



Hawk 物联网应用平台

用户手册

北京金桥得安科技有限公司

版权所有 违者必究

目录

1	软件简介.....	7
2	“十大”经典问答	10
3	名词解释.....	14
3.1	设备类型.....	14
3.1.1	概述.....	15
3.1.2	数据逻辑.....	16
3.1.3	表现视图.....	17
3.2	设备.....	17
3.3	设备列表.....	17
3.4	设备库.....	17
4	工程结构.....	18
4.1	I/O 通信	18
4.2	设备列表.....	19
4.3	业务逻辑.....	19
5	设备状态.....	19
6	“7 语言”	20
6.1	变量类型.....	20
6.2	运算符.....	21
6.3	流程控制.....	21
6.4	其他.....	21
7	运行时.....	21
8	步骤一、产品的安装部署	24
8.1	开发环境 HawkBuilder 的安装部署	24
8.2	运行时 HawkRuntime 的安装部署	24

8.2.1	服务器端.....	25
8.2.2	客户端.....	25
9	步骤二、创建新的设备类型	27
9.1	创建设备组.....	27
9.2	创建用户设备.....	28
9.3	导入用户设备.....	31
9.4	导出用户设备.....	32
10	步骤三、创建物联网应用工程	32
10.1	新建工程.....	33
10.2	复制工程.....	34
10.3	移动工程.....	34
10.4	重命名工程.....	35
10.5	刷新工程.....	35
10.6	关闭工程.....	36
10.7	导入工程.....	36
11	步骤四、快速接入 OPC 数据源	37
12	步骤五、构建工程的设备列表	40
13	步骤六、定义工程的复杂业务逻辑	42
13.1	常用脚本.....	43
13.2	常用转义字符.....	44
14	步骤七、创建监控画面	44
15	步骤八、画面编辑	45
15.1	illustrator 绘制 SVG 的注意事项	45
15.1.1	不能存在符号(symbol)	45
15.1.2	不能使用过渡色.....	46
15.2	画布与大纲视图.....	46

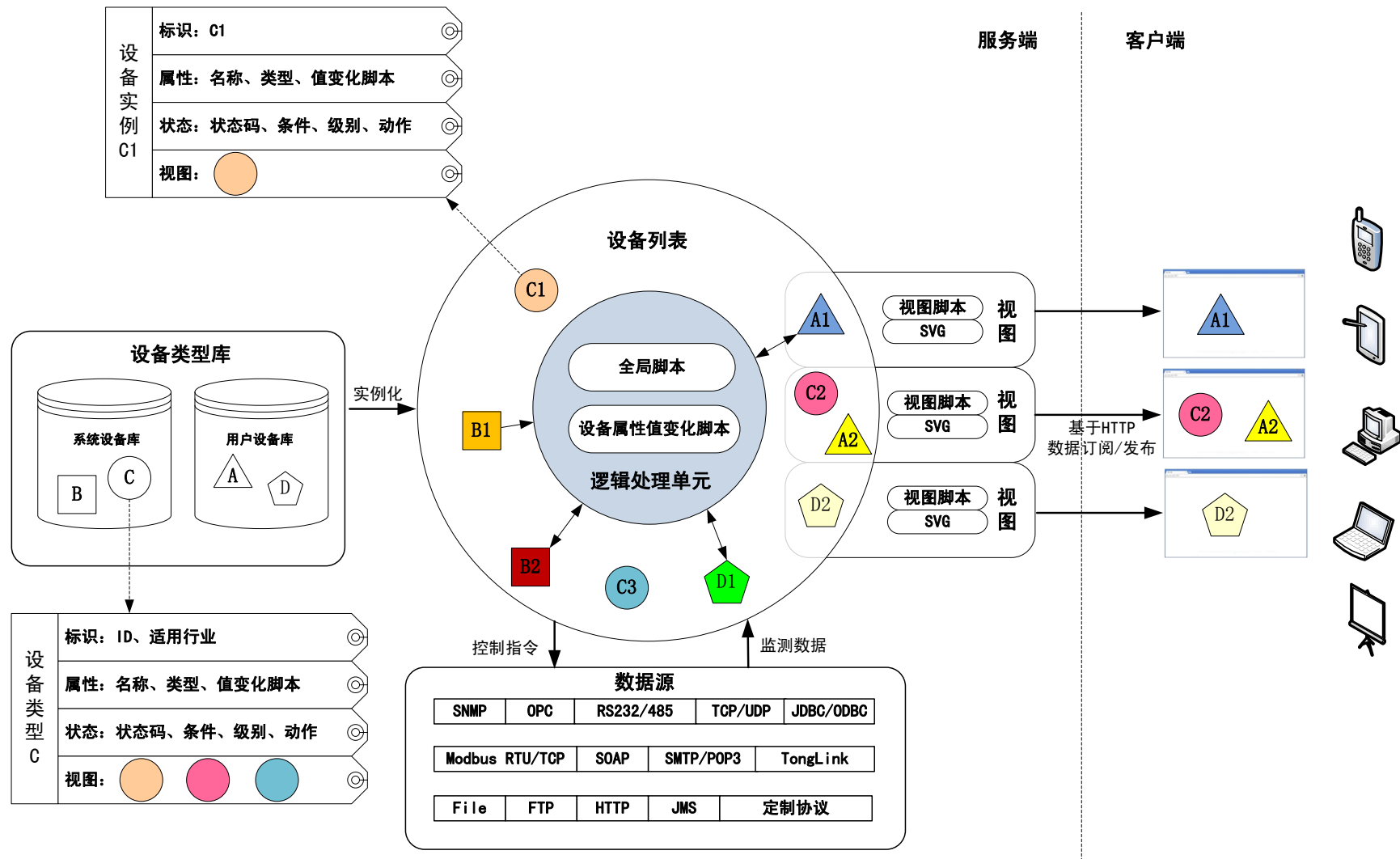
15.2.1	画布.....	46
15.2.2	大纲视图.....	47
15.3	图形元素.....	49
15.3.1	直线工具.....	49
15.3.2	矩形工具.....	50
15.3.3	椭圆工具.....	51
15.3.4	折线工具.....	51
15.3.5	多边形工具.....	52
15.3.6	铅笔工具.....	53
15.3.7	钢笔工具.....	54
15.3.8	添加锚点工具.....	55
15.3.9	删除锚点工具.....	55
15.3.10	转换锚点工具.....	56
15.3.11	图像工具.....	56
15.3.12	文本工具.....	57
15.4	图形编辑.....	58
15.4.1	移动.....	58
15.4.2	复制、粘贴、剪切、删除.....	59
15.4.3	撤消/重做	59
15.4.4	对齐方式.....	59
15.4.5	长宽匹配.....	62
15.4.6	排列分布.....	63
15.4.7	图层顺序.....	66
15.4.8	旋转翻转.....	67
15.4.9	自由旋转和错切.....	69
15.5	视图构建(静态部分).....	72
16	步骤九、创建监控页面的动画效果	72
16.1	可见性.....	73
16.1.1	闪烁动画.....	73
16.1.2	显示/隐藏动画	73
16.2	填充.....	73
16.2.1	区间填充动画.....	73
16.2.2	百分比填充动画.....	74

16.3	缩放.....	74
16.3.1	水平线性缩放动画.....	74
16.3.2	水平区间缩放动画.....	75
16.3.3	垂直线性缩放动画.....	75
16.3.4	垂直区间缩放动画.....	76
16.4	旋转.....	76
16.4.1	旋转启停动画.....	76
16.4.2	区间旋转动画.....	77
16.4.3	线性旋转动画.....	78
16.5	移动.....	79
16.5.1	移动动画.....	79
16.6	过渡.....	80
16.6.1	过渡动画.....	80
16.7	线条.....	80
16.7.1	线条动画.....	80
16.7.2	流动动画.....	81
16.8	文本.....	83
16.8.1	文本值显示动画.....	83
16.8.2	区间值显示动画.....	83
16.8.3	文本颜色动画.....	83
17	步骤十、发布工程至运行平台	84
18	步骤十一、调试监控页面	86
18.1	模拟调试.....	86
18.2	实时快照.....	86
19	第三方程序如何发送数据至 HawkRuntime.....	89
20	如何为监控画面上的设备增加右键菜单项.....	91

第一篇 原理篇

1 软件简介

Hawk 物联网应用平台是完全基于 JAVA 实现的新一代组态软件，其将最先进的 IT 技术和设计理念（如 SVG、HTML5）引入到自动化软件领域，引领了新一代组态软件的技术趋势，可以快速构建以实时智能监测为主要功能的物联网应用。



Hawk 物联网应用平台开创性的实现了“面向对象（设备）”的组态理念，每个“设备”都对应一个具体的物理设备，包含了该物理设备的连接信息、设备属性、报警规则、图形等所有相关信息。完全以“设备”为中心，使得平台具有极强的灵活性和扩展性，工程实施效率大大提高，为物联网技术在各行业中的快速应用(物联网+)提供技术保障。

Hawk 物联网应用平台分为开发工具（HawkBuilder）和运行时（Runtime）两部分。技术领先、引领未来，除了组态工具的基本功能之外，其“八大”创新特性为：

1、跨平台：分为“服务端跨平台”和“客户端跨平台”。服务端依赖 JDK 屏蔽平台差异，系统可部署于 Windows、Unix/Linux 等操作系统之上；客户端依赖浏览器屏蔽平台差异，只需支持 HTML5 规范的浏览器就可以在 PC、平板电脑、智能手机等异构平台上浏览系统功能。

2、零客户端：平台客户端遵从 HTML5 技术规范，100%纯 B/S 结构，只需要支持 HTML5 规范的浏览器即可，无需安装任何插件及客户端程序。

3、面向对象(设备)：平台以“设备”为中心，每个“设备”都由某个“设备类型”实例化而来。组态开发过程中的数据采集、脚本编写、报警识别、图形页面都是针对某个“设备”进行的。

4、设备库：或称“设备类型库”，分为“系统设备库”和“用户设备库”。一个设备类型是对某类物理设备的抽象定义，包含设备的连接信息、设备属性、报警规则、图形等所有相关信息。丰富的设备库将大大提高组态开发的效率和质量，支持用户自定义新的设备类型。

5、SVG 矢量图：与大部分组态工具采用自定义图形格式不同，Hawk 物联网应用平台的图形采用的是 SVG（Scalable Vector Graphics 可伸缩矢量图形），是一种由 W3C 制定的开放式标准图形格式，也是 HTML5 标准协议簇的一部分，具有体积小、易读写、无损缩放等优势。可直接使用 illustrator 等专业作图工具绘制监控画面，画面专业美观，极大提高用户体验度。

6、动画：直观形象的动画是提高用户体验度的重要手段。平台经过简单配置就能完成闪烁、隐藏、填充、旋转、移动、流动等高级动画效果，囊括了几乎所有的页面动态效果。

7、设备状态：平台支持为每个设备定义任意多的状态。每个状态由状态码、状态等级、状态成立条件、状态成立后执行的动作四部分组成，其中的设备状态按重要程度分为蓝色预警、黄色预警、橙色预警和红色预警四个等级。支持设备“挂牌”操作，手动设定设备检修、设备停运等状态。

8、“7 语言”脚本：脚本是组态工具的重要部分，简洁易懂的脚本可以降低对用户的技术要求。其他组态工具的脚本模块大多采用 VBScript、JavaScript 或 C#，工程人员完全掌握这种“大而全”的脚本并非易事。“7 语言”精心设计，语法简单，功能强大，是一种“面向对象”的轻量级脚本语言，只需要 7 分钟就可以学会的脚本语言。

2 “十大”经典问答

1、Hawk 物联网应用平台是一种组态工具吗？与 WinCC、组态王等老牌组态工具相比有什么不同？是否属于重复建设？

答：单纯从产品功能的角度来看，Hawk 物联网应用平台与 WinCC 等组态工具非常类似，两者的差异在于产品定位的不同。WinCC 等组态工具在工业自动化控制领域已经用了几十年，在工厂车间级自动化系统中其地位不可撼动。然而近年来随着物联网技术的发展，智能建筑、智能家居、智慧城市、精细农业等新鲜名词不断涌现，人们希望远程监控的对象越来越多，数据传输网络也比工业以太网更加多样复杂，监控客户端设备也需要支持移动设备、电视、个人电脑等多种设备终端，这就要求组态工具具备灵活的部署方式，较强的开放集成性，更好的界面友好性等特征，这也正是 Hawk 物联网应用平台需要完成的事情。

所以，Hawk 物联网应用平台就是我们在多年从事工业控制工作经验的基础上，利用最先进的信息技术，对在工控领域成熟运用多年的组态工具进行升级改造，面向比工业控制领域范围更广的物联网应用领域提供实时智能监测产品和解决方案，并非重复建设。在工业自动化集成领域，两者甚至是一种共存关系，Hawk 物联网应用平台面向工业调度对多个离散的基于 WinCC 等组态工具实现的自动化系统进行集成整合，进而实现全工厂的综合自动化集中监测系统。

2、听说平台是基于 JAVA 实现的，那是“跨平台”的吗？在 Unix、Linux 等服务器环境下是否可部署运行？

答：目前市场上的其他产品大部分都是基于微软的技术实现的，产品与 Windows 系统紧密捆绑，即使有 Linux 等其他操作系统版本，也大多要维护几份各自针对不同操作系统的产品源代码。Hawk 物联网应用平台 100% 是基于 JAVA 实现的，甚至连 OPC Client 都完全基于 JAVA 实现，所以完全是“跨平台”的，同一个安装程序可以部署在 Windows、Linux 等多种操作系统上，其实际是由 JDK（JAVA 虚拟机）屏蔽了底层系统的差异，产品本身与操作系统松耦合。

另外需要指出的是，一提到“跨平台”，人们往往会只认为是针对服务器端的跨平台，这大概是因为之前客户端被 Windows 牢牢垄断，用户对于客户端别无选择的结果。随着移动终端的兴起，这种局面慢慢在改变，客户端的跨平台其实更加重要，程序员们花了很大的精力在解决怎么去适配不同终端，不同浏览器的问题。市场上的其他组态产品会针对不同的客户端研发不同的监控客户端软件，如 PC 客户端、安卓客户端、iOS 客户端等。

Hawk 物联网应用平台不仅解决了服务端的跨平台，而且也根本解决了客户端的跨平台。服务端的跨平台依靠 JDK，客户端的跨平台依靠浏览器，由遵从 HTML5 规范的浏览器屏蔽客户端的平台差异，进而真正实现客户端的跨平台。所以 Hawk 物联网应用平台完全是轻量级的，“零客户端”，只要个浏览器，浏览器也无需安装任何插件就可以了。

3、听说客户端是“零客户端”，只要浏览器就可以了，不需要安装任何浏览器控件或客户端程序，是真的吗？市场上这么多种浏览器，任何一种浏览器都可以吗？好多企业用户现在还用着 IE7、IE8 等老浏览器，也不需要安装什么客户端程序吗？

答：Hawk 物联网应用平台客户端的实现技术是 HTML、SVG、JavaScript，没有 ActiveX 控件，没有专门针对某一种浏览器进行扩展，客户端完全基于 HTML5 规范，只要是遵从 HTML5 规范的浏览器都可以使用。

总之，我们的基本想法就是不去开发私有的客户端程序，完全依靠浏览器。有的客户也说，开发个移动 APP 看监控画面也挺方便的，不一定非要用浏览器。我们的观点是限于移动终端屏幕大小的限制，移动终端看监控画面还是太小了，但移动 APP 倒是可以用来进行系统报警通知信息的收发，就像我们用 QQ 收发聊天消息，其他的新闻、图片收到链接后还得通过浏览器打开观看。

目前市场上的主流浏览器除了 IE7、IE8 基本都实现了 HTML5 规范，但不可否认各种浏览器对 HTML5 规范的实现确实参差不齐，而且 IE7、IE8 两款非 HTML5 浏览器还有很大的用户群（特别是在工业用户中）。所以对 HTML5 浏览器和非 HTML5 浏览器的处理方式大致为：

HTML5 浏览器：在所有实现 HTML5 规范的浏览器中推荐使用 Chrome

IE7、IE8：这两款浏览器在工业领域还有很大的用户群，特别是在有视频监控系统的情况下，基本都离不开 IE 浏览器。对它们的处理方法一种是通过一些第三方的扩展程序，将 IE 改造成 HTML5 浏览器，该方法需要安装 IE 扩展程序；另一种方法是 Hawk 物联网应用平台对每个监控画面都可以同时生成 HTML5 和非 HTML5 两个版本，根据浏览器版本的不同，访问不同的 URL 进行监控画面的浏览，该方法 IE 客户端需要安装 Adobe SVG Viewer 插件。

4、如果已有一个 Web 工程，可能是基于 JAVA、C#、PHP 等其他程序语言实现的，想在该工程中通过 Hawk 物联网应用平台实现几幅监控页面，可行吗？如果可行的话，该怎么做？

答：Hawk 物联网应用平台的运行时实际是一个标准的 Servlet，遵从 J2EE 规范，是一个普通的 JAVA Web 工程，可以部署于 Tomcat、Jetty、WebLogic 等应用服务器下。

与基于 JAVA 实现的 Web 工程的融合是天然的、无缝的，在标准授权下两者可以分别作为单独的 Webapp 部署于同一个 tomcat 下；在特殊授权下，Hawk 物联网应用平台的运行时可以 jar 包的形式完全融合到其他 Web 工程中。

与基于 C#、PHP 实现的 Web 工程的融合因为技术路线的不同，无法做到代码级、部署级的融合，只能做到数据级和界面级的融合。通过 Hawk 物联网应用平台提供的接口，其他 Web 工程可以实现与 Hawk 物联网应用平台的数据读

写交互；Hawk 物联网应用平台的每个监控页面仅仅为一个普通网页，其他 Web 工程可以通过 iframe 等技术手段在自己的页面中嵌入监控页面，实现两者界面级的融合。

5、什么样的设备才能接入到 Hawk 物联网应用平台中呢？对接入设备有什么具体要求呢？能对接入设备进行控制吗？

答：只要是具备与外界交互能力的设备都可以接入到 Hawk 物联网应用平台中，只是根据设备采集协议规范性的差异，设备接入的复杂度略有不同。大概可以分为三类：

1) 标准化协议：比如在工业领域广泛采用的 OPC 协议，这类协议的设备接入简单，顺利的话可能几分钟就可以搞定。

2) 半标准化协议：比如 Modbus、TCP/UDP 协议，这类协议与设备建立起连接后，需要根据设备厂家提供的协议文档进行数据包的解释，需写少量的程序。

3) 第三方私有协议：由设备厂家自己定义的设备协议，没有遵从任何技术规范，比如有的设备会定期生成一个数据文件，需要定时解释这个私有协议文件读取设备数据，需完全为该设备定制开发采集程序。

只要设备支持控制协议接口，都可以进行设备的控制。

6、听说 Hawk 物联网应用是“面向对象（设备）”的，这一点确实与 WinCC 等组态工具有很大不同，怎么理解这个“面向对象（设备）”？

答：在其他组态工具中，主要是以“点”（也有的称为“位号”）为中心，数据存储是以“点表”的方式，业务报警是基于“点值”的判断，图形动画是绑定的“点名”，这种情况下，“设备”仅仅是一个“组”的概念，是由几个“点”组成的虚拟组，比如一个温湿度传感器，就表示为两个“点”，一个点表示温度，一个点表示湿度。

Hawk 物联网应用平台是以“设备”为中心的，每个设备除了属性(Property)之外，还可以有自己的业务处理方法(Method)，对应于程序语言中的一个类(Class)，对于上述一个温湿度传感器的描述为“一个温湿度传感器对象，有两个属性，一个是温度，一个是湿度”。

“面向对象”或者叫做“面向设备”的好处是可以将每个设备的连接信息、业务逻辑、图形外观融合成一个统一的整体，这也是 Hawk 物联网应用平台中“设备库”的构建基础，面对物联网应用领域众多的设备种类，只有“面向对象（设备）”的设计理念才可从容应对。

所以“点”这个概念在 Hawk 物联网应用平台中已经完全消失，组态的过程就是从“设备库”中选取合适的设备类型进行实例化，形成该工程的“设备实例列表”，再从设备实例列表中选择设备进行监控页面的绘制即可，非常的快捷高效。

7、用户自己是否可以定义设备类型？在制作设备类型库的过程中，贵公司是否提供必要的技术支持？用户自己定义的设备类型，是否可以在不同工程之间复用？

答：Hawk 物联网应用平台是“面向对象（设备）”的，是以“设备”为中心的，每个设备都是从某个“设备类型”实例化而来的，“设备类型”和“设备”的关系其实就是类(Class)和对象(Object)的关系。

“设备库”中的设备类型分为“系统设备类型”和“用户自定义设备类型”两部分，用户可以自己根据项目的实际需求定义新的设备类型。每个设备类型，无论来自用户自定义还是系统内置的，都是可以在不同工程间共享的。实际对于每个行业，虽然觉得设备多种多样，异常复杂，其实仔细抽象建模，任何一个常规行业能够用到的设备类型也就几十种，同一类型的不同设备往往仅仅是外观上的差异，所以每个设备类型可以定义多个设备外观视图，方便实际项目的针对性选择。

其他组态工具中的“图库”，其中的每个组件仅仅是一张图片而已；而 Hawk 物联网应用平台中的一个“设备类型”是 MVC 结构，是由数据(M)、业务逻辑(C)、视图外观(V)组成的整体。

在知识产权双方共享的前提下，我公司免费给客户制作具有通用价值的设备类型。

8、好多项目都要在看监控画面的同时需要查看视频，请问该平台能集成视频监控系统吗？怎么集成？

答：准确的说法，Hawk 物联网应用平台和视频监控系统的地位是一样的，不存在谁集成谁的问题，因为 Hawk 物联网应用平台中的一副监控页面和视频监控系统中的一路视频信号都是一个普通的 HTML 网页，一般在实际项目都会有一个类似 portal 门户的框架通过菜单的合理组织，使得用户在同一个系统中同时浏览监控画面和视频监控画面，从而将两者“集成”在一起。所以既不是 Hawk 物联网应用平台集成了视频监控系统，也不是视频监控系统集成了 Hawk 物联网应用平台，而是两者都具备被第三方框架集成的优势。

9、现在跟前几年不同，用户对监控画面美观度的要求大大提高，仅仅依靠工程实施人员很难组态出漂亮的监控画面，据说 Hawk 物联网应用平台采用的是 SVG 矢量化图形，能组态出比其他组态工具更美观的监控图吗？

答：在功能满足的前提下，最终用户对界面美观度的期望值越来越高，特别是面向大众服务的应用场景，如果还像在工业控制领域那样由实施工程师来随意组态“丑陋”的监控画面，已经很难满足用户的需求。为了获得漂亮美观的图形，必需专业美工的参与。

所有的组态工具都会提供监控画面的图形编辑手段，但其图形编辑能力与 Photoshop 等专业图形工具相比差距很大，只能供工程人员进行简单的图形编辑，专业美工也很难通过这么简单的工具作出多么漂亮的作品。另外，因为两者图形格式的不同（组态工具的图形格式一般为自定义的私有文件格式），两种图形编辑工具很难直接编辑对方的图形格式，在这种情况下，即使有美工的参与，往往也仅是局部的效果提升。

Hawk 物联网应用平台的图形格式采用的 W3C 的规范格式: SVG, 与 photoshop 齐名的 illustrator 是专业的 SVG 矢量化图形编辑工具，Hawk Builder 中的图形编辑模块也是一个 SVG 图形编辑工具。相同的图形格式使得实施工程师和专业

美工可以协同工作，美工通过 illustrator 作出的图形，实施工程师可以直接在 HawkBuilder 中打开，继续组态。

另外，丰富的设备库中每个设备的视图部分都由专业美工提前设计好了若干个视图样本，实施工程师大部分情况下只需要选择使用即可。我们的目标就是尽量不要让实施工程师从头开始用线、矩形、曲线这些基本工具绘制图形。

10、以前的监控页面主要都是以简单的数字变化为主，再加上几个颜色变化表示设备的启停状态，除此之外，页面的大部分都是静态的，整个页面显得很呆板；听说 Hawk 物联网应用平台能够作的动画效果很丰富，是真的吗？需要会写程序才能作出这么多生动的动画效果吗？

答：在制作页面动画效果方面，Hawk 物联网应用平台与其他组态工具相比，具有突出的优势。其他组态工具往往是采用 GIF 图片的方式，或者是多个图片不停切换的方式实现动画效果，不仅制作复杂，效果上也大打折扣。

在 Hawk 物联网应用平台中实施工程师通过简单的配置就可以实现移动、闪烁、流动、填充等动态效果，技术要求低，仅仅要求会写简单的“7 语言”脚本即可。从零起点开始，几个工作日内即可学会所有页面效果的制作。

3 名词解释

3.1 设备类型

设备类型是对物理设备的抽象定义，包括设备属性、业务逻辑和外观等。比如对于市场上不同厂家生产的温湿度传感器，可以抽象定义成一个“温湿度设备类型”，定义大致如下所示：

- 1、两个 float 类型的属性 temperature(温度)、humidity(湿度)
- 2、温度高于设定值发出“高温”预警信号
- 3、指定若干个不同厂家生产的温湿度设备外观图

Hawk 物联网应用平台中的设备类型包括系统内置设备类型和用户自定义设备类型两种。

例：智能交通领域中的“车速检测仪（VD）”设备，定义如下：



3.1.1 概述

车速检测仪

概述: 车速检测仪

基本属性

名称、应用行业等相关信息

类别:

ID:

名称:

适用行业:

标签:

使用说明

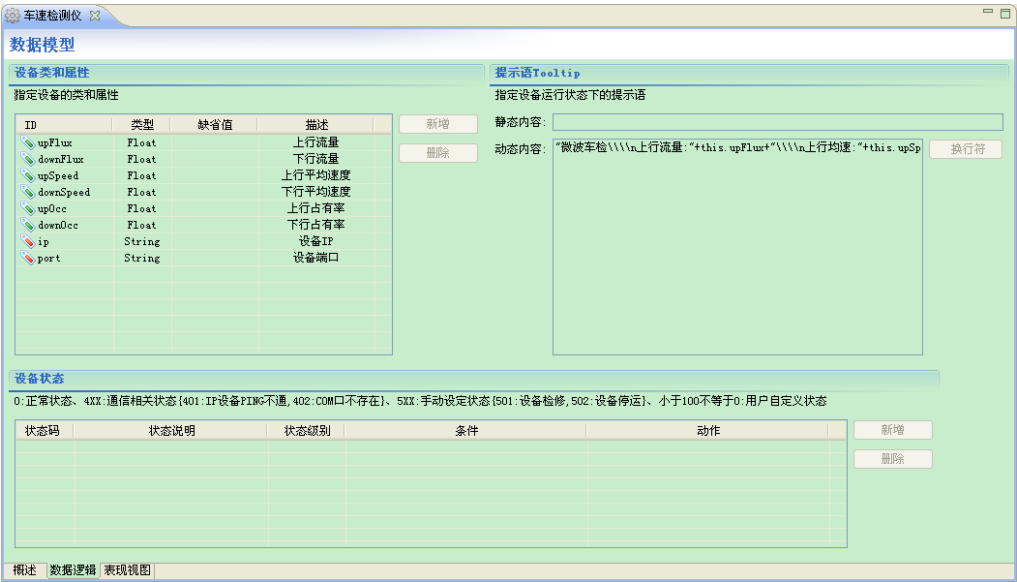
该组件的开发者说明文档

概述 数据逻辑 表现视图

- 类别: 类似于 JAVA 语言中的 package, 防止不同用户定义设备的名称冲突, 推荐采用本公司的域名, 从而保证全球唯一性
- ID: 设备的英文简称
- 名称: 设备的中文名称
- 适用行业: 标识设备的适用行业, 方便按行业检索
- 标签: 定义设备的特性标签, 方便按 Tag 进行检索

- 使用说明：该设备的使用帮助文档

3.1.2 数据逻辑

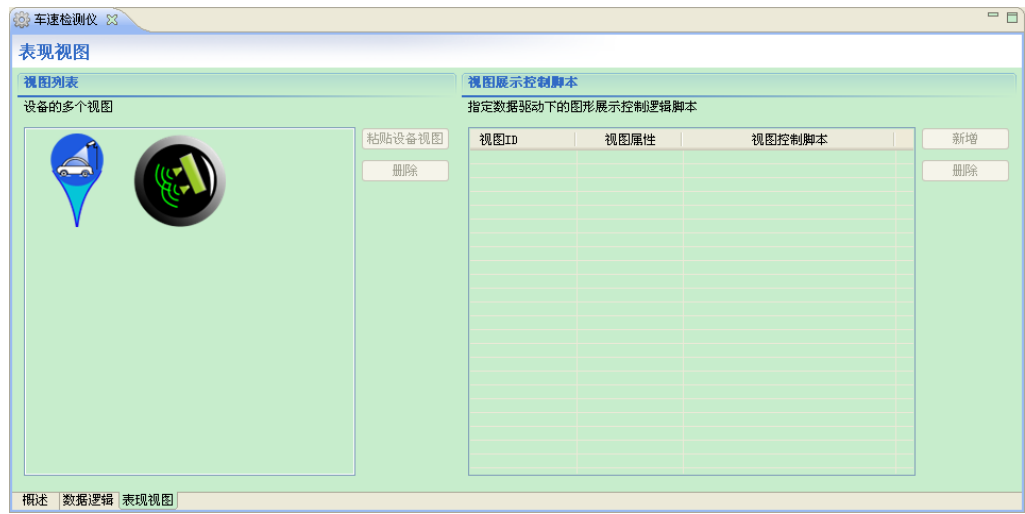


- 设备属性：包括上行流量、上行平均速度、上行占有率等属性，属性类型包括整型、浮点型、字符型、布尔型等
- 提示语：“静态内容”为一些静态信息，比如安装位置、设备 IP 等，监控画面初始状态下显示该内容（如果配置了该内容）；“动态内容”为随着设备监控数据实时变化的提示信息，比如在 Tooltip 上实时显示设备的观测数据，“动态内容”更新时会覆盖掉“静态内容”。



另外可以通过脚本的方式指定状态发生的条件以及状态发生后执行的动作。

3.1.3 表现视图



- 设备视图：每种设备因使用场合和生产厂家的不同，会有多种外观
- 视图展示控制脚本：以“7 语言”定义设备数据与视图的互动逻辑，及定义设备属性值变化后图形如何通过颜色、动画等形式展示。

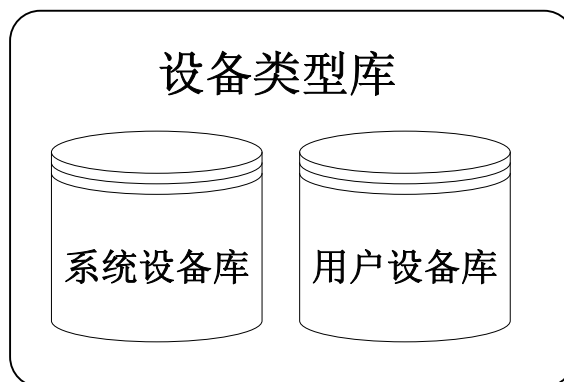
3.2 设备

设备，或称“设备实例”，是某个“设备类型”在某个具体工程中的实例化。该设备实例初始时的属性、状态定义、tooltip 等均来自于“设备类型”中的定义，并可以根据项目的实际需求进行定制修改（设备属性除外）。

3.3 设备列表

某个工程的“设备列表”是该工程中所有设备的集合。

3.4 设备库



设备库是 Hawk 物联网应用平台的重要资源，汇集各个行业、各种类型的智能设备，分为系统设备库和用户设备库两部分。系统设备库为内置于平台中，供用户使用的设备资源；用户设备库提供用户自定义私有设备类型。

与传统组态工具中的图库不同（传统组态工具的图库只是图片），设备库中的每个设备类型由数据、业务逻辑、图形展示三部分组成，即“MVC”结构，完全以面向对象的方式定义设备。

4 工程结构

每个 Hawk 物联网应用工程包括 I/O 通信、设备列表、业务逻辑、监控页面四部分组成。

4.1 I/O 通信

实现数据源的管理。按照协议的标准化程度，数据源可分为如下三类：

1、**标准化数据源**：数据源采用的数据传输协议完全符合业界技术规范，如 OPC 数据源。

2、**半标准化数据源**：数据源提供的数据需要进行二次解释才能利用，如 TCP/UDP 的数据包，Modbus 等。

3、**私有协议数据源**：数据源完全采用自定义的 API 数据访问接口，常见于年代较早的自动化系统和在市场上处于垄断地位的设备。

在项目实施过程中，需要采用中医的“望闻问切”，全面分析数据源的特点，确定数据源协议的标准化程度，从而确定最终的数据采集方案。

通过必要的定制开发，Hawk 物联网应用平台可以接入所有数据源，换句话

说，除非是铁板一块，不然都有办法将数据源接入 Hawk 物联网应用平台。

按照数据源采集数据的主从性，数据源可分为两类：

1、“拉”数据源：数据源处于被动地位，由 Hawk 物联网应用平台根据设定的定时器，定时采集数据库的数据。该类数据源的实时性会大打折扣，并且会增加不必要的网络传输流量，比如关系数据库。

2、“推”数据源：数据源处于主动地位，或称数据源采用“订阅/发布”机制，由 Hawk 物联网应用平台订阅感兴趣的数据源数据，当被订阅的数据发生变化时，数据源会主动将变化的数据发布给 Hawk 物联网应用平台。该类数据源具有很好的实时性，并且避免了不必要的数据传输流量，比如 OPC Server。

4.2 设备列表

设备列表管理工程中的所有设备实例。每个设备实例均由设备库中某个设备类型实例化而来。设备列表中的每个设备都需要有唯一的 ID 标识。

4.3 业务逻辑



业务逻辑主要以“7 语言”脚本实现，用以实现项目实施过程中的复杂业务需求。

5 设备状态

每个设备可以定义多个状态，定义方式类似于 HTTP 协议的状态码。状态码约定如下表所示：

分类	区间	状态码	状态说明
系统内置	0	0	正常状态
	4** 通信相关	400	通信故障
		401	设备 PING 不通
		402	COM 口不存在或异常
	5** 手动设定	501	设备检修
		502	设备停运
用户自定义	$(-\infty, 100)$ 且不等于 0	$(-\infty, 100)$ 且不等于 0	用户可以在小于 100 且不等于 0 的范围内任意定义设备状态

设备状态按重要程度分为蓝色预警、黄色预警、橙色预警和红色预警共四个等级。

报警级别	颜色对应
一般	
较重	
严重	
特别严重	

6 “7 语言”

7 语言，只需要 7 分钟就可以学会的脚本语言。

7 语言是一种面向对象的脚本语言，简单高效，其语法规则如下所示：

6.1 变量类型

■ 三种变量类型：数字型、字符型、布尔型

- **动态类型:** 变量类型是隐含的, 声明一个变量, 直接赋值即可。比如 `n=2;`
`n=2.1;` `n=true;` `n=“a”`。

6.2 运算符

- **算术运算符:** `+`、`-`、`*`、`/`、`%` (模运算)
- **字符串的拼接:** `.` 例如: `c="hello"+"world". "!"` 等价于 `c="hello world!"`
- **逻辑运算符:** 逻辑非 (`!`), 逻辑与 (`&&`), 逻辑或 (`||`)
- **关系运算符:** 大于 (`>`)、大于等于 (`>=`)、小于 (`<`)、小于等于 (`<=`)、等于 (`==`)、不等于 (`!=`)

6.3 流程控制

- `if() { }`
- `if() { }else{ }`

6.4 其他

- `return` 语句
- 不支持代码注释, 代码大小写不敏感
- `$` 表示当前变量, 常用于变量表达式

7 运行时

与传统组态工具的组态开发时和运行时大都在一起不同, Hawk 物联网应用平台的运行时单独存在。HawkRuntime 遵从 J2EE 的 Servlet 规范, 整个 HawkRuntime 为一标准 war 包, 可以部署在 Tomcat、Jetty、JBoss 等应用服务器中。

Hawk 物联网应用运行平台的目录结构如下:



- 1、assets:平台运行时所需要的资源文件，如 css、js、模版文件等。
- 2、context: 平台运行时所需要的上下文文件，该目录用户一般不用关注
- 3、publish:存放 HawkBuilder 发布的工程文件(*.har)。
- 4、HawkDB:平台运行时内置的实时数据库，该目录用户一般不用关注
- 5、userLibs:存放用户自定义的设备类型描述文件。
- 6、view:监控画面的存放路径，每个监控画面都会生成一个 jsp 文件。
- 7、machineid.txt:产品的注册申请文件

工程发布后生成 har 文件，将该文件上传至 publish 目录下，重启运行平台即可。每个监控页面的访问 url 形式如下所示：

<http://ip地址:端口号/hawk/view/监控画面名称.jsp>

如：<http://127.0.0.1:8080/hawk/view/test.jsp>

其中的 IP 地址，端口号以实际部署环境为准，test.jsp 代表的监控画面名称以实际监控画面为准

第二篇 实操篇

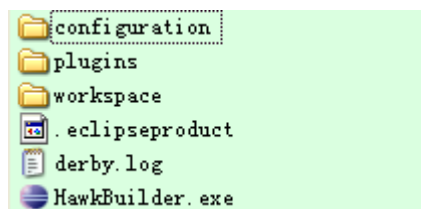
8 步骤一、产品的安装部署

Hawk 物联网应用平台分为集成开发环境 HawkBuilder 和运行时 HawkRuntime 两部分，均基于 JAVA 实现，只需要安装 JDK 就可以运行于 Windows、Linux、Unix 等不同操作系统上；对 CPU 和内存的要求依据监控规模和业务复杂度而定，大多数场合下主流 CPU、大于 2G 内存就可以满足要求。对客户端要求只需要有浏览器即可（推荐采用支持 HTML5 的浏览器，如 Chrome）。

8.1 开发环境 HawkBuilder 的安装部署

1、安装 JDK7+以上，下载地址为
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>（如已安装 JAVA 虚拟机环境，则可以略过此步骤）

2、直接解压 HawkBuilder.zip 至任意目录下



3、双击 HawkBuilder.exe，启动 Hawk 物联网应用平台的集成开发环境



8.2 运行时 HawkRuntime 的安装部署

8.2.1 服务器端

8.2.1.1 硬件要求

处理器： 1 GHz 或更快

内存： 1 GB RAM (32 位) 或 2 GB RAM (64 位)

硬盘空间： 16 GB (32 位) 或 20 GB (64 位)

8.2.1.2 软件要求

操作系统： Windows、Unix、Linux

JDK 7+

Tomcat、Jetty、WebLogic、JBoss 等

8.2.1.3 安装步骤

1、安装 JDK7+以上，下载地址为
<http://www.oracle.com/technetwork/java/javase/downloads/index.html> (如已安装 JAVA 虚拟机环境，则可以略过此步骤)

2、安装 Tomcat 等应用服务器，Tomcat 下载地址为
<http://tomcat.apache.org/>

3、将 Hawk.war 文件复制到 Tomcat 的 webapps 目录下，启动 Tomcat 即可

8.2.2 客户端

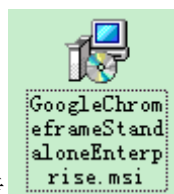
平台客户端遵从 HTML5 技术规范，100%纯 B/S 结构，只需要支持 HTML5 规范的浏览器即可，无需安装任何插件及客户端程序。

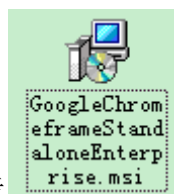
推荐使用 Chrome 浏览器。

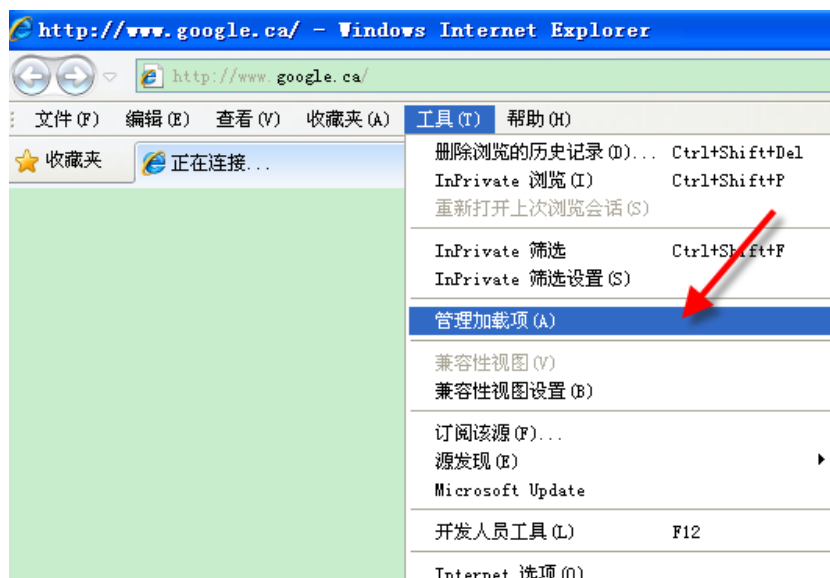
8.2.2.1 浏览器支持

分类	浏览器名称	说明
 IE 浏览器 (非 HTML5)	IE6	需安装第三方插件
	IE7	需安装第三方插件
	IE8	需安装第三方插件
	IE9	需安装第三方插件
	IE10	本地支持，“0” 安装
	Firefox	本地支持，“0” 安装
	Chrome	本地支持，“0” 安装，推荐方式
	Safari	本地支持，“0” 安装
	Opera	本地支持，“0” 安装

8.2.2.2 IE 第三方插件安装



鼠标双击  图标，启动插件安装。安装过程均采用缺省配置，打开 IE 浏览器的“管理加载项”



保证 ChromeFrame 加载项的状态为“已启用”状态（如果为“禁用”状态，可手动改为“已启用”状态）




插件安装完毕

9 步骤二、创建新的设备类型


在项目实施之前，必需为项目中的每类设备定义设备类型，工程创建后再通过该设备类型实例化某个具体的物理设备。Hawk 物联网应用平台的“设备库”中是平台内置的设备类型，用户可以直接使用。如果无法在“设备库”中找到合适的设备类型，则需要用户自定义新的设备类型。

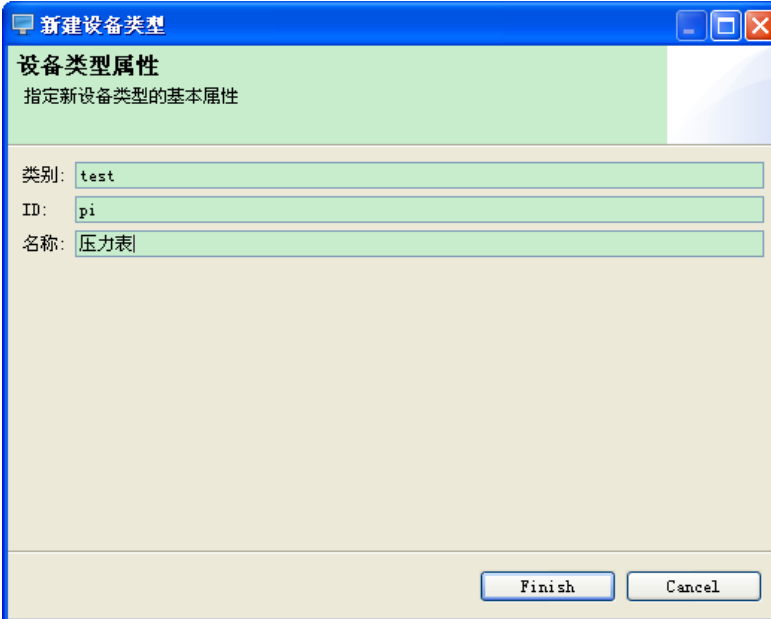
用户自定义设备类型，定义步骤如下所示：

9.1 创建设备组

- 1、点击  图标，在“设备库”的根目录下或者选中的某个组下，创建一个“设备组”
- 2、右击某个组，在快捷菜单中选择“新建组”命令，在该组下创建一个组。
- 3、右击某个组，在快捷菜单中选择“重命名”命令，可以重命名该组。
- 4、右击某个组，在快捷菜单中选择“删除”命令，可以删除该组。

9.2 创建用户设备

- 1、单击  图标，打开“新建设备类型”窗口
- 2、在“新建设备类型”窗口中，输入如图所示的类别、ID 和名称信息



新建设备类型

设备类型属性
指定新设备类型的基本属性

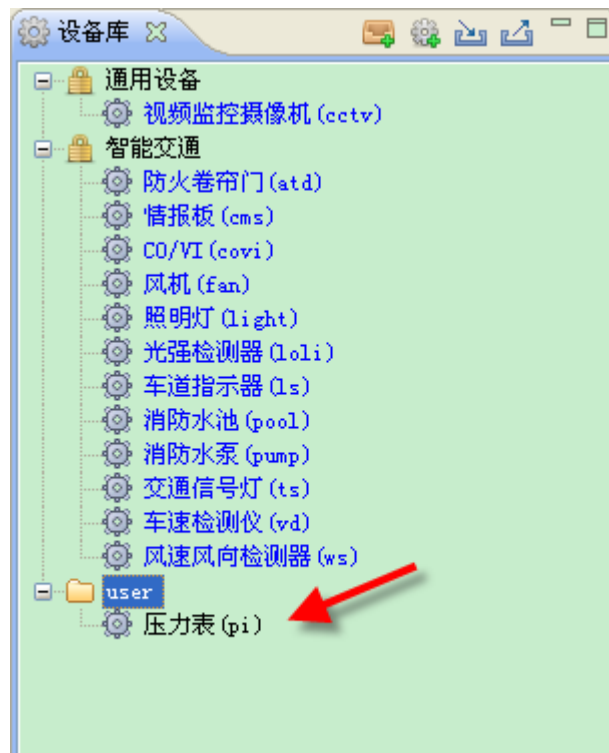
类别: test

ID: pi

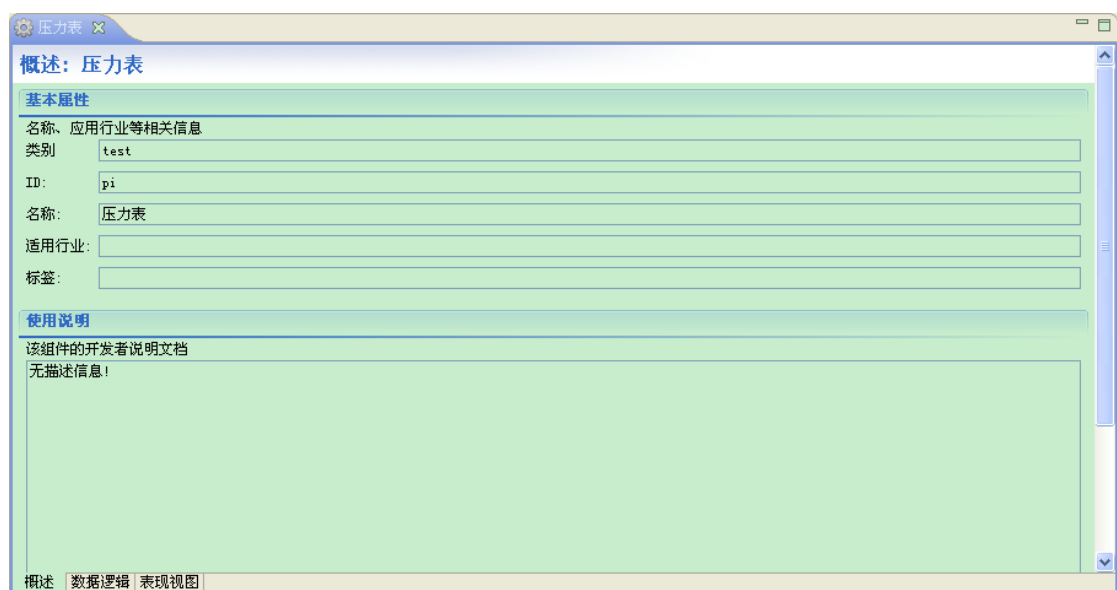
名称: 压力表

Finish Cancel

- 3、单击“确定”按钮创建该设备。



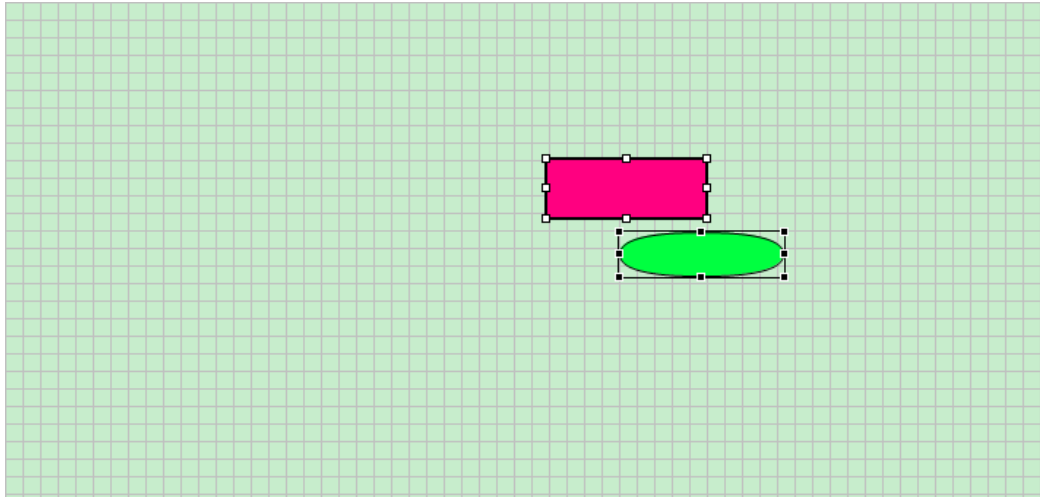
4、双击该设备，打开该设备的编辑器。



5、选填适用行业、标签、使用说明等信息

6、在“数据逻辑”Tab 页中定义一个类型为 Float 的 PV 属性

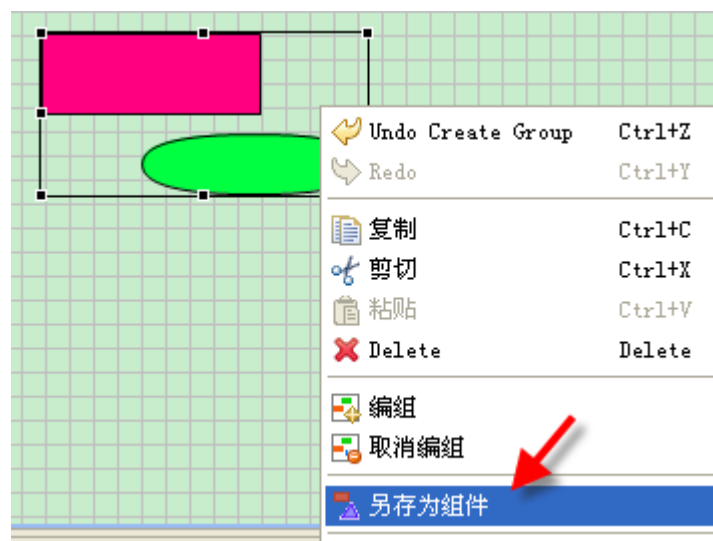
7、在图形编辑器中绘制该设备视图，简单起见，假设该设备就是一个矩形和椭圆的组合。



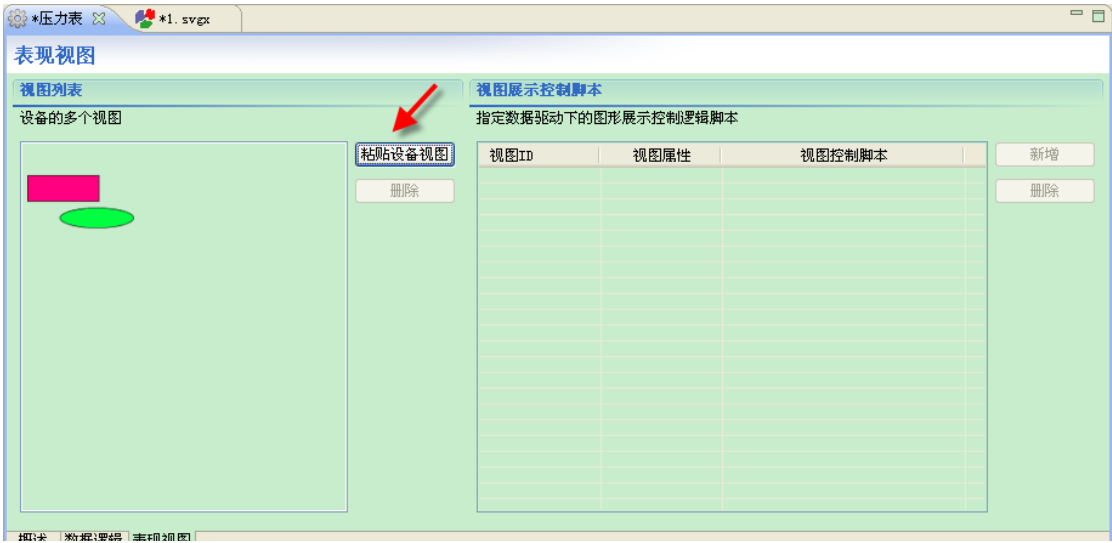
8、将设备进行“编组”操作



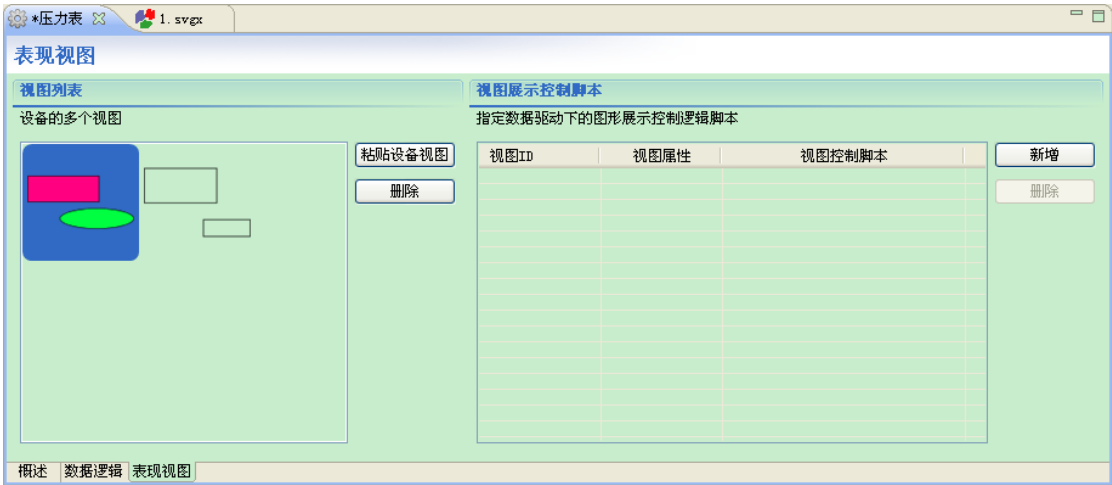
9、右键该组选择“另存为组件”命令




10、在“表现视图”中单击“粘贴设备视图”按钮，创建设备视图




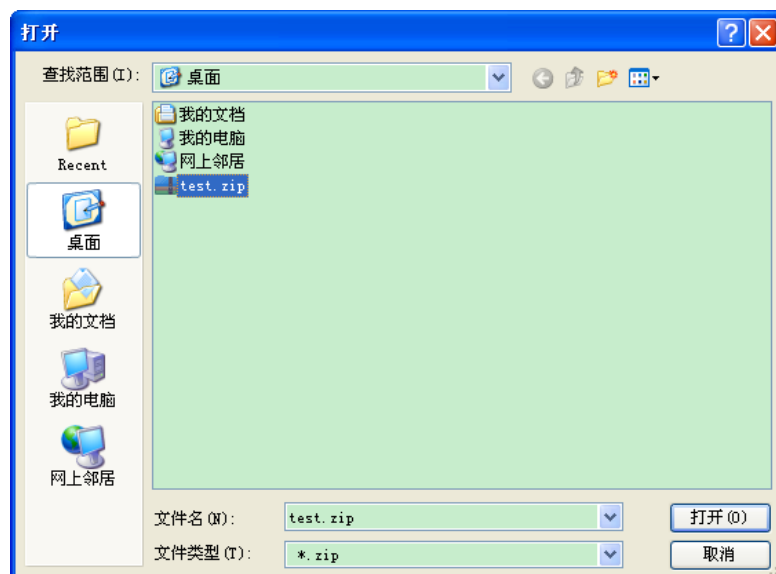
11、按照上述方法，可以为设备创建多个视图



12、单击  按钮，保存该设备信息。


9.3 导入用户设备

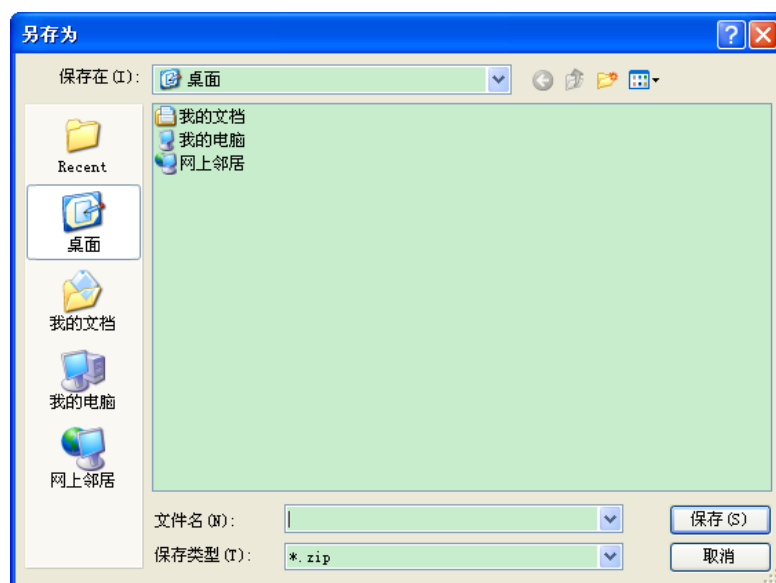
1、单击  图标，打开“打开”窗口



2、选中要导入的用户设备库文件，单击“打开”按钮，导入该用户库。

9.4 导出用户设备

1、单击  图标，打开“另存为”窗口

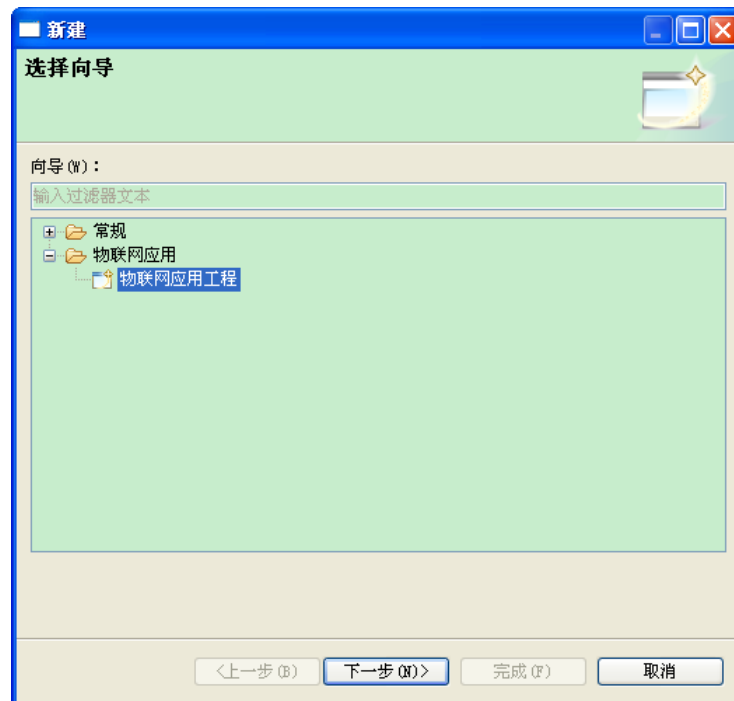


2、指定“文件名”，单击“保存”按钮，导出用户设备。

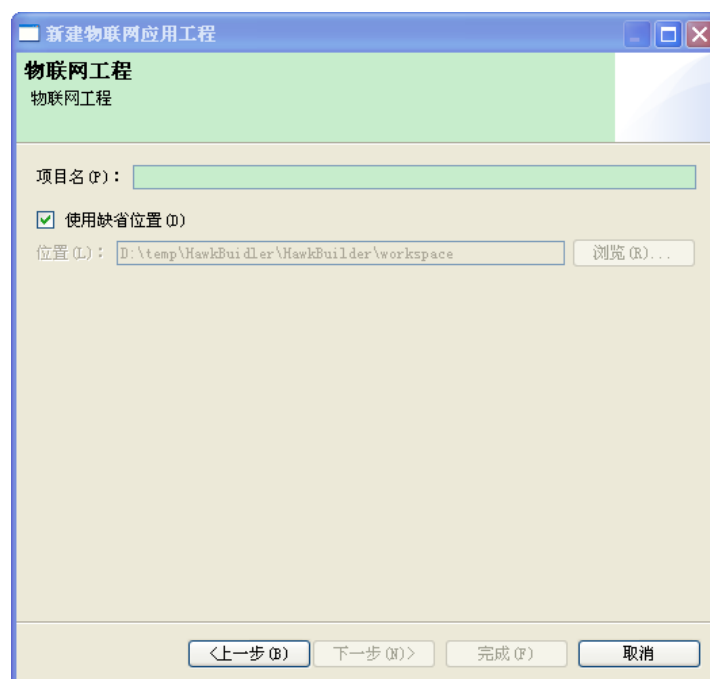
10 步骤三、创建物联网应用工程

10.1 新建工程

- 1、在“文件”菜单中选择“新建”命令，打开“新建”向导对话框。



- 2、选择“物联网应用工程”，单击“下一步”按钮，进入“新建物联网应用工程”对话框。

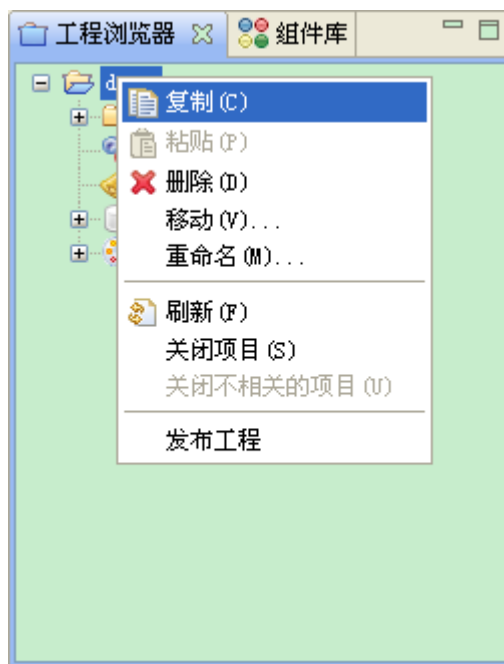


- 3、在“项目名”文本框中输入工程名称，并指定工程的存放路径，单击“完

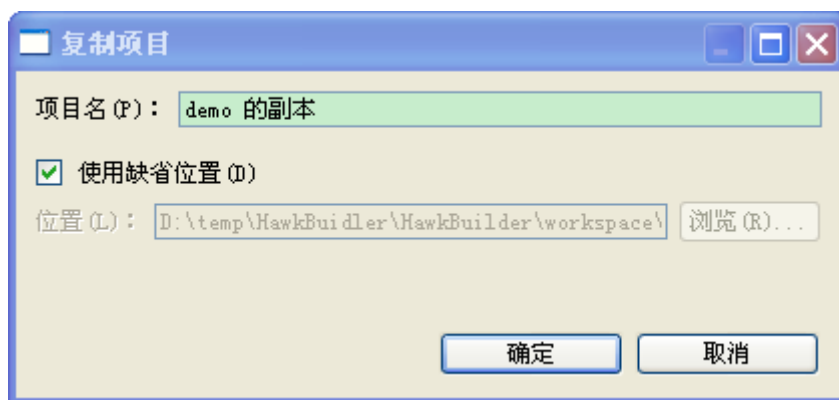
成”按钮，完成物联网应用工程的创建。

10.2 复制工程

1、在工程浏览器中右击任一物联网应用工程，在快捷菜单中选择“复制”命令，实现工程的复制。



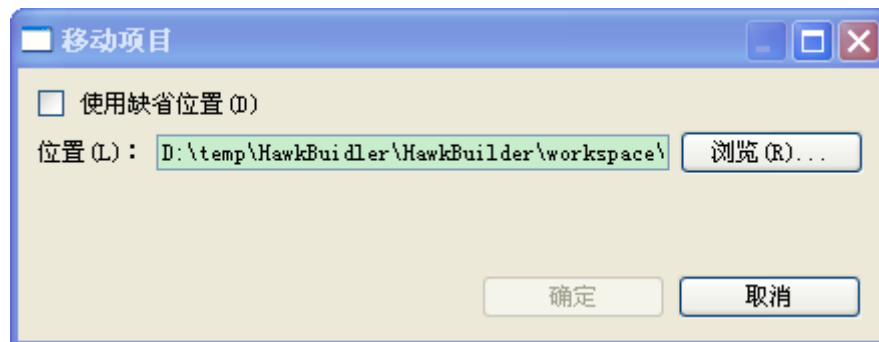
2、在工程浏览器中的任意位置点击鼠标右键，在快捷菜单中选择“粘贴”命令，打开“复制项目”对话框。



3、在“复制项目”对话框中输入项目名称，并指定工程的存储位置，点击“确定”按钮，实现工程的复制操作。

10.3 移动工程

1、在工程浏览器中右击任一物联网应用工程，在快捷菜单中选择“移动”命令，打开“移动项目”对话框。



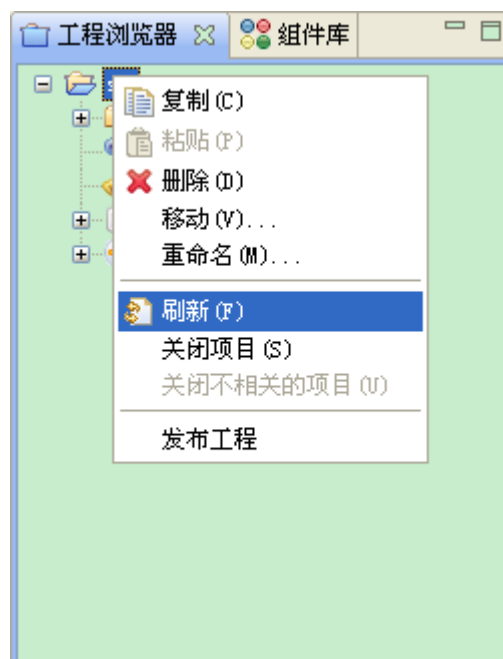
2、在“移动项目”对话框中指定新的存放位置，点击“确定”按钮，实现项目的移动操作。

10.4 重命名工程

1、在工程浏览器中右击任一物联网应用工程，在快捷菜单中选择“重命名”命令，工程名称变为可编辑状态。

2、输入新的工程名称，并按下回车键确认名称修改完成，实现重命名工程操作。

10.5 刷新工程



在工程浏览器中右击任一物联网应用工程，在快捷菜单中选择“刷新”命令，实现刷新工程操作。

10.6 关闭工程



1、在工程浏览器中右击任一物联网应用工程，在快捷菜单中选择“关闭项目”命令，关闭选中的物联网应用工程。

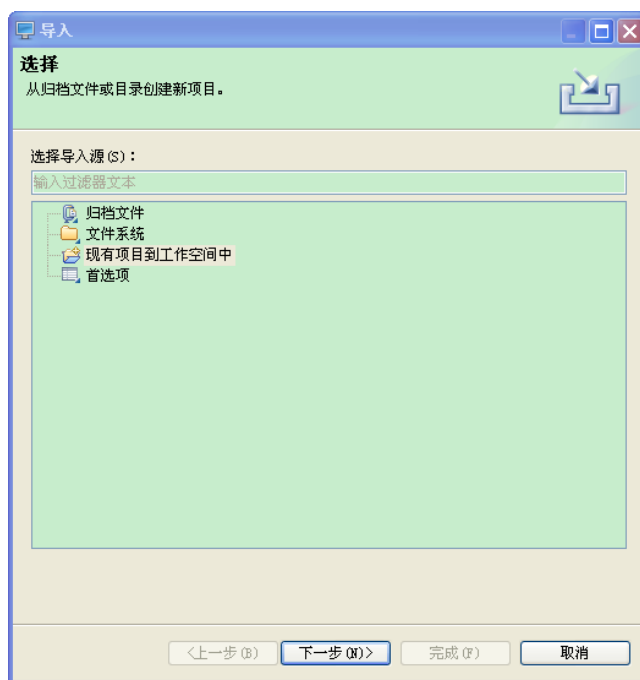
2、右击已处于关闭状态的物联网应用工程，在快捷菜单中选择“打开项目”命令，打开选中的物联网应用工程。

10.7 导入工程

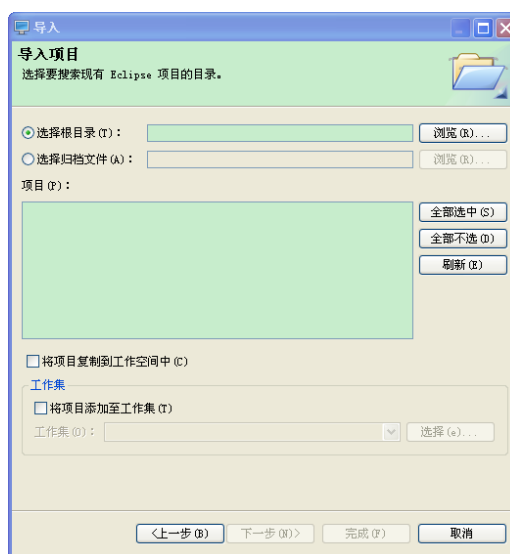
导入工程是将已经存在的工程导入到 HawkBuilder 中，导入工程具体步骤如下所示：

1、单击“工程”菜单中的“导入”菜单项，打开“导入”窗口。





2、选择“现有项目到工作空间中”，单击“下一步”按钮。



3、单击“浏览”按钮，选择现有工程所在的目录，选择需要导入的项目，单击“完成”按钮，实现工程的导入。

11 步骤四、快速接入 OPC 数据源

Hawk 物联网应用平台可以作为一个 OPC Client,连接任意厂家的 OPC Server。
下面以 Matrikon OPC 为例

(<http://www.matrikonopc.cn/products/opc-drivers/opc-simulation-serve>

r.aspx), 介绍 OPC 数据源接入 Hawk 物联网应用平台的方法步骤:



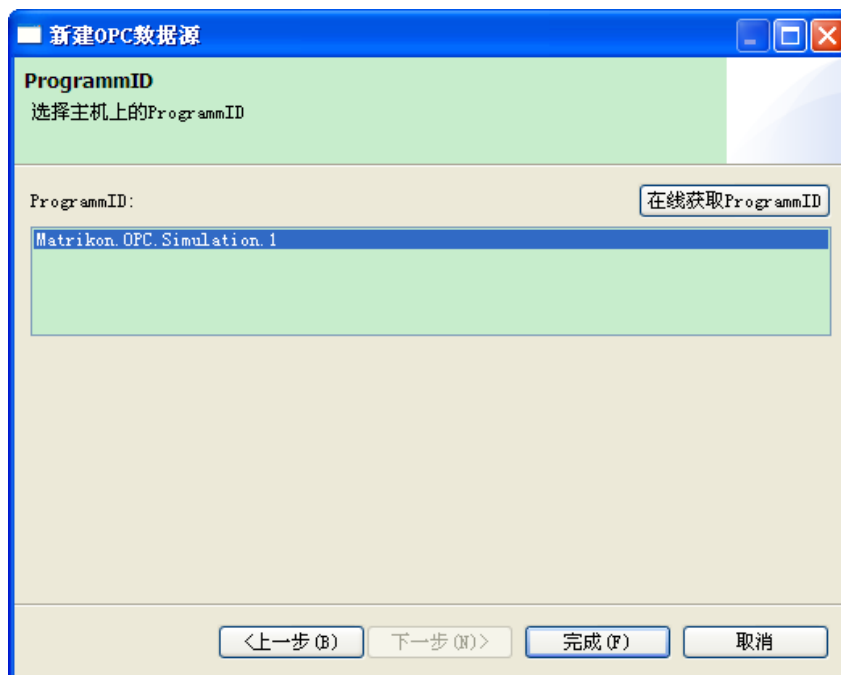
1、在打开的 Hawk 物联网应用工程中, 右击 “I/O 通信” 中的 “OPC” 项, 在快捷菜单中选择 “新建 OPC 数据源” 命令, 打开 “新建 OPC 数据源” 向导对话框。



2、输入数据源 ID、主机 IP、用户名、密码等信息, 单击 “下一步” 按钮, 进入选择主机上的 ProgrammID 对话框。

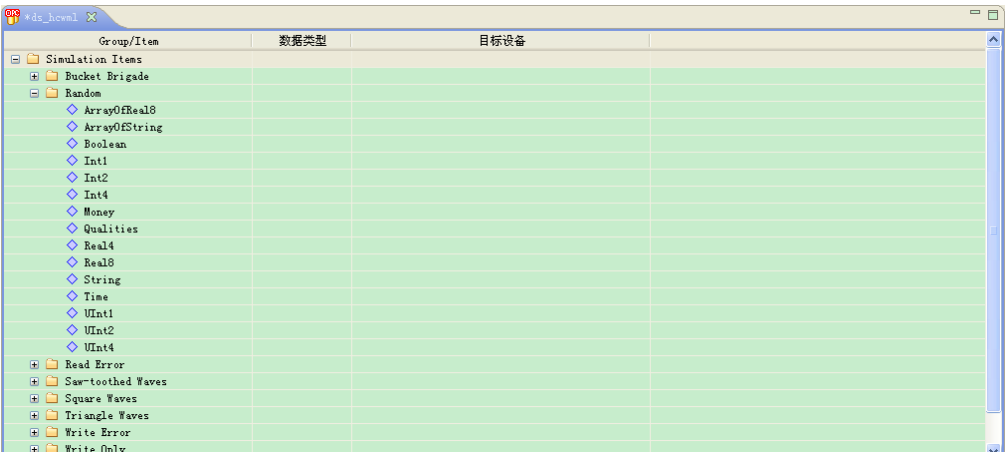




3、点击“在线获取 ProgrammID”按钮，在 DCOM 安全配置正确的情况下，会列出该主机上的所有 ProgrammID。



4、在 ProgrammID 列表中选择正确的 ProgrammID，单击“完成”按钮，创建该 OPC 数据源。

5、在工程树中双击刚刚创建的 OPC 数据源，打开数据源编辑器。



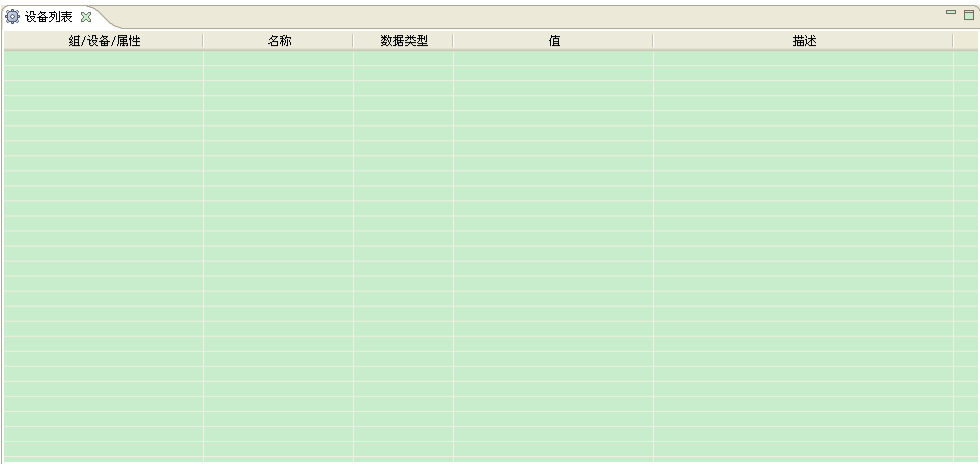
- 6、单击工具栏上的“”按钮，可以在线获取 OPC 数据源的点表信息。
- 7、在“目标设备”列中可以指定该数据关联的设备属性
- 8、单击工具栏上的“”按钮，保存数据源的原始点表信息。


12 步骤五、构建工程的设备列表

“设备列表”是对被监控物理设备的“建模”过程，是将设备类型库中的设备实例化的过程，每个设备都具有唯一的 ID 标识。

设备列表的创建过程如下所示：

- 1、双击工程树中的“设备列表”项，打开“设备列表”编辑器。



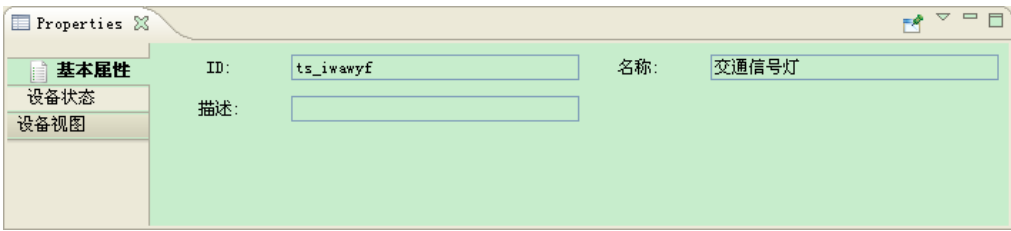
- 2、单击工具栏中的图标，可以建立任意多级“设备组”，方便有效管理

设备列表。

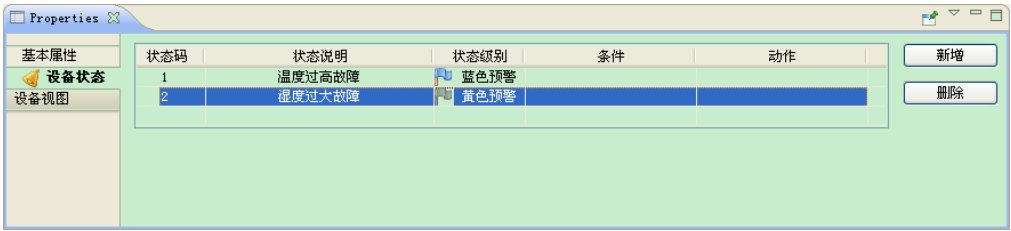
3、从“设备库”中拖动某个设备类型至“设备列表”编辑器中，实现该设备类型的实例化。



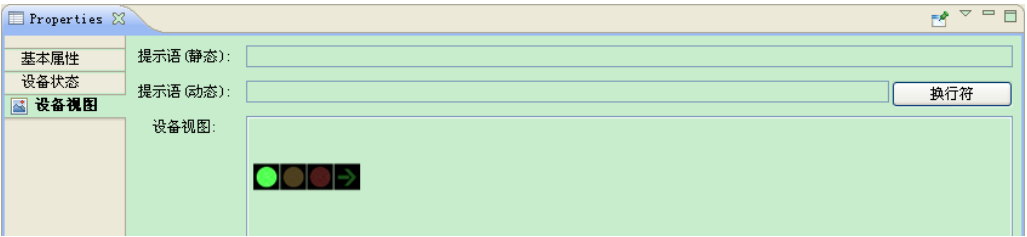
4、在“基本属性”中指定设备的 ID、名称和描述信息，按下回车键后完成修改。



5、在“设备状态”中指定设备的报警条件以及报警后执行的动作。每个设备的状态定义的初始值来自于设备库，除此之外，用户可以任意修改设备状态的定义。




6、在“设备视图”中指定设备的静态提示语、动态提示语和设备视图。

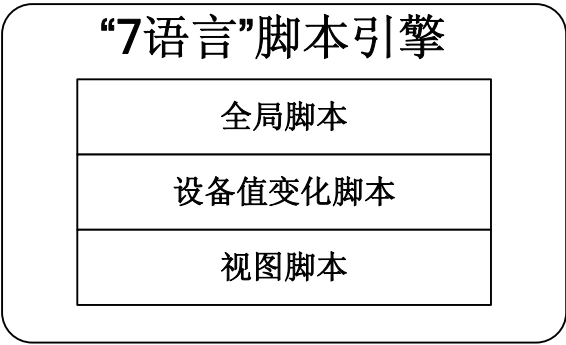


7、单击设备的某个属性，在“基本属性”中可以指定该设备的缺省值和“值变化脚本”



8、单击工具栏中的  按钮，进行保存。保存后在工程树中可以查看该工程中当前的设备列表。

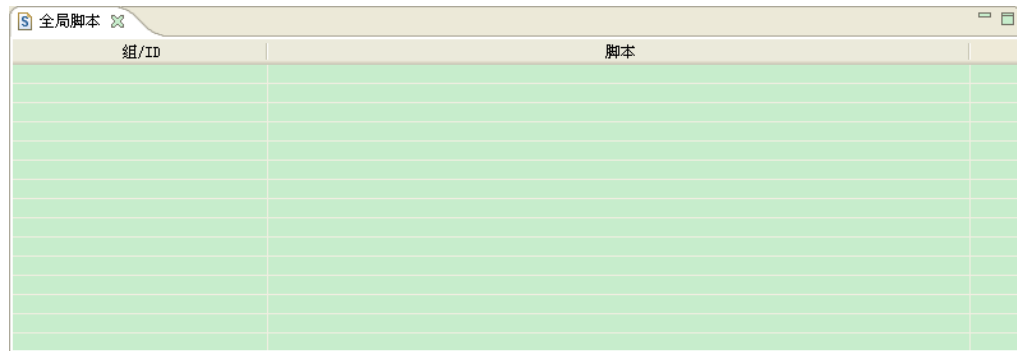
13 步骤六、定义工程的复杂业务逻辑





Hawk 物联网应用平台的业务逻辑通过“7 语言”脚本形式编写，包括全局脚本、设备值变化脚本和视图脚本三类。

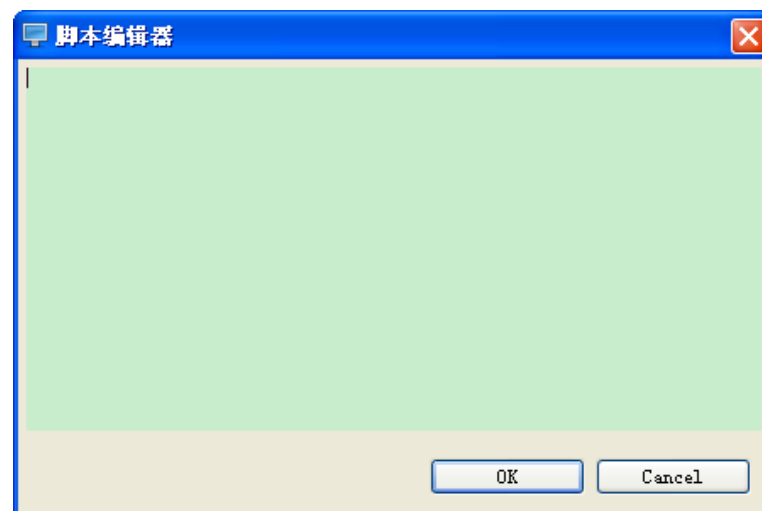
全局脚本的创建过程如下所示：


- 1、双击工程树中的“业务逻辑”项，打开“全局脚本”编辑器。



- 2、单击工具栏中的  图标，创建多级“脚本组”，有效管理所有的全局脚本片段。

- 3、单击工具栏中的  图标，创建一个全局脚本，指定该全局脚本的 ID，并打开“脚本编辑器”窗口，创建脚本内容。



- 4、单击工具栏中的  按钮，保存全局脚本列表。

13.1 常用脚本

- 1、字符串的比较：`c="hello world!"; if(c.equals("inAccess"))`

c=" another" ;

2、调试打印: c=" hello world!" ;print(c);

3、引用某个类: import bsh.Interpreter; c=" hello world!" ;

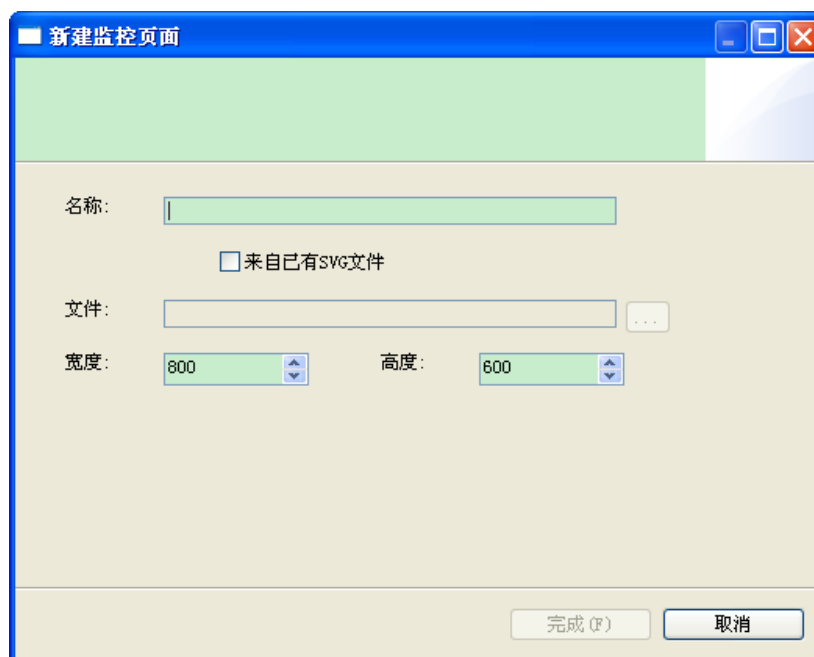
13.2 常用转义字符

<	<	小于号
>	>	大于号
&	&	和
'	'	单引号
"	"	双引号

每个转义字符必须以符号"&"开头，以符号";"结尾。

14 步骤七、创建监控画面

1、打开当前工程，右击“监控画面”，在快捷菜单中选择“新建页面”命令，打开“新建监控页面”窗口。



2、在“名称”文本框中输入画面的名称。

3、如果从已有 SVG 文件中创建监控页面，则勾选“来自已有 SVG 文件”，并选择相应的 SVG 文件。

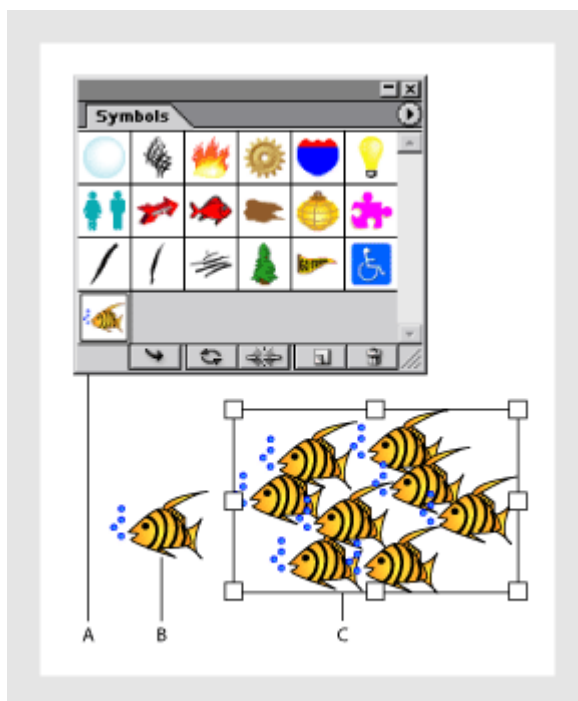
4、设定监控页面的宽度和高度（建议页面宽度为 1000，高度为 600），单击“完成”按钮，完成监控页面的创建。

15 步骤八、画面编辑

15.1 illustrator 绘制 SVG 的注意事项

15.1.1 不能存在符号(symbol)

符号工具（symbols tool）可以方便、快捷地生成很多相似的图形实例，比如一片树林、一群游鱼、水中的气泡等等。同时还可以通过符号工具（Symbolism tools）来灵活、快速地调整和修饰符号图形的大小、距离、色彩、样式等。这样，对于群体、簇类的物体就不必通过拷贝命令来一个一个的复制了，还可以有效地减小设计文件的大小。



使用 illustrator 绘制过程中可以采用符号工具，但完成后在通过 HawkBuilder 打开之前，必需将 SVG 图中所有的符号都“断开符号连接”。

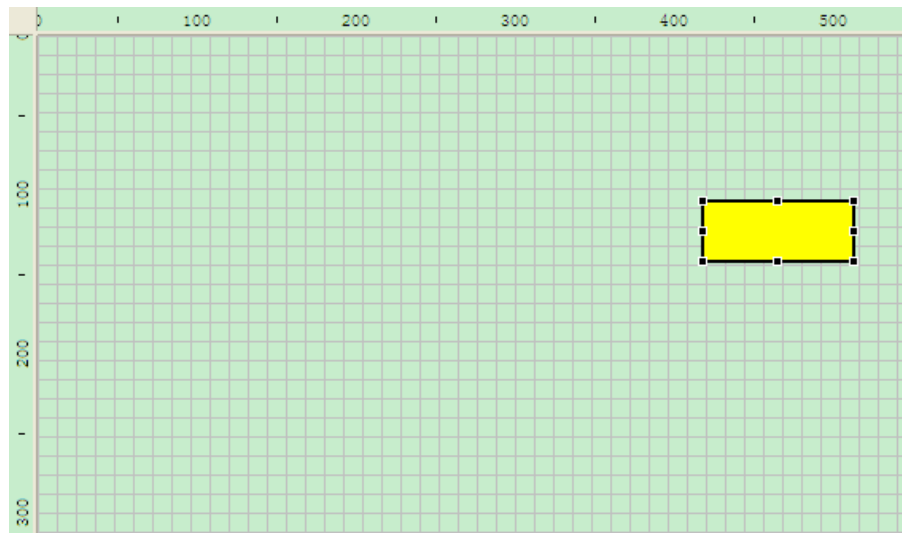
15.1.2 不能使用过渡色

作为一款专业的图形制作工具，illustrator 可以制作任意复杂的过渡色效果；相反，作为一款面向工程应用的图形制作工具，HawkBuilder 只支持相对简单的两种颜色之间的线性过渡色（实践证明基本上可以满足大部分的工程应用场合）。

推荐在使用 illustrator 绘制过程中，仅采用纯色填充，过渡色的效果放到 HawkBuilder 中实现。

15.2 画布与大纲视图




15.2.1 画布



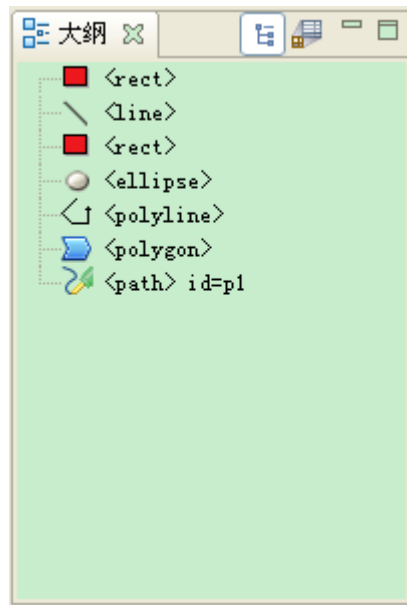
画布是图形绘制的主体区域，画布的大小没有限制，其左上角为坐标系原点，网格线方便精确定位放置到画布上的图形元素，画布的上方和左侧为标尺。

画布的常用操作包括放大、缩小、显示/隐藏网格线、显示/隐藏标尺和坐标吸附。



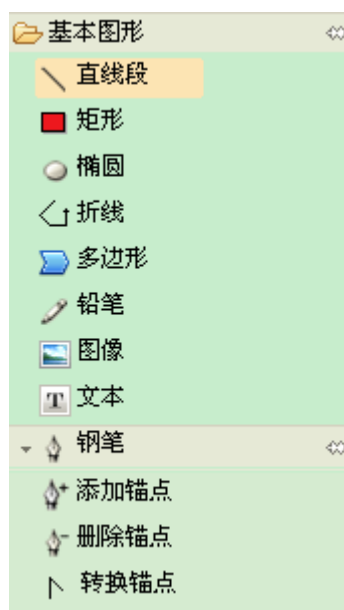
- 1、**放大**：放大显示整个画布
- 2、**缩小**：缩小显示整个画布
- 3、**显示/隐藏网格线**：单击  可以在显示或隐藏网格线之间进行切换
- 4、**显示/隐藏标尺**：单击  可以在显示或隐藏标尺之间进行切换
- 5、**坐标吸附**：单击  可以实现坐标吸附

15.2.2 大纲视图



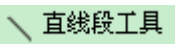
大纲视图可以通过树形结构和缩略图的方式查看画布中的图形元素，从而快速浏览和选择画布上的某个图形元素。

15.3 图形元素

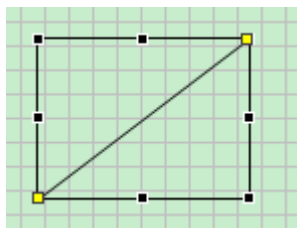



基本图形是二维图形的基础，包括直线工具、矩形工具、椭圆工具、折线工具、多边形工具、铅笔工具、图像工具、文本工具和钢笔工具组；丰富的图形组件可以快速构建专业的监控图形。

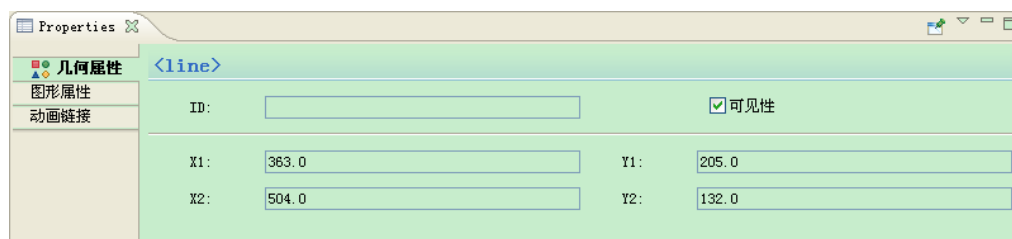
15.3.1 直线工具

1、在工具箱中单击如图  所示的“直线段工具”，选中“直线段工具”。

2、在画布上的直线起点处按下鼠标左键，并保持鼠标左键不松开，移动鼠标至直线终点处，松开鼠标左键，实现直线的绘制。



3、将鼠标移动至操作柄（黄色的方块）的上方，鼠标形状变成 ，按下鼠标左键并保持不松开，可以改变直线起点或终点的坐标。



4、在直线为选中状态下，通过属性视图的“几何属性”改变直线的 id、可见性、起点坐标（x1、y1）和终点坐标(x2、y2)。

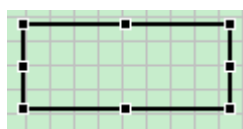


5、在直线为选中状态下，通过属性视图的“图形属性”可以改变直线的线条颜色和线条样式。

15.3.2 矩形工具

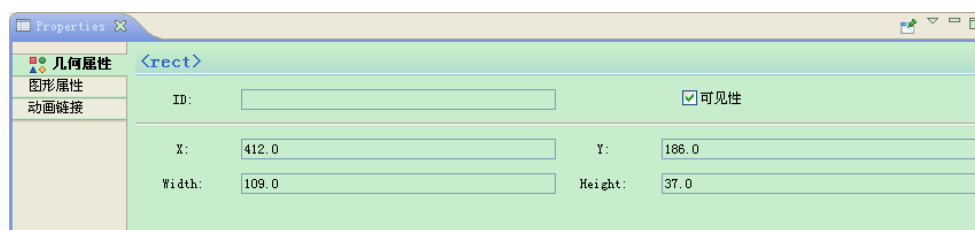
1、在工具箱中单击如图  矩形工具 所示的“矩形工具”，选中“矩形工具”。

2、在画布上的某点按下鼠标左键，并保持鼠标左键不松开，移动鼠标，当矩形大小适宜时松开鼠标左键，实现矩形的绘制。



3、用鼠标左键移动矩形周围的 8 个操作柄，可以任意改变矩形的位置和大小。


4、在矩形为选中状态下，通过属性视图中的“几何属性”改变矩形的 id、可见性、左上角坐标、长宽等属性。



5、在矩形为选中状态下，通过属性视图的“图形属性”可以改变矩形的线条颜色、填充颜色图案和线条样式。

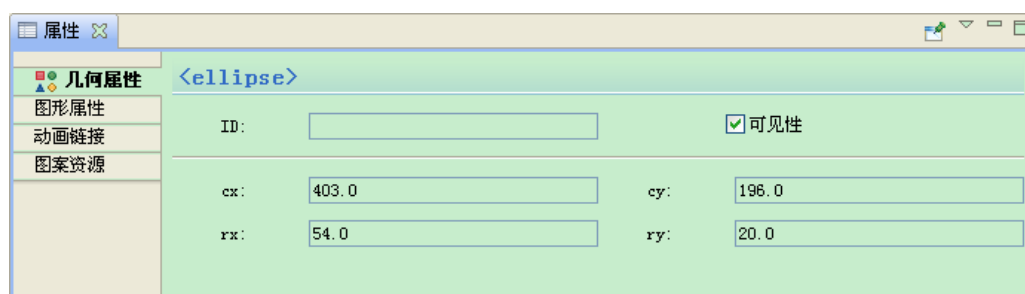


15.3.3 椭圆工具

- 1、在工具箱中单击如图  椭圆工具 所示的“椭圆工具”，选中“椭圆工具”。
- 2、在画布上的某点按下鼠标左键，并保持鼠标左键不松开，移动鼠标，当椭圆大小适宜时松开鼠标左键，实现椭圆的绘制。



- 3、用鼠标左键移动椭圆周围的 8 个操作柄，可以任意改变椭圆的位置和大小。
- 4、在椭圆为选中状态下，通过属性视图中的“几何属性”改变椭圆的 id、可见性、椭圆圆心坐标、长半径和短半径等属性。

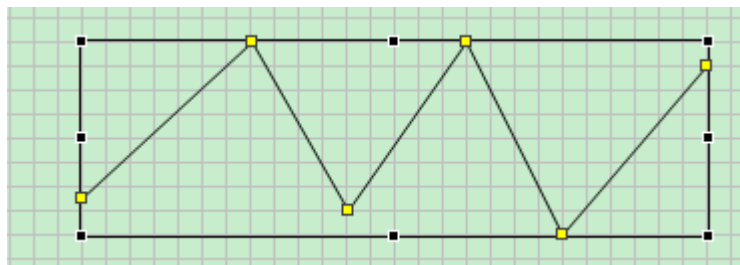



6、在椭圆为选中状态下，通过属性视图的“图形属性”可以改变椭圆的线条颜色、填充颜色图案和线条样式。

15.3.4 折线工具

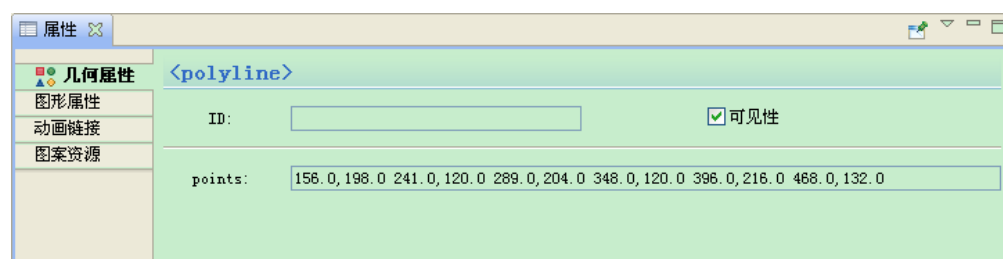
1、在工具箱中单击如图  所示的“折线工具”，选中“折线工具”。

2、在画布上的多个位置持续按下鼠标左键，并在最终位置处双击鼠标左键，实现折线的绘制。



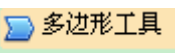
3、将鼠标移动至操作柄（黄色的方块）的上方，鼠标形状变成 ，按下鼠标左键并保持不松开，可以改变折线上该点的坐标。

4、在折线为选中状态下，通过属性视图中的“几何属性”改变折线的 id、可见性、折线坐标值等属性。

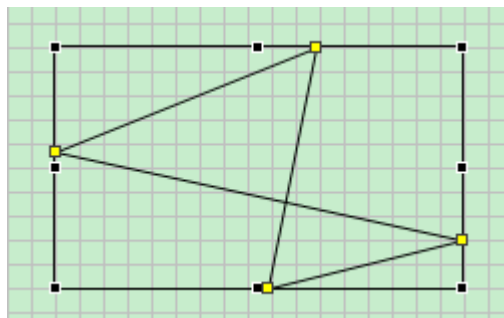



5、在折线为选中状态下，通过属性视图的“图形属性”可以改变折线的线条颜色、线条样式。

15.3.5 多边形工具

1、在工具箱中单击如图  所示的“多边形工具”，选中“多边形工具”。

2、在画布上的多个位置持续按下鼠标左键，并在最终位置处双击鼠标左键，实现多边形的绘制。



3、将鼠标移动至操作柄（黄色的方块）的上方，鼠标形状变成 ，按下鼠标左键并保持不松开，可以改变多边形上该点的坐标。

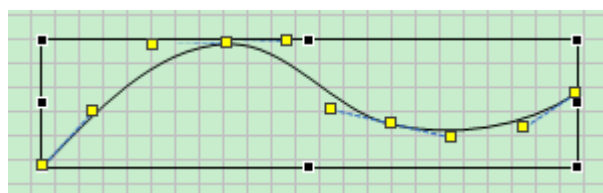
4、在多边形为选中状态下，通过属性视图中的“几何属性”改变椭圆的 id、可见性、多边形坐标值等属性。


5、在多边形为选中状态下，通过属性视图的“图形属性”可以改变多边形的线条颜色、填充颜色图案和线条样式。

15.3.6 铅笔工具

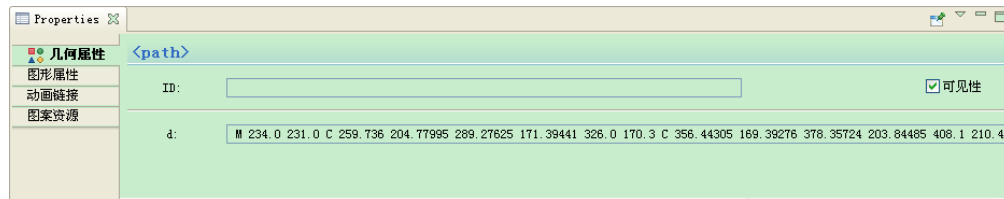
1、在工具箱中单击如图  铅笔工具 所示的“铅笔工具”，选中“铅笔工具”。

2、在画布上按下鼠标左键，保持鼠标左键不松开，鼠标移动的轨迹，形成一条自由曲线。



3、将鼠标移动至操作柄（黄色的方块）的上方，鼠标形状变成 ，按下鼠标左键并保持不松开，可以改变曲线的形状。

4、在曲线为选中状态下，通过属性视图中的“几何属性”改变曲线的 id、可见性、曲线形状等属性。

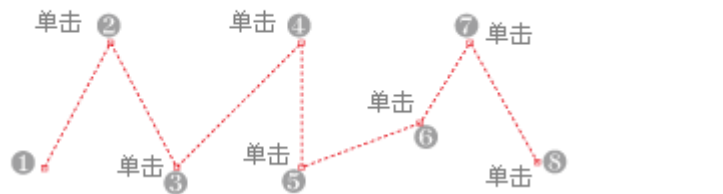


5、在曲线为选中状态下，通过属性视图的“图形属性”可以改变曲线的线条颜色、填充颜色图案和线条样式。

15.3.7 钢笔工具

1、在工具箱中单击如图  所示的“钢笔工具”，选中“钢笔工具”。

单击：使用钢笔工具绘制直线



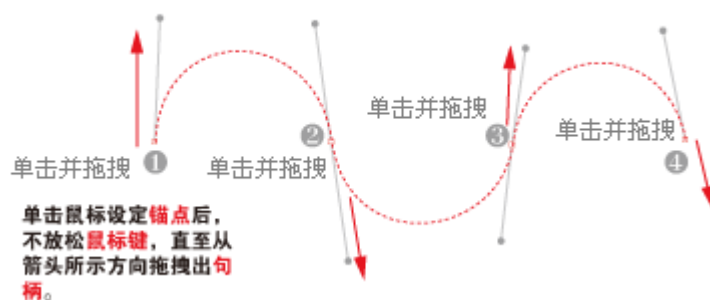
2、使用钢笔工具连续单击鼠标左键，可以绘制折线，按住 Ctrl 键在任意位置处点击鼠标左键结束绘制。

Shift+单击：使用钢笔工具绘制水平、垂直以45°角的直线



3、使用钢笔工具，按下 Shift 键连续单击鼠标左键，可以绘制水平、垂直和成 45 度角的折线，按住 Ctrl 键在任意位置处点击鼠标左键结束绘制。

用钢笔工具单击并拖拽，从锚点上拖出句柄。




4、使用钢笔工具，单击鼠标设定锚点后，不放松鼠标左键，直至从箭头所示方向拖拽出句柄，按住 Ctrl 键在任意位置处单击鼠标左键结束绘制。

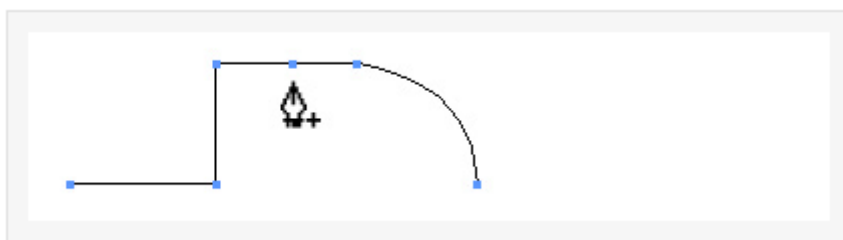
5、在曲线为选中状态下，通过属性视图中的“几何属性”改变曲线的 id、可见性、曲线形状等属性。

6、在曲线为选中状态下，通过属性视图的“图形属性”可以改变曲线的线条颜色、填充颜色图案和线条样式。


15.3.8 添加锚点工具

1、在工具箱中单击如图  添加锚点工具 所示的“添加锚点工具”，选中“添加锚点”工具。

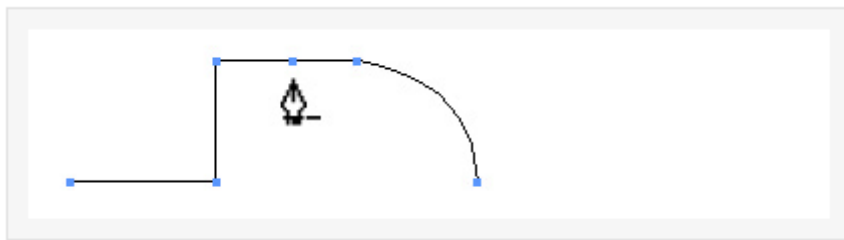
2、在某条路径上单击鼠标左键，可以添加锚点。



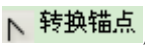
15.3.9 删除锚点工具

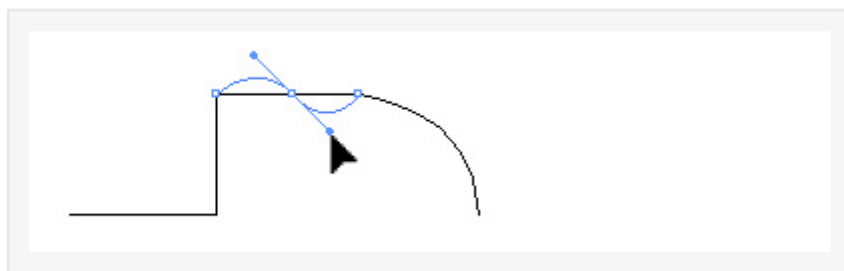
1、在工具箱中单击如图  删除锚点 所示的“删除锚点工具”，选中“删除锚点”工具。

2、在某条路径上单击可以删除已有锚点。

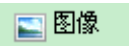


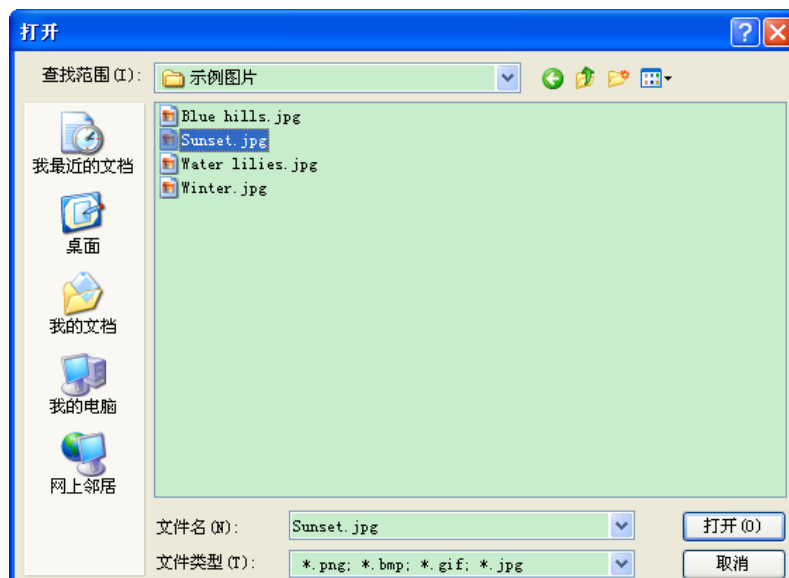
15.3.10 转换锚点工具

- 1、在工具箱中单击如图  所示的“转换锚点工具”，选中“转换锚点”工具。
- 2、创建一个锚点并且拖拽锚点的布赛尔手柄，可以把直角点变平滑点，同样在一个曲线点上点击转换锚点工具可以变成直角点。



15.3.11 图像工具

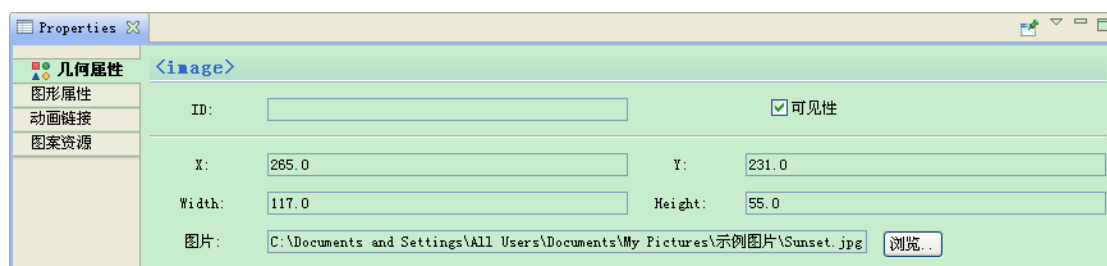
- 1、在工具箱中单击如图  所示的“图像工具”，选中“图像”工具。
- 2、在画布上单击鼠标左键，并按住左键拖动鼠标，达到适宜区域大小后松开鼠标，弹出图片文件选择窗口



3、选中图片后，单击“打开”按钮，完成图片的插入。

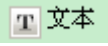


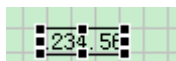
4、在图片为选中状态下，通过属性视图中的“几何属性”改变图片的 id、可见性、大小、位置、和图片 URL 等属性。



注：在实际应用中，过多的使用图片会让矢量化图形的优点大打折扣，所以该工具不推荐经常使用。

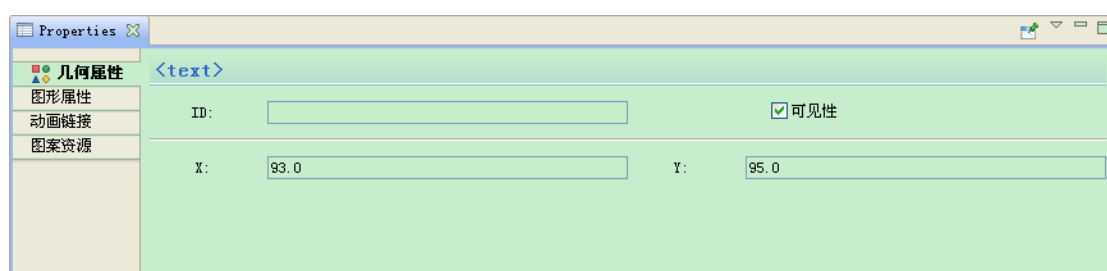
15.3.12 文本工具

- 1、在工具箱中单击如图  所示的“文本工具”，选中“文本”工具。
- 2、在画布上单击鼠标左键，实现文本的绘制。

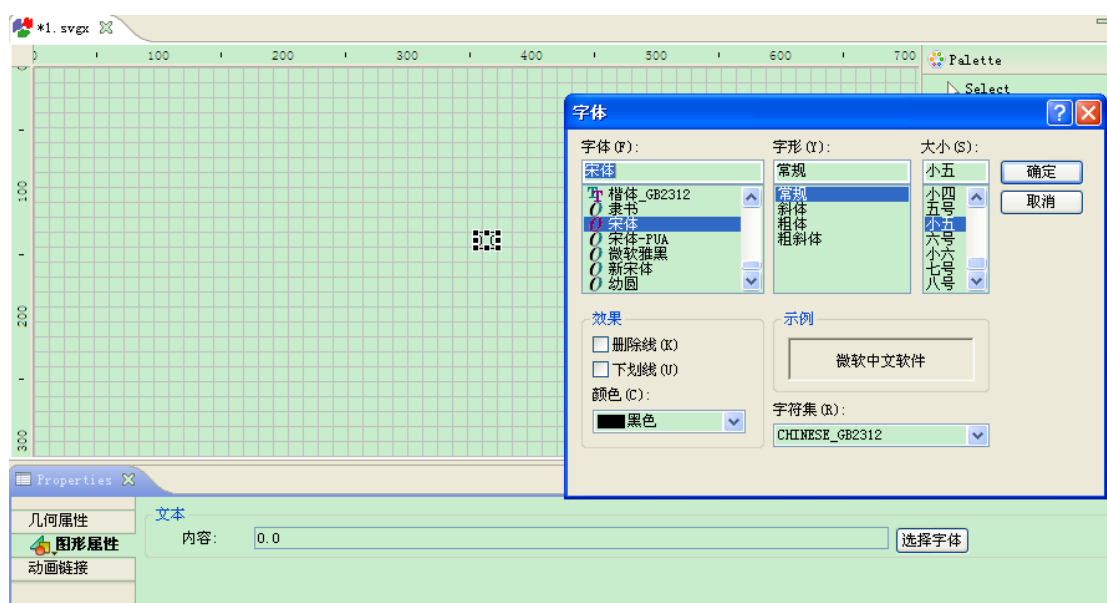


3、在 Text 区域双击，实现文本初始值的输入。

4、在文本为选中状态下，通过属性视图中的“几何属性”改变图片的 id、可见性、位置属性。



5、在“图形属性”中可以指定字体的内容和格式



15.4 图形编辑

图形编辑功能包括复制、粘贴、剪切、删除、重做、撤消、对齐方式、长宽匹配、排列分布、图层次序、旋转翻转等。

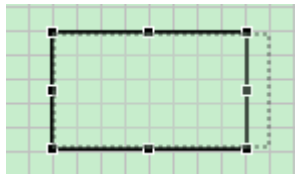
15.4.1 移动

除了支持通过鼠标直接对画布上的图形元素进行移动操作外，HawkBuilder 还支持通过键盘上的方向箭头移动图形元素。



1、在选中某个图形元素的前提下，按下键盘上的如图所示键，切换至“箭头移动”状态，连续按下多次，可在多种方式下切换。

2、通过方向箭头进行图形元素大小和位置的调整，如图所示。



3、按下回车键结束调整操作。

15.4.2 复制、粘贴、剪切、删除

支持图形元素的复制、剪切、粘贴和删除操作。

15.4.3 撤消/重做

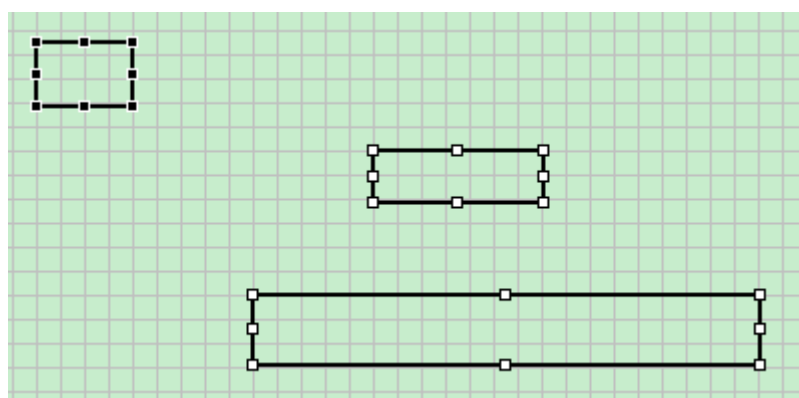
支持图形编辑操作的撤消 undo 和重做 redo。

15.4.4 对齐方式

图形编辑器支持左对齐、右对齐、左右居中对齐、上对齐、下对齐、上下居中对齐共六种方式。



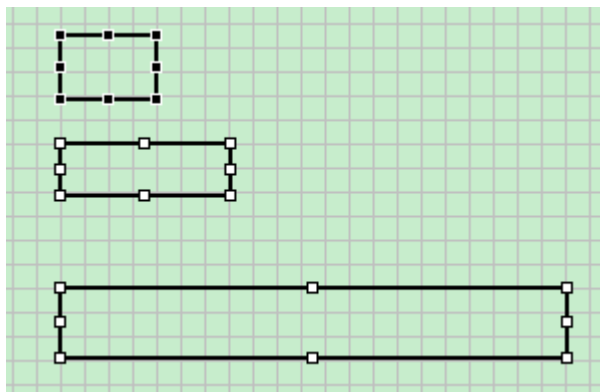
假设有三个矩形初始位置如图所示：



其中操作柄为黑色方块的为“主选择图形”，以其作为对齐的参考。

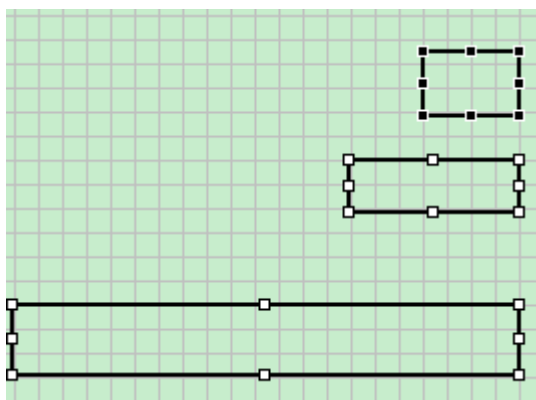
15.4.4.1 左对齐

点击“左对齐”按钮后，三个矩形以“主选择图形”的左边界对齐排列。




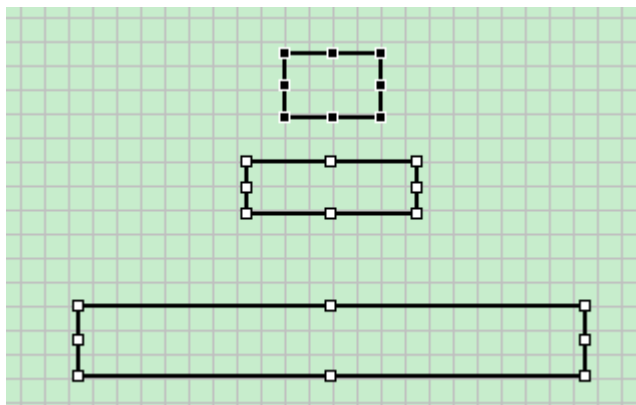
15.4.4.2 右对齐

点击“右对齐”按钮后，三个矩形以“主选择图形”的右边界对齐。



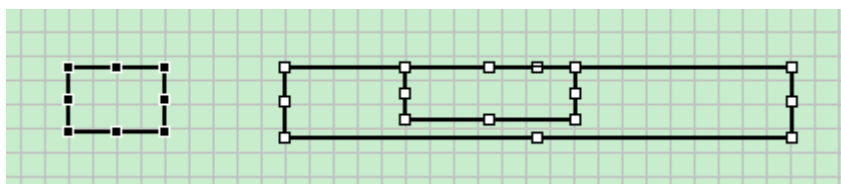
15.4.4.3 左右居中对齐

点击“左右居中对齐”按钮后，三个矩形以“主选择图形”的左右中心线对齐。



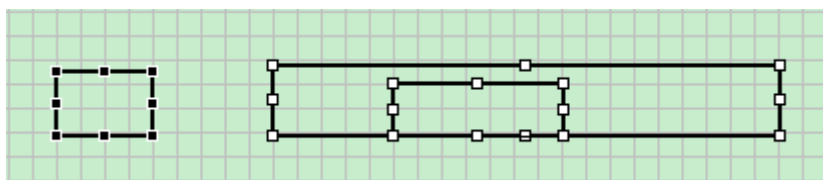
15.4.4.4 上对齐

点击“上对齐”按钮后，三个矩形以“主选择图形”的上边界对齐。




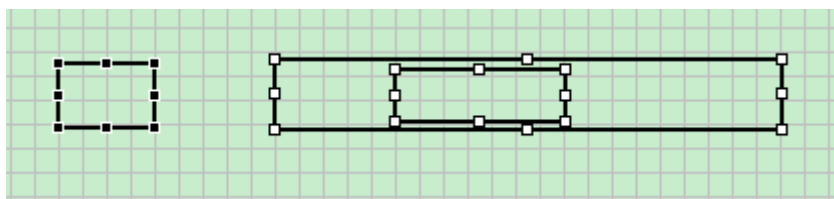
15.4.4.5 下对齐

点击“下对齐”按钮后，三个矩形以“主选择图形”的下边界对齐。



15.4.4.6 上下居中对齐

点击“上下居中”按钮后，三个矩形以“主选择图形”的上下中心线对齐。

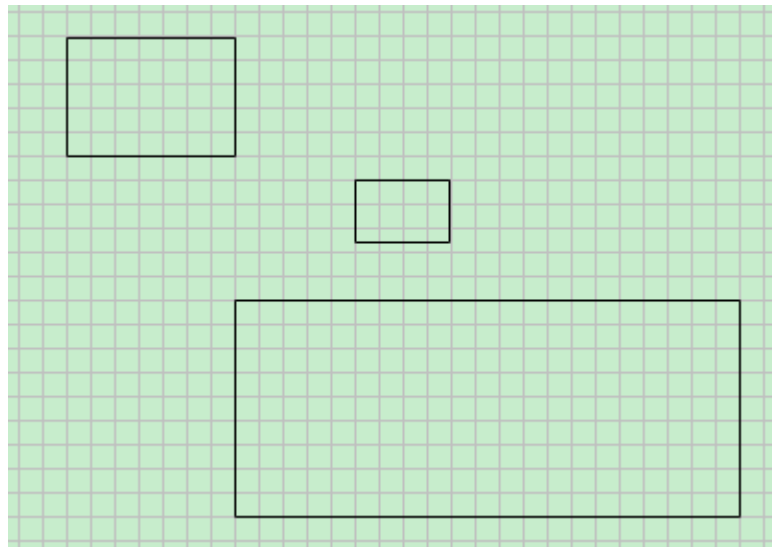


15.4.5 长宽匹配


图形编辑器支持长度匹配、宽度匹配和长宽同时匹配三个方式。

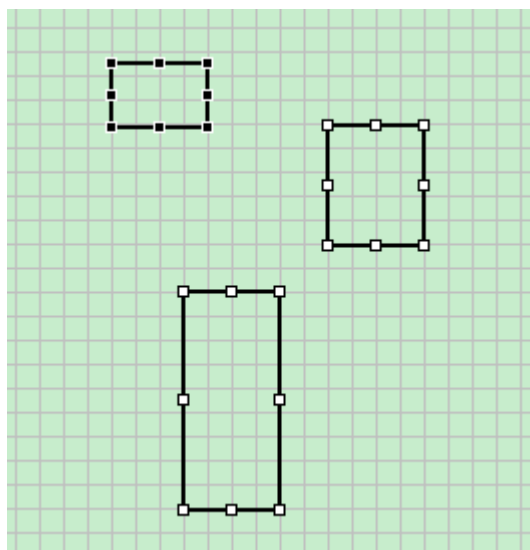


假设有三个矩形，他们的初始大小如图所示：




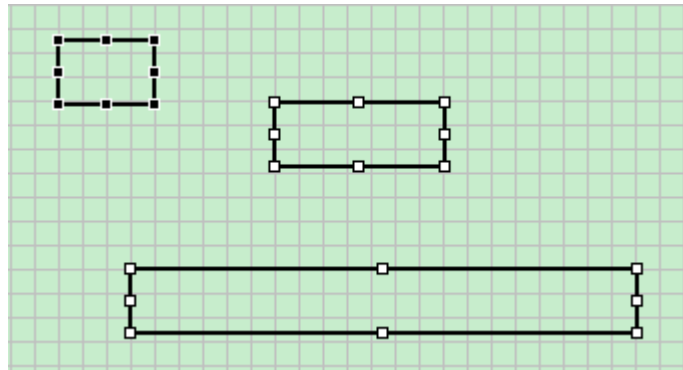
15.4.5.1 长度匹配

点击“长度匹配”按钮后，三个图形的长度都与“主选择图形”的长度保持一致。




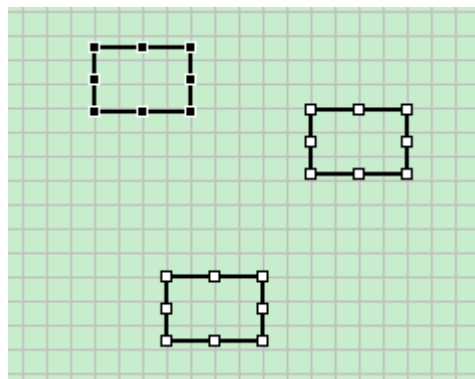
15.4.5.2 宽度匹配

点击“宽度匹配”按钮后，三个图形的宽度都与“主选择图形”的宽度保持一致。



15.4.5.3 长宽匹配

点击“长宽匹配”按钮后，三个图形的长宽都与“主选择图形”的长度和宽度保存一致。

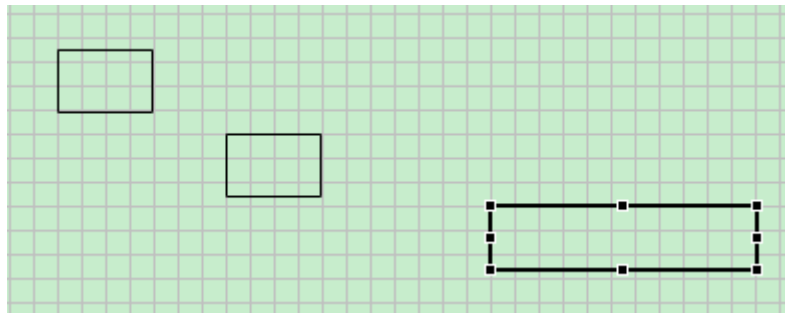


15.4.6 排列分布

图形编辑器支持使对象之间的水平间隙相等、水平方向中心等距分布、水平方向紧密排列分布、使对象之间的垂直间隙相等、垂直方向中心等距分布和垂直方向紧密排列分布共六种排列分布。

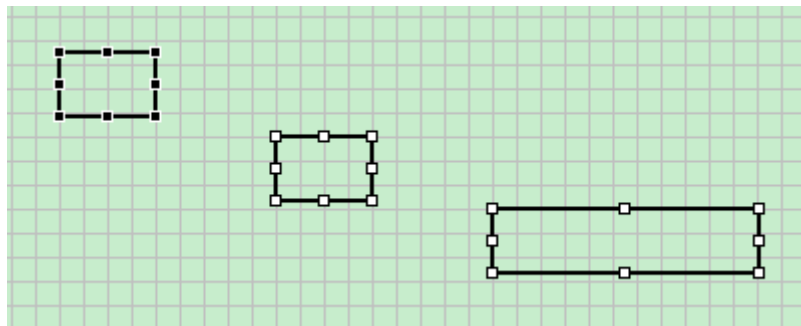


假设有三个矩形，其起始位置如图所示：




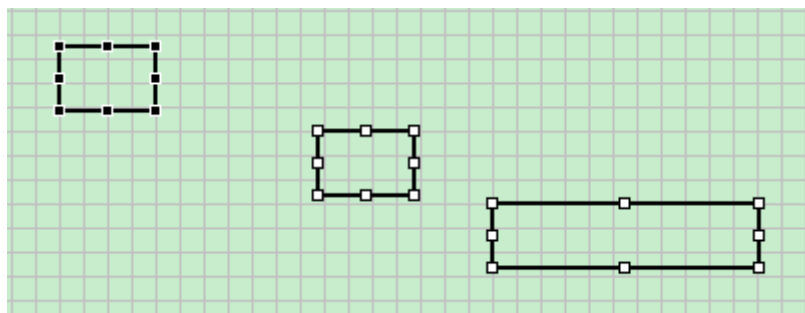
15.4.6.1 水平间隙相等分布

点击“水平间隙相等”按钮后，三个矩形的水平间隙保持相等。



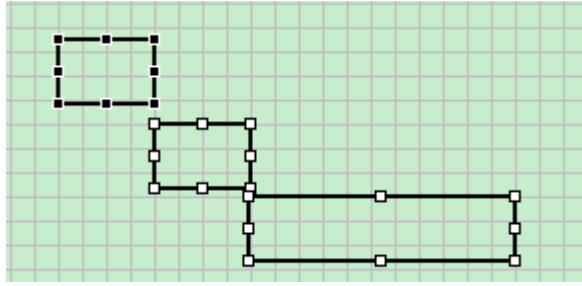
15.4.6.2 水平中心等距分布

点击“水平中心等距分布”按钮后，三个矩形的中心点之间的距离依次相同。



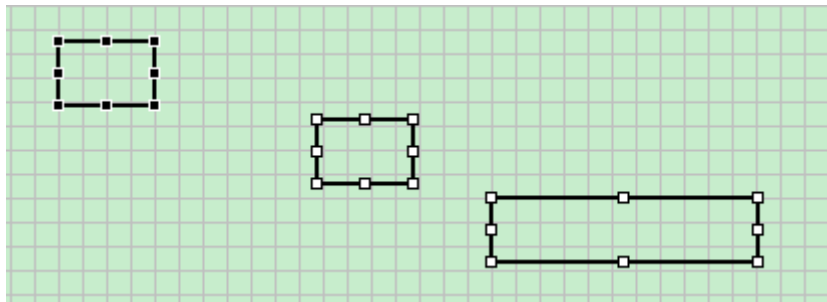
15.4.6.3 水平紧密排列分布

点击“水平紧密排列分布”按钮后，三个矩形依次紧密排列。




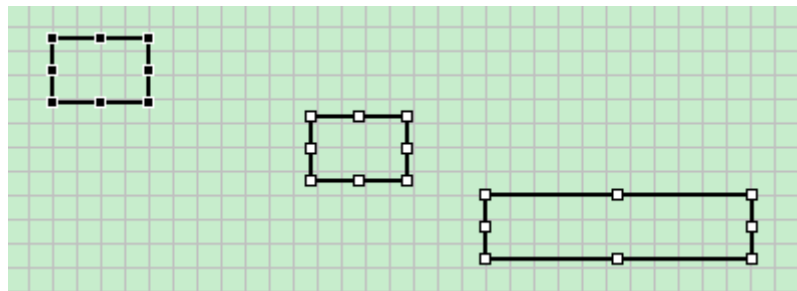
15.4.6.4 垂直间隙相等分布

点击“垂直间隙相等分布”按钮  后，三个矩形的垂直间隙保持相等。




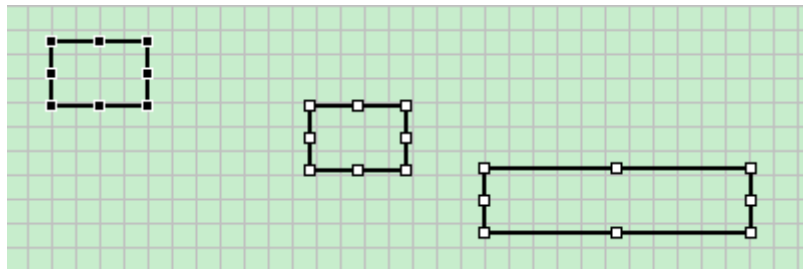
15.4.6.5 垂直中心等距分布

点击“垂直中心等距分布”按钮  后，三个矩形的中心点在垂直方向上依次间距相等。



15.4.6.6 垂直紧密排列分布

点击“垂直紧密排列分布”按钮  后，三个矩形的在垂直方向上依次紧密排列。

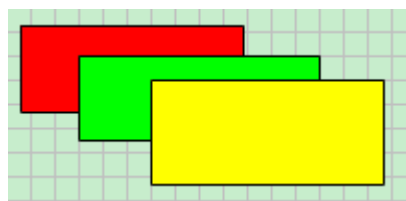


15.4.7 图层顺序


图形编辑器支持上移一层、下移一层、移至顶层和移至底层四种操作。

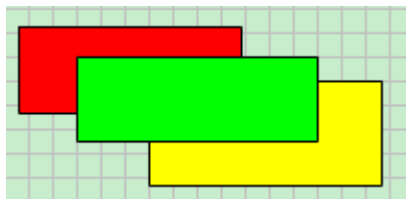


假设有三个矩形，其初始位置如图所示。




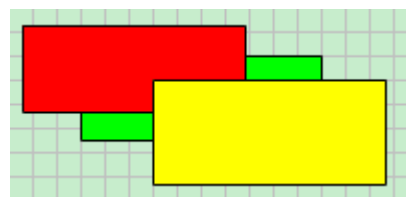
15.4.7.1 上移一层

选中绿色矩形，单击“上移一层”按钮后，三个矩形变化为




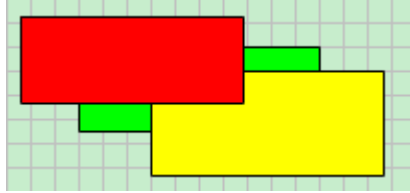
15.4.7.2 下移一层

选中绿色矩形，单击“下移一层”按钮后，三个矩形变化为




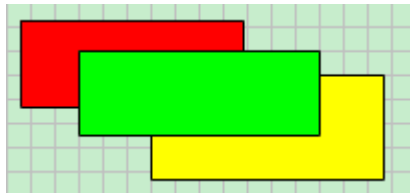
15.4.7.3 移至顶层

选中红色矩形，单击“移至顶层”按钮后，三个矩形变化为



15.4.7.4 移至底层

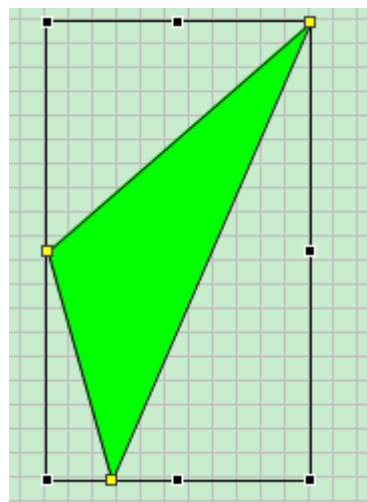
选中黄色矩形，单击“移至底层”按钮后，三个矩形变化为




15.4.8 旋转翻转

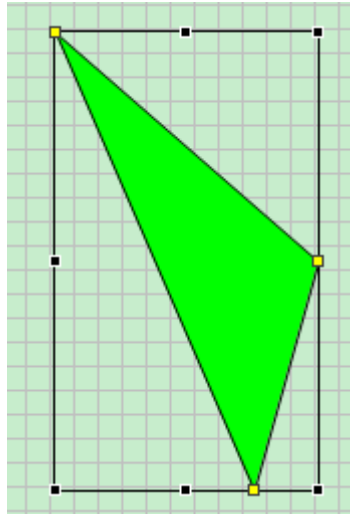
图形编辑器支持水平翻转、垂直翻转、顺时针旋转 90 度和逆时针旋转 90 度四种固定角度的旋转方式。

假设有一多边形，其原始状态如图所示




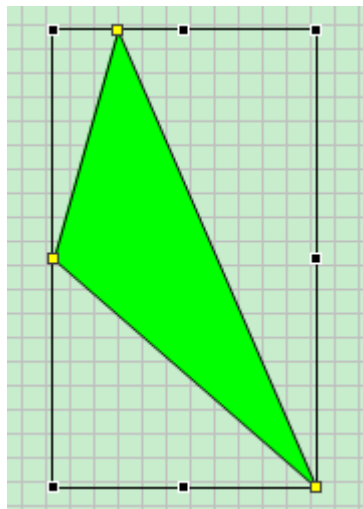
15.4.8.1 水平翻转

点击“水平翻转”按钮后，多边形变化为如图所示，其实际为水平方向的轴对称图形。



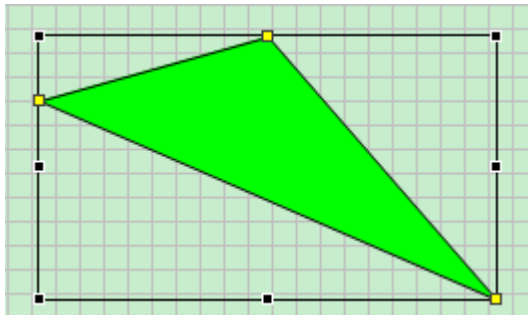
15.4.8.2 垂直翻转

点击“垂直翻转”按钮后，多边形变化为如图所示，其实际为垂直方向上的轴对称图形。



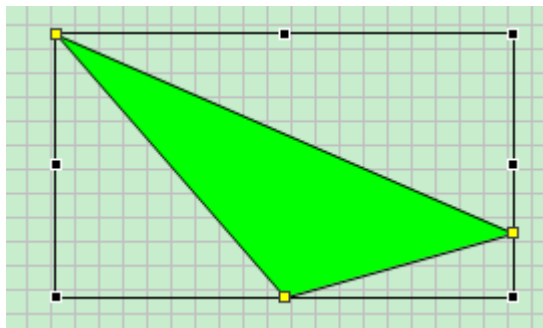
15.4.8.3 顺时针旋转 90 度

点击“顺时针旋转 90 度”按钮后，多边形变化为如图所示。






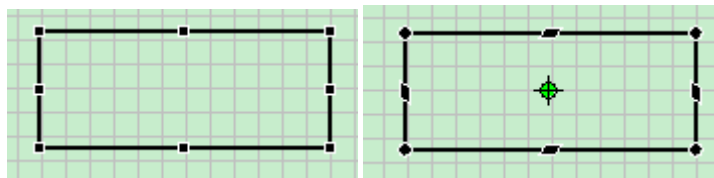
15.4.8.4 逆时针旋转 90 度




点击“逆时针旋转 90 度”按钮后，多边形变化为如图所示。

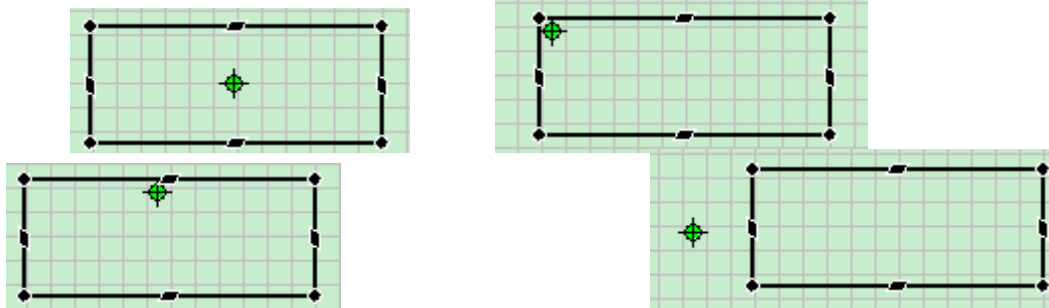


15.4.9 自由旋转和错切

图形编辑器通过“旋转”工具和“倾斜”工具实现自由角度的旋转和错切操作。以矩形为例，如图所示，前者的黑色方块操作柄主要用来改变矩形的大小。后者的黑色圆形操作柄用来自由旋转矩形，黑色的平行四边形操作柄用来错切矩形。



圆心点的初始位置在矩形的中心点，无论在何处，在其上双击鼠标左键，圆心点都会回归至起始位置。



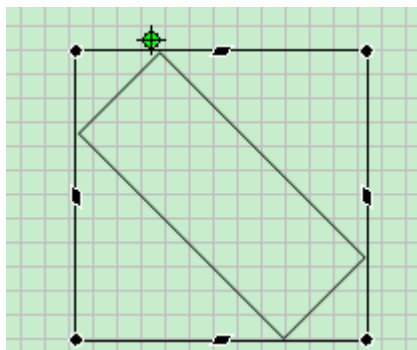
15.4.9.1 自由旋转

以矩形的左上角为中心点，自由旋转的操作方法如下所示：

- 1、将  圆心点拖动至矩形的左上角，如图所示。

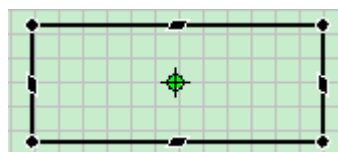


- 2、拖动旋转句柄（黑色圆形）至任意角度，矩形形状变化为如图所示。

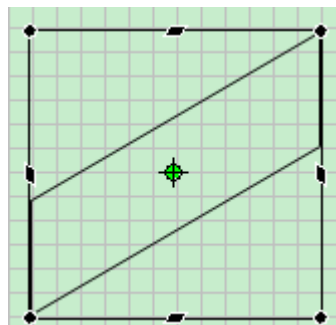
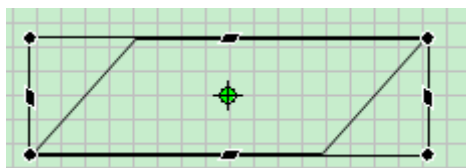


15.4.9.2 错切

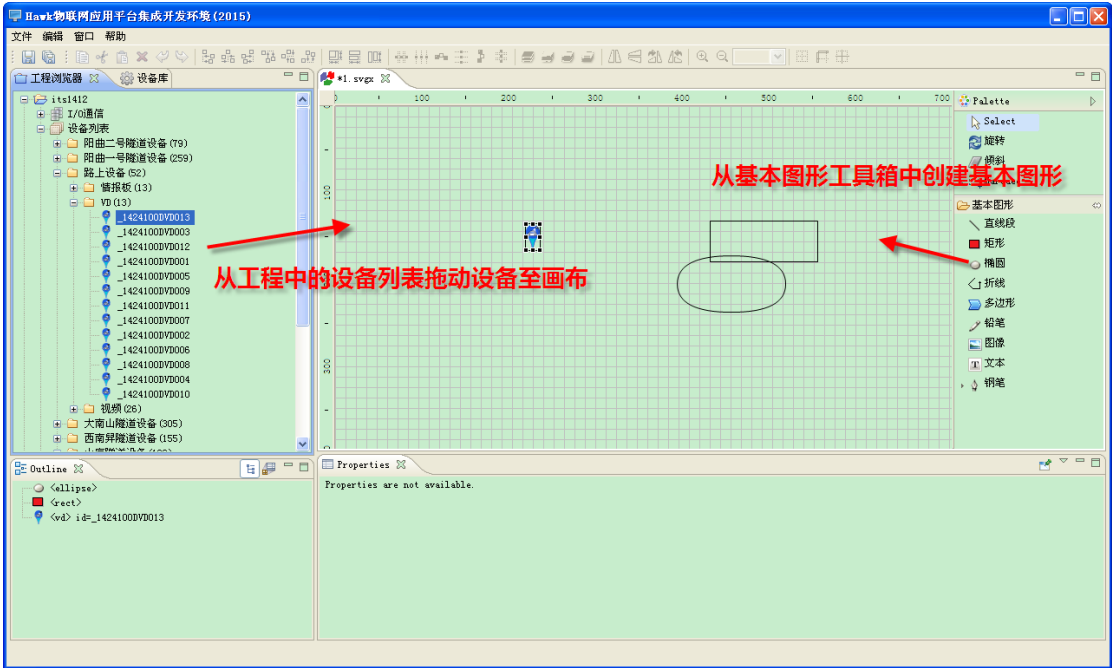
- 1、假设矩形的初始状态如图所示。



- 2、拖动错切句柄（黑色平行四边形），矩形形状变化为如图所示。



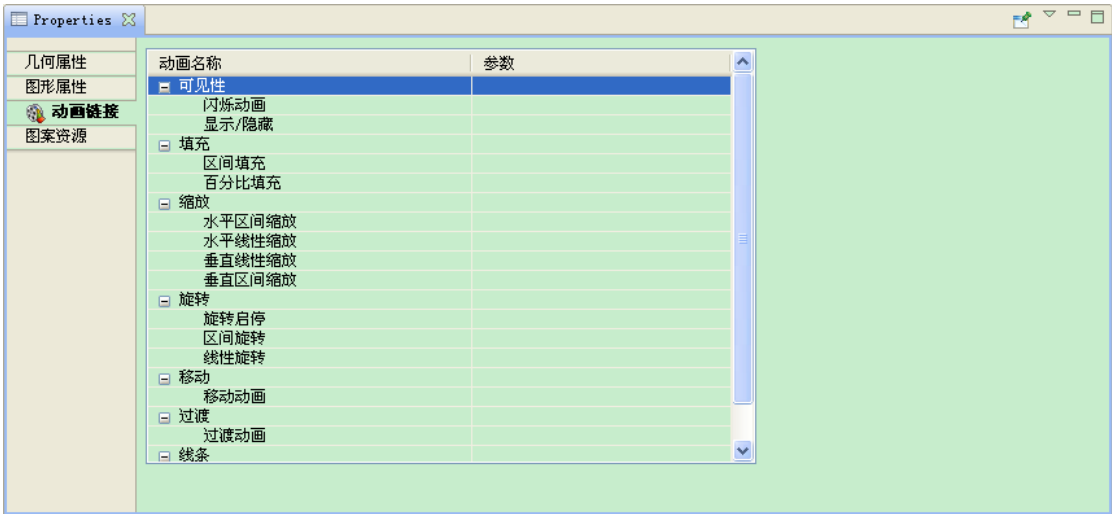
15.5 视图构建(静态部分)



视图是工程设备列表的“表现”，每个视图中的图形元素既可以来自于设备列表（如果该设备指定了设备外观的话），也可以来自于基本图形。

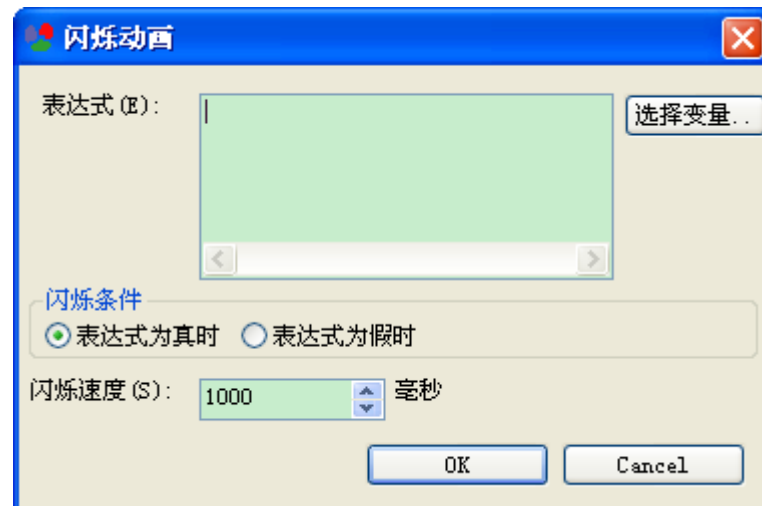
16 步骤九、创建监控页面的动画效果

在画布中选中某个图形元素，打开属性视图中的“动画链接”窗口，为该图形配置相关动画链接。



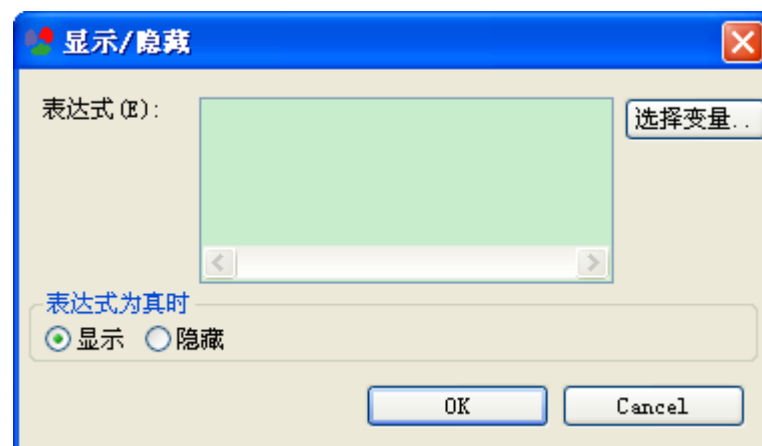
16.1 可见性

16.1.1 闪烁动画



配置闪烁动画的触发条件、闪烁速度等参数。

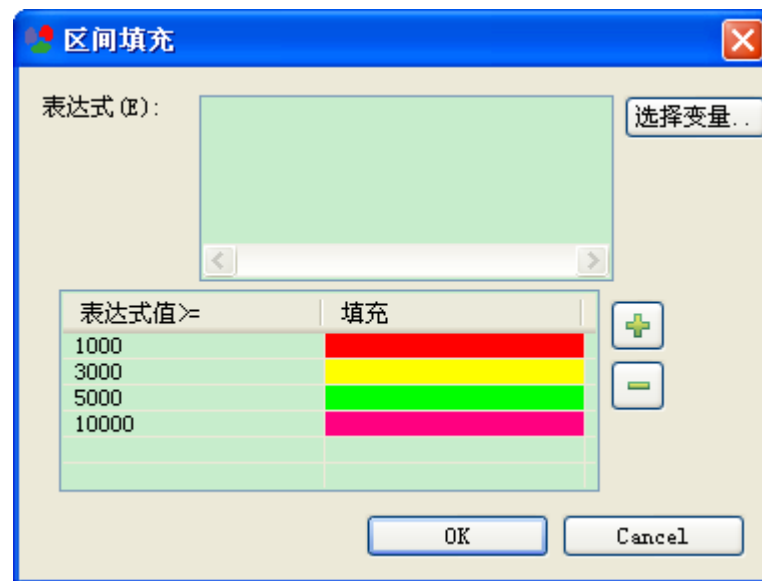
16.1.2 显示/隐藏动画



配置显示/隐藏动画的触发条件等参数。

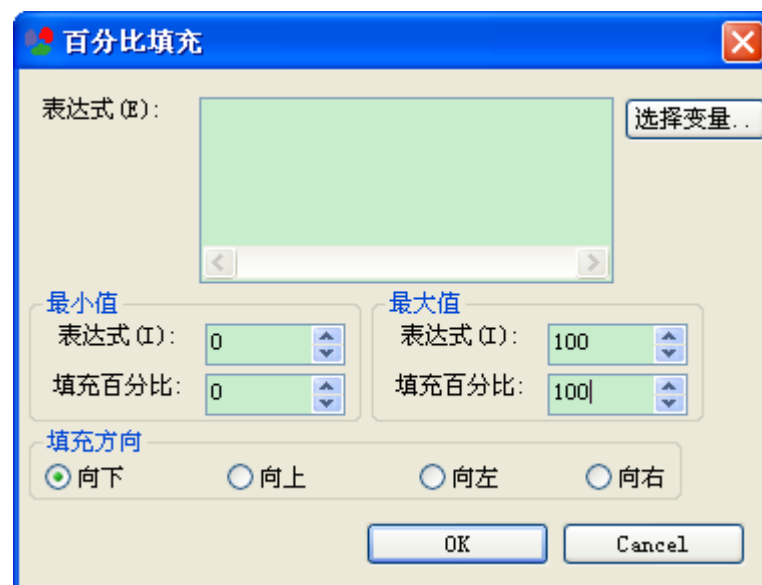
16.2 填充

16.2.1 区间填充动画



区间填充动画根据表达式的运行值所在区间的不同，显示不同的颜色值。

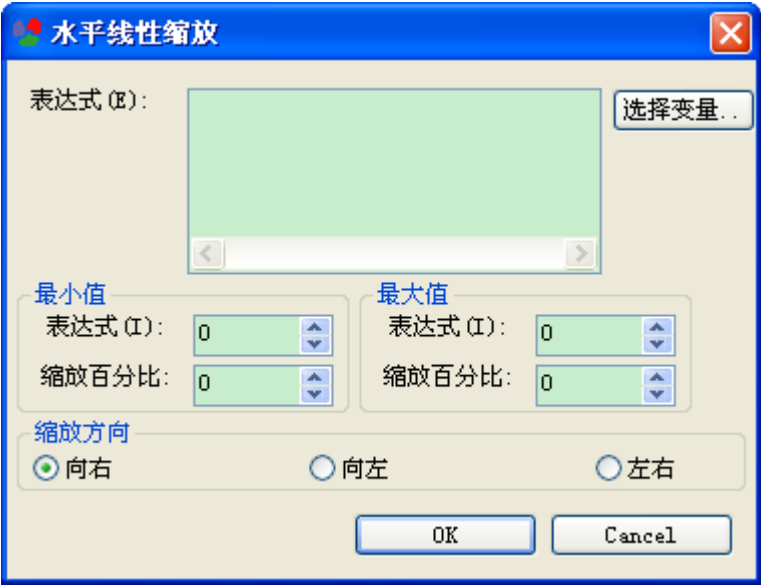
16.2.2 百分比填充动画



百分比填充动画根据表达式的运行值为最大值的百分比，实现填充不同的高度。该动画可以实现类似于温度计液位、水箱水位等效果。

16.3 缩放

16.3.1 水平线性缩放动画



设定水平线性缩放动画的缩放百分比和缩放方向等参数。

16.3.2 水平区间缩放动画



设定水平区间缩放动画的缩放百分比和缩放方向等参数。

16.3.3 垂直线性缩放动画



设定垂直线性缩放动画的缩放百分比和缩放方向等参数。

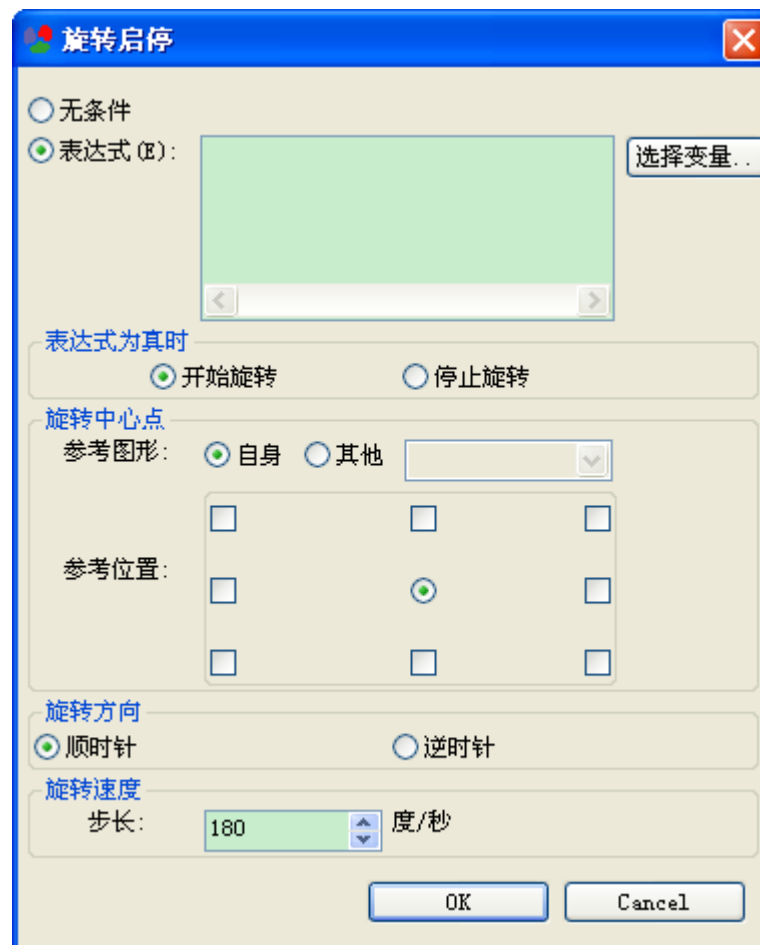
16.3.4 垂直区间缩放动画



设定垂直区间缩放动画的缩放百分比和缩放方向等参数。

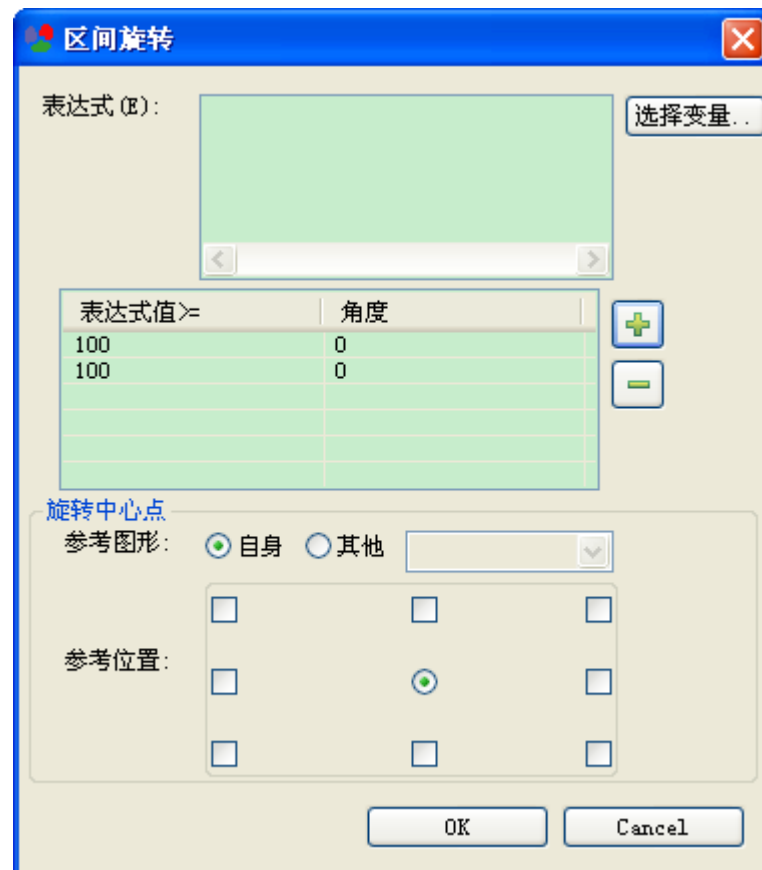
16.4 旋转

16.4.1 旋转启停动画



设定旋转启动动画的触发条件、旋转中心点、旋转方向和旋转速度等参数。

16.4.2 区间旋转动画



设定区间旋转动画的旋转角度、旋转中心点等参数。

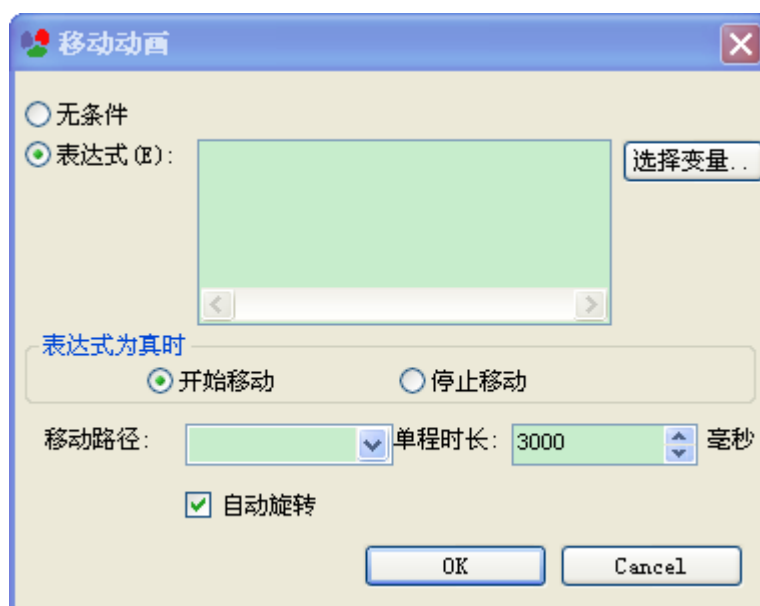
16.4.3 线性旋转动画



设定线性旋转动画的旋转角度和旋转中心点等参数。

16.5 移动

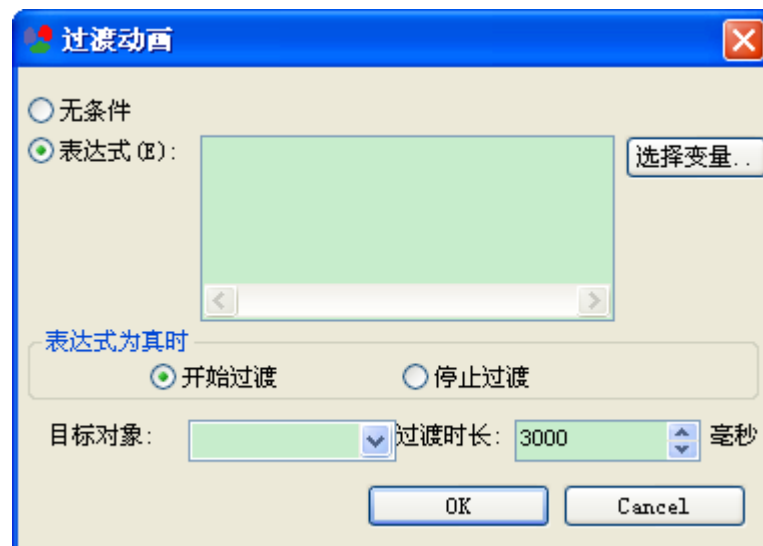
16.5.1 移动动画



设定移动动画的触发条件、移动路径、移动速度和是否自动旋转等参数。

16.6 过渡

16.6.1 过渡动画



过渡动画可以使得一个图形元素在设定时长内变化为同类型的另一个图形元素。可以设定触发条件、目标对象和所需时长等参数。

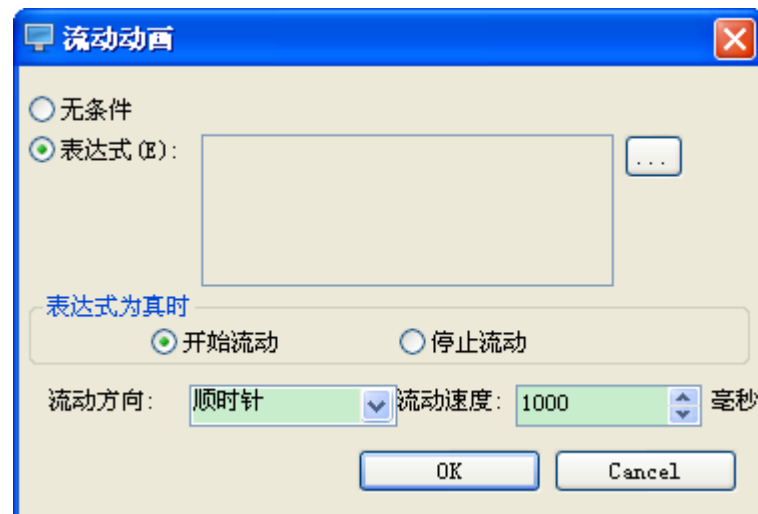
16.7 线条

16.7.1 线条动画



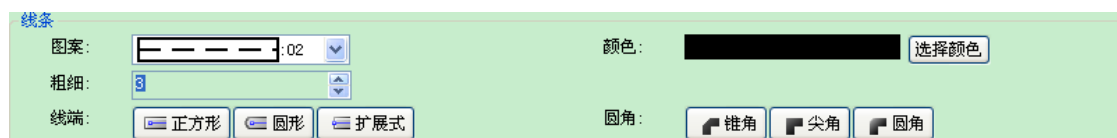
线条动画可以设定当表达式的运行值在不同区间内，显示不同颜色的线条。

16.7.2 流动动画



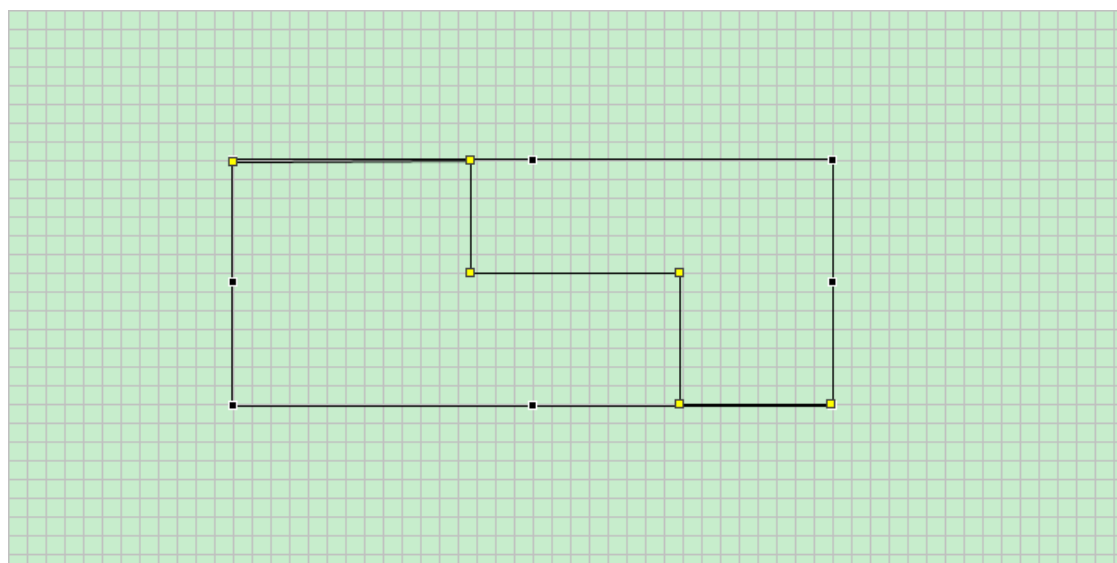
流动动画实现液体流动的效果，该流动效果适用于所有具有“线条”属性的对象，比如直线、矩形、折线、曲线等。

流动效果必需将图形的线条设置成虚线的形式，并可以设置粗细、线端、颜色等属性，实现不同的流动效果。

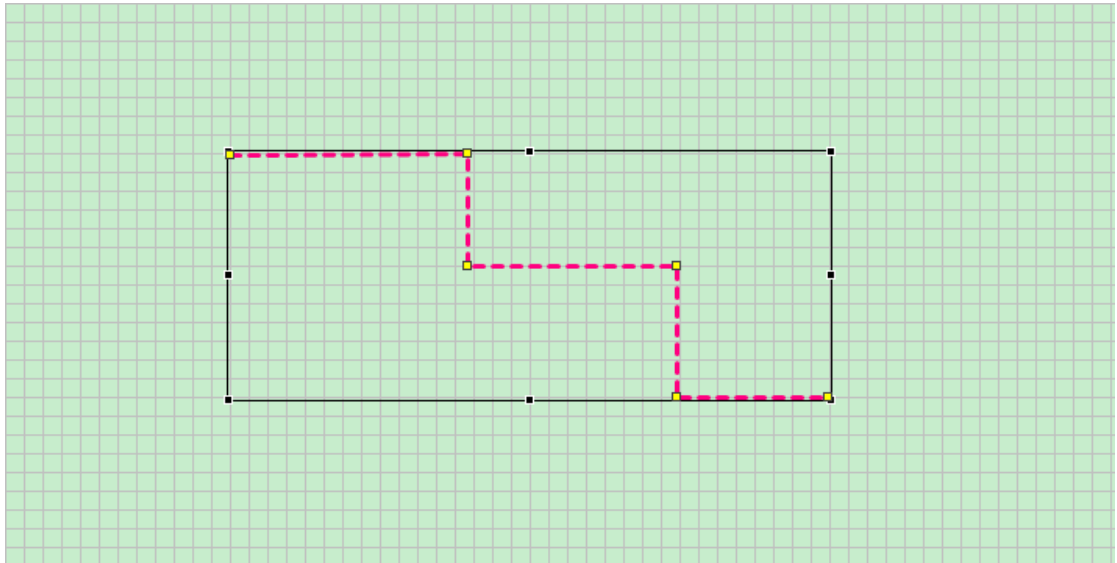


以给折线添加流动效果为例，进行说明：

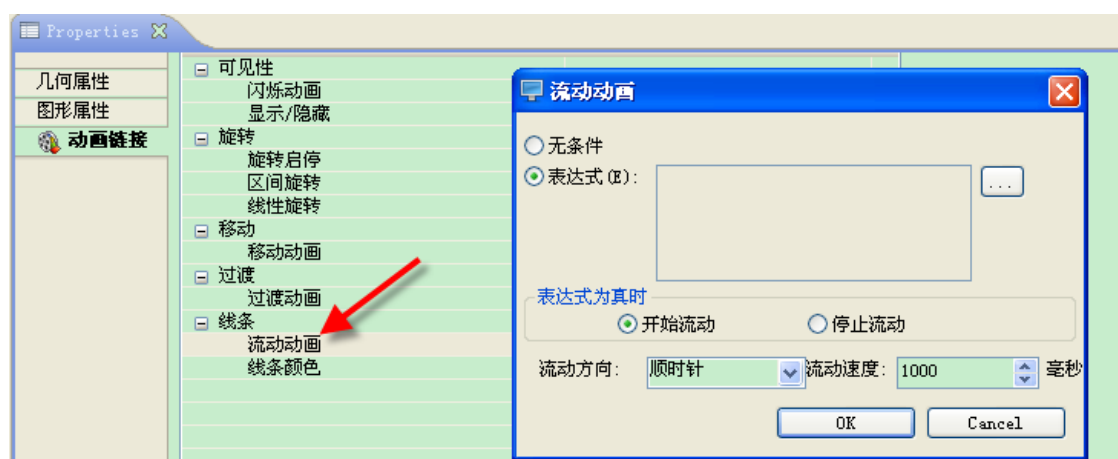
1、在画布中创建折线



2、在“图形属性”中设置线条的图案为“02”、粗细为 3、线端为“圆形”、颜色等属性。



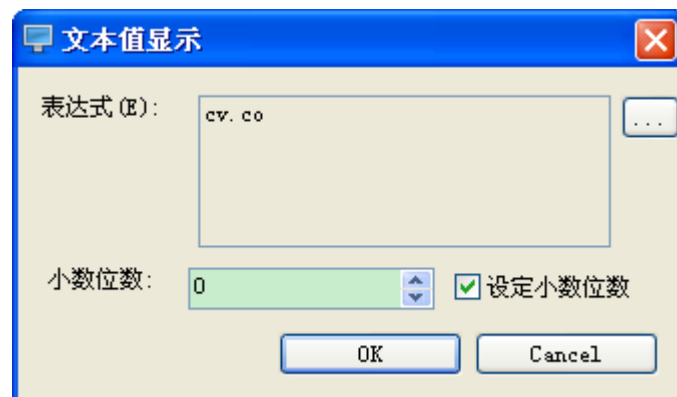
3、在“动画链接”中选择“流动动画”，打开“流动动画”对话框。



4、设置流动动画的启动条件、流动方向和流动速度，点击“确定”按钮，完成设置。

16.8 文本

16.8.1 文本值显示动画



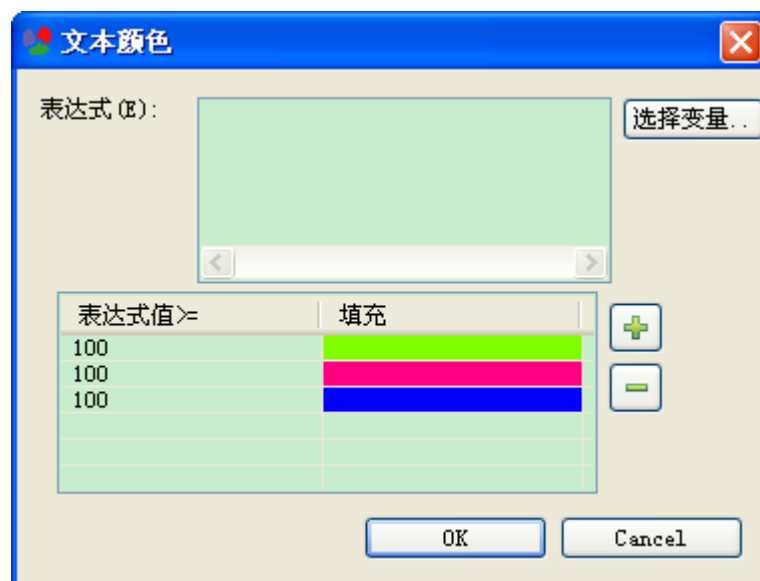
设定文本显示值和小数点的位数。注意只有数字型才能设定小数位数。

16.8.2 区间值显示动画



设定表达式的运行值在不同的区间内，显示不同的字符串值。

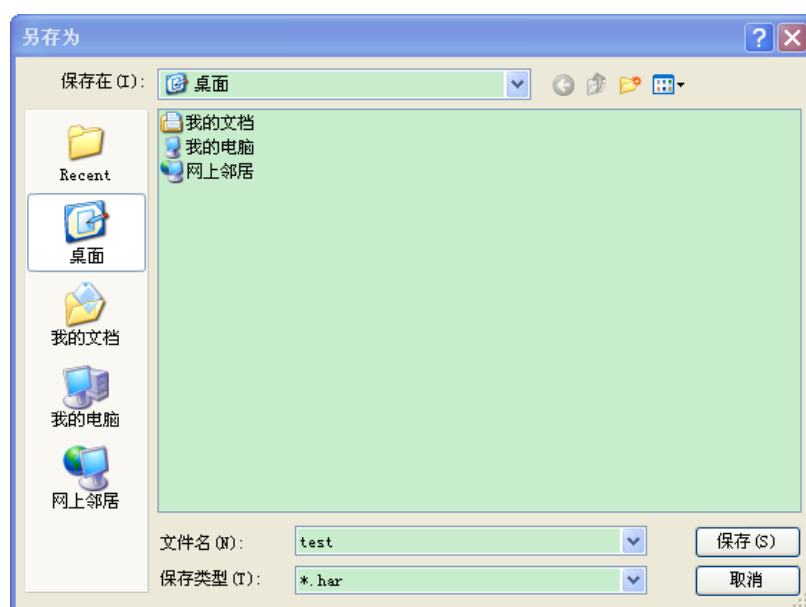
16.8.3 文本颜色动画



设定表达式的运行值在不同的区间内，文本显示不同的颜色。

17 步骤十、发布工程至运行平台

1、在工程浏览器中右击要发布的物联网应用工程，在快捷菜单中选择“发布工程”命令，打开“另存为”窗口。



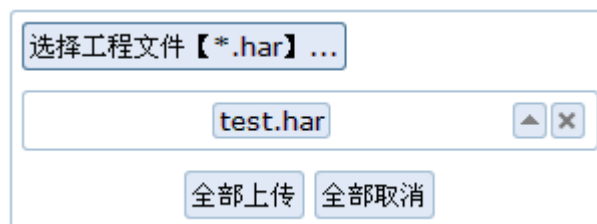
2、整个工程将打包成扩展名为“har”的压缩文件，输入正确的文件名（缺省为工程名称），单击“保存”按钮，完成“发布工程”操作。

3、打开“Hawk 物联网应用平台管理控制台”（URL 类似于

<http://192.168.1.2:8080/hawk> 形式，IP 和端口以实际情况为准)



4、单击“选择工程文件”按钮，打开“打开”窗口，浏览选择工程发布后的 har 文件，单击“全部上传”按钮，将其上传至运行平台，无需重新启动运行平台，对于图形元素发生变化的监控页面，需要用浏览器重新打开监控页面。



18 步骤十一、调试监控页面

18.1 模拟调试



数据模拟器可以对设备某个属性单次发送一个值，进行功能调试，从而方便工程实施。

- 1、在设备列表中选中某个设备的某个属性。
- 2、在输入框中输入某个值
- 3、单击“发送模拟值”发送数据至运行平台。

18.2 实时快照



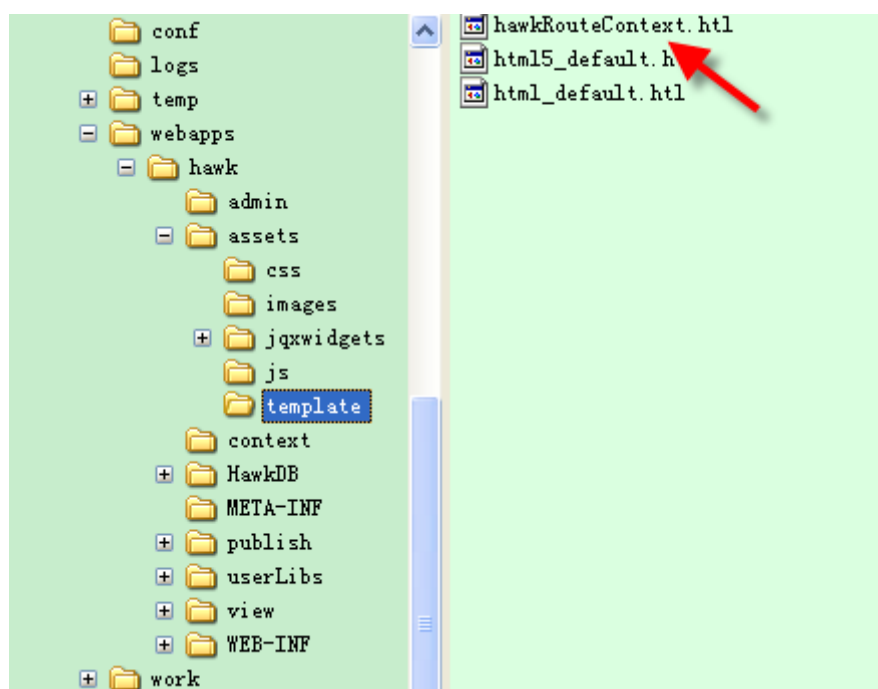
设备实时快照可以查看当前所有设备属性的最新值，方便工程实施。

第三篇 开发篇

19 第三方程序如何发送数据至 HawkRuntime

19.1 如何配置 UDP 的端口号和 IP

- 1、打开 hawk/assets/template 目录下的 hawkRouteContext.html 文件



- 2、编辑 IP 和端口号（文件的其它地方不要改动），如下图所示：

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:camel="http://camel.apache.org/schema/spring" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans
    http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring.xsd">

  <bean id="hawkDTSPProcessor" class="com.jintuotech.hawk.runtime.core.DTSPProcessor"/>
  <bean id="hawkJSONDTSPProcessor" class="com.jintuotech.hawk.runtime.core.JSONDTSPProcessor"/>

  <bean id="length-decoder" class="org.apache.camel.component.netty4.ChannelHandlerFactories" factory-method="lengthDecoder"
    <constructor-arg value="1048576"/>
    <constructor-arg value="0"/>
    <constructor-arg value="4"/>
    <constructor-arg value="0"/>
    <constructor-arg value="4"/>
  </bean>
  <bean id="string-decoder" class="io.netty.handler.codec.string.StringDecoder"/>

  <camelContext id="camelContext" xmlns="http://camel.apache.org/schema/spring">
    <route>
      <from uri="netty4:udp://127.0.0.1:50000?decoders=#length-decoder,#string-decoder&sync=false" />
      <process ref="hawkJSONDTSPProcessor"/>
    </route>
  </camelContext>

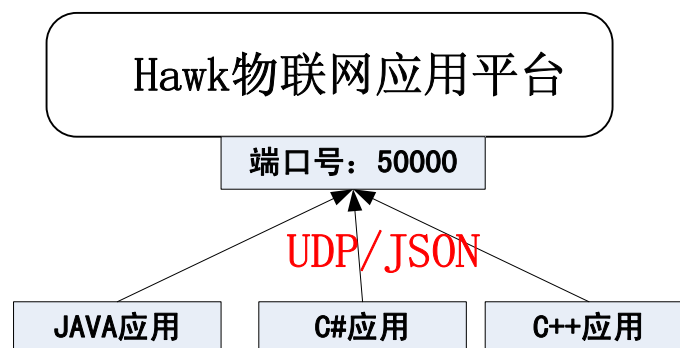
```

端口号

IP

3、重新发布工程至运行平台即可。

19.2 客户端例程



Hawk 物联网应用平台通过 UDP 协议在 50000 端口监听来自于其他应用的 JSON 数据。

JAVA 客户端例程如下所示：

```

private void send(String ip) throws IOException {
    Map<String, Object> dataMap = new HashMap<String, Object>();
    dataMap.put("pi001.PV", 298);
    dataMap.put("pi001.timestamp", 298);
    String content = JSONObject.fromObject(dataMap).toString();
    DatagramSocket clientSocket = new DatagramSocket();
    InetAddress IPAddress = InetAddress.getByName(ip);

```

```
byte[] sendData = new byte[1024];
sendData = content.getBytes("utf-8");
DatagramPacket sendPacket = new DatagramPacket(sendData,
        sendData.length, IPAddress, 50000);
clientSocket.send(sendPacket);
clientSocket.close();

}
```

客户端程序将需要发送的数据设置在 Map 中, 根据指定的 IP 地址发送到 Hawk 物联网应用平台上。

20 如何为监控画面上的设备增加右键菜单项

右键菜单项大多与项目的实际业务需求有关, 需要通过 JavaScript 代码定制开发实现。

位于 HawkRuntime 运行时 assets\js\目录中的 hawk.contextMenu.js 文件, 用来为 Hawk 物联网应用平台的监控画面添加右键菜单。

举例如下所示:

```
function initPopupMenu() {
    $("[device='ts']").contextPopup({
        title: ' 交通信号灯',
        items: [
            {label:'绿灯',icon:'../images/152.png',action:function(evt){menuAction(evt.target,'green')}}},
            {label:'黄灯',icon:'../images/brightness.png',action:function(evt){menuAction(evt.target,'yellow')}}},
            {label:'红灯',icon:'../images/151.png',action:function(evt){menuAction(evt.target,'red')}}},
            {label:'左转',icon:'../images/navigationpng',action:function(evt){ menuAction(evt.target,'turn') }} }
        ]
    });
    $("[device='ls']").contextPopup({
        title: ' 车道指示器',
        items: [
            {label:'正向通行',icon:'../images/arrow.png',action:function(evt){menuAction(evt.target,'front')}}},
            {label:'反向通行',icon:'../images/arrow.png',action:function(evt){menuAction(evt.target,'back')}}},
            {label:'左向通行',icon:'../images/arrow.png',action:function(evt){menuAction(evt.target,'left')}}},
            {label:'禁止通行',icon:'../images/cross.png', action:function(evt){ menuAction(evt.target,'stop') }} }
        ]
    });
}
```



该例中为交通信号灯和车道指示器两类设备分别添加了不同的右键菜单项。