

# CVS 用户手册(V3.0.0)

---

**CVS** (Configuration View Service-组态展示服务) 是将传统的监控系统与最新 Web 技术相结合的平台产品。CVS 可以让用户在浏览器环境中搭建面向Web和移动设备的工业监控系统,通过“搭积木”、“零编程”的方式在多种终端设备上快速、简单、安全的访问和查看用户数据,降低运营成本,提高工作效率,并最终为您的企业提供基于数据的决策依据。

该文档描述如何使用 CVS 系统,您可以先从 **开始使用** 章节开始,该章节描述了一个简单的完整的使用流程,可以使您很快的搭建监控系统并实际使用。接下来,您可以根据实际遇到的问题来查阅各个章节,每个章节都有不同的主题。

---

## [1. 开始使用](#)

- [1.1 安装部署](#)
- [1.2 数据代理](#)
- [1.3 运行Demo](#)
- [1.4 系统许可证](#)
- [1.5 CVS 配置说明](#)
- [1.6 CVS 系统升级](#)

## [2. 数据代理配置](#)

## [3. 系统管理](#)

- [3.1 数据源管理](#)
- [3.2 画面管理](#)
- [3.3 图元管理](#)
- [3.4 变量管理](#)
- [3.5 权限管理](#)
- [3.6 帮助](#)

## [4. 画面设计](#)

- [4.1 画面设计器介绍](#)
- [4.2 功能菜单](#)
- [4.3 工具栏](#)
- [4.4 元素](#)
- [4.5 属性配置](#)
- [4.6 脚本](#)

## [5. 图元设计](#)

- [5.1 图元编辑](#)
- [5.2 图元属性](#)
- [5.3 控件配置](#)
- [5.4 图元脚本编辑](#)
- [5.5 系统图元](#)

## [6. 画面运行](#)

- [6.1 趋势图运行](#)
- [6.2 报警控件运行](#)

- [6.3 历史报警运行](#)
- [6.4 历史回放](#)
- [6.5 前端消息查看](#)
- [6.6 html文件上传运行](#)

## 7. 常见问题

---

# 1. 开始使用

CVS 系统有两种安装方式，一种是通过xInsight大数据平台进行集成的方式，另一种是独立安装运行。云服务方式只需要申请租户，访问网址即可以使用组态服务。独立安装方式请参照 [1.1 安装部署](#) 章节。

## 1.1 安装部署

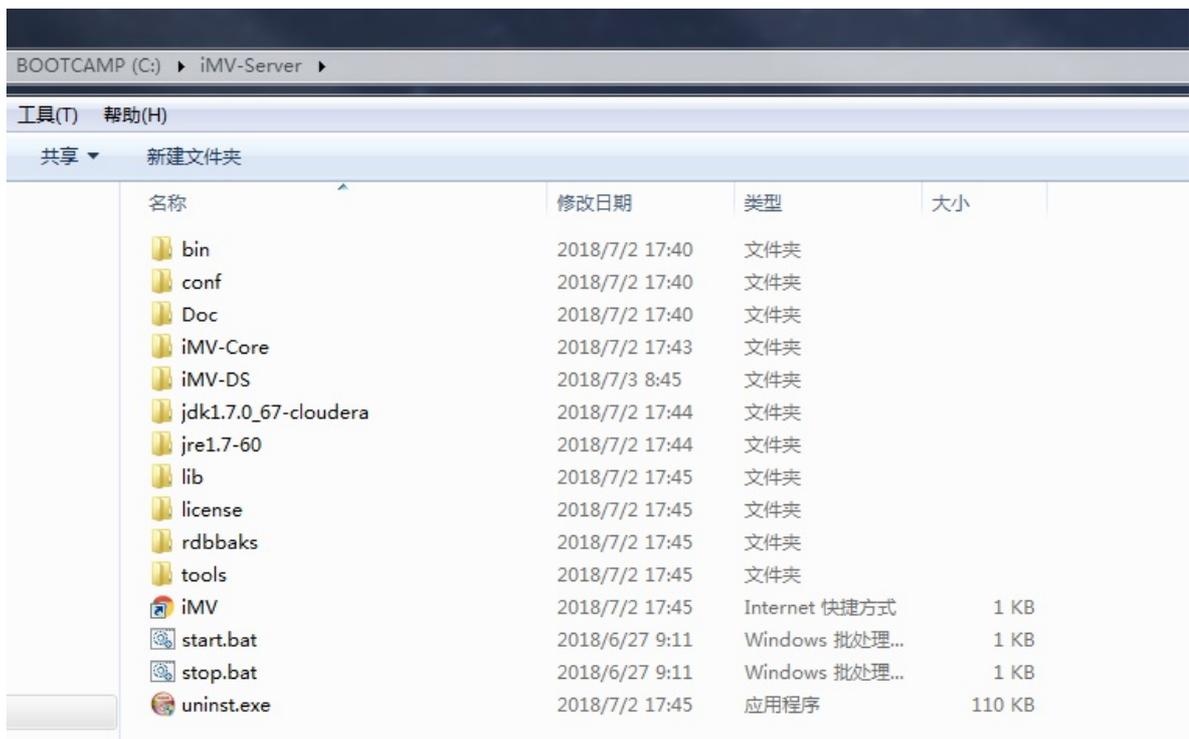
**注意：**请勿安装在中文路径或包含有空格的路径下。

软硬件要求：

项目	服务端	客户端
操作系统	Windows Server 2008 R2 及后续版本 RHEL/CentOS 6.6及后续版本 Solaris 11.2以及后续版本	Windows 7 及后续版本 带有图形化界面的 Linux/Solaris 系统
处理器	8核 Intel 或 SPARK 处理器，2.0Ghz 以上 例如 Intel Xeon E5-2600 V4	Intel Core i3 或同等级及以上的处理 器
内存	32GB 或更多	8GB 或更多
硬盘	500GB 或更多	500GB 或更多
网卡	1000 Mbps	100/1000 Mbps
其他	默认对外端口 18080，请检查端口是否被其他应用占用或者被 防火墙阻拦 数据库 PostgreSQL 9.1 以上版本	客户端浏览器支持 Chrome 60 及以 上版本

Windows 安装步骤：

1. 双击CVSSetup.exe程序，点击下一步，选择安装目录。（注意：安装路径不能包含中文或者空格）再点击下一步。
2. 如果已有PostgreSQL数据库，则点击下一步跳过该步骤；如果没有可用的PostgreSQL数据库，则点击'安装'按钮，等待PostgreSQL安装完成，点击下一步。
3. 在该页面中输入PostgreSQL数据库连接的各个参数，点击下一步。
4. 在该页面中配置Redis连接参数，一般情况使用默认即可，点击安装。
5. 安装完成后目录结构如下图：



6. 双击start.bat启动CVS系统，待启动完成后，使用Chrome浏览器访问 <http://localhost:18080/cvs/conf> 页面，进入安装与升级子页面，点击 系统初始化 按钮，系统安装完成！

使用 Chrome 浏览器访问 <http://localhost:18080/cvs/console> 地址，使用用户名admin 密码 admin 登录系统。

## 1.2 数据代理

从 CVS 3.0.0 版本开始不再使用数据代理程序，iCV、iHD数据接入通过配置数据源信息即可。具体参照3.1 数据源管理章节。

## 1.3 模拟数据源

CVS 系统发布附带了模拟数据源和示例画面，方便用户对系统快速掌握。初始安装的系统可以看到有testds 数据源存在系统中，同时可以看到画面管理下的示例画面文件夹，其中包含了多种类型的示例画面，可以通过查看这些画面学习CVS系统的使用。

## 1.4 系统许可证

CVS 系统许可证分为临时许可证和永久许可证两种，若无许可证，CVS 系统只能运行 2 小时、支持 1000 个数据点。

临时许可证文件可以联系 CVS 产品组签发。签发后的 license.xml 文件放置在 %iMV-WebServer%\license 文件夹下即可。

## 1.5 CVS 配置说明

CVS 可以通过浏览器访问 <http://localhost:18080/cvs/conf/> 修改。完成编辑后需要使用 cvsmgr\cvsmgr 的用户名密码重启服务才会生效。

### 1.5.1 Redis

Redis 数据库已经被集成到 iMV 的 zip 文件中，位于 iMV-DS 目录下。

- pde\_host Redis 服务的 IP 地址，若与 iMV 运行在同一个服务器上，可以配置为 127.0.0.1。
- pde\_port、pde\_auth、pde\_dbnum 分别对应 Redis 配置，若无特殊需要请保持默认。

### 1.5.2 PostgreSQL

有关 PostgreSQL 的参数配置请按照数据库安装、设置过程中的信息填写。

- postgres.host PostgreSQL 数据库的地址，形式为 IP:端口。
- postgres.user PostgreSQL 数据库的用户名。
- postgres.password PostgreSQL 数据库的密码。
- postgres.dbname PostgreSQL 数据库中配置给 CVS 系统使用的数据库名称，请确保 postgres.user 和 postgres.password 有权限访问。

### 1.5.3 网关配置

CVS 可以将组态画面的 URL 地址分享给其他用户，或者集成到第三方系统中。考虑到有可能 CVS 系统被放置在 Nginx 或者其他服务器之后，因此需要配置这三项以供访问：

- proxy\_host 服务对外的 IP 地址
- proxy\_port 服务对外的端口
- proxy\_path 服务对外的相对路径

则分享画面功能里生成的 URL 地址形式

为：`http://proxy_host:proxy_port/proxy_path/share/.....`

**注意：**本功能目前生成的 URL 默认使用 http 协议。

### 1.5.4 集群模式配置

CVS 可以支持从多个子CVS系统中获取数据、报警、发送控制和报警确认等操作。需要配置一个主节点CVS和多个子节点CVS。这样，可以适用大规模点数的情况。

需要在 bpconf.properties 文件中，做如下配置

**注意：**

1.该配置只在产品模式下生效；

2.单个节点只能支持一个数据代理。

1. 配置 cluster\_pde\_enabled 为yes，默认为no。yes即启用集群模式
2. 在 cluster\_pde\_set 中配置各个子节点，子节点的配置模式为 host:port?agent=testds[&pass=secret&iscenter=true&isauth=true]，多个子节点用英文逗号连接。其中 [ ] 中内容为可选项
  - agent 为数据代理名称
  - pass 为数据代理密码
  - iscenter 是否是中心节点，默认为 false
  - isauth 是否是授权节点，默认为 false。
3. 各个子节点无需做额外配置
4. 重启主节点服务器

配置完成后，可以在变量管理中看到来自于各个子节点的内容。

## 1.6 CVS 系统升级

### 1.6.1 将 CVS 系统升级到 2.3.0

CVS系统从低于 2.3.0 版本高于 2.0.0 版本升级到 2.3.0 版本步骤；

1. 运行 stop\_core.bat 脚本停止服务，把 [cvs\\_2.3.0\\_08090618.war](http://product.baosight.com/SupportCenter/content/iMV/download/cvs_2.3.0_08090618.war) ([http://product.baosight.com/SupportCenter/content/iMV/download/cvs\\_2.3.0\\_08090618](http://product.baosight.com/SupportCenter/content/iMV/download/cvs_2.3.0_08090618)) 放置到 /iMV-Server/iMV-Core/webapps/ 目录下并重命名为 cvs.war；
2. 在 /iMV-Server/iMV-Core/ 下建立一个名为 cvs 的文件夹，并在 cvs 文件夹下建立名为 data 的文件夹；
3. 从 [CVS\\_2.3.0\\_08090618.zip](http://product.baosight.com/SupportCenter/content/iMV/download/CVS_2.3.0_08090618.zip) ([http://product.baosight.com/SupportCenter/content/iMV/download/CVS\\_2.3.0\\_08090618](http://product.baosight.com/SupportCenter/content/iMV/download/CVS_2.3.0_08090618)) 中解压 iMV-Server/iMV-Core/cvs/conf 文件夹，将该文件夹拷贝到第二步中建立的 /iMV-Server/iMV-Core/cvs 文件夹下；
4. 修改 /iMV-Server/iMV-Core/cvs/bpconf.properties 文件，更新 Redis 地址和 PostgreSQL 地址（参见 1.5.1 和 1.5.2 章节）；
5. 移动 /iMV-Server/iMV-Core/webapps/cvs/lessees 文件夹到 /iMV-Server/iMV-Core/cvs/data/ 文件夹下；
6. 修改 /iMV-Server/iMV-Core/conf/server.xml 文件，在 <host /> 标签中添加如下内容：

```
<Context docBase="${catalina.home}/cvs/data/lessees"  
path="/cvs/lessees"/>
```

7. 在 /iMV-Server/iMV-Core/conf/tomcat-users.xml 中添加

```
<role rolename="admin" />  
<role rolename="manager-gui" />  
<role rolename="manager-script" />  
<user username="admin" password="admin" roles="manager-gui,manager-  
script,admin" />
```

用户名密码可自定义，此为修改配置后**重启**（使用 stop\_core.bat 和 start\_core.bat 脚本）使其生效。

8. 访问 <http://ip:18080/cvs/conf/install.html> 页面，在 系统升级 一栏点击升级按钮。  
**注意：**登陆如出现租户信息不存在，可先用 admin 用户登陆，然后访问 <http://ip:18080/cvs/conf/debug.html> 添加租户。

### 1.6.2 将 CVS 系统升级到 2.4.0

CVS 系统在 2.4.0 中引入了画面和图库的版本控制功能，并且升级了画面运行态格式，必须按照以下步骤升级系统。（若原先系统版本低于 2.3.0，请先按照 1.6.1 章节准备配置文件。）

1. 访问 <http://ip:18080/cvs/console/index.html> 登录系统
2. 访问 <http://ip:18080/cvs/conf/install.html> 点击升级按钮

升级过程需要的时间依据数据库中的画面总数和图元总数所确定，请耐心等待。

### 1.6.3 将 CVS 系统升级到 2.5.0

CVS 系统在 2.5.0 中提供了日志下载的功能，需要做额外的配置来支持该功能。

修改 /iMV-Server/iMV-Core/conf/server.xml 文件，在 <host /> 标签中添加如下内容：

```
```\xml
<Context docBase="${catalina.home}/logs" path="/cvs/logs"/>
```\
```

### 1.6.4 将 CVS 系统升级到 2.5.1

cvcs系统从 2.5.0 版本升级到 2.5.1 版本前需要进行的配置操作：

将发布包中的war文件拷贝到 /iMV-Server/iMV-Core/webapps/ 下，重命名为cvs.war，然后重启cvs服务。

### 1.6.5 将 CVS 系统升级到 3.0.0

cvcs系统从 2.5.0 版本升级到 3.0.0 版本前需要进行的配置操作：

由于 3.0.0 使用新版本 tomcat，所以完整更新到 3.0.0 需要获取相对应的 tomcat 和 lib 库，所以需要更新整体发布版本。

1. 备份 2.5.0 版本安装目录根目录下 license/license.xml 文件与根目录下 iMV-Core/cvs 文件夹；
2. 解压 cvs 3.0.0 发布包（或者使用 exe 文件安装，可以安装在跟老版本不同的路径下），上一步备份的文件和文件夹对应替换到 3.0.0 中
3. 启动 3.0.0 版本，并进入 http://127.0.0.1:18080/cvs/conf/install.html 页面，点击升级数据库到最新版本 3.0.0 按钮；
4. 重新启动 cvs，即可更新完成。

---

## 2. 数据代理配置

数据代理服务主要负责将底层数据源的数据内容采集到 CVS 的数据服务器中，它的主要配置过程在上面的安装部署[章节](#)已经描述过，可以参照上面的内容进行配置。

如果使用的是最新版本的iCV和iHD系统，可以通过其自带的配置界面设置数据代理相关的属性。

## 3. 系统管理

CVS 系统向用户提供了黑色、蓝色两种主题的界面，用户可以在登录时自由选择（两种界面下功能完全一致）。

以下是 CVS 系统登录后的首页画面，该首页中展示了当前系统的基本信息，包括画面数、图元数、分享画面占比、当前租户等信息。

组态配置服务

[基本信息](#)
[画面管理](#)
[图元管理](#)
[变量管理](#)
[权限管理](#)
[文件下载](#)
[帮助](#)
admin

## 基本信息 许可、数据源等

控制台 > 基本信息

### 许可状态

项目	内容
许可-模式	临时许可
许可-过期日期	2016-09-28
许可-最大终端个数	10
当前-终端个数	0
许可-最大点数	100000
当前-点数	70073

### 数据统计

分享画面占比

未分享239  
已分享30

图元与画面

数量

### 服务信息

画面总数  
**269**

图元总数  
**100**

可用图库  
**4500+**

当前租户  
**user**

### 数据代理

代理名称	类型	是否授权	状态
ihd	IHD	否	未连接
tempds	demojava	否	未连接
testwxn	iCV	否	未连接
lq	iHD	否	未连接
wxnihd	iHD	否	未连接
ihd22	IHD	否	未连接
testds	demojava	否	运行中
hdds	iHD	是	运行中
hdds2	iHD	否	未连接

上海宝信软件股份有限公司 版权所有 1998-2016
Version 1.9.0

最上面是系统导航栏，包括基本信息、画面管理、图元管理、变量管理和文件下载。最右侧是当前登录用户信息。

### 3.1 数据源管理

该模块对当前系统中所有的常规数据源和关系型数据代理进行管理，包括对数据源的添加、修改、查看、删除功能。通过选择数据源列表上方不同的选项卡，可以进入不同的数据源配置界面。对数据源进行添加、修改、删除的操作后，用户在基本信息页的数据代理板块和变量管理的数据源下拉列表中均能直接看到操作所带来的变化。

#### 3.1.1 常规数据源管理

常规数据源目前包括模拟数据源（testds）、iCV数据源、iHD数据源、TSDB数据源，其中模拟数据源是系统自带的。在没有其他数据源的情况下，模拟数据源被设为系统的授权中心。产品模式下不支持TSDB数据源，云模式下只支持TSDB数据源。

常规数据源		关系型数据代理				
常规数据源 <span style="float: right;">+ 添加   -</span>						
名称	类型	主IP	主端口	授权中心	配置中心	是否启用
ihdtest	iHD	192.168.128.141	5673	否	否	是
testds	demojava			否	否	是
scada1	iCV	168.2.9.135	50000	否	是	是
icvtestKeep	iCV	192.168.128.140	50000	是	否	是

- **添加数据源**：点击数据源列表顶部右侧的 添加 按钮，会弹出相应的数据源添加表单。可供添加的常规数据源类型共有3种：iCV数据源、iHD数据源、TSDB数据源。用户应当先选择数据源类型，选定类型后，该类型数据源的必填项会用红色星号标出，且对应的部分配置项也会在表单底部显示（授权中心和配置中心），其中添加iCV数据源时可选择是否为设配置中心，添加iCV和iHD数据源时可选择是否设为授权中心。所有数据源在添加时默认被设置为启用状态。填写好数据源信息，点击确定，即可完成添加数据源的操作。

**注意：**

1. CVS系统中至多只能有一个iHD数据源。
2. iCV数据源的名称必须与某一存在的scada名称一致。
3. iCV数据源需要有一个被设为配置中心，所以添加第一个iCV数据源时必须设为配置中心。
4. TSDB数据源添加前要判断租户：若当前租户下已有TSDB数据源，则无法添加；大数据平台上，当前租户只能看见与之关联的TSDB数据源。

### 添加数据源 ×

数据源类型 *	icv
名称 *	icvtestKeep
主机IP *	192.168.128.140
主机端口 *	50000
备机IP	备机IP
备机端口	备机端口
内网地址	内网地址
外网地址	外网地址
用户名 *	admin
密码 *	admin

是否授权中心       是否配置中心

确定 取消

- **修改数据源**：选中某一数据源后，点击数据源列表顶部右侧的 **修改** 按钮，会弹出相应的数据源修改表单。所有数据源的名称和类型均不可修改。修改状态下，只有模拟数据源可以更改启用状态。修改好数据源信息，点击确定，即可完成数据源的修改操作。
- **查看数据源**：选中某一数据源后，点击数据源列表顶部右侧的 **查看** 按钮，会弹出相应的数据源查看弹窗，用户可以看到该数据源的详细信息。
- **删除数据源**：选中某一数据源后，点击数据源列表顶部右侧的 **删除** 按钮，会弹出相应的数据源删除弹窗。若该数据源为系统的授权中心或配置中心，则无法被删除。点击确定，即可完成数据源的删除操作。

添加数据源的操作。

**注意:**

1.如果用户想要删除某不是授权中心的iCV数据源，且系统中已没有其他iCV数据源，则虽然该iCV数据源为配置中心，但它也可以被删除。

### 3.1.2 关系型数据代理管理

关系型数据代理目前支持的数据库类型为PostgreSQL。

名称	描述	数据库	数据表	时间戳	刷新周期(秒)
rdbtest	这是rdbtest	postgres	testrdb	ts	3000
rdbtest2	这是rdbtest2	db1	datasourceinfo	id	60

- **添加数据代理**：点击数据代理列表顶部右侧的 添加 按钮，会跳转至数据源添加页面。用户需要填写名称、描述（可选）、刷新周期，之后需要选择数据库，在得到的数据表列表中选择一张数据表并将该数据表的某个字段选中设为时间戳，点击确定，即可完成数据代理的添加操作。添加完成后，会跳转回数据源管理主页面。
- **修改数据代理**：选中某一数据代理后，点击数据代理列表顶部右侧的 修改 按钮，会跳转至数据代理修改页面。数据代理的名称不可修改。跳转到该页面时，数据代理的数据库、数据表、时间戳都会在表格中用深色标记出来。修改步骤与添加数据代理类似。修改好数据代理信息，点击确定，即可完成数据源的修改操作。
- **删除数据代理**：选中某一数据代理后，点击数据代理列表顶部右侧的 删除 按钮，会弹出相应的数据代理删除弹窗。点击确定，即可完成数据代理的删除操作。

在添加或修改数据代理的过程中，均可对数据库进行添加、修改、删除的操作。

- **添加数据库**：点击数据库列表顶部右侧的 添加 按钮，会弹出相应的数据库添加表单。选定数据库类型后，表单中的URL和用户名会自动填充。填写好数据库信息，点击测试连接，会提示用户当前配置的数据库是否能连上；点击确定，即可完成数据库添加操作。
- **修改数据库**：选中某一数据库后，点击数据库列表顶部右侧的 修改 按钮，会弹出相应的数据库添加表单。修改好数据库信息，点击测试连接，会提示用户当前配置的数据库是否能连上；点击确定，即可完成数据库修改操作。
- **删除数据库**：选中某一数据库后，点击数据库列表顶部右侧的 删除 按钮，会弹出相应的数据库删除弹窗。点击确定，即可完成数据库的删除操作。

## 3.2 画面管理

该模块对当前系统中所有的监控画面进行管理，包括新增、编辑、删除、分享、预览等功能，同时支持对画面文件夹的创建新增等操作，画面文件夹可以嵌套。



根目录

共 1 个对象

新建画面

新建文件夹



示例画面

- 新建画面：点击新建画面在弹出窗口中输入新画面名称，点击确定，即可新增画面
- 编辑画面：双击画面缩略图，或者点击工具栏上的 编辑 按钮可以打开画面设计器，对画面进行设计
- 预览画面：选中画面后，点击预览按钮，可以查看该画面的运行态效果
- 重命名画面：选中画面后，点击重命名按钮，在弹出的对话框中输入新名称，点击确定，可以对画面进行重命名
- 删除画面：选中画面后，点击删除按钮，在弹出对话框中点击确定，可以删除该画面；点击取消，将不做操作。可以选中单个画面进行删除或同时选中多个画面进行批量删除。也可以同时选中文件夹和画面进行批量删除。
- 导出画面：选中画面后，点击导出按钮，在弹出对话框中点击确定，可以导出画面；点击取消，将不做操作；文件名自动保存到当前设置的浏览器下载路径中（文件格式为pml）
- 导入画面：点击导入画面，弹出选择本地画面文件的弹出框，点击 选择文件 选择多个本地 pml 格式文件，在弹出对话框中点击确定，可以导入画面；点击取消，将不做操作；点击单个文件后的 del 按钮可删除单个选中上传文件；点击 清空 按钮可删除所有选择文件（如下图）。若导入的画面文件中包含图片，则会在文件下方依次列出来。系统中不存在的图片将会上传，已存在的图片可以取消上传勾选。可以点击图片文件右侧的链接预览图片，或者查看服务器上的原图片。

- 全选画面：点击全选画面可选择当前文件夹下全部画面（只含画面）。

## 导入画面

选择文件

- 总览图  del
  - dir1/背景.png[预览]
  - dir2/图标.png[预览]
  - dir2/logo.png[预览][查看原图]

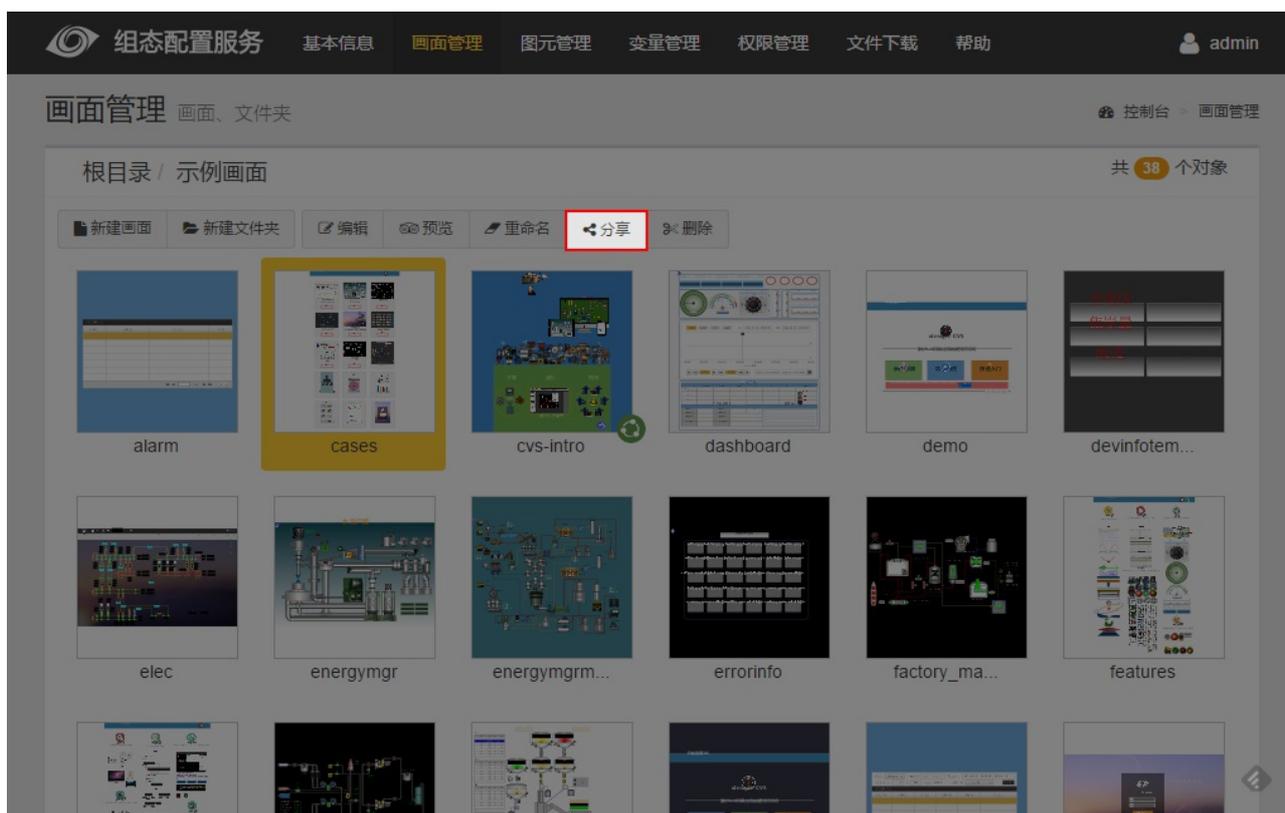
清空

取消

确定

- 分享画面：设计完成的画面可以通过分享操作，嵌入到其他第三方系统中。

首先选中需要分享的画面，点击分享按钮（如下图）



打开分享画面之后，默认为公开链接，输入用户名密码点击生成（因为公开链接可以内嵌在其他系统，或者用浏览器直接打开，不需要验证用户名密码，所以生成前需要输入用户名密码）如下图

# 分享



分享类型:  公开链接  私有链接

admin

admin

http://localhost:18080/cvs/share/eyJpZCI6OTUsInVzZXJuYW1lIjoiYWRtaW4iLCJ0ZW5hbnRJRCI6InRfdXNlcilzInBhc3N3b3JkljoiYWRtaW4iLCJzaGFyZVR5cGUiojB9



取消分享 取消 生成

可看见生成的链接和二维码（二维码可直接扫码登陆）

点击私有链接，不用输入用户名密码可以直接生成（内嵌在其他系统或者用浏览器直接打开需要输入用户名密码如下图）

分享类型:  公开链接  私有链接

`http://localhost:18080/cvs/share/eyJpZCI6OTUsInVzZXJpudWxsLCJ0ZW5hbnRJRjRCI6InRfdXNlcilzInBhc3N3b3JkIjpuZDVsLjZaGFyZVR5cGUoJF9`



取消分享

取消

生成

已分享的画面会在名称前面生成分享图标如下图



右下角的图标为分享图标，带锁的图标为私有链接，不带锁的图标为公开链接

点击取消分享，可以使公开分享链接访问失效（若想再次访问可再次生成）

**注意：**

1. 分享画面所产生的URL链接是根据 `bpconf.properties` 文件中的有关 `proxy` 的三个配置项定义的，请参见 1.5.4 章节。
2. 公开链接、私有链接使用的用户名、密码来自数据代理中配置的权限节点。

- 批量分享：画面管理中多个画面可进行批量分享。

首先选中多个画面，点击批量分享按钮（如下图）



弹出批量分享页面，类似单个分享，可选择公开链接和私有链接（如下图）



分享操作如同单个分享，但不会直接显示分享链接和二维码，可选择单个画面点击分享查看

### 3.3 图元管理

该模块对当前系统中的所有图元进行新增、删除、编辑、导入图元，导出图元，全选图元等操作。（类似画面管理操作，详见画面管理，PS:导入图元不同于画面导入，下详解）

导入图元：点击导入图元，弹出选择本地图元文件的弹出框，点击"选择文件"选择多个本地gml格式文件，在弹出对话框中点击确定，可以导入图元；点击取消，将不做操作；点击单个文件后的"del"按钮可删除单个选中上传文件；点击"清空"按钮可删除所有选择文件；填写新数据源名称则会更新图元中的所有数据源名称（如下图）。有关图元中附带的图片管理和画面导入相同。导入时如选择图元中有同名图元，则会出现提示是否覆盖原有图元，若覆盖则更新至最新导入图元，不可逆；如取消则返回上一级选择图元界面。

## 导入图元



选择文件

- a1.gml
- a2.gml

是否替换画面中已有的引用

新数据源名称

清空

取消

确定

### 说明：

1. 所有的图元都将自动集成到画面设计器中
2. 图元修改之后可以自动更新
3. 图元必须创建在图库文件夹中



- 图元编辑：双击图元或者选中图元后点击工具栏中的编辑按钮

### 3.4 变量管理

该模块对当前系统底层连接的所有数据源进行查看，包括该数据源下的所有点。



在请输入关键字框中输入需要查询的关键字，点击查询，显示与名称字段或描述字段符合的变量信息，查询还支持模糊查询，例如：输入关键字'tag'，点击查询，会显示所有包含'tag'的点。云模式下同样支持查询，但不支持对描述字段的查询。模糊查询的规则以具体数据代理查询的规则为准。

### 3.5 权限管理

该模块对当前系统中的权限进行查看，包括：群组信息、用户信息和授权项信息。

The screenshot shows a web interface for '组态配置服务' (Configuration Service) with a navigation bar containing '基本信息', '画面管理', '图元管理', '变量管理', '权限管理', '文件下载', and '帮助'. The user 'admin' is logged in. The main content area is titled '权限管理' and includes a breadcrumb '组、用户、权限'. It features three expandable sections:

- 群组信息 (Group Information):** A table with columns '名称' (Name) and '描述' (Description). The '所有群组' (All Groups) row is highlighted in yellow. Other rows include 'mvgroup', 'admin', 'groupC' (description: 'goupC desc'), 'aaaaa', 'groupB' (description: '群组描述'), and 'groupA' (description: 'group A desc').
- 用户信息 (User Information):** A table with columns '名称' (Name), '描述' (Description), and '群组' (Group). Rows include 'userA' (description: '描述', group: 'admin'), 'admin', 'userB' (description: '描述'), 'userC' (description: '描述'), 'mvhd' (group: 'mvgroup'), 'userD', 'aaaaa' (group: 'aaaaa'), 'userAB', and 'userABC' (group: 'groupC groupB groupA').
- 授权项 (Authorization Items):** A table with columns '名称' (Name), '描述' (Description), and '权限' (Permissions). Rows include 'authB', 'authC', 'authA', and 'mvitem'.

- 群组信息：可以点击【所有群组】或者单击某个群组。点击【所有群组】，用户信息和授权项信息会显示当前系统内的所有用户信息以及所有授权项信息。点击单个群组，用户信息和授权项信息只显示当前所选中的群组内的用户以及该群组用户拥有的授权项。
- 用户信息：默认显示所有用户信息，包括用户名称、描述和所属群组。只有选择特定的群组后才会显示当前群组的用户。
- 授权项：默认显示所有授权项，包括授权项名称、描述和权限。只有选择特定的群组后才显示当前群

组的授权项。

## 3.6 帮助

该界面提供帮助手册和用户指南的查看，以及系统备份和文件下载功能



其中帮助手册为全面系统介绍CVS系统使用和配置的文档，可以作为参考手册使用；用户指南是一个快速指南文档，用户在初次使用的时候快速搭建可用的系统。

备份系统功能是将当前系统的所有用户配置内容，包括画面、图元、脚本、图片以及上传文件进行全部打包，并提供用户下载，使用下载后的打包文件，可用在相同版本的CVS系统中进行系统恢复，恢复后的系统与之前的备份系统保持完全一致。

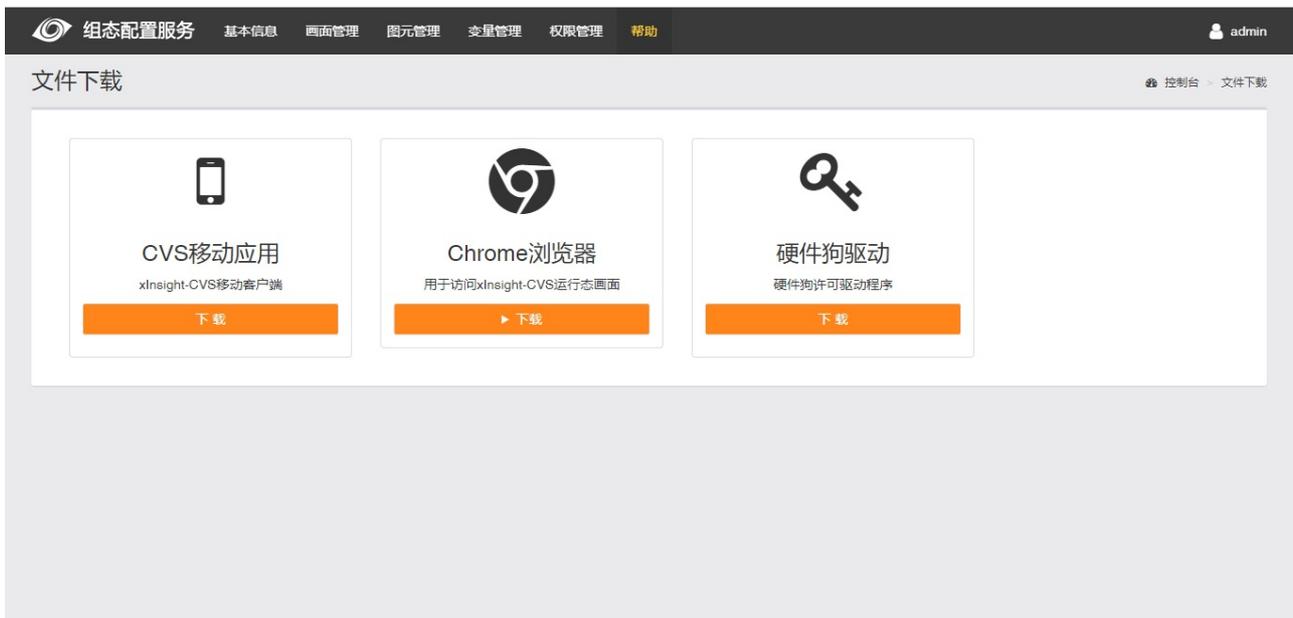
系统恢复将清空目标系统的所有内容，恢复为打包文件的系统内容。



点击立即备份按钮，生成系统备份文件，将该文件保存到本地磁盘。

系统恢复首先点击 '选择备份文件' 按钮，在弹出对话框中选择已经备份的zip文件，选择上传，上传成功后，点击'恢复系统'按钮，等待系统恢复完成。

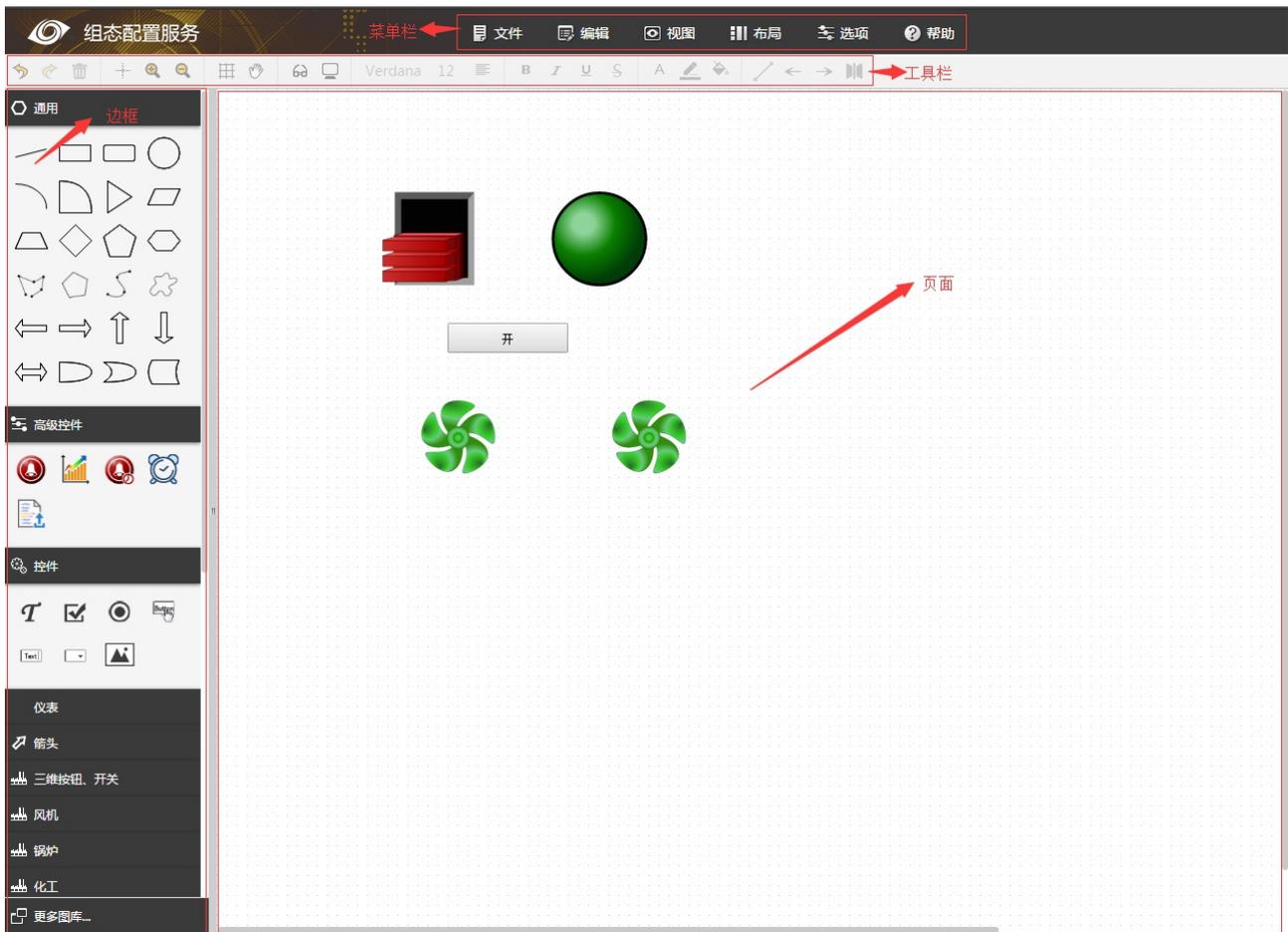
文件下载功能为用户提供了当前系统中他们所需的下载文件。点击帮助界面上的文件下载，会跳转到下图所示的文件下载界面，点击下载按钮，即可下载对应文件。



## 4. 画面设计

通过双击画面管理界面中的文件，或者点击编辑按钮可以打开画面进行设计。设计完成的画面使用F6快捷键，可以预览查看。

## 4.1 画面设计器介绍



- 边框中包含所有可以绘制的图形、图元和图库。鼠标移到图形上会有预览框，及信息提示。
- 中间空白区域为页面绘图区
- 右上为功能菜单
- 左上图标是工具栏。鼠标移到图标上会有信息提示。

## 4.2 功能菜单

设计器的菜单区域包含了文件、编辑、视图、布局、选项和帮助菜单内容。

### 4.2.1 文件菜单

文件

编辑

保存

Ctrl+S

签入

导出

导入...

页面统计信息

查找与替换

资源配置

画面历史

- 保存：保存当前画面（快捷键 ctrl + s），保存结果会在右上角提示。



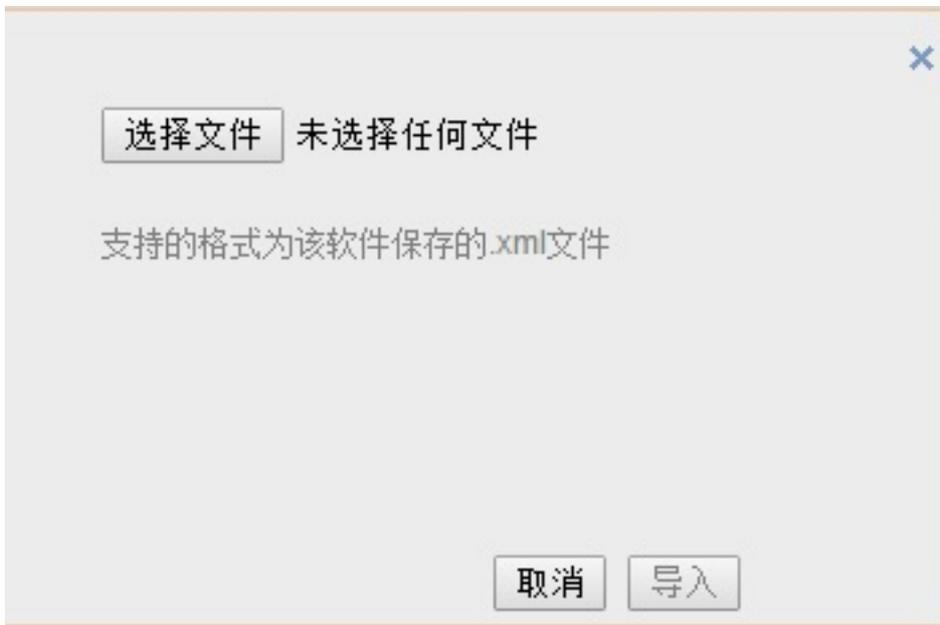
- 签入：将会弹出日志提示框。作者、邮箱、备注三个字段均可留空，但是建议按照实际情况填写，以便于画面历史记录查询。该操作同样会**保存**画面。

**保存日志** ✕

作者	<input type="text"/>
邮箱	<input type="text"/>
备注	<input type="text"/>

- 页面统计信息：弹出对话框展示当前页面中元素信息

- 页面导出：导出当前页面并存在当前浏览器设置的下载路径（文件格式为pml，文件名为当前页面名）
- 页面导入：导入本地存储的画面（文件格式为pml）。pml 文件中附带的图片管理和画面管理中一致。



- 点击刷新按钮，刷新显示信息
- 在查询框输入查询条件，可以列出查询结果
- 查找与替换：弹出对话框，可以查找与替换变量/文本内容

### 查找与替换

操作对象 变量

查找内容  查找

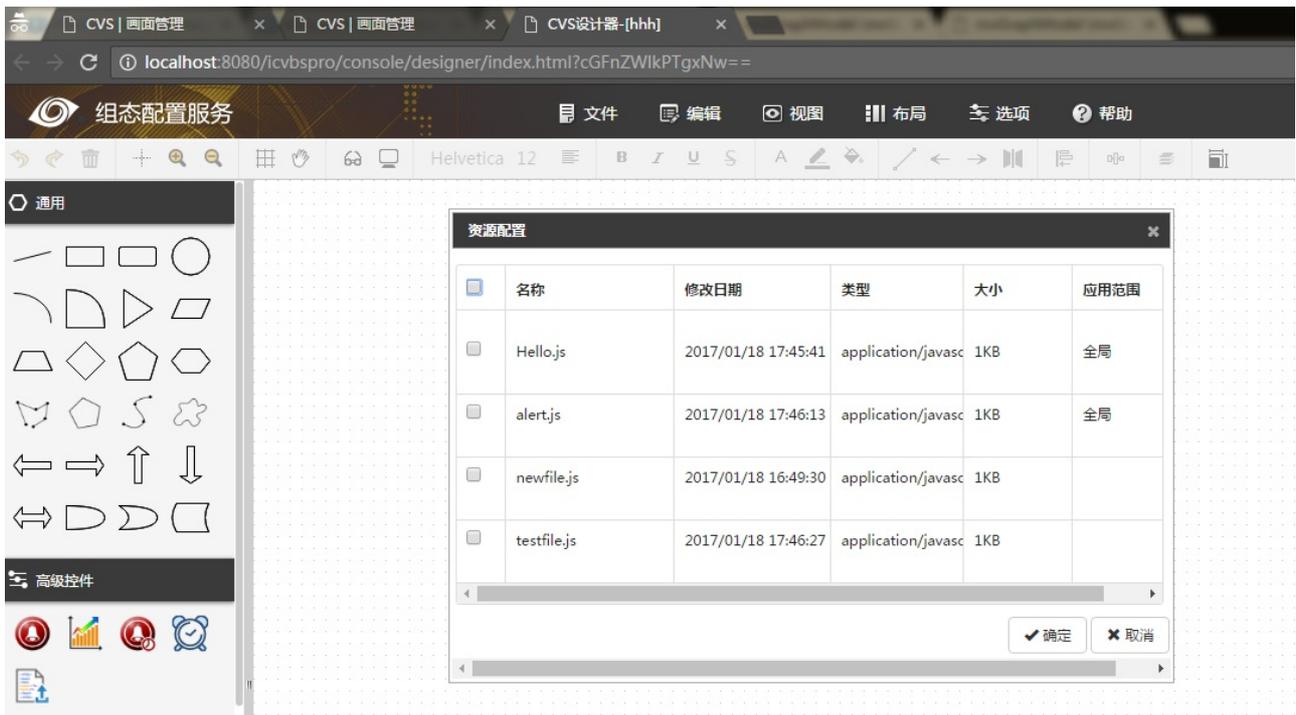
替换为  替换

只替换选中对象

选择	对象	属性	内容
----	----	----	----

查找与替换支持三种操作对象：变量、文本、模板对象。都支持模糊查询。

- 下拉框中选择变量，输入查询变量，点击查询按钮，将配置该变量的元素及属性显示在列表框中
- 在列表框中勾选要替换的元素属性，勾选只替换选中对象
- 输入要替换后的变量，点击替换，将勾选的对象属性替换
- 若不勾选只替换选中对象，默认替换全部变量
- 在下拉框中选择文本，输入查询内容，点击查询按钮，将配置了该内容的文本对象全部列出
- 输入要替换的内容，将选中的文本对象中的内容全部替换
- 在下拉框中选择模板对象，输入查询内容，点击查询按钮，将配置了该内容的模板对象全部列出
- 输入要替换的内容，将选中的模板对象中的内容全部替换



- js类型文件的配置对话框
- 选中一行，然后在应用范围一列中选择该文件的应用范围
- 空白，则是不配置
- 全局表示该js文件所有的画面都可以使用
- 本地表示该js文件只有当前画面可以使用



点击画面历史按钮会弹出查看历史对话框。对话框内由近及远列出这个画面的所有历史操作。版本列表中每条记录会依次显示版本号（7位十六进制数，用蓝色显示）、编辑时间、备注和作者。

- 由近及远的含义表示历史版本越往上，约接近查看的时间。最上方的记录表示最新版本。
- 历史操作包含：创建、导入、重命名（含文件夹）、删除。

可以勾选任意一条版本记录，并点击 查看该版本 按钮在设计器中查看该历史记录。（请注意，当前画面中未保存的内容将会丢失）可以通过查看该历史版本的功能再次保存或者编辑画面。

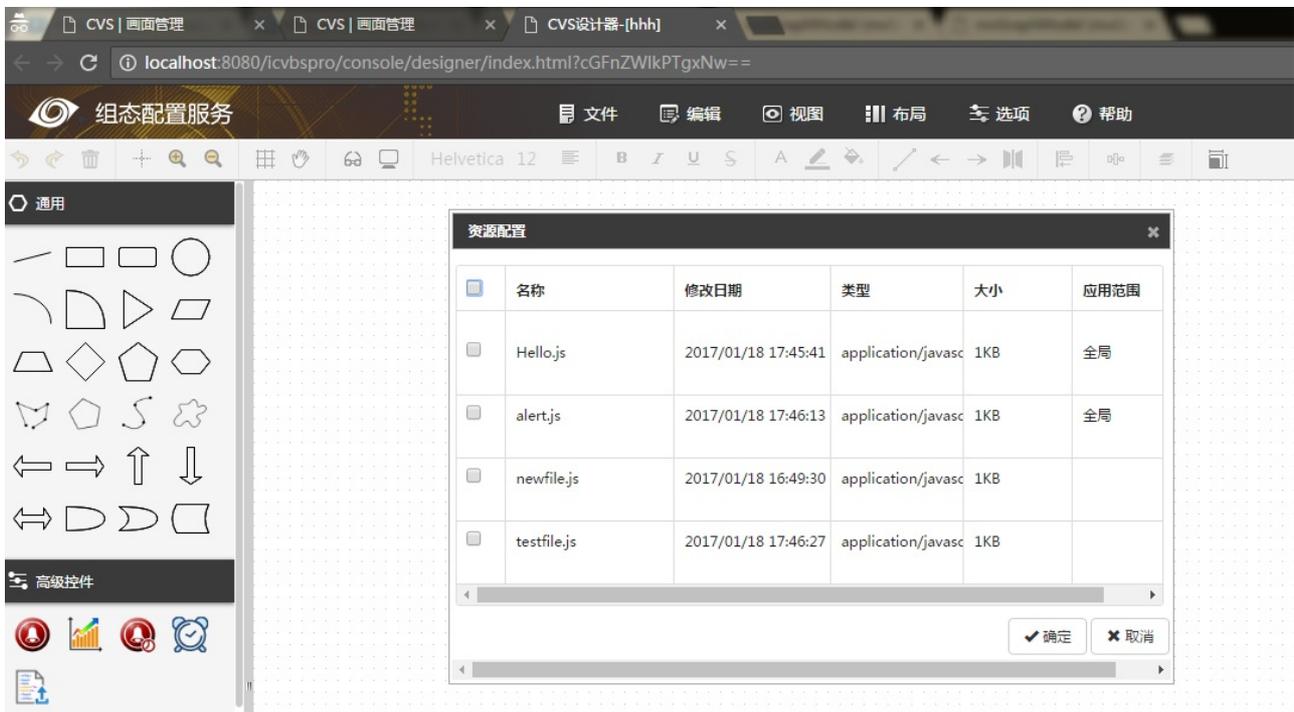
可以勾选任意两条版本记录，并点击 对比差异 按钮，将会弹出对比结果。

```

example.pml CHANGED
@@ -4,9 +4,12 @@
4   <root>
5   <mxCell id="0"/>
6   <mxCell id="1" parent="0"/>
7 - <mxCell id="2" value="中文" style="text;
8     <mxGeometry x="70" y="125" width="120"
9   </mxCell>
10  </root>
11  <script><![CDATA[]]></script>
12 </mxGraphModel>

4   <root>
5   <mxCell id="0"/>
6   <mxCell id="1" parent="0"/>
7 + <mxCell id="2" value="中文" style="text;
8     <mxGeometry x="70" y="125" width="120"
9   </mxCell>
10 + <mxCell id="3" value="测试" style="text;
11 +   <mxGeometry x="170" y="120" width="120"
12 + </mxCell>
13 </root>
14 <script><![CDATA[]]></script>
15 </mxGraphModel>

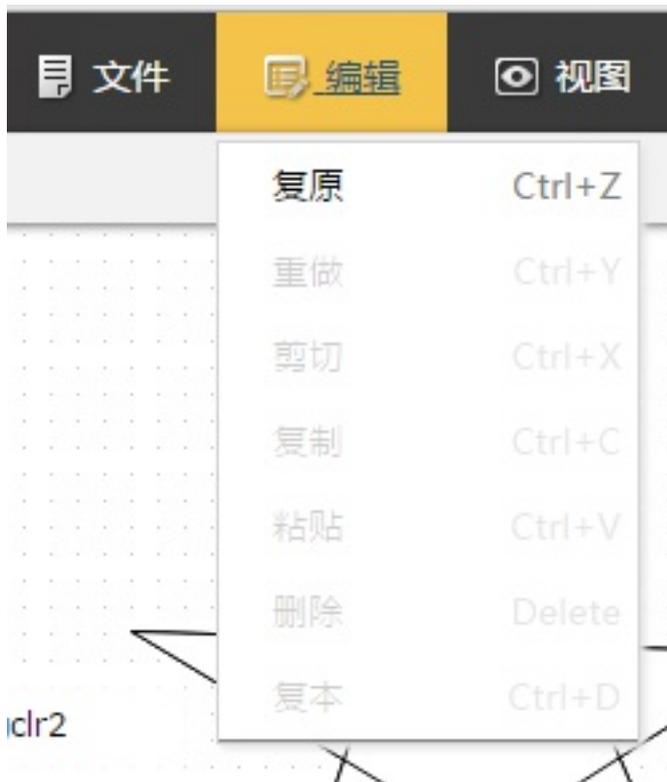
```



- 对当前画面的校验。
- 如果检验正确显示：XML文件通过XSD文件校验成功
- 如果校验失败显示：XML文件通过XSD文件校验失败！
- xsd文件通过trang.jar生成，生xsd的脚本例如：  
java -jar c:\trang.jar standard\_designer.xml standard\_designer.xsd
- 可以校验mxCell是否唯一

#### 4.2.2 编辑菜单

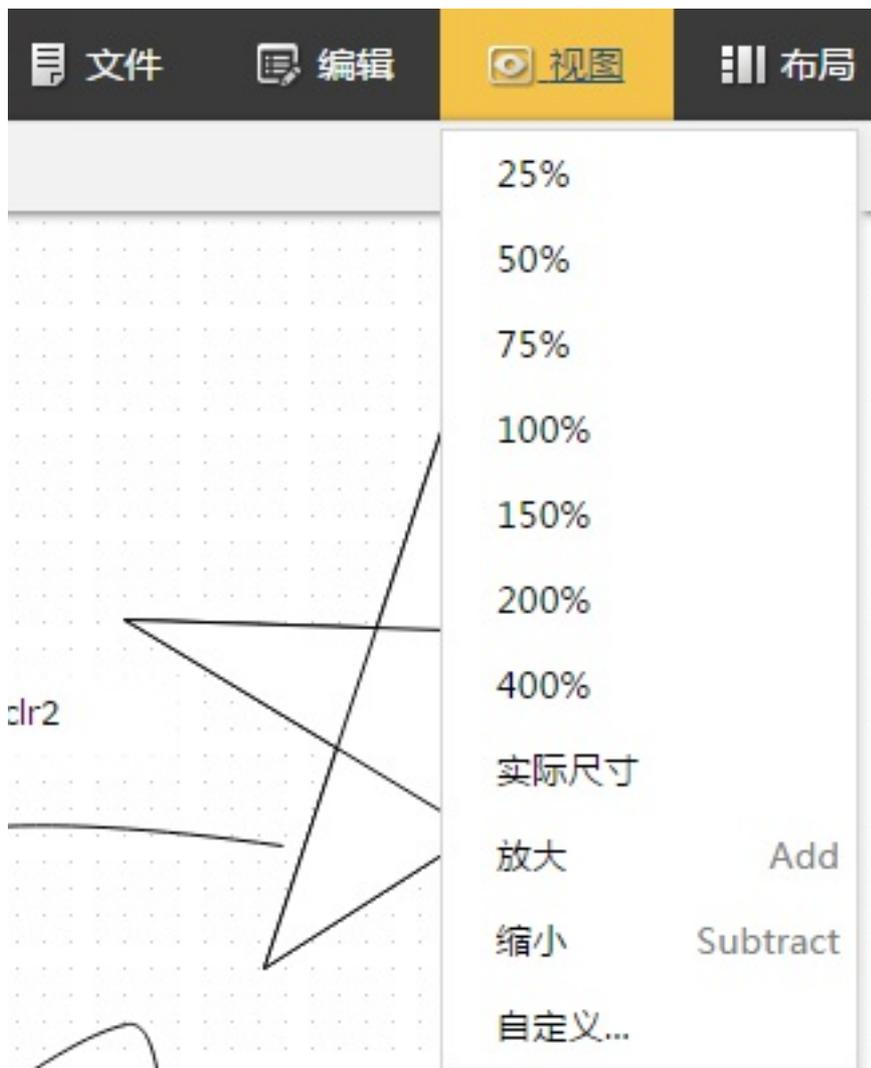
可以对元素进行复原、重做、剪切、复制、粘贴、删除和复本等操作



- 撤销：点击按钮，可以撤销前一次执行的操作。连续激活此命令，可以撤销到打开或者新建画面以来的多次执行的操作。快捷键：Ctrl+Z
- 重做：点击按钮可以重复执行前一次撤销的操作。快捷键：Ctrl+Y
- 剪切：首先选择要剪切的对象，点击按钮。当前所选中对象被拷贝到剪贴板上，对象本身从页面中清除。快捷键：Ctrl+X
- 复制：首先选择要复制的对象，点击按钮。当前所选中对象被复制到剪贴板上，对象本身不从页面中清除。快捷键：Ctrl+C
- 粘贴：首先进行复制或剪切操作，把要粘贴的内容拷贝到剪贴板上，然后点按钮剪贴板上的内容被拷贝到页面上。注意：被复制或剪切的对象会与它的属性一起保存在剪贴板中。在该对象被粘贴到页面后，它的所有特性均会保持不变。快捷键：Ctrl+V
- 删除：首先选中要删除的对象，点击按钮，将对象从页面中删除。快捷键：Delete
- 复本：首先选择要复本的对象，点击按钮，将对象复制一份生成到页面中。相当于复制+粘贴操作之和。快捷键：Ctrl+D

### 4.2.3 视图菜单

可以对当前视图进行缩放操作



#### 4.2.4 布局菜单



- 选中画面中单个元素，通过
- /向后，可以调整元素的z轴位置，
- 向上：将对象上移一层
- 向下：将对象下移一层
- 置顶：将对象放置到所属图层的顶层
- 置底：将对象放置到所属图层的底层
- 选中画面中元素，通过方向菜单，可以绘制元素的镜像
- 水平镜像：将对象做一个水平镜像
- 垂直镜像：将对象做一个垂直镜像

选中要对齐的多个元素，点击对齐菜单，选择对齐方式



将选中元素按照对齐方式对齐。

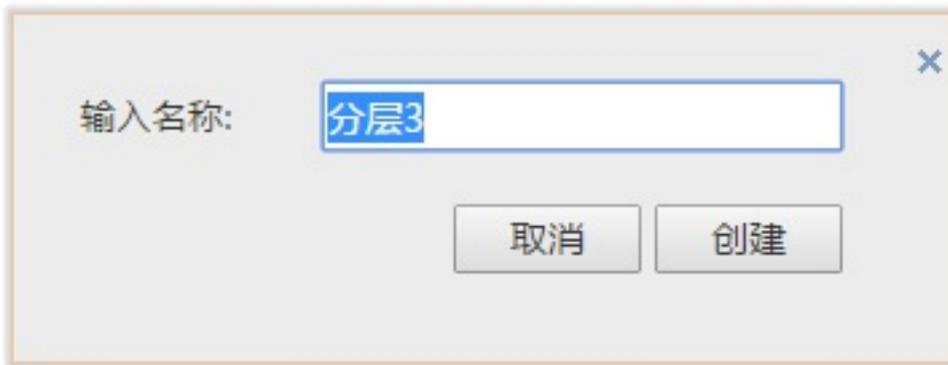
可以对多个元素进行分布操作，选中多个元素，点击菜单中的分布子菜单，选择水平或者垂直。水平菜单将对所有元素在水平方向上进行平均分布；选择垂直菜单，将对选中所有元素进行垂直方向的平均分布。

- 选中多个元素，可进行组合，取消组合等操作
- 组合：可以对多个对象进行成组操作，选中要组合的多个对象，点击按钮，或者右键菜单中选择“组合”，完成组合操作。
- 取消组合：选中需要拆分的已组合的对象，点击按钮或右键菜单选择“取消组合”，完成拆分操作。

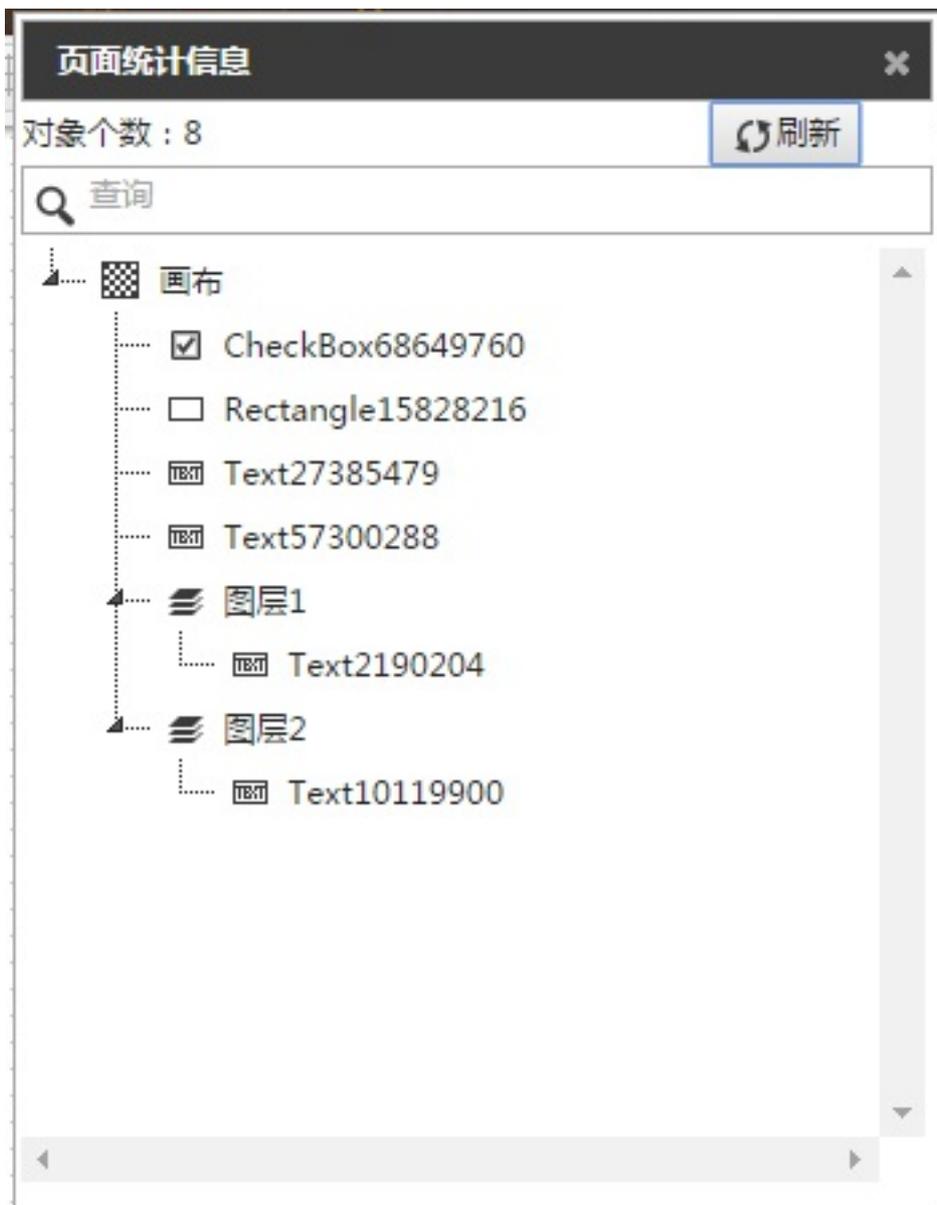
可以对元素进行分层操作，以便控制元素的显示隐藏。点击分层按钮，弹出图层对话框



对话框中会显示画面中已经配置的图层。默认图层为背景。  
点击新增按钮，弹出新增图层对话框



输入名称，创建即可增加图层。  
点击图层，在画面中选择要移动到该图层的元素，点击按钮“将选择移动到 图层2”，系统会将选中的元素移动到相应的图层。在菜单文件--页面统计信息中可查看当前元素分组情况



点击图层后面上移下移箭头，可以移动图层显示顺序  
点击删除按钮可删除图层，图层中的元素会直接设置到背景图层中。  
点击图层前的选择框，可以设置图层是否显示隐藏。

#### 4.2.5 选项菜单

网格：画图区的网格显隐



指导：画图区的指导线功能是否启用进行配置。

## 4.2.6 帮助菜单

可以查看帮助信息和用户手册

## 4.3 工具栏

在画面设计器的上方会提供元素属性设置的快捷方式，用户可以通过工具栏上提供的工具按钮对元素的填充颜色、字体等属性进行快速的配置。工具栏见下图



鼠标置于图标上会显示提示信息。工具栏提供的功能主要包括：复原、重做、删除、实际尺寸、放大、缩小、网格、指导、预览、视图、字体、字体大小、对齐、粗体、斜体、下划线、删除线、字体颜色、线条颜色、填充颜色、线、终点箭头、起点箭头、方向、对齐、分布、分层、大小。

- **复原** 在画面和图元设计器中，将控件拖放到画布、删除、设置字体等操作可以通过复原操作恢复。



- **重做** 重做操作可以对已经复原的操作进行回退。



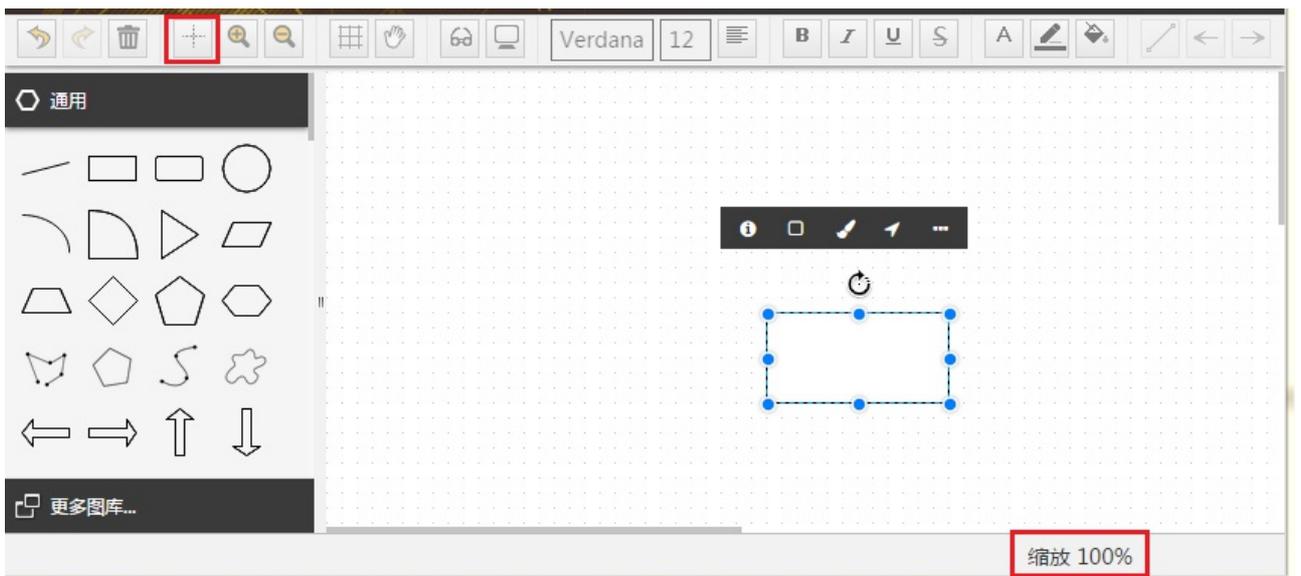
- **删除** 在画面设计器和图元设计器中，对已经拖放到画布的控件选中后点击删除按钮即可删除。



- **实际尺寸** 画布缩放和放大以后,点击这个按钮，画布回复到原来的实际尺寸。



即缩放比例为100%，在画布的右下角可以看到，如下图。



- **放大** 点击放大按钮后画布在原来的大小基础上放大20%。



- **缩小** 点击缩小按钮后画布在原来的基础上缩小20%。



- **网格** 点击网格可以为画布设置网格或者取消网格。



- **预览** 点击预览按钮，可以查看运行态。



- **视图** 视图工具栏提供四种可以查看画面设计器运行态的方式。包括：ios,android,ipad和pc。



窗口高度分别如下：

- ios 360x548
- android 548x348
- ipad 1000x768
- pc 根据当前pc分辨率决定



- **字体** 设计器中的text、checkBox、radioButton、button、input和comboBox支持字体设置。



- **字体大小** 设计器中的text、checkBox、radioButton、button、input和comboBox支持字体大小设置。



- **对齐** 设计器中的text支持对齐设置。



对齐的样式包括：左、垂直居中、右、上、水平居中、下。



- **粗体** 设计器中的text、checkBox、radioButton、button、input和comboBox支持粗体设置。



粗体效果如下：

**Text**

- **斜体** 设计器中的text、checkBox、radioButton、button、input和comboBox支持斜体设置。



斜体效果如下：

*Text*

- **下划线** 设计器中的text、checkBox、radioButton、button、input和comboBox支持下划线设置。



下划线效果如下：

Text

- **删除线** 设计器中的text、checkBox、radioButton、button、input和comboBox支持删除线设置。



删除线效果如下：



- **字体颜色** 设计器中text支持字体颜色设置。



- **线型颜色** 设计器中除了checkbox、radioButton、button、input和comboBox、image外所有控件都支持线型颜色的设置。



- **填充颜色** 设计器中除了checkbox、radioButton、button、input和comboBox、image外所有控件都支持线型颜色的设置。



**注：**所有的颜色选择框中已经被选中的颜色的十六进制码会显示在input框中，点击应用即可生效。

颜色配置框如下：



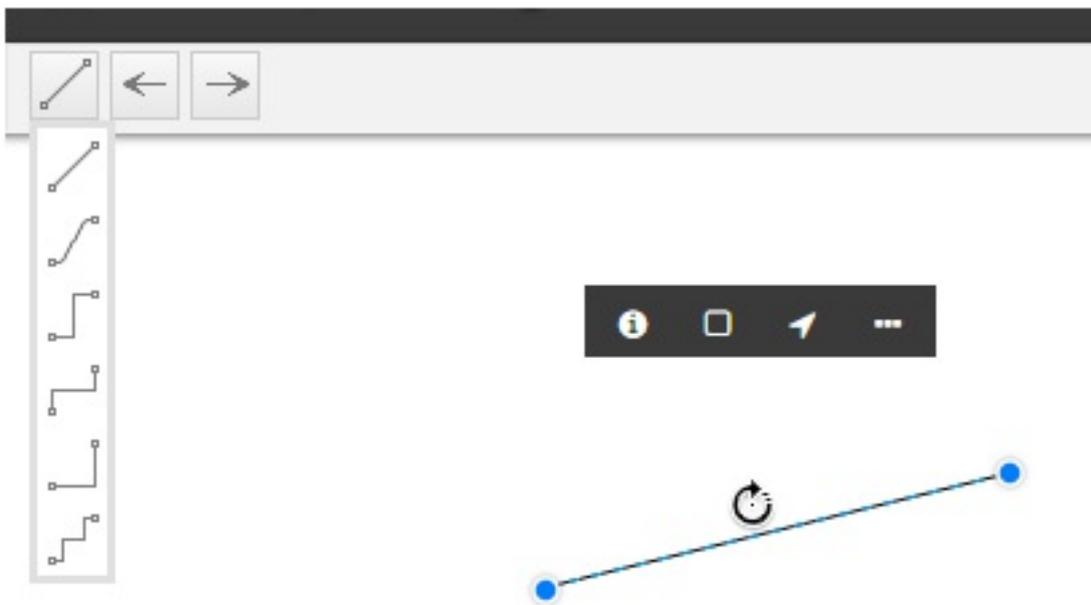
- **线** 直线支持线的设置。



可以设置为直线、实体关系、水平、垂直、手动和自动6模式。



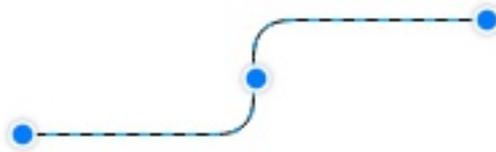
- 直线，效果如下：



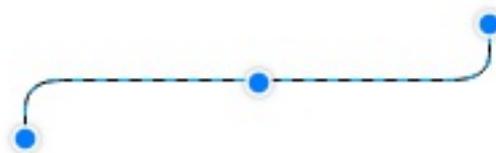
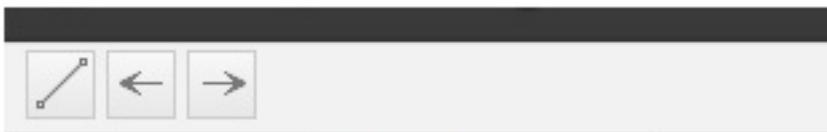
- 实体关系，效果如下：



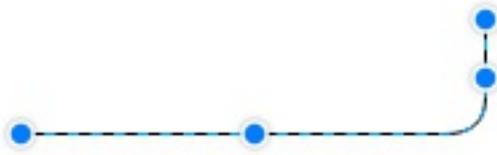
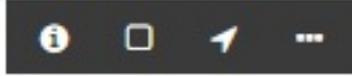
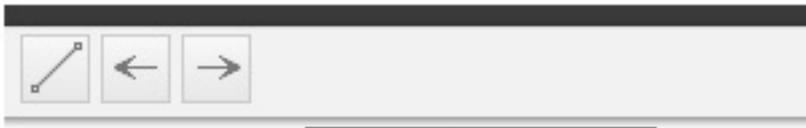
- 水平，效果如下：



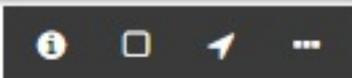
- 垂直，效果如下：



- 手动，效果如下：



- 自动，效果如下：



注：其中，手动和自动时除了终点和起点外的点移动后会多生成一个点来绘制图形。

- **起点箭头** 设计器中除了checkbox、radioButton、button、input和comboBox、image外所有控件都支持线型颜色的设置。



直线可以设置起点箭头。起点箭头的类型包括：

- 实心经典、开放、块、椭圆、菱形、菱形（细）
- 空心经典、开放、块、椭圆、菱形、菱形（细）
- **终点箭头** 设计器中除了checkbox、radioButton、button、input和comboBox、image外所有控件都支持线型颜色的设置。



直线可以设置终点箭头。终点箭头的类型包括：

- 实心经典、开放、块、椭圆、菱形、菱形（细）
- 空心经典、开放、块、椭圆、菱形、菱形（细）
- **方向** 设计器中除了线、checkbox、radioButton、button、input和comboBox、控件、组合图表控件外都支持方向的设置。

方向包括水平镜像和垂直镜像。

- **对齐** 设计器中提供对齐功能，选择两个及以上元素，在工具栏上点击对齐按钮，选择对齐方式，元素自动按照选择的对齐方式对齐。



对齐方式与菜单中的对齐方式一样，包括：左对齐，垂直居中，右对齐，上对齐，水平居中，下对齐。

- **分布** 设计器中提供设置元素分布功能，选择两个及以上元素，在工具栏上点击分布方式，元素自动按照选择的分布方式分布。

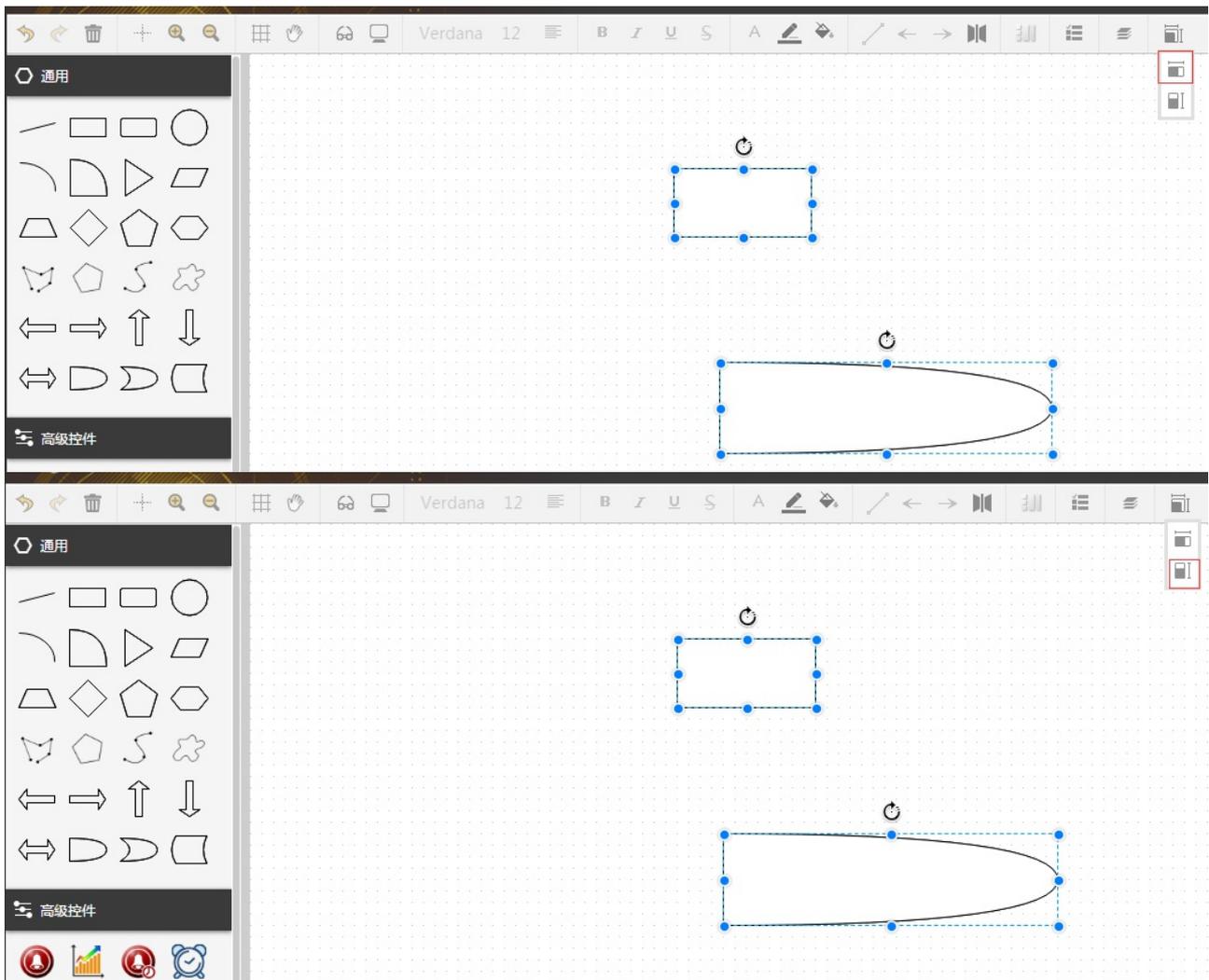


分布方式与菜单中的分布方式一样，包括：水平分布和垂直分布。

- **分层** 点击工具条上的分层按钮，可以配置元素的分层。



- **等宽等高** 设计器中除了line、Arc、Polyline、Curveline外所有控件都支持线等宽登高设置。当选中元素超过2个时，可设置等宽和等高。



## 4.4 元素

通用中包含直线，矩形，圆角矩形，椭圆，折线，多边形，曲线，闭合曲线,弧形，梯形、箭头等

高级控件中包括趋势图控件，报警控件，历史报警控件

控件中包括文本、复选框、单选框、按钮、输入框、下拉框、图片

图表中包括柱状图、饼图、折线图

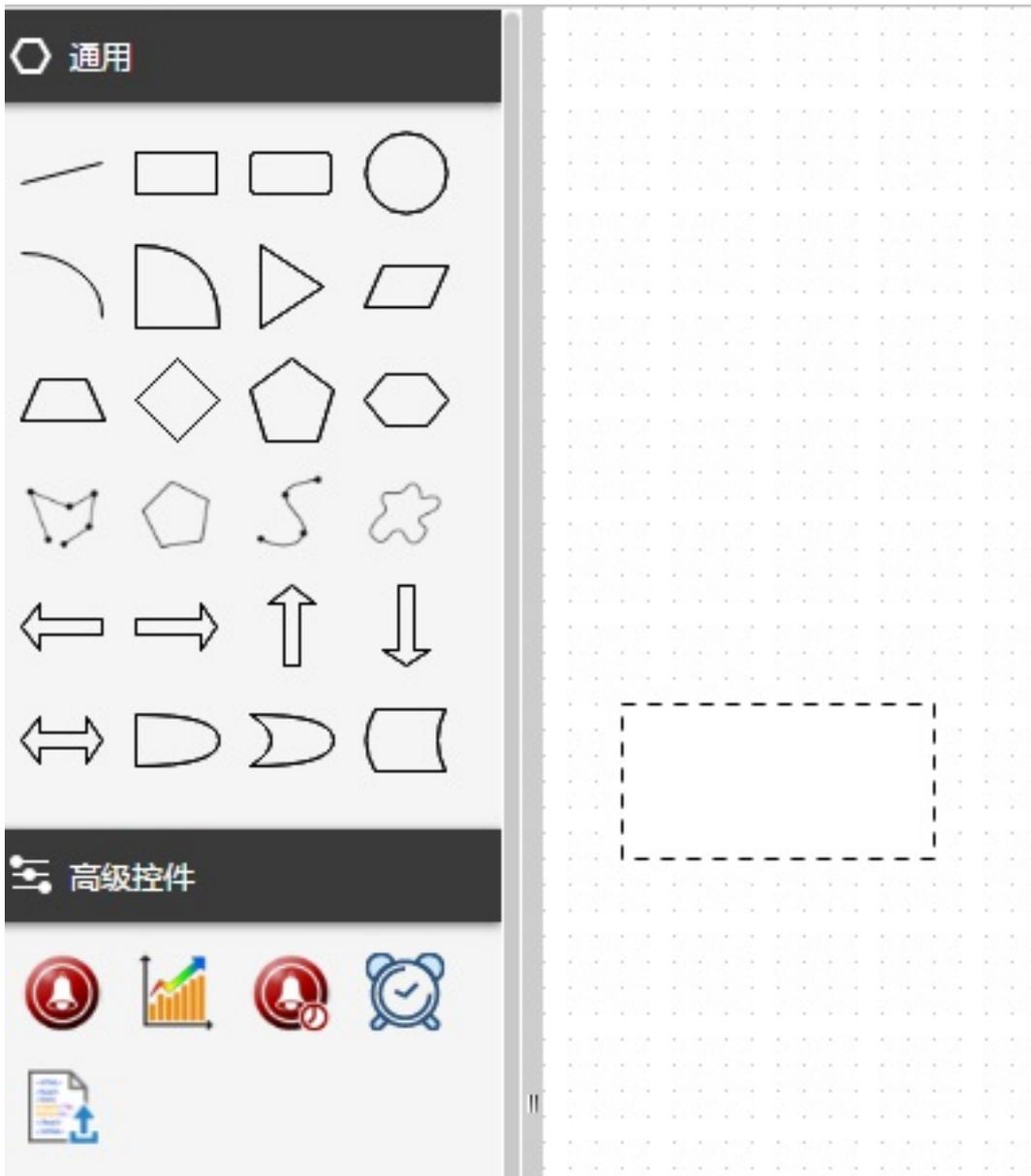
### 4.4.1 基本元素

基本元素包括通用分类下面的直线、矩形、圆角矩形、椭圆、折线、多边形、曲线和闭合曲线元素。

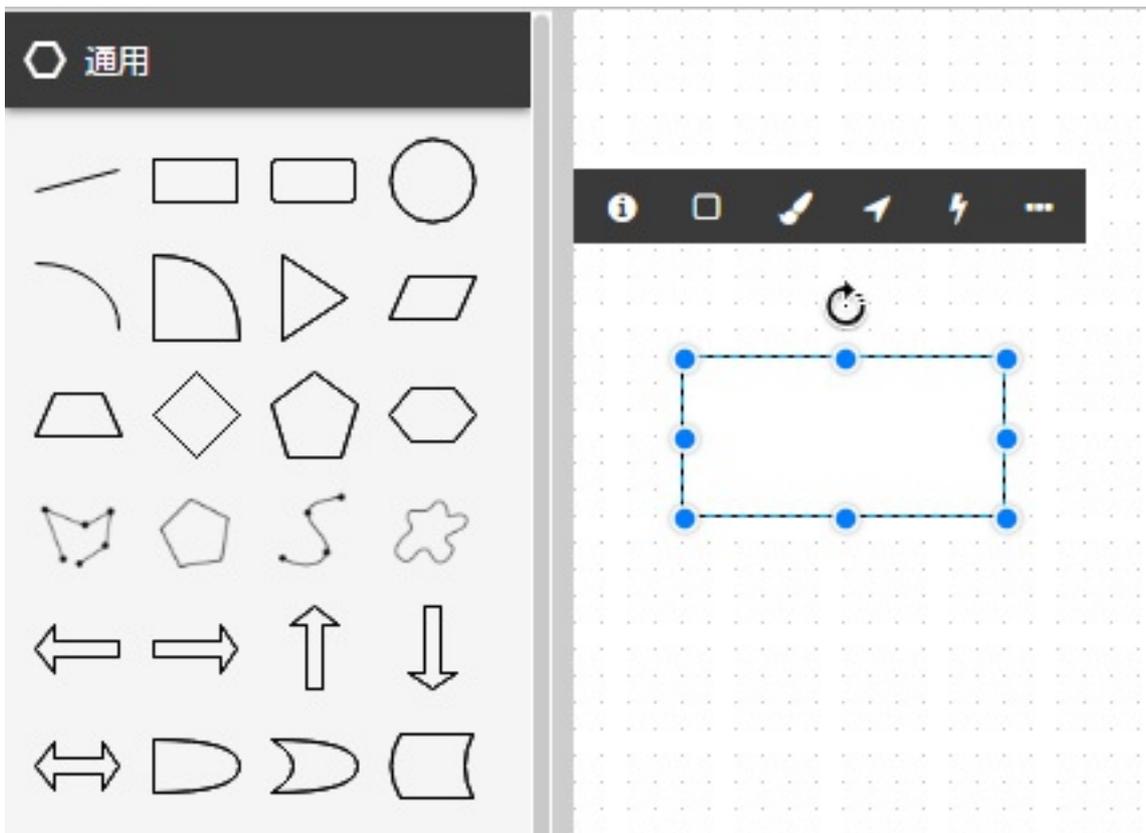
鼠标移动元素上方之后将显示该元素的缩略图。



除过线型和图片元素外，其他所有图形元素拖放到画面上，会出现元素的虚线框。



鼠标移动到要放置的位置时，松开鼠标即可将元素添加到画面上。按住元素的顶点可以对元素进行移动和缩放。点击旋转按钮可对元素进行旋转。



#### 4.4.2 报警控件

在画面设计器左侧“高级控件”栏，双击或选中报警控件拖动到画面设计器中，进行编辑。为了保证画面报警控件的可读性、美观，请依据配置的列宽合理设置报警控件的宽度和高度。



右键报警控件，弹出菜单，选中报警配置进入报警配置界面。

✕	删除	Delete
✂	剪切	Ctrl+X
📄	复制	Ctrl+C
📄	复本	Ctrl+D
▶▶	向前	
◀◀	向后	
⚙	报警配置...	

- 基本设置

基本设置包含九个配置项：隐藏标题、隐藏表尾、固定表头、固定表尾、是否多选、是否使用自定义通知脚本、分页设置、默认分页和刷新周期。

基本设置	过滤配置	列选项	报警设置	✕
隐藏标题	<input type="checkbox"/>	隐藏表尾	<input type="checkbox"/>	
固定表头	<input type="checkbox"/>	固定表尾	<input type="checkbox"/>	
是否多选	<input checked="" type="checkbox"/>	自定义通知脚本	<input type="checkbox"/>	
分页设置	<input type="text" value="10,20,30"/>			
默认分页	<input type="text" value="1"/>			
刷新周期	<input type="text" value="5000"/>			

默认情况下的配置如图所示。

当勾选上“隐藏标题”时，运行态将不会显示“实时报警列表”字样。

“隐藏表尾”指的是运行态将不再显示过滤设置、翻页、确认报警等交互按钮。这种情况下可以通过脚本来实现报警过滤设置和确认报警的功能，详情请参见运行态报警控件截图和系统函数中有关说明。

“固定表头”和“固定表尾”分别是指表头和导航栏分别固定在控件的最上方和最下方，不随控件的竖直滚动而改变位置。

不启用多选时，运行态将无法同时选中多条报警。

自定义通知脚本允许用户替换掉默认的 toastr 通知方式。启用后需要定义 errorMsgCallback 函数，相关说明参见 4.6.4.4 章节。

报警控件默认启用分页方式显示报警数据，默认情况下可以选择按照每页 10 条、20 条或者 30 条数据的显示方式。当配置了分页设置的值后，默认显示方式将失效。配置示例如下：

### 输入 规则

10 按照每页 10 条记录的方式分页。

10,20 默认按照每页 10 条记录的方式分页，用户可以在下拉列表中选择每页 10 条或者 20 条记录的方式。

0 在一页里显示**所有记录**。请注意，报警记录较多的情况下，这种方式可能加载速度会很慢，请谨慎使用。

a **不合法的**配置字符串，将按照默认 "10,20,30" 的分页模式进行。合法的字符串形式参见以上示例：非负整数，或者使用半角逗号，分割的非负整数。

指定默认分页号，将会在画面刚运行时就显示该页的报警信息。

可以按照项目需求调整报警数据的刷新周期，时间单位为毫秒。最小刷新周期为 10 毫秒，小于该数值的配置在运行态不会生效。

- 过滤配置

每种配置前选中框可选择是否启动此项过滤，齿轮状按钮可打开配置此项过滤界面。画面运行态可以修改过滤条件，但是无法保存配置。

基本设置	过滤配置	列选项	报警设置	✕
<input type="checkbox"/>	启用报警区域过滤	<input type="text"/>		
<input type="checkbox"/>	启用报警状态过滤	<input type="text"/>		
<input type="checkbox"/>	启用报警时间过滤	<input type="text"/>		
<input type="checkbox"/>	启用报警优先级过滤	<input type="text"/>		
<input type="checkbox"/>	启用变量名称过滤	<input type="text"/>		
<input type="checkbox"/>	启用报警类型过滤	<input type="text"/>		
<input type="checkbox"/>	启用数据代理过滤	<input type="text"/>		
<input type="checkbox"/>	启用报警点描述过滤	<input type="text"/>		
<input type="checkbox"/>	启用子系统过滤	<input type="text"/>		

1. 报警区域过滤设置，可增加或删除报警区域。配置区域报警后，只要报警数据中六个区域数据符合列表中的任意一条名称记录，该条报警将显示。

### 报警区域过滤设置 ✕

<input type="checkbox"/>	报警区域名称
--------------------------	--------

---

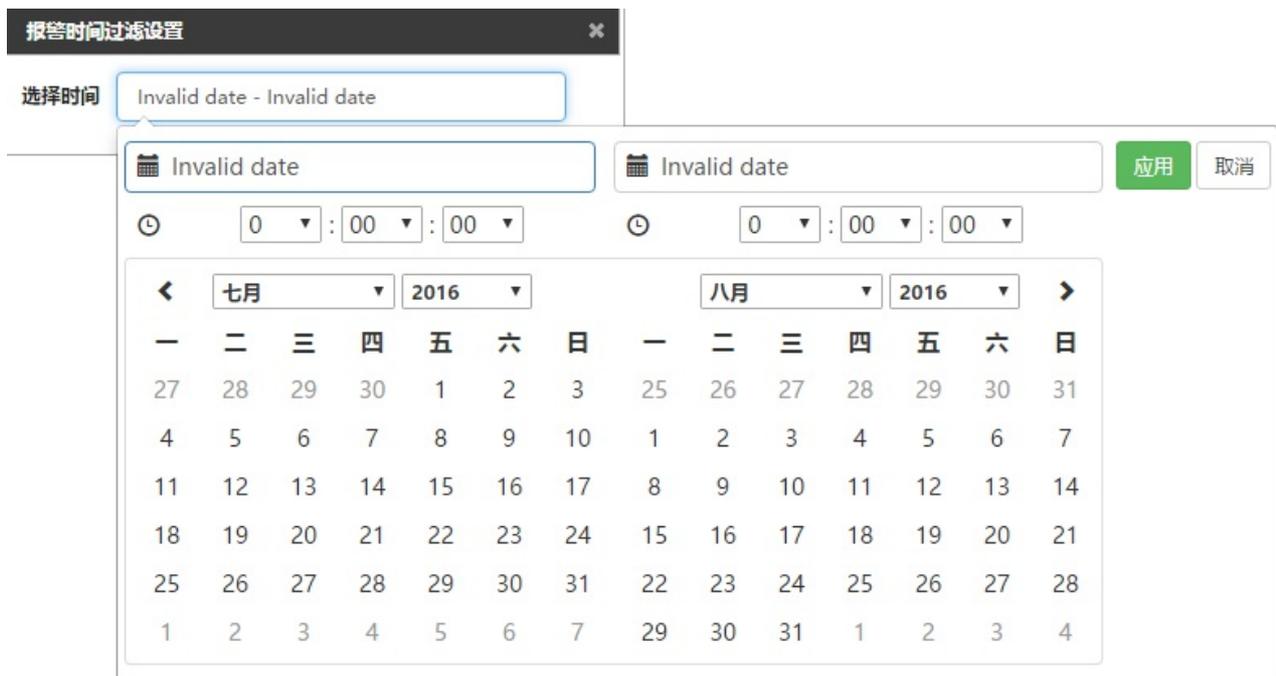
增加 删除

2. 报警状态过滤设置，可勾选不同报警状态，勾选后只显示对应报警状态。

### 报警状态过滤设置 ✕

未确认未恢复       已确认未恢复       未确认已恢复

3. 报警时间过滤设置，点击选择时间输入框可进入选择报警时间界面。



4. 报警优先级过滤设置，输入显示区域，例如左右单元格分别输入 1、3，则显示 1、2、3 三个优先级的报警信息。



5. 报警变量名称过滤设置。在弹出的配置界面中输入变量表达式，变量表达式支持通配符、正则等多种形式。该功能与 [icv 的报警变量表达式](http://product.baosight.com/SupportCenter/doc/icv_doc_new/index.html?icv_alarm_monitor.htm) ([http://product.baosight.com/SupportCenter/doc/icv\\_doc\\_new/index.html?icv\\_alarm\\_monitor.htm](http://product.baosight.com/SupportCenter/doc/icv_doc_new/index.html?icv_alarm_monitor.htm)) 相兼容。

表达式说明：

1. 字符串，匹配完整字符串，例如，表达式为 alarm，则匹配值为 alarm 的选项。
2. ? 表达式，匹配任何单个字符，例如，表达式为 a?a，则匹配值为 aaa aba等选项。
3. 通配符 \*，例如表达式为 \*aaa\*，则匹配所有包含 aaa 的选项。
4. 正则表达式匹配，需要将表达式写在 @qreg() 的括号内。示例表达式为 @qreg(^[a-zA-Z]{1}([a-zA-Z0-9]|[.\_]){4,19}\$)，则匹配所有 5-20个以字母开头、可带数字、“\_”、“.”的字符串 的选项。

## 报警变量名称过滤设置



变量表达式

|

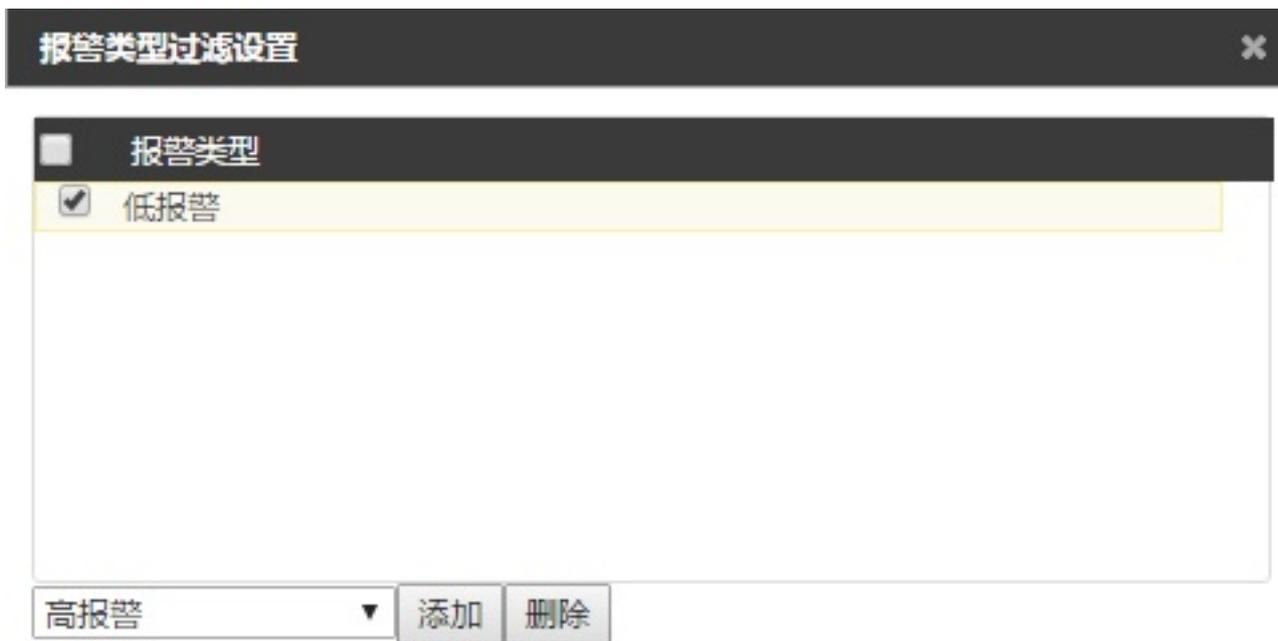
表达式说明：

- 字符串，匹配完整字符串，例如，表达式为“alarm”，则匹配值为alarm的选项。
- ?表达式，匹配任何单个字符，例如，表达式为“a?a”，则匹配值为aaa，aba等选项。
- 通配符\*，例如，表达式为“\*aaa\*”，则匹配所有包含aaa的选项。
- 正则表达式匹配，示例表达式为“@qtrereg(^[a-zA-Z]{1}([a-zA-Z0-9][.\_]){4,19}\$)”，则匹配所有“5-20个以字母开头、可带数字、“\_”、“.”的字符串”的选项

6. 报警类型过滤设置。用户可以从配置界面中选择报警类型，添加到过滤表格中，运行时符合表格中配置的类型报警条数将被筛选显示出来。删除已配置的报警类型只需勾选上表格中的项目，点击删除按钮。

支持的报警类型包括：

- 低低报警
- 低报警
- 高报警
- 高高报警
- 变化率报警
- 偏差报警
- 变位报警
- 0->1报警
- 1->0报警



7. 数据代理过滤：与 5) 报警名称过滤的配置相似。
  8. 报警点描述过滤：与 5) 报警名称过滤的配置相似。
  9. 子系统过滤：与 5) 报警名称过滤的配置相似。
- 列选项配置

配置列选项，报警列表可显示列选项，初始为**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**等 9 项，可根据自己项目的需求，进行勾选。

可以点击某一行的空白位置切换到编辑状态，可以输入该列数据的替代名称，并且调整该列显示的宽度。替代名称不影响该列关联的数据，只影响运行态报警表头显示的名称。编辑状态下可以通过点击“上移”“下移”按钮调整列的显示先后顺序。

变成替代名称编辑、顺序调整后，再次点击该行的空白位置退出编辑状态。**请注意**：退出编辑后配置才会被保存；列宽数值应为正数。

启用	列名	替代名称	列宽
<input checked="" type="checkbox"/>	数据代理		100
<input checked="" type="checkbox"/>	报警点		100
<input checked="" type="checkbox"/>	优先级		100
<input checked="" type="checkbox"/>	报警类型		100
<input checked="" type="checkbox"/>	报警值		100
<input checked="" type="checkbox"/>	发生时间		200
<input type="checkbox"/>	恢复时间		200

上移 下移

• 报警设置

可以选择是否在运行态启用声音报警。画面运行中可以变更声音报警开启、关闭的状态。

可为不同的优先级配置各自的报警选项。未确认未恢复、未确认已恢复、已确认未恢复三种状态的选项独立配置。其中三种状态均可以配置报警记录的前景色、背景色，以及闪烁颜色。闪烁颜色可以结合闪烁配置的下拉框，分为不闪烁、前景闪烁、背景闪烁三种方式，可以自行配置闪烁间隔（即颜色切换的间隔）时间，默认值为 0.5 秒。

对于未确认未恢复的报警，用户可以选择声音报警，并选择是否循环播放。配置界面提供试听功能，试听过程中再次点击该按钮将停止声音播放。CVS 系统为用户提供了两种默认的报警声音（分别是 crsdefaultsnd.mp3 和 defaultsnd.mp3），上传自定义的报警声音需要打开后台 cvs/data/lessees/<租户名>/audio 的文件夹，将声音文件放置在该文件夹中。支持的声音格式包括 mp3 和 wav，建议使用英文文件名。自定义的声音文件不能与默认报警声音重名。

需要注意的有以下几点：

1. 配置界面中，新增优先级将复制最后选中的那条优先级配置信息，包括描述、颜色、闪烁、声音配置。
2. 默认优先级可以编辑，但是无法删除。
3. 添加、修改界面只需要填写优先级和描述信息即可，优先级不能与已有配置项重复。
4. 可以为多个优先级配置报警声音，运行状态下，同一批次报警记录仅应用当时分页中配置了报警声音的、未确认未恢复状态下优先级最高的那条报警声音配置。
5. 可以添加、删除自定义的声音文件。默认声音无法删除。

基本设置 过滤配置 列选项 **报警设置** ✕

启用报警声音

优先级	描述
默认	默认设置

未确认未恢复
未确认已恢复
已确认未恢复

前景色	背景色	闪烁色
<span style="color: red;">■</span> ▼	<span style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></span> ▼	<span style="background-color: gray; display: inline-block; width: 15px; height: 15px;"></span> ▼

闪烁配置 不闪烁 ▼ 闪烁间隔 0.5 (单位: 秒)

报警声音 ▼

▶ 试听
+
-

循环播放报警声音

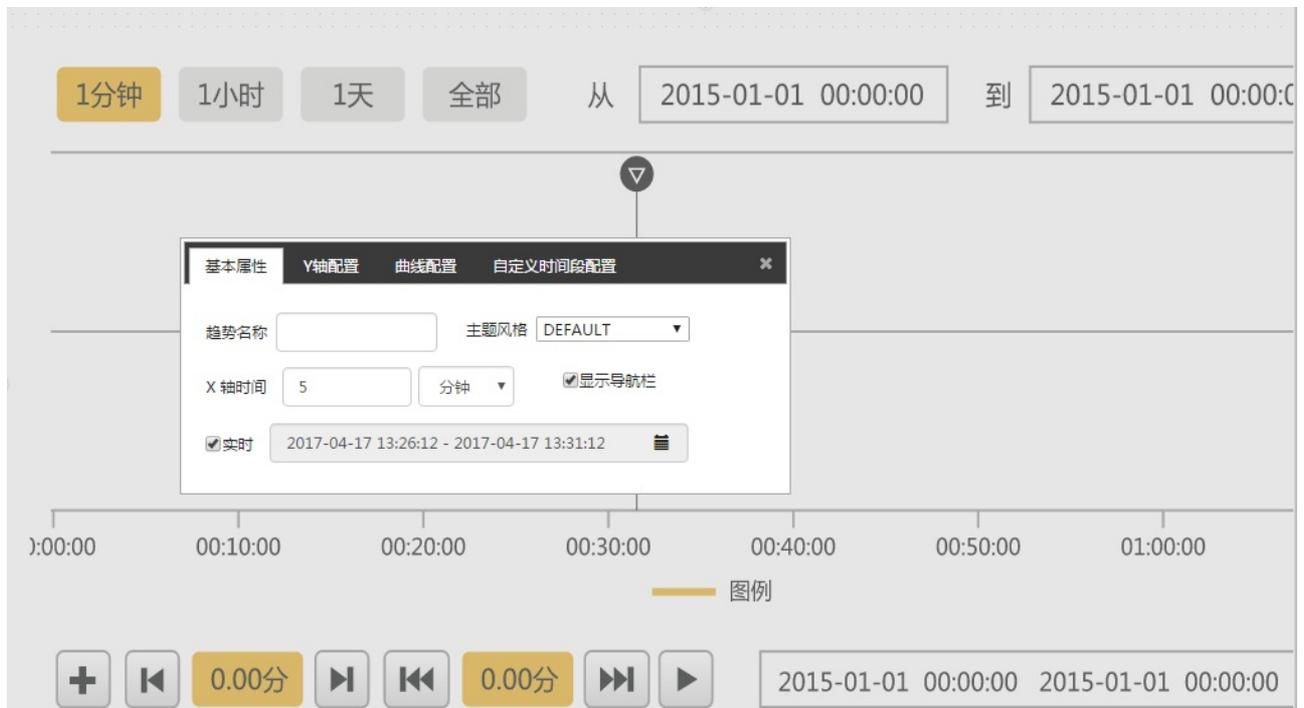
+ 添加
- 删除
✎ 修改

#### 4.4.3 趋势控件

在画面设计器左侧“高级控件”栏，双击或选中趋势控件拖动到画面设计器中，进行编辑。

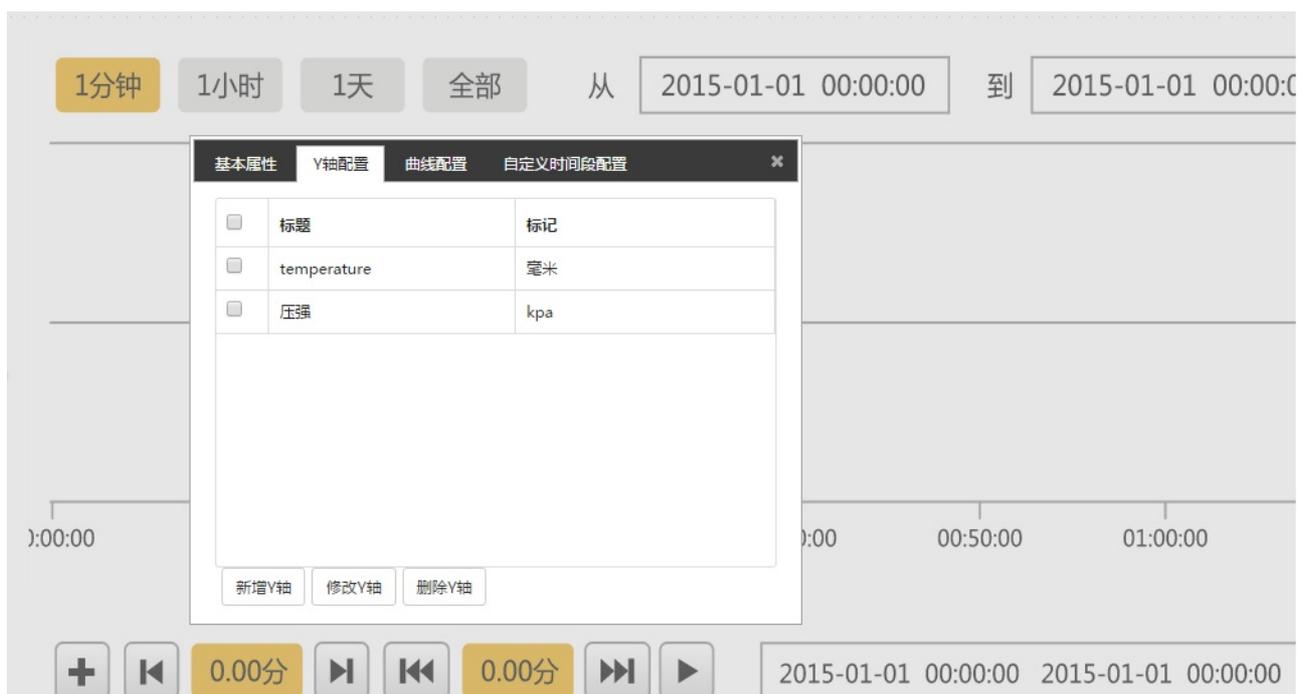
- 基本属性

1. 可配置趋势名称
2. 可配置运行态是否显示导航栏
3. 可配置趋势图的主题风格，默认是四种风格
4. 可配置是否是实时趋势，选择实时趋势时，不能配置时间
5. 不开启实时趋势时，可配置趋势发生时间区间
6. X轴时间范围（仅对实时状态时有效）

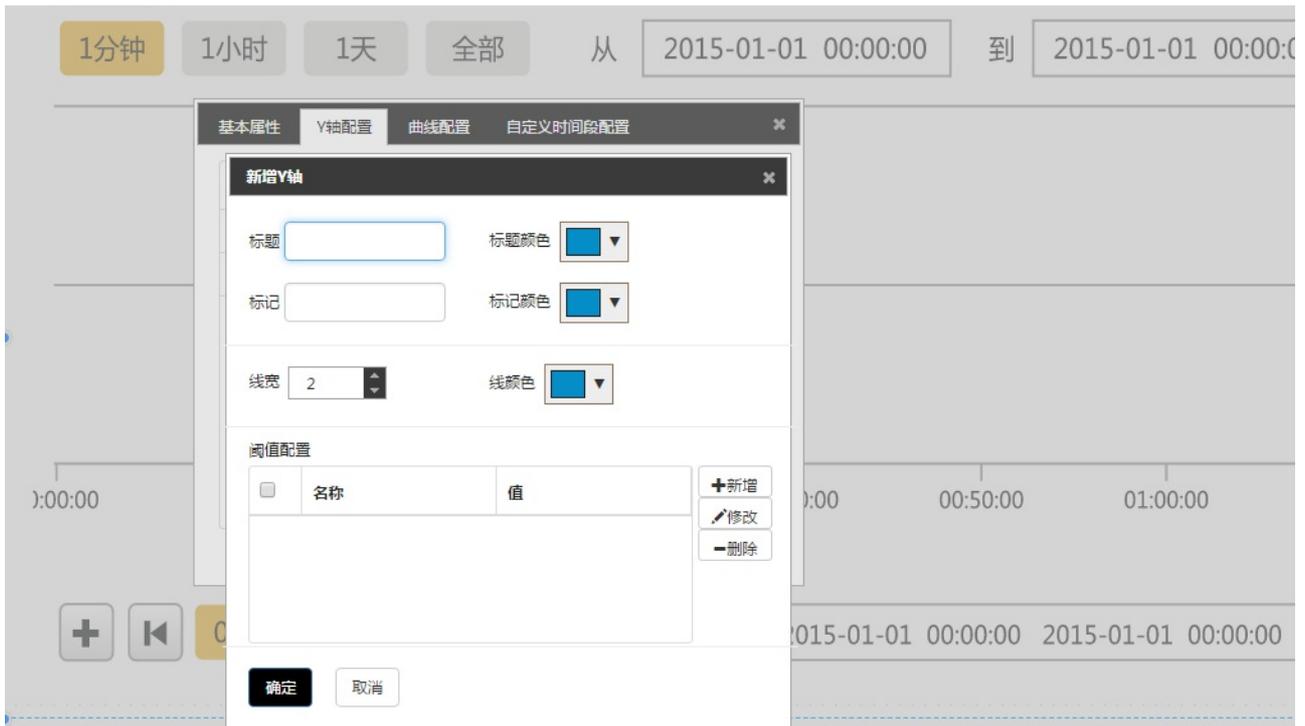


- Y轴配置

趋势2.3版本以后，支持多个Y轴显示，不同的曲线可以关联到不同的Y轴上，可以对Y轴的标题，标记，线的宽度，颜色进行配置。默认带一条默认Y轴“default”。不同的Y轴也可以配置不同的阈值线，之前的阈值线不在是高级配置，作为Y轴的一个属性进行配置。



新增Y轴，输入Y轴配置信息,Y轴名称不可重复



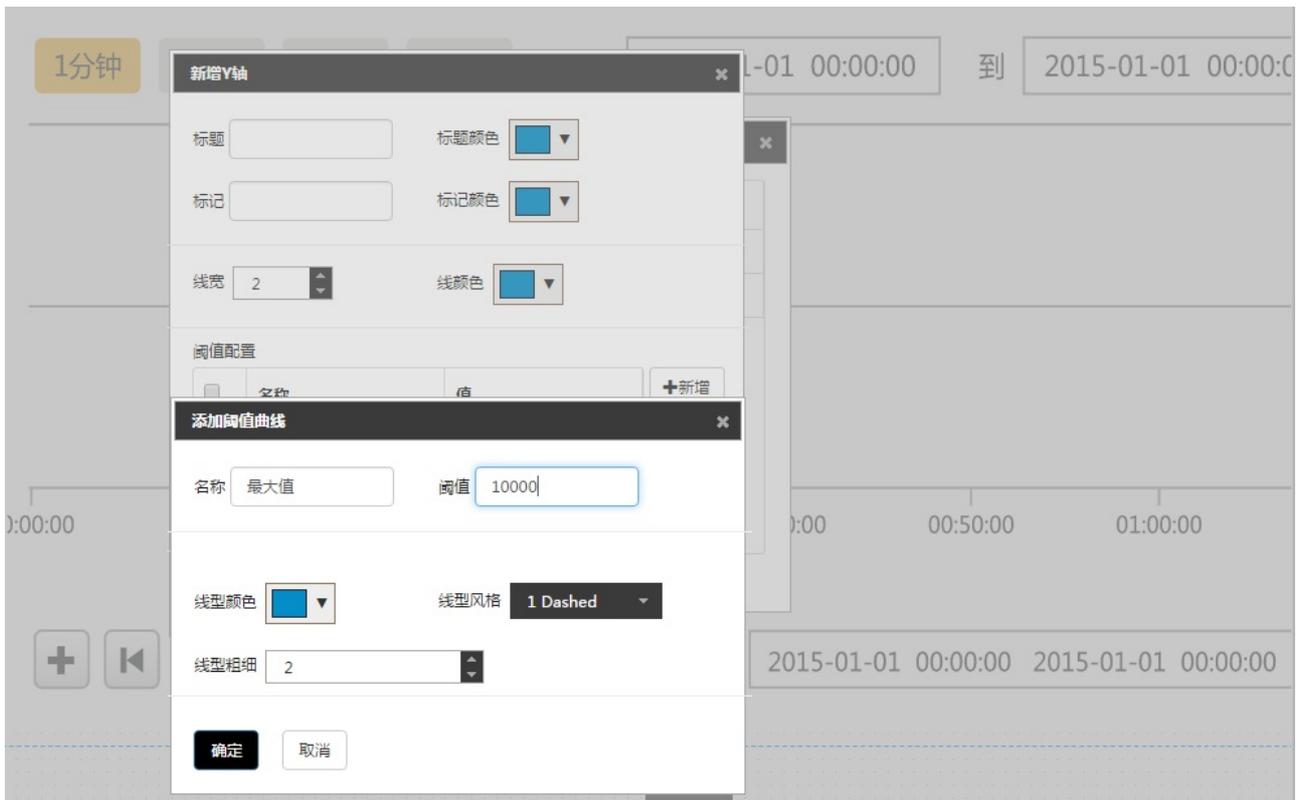
修改Y轴 点击修改Y轴按钮，可以修改配置的Y轴属性，Y轴名称不可修改

删除Y轴 选中需要删除的Y轴，点击删除按钮，该Y轴被删除

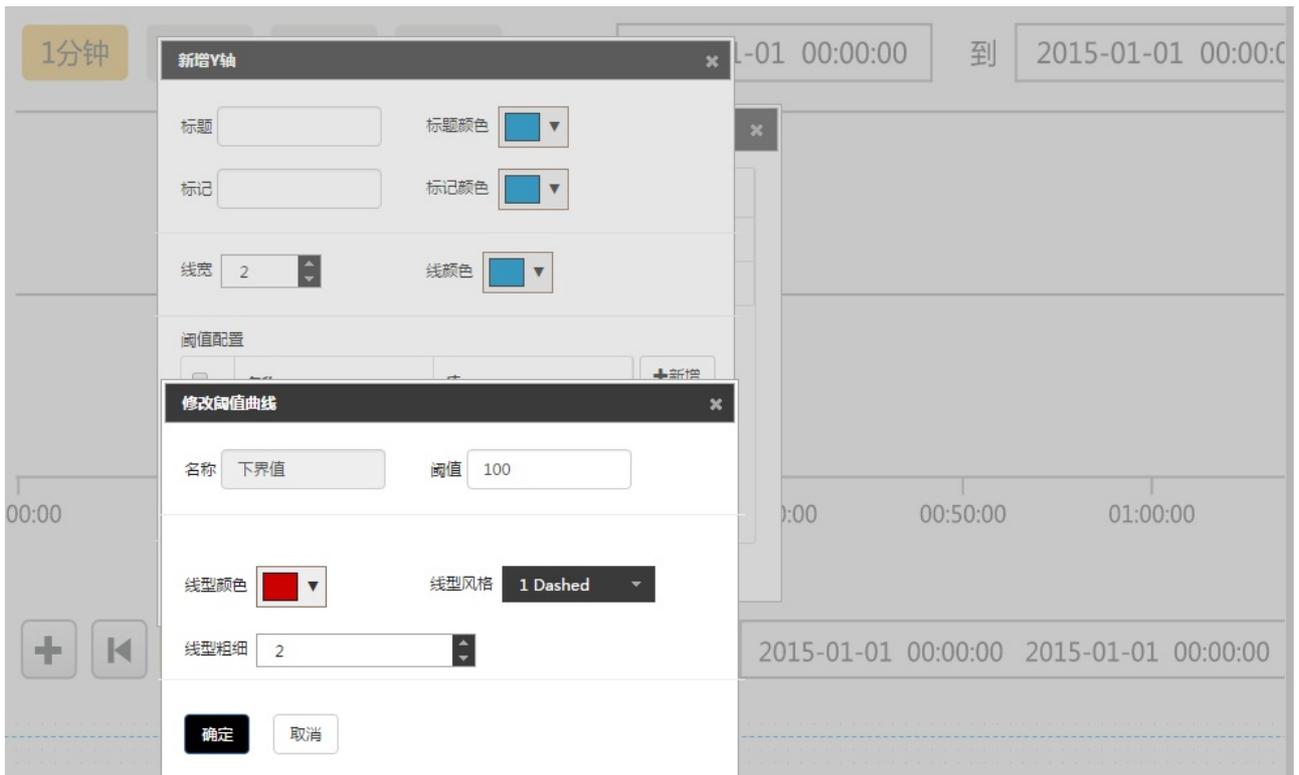
#### 阈值线配置

用户可以根据Y轴的范围设置一些阈值线，来查看阈值线附近的数据变化，提高数据分析的效率。首先要在趋势图编辑态配置阈值曲线。

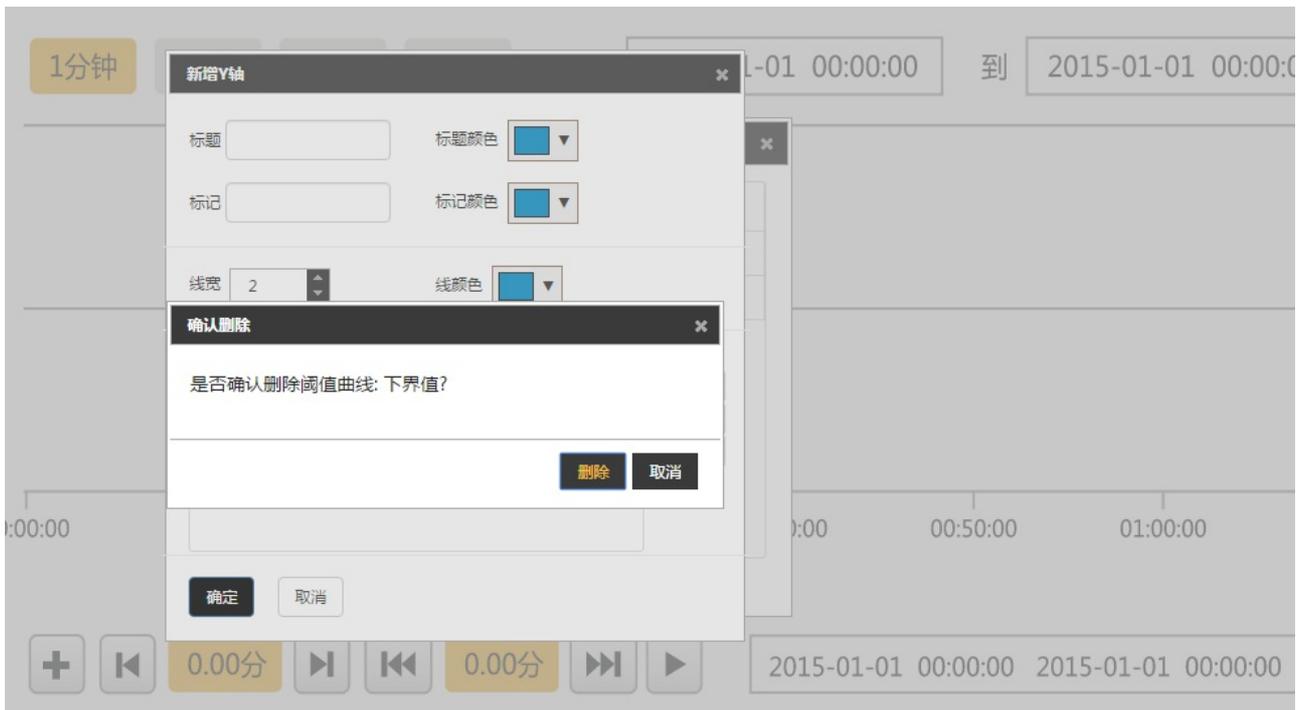
添加阈值线配置；



修改阈值线配置；

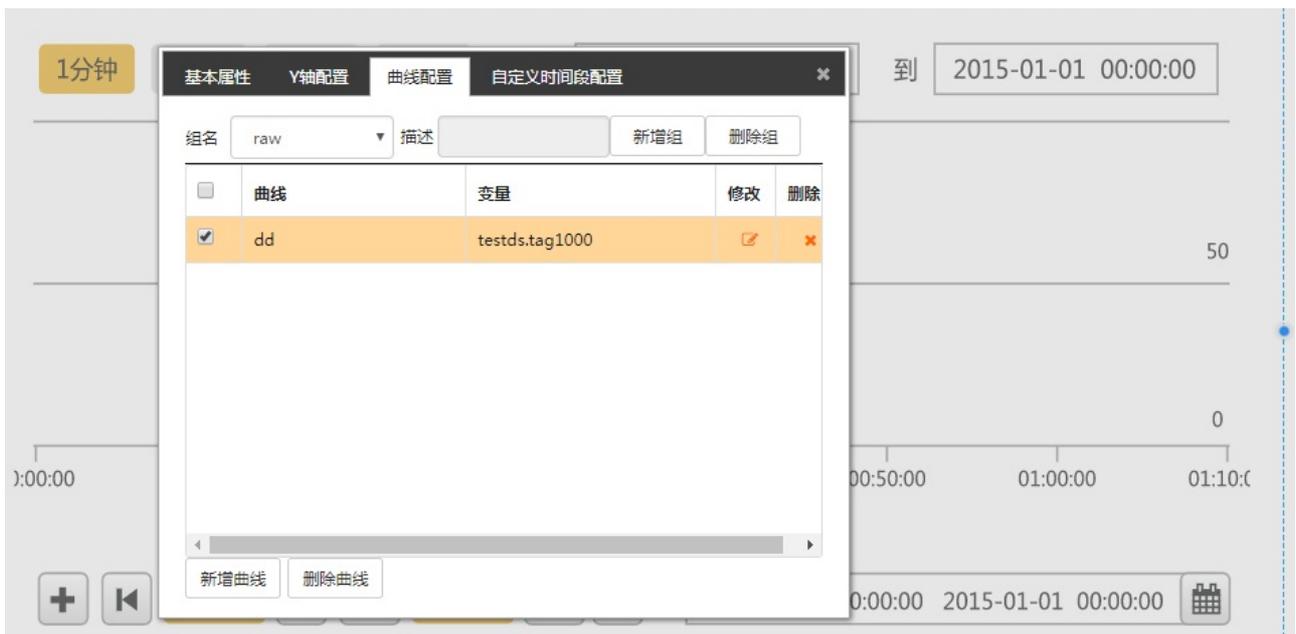


删除阈值曲线；

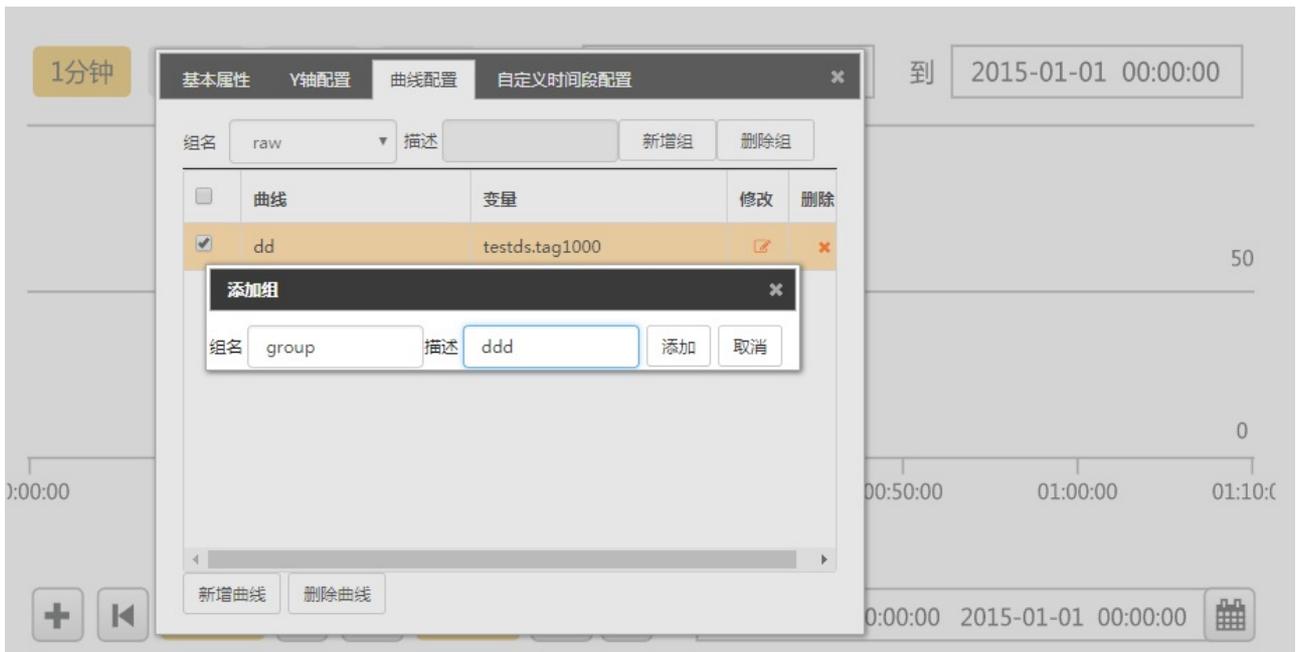


- 曲线配置

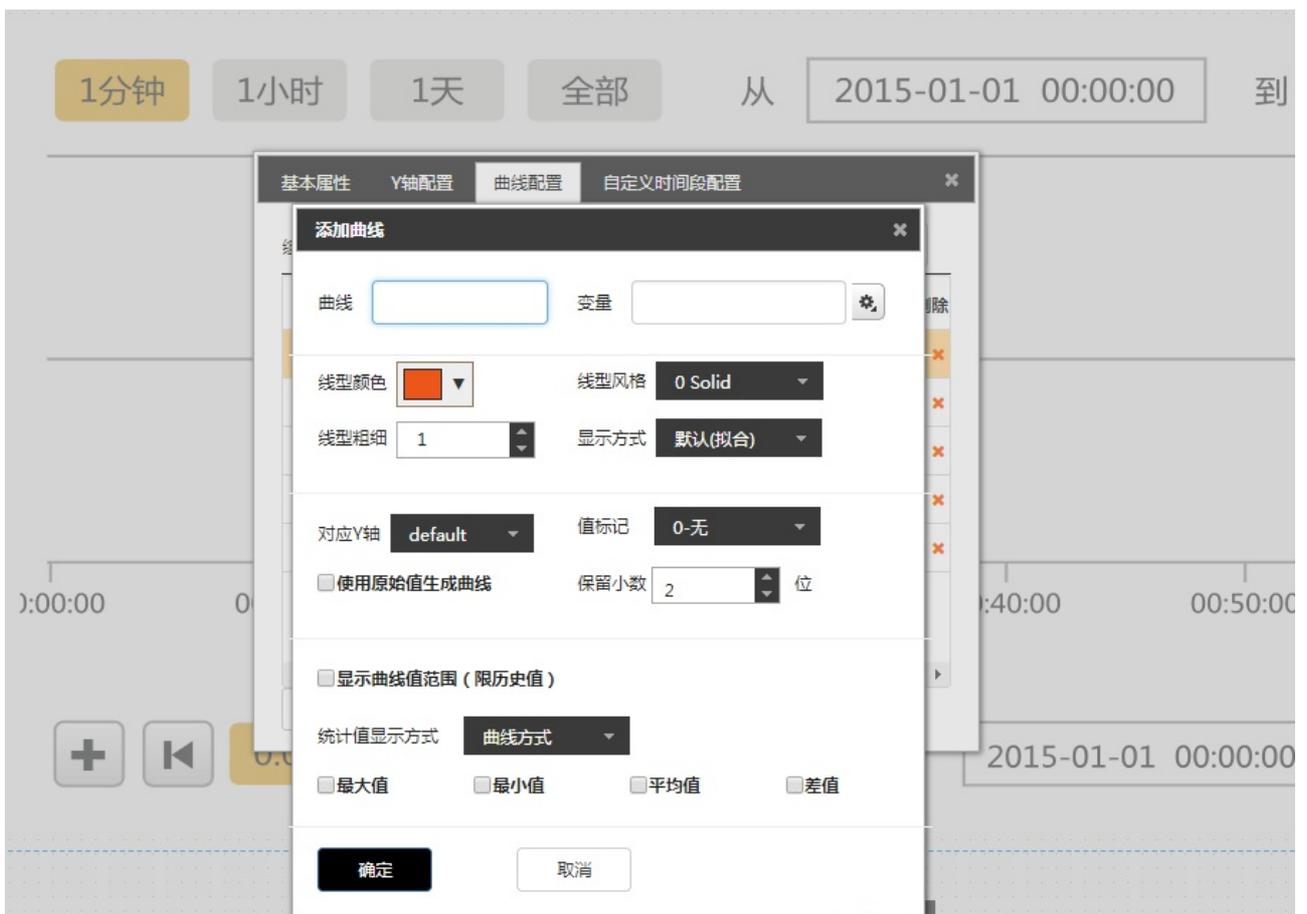
可添加不同的组，并且在相应的组下添加曲线；



新增组，输入组名和描述



新增曲线(必须在一个已有的组下添加曲线)，后面齿轮状按钮可配置曲线所关联的点，具体选点配置请参照选点；



说明

1. 保留位数：主要是数值型数据，设置数值的小数点后保留位数，如果是整型数据5，设置为2位，就是5.00，如果是浮点型数据4.5673，设置为2位就是4.57。位数只能设置为[0-6]，默认为2位。
2. 线型颜色：通过点击颜色按钮，配置想要的线型颜色。
3. 线型风格：支持五种线型风格配置，通过下拉框选择。
4. 线型粗细：支持1~100之间的线型粗细配置。
5. 最大值，最小值，平均值，差值是统计曲线，勾选上以后，当趋势运行起来，根据原始值计算出对应的统计值，并画出统计曲线，统计曲线风格和原始值曲线风格一致。差值是最大值减去最小值所得的值。
6. 趋势点变量暂时不支持表达式类型，只能查看原点数据。
7. 显示方式表示用户曲线是拟合方式还是阶跃方式显示
8. 对应Y轴：配置此条曲线和那条Y轴对应，如果不配置，就是默认Y轴，如果该条曲线配置的Y中被删除，也自动使用默认Y轴
9. 值标记：趋势曲线是连续的没法看出某一个点上是否有数据，启用值标记功能，就可以显式的看出，那个点上有值
10. 使用原始值生成曲线：默认使用插值方式获取数据显示，如果用户配置了使用原始值，则获取原始数据画出曲线，目前只有ihd数据源支持原始数据。
11. 统计值显示方式：默认是曲线方式，用户勾选最大值等之后，趋势图会画一条最大值曲线，如果用户选择文本提示方式，则为用户鼠标放到趋势图一个具体的点上显示该点的最大值。
12. 显示曲线值范围：提供在插值情况下查看趋势时，当前点的大致范围，最大值和最小值，这个属性只有在查看历史趋势的时候有效

修改曲线 点击该条曲线所在行的修改按钮，以修改配置的曲线属性，曲线名称不能修改，一次只能修改一条曲线

删除曲线 选中需要删除的曲线，点击删除按钮，该曲线被删除，也可以通过勾选曲线前面勾选框，选择多条曲线一起删除

注：2.4版本以后，添加曲线可以批量添加，用户在曲线点击新增曲线，然后再点的选择界面可以选择多个tag点，然后点击确定，这些点都被加到曲线配置中，默认曲线名称为点名+下划线+配置的曲线名，例如：曲线名为line，点名为testds.tag100，则新增的曲线名称为testds.tag100\_line。总的曲线名长度限制为1024，配置的曲线名称最长为64。

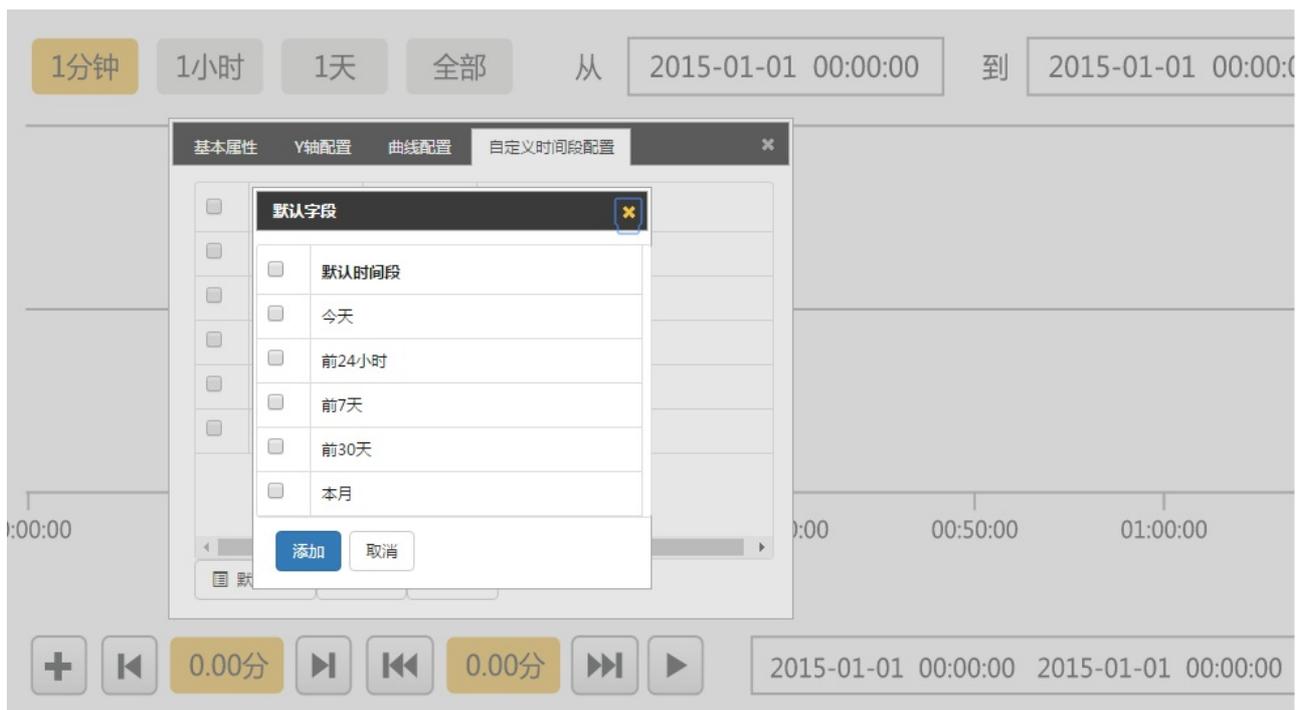
最后用户需要通过勾选曲线前面的checkbox，来配置曲线显示，如果不勾选，则默认不显示曲线。

- 自定义时间段配置

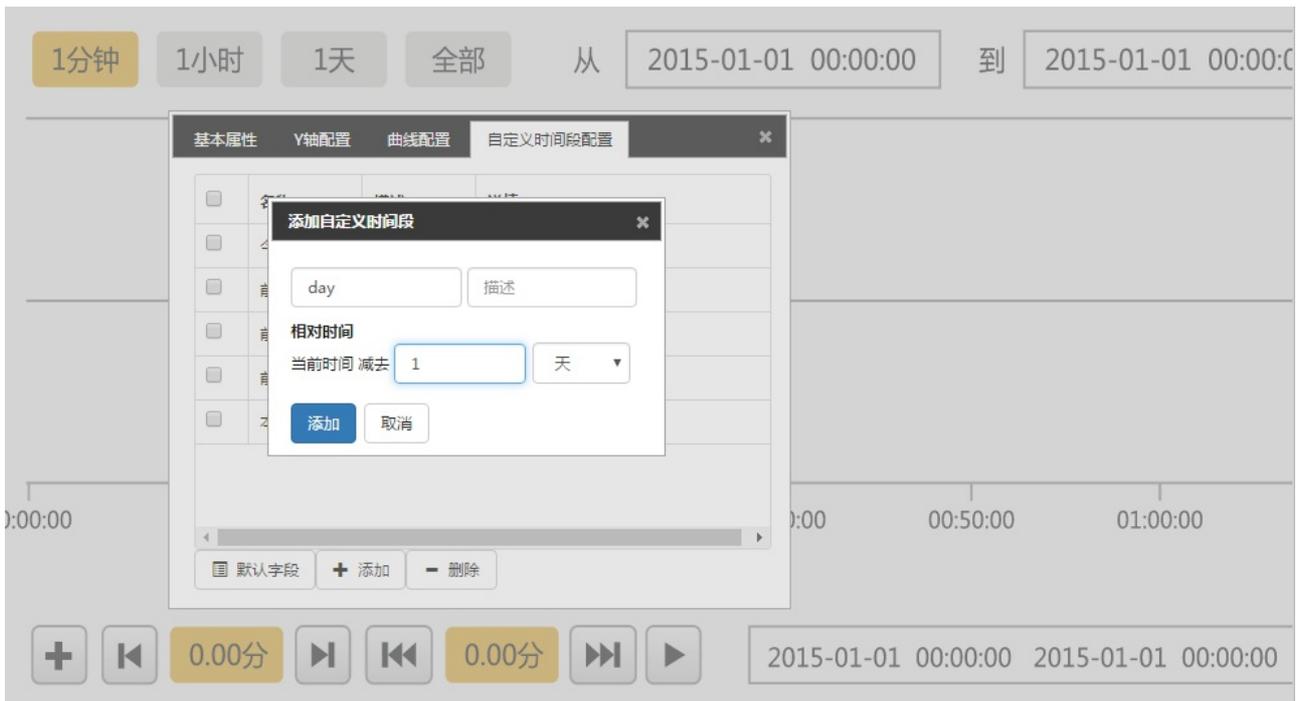
配置用户常用时间段，在趋势运行时，用户可以迅速切换到常用时间段的趋势图，进行查看。



可添加默认时间段；



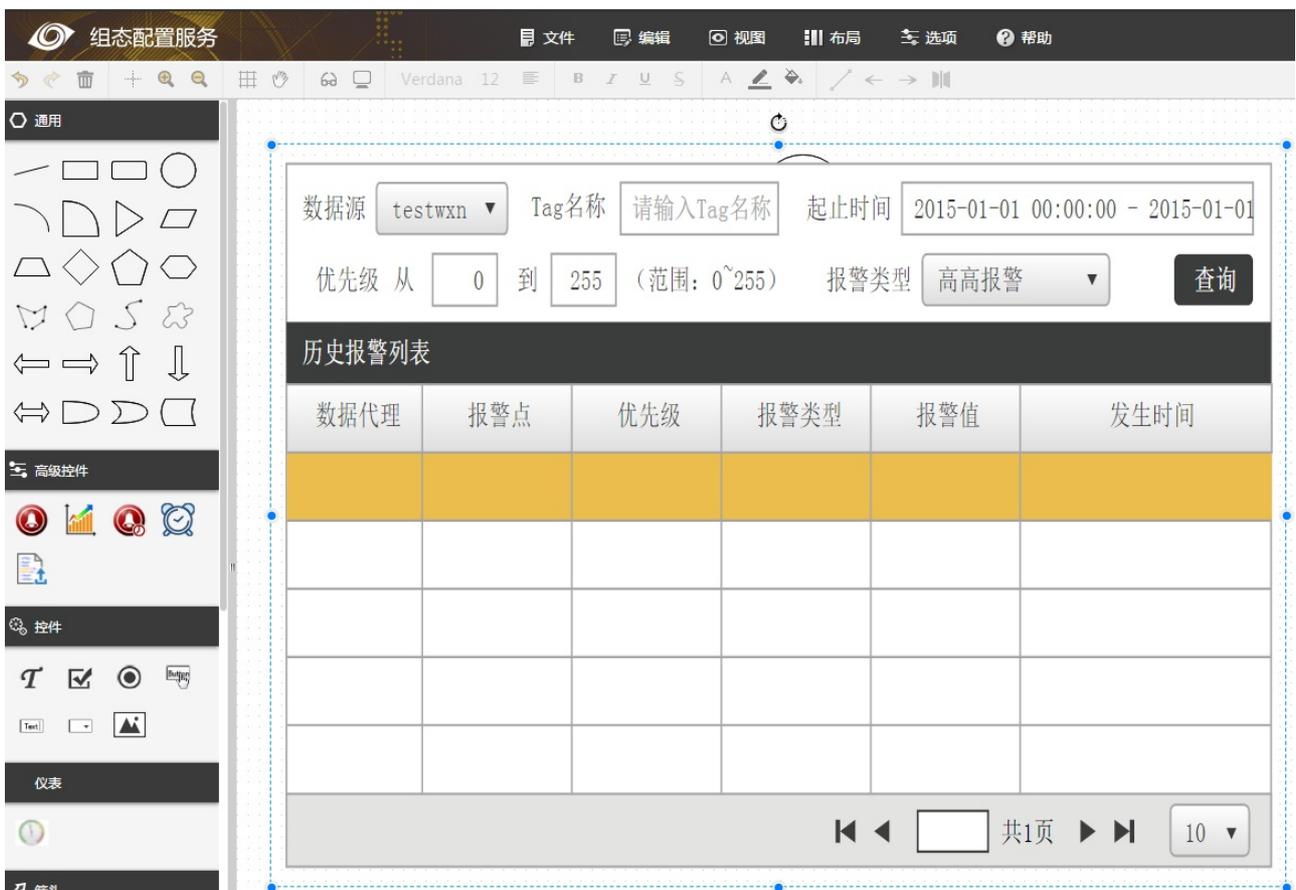
添加自定义时间段，是指趋势运行态的当前时间减去某一个时间，几天，几小时，几分钟或者几秒；



删除配置的自定义时间段；

#### 4.4.4 历史报警控件

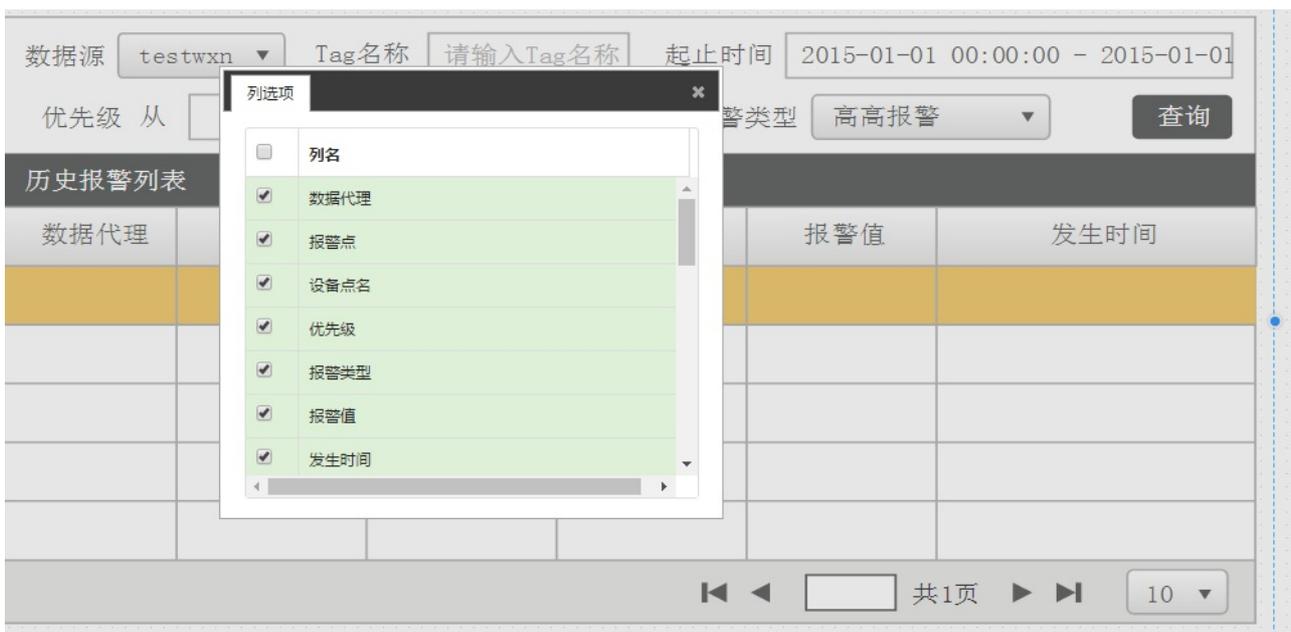
在画面设计器左侧“高级控件”栏，双击或选中历史报警控件拖动到画面设计器中，进行编辑。



在历史报警控件上右击，弹出右键菜单，选中历史报警配置进去历史报警配置界面，在这里用户可以配置在运行态需要显示的报警列。

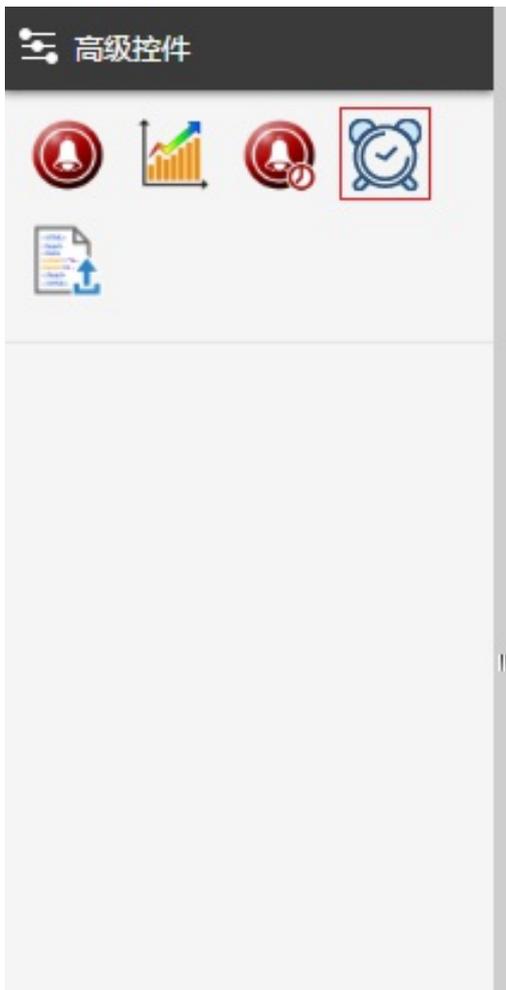


- 列选项配置  
配置列选项，报警列表可显示列选项，初始为**数据代理、报警点、优先级、报警类型、报警值、发生时间、报警状态、确认状态、节点名称**等 9 项，可根据自己项目的需求，进行勾选。



#### 4.4.5 定时器控件

定时器控件可以用于在画面中设定一个或者多个任务，每个任务可以设定自己的执行周期。



在定时器中，**任务**一词指的是定周期的执行一个 JavaScript 函数，执行周期的单位是毫秒。任务函数在调用的时候不会传递参数。

可以从边栏中拖动定时器控件到画面中，可以放置在任意位置，运行态报警控件不现实。

右击画面中的报警控件，点击“定时器配置”菜单项，弹出定时器控件的配置窗口。



点击“添加”按钮会在任务列表中新增一条记录，点击一条记录进入选中状态，可以进行删除、编辑等操作。删除后，该行记录从任务列表中被移除。

点击“编辑”按钮后，该行记录中函数名称、执行周期分别出现文本编辑框，同时“保存”“撤销”两个按钮进入可点击状态。“保存”“撤销”两个按钮只对该行起作用。

示例配置图中的两个函数 foo bar 是写在脚本编辑器里的函数，代码如下所示。用户可以配置任意函数名称（不包含括号、参数）。一条任务配置错误不会影响其他任务。

如示例函数所示，如果需要调用**系统函数**或者其他库函数需要传递参数时，可以在任务函数中指定。

```
function foo() {  
    console.log("foo");  
}  
  
function bar() {  
    console.log("bar");  
}
```

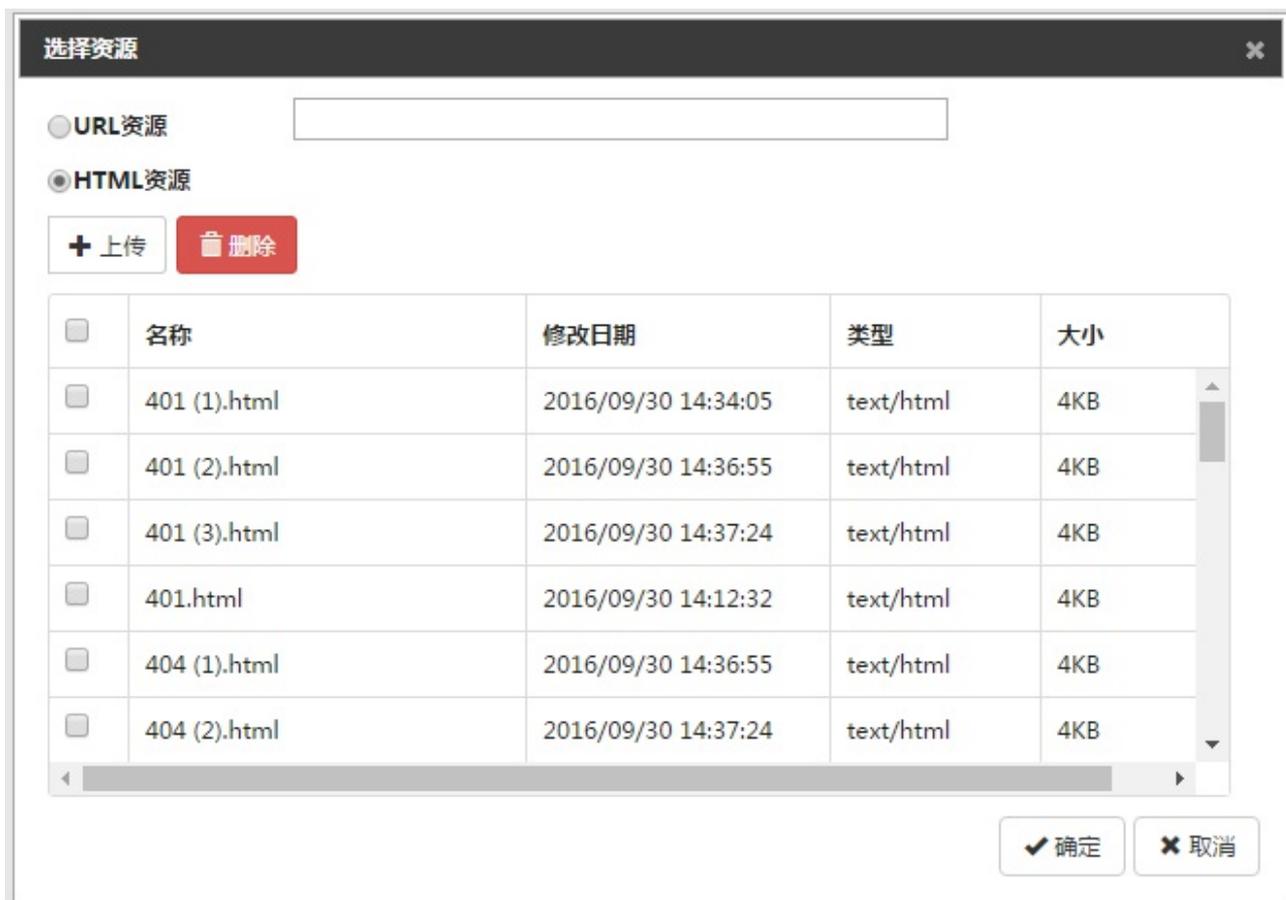
#### 4.4.6 html文件上传控件

html文件上传控件给用户提供一个上传自定义html文件的途径，通过选择用户自定义的html文件，在运行态显示出来，极大的增强的产品的多元应用，用户可以不用局限于已有的一些控件，可以自定义一些内容，显示出来，适用性更强。

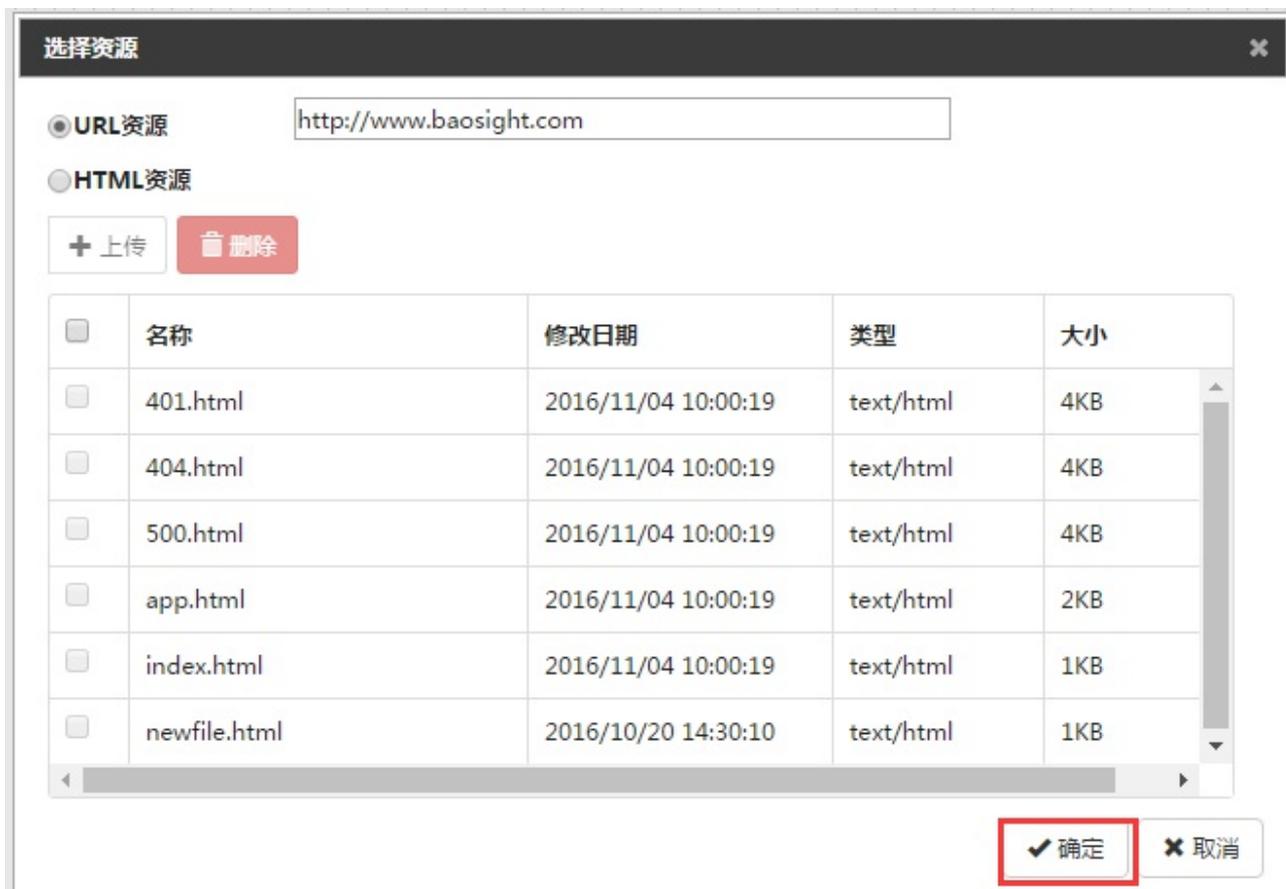
首先在画面设计器的高级控件中选择html文件上传控件



拖到画布上，弹出资源配置界面

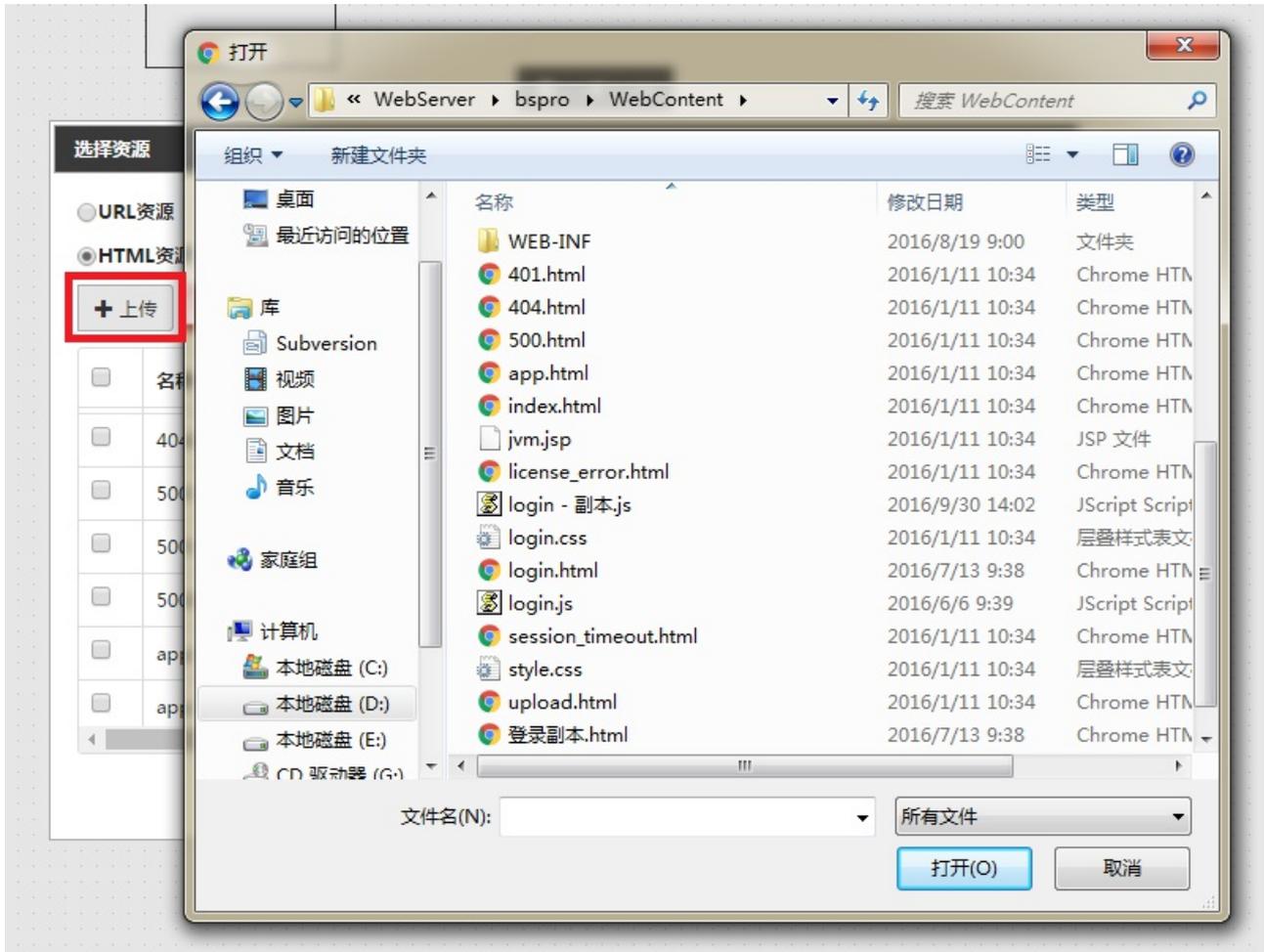


默认是选择html资源，用户点击URL资源按钮，输入可用的URL资源，如图：



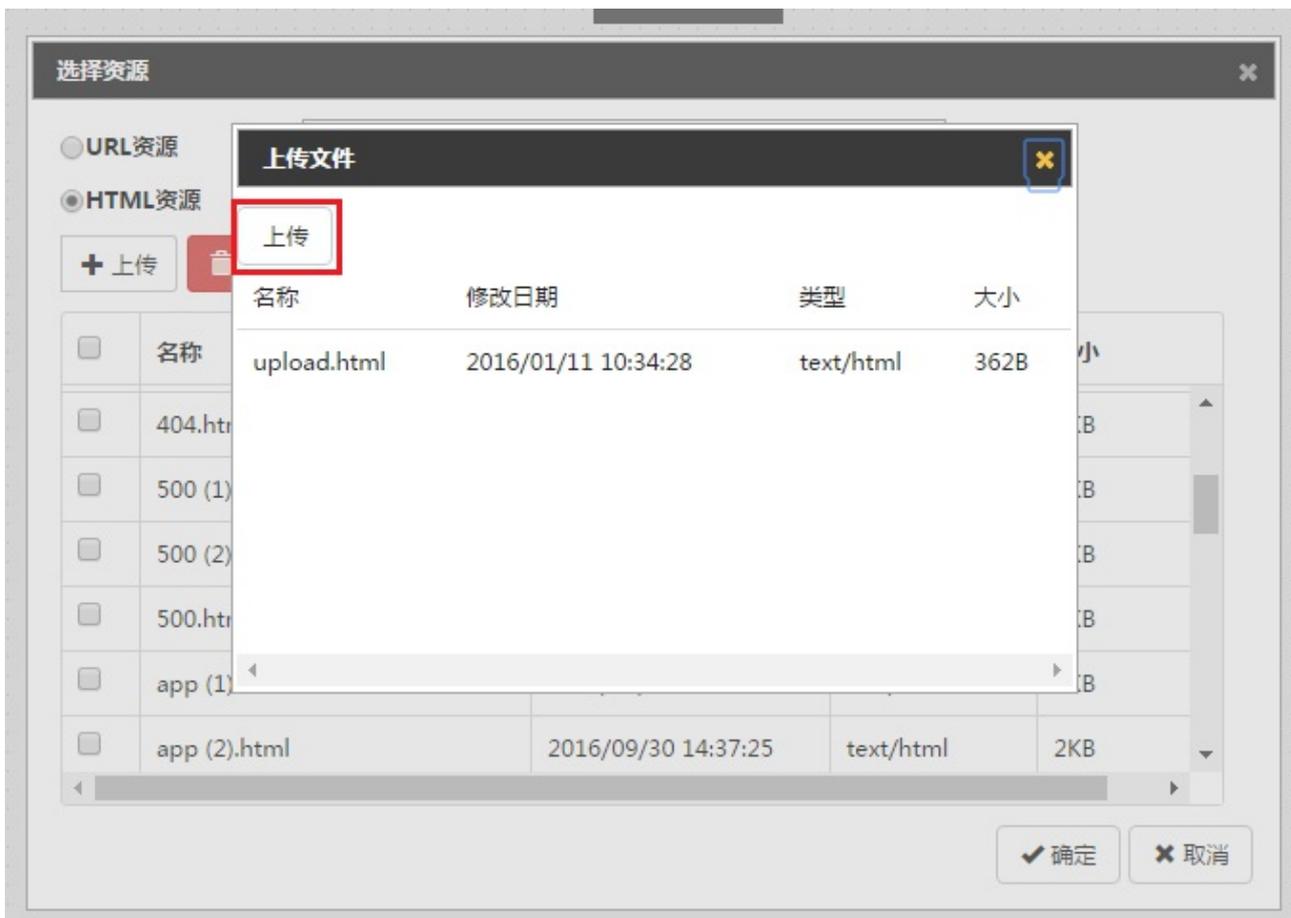
点击确定按钮。该配置被保存到html文件上传控件中。

用户点击HTML资源按钮，点击上传按钮，弹出文件选择对话框：



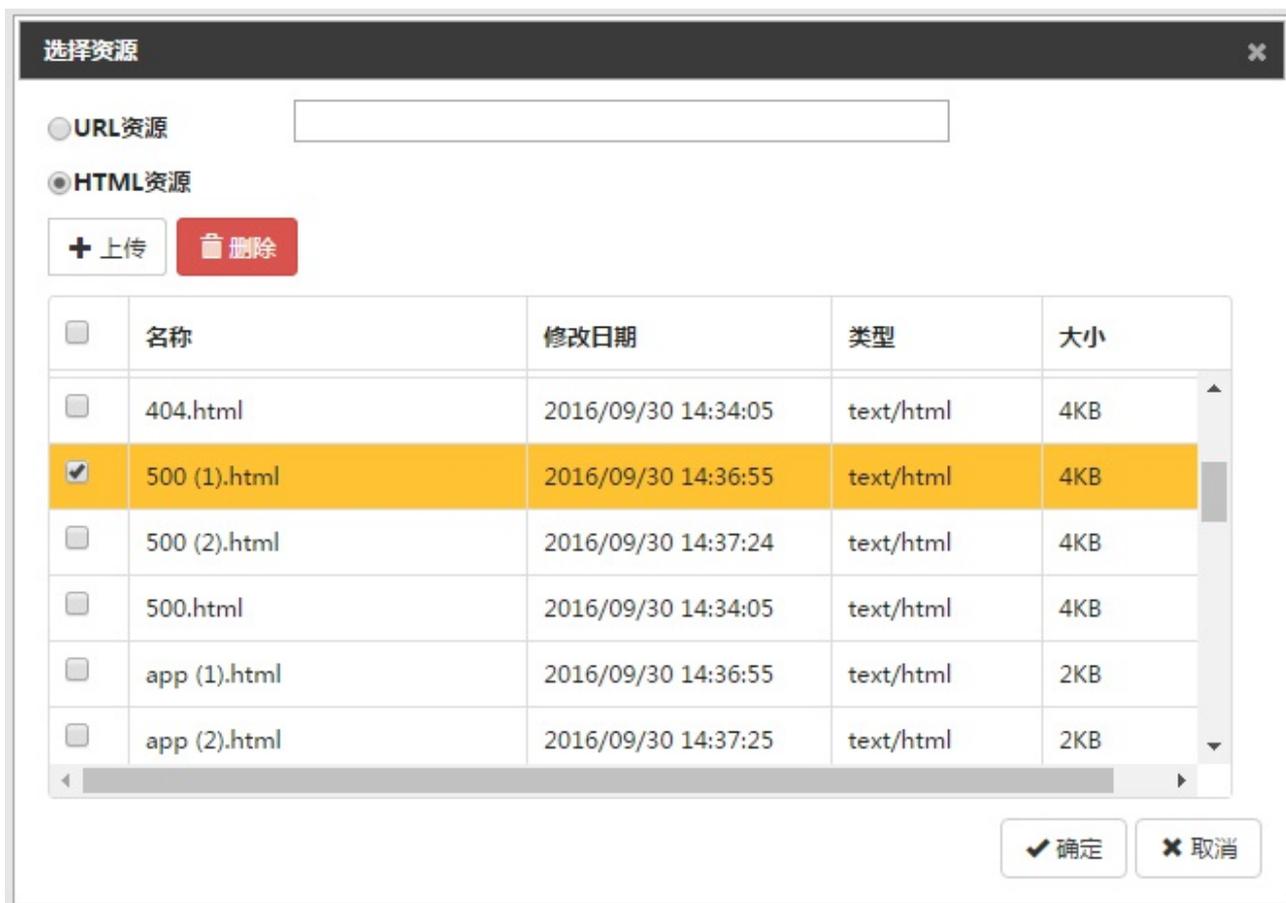
用户可以选择文件后缀为html, js和CSS，否则报错

选择好文件以后，点击确定，弹出确认上传对话框：



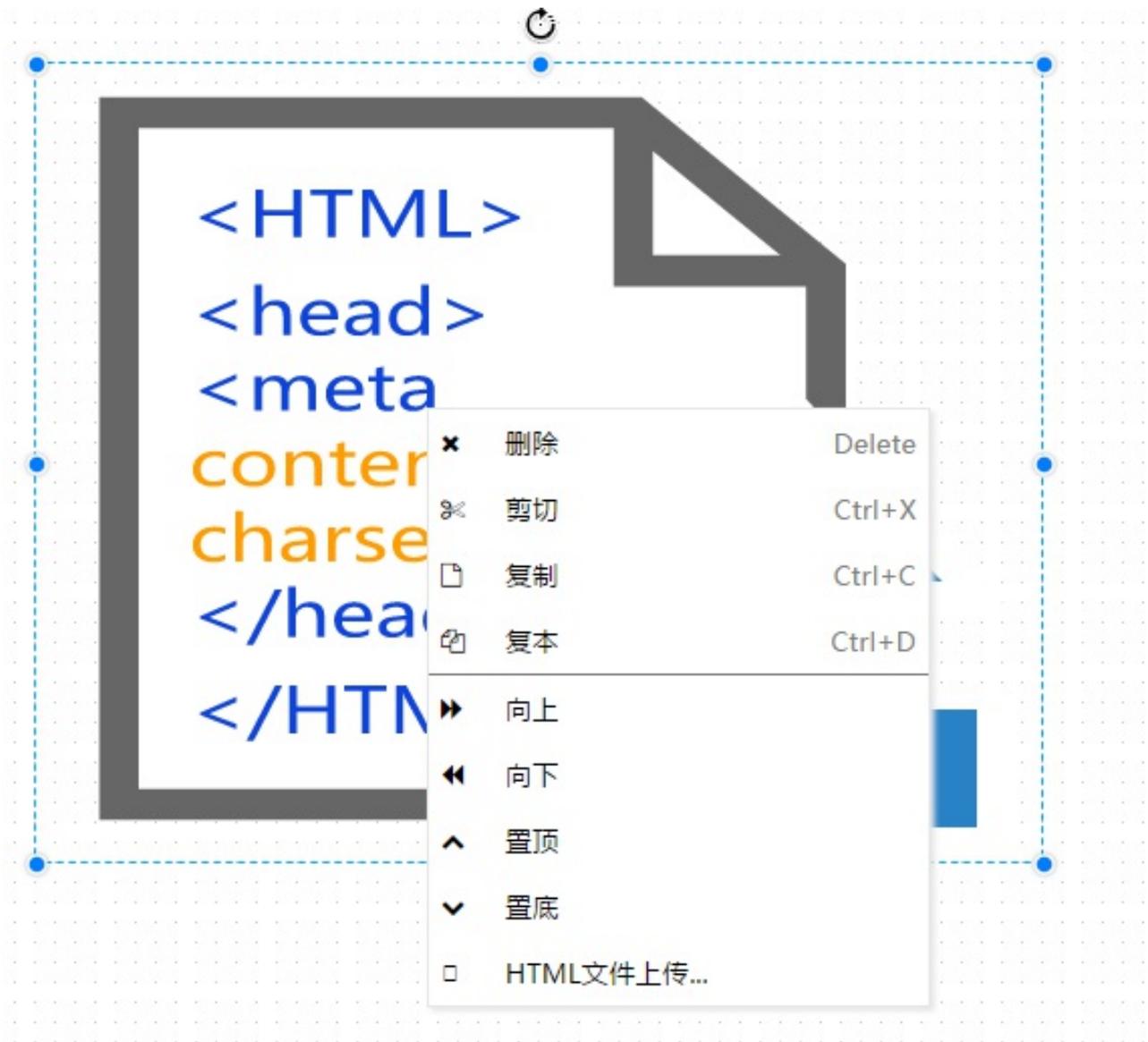
点击上传按钮，选择的文件被上传到服务器上。

上传后文件以后，选择一个要显示的html文件（注意，这里只支持html文件的选择，如果同时选择了js和css文件，将会报错）点击确定



配置的html文件被保存在html文件上传控件中。

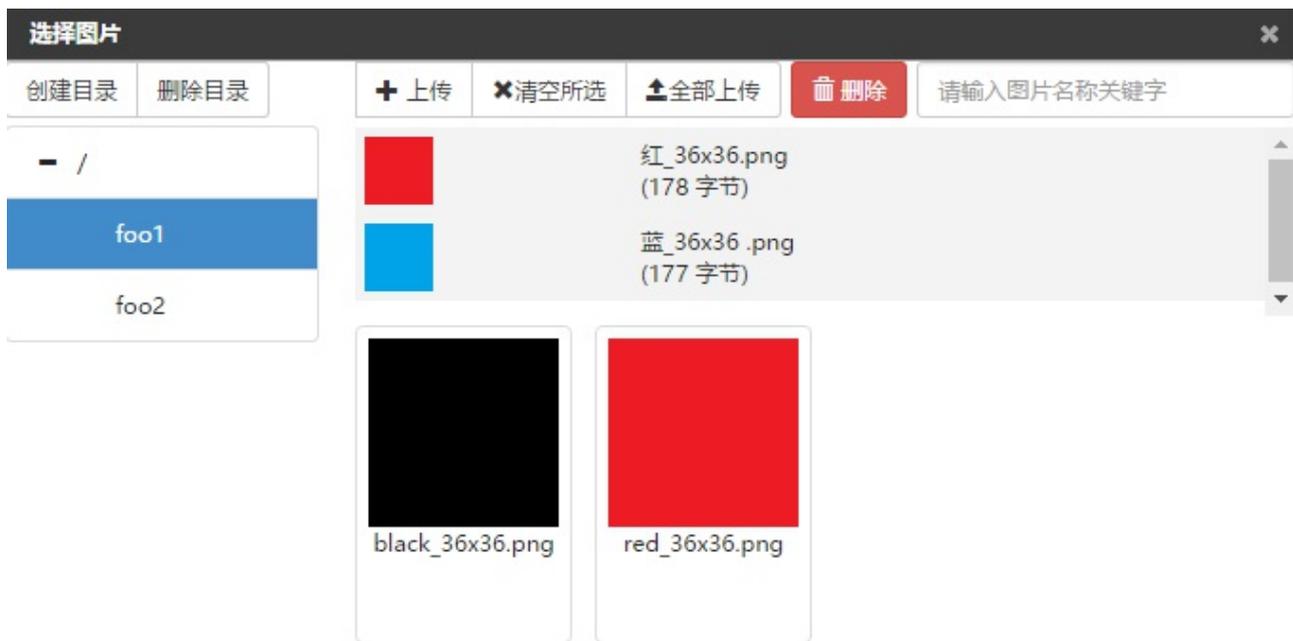
同时该控件还提供了修改配置的功能，在画面的html文件上传控件上右击



点击按钮，弹出修改对话框，操作同第一次配置一样。

#### 4.4.7 图片

将图片控件拖拽到画面上，弹出图片选择框

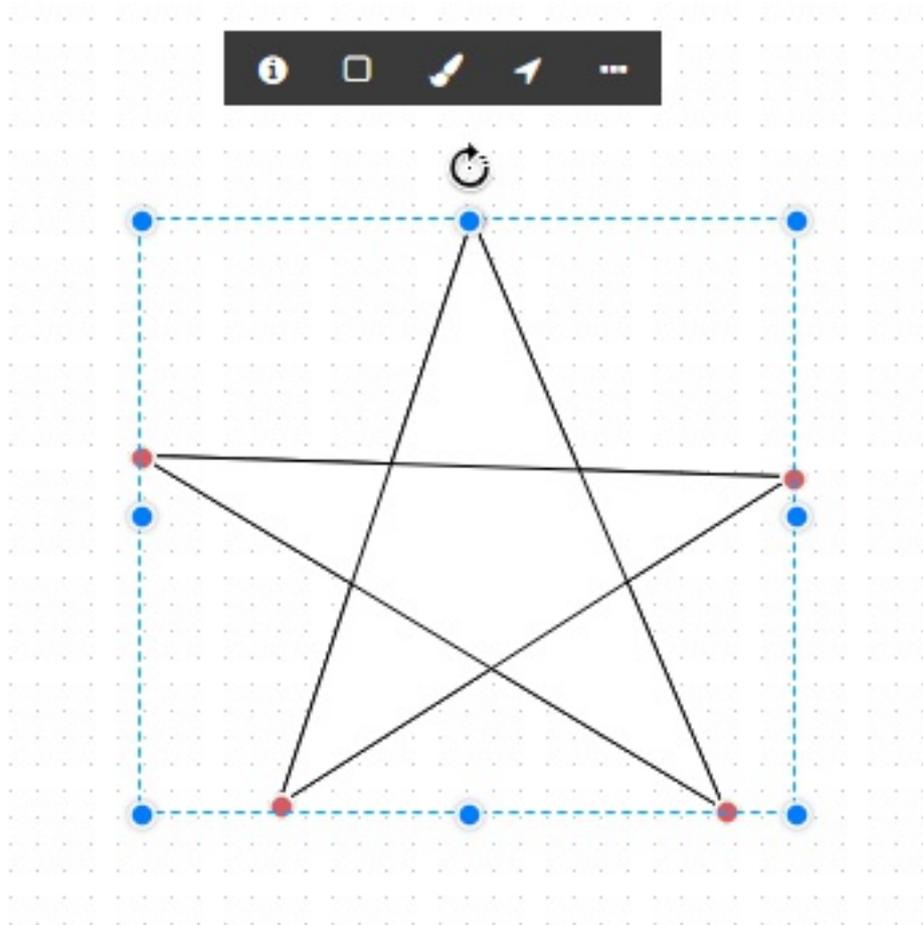


尺寸：大小：修改日期：✓ 选中 ✕ 取消

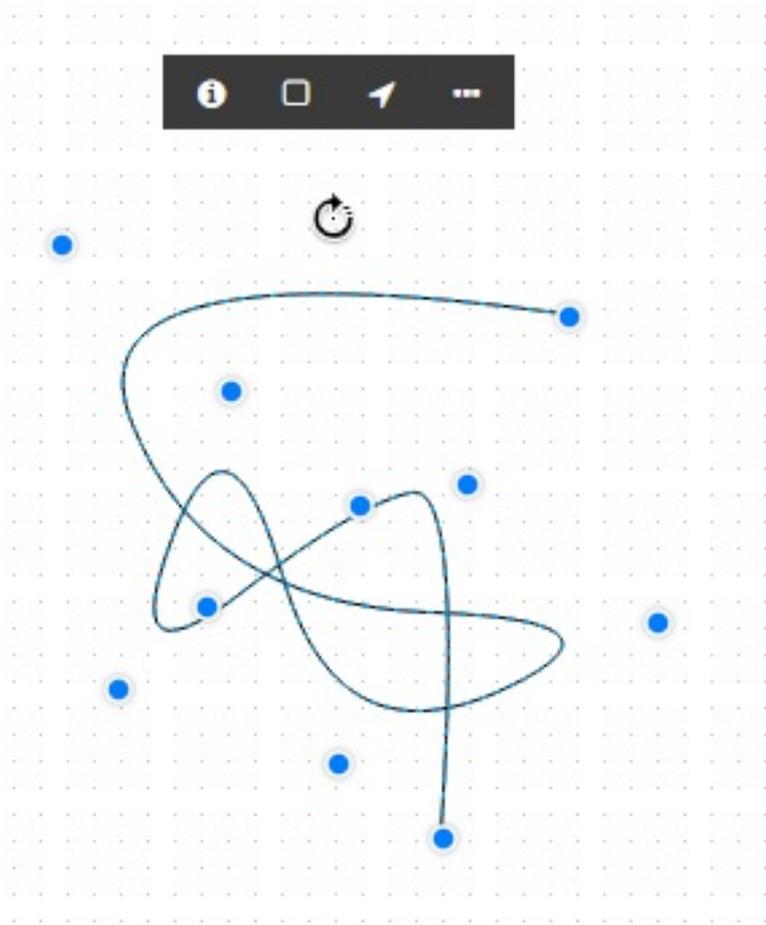
- 点击上传，弹出选择框，选择要上传的一个或者多个图片上传到服务器。选择图片后，预览缩略图将显示在菜单栏的下方。点击清空所选，则待上传图片将被清空。一次最多可以上传 20 张图片。
- 选中服务器预览区域中的一个或者多个图片，点击删除可以从**服务器**中删除对应的图片。多选图片将同时删除。
- 选中服务器预览区域中的一个或者多个图片，点击选中，该图片会添加到画面中。多选图片将依次堆叠显示。
- 在文本框中输入关键字，将会从当前图片列表中过滤出文件名包含此关键字的所有图片。
- 点击创建目录，在弹出的对话框中输入目录名称，点击确定按钮，将创建用户图片子目录。
- 点击某个目录，对于上传、删除、过滤图片的操作将在该目录下进行。
- 点击某个目录，点击删除目录，在确认窗口中点击确认按钮，将删除该目录及该目录下的所有图片。目录名称为“/”的表示根目录，不可被删除。

#### 4.4.8 折线/多边形/曲线/闭合曲线

折线等图形点击后，鼠标会变成 + 的样式，左键点击开始，在画布上拖动，再次点击左键为折点，双击左键结束  
 对于多边形和闭合曲线，绘制完成后，各个顶点为红色，用户可以继续拖动顶点来调整元素的形状；拖动边框的大小可以按比例缩放整个图形。

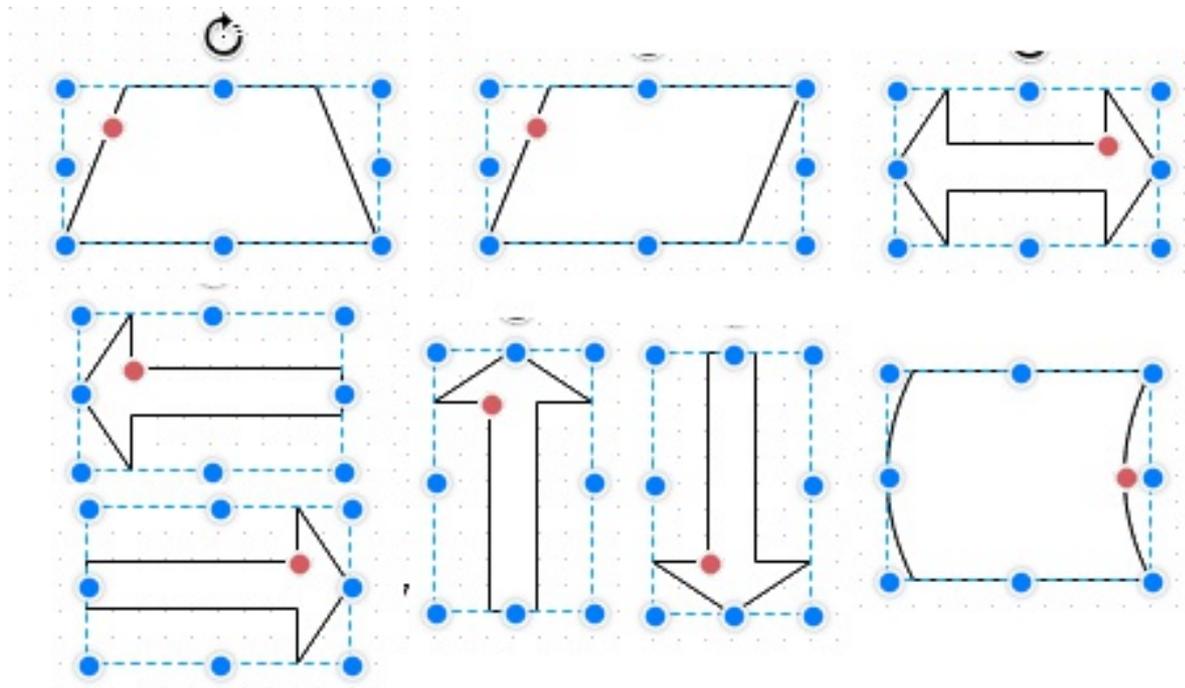


对于折线和曲线，顶点为蓝色，用户可以直接拖动顶点改变形状。



#### 4.4.9 梯形/平行四边形/数据存储图形/上下左右箭头/双箭头

这些图形拖动到画布之后，会有红点标记。拖动红点可以按比例修改图形。



#### 4.4.10 弧形

将弧形拖动到画布上，会显示默认的弧形。点击弧形属性页按钮 ，可以配置弧形属性。



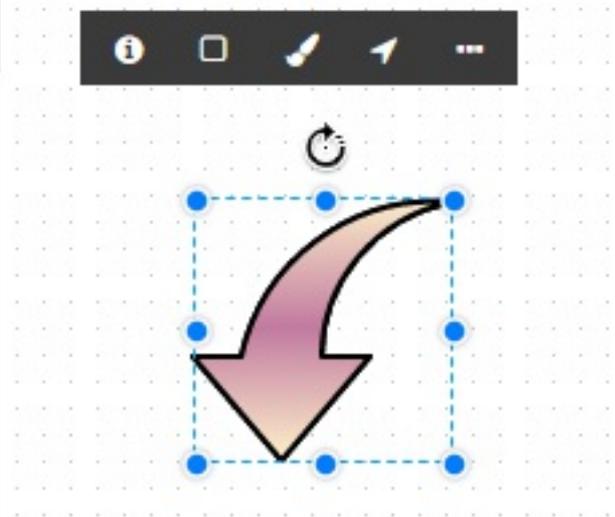
- X轴半径：要绘制弧形的x轴半径
- Y轴半径：要绘制弧形的y轴半径
- X轴旋转角度：弧形沿x轴旋转角度
- 弧线角度：弧线是大于180度还是小于180度，0表示小于180度，1表示大于180度
- 弧线方向：弧形的绘制方向。0表示从起点到终点沿逆时针方向，1表示沿顺时针方向

在位置属性页中，可以修改弧线的起点和终点位置。



#### 4.4.11 箭头

在左侧工具栏中增加了箭头tab。展开有各种箭头样式。



选中后拖动到画布上，可以对箭头进行填充线型动画的配置。

#### 4.4.12 图元

图元是用户自定义的具有业务逻辑的可复用元素。用户可以根据自身业务需要创建所需的图元，并在画面中为图元关联不同对象，从而大大扩展画面可展示的内容，并提高画面创建效率。

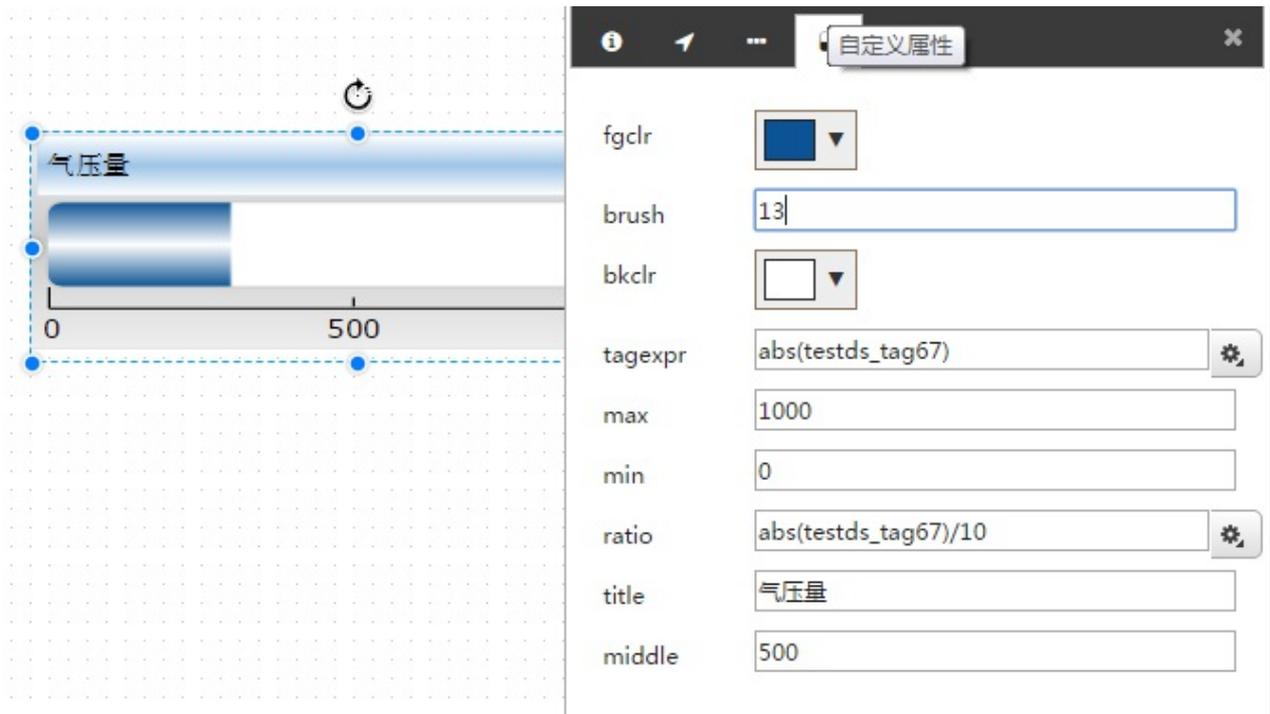
图元设计请参照 5 图元设计章节，本节所描述的是图元应用。

在设计器的左边栏中，图元根据用户在图元管理界面中的配置显示。

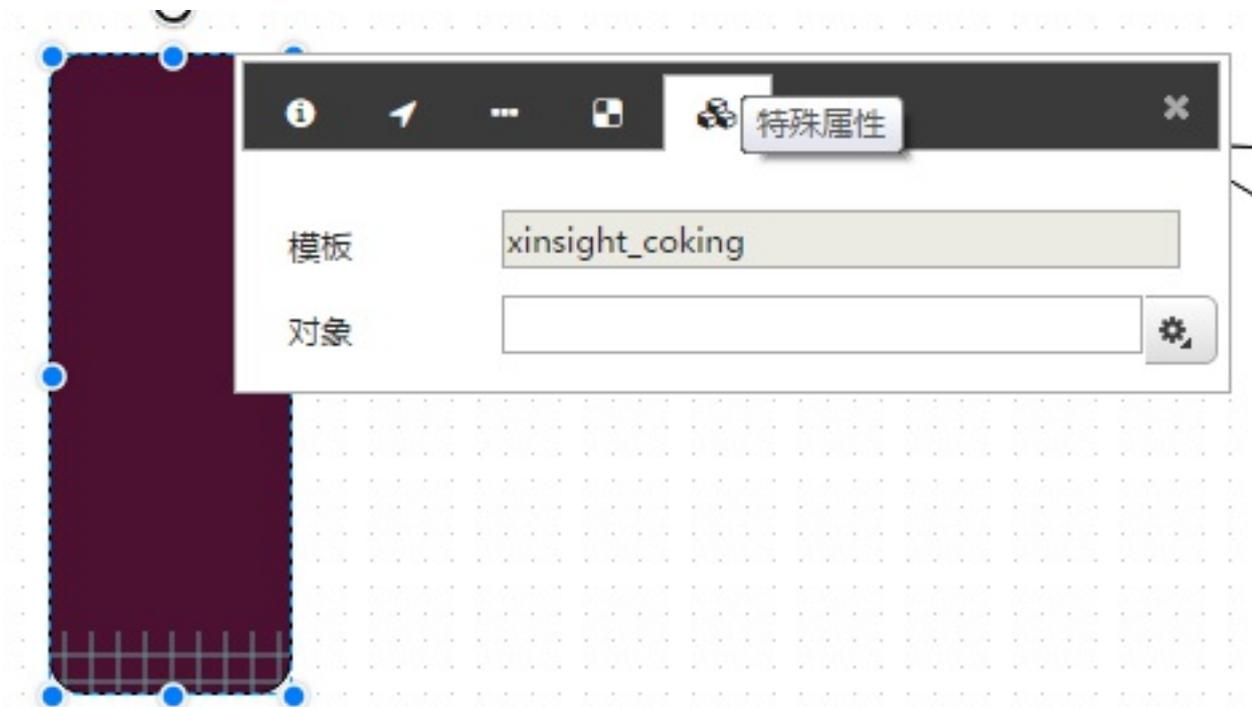
- 点击图元，拖拽到画面上。选中图元，显示属性配置标题栏
- 图元可以配置基本属性，位置属性，其他属性，方法与其他元素配置相同
- 图元还有自定义属性和特殊属性。
- 自定义属性中会列出在图元设计器中编辑图元时配置的所有自定义属性信息。根据不同类型显示相应的输入格式。如果类型是color，则显示颜色选择框；如果类型是var，则显示输入框；如果类型是变量，则在输入框后增加选点按钮，可以选择变量信息。
- 对于color类型和var类型的变量，值改变后会直接更新图元中的配置。

- 用户也可以通过脚本设置图元自定义属性的值。

如 `sys.SetAttribute("can39939680", "#fgclr", v);`



如果图元中配置了模板，在特殊属性中会显示配置的模板名，并在对象输入框中可以选择派生于该模板的对象。



点击选对象按钮，弹出对话框

数据源 xinsight

name	desc
▼ coking	
A050300	烧结
CVS	CVS

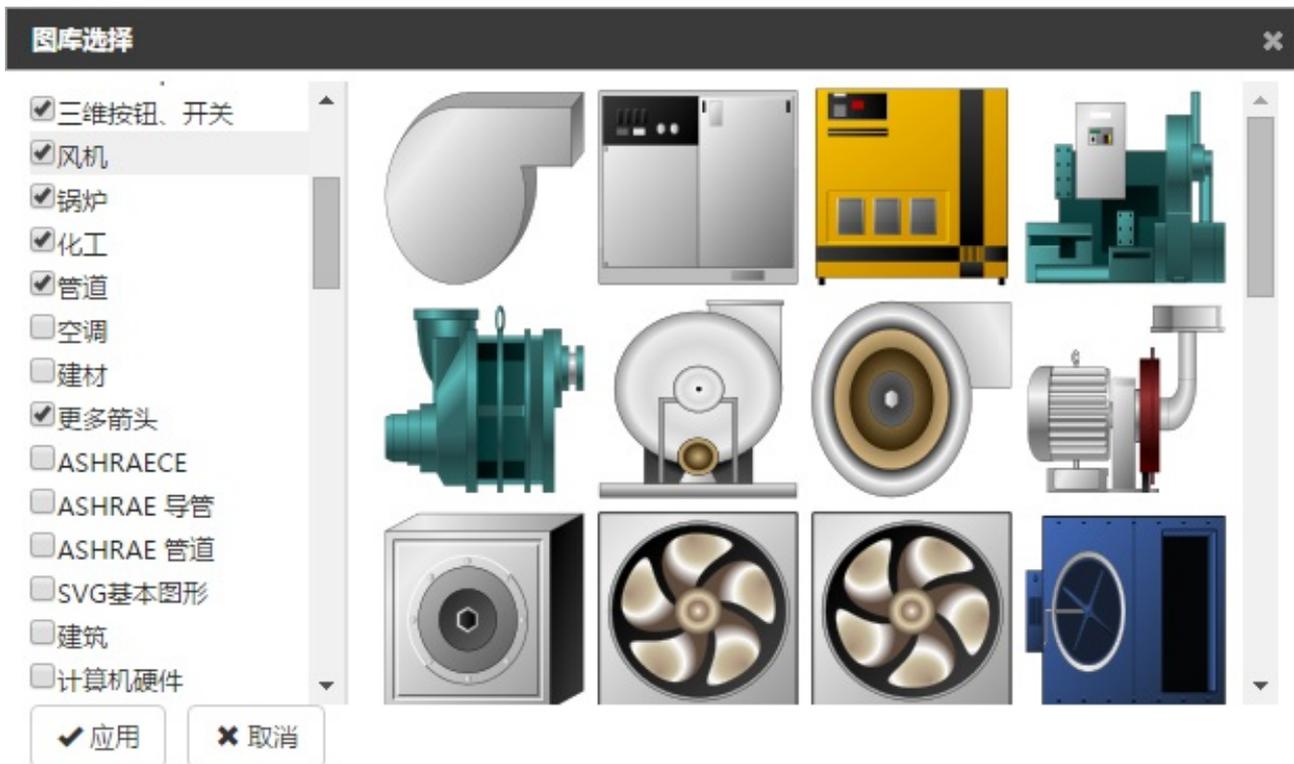
xinsight\_A050300

确认

选择对象后，该图元就可以实际运行了。

#### 4.4.13 工业图库

工业图库是由系统提供的一套样式丰富的矢量图片控件，共计 62 个大类。用户在首次访问设计器时会在边栏默认加载**三维按钮、风机、锅炉、化工、管道**等 5 个图库。可以在组态设计器或者图元设计器中拖拽使用，可以配置旋转闪烁等属性，可以与其他画面元素组合，也可以给图库元素关联脚本。



## 图库的选择

点击设计器边栏下方的 **更多图库** 按钮，可以弹出“图库选择”对话框，如下图所示。点击图库名称可在对话框右侧预览该图库，对于不需要在边栏显示的图库，可以取消勾选。边栏隐藏图库不会影响画面中已经拖拽出来的图库元素。点击 **应用** 按钮将会关闭图库选择对话框，并更新边栏的项目。

## 图库元素的使用

展开任意图库，鼠标移动到图标上方是将会出现预览浮层。鼠标拖拽任意图标至画面上释放，将在该位置上出现对应的图标。具体的属性配置(包括名称属性、位置属性、其他属性)、脚本设置，请参照图片控件中的相关章节。

完成编辑后保存画面，即可在运行态查看图库的使用情况。

### 4.4.14 图标图库

CVS 系统为用户提供了两组时下网页最流行的图标图库，分别叫做 **Font Awesome 图标**和 **Bootstrap 图标**。

包含 800 多个常见图标，包括加减、缩放、勾选等等图标。用户可以为图标图库配置颜色、旋转、显隐等属性。

## 图库的选择

点击设计器边栏下方的 **更多图库** 按钮，可以弹出“图库选择”对话框，勾选“Font Awesome 图标”或者“Bootstrap 图标”，点击 **应用** 按钮后边栏将出现对应图标图库。

## 图库元素的使用

图标图库与其他图形、图库相似，拖动相关图标至画面上释放，将在该位置上出现对应图标。用户可以选中创建出来的图标，对其进行缩放、旋转，并可通过属性条配置基本属性、字体属性、位置属性以及其他属性四个项目。其中基本属性、位置属性、其他属性这三个界面可以参考前面的章节。用户可以在字体属性界面中配置图标的颜色。

值得注意的是，图标图库缩放与图片、工业图库有所不同。图标图库在缩放中维持宽高比，当水平、竖直中某一个达到当前最大时，拖动另一侧边框不会增大图标尺寸。

完成编辑后保存画面，即可在运行态查看图库的使用情况。

#### 4.4.15 仪表

仪表工具栏中支持以下仪表：

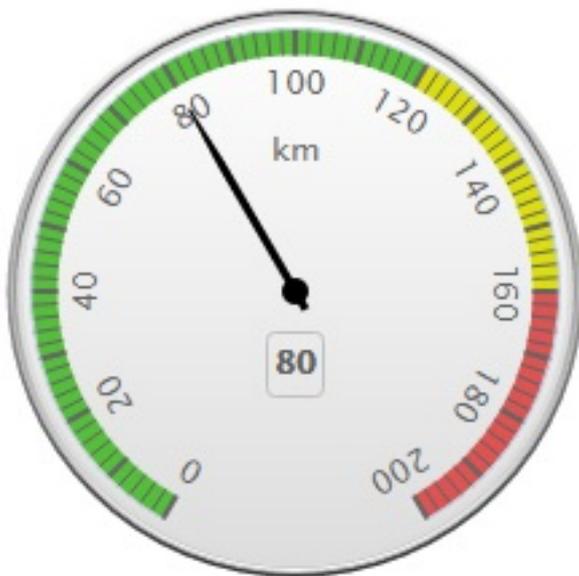


- 速度仪
- 仪表盘
- 仪表盘2
- 伏压图

#### 速度仪

双击速度仪或者直接拉到画面上，可以看到速度仪如下图

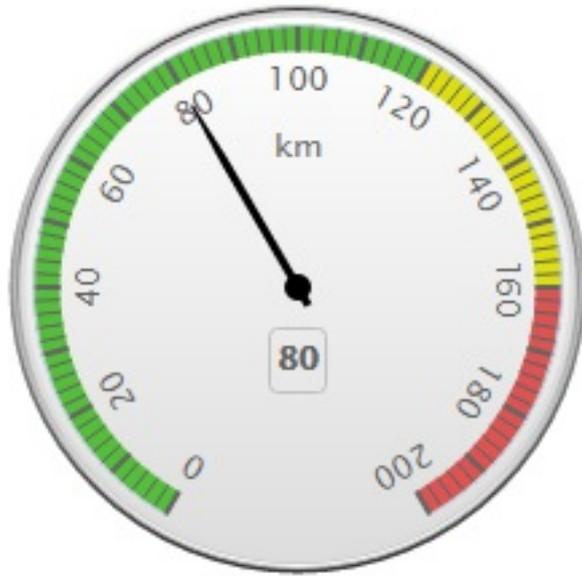
### Speedometer



在属性框中选择仪表属性，可以配置速度仪的属性



Speedometer



标题 Speedometer

单位 km

起始角度 -150 结束角度 150

最小值 0 最大值 200

tag点

- 标题，可以为空
- 单位，可以为空
- 起始角度， $-180^{\circ}$ 到 $180^{\circ}$ ，必须为数字
- 结束角度， $-180^{\circ}$ 到 $180^{\circ}$ ，必须为数字
- 最大值，必须为数字
- 最小值，必须为数字
- tag点，关联tag点后运行态根据tag点进行变化。当前值没有超过仪表盘60%时显示绿色，60%~80%显示黄色，80%~100%显示红色

### 仪表盘

双击仪表盘或者直接拉到画面上，可以看到仪表盘如下图

0 50 100 150 200

## Speed



在属性框中选择仪表属性，可以配置仪表盘的属性



Speed



ⓘ ↶ 📈 ✕

标题

单位

---

起始角度  结束角度

最小值  最大值

---

tag点  📺 ⚙️

---

- 标题，可以为空
- 单位，可以为空
- 起始角度， $-90^{\circ}$ 到 $90^{\circ}$ ，必须为数字
- 结束角度， $-90^{\circ}$ 到 $90^{\circ}$ ，必须为数字
- 最大值，必须为数字
- 最小值，必须为数字
- tag点，关联tag点后运行态根据tag点进行变化。

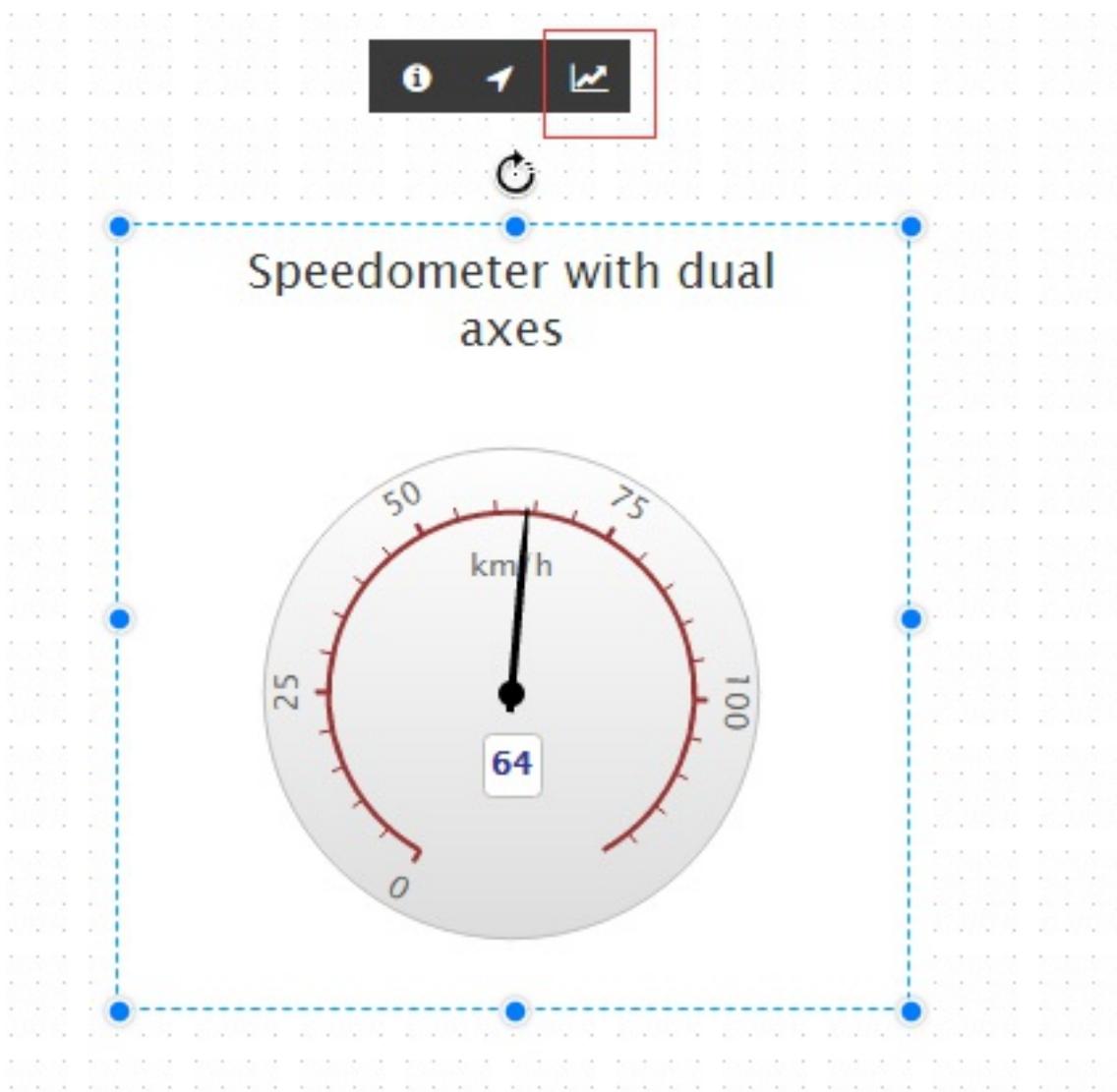
## 仪表盘2

双击仪表盘2或者直接拉到画面上，可以看到仪表盘2如下图

## Speedometer with dual axes



在属性框中选择仪表属性，可以配置仪表盘2的属性



ⓘ
➔
📈
✕

标题

单位

---

起始角度  结束角度

最小值  最大值

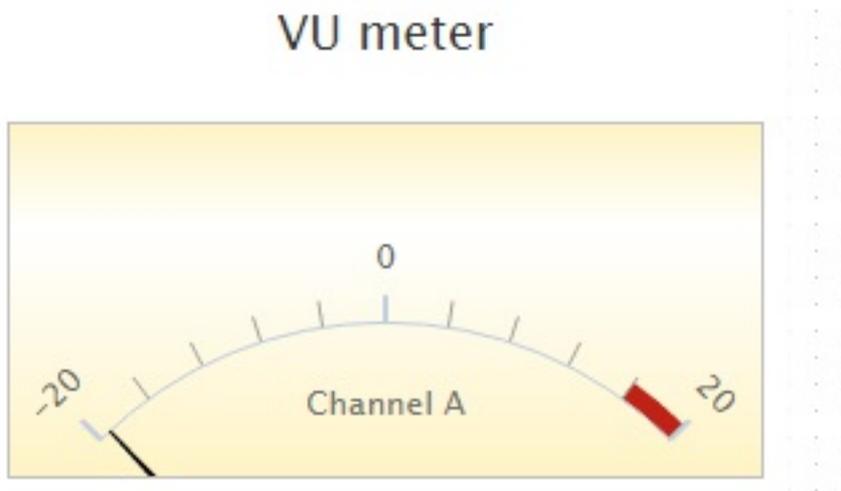
---

tag点  📄 ⚙️

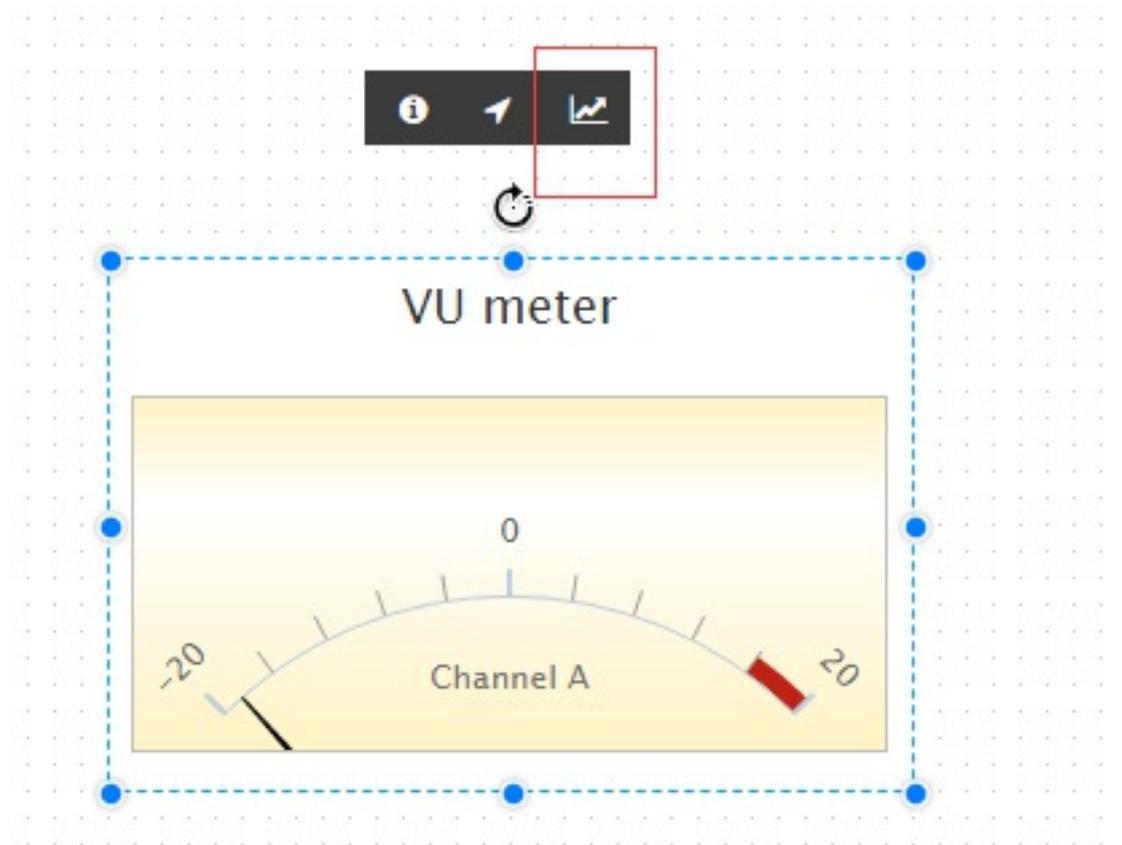
- 标题，可以为空
- 单位，可以为空
- 起始角度， $-180^{\circ}$ 到 $180^{\circ}$ ，必须为数字
- 结束角度， $-180^{\circ}$ 到 $180^{\circ}$ ，必须为数字
- 最大值，必须为数字
- 最小值，必须为数字
- tag点，关联tag点后运行态根据tag点进行变化。

## 伏压图

双击伏压图或者直接拉到画面上，可以看到伏压图如下图



在属性框中选择仪表属性，可以配置伏压图的属性



标题 VU meter

单位 Channel A

起始角度 -45 结束角度 45

最小值 -20 最大值 20

tag点

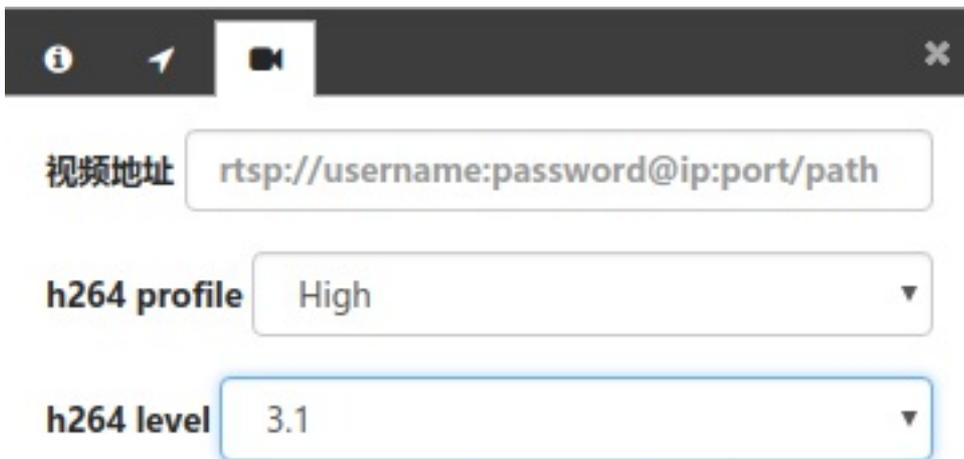
- 标题，可以为空
- 单位，可以为空
- 起始角度， $-45^{\circ}$ 到 $45^{\circ}$ ，必须为数字
- 结束角度， $-45^{\circ}$ 到 $45^{\circ}$ ，必须为数字
- 最大值，必须为数字
- 最小值，必须为数字
- tag点，关联tag点后运行态根据tag点进行变化。当前tag点值范围超过最大值的90%显示 红色。

#### 4.4.16 简单视频控件

简单视频控件提供了在运行态播放来自网络摄像头或者硬盘录像机实时画面的功能。简单视频控件不提供摄像头云台控制、缩放、截屏与录像等功能。

从高级控件一栏中将简单视频控件的图标拖拽到画面中需要的位置。默认的控件大小为 1280\*720 像素，可以通过位置属性调整到合适的大小。运行态的实际大小会受到视频宽高比以及画布自适应选项的影响。（视频控件宽高比与视频源保持一致）

视频属性的配置如下图所示。视频地址一栏需要填入网络摄像头或者硬盘录像机提供的 RTSP 流地址，可以通过 [VLC \(https://www.videolan.org/\)](https://www.videolan.org/) 或者类似的播放器打开该地址校验是否可以播放。



视频地址

h264 profile

h264 level

H264 profile/level 是于视频编解码有关的选项，如果底层设备的配置界面中没有给出相关的信息，可以通过 [FFmpeg \(https://www.ffmpeg.org/\)](https://www.ffmpeg.org/) 录制一小段视频文件再通过 [MediaInfo \(https://mediaarea.net/en/MediaInfo/Download\)](https://mediaarea.net/en/MediaInfo/Download) 工具获取视频信息。ffmpeg 的命令如下，请保留 rtsp 链接前后的半角引号，以免链接中的 & 符号终止命令。

```
# 请在命令行执行这条命令，运行几秒之后可以通过 ctrl+c 终止录制。
ffmpeg -i "rtsp://username:password@ip:port/path" -an -c:v copy -f mp4 -
movflags +frag_keyframe+empty_moov+default_base_moof pipe:1 > sample.mp4
```

通过 MediaInfo 查看上述示例输出的 sample.mp4 文件，输出的信息如下示例。Format profile 一行的显示说明这一个网络地址提供的是 Profile high + Level 3.1 方式编码的 H264 视频。请依据实际设备的编码信息配置控件。如果文件信息中没有 Format profile 这一行，则视频控件可能无法播放，需要借助其他实时的转码服务将视频转换为标准 H264 编码方式。

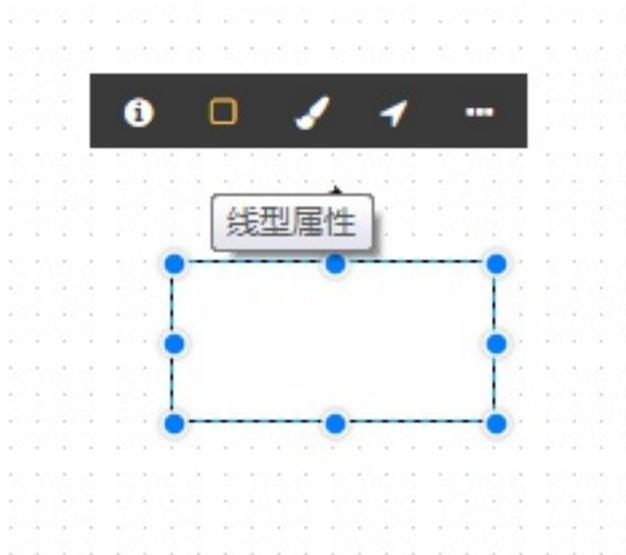
```
Video
ID : 1
Format : AVC
Format/Info : Advanced Video Codec
Format profile : High@L3.1
Format settings : CABAC / 2 Ref Frames
Format settings, CABAC : Yes
Format settings, ReFrames : 2 frames
Codec ID : avc1
Codec ID/Info : Advanced Video Coding
Duration : 4 s 190 ms
```

#### 说明：

1. 由于 Chrome 安全策略的限制，运行态视频控件不能自动播放，需要点击控件上的播放按钮。
2. 由于网络缓冲的缘故，网页上播放的视频可能会比通过 VLC 播放滞后 3~4 秒。

## 4.5 属性配置

在画面设计过程中，为每个元素设置其属性，是经常用到的功能。CVS中的属性设置，通过在画面中选择图形后，上方会出现浮动的属性框工具栏。点击相应按钮会弹出配置框，所有的配置在改变后会立即生效，鼠标移动到图标上会有对应的提示信息。



#### 4.5.1 画面属性

- 在画面空白处，右键菜单选择“画面属性”，弹出画面属性对话框，包括基本信息和填充属性框
- 在基本信息中可以配置画面描述，刷新周期，画布高宽以及画面自适应风格
- 画布自适应选择宽度自适应，则在运行时按照运行环境的宽度来缩放画面
- 画布自适应选择高度自适应，则在运行时按照运行环境的高度来缩放画面
- 画布自适应选择原始大小，则在运行时不进行缩放，按照原始大小显示
- 画面授权项，在云模式下，画面授权项不能配置，如果勾选已启用，默认就是画面的名称。在产品模式下可以选择授权项。如果不启用需要点击×删除该授权项。
- 勾选上 运行时启用画面缩放功能 则可以在运行态按下 F9 功能键，通过鼠标滚轮放大、缩小画面。缩放位置以转动鼠标滚轮时所在的位置为中心。目前缩放的倍率为 1 / 1.25 / 1.5 / 2 / 3 / 4 / 5。  
(1 为画面初始显示的大小)



窗口标题

demo

刷新周期

1000

画面描述

画布 宽度

1440

画布 高度

900

画布自适应

原始大小

画面授权项



运行时不显示控制条

运行时启用画面缩放功能



窗口标题

ggg

刷新周期

1000

画面描述

画布 宽度

1440

画布 高度

900

画布自适应

宽度自适应

画面授权项



窗口标题

abc

刷新周期

1000

画面描述

---

画布 宽度

画布 高度

画布自适应 高度自适应 ▼

画面授权项 resourceC × 

---

在填充属性中，可选择画面的填充前景色、背景色、填充图案或背景图片



是否填充

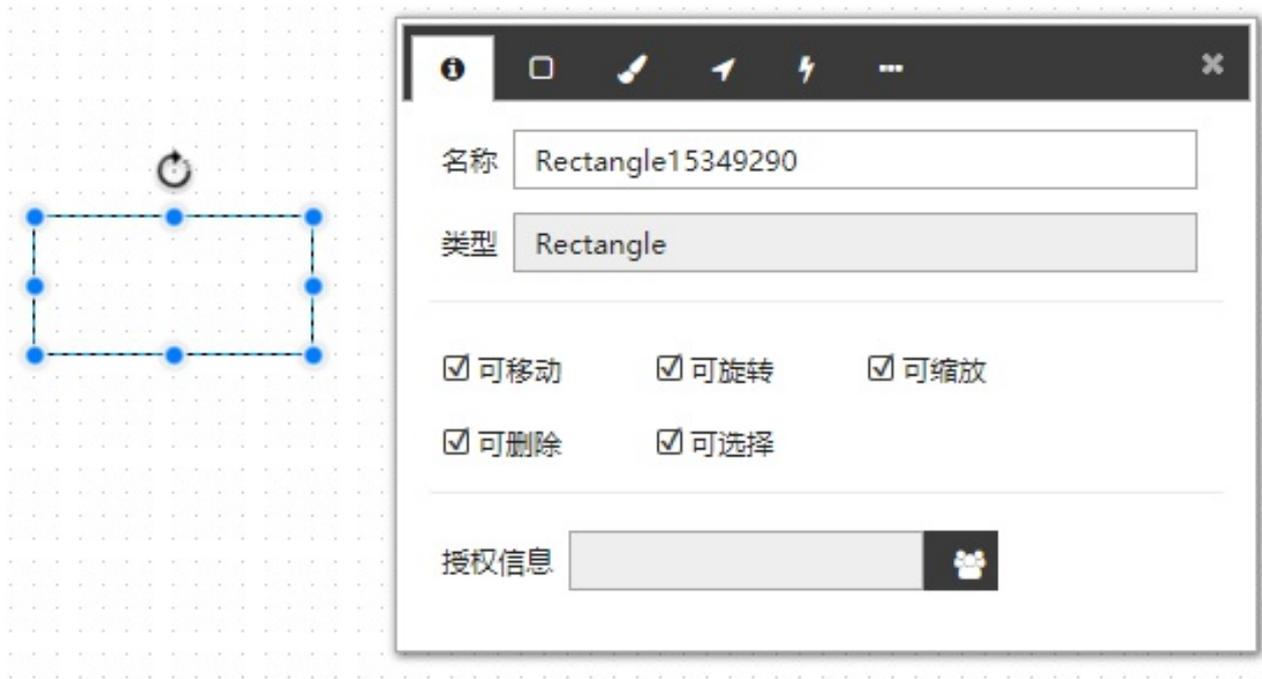
前景颜色  背景颜色

填充图案  ▼

背景图片  

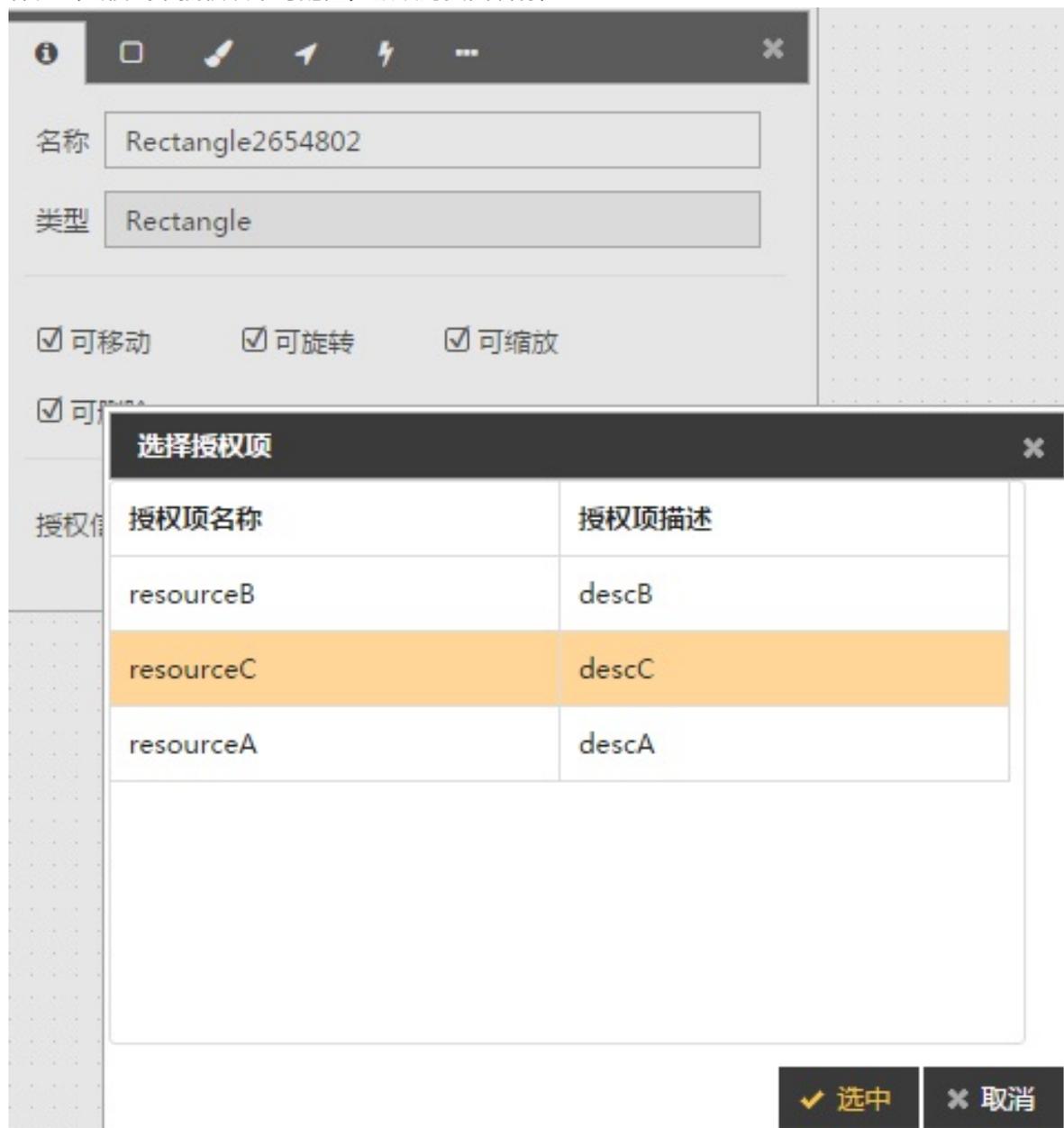
#### 4.5.2 基本属性

点击基本属性，弹出属性框内容



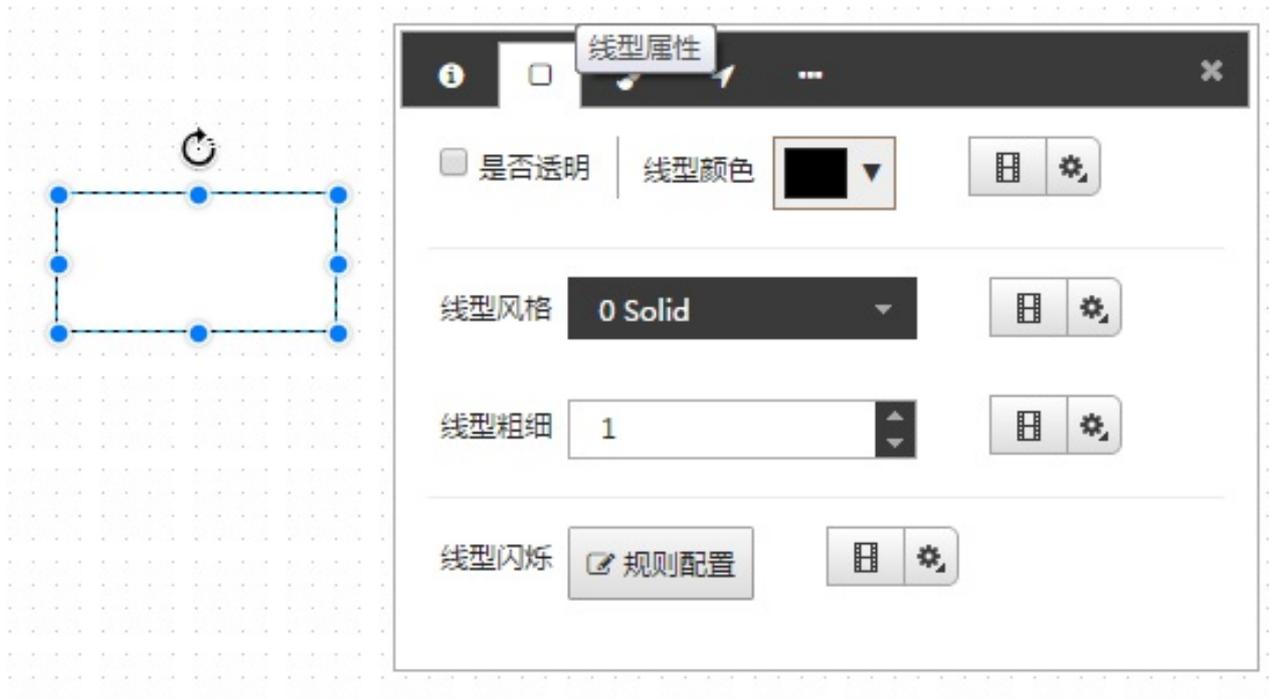
- 可以查看元素名称和类型，可以修改名称。如果元素来自图元，还可以查看元素在系统中的路径。
- 可以配置元素是否可移动，可旋转，可缩放，可删除，可选择属性。当元素处于不可选择状态时，点击元素不会出现属性框工具栏和缩放框，这时可以通过顶部菜单的**文件-页面统计信息**点击元素来选中。
- 授权信息：元素可以配置授权信息，如果当前用户没有该授权项，则不显示该元素。如果拥有该授权项，则运行态可以显示该元素，并且拥有可见功能，则该用户可以看到这个元素,但是不能触发OnClick\OnDbClick\OnMouseOver\OnMouseDown\OnMouseUp 这些事件。如果拥有该授权

项并且拥有可控功能， 则该用户不仅可以看到这个元素,还可以通过脚本控制该元素,可以触发以上事件。（云模式下授权项不可配置，默认为页面名称）



### 4.5.3 线型属性

点击线型属性tab，弹出属性框内容

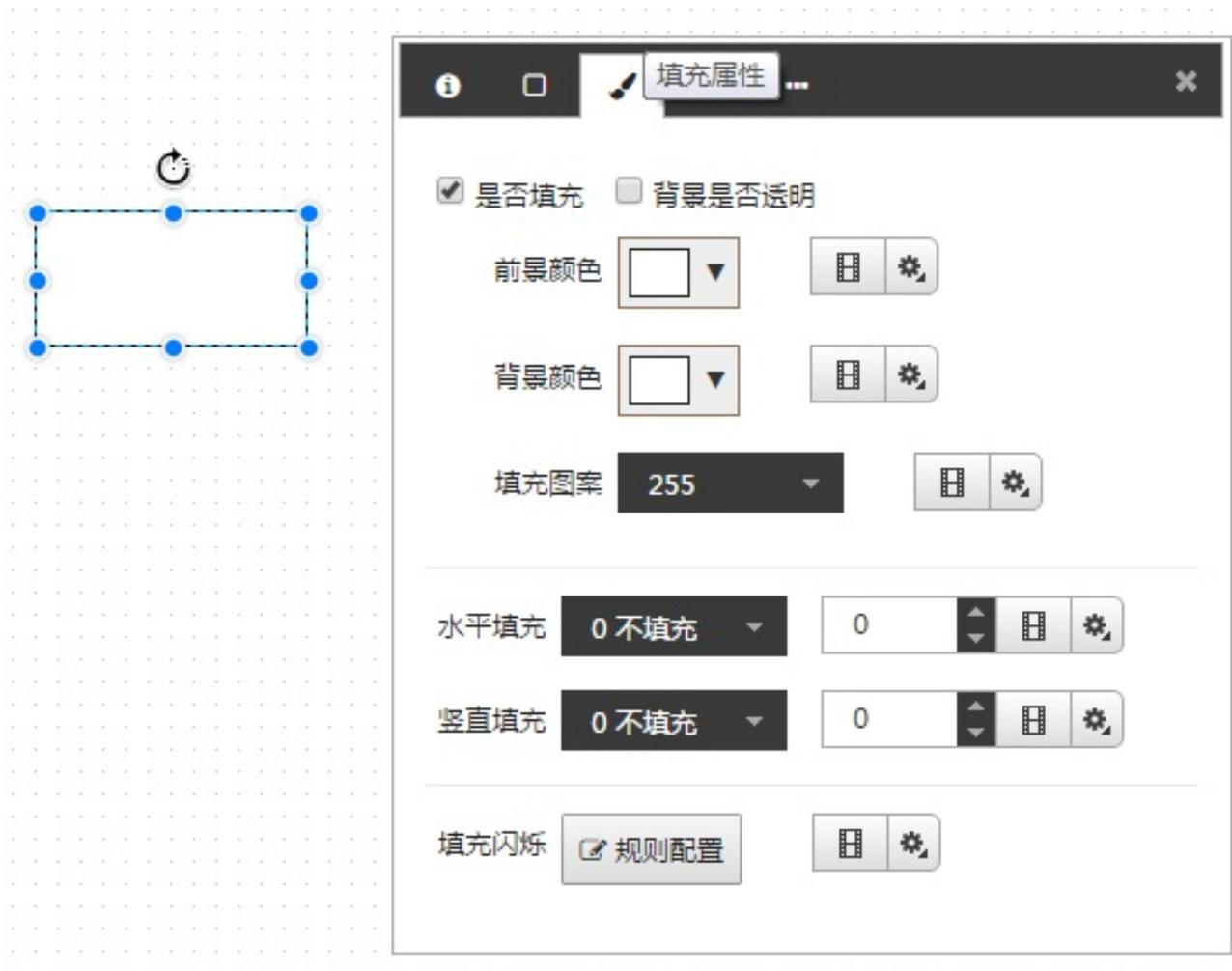


- 可以选择线型颜色，风格，粗细。
- 若勾选是否透明，则线型为透明，不显示。

线型属性支持颜色规则匹配和闪烁规则匹配，需要搭配动画功能。请参阅[元素动画配置](#)、[颜色规则配置](#)以及[闪烁规则配置](#)等章节。

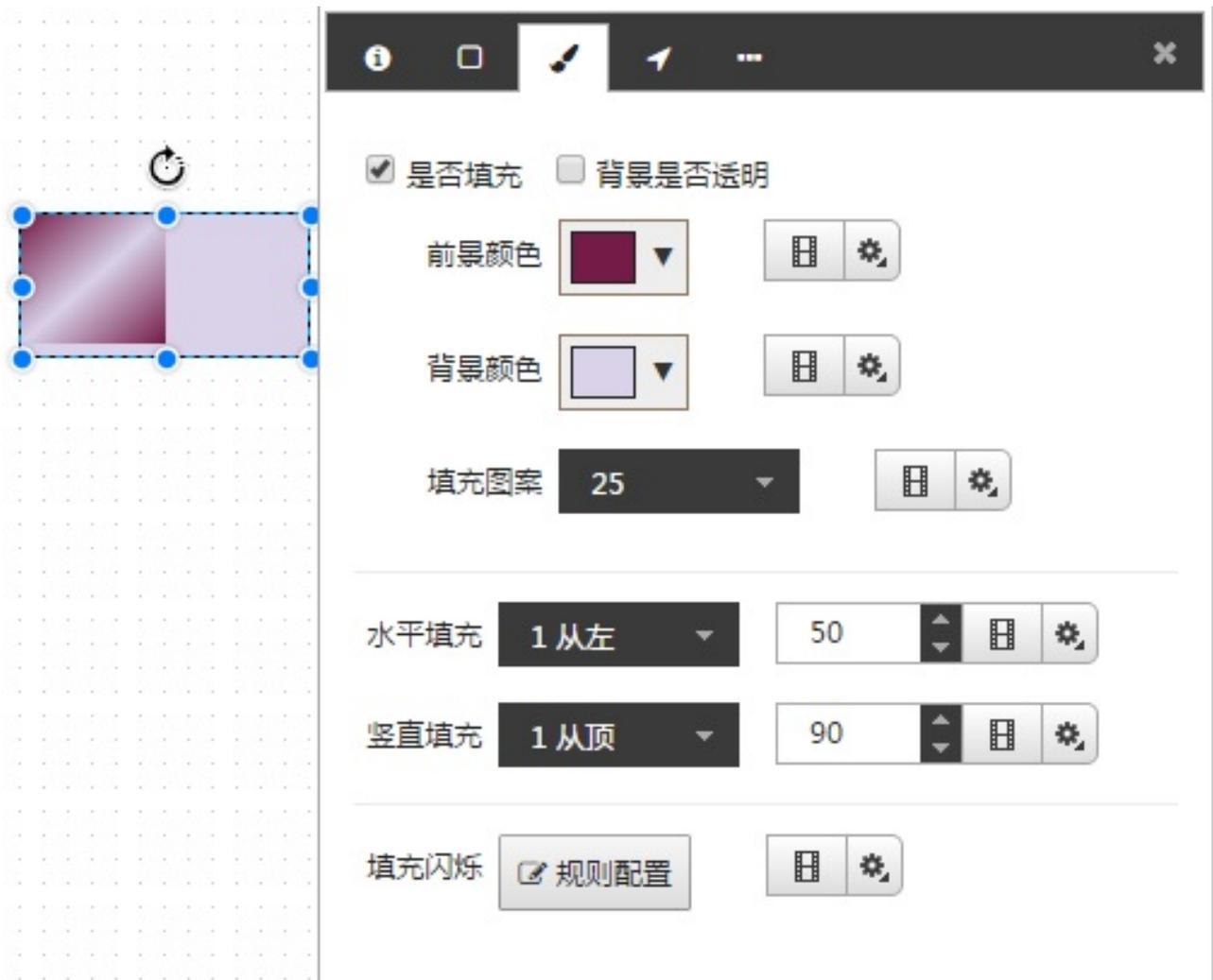
#### 4.5.4 填充属性

点击填充属性，弹出属性框内容



- 若勾选背景透明，则背景色为透明，不显示。
- 可以配置前景色，背景色，填充图案，水平填充比例，竖直填充比例及填充闪烁。
- 填充的时候，如果设置了画刷风格，背景透明不起作用。

填充属性支持颜色规则匹配和闪烁规则匹配，需要搭配动画功能。请参阅[元素动画配置](#)、[颜色规则配置](#)以及[闪烁规则配置](#)等章节。



#### 4.5.5 位置属性

点击位置属性，弹出属性框内容



- 可以配置图形的左上宽高信息，水平缩放倍数，竖直缩放倍数及缩放方向信息。
- 根据配置会将图形进行相应的缩放和位移。
- 快捷操作，可以选中一个元素，按住ctrl + up, left, right, down 键，元素大小进行微调，每次调整一个像素。
- 对于直线类型，还可以配置起点和终点的信息



The image shows a software interface with a grid background. On the left, a blue line connects two points on the grid, with a circular arrow icon indicating rotation. On the right, a settings panel is displayed with various input fields and controls.

Settings panel details:

- 左: 660
- 上: 120
- 宽: 100
- 高: 100
- 起点X: 660
- 起点Y: 220
- 终点X: 760
- 终点Y: 120
- 水平缩放倍数: 1
- 竖直缩放倍数: 1
- 水平缩放方向: 水平从左
- 竖直缩放方向: 竖直从顶

#### 4.5.6 其他属性

其他属性配置



- 可以配置旋转角度，旋转速度，显隐频率，声音，提示信息等。
- 旋转角度可以在设计器直接展示。其他的属性只有在运行态才会生效。

#### 4.5.7 文本属性

对于文本元素，可以配置文本



- 可以配置显示的文本，显示方式，水平对齐方式，竖直对齐方式，输出格式等信息。
- 如果勾选自适应，系统会根据输入的文本内容改变文本框的大小。
- 在画面中双击元素，可以直接输入文本内容。

填充属性支持动画功能和文本规则匹配。请参阅[元素动画配置](#)和[文本规则配置](#)等章节。动画可单独使用，显示变量表达式的值。在配置文本规则后，依据变量表达式的值显示对应内容。

#### 4.5.8 字体属性



可以配置字体，字号，颜色，粗体，斜体，下划线，删除线等属性。

字体属性颜色支持颜色规则匹配，需要搭配动画功能。请参阅[元素动画配置](#)、[颜色规则配置](#)以及[闪烁规则配置](#)等章节。



#### 4.5.9 元素动画配置

画面中元素的属性，可以关联数据点，实现动态展示画面效果。点击是否启用动画按钮，可以选择是否启用动画配置。



点击选点按钮，弹出选点对话框



对话框左边树形栏可选择对应的所有设备变量或者在子系统下属性

选点

testds

Search

所有设备变量

子系统

- subsys1
- subsys2

名称	描述
cus_home_light	
cus_home_tv	
cus_mall_persons	
cus_mine_speed	
cus_mine_status	
cus_mine_sys_switch	
cus_mine_today_finish	
cus_mine_whole	
cus_mine_yesday_finish	
cus_profit	
cus_sale	

1 共 1,503 页 20 共30,041条

确认

选点时，可以在对话框右上角的搜索框中输入关键字对点进行查询，按下回车，会得到与名称字段或描述字段相符合的点。云模式下同样支持查询，但不支持对描述字段的查询。选点后，点击确定，将该变量关联到对应的属性上。点击右上角 fx 按钮会弹出表达式配置框，配置表达式。也可以直接在输入框中输入表达式。

表达式
✕

**基本运算**

7	8	9	+	<<	>	<
4	5	6	-	>>	(	)
1	2	3	*	&&	!	^
0	%	/	&			

**函数运算**

sin	cos	tan	abs
asin	acos	atan	log
ceil	max	min	pow
sqrt	eq	neq	conret
strconcat			

#### 4.5.10 闪烁规则配置

点击闪烁规则，弹出对话框。

闪烁配置
✕

范围比较  完全匹配

	低值	高值	闪烁频率	显示颜色ON	显示颜色OFF
1	10	50	慢速 ▼	<span style="color: red;">■</span> ▼	<span style="color: green;">■</span> ▼

+ 添加

- 删除

✓ 保存

- 可以添加、删除匹配规则。点击保存按钮，保存规则配置。
- 可以在范围比较、完全匹配间切换。
- 闪烁在配置的显示颜色 ON/OFF 间变换。

- 若选用范围比较，则在 变量值  $\geq$  低值 且 变量值  $<$  高值 时满足规则条件。
- 若选用完全匹配，则在 当前值  $==$  变量值 时满足规则条件。
- 当有多条规则条件同时满足时，序号排名最小的规则被应用。



#### 4.5.11 颜色规则配置

颜色规则匹配的按钮藏于颜色选择框中，点击 规则匹配 按钮弹出配置对话框。



- 可以添加、删除配置规则。点击保存按钮，保存规则配置。
- 可以在范围比较、完全匹配间切换。

- 若选用范围比较，则在 变量值  $\geq$  低值 且 变量值  $<$  高值 时满足规则条件。
- 若选用完全匹配，则在 当前值  $==$  变量值 时满足规则条件。
- 当有多条规则条件同时满足时，序号排名最小的规则被应用。没有规则满足时，使用默认颜色。

**颜色配置**
✕

范围比较     完全匹配
 默认颜色            ▼

	低值	高值	对应颜色
1	1	2	<span style="background-color: red; color: red;">          </span> ▼

+ 添加
- 删除
✓ 保存

#### 4.5.12 文本规则配置

文本规则匹配可以独立于动画配置，但如果需要在运行态生效，则必须先启动动画配置。点击铅笔图标，即可打开文本规则匹配的配置界面。

ⓘ
T
A
□
✎
↖
⋮

文本

✎
🎬
⚙️

- 文本规则匹配有字符串比较和范围比较两种模式，默认使用字符串匹配。
- 若选用范围比较，则在 变量值  $\geq$  低值 且 变量值  $<$  高值 时满足规则条件。
- 若选用字符串匹配，则在 输入值  $==$  变量值 时满足规则条件。
- 当有多条规则条件同时满足时，序号排名最小的规则被应用。没有规则满足时，使用默认显示值。

### 文本匹配配置 ✕

默认显示值

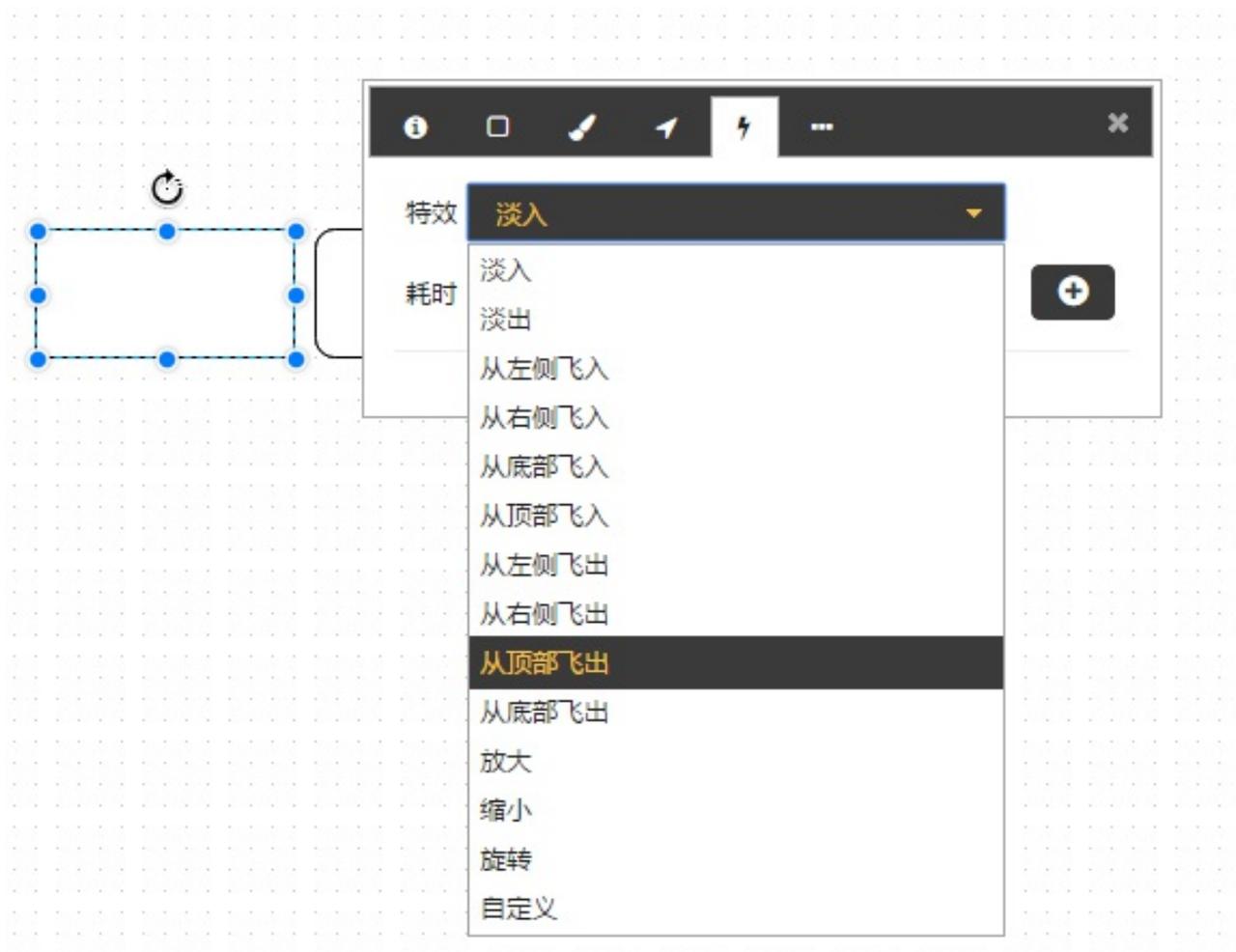
范围比较  字符串匹配

	输入		输出
1	1		foo
2	2		bar

#### 4.5.13 动画配置

可以对画面中的各个通用图形，文本、图像、组合、图库和图元进行动画配置，可配置的动画包括两种类型，一种是特效动画，一种自定义动画。其中特效动画包括了淡入、淡出、从左侧飞入、从右侧飞入、放大、缩小、旋转等；自定义动画包括两种形式，一种是位置动画，一种是路径动画。

选中需要配置动画的元素，在弹出的属性框中选择**动画属性**标签页，如下图



### 特效动画配置

在特效下拉框中选择一种类型，比如淡出，再选择耗时和延迟，单位是秒，然后点击右侧的添加按钮，这样就可以为该元素增加一个淡出效果的动画。



### 自定义动画配置

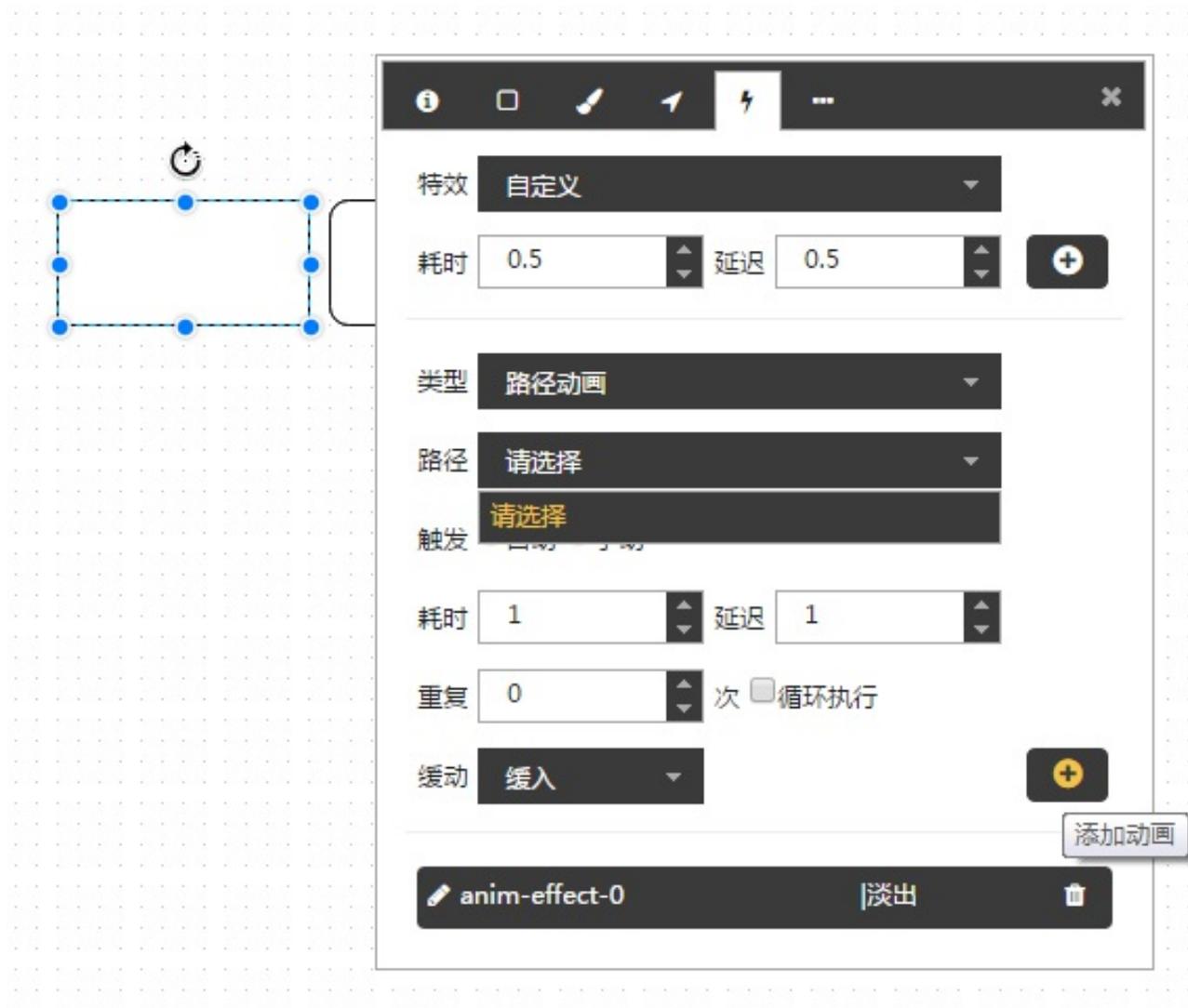
自定义动画可以选择位置动画或者路径动画，位置动画是从A点B点直线运动，路径动画是让元素可以按照一条用户指定的路径运动，可以是曲线、折线、椭圆等。

### 位置动画配置

项目	说明
目标	最终要到达位置的坐标
触发	自动是不需要脚本去触发该动画；手动需要通过脚本去触发。（脚本使用 PlayAnimation 和 PauseAnimation）
耗时和延迟	单位为秒
重复	可以设置重复次数或者持续循环执行
缓动	有四种缓动效果可以选

路径动画配置需要首先选择一个路径，也就是让元素按照该路径去运动，其余配置与 **位置动画** 一致。

配置完成后，点击添加动画按钮。



### 修改动画名称

在动画配置窗口的下方动画列表中，可以点击动画名称修改，后续可以通过脚本来操作该名称的动画。

## 4.6 脚本

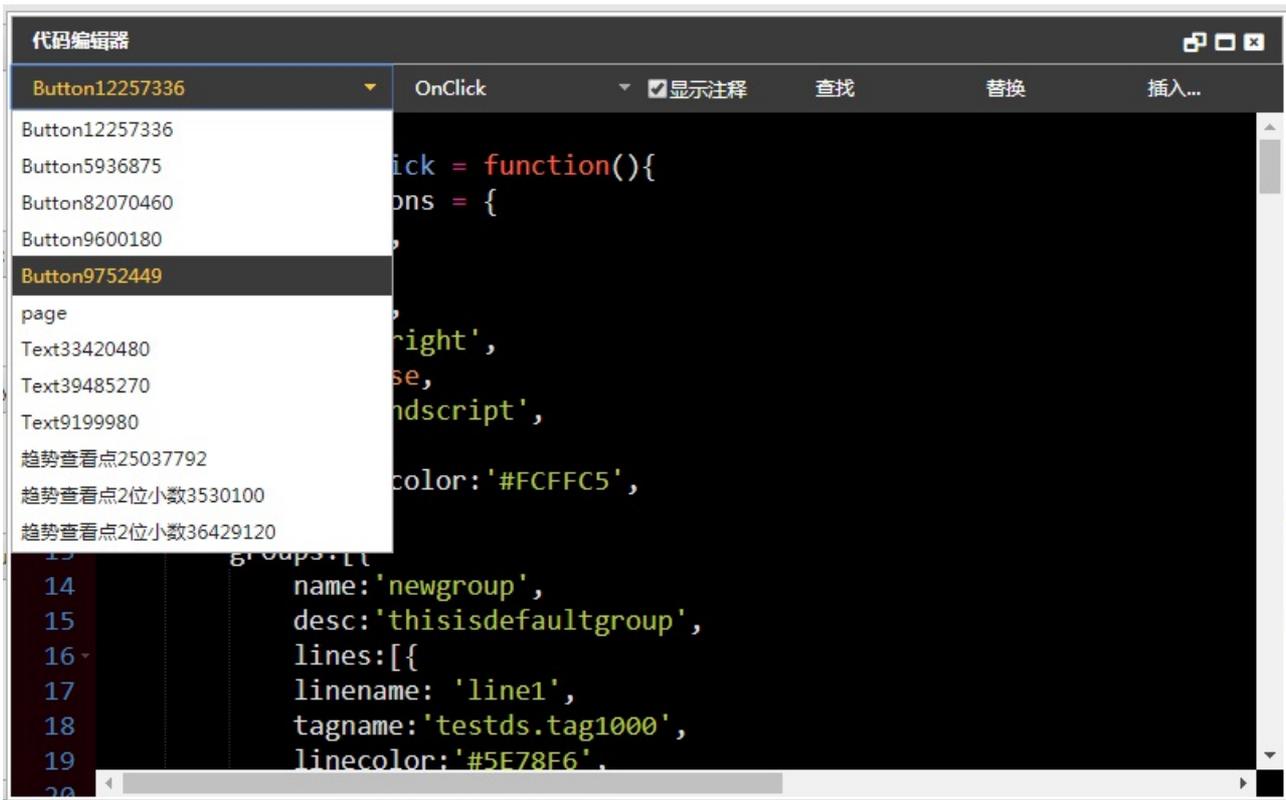
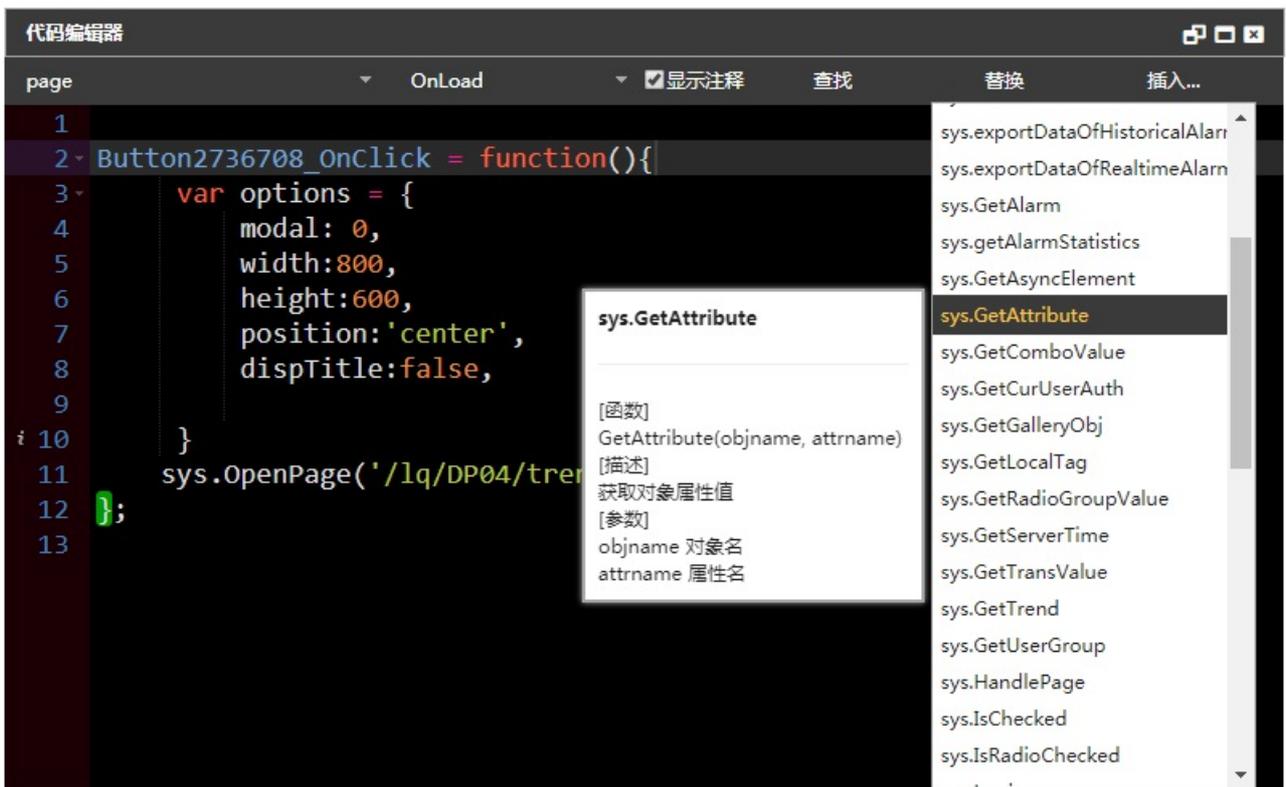
CVS系统的所有脚本使用JavaScript语言描述，通过为元素或者画面关联事件，可以实现复杂的业务逻辑。

**注意：**脚本中请勿包含 ]]> 字符。相关数组元素的获取和使用可以使用 ]] > 或者 ] ]> 之类代替，或者使用变量取代数组元素获取。

1. 在画面空白处，右键菜单中选择脚本编辑，弹出脚本编辑框；
2. 在画面中选中元素，右键选择脚本编辑，弹出脚本编辑框，并且自动选择该元素

```
1
2 Button2736708_OnClick = function(){
3     var options = {
4         modal: 0,
5         width:800,
6         height:600,
7         position:'center',
8         dispTitle:false,
9     }
10 }
11 sys.OpenPage('/lq/DP04/trendscript', options);
12 };
13
```

- 点击事件下拉框，会自动生成方法。
- 点击插入按钮，会弹出所有系统函数，鼠标移上去显示函数说明。
- 点击查找按钮，会弹出查找对话框，在输入框中输入想要查找的内容，随着用户输入，代码编辑器会自动在当前脚本中进行搜索；可以选择是否启用正则表达式；是否大小写敏感；以及是否整词搜索；
- 点击替换按钮，会弹出替换对话框，在上一行的查找输入框中输入想要替换的字符串，在第二行中输入想要替换为的字符串，可以进行单个替换或替换所有
- 在编辑器中使用代码补全输入函数，会显示出函数的参数列表
- 勾选显示注释checkbox,用户输入系统函数，显示注释，如果不勾选，则不显示
- 点击对象选择下拉框，所有对象都按照字母顺序排列



#### 4.6.1 系统函数

CVS提供了多个系统函数，供用户调用。

##### 4.6.1.1 CloseModalPage()

关闭模态画面。

- 引入版本：1.8

#### 4.6.1.2 ClosePage(pageID)

关闭指定ID的画面。

- 参数：
  - pageID 画面ID
- 引入版本：1.8

示例：

```
var options = {
    modal:3,
    oldtags:"testds.tag10",
    newtags:"testds.tag100",
    width:"400",
    height:"400",
    position:"center",
    dispTitle:"no",
    title:'w2',
    pageID:"test_close"
};
sys.OpenPage("/test/测试", options);

sys.ClosePage("test_close")
```

#### 4.6.1.3 confirmAlarmGlobal(options)

全局确认报警接口，该接口无需报警控件。

- 参数：
  - options 对象，却仍报警所需的过滤参数及回调函数。过滤参数与 alarm.setFilter 兼容，filterPrioritySetting 除外（参见下方说明）。
    - confirmAll: Number 类型，0 或者 1，默认值为 0。1 表示确认所有报警，过滤条件不生效。
    - filterPrioritySetting: String 类型，用于优先级过滤，可以使用半角逗号分隔多个优先级段。例如 0,2|4 表示过滤优先级为 0、2、3、4 的优先级。
    - success: 函数类型，可选参数，用于确认成功后调用，接受三个参数 data, status 和 xhr，含义参见 [jQuery AJAX API 说明 \(http://api.jquery.com/jquery.ajax/\)](http://api.jquery.com/jquery.ajax/)。
    - error: 函数类型，可选参数，用于确认失败时调用，接受三个参数 xhr, status 和 err，含义参见 [jQuery AJAX API 说明 \(http://api.jquery.com/jquery.ajax/\)](http://api.jquery.com/jquery.ajax/)。
- 返回值：无返回值
- 引入版本：2.3

示例：

```

sys.confirmAlarmGlobal({
  enableFilterPriority: 1,
  filterPrioritySetting: "0,2|4,6", // <---- 确认 0、2、3、4 和 6 级报警
  success: function(data) {
    console.log(data);
  },
  error: function(xhr, status, err) {
    console.log(status, err);
  }
});

sys.confirmAlarmGlobal({ confirmAll: 1 });

```

#### 4.6.1.4 exportDataOfHistoricalAlarms(options)

导出历史报警数据，保存为 cvs 或者 tsv 格式。

- 参数：
  - options 对象，配置参数，可以为空。它包含导出格式、过滤条件，说明如下：
    - filetype 选填，字符串类型，用于导出的文件格式判断说明，支持的文件格式包括使用逗号分隔值("csv")和制表符分隔值("tsv")，默认值为 "csv"；
    - colModel 选填，数组，需要显示的报警数据字段。默认值为全部报警数据字段。
    - colName 选填，对象，它的键名为报警数据字段名，键值为打算替换的字段名。导出数据的列名顺序按照 colModel 数组中的字段顺序显示，与 colName 无关。
    - tagname 选填，选填，字符串，代表需要查询的点名。为空代表所有点。
    - priority 选填，字符串，优先级，"1|3" 代表优先级为 1、2、3 级的报警。为空代表优先级为"0|255"。
    - amtype 选填，字符串，表示报警类型，默认值为 "0"。有效值为 0~10 的整数，以此代表"所有报警", "高高报警", "高报警", "低报警", "低低报警", "变位报警", "偏差报警", "0状态报警", "1状态报警", "0->1报警", "1->0报警"。
    - da 选填，字符串，数据源名称。为空代表所有数据源。
    - time 选填，字符串，导出的起止时间。为空表示全部时间段。格式参见示例。
- 返回值：无返回值
- 引入版本：2.3

colModel 和 colName 默认的字段的、显示名称和顺序如下：

No.	colModel	colName
1	daName	数据代理
2	alarmTag	报警点
3	priority	优先级
4	alarmType	报警类型
5	alarmTagValue	报警值
6	occurTime	发生时间
7	recoverTime	恢复时间
8	alarmState	报警状态
9	confirmStatus	确认状态
10	nodeName	节点名称
11	alarmID	报警ID

12 alarmTagDesc 报警点描述  
13 confirmor 确认者  
14 confirmTime 确认时间  
15 className 所在类  
16 subSysName 子系统  
17 alarmCategory 报警类别  
18 recoverTagValue 恢复值  
19 repeatCount 重复次数  
20 alarmArea1 报警区域1  
21 alarmArea2 报警区域2  
22 alarmArea3 报警区域3  
23 alarmArea4 报警区域4  
24 alarmArea5 报警区域5  
25 alarmArea6 报警区域6

示例：

```
sys.exportDataOfHistoricalAlarms({
  filetype: "csv",
  colName: {
    "daName": "站点名"
  },
  amtype: 0,
  time: "2017-01-01 00:00:00|2017-02-01 00:00:00"
});
```

#### 4.6.1.5 exportDataOfRealtimeAlarms(options)

使用脚本导出实时报警数据，保存为 cvs 或者 tsv 格式。

- 参数
  - filetype 选填，字符串类型，用于导出的文件格式判断说明，支持的文件格式包括使用逗号分隔值("csv")和制表符分隔值("tsv")，默认值为 "csv"。
  - colModel 选填，数组，需要显示的报警数据字段。默认值为全部报警数据字段。
  - colName 选填，对象，它的键名为报警数据字段名，键值为打算替换的字段名。导出数据的列名顺序按照 colModel 数组中的字段顺序显示，与 colName 无关。
  - 其余过滤参数与 alarm.setFilter() 接口一致。
- 返回值：无返回值
- 引入版本：2.3

示例

```
sys.exportDataOfRealtimeAlarms({
  filetype: "csv",
  colName: {
    "daName": "站点名"
  },
  enableFilterPriority: 0,
  filterPrioritySetting: "1|3"
});
```

#### 4.6.1.6 GetAlarm(name)

获取画面上的报警控件。获取到报警控件后可以进一步调用报警接口。

- 参数：
  - name 字符串类型，报警控件的名字，参见报警控件的基本属性设置
- 返回值：报警控件对象
- 引入版本：2.0

示例：

```
var alarm = sys.GetAlarm('alarmWidget24321890');
```

#### 4.6.1.7 getAlarmStatistics(callback)

获取未确认、未恢复的报警总数。该接口统计所有报警记录，与是否存在报警控件、报警控件显示的分页条数无关。

- 参数：
  - callback 函数类型，回调函数，会接受两个字符串类型的参数，第一个参数代表未恢复的统计总数，第二个参数代表未确认的统计总数。使用请参看示例。
- 引入版本：2.2

示例：

```
sys.getAlarmStatistics(function(unrestored, unconfirmed) {  
    // 将未恢复的统计总数显示在名为 Text12345 的文本控件上  
    // 而代表未确认的变量 unconfirmed 没有使用  
    sys.SetAttribute("Text12345", "text", unrestored);  
});
```

#### 4.6.1.8 GetAttribute(objname, attrname)

获取对象属性值。对于脚本对象（矩形、直线、椭圆、多边形、图片）可以设置/获取的属性有：

类型	属性名	描述	类型/形式
位置属性	x	距左边距离	字符串数值
	y	距上边距离	字符串数值
	width	宽度	字符串数值
	height	高度	字符串数值
填充属性	fgclr	填充前景色	十进制/字符串(#ffffff)
	bkclr	填充背景色	十进制/字符串(#ffffff)
	brush	风格画刷	1/2/3...
	fillhorscale	水平填充比例	0-100
	fillverscale	竖直填充比例	0-100
	fill-opacity	背景透明	0/1
线型属性	stroke	线型颜色	十进制/字符串(#ffffff)
	stroke-dasharray	线型风格	1/2/3... (与线型风格下拉选项对应)
	stroke-width	线宽风格	1/2/3... (与线宽风格下拉选项对应)

<b>字体属性</b>	fontclr	字体颜色	十进制/字符串(#ffffff)
<b>其他属性</b>	visibility	是否可见	1/0
	sound	声音	(与声音下拉选项对应)
	rotatedegree	旋转角度	度数
	rotaterate	旋转速度	1/2/3... (与旋转速度下拉选项对应)
<b>文本属性</b>	text	文本框(Text)的内容	字符串
<b>控件</b>	value	输入框(Input)的内容	字符串, 只能获取 按类型设置 var: 值 color: 颜色值, 如#ffffff
<b>图元自定义属性</b>	#+图元自定义属性 如 #显示文本 设置图元自定义属性值		

- 参数：
  - objname 对象名称, 字符串类型
  - attrname 属性名称, 字符串类型
- 返回值：对象属性值, 类型参见上表
- 引入版本：1.8

示例：

```
var a=sys.GetAttribute("Line10","stroke-dasharray"); // 获取名为 "Line10" 的控件
的线形风格
sys.SetAttribute("Line10","stroke-dasharray","6"); // 设置名为 "Line10" 的控件
的线形风格为
var user = sys.GetAttribute("Edit5", "value"); // 对于Edit控件, 可以通过
value获取输入的内容
```

#### 4.6.1.9 GetComboValue(objname)

返回给定的combobox当前选中的值

- 参数：
  - objname combobox对象的对象名
- 返回值：给定的combobox当前选中的值
- 引入版本：1.8

#### 4.6.1.10 GetCurUserAuth(authItem)

获取当前用户对某个授权项的权限（同步函数）

- 参数：
  - authItem 授权项名称
- 返回值：
  - 0 没有该权限或者不可见
  - 1 可见
  - 2 可控
- 引入版本：2.2

示例：

```
var authRight = sys.GetCurUserAuth("resourceC");
```

#### 4.6.1.11 GetCurUserMultiAuth(multiAuthItem)

获取当前用户对某个授权项的权限（同步函数）

- 参数：
  - multiAuthItem 授权项名称数组
- 返回值：授权项名称对应的授权项信息数组
  - 0 没有该权限或者不可见
  - 1 可见
  - 2 可控
- 引入版本：2.6

示例：

```
var multiAuthItem = [ "resourceC", "resourceA" ],
    multiAuth = sys.GetCurUserMultiAuth(multiAuthItem),
    authRightC = multiAuth["resourceC"],
    authRightA = multiAuth["resourceA"];
```

#### 4.6.1.12 GetGalleryObj(objname)

获取图元关联的对象

- 参数：
  - objname 图元名称
- 返回值：图元关联对象的名称，字符串类型
- 引入版本：2.0

示例：

```
sys.GetGalleryObj("obj1");
```

#### 4.6.1.13 GetLocalTag(tagname)

获取本地变量值

- 参数：
  - tagname 本地变量名，字符串
- 返回值：本地变量值
- 引入版本：2.2

示例：

```
var value = sys.GetLocalTag('cc');
alert(value);
```

#### 4.6.1.14 GetPropExpr(objname, prop)

获取图元自定义属性关联的表达式值

- 参数：
  - objname 图元的名称；若在图元编辑器中使用，设为 "SELF"
  - prop 图元的自定义属性名称，需要加 #；如属性名为 "abc"，则输入 "#abc"
- 返回值：当前图元的自定义属性关联的变量表达式值
- 引入版本：1.8

示例：

```
var a=sys.GetPropExpr("SELF", "#abc"); // 在图元脚本中运行态这个图元自定义属性关联的表达式
```

#### 4.6.1.15 GetRadioGroupValue(groupname)

获取一组radiobutton的选中情况

- 参数：
  - groupname radiobutton的名称
- 返回值：当前选中radiobutton的值。如果groupname为空，则返回没有设置分组的radiobutton的选中的值。
- 引入版本：1.8

示例：

```
var a=sys.GetRadioGroupValue("Group13"); // 获取Group13这个组中选中的radio
```

#### 4.6.1.16 GetTransValue(key)

配合 OpenPage 函数使用，从打开页面传值。（OpenPage 第三个参数为传值参数）

- 参数：
  - key 打开页面传入的parameter对象中的键
- 返回值：
- 引入版本：2.0

示例：

```
sys.GetTransValue("test1"); // 参考 OpenPage 脚本中的例子
```

#### 4.6.1.17 GetTrend(name)

获取画面上已经存在的趋势控件。获取到趋势控件后，可以进一步调用趋势接口。

- 参数：
  - name 趋势控件的名称
- 返回值：返回趋势控件的对象

- 引入版本：2.0

示例：

```
var trend = sys.GetTrend('trendWidget28315350');
```

#### 4.6.1.18 GetUserGroup(username)

通过用户名称获取该用户所属群组

- 参数：
  - username: 用户名，字符串类型，可以为空。
- 返回值：用户名所属群组，数组类型。当用户名为空时返回当前登录用户名所属群组。当用户名不存在时返回的是空的数组。
- 引入版本：2.2

示例：

```
var usergroup = sys.GetUserGroup("userA"); // usergroup = ["groupA", "groupB"]
```

#### 4.6.1.19 HandlePage(pagename, oldtags, newtags, pageparam,newWindow)

在主视图中根据不同的参数打开画面

- 参数：
  - pagename 画面名，字符串
  - oldtags 被替换的变量名称，字符串/数组，可以不设
  - newtags 替换的变量名称，字符串/数组，可以不设
  - pageparam 画面参数，字符串
  - newWindow 画面是否在浏览器新窗口打开（参数"1","true","yes"）
- 引入版本：1.8

示例：

```
// 跳转到根目录下aa.html画面  
sys.HandlePage("/aa");  
  
// 跳转到test文件夹下objtest画面  
// 并替换数据源 testds 的 tag67 变量成 testds 的tag99  
sys.HandlePage('/test/objtest', 'testds_tag67', 'testds_tag99');  
  
// 在浏览器新窗口打开目录下aa.html画面，  
// oldtags、newtags、pageparam 无需设置，使用空字符串  
sys.HandlePage("/aa", '', '', '', '1');
```

#### 4.6.1.20 IsChecked(objname)

返回给定的checkbox是否选中

- 参数：
  - objname combobox 对象的对象名

- 返回值：
  - true 选中
  - false 没有选中
- 引入版本：1.8

#### 4.6.1.21 IsRadioChecked(objname, groupname)

返回指定的radiobutton是否被选中

- 参数：
  - objname radio对象的对象名
  - groupname 所属的组名，若没有则可以不写
- 返回值：
  - true 选中
  - false 没有选中
- 引入版本：1.8

示例：

```
var a = sys.IsRadioChecked("RadioButton13");
```

#### 4.6.1.22 Login(name, password)

同步方式登录系统

- 参数：
  - name 用户名 (只需输入用户名，不需要添加租户)
  - password 用户密码
- 返回值：
  - 0=登录成功；
  - 其他=登录失败
- 引入版本：1.8

示例：

```
sys.Login("admin","admin");
```

#### 4.6.1.23 LoginAsync(name, password, nextPage)

异步方式登录系统，并可以指定登录成功后跳转的画面

- 参数：
  - name 用户名 (只需输入用户名，不需要添加租户)
  - password 用户密码
  - nextPage 登录成功后跳转画面，需要全路径。例如 "/a/b/c"

- 引入版本：1.8

示例：

```
sys.LoginAsync("admin","admin","/a/b/c");
```

#### 4.6.1.24 OpenNewTrend(tagName, objectOptions)

打开趋势图

- 参数：
  - tagName 要查看的变量点//2.3版本后可以设置成null,点信息在objectoptions中设置
  - objectOptions 新的趋势图的配置项，对象类型。新的趋势图的配置项包含以下参数：
    - openMode 新趋势的打开方式，0 表示新窗口模式，1 表示新页面模式
    - width 打开新窗口的宽度 **新窗口下有效**
    - height 打开新窗口的高度 **新窗口下有效**
    - position 新窗口在画面上显示的位置，例：[100,200] **新窗口下有效**
    - bModal 打开的新窗口是模态还是非模态窗口 true 表示模态，false 表示非模态 **新窗口下有效**
    - title 新的趋势控件的标题 字符串
    - backgroundColor 窗口背景色 设置好以后，当趋势以窗口模式打开以后，窗口和趋势呈现出设置的颜色
    - type 趋势类型 0表示实时趋势，1表示历史趋势
    - lines 新的趋势控件的配置的曲线数组包括5个参数：linename, tagname, linecolor, linewidth, linestyle （丢弃）
    - YAxes 趋势控件要添加的Y轴数组，每个数组包括参数有Y轴的title, tcolor,linewidth,和阈值数组thresholds,每个阈值但对象包括参数：name,color, style, width,只有该条Y轴被设置了对应曲线，该条Y轴才显示，否则不显示。
    - xAxisTimeSpan 新的趋势控件的X轴坐标范围,包括2个参数：timenumber, timeunit
    - groups 存储趋势组的信息包括组的基本信息和趋势曲线点的信息包括：
      - name 组名
      - desc 组描述 中间不要有空格
      - lines 趋势曲线数组，可配置的参数包括：
        - linename 曲线名称,
        - tagname 点名,
        - linecolor 曲线颜色,
        - linewidth 曲线宽度,
        - linestyle 曲线风格,
        - max 是否显示最大值,
        - min 是否显示最小值,
        - average 是否显示平均值,
        - difference 是否显示差值,
        - valuetype 显示原始值还是插值， 0表示插值 1表示原始值,
        - markertype 0表示没有marker 1表示是三角形 2是反三角形3是正方形4是圆形,
        - yAxis 此条曲线对应的Y轴
        - decimal 设置小数点的位数
        - statsvalueshowtype 统计值显示类型0表示曲线方式，1表示

文本提示方式

- savetolocal 运行态添加的曲线是否保存到本地
  - isshow 曲线是否显示0不显示 1显示
  - tags 趋势点的信息数组每个点的信息也是一个数组，包括：点名，曲线名，值类型（是否用原始值生成曲线）
  - selgroup 当前趋势组
  - tags 当前选中的趋势组下的点数组
- 引入版本：2.0

示例：

```
// 新窗口模式：
Text25419436_OnClick = function(){
    var objectOptions = {
        openMode:0,
        width:800,
        height:600,
        position:[100,200],
        bModal:true,
        title:'testtrend',
        type:'0', //实时或者历史趋势 如果是1表示历史还需要设置开始和结束时间
        starttime:'',
        endtime:'',
        groups:[{
            name:'group',
            desc:'description', //中间不能有空格
            lines:[{
                linename:'line1',
                tagname:'testds.tag1000',
                linecolor:'#5E78F6',
                linewidth:1,
                linestyle:'Solid',
                max:'1',
                decimal:'1',
                showvalue:'0',
                statsvalueshowtype:'1',
                yAxis:'testYAxis1'
            },
            {
                linename: 'line2',
                tagname: 'testds.tag1003',
            }
        ]],
        tags:[['testds.tag1000', 'line1'],['testds.tag1003', 'line2']],
        selgroup:'group',
        YAxes:[{
            title:'test',
            tcolor:'#22f768',
            linewidth:'10'
        }],
    },
    {
```

```

        title: 'testYAxis1',
        tcolor: '#ffee99',
        linewidth: '5',
        thresholds: [{
            name: 'ht1',
            value: 100
        }]
    },
    {
        title: 'testYAxis12',
        tcolor: '#cce99',
        linewidth: '5'
    }
}],
xAxisTimeSpan: {
    timenumber: 20,
    timeunit: 'minute'
}
};
sys.OpenNewTrend(null, objectOptions);
};

// 新页面模式：
Text12260030_OnClick = function(){
    var objectOptions = {
        openMode: 1,
        title: 'testtrend'
    };
    sys.OpenNewTrend('testds.tag100', objectOptions);
};

```

注：新页面下打开趋势图，不支持浏览器刷新页面。

#### 4.6.1.25 OpenPage(pagename, options, parameter)

通过输入不同参数以不同方式打开画面，支持页面传值

- 参数：
  - pagename 画面名，字符串
  - options 属性对象(可选)，例如 {modal:"3"}
    - modal 模式（数字），可以有四种方式
      - 0 页面跳转
      - 1 新页面跳转
      - 2 弹出非模态画面
      - 3 弹出模态画面
    - width 宽度（字符串）
    - height 高度（字符串）
    - position 位置，可以有三种方式
      - 1)窗口中间 center

- 2)left center
      - 3)[left,top] (字符串)
    - dispTitle 是否显示标题(可选) (字符串)
    - title 标题(可选) (字符串)
    - oldtags 被替换的变量名(可选) (字符串)
    - newtags 替换的变量名(可选) (字符串)
    - pageID 用于关闭画面ID(可选) (字符串)
    - dispFrame 是否显示边框(可选) (布尔类型)
    - windowName 打开新窗口的名字(可选,只在modal=1的情况下起作用) (字符串)
  - parameter 传值对象(可选, 字符串, 获取传值可在 sys.GetTransValue 中获取), 例如 {test1:"3"}
- 引入版本: 2.0

示例:

```
var options = {
  modal:1,
  oldtags:"",
  newtags:"",
  width:"500",
  height:"500",
  position:"center"
};

var parameter = {
  test:"1",
  test2:"2"
};

sys.OpenPage("/test2", options, parameter);
```

打开模态窗口, 并且要在打开窗口内关闭自身时, 必须要在被打开父类画面的脚本中加入一个函数, 如下所示:

```
function closePage(id){
  $("#" + id).remove();
  $("#" + id).dialog("destroy");
}
```

#### 4.6.1.26 PauseAnimation(objName, animName)

暂停动画

- 参数:
  - objName 对象名称
  - animName 动画名称
- 引入版本: 1.9

#### 4.6.1.27 PlayAnimation(objName, animName)

## 播放动画

- 参数：
  - objName 对象名称
  - animName 动画名称
- 引入版本：1.9

### 4.6.1.28 ReadMultiValues(variableArray)

读取多个变量的相应域值，目前只能获取已经注册的点的值，最多可以一次性获取32个点。

- 参数：
  - variableArray 字符串数组，其中的每个元素是 数据源.点名 或者 数据源.点名.域名 形式的字符串
- 返回值：对象，可以是散列表，键为 variableArray 中注册的点名，值为获取到的对应的值
- 引入版本：2.0

示例：

```
var a = sys.ReadMultiValues([
    "testds.tag1.value",
    "testds.tag2.A_DESC"
]);

console.log(a);
// a = {
//   testds.tag1.value:"1",
//   testds.tag2.A_DESC:"设备"
// }

console.log(a["testds.tag2.A_DESC"]); // "设备"
// 请勿使用 a.testds.tag2.A_DESC 的方式获取值
```

### 4.6.1.29 ReadMultiValuesAsync(variableArray, callback)

ReadMultiValues 接口的异步版本。

- 参数：
  - variableArray 字符串数组，其中的每个元素是 数据源.点名 或者 数据源.点名.域名 形式的字符串
  - callback 回调函数，用于取完值后调用
- 返回值：无返回值
- 引入版本：2.3

示例

```
sys.ReadMultiValuesAsync(["testds.tag1.value", "testds.tag2.A_DESC"],
callback(values) {
    console.log(values["testds.tag2.A_DESC"]);
});
```

#### 4.6.1.30 ReadValue(tagname)

读取变量的相应域值，目前只能获取已经注册的点的值。

- 参数：
  - tagname 字符串类型，点名，必须是 数据源.点名 或者 数据源.点名.域名 的形式
- 返回值：返回读取到的域值
- 引入版本：1.8

#### 4.6.1.31 SendUserAppInvoke(userPath, options, callback)

调用用户应用程序

- 参数：
  - userPath 所要调用的用户应用的path信息 http://host:port/{userPath}
  - options 调用参数[可选]
    - method GET/POST 默认GET
    - async 是否异步 默认false；如果是异步，需要提供callback函数
    - data: 【可选】post的数据
  - callback 异步调用的回调函数[可选]
- 返回值：
- 引入版本：2.2

示例：

```
var options = {
    method:"POST",
    async:true,
    data:"abcdef"
};

sys.SendUserAppInvoke("testweb/userService", options, function(msg) {
    console.log("async return data = " + msg);
});
```

#### 4.6.1.32 SetAttribute(objname, attrname, value)

设置对象的属性值

- 参数：
  - objname 对象名称
  - attrname 属性名称
  - value 值

- 引入版本：1.8

#### 4.6.1.33 SetLayerVisible(layer, visible)

设置图层的显示隐藏

- 参数：
  - layer 图层名称
  - visible 是否显示图层，true 显示图层，否则隐藏
- 引入版本：1.9

#### 4.6.1.34 SetLocalTag(tagname, value)

给本地变量设置值

- 参数：
  - tagname 本地变量名，字符串。注：本地变量名与其他变量名不能重名。
  - value 值
- 引入版本：2.2

示例:

```
sys.SetLocalTag('cc',10); //给本地变量cc设置值10
```

#### 4.6.1.35 ShowModalPage(pagename, width, height, position, dispTitle, title, oldtags, newtag, pageid)

按模态对话框方式打开画面

- 参数：
  - pagename 画面名称
  - width / height 打开画面的宽高
  - position 指定打开画面的位置，可以有三种方式。
    - 1. 窗口中间 'center' ;
    - 2) "left center";
    - 3)[left,top]
  - dispTitle 是否不显示标题（可选）
  - title 标题（可选）
  - oldtags / newtags 变量替换，各为数组，一对一对应（可选）
  - pageid 用于关闭画面ID（可选）
- 返回值：打开对话框的ID
- 引入版本：1.8

#### 4.6.1.36 ShowMsgBox(type, title, content, buttons, functions)

消息框、确认框

- 参数

- type messagebox类型, 包括 "info" 和 "confirm" 两种
- title messagebox标题
- content messagebox内容
- buttons messagebox按钮数组
- functions buttons数组触发的事件, 分别对应每个buttons

- 引入版本: 2.2

示例:

```
var buttons = [
    { value: "Yes" },
    { value: "No" },
    { value: "Cancel" }
],
func1 = function() {
    console.log("foo");
},
func2 = function() {
    console.log("bar");
},
functions = [ func1, "", func2];
sys.ShowMsgBox("confirm", "Msgbox Title", "msgbox content", buttons,
functions);
```

#### 4.6.1.37 ShowPage(pagename, width, height, position, dispTitle, title, oldtags, newtags, pageid)

以对话框方式打开画面

- 参数:
  - pagename 页面名称
  - width / height 打开页面的宽高
  - position 指定打开画面的位置, 可以有三种方式。
    1. 窗口中间 'center';
    2. 'left center';
    3. 数组 [left,top], 例如 [100, 100]
  - dispTitle 是否显示标题 (可选)
  - title 标题 (可选)
  - oldtags / newtags 变量替换, 各为数组, 一对一对应 (可选)
  - pageid 用于关闭画面ID (可选)
- 返回值: 打开对话框的ID
- 引入版本: 1.8

#### 4.6.1.38 WriteValue(tagname, tagvalue)

向指定的Tag点写值

- 参数:
  - tagname 要写入值的tag点
  - tagvalue 新写入的值

- 引入版本：1.8

## 4.6.2 趋势脚本函数

趋势控件目前提供一些脚本函数供用户调用，这些函数包括曲线的添加，删除，修改和获取。

### 4.6.2.1 AddCurves(group, lines)

在已经存在的趋势组下面添加曲线

- 参数：
  - group 趋势组名，字符串
  - lines 要添加的曲线对象的数组，数组
- 引入版本：2.0

脚本示例:

```
Button1255490_OnClick = function(){
    var trend = sys.GetTrend('trendWidget28315350');
    var group = 'trendgroup';
    var lines = [
        {
            linenoame: 'testline',
            tagname: 'testds.tag1000',
        },
        {
            linenoame: 'testline1',
            tagname: 'testds.tag1002',
            color: '#d35400',
            width: 1,
            style: 'ShortDash',
            yAxis: 'testYAxis1'
        }
    ];
    trend.AddCurves(group, lines);
};
```

说明：

- trendgroup 为趋势组名称
- lines 为要添加的曲线对象数组
- line 对象可以配置5个属性：
  - linenoame 曲线名称
  - tagname 曲线对应的变量名称
  - color 曲线颜色 可以不配
  - width 曲线宽度 可以不配 只能是数字 例如width:1, 不支持'1'字符串
  - style 曲线类型包括五种 (Solid, ShortDash, ShortDot, ShortDashDot, ShortDashDotDot) 可以不配
  - yAxis 该条曲线对应的Y轴，如果配置的Y轴不存在或者才参数不配置就用默认Y轴。

后面三个属性如果不配置就给出默认的，预配置颜色数组中第一个未使用的颜色 / 1 / Solid。

如果要输入的趋势组不存在，则添加一个名字为trendgroup的新的趋势组，曲线添加在该组下面。

#### 4.6.2.2 GetCurve(group, curvename)

在已经存在的趋势组下面获取已经存在的曲线

- 参数：
  - group 趋势组名，字符串
  - curvename 要获取的曲线名称，字符串
- 引入版本：2.0

示例:

```
Button1255490_OnClick = function(){
    var trend = sys.GetTrend('trendWidget28315350');
    var oldline = trend.GetCurve('trendgroup', 'testline1');
    if(oldline === null){
        alert('trend line is not exist');
    }
};
```

#### 4.6.2.3 ModifyCurve(group, newcurve)

在已经存在的趋势组下面修改已经存在的曲线

- 参数：
  - group 趋势组名，字符串
  - newcurve 修改后的新曲线，曲线对象
- 引入版本：2.0

示例:

```
Button1255490_OnClick = function(){
    var trend = sys.GetTrend('trendWidget28315350');
    var oldline = trend.GetCurve('trendgroup', 'testline1');
    if(oldline === null){
        alert('trend line is not exist');
    }
    oldline.tagname = 'testds.tag204';
    oldline.color = '#FF0000';
    trend.ModifyCurve('trendgroup', oldline);
};
```

说明：

- 要想修改曲线，必须先获取到这条曲线，也就是调用GetCurve方法。
- 目前只支持修改曲线的color, width, style三个属性
- 颜色属性只能是'#FF0000'格式，不支持直接color='black'格式

#### 4.6.2.4 DeleteCurve(group, curvename)

在已经存在的趋势组下面删除已经存在的曲线

- 参数：
  - group 趋势组名，字符串
  - curvename 要删除的曲线名称，字符串
- 引入版本：2.0

示例:

```
Button1255490_OnClick = function() {  
    var trend = sys.GetTrend('trendWidget28315350');  
    trend.DeleteCurve('trendgroup', 'testline1');  
};
```

如果需要在 page\\_onload 的方法里面调用添加，删除，修改，获取等方法，需要等到趋势加载完成以后再调用。

示例:

```
page_OnLoad = function() {  
    var trend = sys.GetTrend('trendWidget28315350');  
    if(trend === null) {  
        alert('get trend failed');  
    }  
    //如果需要调用AddCurves方法，需要在trend的onload函数里面调用  
    trend.onload = function() {  
        var group = 'testgroup';  
        var lines = [  
            {  
                name: 'testline',  
                tagname: 'testds.tag1000',  
            },  
            {  
                name: 'testline1',  
                tagname: 'testds.tag1002',  
                color: 'd35400',  
                width: 1,  
                style: 'ShortDash'  
            }  
        ];  
        trend.AddCurves(group, lines);  
    };  
};
```

#### 4.6.2.5 SetYAxisRange(id, max,min)

设置Y轴的范围

- 参数：
  - id 想要设置的Y轴id
  - max Y轴的最大量程

- min Y轴的最小量程
- 引入版本: 2.5

示例:

```
Button20410696_OnClick = function(){
    var trend = sys.GetTrend('trendWidget28315350');
    trend.SetYAxisRange('tempeture', 500, 100');
};
```

说明: max 和 min 也可以写成null, 表示不设置

### 4.6.3 报警控件脚本函数

报警控件目前包含设置过滤条件和确认报警两个接口, 使用前请务必先调用 `sys.GetAlarm()` 函数获取报警控件。

#### 4.6.3.1 confirmAlarm(gridNo, callback, error)

通过脚本确认报警。

**请注意**, `confirmAlarm()` 函数不同于 `confirm()` 函数, 后者是浏览器内置的函数。

- 参数:
  - `gridNo` 数值类型, 待确认的报警位于表格中的编号
  - `callback` 函数类型, 可选参数, 用于确认报警成功后执行 (即异步回调函数)。`callback` 函数可以声明一个参数, 其在被执行的时候, 会被传递确认报警接口的数据。
  - `error` 函数类型, 可选参数, 用于确认失败时调用, 接受三个参数 `xhr`, `status` 和 `err`。该参数的引入版本 2.3。
- 引入版本: 2.0

示例:

```
var alarm = sys.GetAlarm('alarmWidget24321890');

// 不传递 callback
alarm.confirmAlarm(0);

// 传递 callback 函数, 但 callback 函数不包含参数
alarm.confirmAlarm(0, function() {
    console.log("confirmed");
});

// 传递 callback 函数, 并包含参数
alarm.confirmAlarm(0, function(APIData) {
    // callback 函数最多一个参数, 超出的参数不生效
    console.log(APIData);
});
```

#### 4.6.3.2 confirmAllAlarms(callback, error)

确认当前报警页的所有报警。

- 参数：
  - callback 函数类型，可选参数，用于确认报警成功后执行（即异步回调函数）。callback 函数可以声明一个参数，其在被执行的时候，会被传递确认报警接口的数据。
  - error 函数类型，可选参数，用于确认失败时调用，接受三个参数 xhr, status 和 err。该参数的引入版本 2.3。
- 引入版本：2.2

#### 4.6.3.3 confirmSelectedAlarms(callback, error)

确认当前报警页所有选中的报警。

- 参数：
  - callback 函数类型，可选参数，用于确认报警成功后执行（即异步回调函数）。callback 函数可以声明一个参数，其在被执行的时候，会被传递确认报警接口的数据。
  - error 函数类型，可选参数，用于确认失败时调用，接受三个参数 xhr, status 和 err。该参数的引入版本 2.3。
- 引入版本：2.2

#### 4.6.3.4 setFilter(options)

设置报警过滤条件

- 参数：
  - options 对象类型，可以包含如下属性
- 引入版本：2.0（部分属性在 2.0 之后的版本添加，参见具体说明）

属性名	数据类型	说明
enableFilterArea	number	0 或 1, 0 表示不启用报警区域过滤, 1 表示启用
enableFilterStatus	number	0 或 1, 0 表示不启用报警状态过滤, 1 表示启用
enableFilterTime	number	0 或 1, 0 表示不启用报警时间过滤, 1 表示启用
enableFilterPriority	number	0 或 1, 0 表示不启用报警优先级过滤, 1 表示启用
enableFilterTag	number	0 或 1, 0 表示不启用变量名称过滤, 1 表示启用
enableFilterType	number	0 或 1, 0 表示不启用报警类型过滤, 1 表示启用
enableFilterAgent	number	0 或 1, 0 表示不启用数据代理过滤, 1 表示启用 (引入版本: 2.2)
enableFilterDesc	number	0 或 1, 0 表示不启用报警点描述过滤, 1 表示启用 (引入版本: 2.2)
enableFilterSubsysname	number	0 或 1, 0 表示不启用子系统过滤, 1 表示启用 (引入版本: 2.2)
filterAreaSetting	array	报警区域过滤设置。输入数组, 每个元素为字符串。每个字符串对应一个报警区域名称。使用详情参见 4.4.2 章节。
filterStatusSetting	string	报警状态过滤设置。输入 "x y z" 格式的字符串, x y z 分别代表“未确认未恢复”“已确认未恢复”“未确认已恢复”, 可以分别设置为

filterTimeSetting	string	0 或者 1。例如配置为 "1 0 1" 表示过滤出所有未确认的报警。报警时间过滤设置。输入 "t1 t2" 格式的字符串，t1 t2 分别代表起止时间，时间格式为 yyyy-MM-dd hh:mm:ss。示例配置："2016-10-04 00:00:00 2016-11-08 00:00:00"。
filterPrioritySetting	string	报警优先级过滤设置。输入 "p1 p2" 格式的字符串，p1 p2 分别代表最小优先级和最大优先级（0~255 之间的整数）。示例配置："10 20"。
filterTagSetting	string	变量名称过滤，输入变量名称表达式字符串。详情参见 4.4.2 章节。
filterTypeSetting	string	报警类型过滤，输入包含一个或者多个类型的字符串，多个类型间使用竖线   分隔。示例配置："低低 偏差 0->1"。
filterAgentSetting	string	数据代理过滤，输入数据代理表达式字符串。详情参见 4.4.2 章节。（引入版本：2.2）
filterDescSetting	string	报警点描述过滤，输入报警点描述表达式字符串。详情参见 4.4.2 章节。（引入版本：2.2）
filterSubsystemSetting	string	子系统过滤，输入子系统表达式字符串。详情参见 4.4.2 章节。（引入版本：2.2）

options 对象无需提供每项的配置。调用时没有设定的属性将按照上一次的设定执行（不区分界面、脚本设定）。

示例：

```
var alarm = sys.GetAlarm('alarmWidget24321890');
alarm.setFilter({
    enableFilterArea: 1,
    filterAreaSetting: [ "正门", "办公室", "走廊" ]
});
```

#### 4.6.3.5 setGlobalMute(value)

设置全局静音

- 参数：
  - value 布尔类型，true 表示全局静音，false 表示取消静音
- 引入版本：2.2

#### 4.6.3.6 setPage(num)

设置翻页

- 参数：
  - num 整数类型，需要跳转的页面序号，从 1 开始索引。
- 引入版本：2.2

#### 4.6.3.7 setPriorityMute(priority, value)

设置优先级是否静音

- 参数：

- priority 优先级, "default" 字符串或者报警级别的整型数值
- value 布尔值, true 表示该级别静音, false 表示取消静音

- 引入版本: 2.2

#### 4.6.4 调试相关的脚本

可以使用 `console.mvError`、`.mvLog`、`.mvWarn` 函数记录日志, 运行态使用 `Alt + Q` 快捷键或者通过控制条打开日志系统, 详细参照 6.5 章节 [前端消息查看](#)。这三个函数分别是 `console.error`、`console.log` 和 `console.warn` 的替代, 浏览器控制台会输出对应级别的日志。

##### 4.6.4.1 console.mvError(str)

新建一条错误日志。

- 参数:
  - str 日志内容
- 引入版本: 1.9

示例:

```
console.mvError("error...");
```

##### 4.6.4.2 console.mvLog(str)

新建一条日志。

- 参数:
  - str 日志内容
- 引入版本: 1.9

示例:

```
console.mvLog("log...");
```

##### 4.6.4.3 console.mvWarn(str)

新建一条警告日志。

- 参数:
  - str 日志内容
- 引入版本: 1.9

示例:

```
console.mvWarn("warn...");
```

##### 4.6.4.4 errorMsgCallback(src, errMsg, errCode, options)

错误消息回调函数, 在画面脚本或者全局脚本中定义, 趋势控件和报警控件可以使用该函数处理错误消息。

注意：报警控件中需要配置启用。

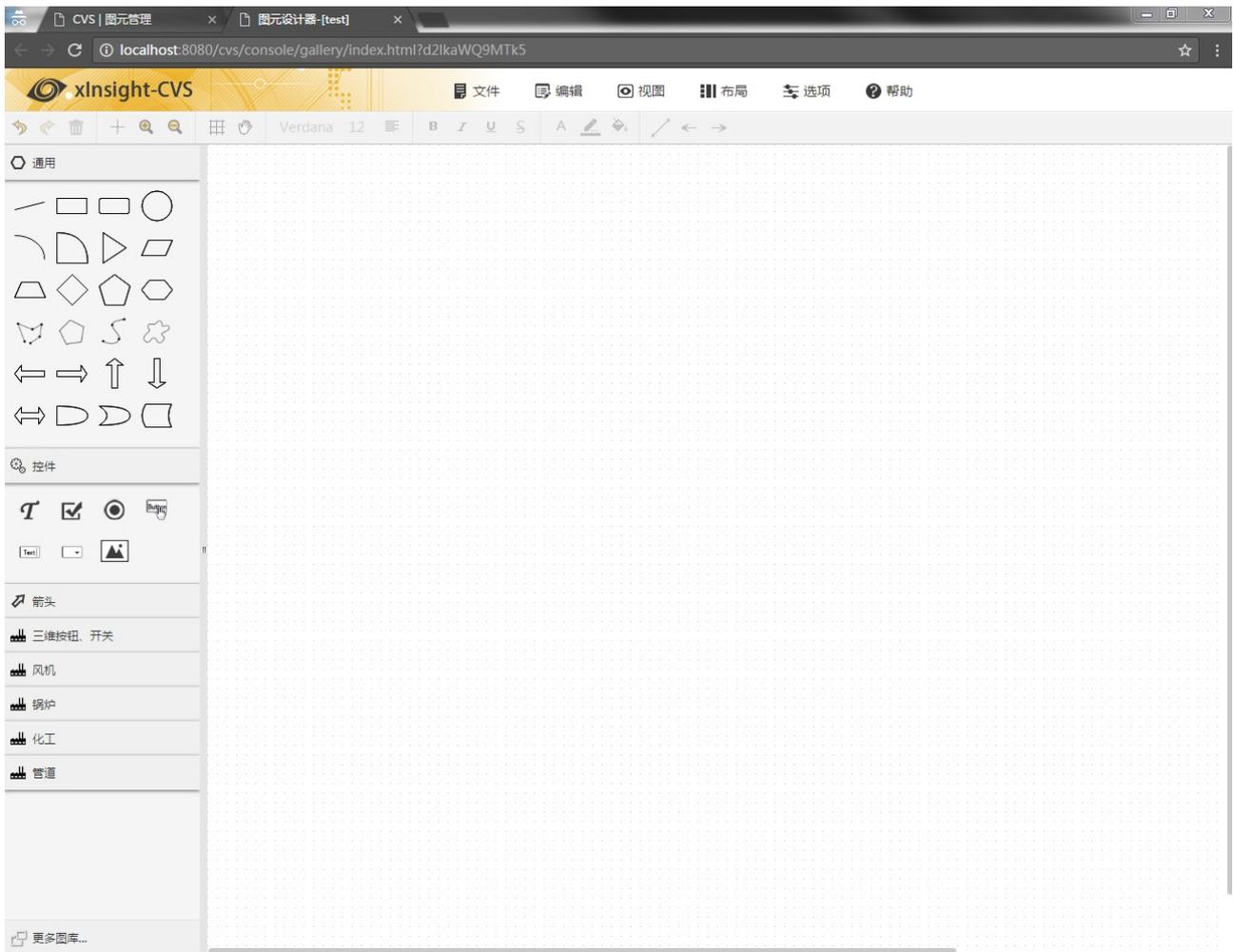
- 参数：
  - src 出错的消息来源，目前仅支持"trend"、"alarm" 两种
  - errMsg 需要显示的错误消息内容
  - errorCode 需要显示的错误消息的错误码
  - options 可选参数，对象类型
    - level 消息级别，字符串类型，可以更细致地处理日志
- 引入版本：2.5.1
- 示例（错误消息alert展示）：

```
function errorMsgCallback(src, errMsg, errorCode, options)
{
    if (src == "trend") {
        alert("用户自定义消息："+errMsg);
    } else if (src == "alarm") {
        if (options.level == "info") {
            // 不做处理
        } else {
            console.mvLog("报警控件自定义消息："+errMsg)
        }
    }
}
}
```

## 5. 图元设计

### 5.1 图元编辑

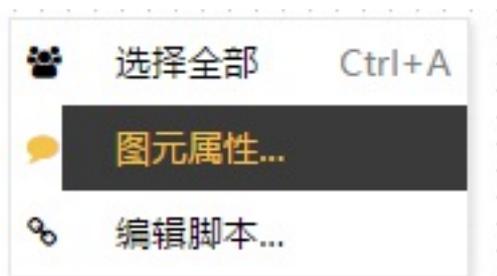
在图元管理页面双击图元或者选中图元点击编辑就会打开相应的图元，进行编辑。



在打开的图元编辑器中左侧列中的控件使用同画面设计器中的使用。具体元素的使用和属性配置请参照第四章内容。

## 5.2 图元属性

在图元的画布空白处双击画布，弹出对话框，点击图元属性，打开图元属性配置界面





### 5.2.1 模板及属性

**类模板**：点击选择模板，弹出模板选择界面。选择图元需要关联的模板。

选择模板 ✕

数据源 xinsight

name	desc
▶ template1	模版1
▼ coking	
E2101_D_day	
E3301_D_day	
E4001_D_day	
E4001_R_second	
E3301_R_second	
E2101_R_second	
▶ wood	
▶ sintering	
▶ raw_material	

xinsight\_coking

确认

点击确认，在模板属性tab页可以看到选中的模板，以及该模板对应的属性。



类模板

属性：

- E2101\_R\_second
- E3301\_R\_second
- E4001\_R\_second
- E4001\_D\_day
- E3301\_D\_day
- E2101\_D\_day

**属性：**模板对应的属性。在选择模板tab页选择了模板后，点击确定按钮，模板的属性就会显示在属性中。只读。

### 5.2.2 自定义属性



自定义属性

属性：

属性名称	类型	默认值	描述
------	----	-----	----

在自定义属性tab页中包括新增和删除自定义属性以及自定义属性的列表。

## 新增自定义属性

点击加号，弹出新增自定义属性对话框



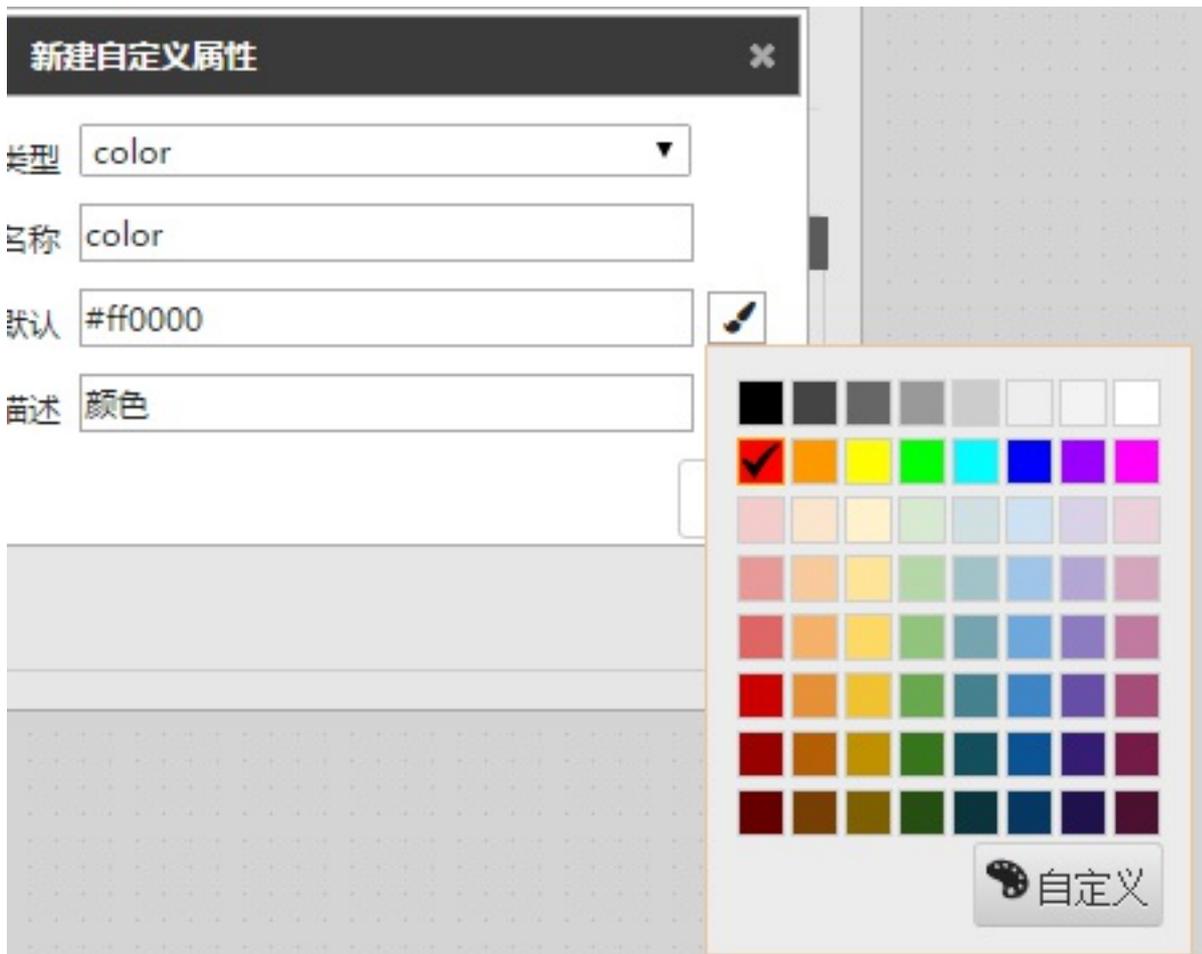
The screenshot shows a dialog box titled '新建自定义属性' (New Custom Attribute) with a close button (X) in the top right corner. It contains four input fields: '类型' (Type) is a dropdown menu with 'var' selected; '名称' (Name) is an empty text box; '默认' (Default) is an empty text box; and '描述' (Description) is an empty text box. A '确认' (Confirm) button is located at the bottom right.

- 类型：自定义属性共三种，包括var、tag和color
- 名称：自定义属性名称。名称不能为空。
- 默认：自定义属性的默认值。
- 当自定义属性为tag时，可以选点



The screenshot shows the same dialog box, but with 'tag' selected in the '类型' (Type) dropdown. The '名称' (Name) field now contains the text 'tag'. A '选择点' (Select Point) button is positioned to the right of the '默认' (Default) field. The '确认' (Confirm) button remains at the bottom right.

- 当自定义属性为color时，可以选择颜色



### 删除自定义属性

在自定义属性tab页选中需要删除的自定义属性，点击删除自定义属性按钮即可



## 自定义属性列表

所有的自定义属性都会显示在列表中。

属性名称	类型	默认值	描述
tag	tag	xinsight_test	
color	color	#ff0000	颜色
测试	var	测试	测试

### 5.2.3 其他属性

在其他属性中可以配置图元的背景色和显隐闪烁属性。



背景颜色只在图元中有效。

图元大小可以设置图元画布的高度和宽度。

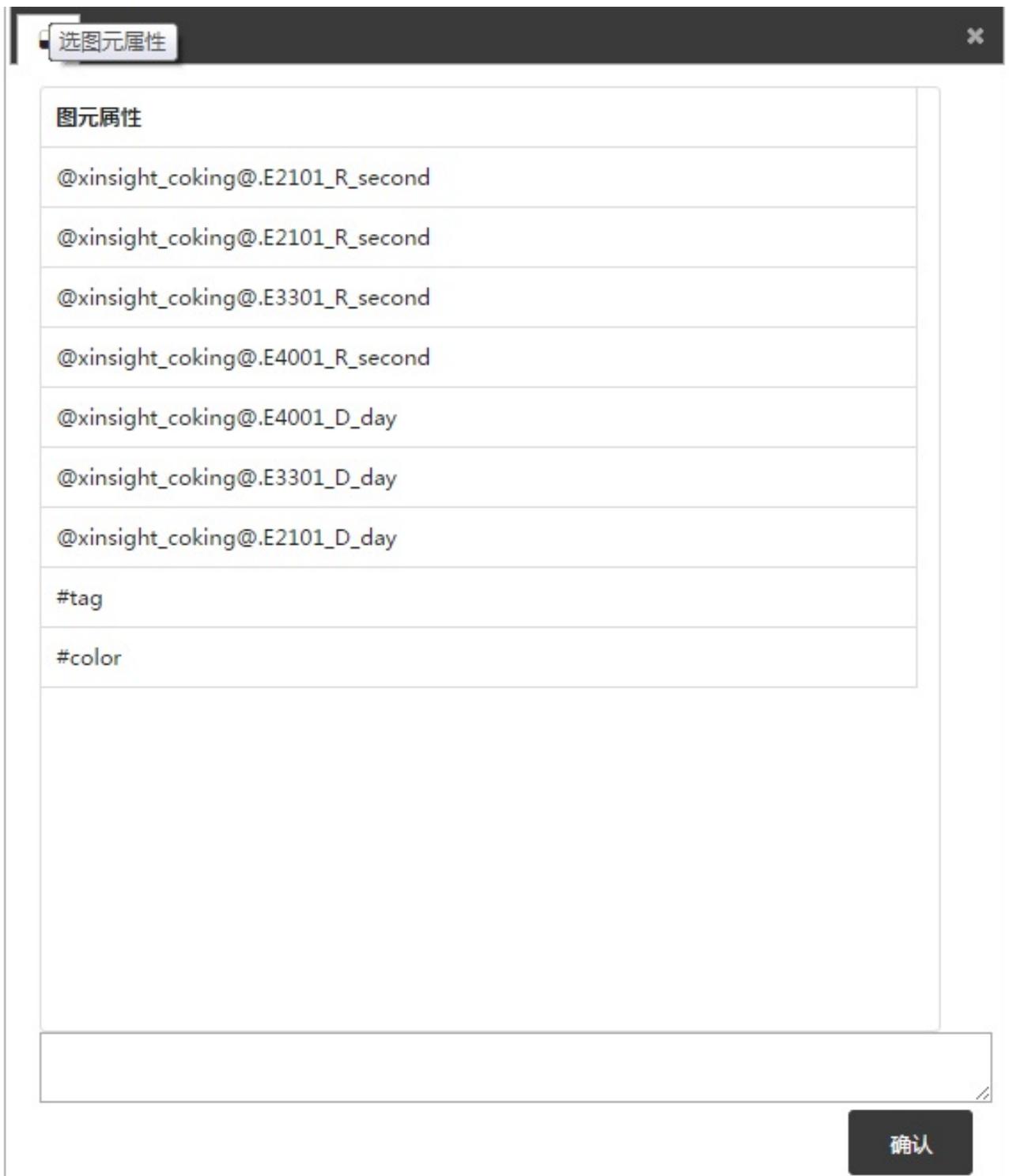
显隐和闪烁的配置和设计器中一致。

拖放到设计器中后，以图元中配置为准，在画面设计器中不能配置。

## 5.3 控件配置

控件使用和设计器中一致。需要注意的是，在控件中关联动画时，可以关联点包括图元属性以及tag点两种类型。

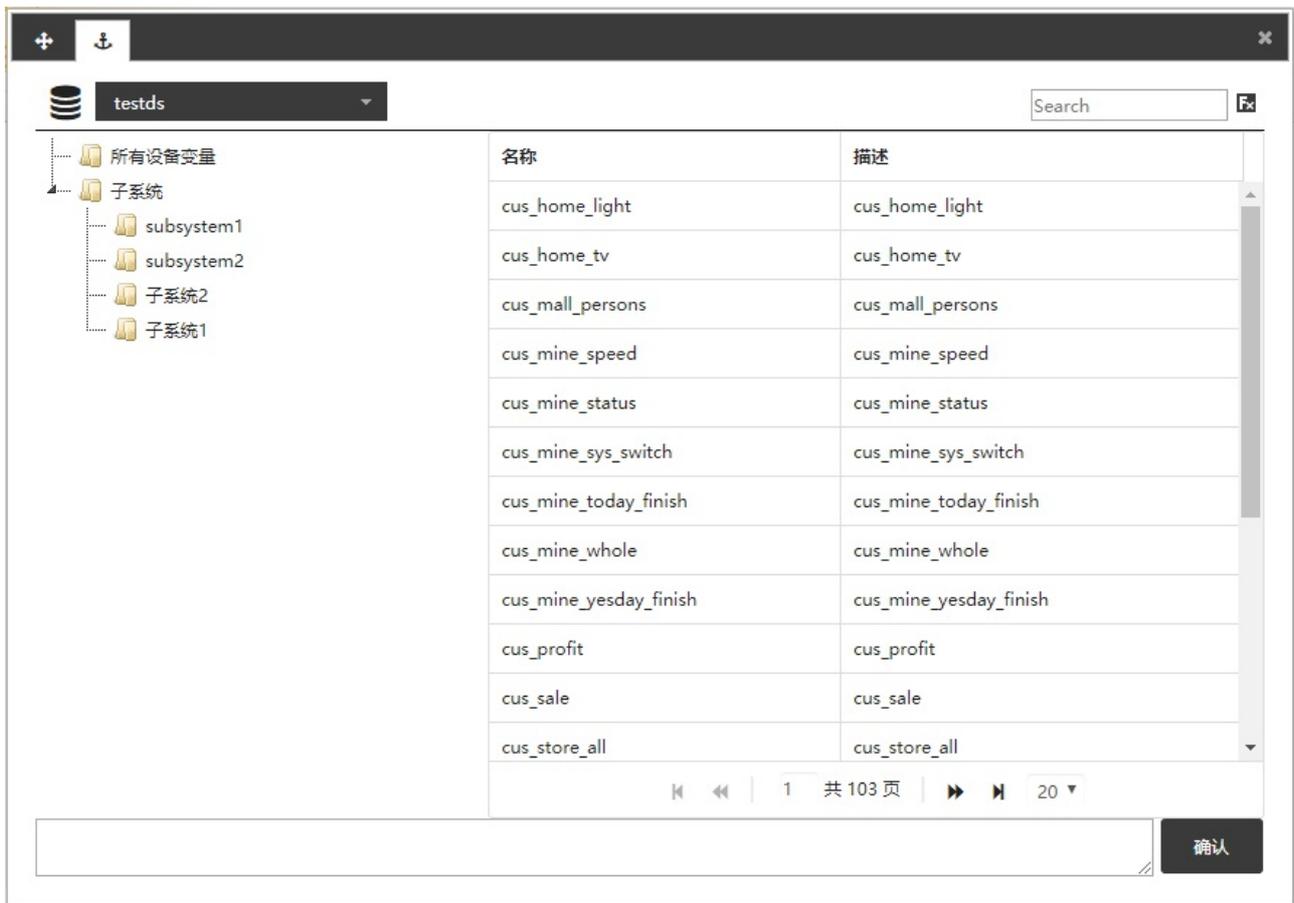
### 5.3.1 选图元属性



选图元属性页面包括图元属性中关联的模板以及自定义属性

### 5.3.2 选择Tag点

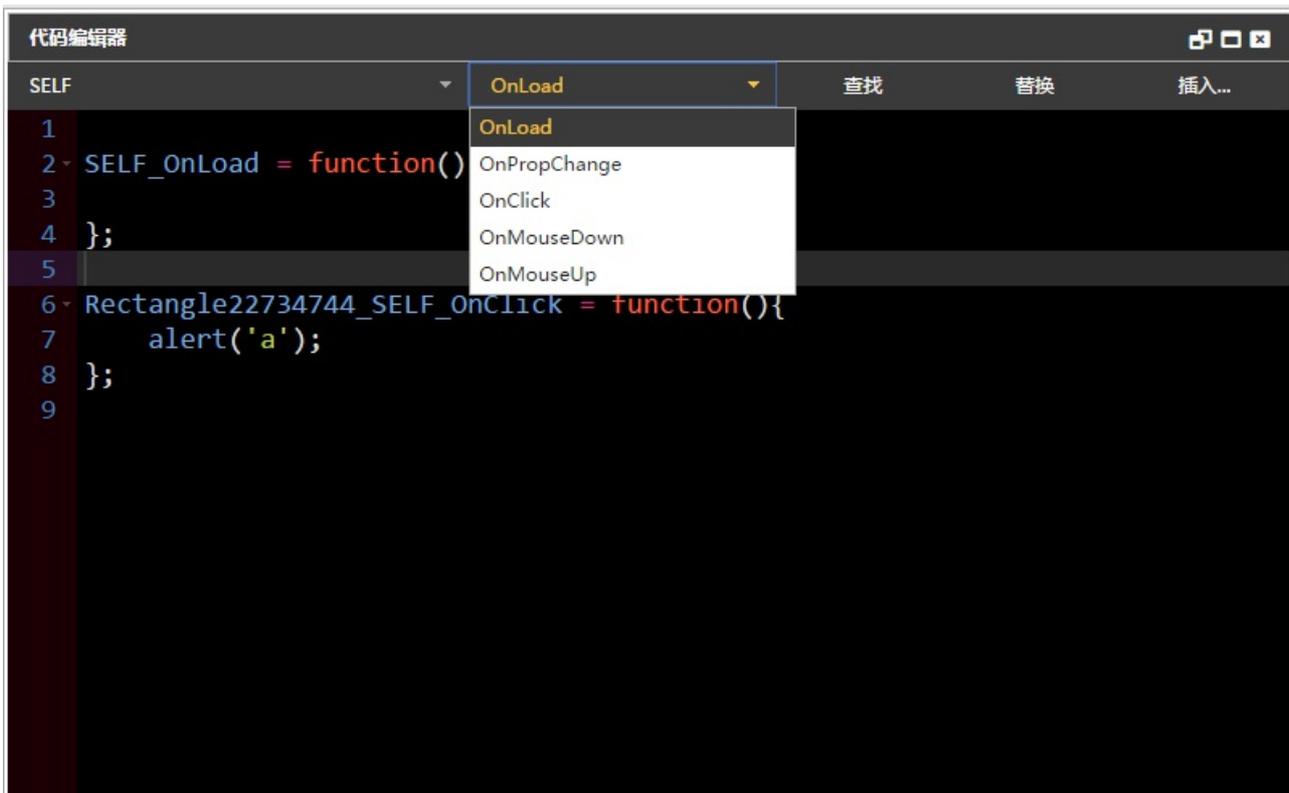
选tag点页面包括数据源的选择以及数据源下tag点的选择。tag点的查询规则和画面设计器中的点查询规则相同（详见4.5.9）。



## 5.4 图元脚本编辑

在画布空白处点击右键，选择编辑脚本就可以对图元进行相应的脚本编辑

### 5.4.1 图元脚本--SELF



图元的事件包括：OnLoad、OnPropChange、OnClick、OnMouseDown、OnMouseUp

- OnLoad：图元加载时调用该方法
- OnPropChange：图元关联的变量(包括图元的自定义属性以及 tag 点等变量)改变时调用该方法
- OnClick：图元点击时触发
- OnMouseDown：鼠标在图元上点下时触发
- OnMouseUp：鼠标从图元上抬起时触发

示例：

```
SELF_OnPropChange = function(evt, prop,value) {  
    if(!prop)  
        return;  
    if(prop == "#open") {  
        console.log('page name ' + value);  
    }  
}
```

#### 5.4.2 图元控件脚本

控件的事件包括：OnDbClick、OnMouseOver、OnClick、OnMouseDown、OnMouseUp

- OnDbClick：双击图元时调用该方法
- OnMouseOver：鼠标在该控件上移动时触
- OnClick：控件被点击时触发
- OnMouseDown：鼠标在控件上点下时触发

SELF 关键字：获取图元在画面设计器中的唯一标识（图元id会在运行态中被唯一标识替换）

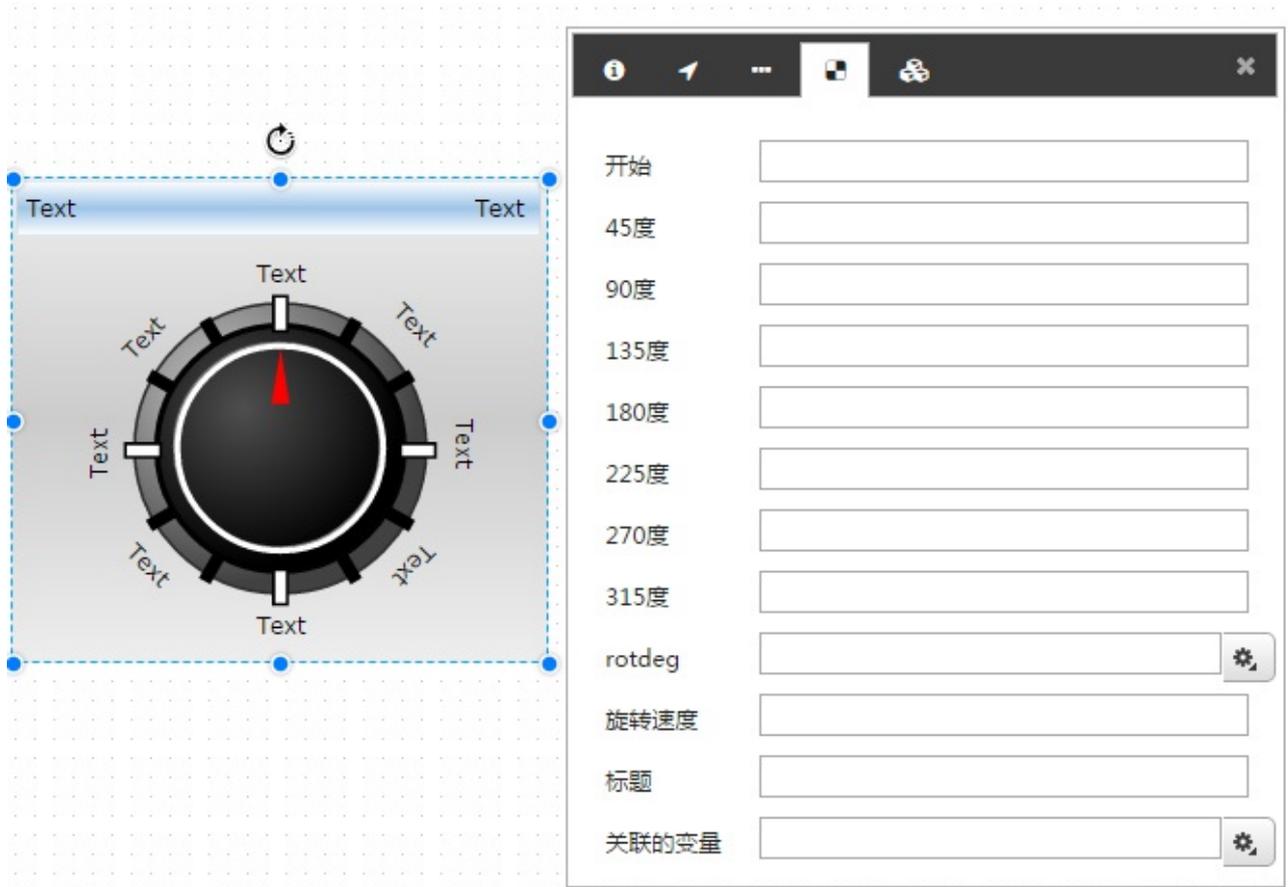
示例：

```
sys.GetAttribute("SELF","x"); // 获取图元在画面设计器中当前X值（获取其他属性与4.9.1画面获取列表一致）
```

## 5.5 系统图元

系统默认会发布一些常用图元，用户可以直接使用或者根据需要改动后使用，也可以打开系统图元文件学习如何制作图元。目前共有17个图元，以下将对主要图元的使用方法进行介绍。

### 5.5.1 仪表盘

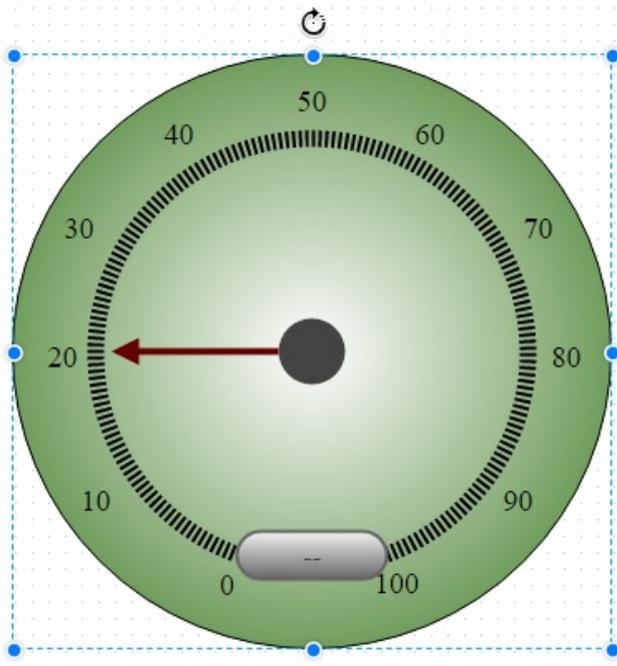


仪表盘图元的功能是根据关联的数据值，仪表盘指针根据当前值指向表盘中的刻度。目前系统共提供三种样式的仪表盘，用户可以根据需要选用。

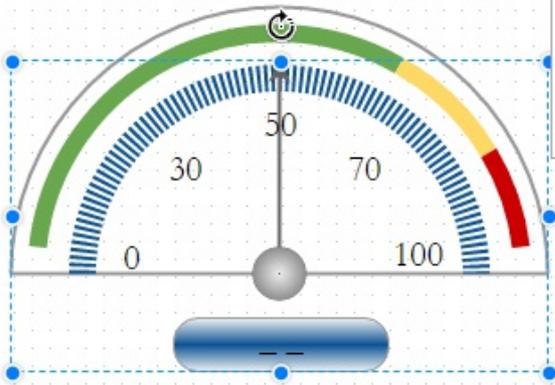
主要配置属性介绍：

1. 开始/45度.../315度：配置表盘各个角度的数据值，比如该关联值的值范围是0-160，那么每个刻度分别是0/20/40.../140。
2. rotdeg：旋转角度，即表盘指针的角度。一般需要关联点，比如关联testds\_tag66，按照总角度和最大值范围，需要配置计算公式  $\text{testds\_tag66} * 360 / 160$ 。注意160替换为当前表盘的最大值。
3. 旋转速度：暂时无意义。
4. 标题：表盘左上角的标题。
5. 关联的变量：右上角显示的值，一般和rotdeg中关联的值一致。

其他类型的仪表盘截图如下：

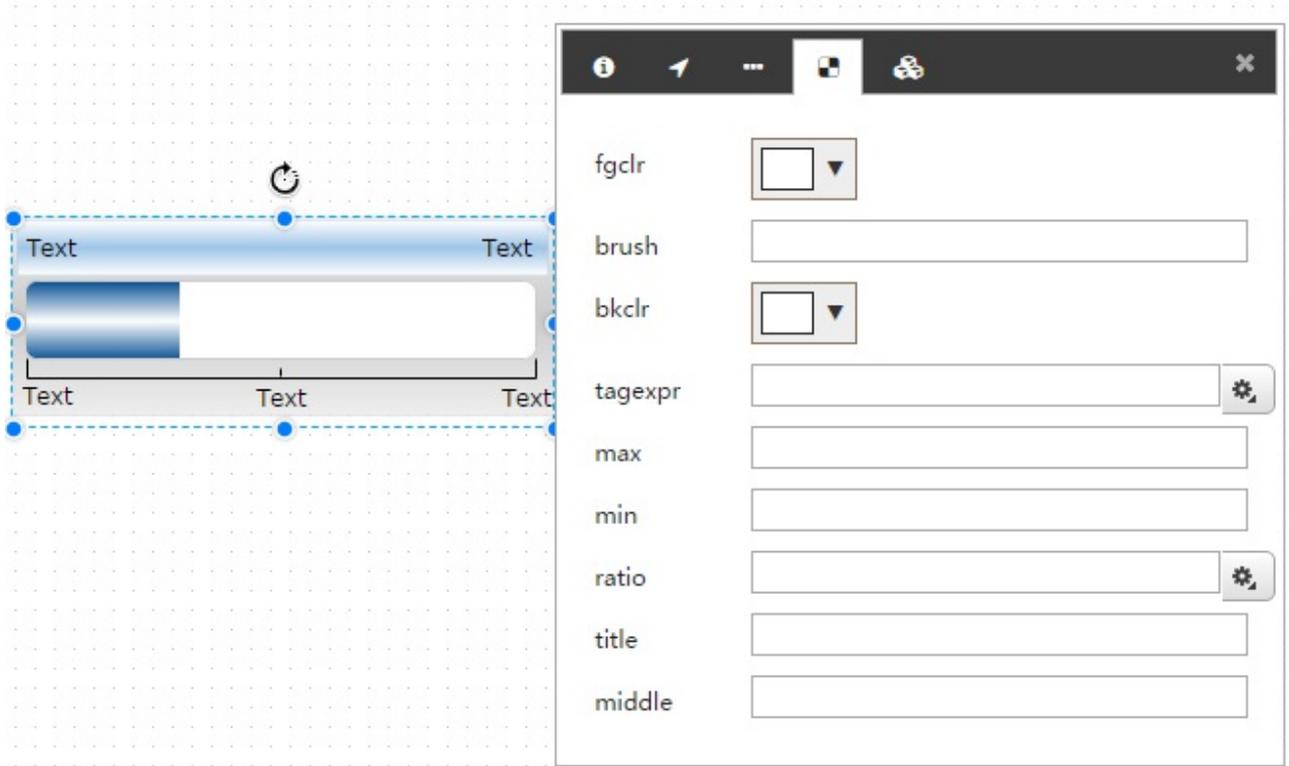


表盘颜色	<input type="color"/>
表盘值	<input type="text"/>
箭头X值	<input type="text"/>
箭头Y值	<input type="text"/>
最大值	<input type="text" value="100"/>
最小值	<input type="text" value="0"/>



Tag值	<input type="text"/>
指针角度X	<input type="text"/>
指针角度Y	<input type="text"/>

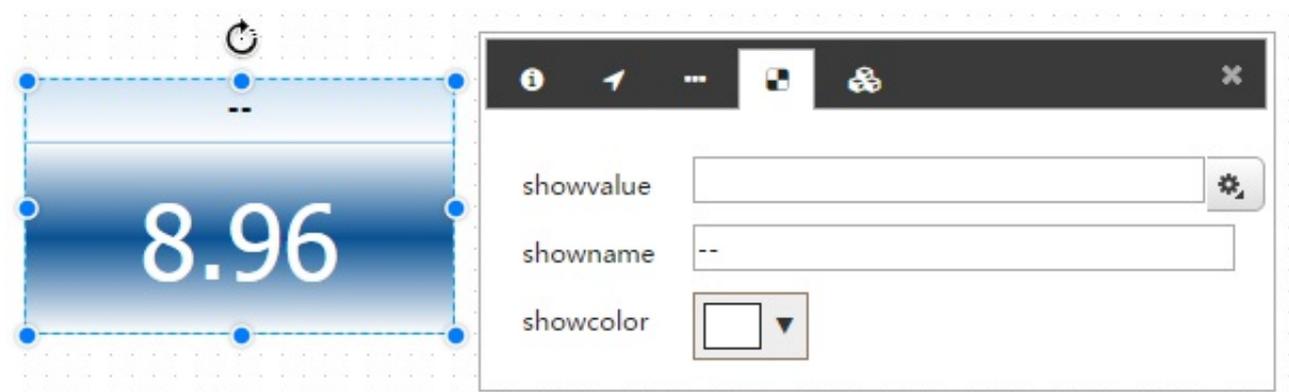
### 5.5.2 数据刻度计



数据刻度计根据关联的变量，以温度计的形式显示当前值，随着数据值变化液位高低发生变化。主要配置属性介绍：

1. fgclr/bkclr: 分别是前景色和背景色
2. brush：画刷风格
3. min/middle/max:刻度计上的最小、中间和最大值。比如值范围是0-300，那么分别配置0,150,300.
4. tagexpr：是右上角变量值
5. ratio：是刻度计的填充比例，配置为tagexpr/max\*100。因为填充比例是百分比数字，必须小于100.
6. title：左上角标题显示

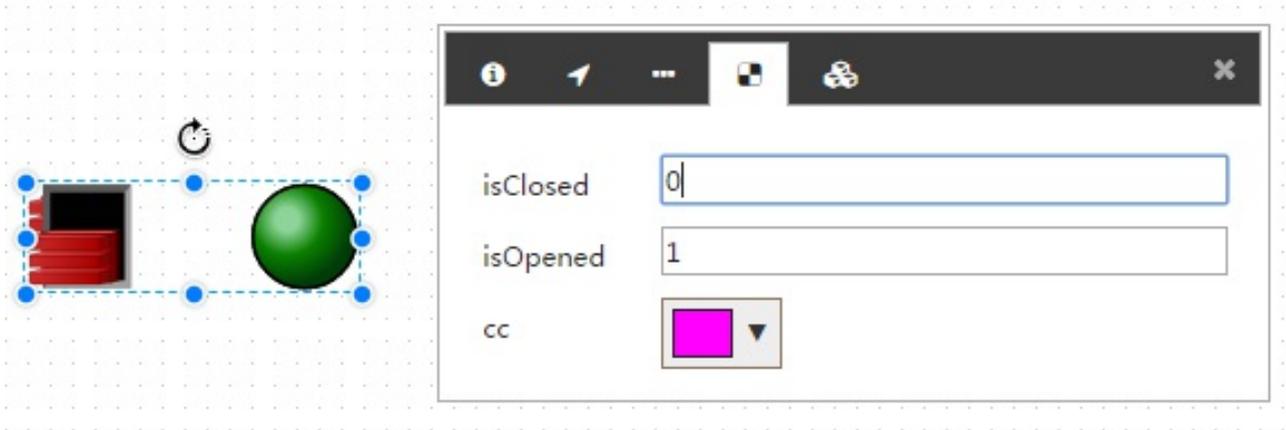
### 5.5.3 数据展板



数据展板以较大的展板方式显示关联数据值。

1. showvalue：要显示的数据值关联点。
2. showname：上方要显示的标题
3. showcolor：展板背景色

### 5.5.4 开关灯



开关灯图元主要功能是通过脚本控制图片的切换，从而实现开关灯效果。

1. isClosed：是否关闭
2. isOpenned：是否打开
3. cc：暂时无意义。

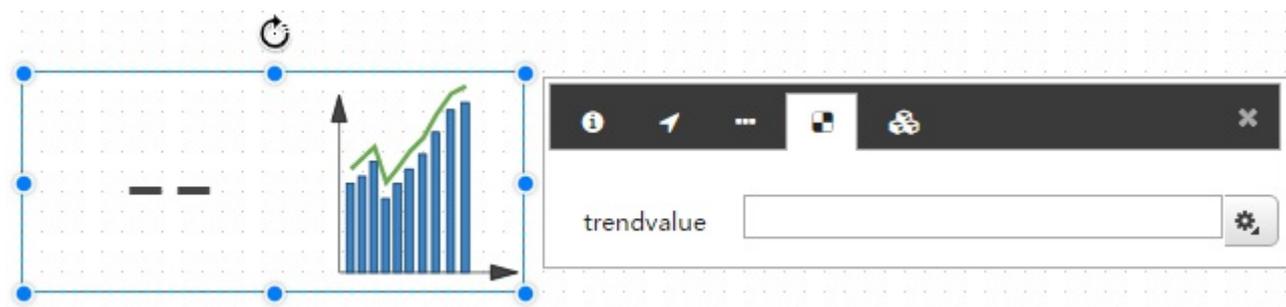
脚本内容：

```
开关灯2249520_OnClick = function() {  
    var isClosed = sys.GetAttribute("开关灯2249520", "#isClosed");  
    if (isClosed == 1) {  
        // open it  
        sys.SetAttribute("开关灯2249520", "#is0pened", 1);  
        sys.SetAttribute("开关灯2249520", "#isClosed", 0);  
    } else {  
        // close it  
        sys.SetAttribute("开关灯2249520", "#is0pened", 0);  
        sys.SetAttribute("开关灯2249520", "#isClosed", 1);  
    }  
};
```

### 5.5.5 灯闪烁

主要功能是根据配置的open属性，切换不同的图片，可以在图元设计器中查看。

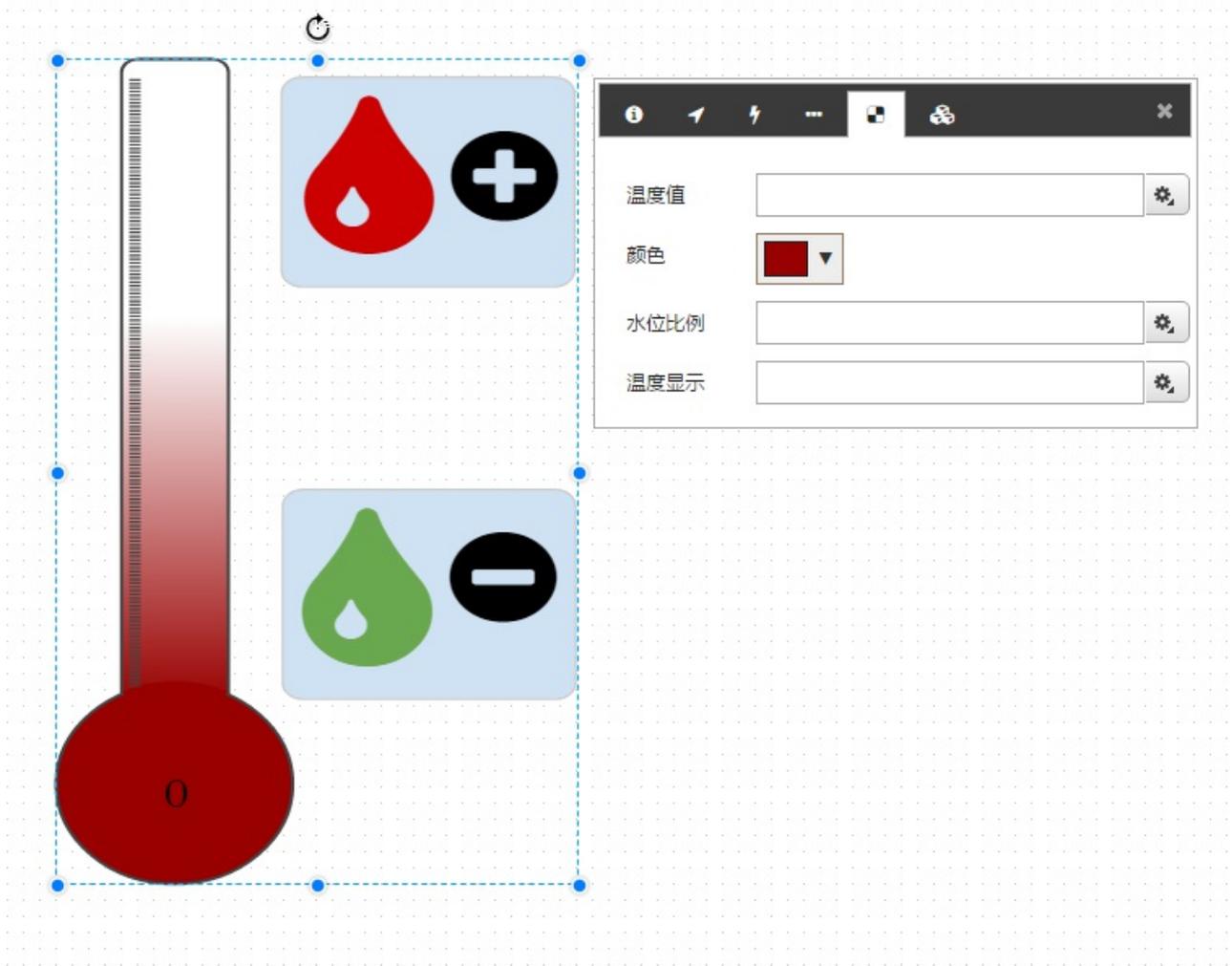
### 5.5.6 趋势查看点



趋势查看点图元通过绑定数据点，可以展示该关联点的实时值，同时点击之后可以直接打开该点的趋势。

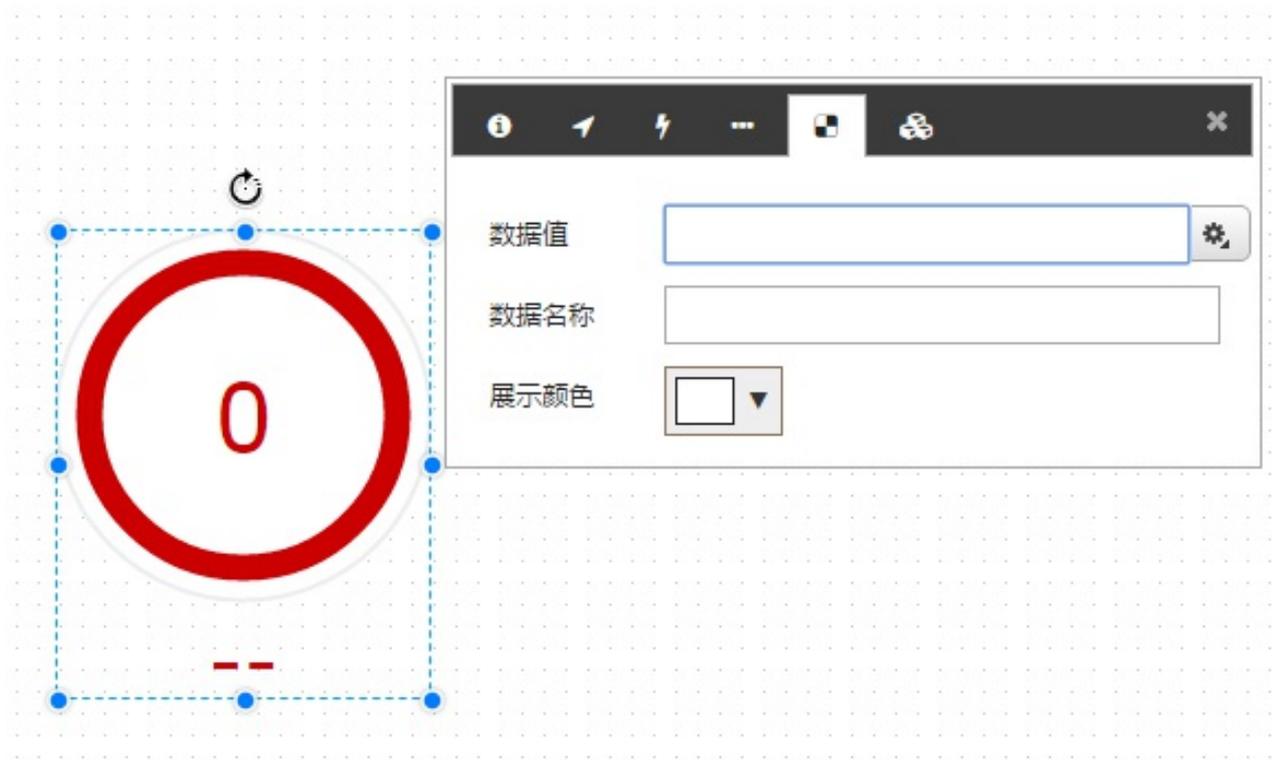
### 5.5.7 温度计

一个温度展示图元，随着底层温度值的变化，图元中的数值和填充比例会变化，同时可以支持对该温度值的写值操作，可以升高和降低温度。



### 5.5.8 数据展示圈

一个两层圆圈状的数据展示图元，可以配置实时数据进行展示。如下所示：



其余图元可以在系统图元管理菜单中看到，用户可以直接打开查看或者在自己的监控画面中使用。

## 6. 画面运行

最终设计完成的画面，可以通过画面管理中的分享按钮生成最终访问的URL，直接拷贝该URL到浏览器中，可以访问设计好的画面。

在画面设计过程中，可以直接使用快捷键 F6 预览画面，也可以点击预览按钮打开。如果您需要设置最终预览的设备屏幕大小，请查阅画面设计器中有关工具栏视图按钮的介绍。

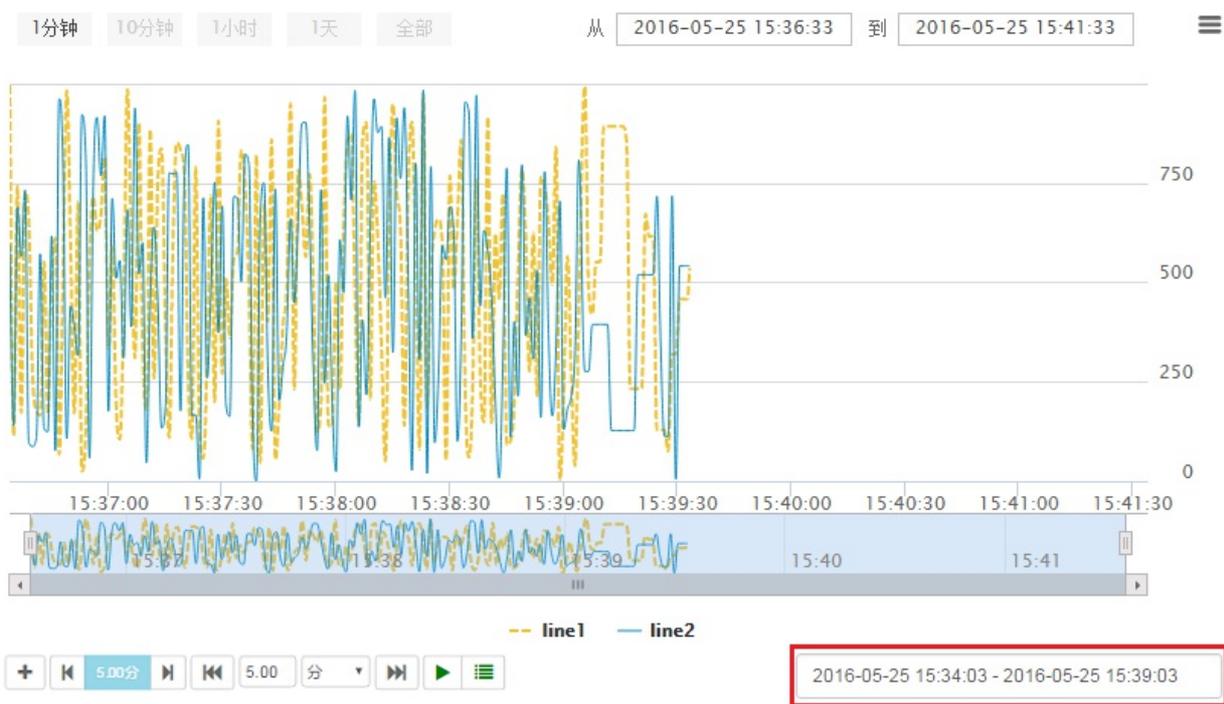
### 6.1 趋势图运行

趋势图运行起来有2种方式：一种是在设计器中使用快捷键 F6 预览画面，还有一种方式是通过脚本函数打开趋势图。

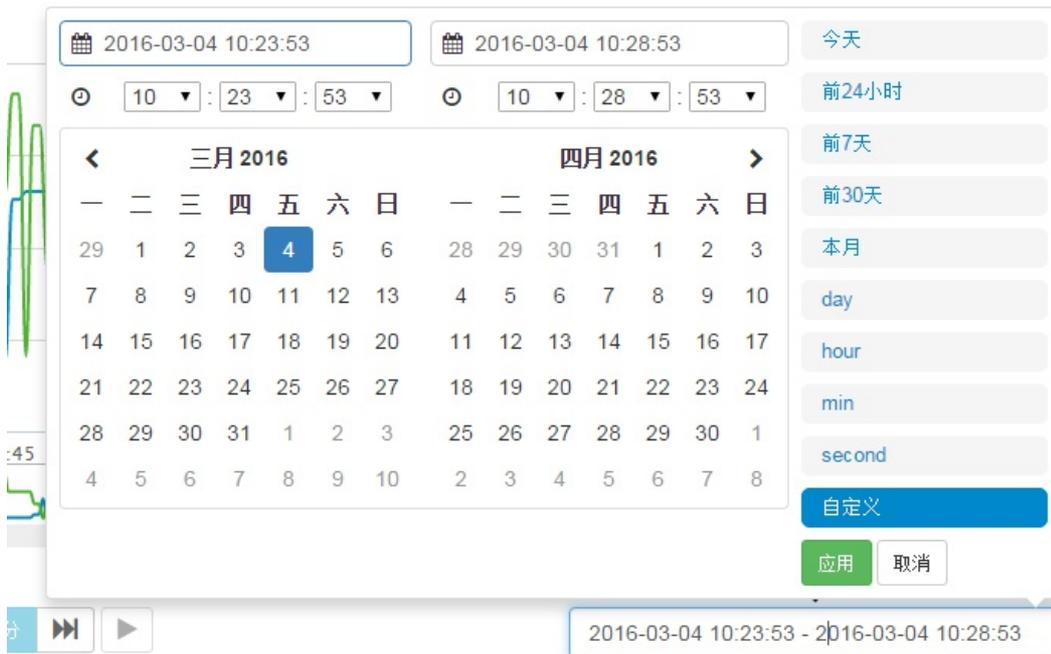
当以预览画面方式打开时，趋势图运行起来以后，界面如下图：



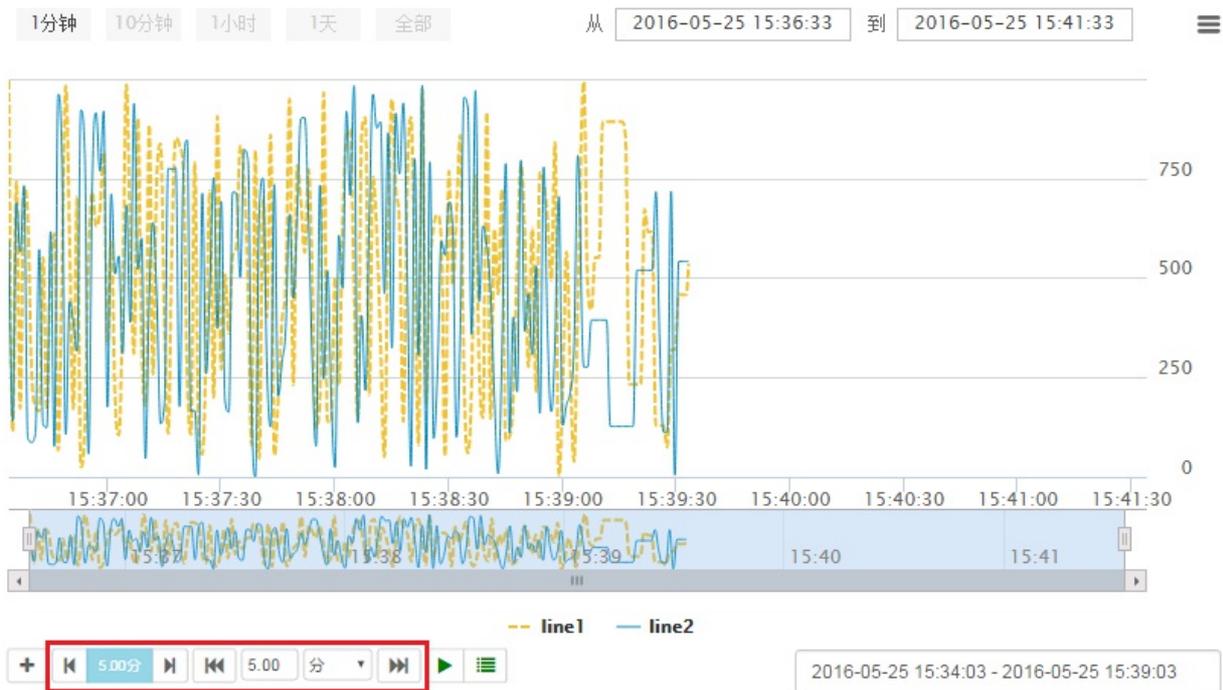
默认显示的实时数据，用户也可以通过配置历史时间查看历史趋势：



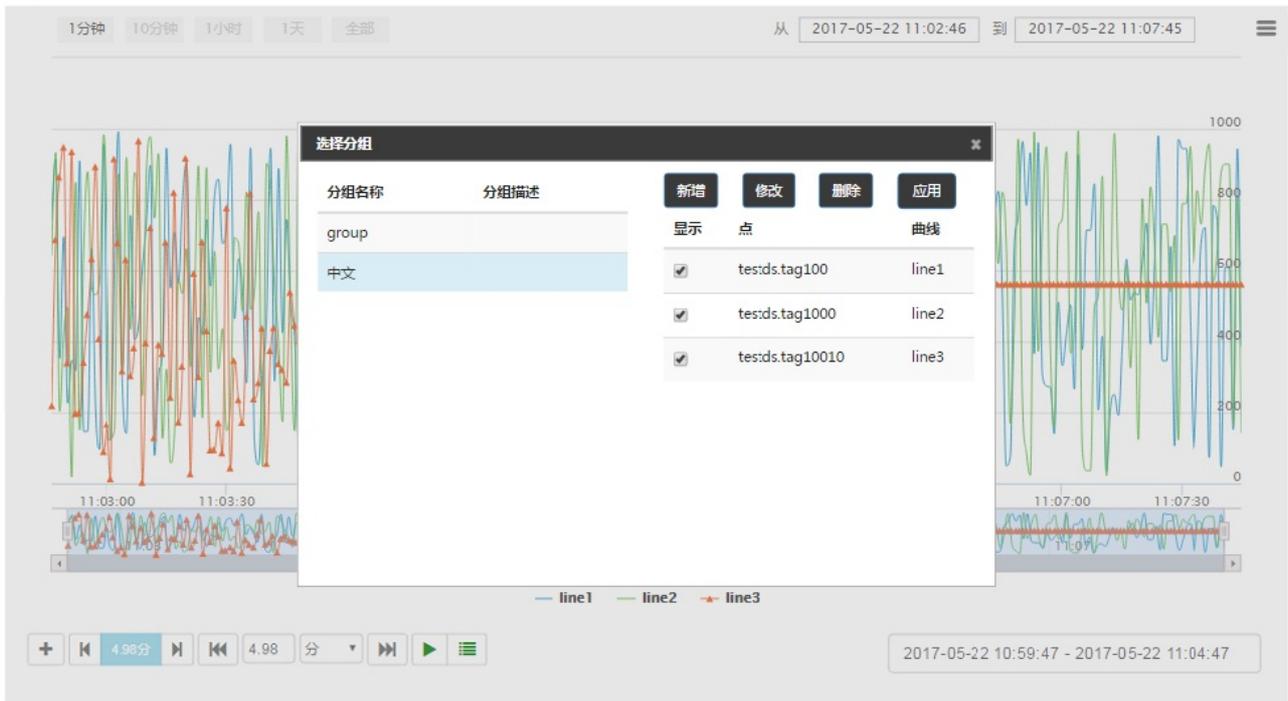
历史趋势时间配置界面：



用户可以通过点击向前或者向后翻页查看时间段前后的数据



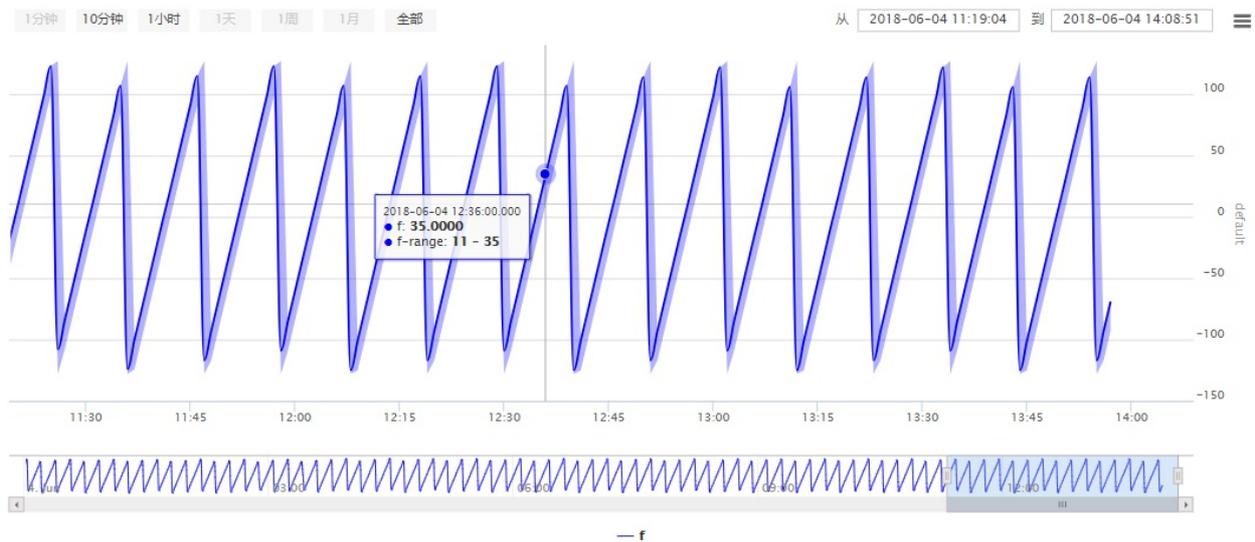
点击加号按钮可以弹出切换趋势组界面



在切换趋势组界面，可以选择想要查看的组点击组名，切换到该组。右侧该组下的曲线列表显示出来，也可以在运行态给已经存在的趋势组配置新的曲线（只在运行态有效）。



在趋势获取到的历史数据中，对于数据质量为停机记录的点，采用黄色圆圈符号标记，对于数据质量为bad的点，采用红色符号标记。



2018-06-04 00:00:00 - 2018-06-04 14:08:51

当以插值模式查询历史趋势时，可以查询当前插值点和上一个插值点之间的最大值和最小值，和插值到的值一起显示在当前时间点上。

注：1. 查看值范围的功能只有在历史趋势下有效。2. 如果用户查询的历史时间范围内的原始值超过65535条，超出的范围不能显示range

- 新增曲线



- 修改曲线 首先选中一条曲线，然后点击修改按钮弹出修改对话框，曲线名称不能修改



- 删除曲线 首先选中一条曲线，然后点击删除按钮，该曲线被删除

以上操作以后，点击“应用”按钮，编辑生效。注意：修改已经存在曲线的统计值曲线，即时生效。

选中组名，右侧列表显示改组下所有的tag点，勾选前面的checkbox，如果勾上，该条曲线显示，如果不勾选，则该条曲线不显示。配置好以后，点击应用按钮，配置生效。如果该条曲线配置为不显示，则对应的统计曲线也不显示。

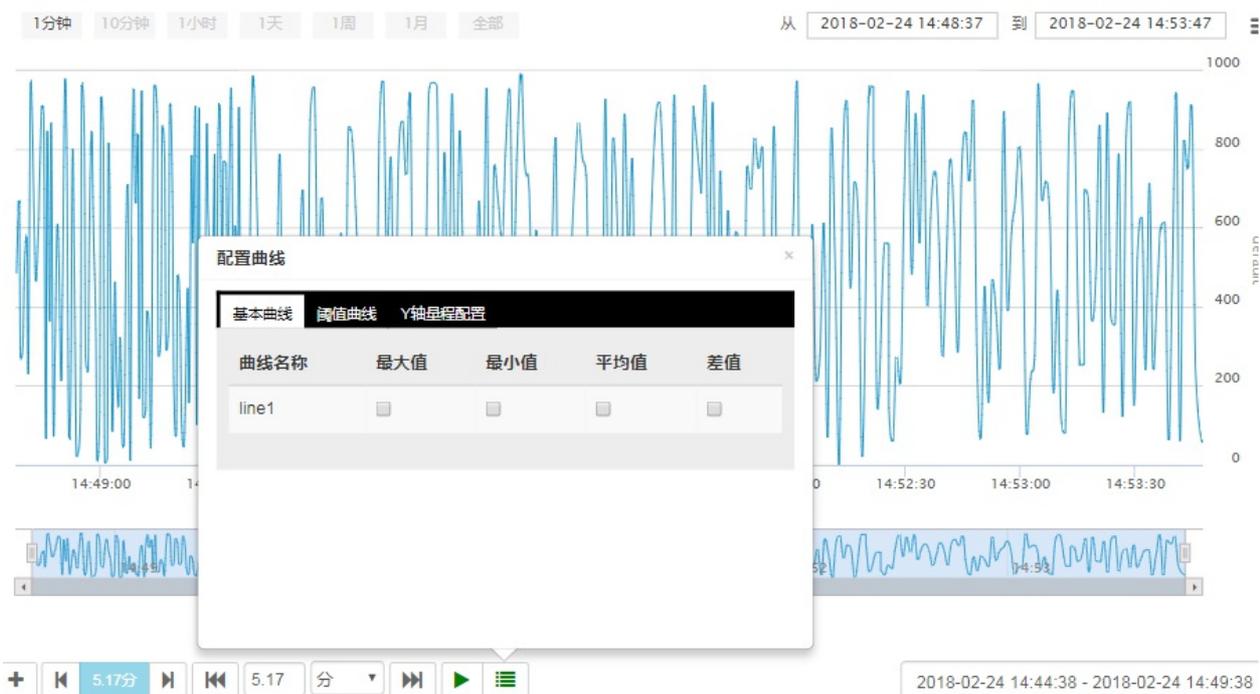
注：1. 当用户时间范围选择的很小时，系统默认界面显示5个点的值，统计曲线统计的是最右上方时间框范围的统计值，所以当时间范围内只有一个点时，这时候最大值和最小值平均值是重叠的。

2. 隐藏掉导航栏以后，运行态会出现一个滚动条，拖动滚动条会改变时间范围，当范围超出当前时间范围时，可能看不到数据值，只需要拖动滚动条到合理时间范围内，就可以看到了。

3. 当主题选择统计时，统计曲线说明字体默认黑色，直接看可能会看不清楚，选中是可以看到的。

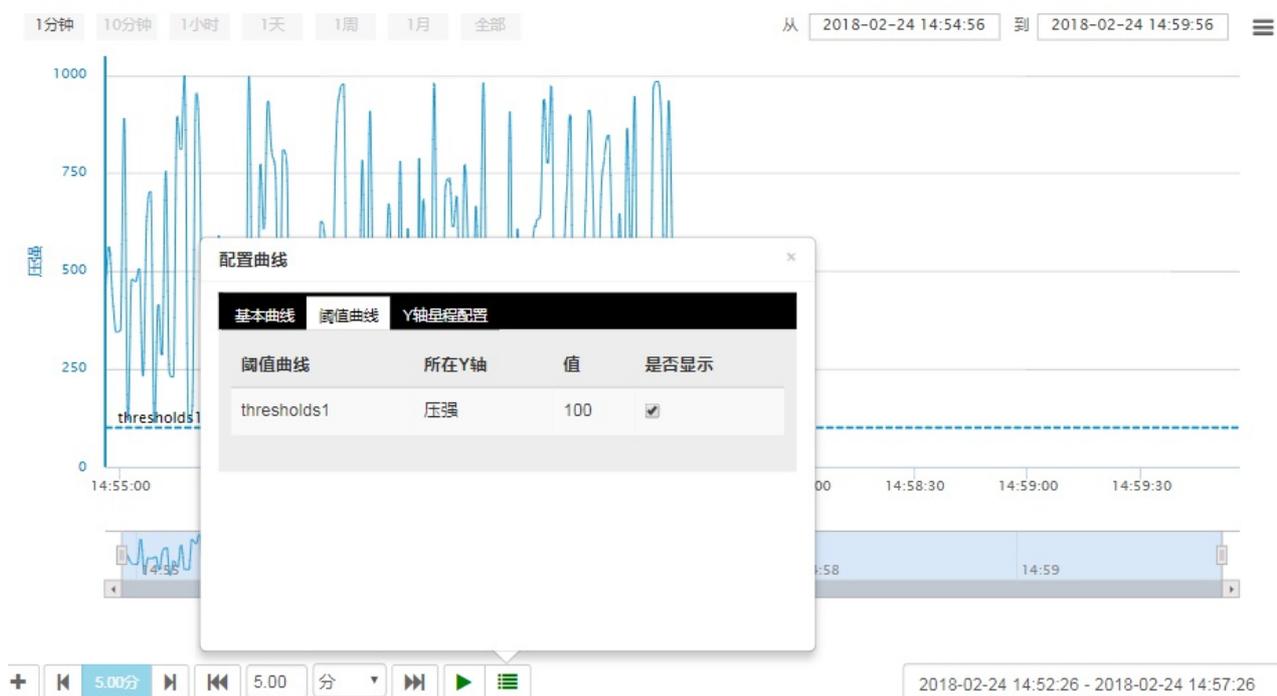
统计曲线配置界面：

点击如图按钮：弹出统计曲线配置界面



勾选想要查看的统计曲线，该曲线显示在主界面上（注，当该曲线值与Y轴值一致时，该曲线显示看不到）

阈值曲线界面配置：

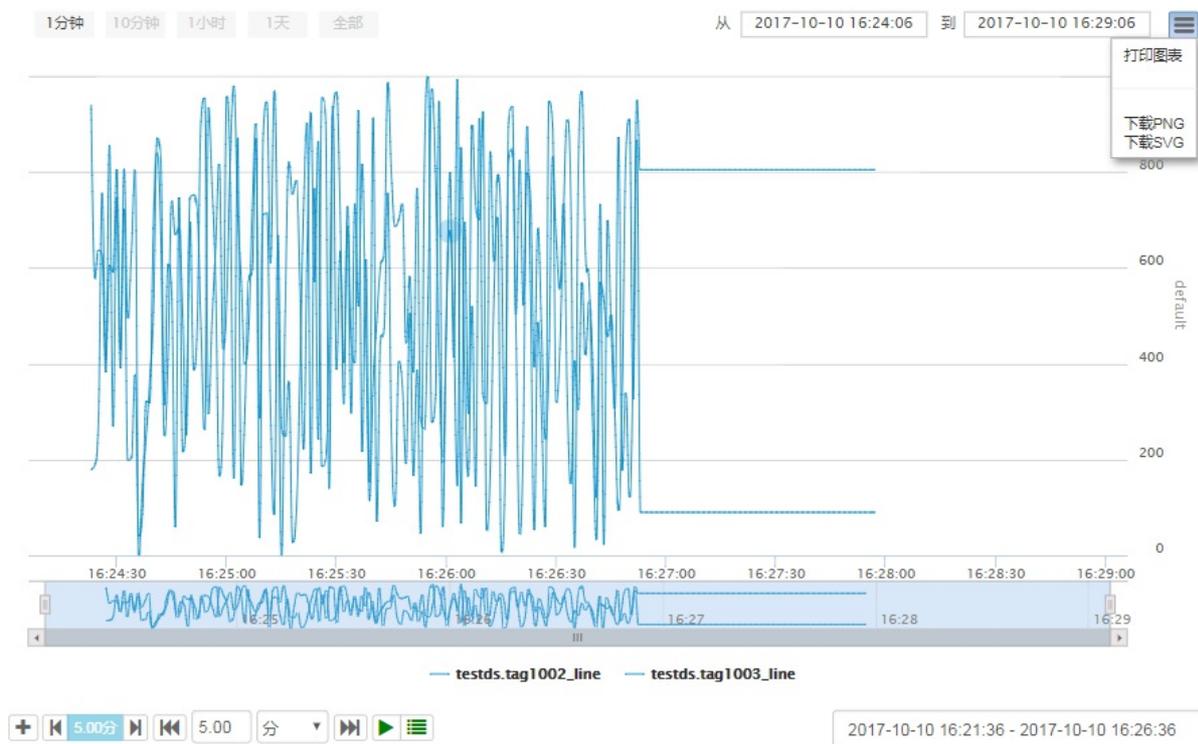


勾选想要查看的阈值曲线，该曲线显示在界面上。

Y轴范围配置：



输入想要的Y轴范围，点击应用按钮，Y轴的量程被改变，点击恢复默认按钮，Y轴量程恢复到系统默认  
趋势图打印和下载



- 打印趋势图 在趋势图运行态时，点击右上角的菜单，弹出打印按钮，点击，可以打印当前趋势图
- 下载趋势图 在趋势图运行态时，点击右上角的菜单，选择下载按钮，包括下载PNG， 下载SVG两个按钮，点击不同的按钮趋势图被下载成不同的文件格式。

当在对话框中打开趋势图时，我们可以在脚本中添加对趋势图背景色的设置（见OpenNewTrend函数说明），同时我们也可以通过最大化，最小化，恢复按钮控制趋势图的窗口大小。



## 6.2 报警控件运行

报警控件默认的运行效果如下图显示。实际运行中，会依据用户配置的列选项、报警过滤等信息有所变化。

实时报警列表									
<input type="checkbox"/>	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
<input type="checkbox"/>	hdds	AlarmTagG	2	高高报警	96	2016-07-12 15:54:25.217	未恢复	未确认	hdds
<input type="checkbox"/>	testds	tag5	5	高高	900	2016-07-12 15:50:45.849	未恢复	已确认	tag5
<input type="checkbox"/>	testds	tag1	7	高	90	2016-07-12 15:50:45.849	未恢复	已确认	tag1
<input type="checkbox"/>	testds	tag2	8	高	9990	2016-07-12 15:50:45.849	未恢复	未确认	tag2
<input type="checkbox"/>	testds	tag3	9	高高	99999	2016-07-12 15:50:45.849	已恢复	未确认	tag3
<input type="checkbox"/>	testds	tag4	4	低低	9	2016-07-12 15:50:45.849	已恢复	未确认	tag4
<input type="checkbox"/>	testds	tag6	6	高	90	2016-07-12 13:20:44.804	未恢复	已确认	tag6
<input type="checkbox"/>	lq	amTag3	3	高高报警	8299	2016-07-01 16:42:01.760	未恢复	已确认	lq
<input type="checkbox"/>	lq	amTag3	1	低报警	429	2016-07-01 16:42:00.773	未恢复	未确认	lq
<input type="checkbox"/>	lq	amTag4	3	低低报警	17	2016-07-01 16:41:59.811	未恢复	未确认	lq

确认报警     收起报警过滤    1 / 共 3 页    10 / 1 - 10 共 21 条

<input type="checkbox"/> 启用区域过滤	<input type="text"/>	配置	<input type="checkbox"/> 启用优先级过滤	<input type="text"/>	配置
<input type="checkbox"/> 启用状态过滤	<input type="text"/>	配置	<input type="checkbox"/> 启用变量名称过滤	<input type="text"/>	配置
<input type="checkbox"/> 启用时间过滤	<input type="text"/>	配置	<input type="checkbox"/> 启用类型过滤	<input type="text"/>	配置
<input type="checkbox"/> 启用报警声音					

报警控件每间隔5秒自动从服务器抓取一次最新的报警信息，并显示在运行态。用户可以通过点击刷新按钮手动刷新。

运行态向用户提供了报警分页功能，提供每页 10、20、30 条报警信息分页选项。打开运行态的时候，会默认采用 10 条报警信息分页的配置，并且停留在第一页。用户可以通过报警列表下方的分页导航键（第一页、上一页、下一页、最后一页）进行分页跳转，也可以通过输入相对应的页码后按下回车确认，快速跳转到对应页码。

用户可以通过点击表格第一栏的勾选框选中当前页的所有报警信息，也可以通过逐一勾选，选中当前页的部分报警信息。在选中报警信息后，用户可以通过点击**确认报警**按钮，会通过后台改变**未确认**状态的报警信息为**已确认**。确认成功后，页面上将给出提示信息，同时清除已勾选的项目。

报警控件提供了双击确认报警的功能，只需要在某一行报警记录上快速点击两次鼠标，即可完成确认操作。

用户可以在运行态点击表格中的列名触发报警排序。目前支持报警排序的列包括：

- 变量名称
- 发生时间
- 报警状态
- 优先级

第一次点击列名的时候触发升序排列，第二次点击则触发降序排列。

用户可以在运行态修改报警过滤条件。点击报警控件上的**展开报警过滤**按钮，弹出过滤配置界面。若画面设计的时候已经配置了过滤条件，则配置界面中将加载有关的配置信息。通过勾选单选框可以启用、停止某一过滤条件。点击配置按钮则会弹出相对应的配置界面。配置方法与画面设计中一致，请参见画面设计中有关报警控件使用的章节。运行态的过滤条件不会保存，下一次打开画面时依旧按照画面配置的过滤条件运行。

用户可以在过滤配置界面中勾选或者取消勾选来决定是否在运行态启用声音报警。

若配置了精简模式，则运行态下的报警控件显示成如下样式

testds	tag2	8	高	9990	2016-10-24 13:49:05.918	未恢复	未确认	tag2
testds	tag5	5	高高	900	2016-10-24 13:19:05.697	未恢复	已确认	tag5
testds	tag6	6	高	90	2016-10-24 13:19:05.697	未恢复	已确认	tag6
testds	tag4	10	高高	99999	2016-10-24 13:19:05.697	已恢复	未确认	tag4
testds	tag3	3	低	99	2016-10-24 13:19:05.696	已恢复	未确认	tag3
testds	tag1	1	高高	999	2016-10-24 12:49:05.676	未恢复	未确认	tag1
hdds	AlarmTagG	2	高高报警	83	2016-10-21 16:10:58.666	未恢复	未确认	hdds
lq	amTag4	3	低低报警	53	2016-10-20 16:23:19.924	未恢复	未确认	lq
lq	amTag3	3	高高报警	27941	2016-10-20 16:22:43.917	未恢复	已确认	lq
lq	amTag3	1	低报警	338	2016-10-20 16:22:41.916	未恢复	未确认	lq

## 6.3 历史报警运行

历史报警控件默认的运行效果如下图显示。实际运行中，会依据用户配置的列选项、查询条件等信息有所变化。

数据源 testds Tag名称 \*tag\* 起止时间 2016-06-02 09:01:12 - 2016-06-02 09:06:12

优先级 从 0 到 255 (范围:0~255) 报警类型 高高报警 查询

数据代理	报警点	优先级	报警类型	报警值	发生时间	恢复时间	报警
testds	alarmtag0	0	alarmType0	0	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	未
testds	alarmtag1	1	alarmType1	1	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	已
testds	alarmtag2	2	alarmType2	2	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	未
testds	alarmtag3	3	alarmType3	3	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	已
testds	alarmtag4	4	alarmType4	4	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	未
testds	alarmtag5	5	alarmType5	5	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	已

1 共 1,000 页 1 - 10 共 10,000 条

用户可以根据实际需求，输入查询条件，包括数据源名称，Tag名称，历史报警发生的起始时间，优先级，以及报警类型五个条件进行查询。输入好查询条件以后，点击查询按钮，得到查询结果。

- 数据源名称 选择需要查找点所在的数据源名称

数据源 testds Tag名称 \*tag\* 起止时间 2016-06-02 09:01:12 - 2016-06-02 09:06:12

优先级 到 255 (范围:0~255) 报警类型 高高报警 查询

数据代理	报警点	优先级	报警类型	报警值	发生时间	恢复时间	报警
testds	alarmtag0	0	alarmType0	0	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	未
testds	alarmtag1	1	alarmType1	1	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	已
testds	alarmtag2	2	alarmType2	2	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	未
testds	alarmtag3	3	alarmType3	3	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	已
testds	alarmtag4	4	alarmType4	4	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	未
testds	alarmtag5	5	alarmType5	5	2016-05-23 16:11:19 559	2016-05-23 16:11:19 559	已

- tag名称 目前只支持精确名字查询和全部查询。
- 起始时间 点击输入框弹出时间配置界面如下图：默认时间范围是当前时间往前推一个小时

数据源 testds Tag名称 \*tag\* 起止时间 2016-06-02 09:01:12 - 2016-06-02 09:06:12

优先级 从 0 到 255 (范围:0~255) 报警类型 高高报警 应用 取消

历史报警列表

数据代理	报警点	优先级	报警类型	报警值	发生时间
testds	alarmtag0	0	alarmType0	0	2016-05-23 16:11:19 559
testds	alarmtag1	1	alarmType1	1	2016-05-23 16:11:19 559
testds	alarmtag2	2	alarmType2	2	2016-05-23 16:11:19 559
testds	alarmtag3	3	alarmType3	3	2016-05-23 16:11:19 559
testds	alarmtag4	4	alarmType4	4	2016-05-23 16:11:19 559
testds	alarmtag5	5	alarmType5	5	2016-05-23 16:11:19 559

0 共 页 10

- 优先级 直接输入优先级数字，范围为0-255

- 报警类型 点击报警类型下拉框，选择想要查看的报警类型，如下图：

数据源  Tag名称  起止时间

优先级 从  到  (范围:0~255) 报警类型

历史报警列表

数据代理	报警点	设备点名	优先级	报警类型	报警值	恢复时间

报警类型下拉框选项：  
 所有报警  
 高高报警  
 高报警  
 低报警  
 低低报警  
 变位报警  
 偏差报警  
 0状态报警  
 1状态报警  
 0->1报警  
 1->0报警

分页：0 / 共 页 | 10

运行态向用户提供了报警分页功能，提供每页 10、20、30 条报警信息的分页选项。打开运行态的时候，会默认采用 10 条报警信息分页的配置，并且停留在第一页。用户可以通过报警列表下方的分页导航键（第一页、上一页、下一页、最后一页）进行分页跳转，也可以通过输入相对应的页码后按下回车确认，快速跳转到对应页码。

## 6.4 历史回放

在运行态可以按照时间进行画面回放。

在运行态，鼠标移到屏幕左上方，左侧会出现浮动条，点击回放按钮，在屏幕上方弹出回放工具条



默认为回放指定时刻的，选择需要回放的时间点，点击回放按钮，画面会进行回放，点击实时按钮，回到实时刷新状态。

点击工具条左侧按钮，可切换到回放时间段界面。



选择时间段，回放数据的间隔，点击回放，画面会根据设置的时间段，按照间隔进行回放刷新。在当前播放时间中显示当前回放的变量时间戳。点击实时按钮，跳转回实时刷新状态。

点击关闭按钮，如果当时处于回放状态，界面上方会有悬浮图标显示，点击该按钮，打开回放界面。



### 历史回放的回调事件

用户在选择时间点的回放后，将会触发 OnGetHisDataEvt(timems) 函数。用户可以在页面脚本中定义该函数，传入的 timems 是当前选择的毫秒值。

## 6.5 前端消息查看

通过快捷方式 Ctrl + Q调出日志查看器，查看刷新数据中的错误信息和部分其他消息。可显示最新的100条信息。

### 信息显示 ✕

搜索

时间	类型	内容	错误码
2016-06-02 10:39:24	错误	test	11
2016-06-02 10:39:22	信息	parse time 77ms.	

## 6.6 html文件上传运行

当用户配置好html控件以后，就可以在运行态加载起来，例如，用户配置了URL资源：<http://www.baosight.com>,然后运行画面



GO



GO



GO

股价展示

600845 宝信软件  
**20.680 -0.19%** **413万元**  
900926 宝信B

最新动态

- 宝信软件与秦皇岛市政府签约
- 宝信软件参展2016第二十届
- 宝信软件轨道交通项目团队
- 宝信软件信息系统集成及服
- 宝信软件获得首批ITSS运
- 公司签约印度SAIL-ROURKE

如果用户配置了HTML资源(一个上传文件的界面)：



选择文件 未选择任何文件

上传

仅支持zip文件，单个文件限制10MB

## 7. 常见问题

### 7.1 无法正常启动 start\_ds.bat 脚本

首先确认当前系统是64位，再查看当前系统可用内存大小，默认redis配置了1G的启动内存，如果实际可用内存小于该值，将无法启动。

### 7.2 模拟数据源启动不了

启动模拟数据源，需在 iMV-Server\iMV-RandDemoCollector 路径下 collector.properties 中配置 redis 的 IP，并且将 lessee 租户配置成正在使用的租户。

每次配置之后需重启模拟数据源，重启间隔为一分钟以上，不然可能会报错。

### 7.3 在组态运行中调用自定义函数库

本系统使用了标准的 HTML5 用户界面，因此您可以方便的将自定义的 JavaScript 函数库添加到系统中，在运行态使用。**假设**您的函数全部存在在一个叫做 foo.js 的文件中，并且没有使用其他函数库，那么您需要做得操作步骤是：

1. 把 foo.js 文件放置在 iMV-Core\webapps\cvs\js 路径下；
2. 打开需要添加该脚本的画面进行编辑，打开
3. ，选择 Page 对象的 OnLoad 函数。在该函数中动态添加 foo.js 。具体代码如下：

```
var myScript = document.createElement("script");
myScript.type = "text/javascript";
myScript.src = "js/foo.js";
document.body.appendChild(myScript);
```

4. 接下来便可以在该画面中使用foo.js中定义的变量和函数了。

额外的，foo.js 加载在系统脚本函数库之后，因此 foo.js 中可以调用iMV中所包含的库函数，例如 jQuery 的 \$ 函数、Moment.js 中的 moment() 函数。

### 7.4 CVS 网络通信依赖哪些端口

默认安装的 CVS 需要如下端口：

#### 端口号

#### 说明

18080 CVS 应用端口，通过浏览器访问，包括设计态、运行态。

2998 Redis 端口，参见 1.5.1 章节，iCV、iHD 数据转储时需要与该端口通信。

5432 PostgreSQL 端口，参见 1.5.2 章节。

### 7.5 如何与JDT集成？

1. 设置Tomcat的session共享配置，打开 \$TOMCAT\_HOME\$/conf/context.xml 文件，给 <Context /> 标签添加 path="" 和 sessionCookiePath="/" 的两个属性，再给该标签添加如下子元素

```
<Valve  
className="com.orangefunction.tomcat.redissessions.RedisSessionHandlerVa  
lve" />  
<Manager  
className="com.orangefunction.tomcat.redissessions.RedisSessionManager"  
  host="0.0.0.0"  
  port="2998"  
  password="baosightpage"  
  database="0"  
  maxInactiveInterval="60" />
```

注意修改其中的redis地址、端口和密码信息

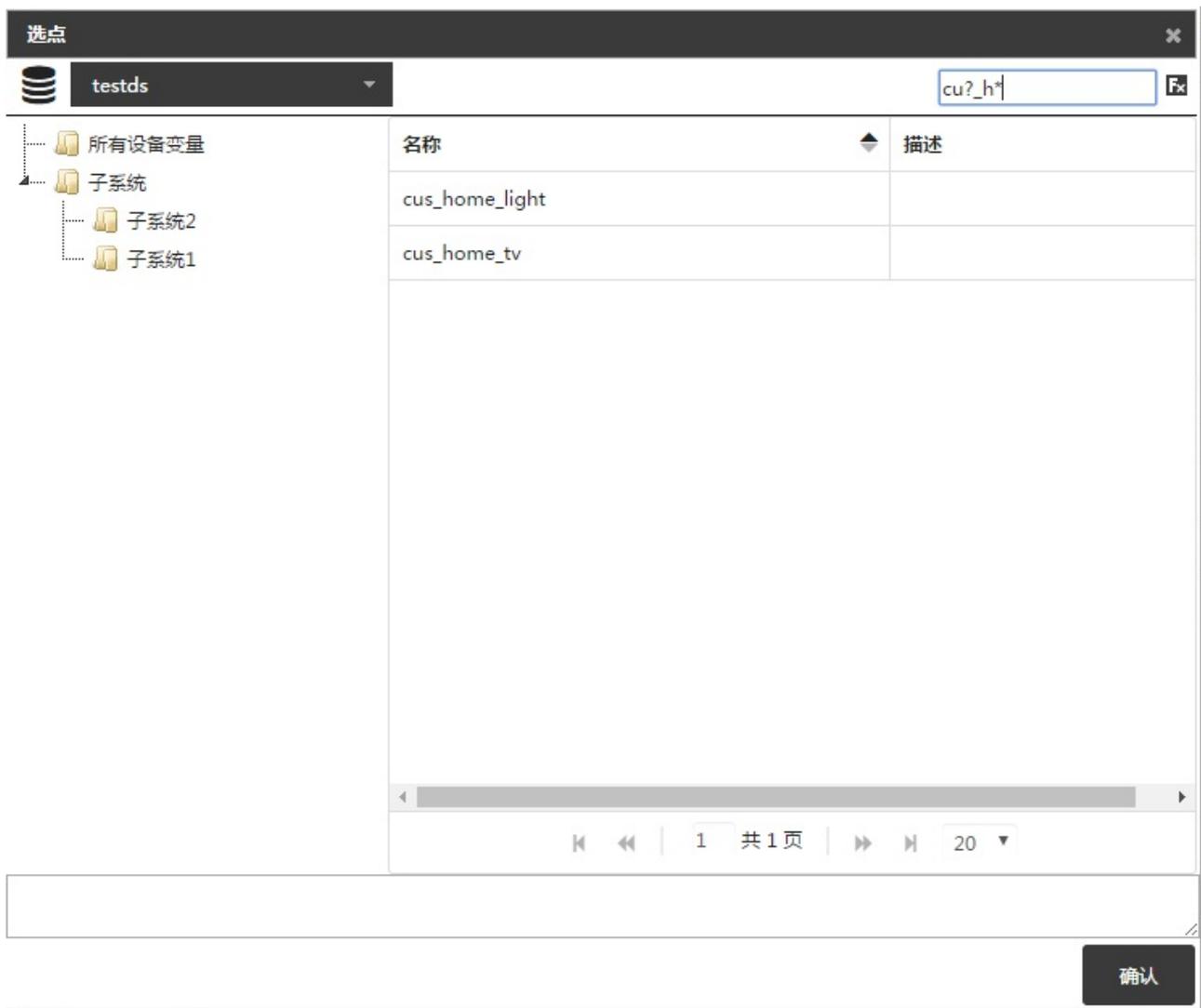
2. 修改CVS配置文件。修改 bpconf.properties 文件中的 aasonly 配置项为 yes，修改 auth\_host、auth\_port 配置项为对应的AAS地址和端口
3. 拷贝 xinsight-cvs-share.jar 文件到Tomcat的 lib 目录下。
4. 重启Tomcat，完成配置。

## 7.6 可以保存颜色选择框中的自定义颜色吗？

可以的。只要在任何一个颜色选择框中选中某个自定义的颜色，它将会出现在选择框的最下面一排。您可以在其他的颜色选择框中复用它。最多可以保存 7 个自定义颜色。如果使用了超过 7 个自定义颜色，则最早使用的将会被从队列中移除。

## 7.7 选点界面里不确定点名该怎么办？

选点界面的右上角提供了筛选输入框，输入关键字即可查询特定范围内的点。? 代表单个未知字符，\* 代表多个未知字符。



## 7.8 关于点击起始页的示例画面跳转到对应分享链接的说明

此功能是为了让用户在未登录的状态下更好地了解cvs系统而设计的，用户点击起始页上的示例画面后，能够跳转到相应的分享链接。以下是两个需要注意的点：

1. 页面上的示例画面均是使用了数据代理testds中的用户名和密码进行分享的，所以需要确保testds已授权且状态为运行中；并且collector.properties中lessee的值必须为用户。
2. 要使用此功能必须进行系统初始化并重新创建user租户，所以对于老用户需要慎重。