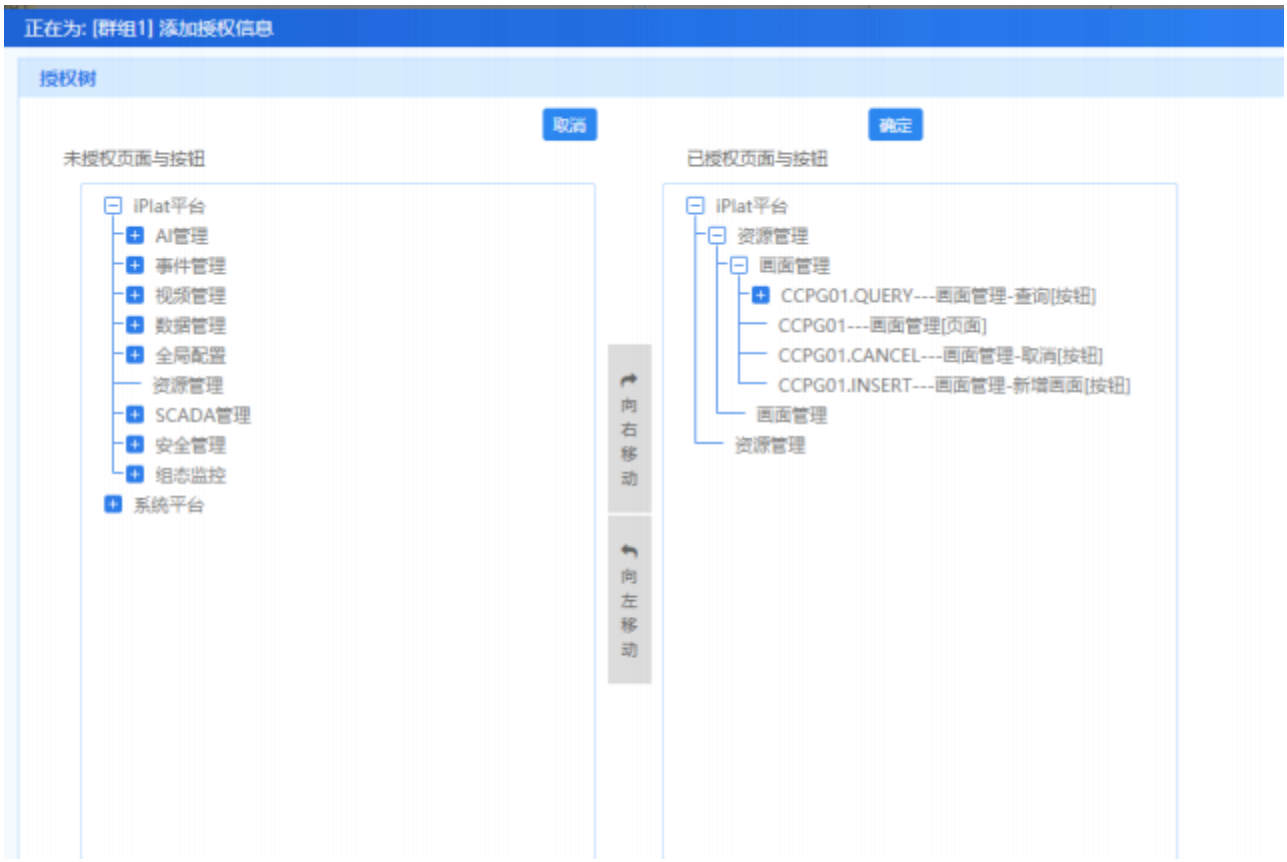
在安全管理→用户信息管理→群组成员管理中为用户配置不同的用户组，如下图：



7. **为用户组配置资源组权限和页面权限**

a. 资源组视角授权，group和group1拥有资源1的权限

b. 用户组视角授权，三个group都需拥有画面配置的菜单权限



画面授权配置

登录用户（user）,再画面做授权

用user登陆后，选择要授权的页面，点击授权属性的选择按钮，在，弹出的框中可以看到配置的资源[画面1]

配置成功后保存画面。

注意：在选择授权项中可以看到除了"页面"和"按钮"外的所有授权项，建议在新建资源时候选择类型为URL.