

© 2018-2020 Lovato Electric S.p.A.

如有更改，恕不通知

本文包含的信息仅供参考。尽管我们采取措施检验本文所包含信息的准确性，但信息按“原样”提供，不具备任何类型的保证。

第三方品牌和名称是其各自所有者的财产。

Microsoft[®]、Windows[®]、Windows XP、Windows Vista、Windows 7、Windows 8、Windows 10、Visual Studio 是 Microsoft Corporation 在美国或其他国家的注册商标或商标。此处提到的其他产品和公司名称是其各自所有者的商标。

此处介绍的示例公司、组织、产品、域名、电子邮件地址、徽标、人员、地点和事件均为虚构。我们没有意图也不得推断与任何真实公司、组织、产品、域名、电子邮件地址、徽标、人员、地点和事件的关联。

内容

1 开始	1	区分页面	66
假设	2	6 项目属性	73
安装应用程序	2	Runtime	74
2 Runtime	7	插件	79
HMI 设备基本设置	8	项目	80
背景菜单选项	8	网页	83
内置 SNTP 设备	11	事件	85
3 我的第一个项目	13	区域设置	85
工作空间	14	7 HMI 模拟器	87
创建一个项目	15	数据模拟方法	88
设计一个页面	17	模拟器设置	88
控件库	18	启动和停止模拟器	89
标签控件	20	8 向 HMI 设备传输项目	91
数据字段控件	26	下载到 HMI 设备	92
信息控件	29	更新包	94
将控件附在标签上	34	Runtime 加载器	97
对话框页	35	上传项目	98
4 编制程序概念	37	9 标签编辑器	99
数据类型	38	通信协议	100
“附属于”参数	39	添加标签	101
公式	44	导出标签	104
事件	48	导入标签	104
控件定位	51	标签查找和重命名	108
管理重叠控件	52	标签查找和替换	110
编组控件	53	10 索引寻址	113
更改多个控件属性	60	创建一个索引地址集	114
根据标签值更改填充颜色属性	61	使用页面中的索引标签集	117
5 页面	63	11 标签索引	119
统一页面	64	在标签索引窗格更新数据	123
项目验证程序	64	12 系统变量(附加至)	125

报警变量	127	FTP 动作	175
响铃变量	127	页面动作	178
通信变量	128	打印动作	183
夏令时间变量	128	配方动作	185
设备变量	129	远程客户动作	189
转储信息变量	130	系统动作	190
FTP客户端变量	131	标签动作	202
键盘变量	131	趋势动作	204
网络变量	131	文本编辑器动作	208
打印变量	132	用户管理动作	208
远程客户变量	133	控件动作	211
版本变量	134	15 LRH SW Client	215
屏幕变量	134	PC上的客户端应用程序	216
SD卡变量	134	HMI上的客户端应用程序	217
服务器变量	135	设置和时区选项	218
时间变量	135	16 使用集成的FTP服务器	223
触摸屏变量	135	FTP 设置	223
USB 驱动变量	136	17 VNC 远程访问	225
用户管理变量	137	在 Linux 设备上启动 VNC 服务器	226
13 系统变量(协议)	139	启动 VNC 查看器	226
协议编辑器设置	140	18 报警	227
标签导入	140	报警编辑器	228
默认变量	141	远程报警确认	231
保留内存变量	154	报警状态机	231
服务变量	160	设置 事件	232
PLCM09 变量	163	激活报警控件	234
14 动作	167	报警历史控件	239
报警动作	168	其他报警控件	239
数据库动作	169	运行时管理报警	240
事件动作	172	运行时启用/停用报警	240
多语言动作	172	显示实时报警数据	241
键盘动作	173	输出报警缓存到 . csv 文件	243
多媒体播放器动作	175	导出报警配置	243

19 配方	247	循环调度	300
管理配方	247	为调度表配置地址	302
配置配方控件	250	配置计划任务控件	303
配方状态	251	运行时计划任务事件	304
上传 / 下载一个配方	252	25 21 CFR 第 11 部分合规性	305
备份和恢复配方数据	253	X.509 证书	309
20 趋势	255	已签名 CSV 文件	311
数据记录	256	已签名 PDF 文件	313
导出趋势缓存数据	260	符合应用程序	316
实时趋势控件	262	26 用户管理和密码	317
历史趋势控件	263	启用/停用安全管理	318
散布图控件	265	配置组和授权	318
趋势控件手势	266	修改访问权	319
表格趋势控件	272	从页面视图指定控件许可	324
21 数据传输	279	配置用户	324
数据传输编辑器	280	默认用户	326
导出数据到 .csv 文件	282	运行时管理用户	326
数据传输限制和建议	282	强制远程登录	327
22 离线节点管理	285	27 日志	329
离线节点管理过程	286	启用/停用日志	330
手动离线节点管理过程	286	电子签名	331
手动离线配置	286	表格日志控件	334
自动脱机节点检测	286	导出日志到 .csv 文件	335
23 多语言	289	配置日志事件	336
多语言编辑器	291	28 报表	339
更改语言	292	添加一个报表	340
多语言控件	292	配置图形报表	340
导出/导入多语言字符串	294	启动打印事件	341
在运行时更改语言	296	默认打印机	342
Unicode 支持限制	296	29 屏幕保护程序	343
24 计划任务	299	30 Runtime 和项目备份/恢复	345
创建一个调度程序	300	31 键盘	347
高分辨率调度表	300	创建和使用自定义键盘	349



删除或重命名自定义键盘	351	Canvas 控件	389
键盘类型	351	组合框控件	392
键盘位置	352	耗量表控件	396
32 外置键盘	355	控制列表控件	397
搜索并过滤	357	日期时间控件	401
显示	357	手势控件	402
删除动作关联	357	IP 摄像机控件	407
键盘布局	358	JavaScript 功能块控件	410
启用/停用键盘	358	多媒体播放器控件	412
将动作与按键关联	358	多状态图像控件	414
33 将数据存储到外部数据库	361	多状态图像多层控件	415
安装 SQL4	362	网络适配器控件	417
配置 SQL4	362	RSS Feed 控件	417
配置 HMI 项目	364	滚动 RSS Feed 控件	418
用 JavaScript 传输数据	364	表格控件	418
数据库表格	366	TextEditor 控件	434
自定义表格	367	变量控件	436
连接限制	367	37 自定义控件	439
34 OPC UA 服务器	369	创建一个自定义控件	440
功能	369	将属性添加到自定义控件中	442
网络	369	使用结构标签	444
验证	369	自定义控件的 JavaScript	447
使用 x.509 证书	370	用户库	450
使用自签名证书	373	38 发送邮件信息	451
使用外部证书	376	配置电子邮件服务器	452
报警映射	378	配置电子邮件	452
35 MQTT 接口	379	39 JavaScript	455
标签配置	379	JavaScript 编辑器	457
主题和负载	381	JavaScript 功能的执行	457
MQTT 代理设置	382	事件	459
36 特殊控件	387	控件事件	460
BACnet 控件	388	页面事件	462
浏览器控件	388	系统事件	463

对象	465	更改 HMI 设备密码	514
控件级对象	465	端口和防火墙	514
控件属性	466	43 改善性能的提示和技巧	517
控件函数	469	静态优化	518
页面对象	470	静态优化方面的常见问题	521
页面对象属性	470	页面高速缓存	521
页面对象函数	471	图像数据库	521
项目对象	473	预缓存	522
项目对象属性	473	预缓存常见问题	522
项目对象函数	474	44 功能规格说明与兼容性	523
项目对象控件	482	功能限制表	524
打印报表对象	483	HMI 设备地址	525
组对象	486	兼容性	525
组对象函数	487	在不同的 HMI 设备间转换项目	526
状态对象	488	45 系统设置	527
状态对象函数	488	Linux 设备	528
关键字	489	46 更新 HMI 设备中的系统组件	551
全局函数	489	从应用程序更新系统组件	552
处理读写文件	490	设置	552
从 JavaScript 登录	493		
在 JavaScript 控件操作的限制	495		
JavaScript 调试	495		
40 处理手势	499		
41 网页连接	501		
支持的平台和浏览器	502		
Web 页面	502		
测试 Web 项目	504		
下载网页项目	504		
网络连接问题	506		
网络支持的功能	507		
检修和 FAQ	510		
隐私	511		
42 保护 HMI 设备使用	513		

1 开始

LRH SW 是一个为创建图表 HMI 页面而设计的软件应用程序。LRH SW 有拖放界面，便于轻松创建复杂页面。在通用 Windows 应用程序中发现的许多功能也适用于 LRH SW。

本文档分为不同章节，描述 LRH SW 的主要功能，并解释如何使用这些功能。每一个章节以独立的方式呈现，您可以根据当前的任务从一章跳到另一章。

假设	2
安装应用程序	2

假设

我们假设读者对计算机、Microsoft Windows 以及应用程序将运行的特定网络环境有基本的了解。

安装应用程序

LRH SW 安装包括：

- **LRH SW:** 一个应用程序，用于以用户友好方式设计自定义 HMI 项目、其内置库中的各种对象以及控件库。
- **LRH SW Client:** 一个轻量级的应用程序，可以在 Windows 计算机上使用，远程查看和管理在 HMI 设备上运行的项目。
- **LRH SW HMI Runtime:** 在 HMI 设备上运行的独立应用程序。HMI Runtime 通过 LRH SW 来安装。

LRH SW 系统要求

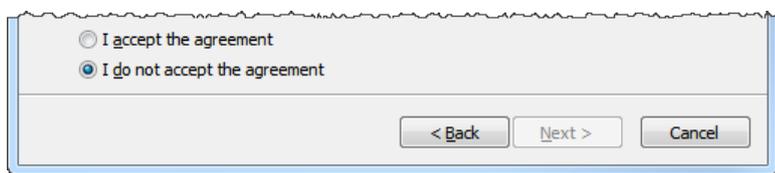
LRH SW 有以下系统要求：

操作系统	Windows Embedded Standard (WES 2009) Windows Server 2003 Windows Vista Business/Ultimate Windows 7 Professional Windows Embedded Standard 7 Windows 8 Windows 10
空间	最小 500 MB
随机存取存储器	512 MB
其他	一个以太网连接

安装步骤

安装 LRH SW：

1. 运行 LRH SW 设置并单击下一步。
2. 阅读 LRH SW 软件许可证并同意协议。



3. 按照屏幕上的说明做。默认的 c 软件位置为 *C:\Program Files\Lovato Electric\LRH SW*，如果需要，可以更改路径。
4. 如果选择组件可用，选择要安装的组件。
5. 选中**创建桌面图标**选项，在桌面添加一个 LRH SW 图标。一个 LRH SW 组将被安装过程自动添加到开始菜单中。



6. 要运行该应用程序，请单击桌面图标或选择**开始 > 所有程序 > LRH SW**。

试用版

LRH SW 有 30 天的免费试用期。安装 30 天之后会显示一个注册表格，以输入许可证激活密钥。



注：试用版在虚拟机上不受支持，只能使用有效的许可证。

许可证

要在试用期结束前注册软件，请转到**帮助 > 注册**。



注：注册过程要求有 Internet 连接。端口 TCP 80 和 443 用于激活。

注册时，许可证文件从许可证服务器下载到电脑。许可证文件根据 OS 保存在以下文件夹中：

%appdata%\Lovato

许可证锁在 **BIOS ID** 中或锁在安装软件的计算机的 **Windows 产品 ID** 中。

如果 LRH SW 无法到达授权服务器(例如，无 Internet 连接可用)，则会显示一个按钮以通过电子邮件激活许可证。

Registration form fields:

- First Name*
- Last Name*
- Job Title
- Company*
- Division or Group
- Industrial Sector (Please choose one)
- Country* (Italy)
- State/Province* (Lazio)
- City* (Rome)
- Phone
- E-Mail* (email@email.it)
- Confirm E-Mail* (email@email.it)
- Terms of Service* (Yes/No)
- Privacy Statement* (Yes/No)
- Activation Key* (XXXXX-XXXXX-XXXXX-XXXXX)
- HostID (BIOS ID: 8DFZG12, ETHERNET ID: ECF4BB4FF8DC, ETHERNET ID: C4D987F1A5EF, ETHERNET ID: 00A0C6000021)

Error: Unable to contact license server. You may request License by email. A valid license will be emailed back to you.
Check your license status [here](#)

Buttons: Send Mail, Register, Close

按“发送邮件”按钮，LRH SW 将显示以下表格：

Email composition details:

- To: license@x-formation.com
- Subject: Request License for LRH SW
- Body: ACTIVATION_KEY:jkhjhjk
HOSTID:BIOS=564d2eb0e27f2ba9-520f56163a7a086f,ETHERNET=000C297A086F

Note: A license will be emailed to you after verifying the activation code is valid and registering the product. Please place the license file in the folder C:\Users\User\AppData\Roaming\Lovato/

Button: OK

在有网络连接时，此电子邮件将在几秒内发送。您也可以激活许可证，然后从以下网站下载许可证文件：<https://licenses.x-formation.com/licenses>。报告在以上表单“正文”中包含的数据。

验证许可证状态

要检查您的许可证的状态：

1. 请点击：<https://license.x-formation.com/licenses>
2. 输入您的激活密钥并单击 **登录** 按钮。

安装多个版本的 LRH SW

您可以在同一台计算机上安装 LRH SW 的不同实例。每一种安装都有各自的设置，可以单独卸载。

三种安装方案可行：

安装方案	结果
在系统中首次安装 LRH SW	软件安装在指定的目标文件夹中
只安装了一个 LRH SW 实例的系统	当前版本可以替换或保留。
安装了多个 LRH SW 实例的系统	上一个安装的版本可以替换或保留。

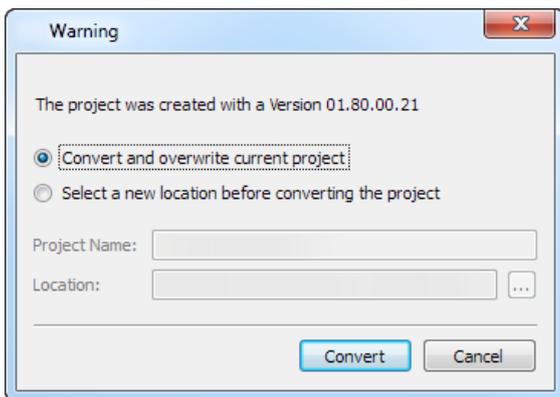
如果尝试再次安装 LRH SW 的已安装版本实例，会显示警告消息。

多个 LRH SW 安装共享一个公共工作空间文件夹，每个子文件夹都有版本号，例如 *C:\Program Files\Lovato Electric\LRH SW 2.5*。每一个安装的版本有其各自的 ID，因此可以独立删除。

每一个安装都在 Windows 开始菜单中单独列出。

打开旧的项目

打开用旧版本软件创建的 LRH SW 项目 (.jpr 文件) 时，LRH SW 会要求将项目转换成当前版本：



选项	说明
转换并覆盖当前项目	项目已转换，原始版本未备份。
转换项目前选择一个新位置	将项目复制到指定文件夹内，然后再转换项目。



警告：请勿使用比创建项目所用版本更旧的 LRH SW 来编辑项目。这会损坏项目并引起 runtime 不稳定。

多语言 LRH SW

LRH SW 支持多种语言。所有语言默认作为 LRH SW 的一部分安装。

默认语言是英语。要进行更改，请转到 [帮助 > 更改语言](#)。

事故报告

LRH SW 停止或崩溃时，显示事故报告对话框。



重要:始终保存事故报告文件，因为其中可能包含对技术支持有用的信息。



注:Windows XP 系统中不支持事故报告。

2 Runtime

LRH SW HMI Runtime 的设计目的是支持不同的平台和不同的操作系统。

HMI 设备基本设置	8
背景菜单选项	8
内置 SNTP 设备	11

HMI 设备基本设置

HMI设备出厂时没有 runtime。如果设备上未安装 Runtime 时，请参阅"[Runtime 加载器](#)"在本页 97 了解详细信息。

Runtime 模式

LRH SW HMI Runtime 由两个逻辑单元组成：

- **服务器**：运行通信协议、收集数据、监控报警、驱动趋势缓存取样。
- **客户端**：显示服务器收集的数据。

服务器单元负责处理 HMI 设备，如通信协议、执行数据获取、驱动趋势缓存取样活动、监控报警等等。

客户端单元负责可视化过程：使用服务器收集到数据以图表信息的形式显示。

服务器单元以两种操作模式工作：

- **配置模式**：服务器闲置（例如当设备上没有加载项目或一些系统文件丢失时）。
- **操作模式**：服务器按照系统文件和加载的应用程序项目定义的设置操作。



注：即使服务器上未运行活动，客户端上也可能显示数据。

背景菜单选项

在 HMI 设备上按住屏幕上的空白区域并持续几秒钟，显示背景菜单。

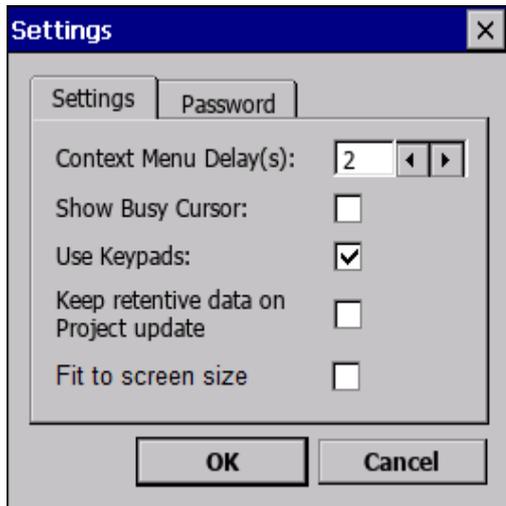
放大/缩小

运行时选择视图尺寸

平移模式

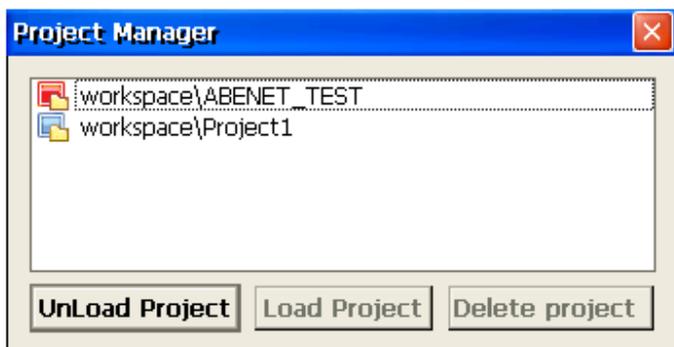
放大后启用/停用平移模式

设置



主要参数	说明
上下文菜单延迟	上下文菜单激活延迟。范围：1–60 秒。
显示忙碌光标	当系统忙碌时显示一个沙漏
使用键盘	当用户触及数字输入字段时显示键盘。 当外接 USB 键盘连接到设备时设置为 禁用 。
项目更新时保留保持数据	下载或更新项目时保留保持数据的内容。
缩放至屏幕大小	将视图调整为屏幕大小
密码	限定以下需要密码保护的操作： <ul style="list-style-type: none"> • 下载项目 / runtime • 上传项目 • 面板管理(BSP, 更新) 请参阅"保护 HMI 设备使用"在本页 513 了解详细信息。

项目经理



这一工具让您可以：

- 卸载当前项目
- 加载另一个项目
- 删除一个项目。

当您加载一个新项目的时候，当前项目会自动卸载。删除一个项目前，您必须将其卸载。

更新

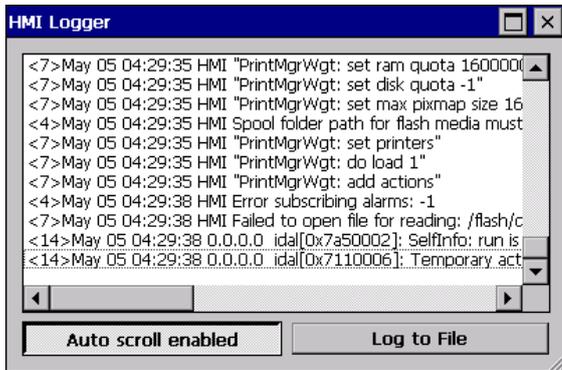
这一功能从外置 USB 装置加载更新包。请参阅["更新包"](#)在本页 94 了解详细信息。

备用

您可以创建 Runtime 和项目的备份。

日志

该功能显示系统操作的记录。



单击**记录到文件**保存数据：logger.txt 文件将保存至 ...\\var\\log 文件夹中。

该文件可以使用 FTP 客户端找回，并且可以抄送给技术支持。



注：一旦启用，记录在重启后仍会保持，必须手动禁用。

在启动时显示记录

该功能使得在启动时会启用记录器。如果**记录到文件**选项已启用，则将从启动时保存日志文件。

注销

注销当前用户。

显示系统设置

允许显示 HMI 设置并管理系统组件。请参阅["系统设置"](#)在本页 527 了解详细信息。

开发人员工具

运行时进行调试的实用功能。仅当已在“项目属性”中启用时才可见(请参阅“开发人员工具”在本页 76了解详细信息)

关于

显示关于 runtime 版本的信息。



警告: 如果从对话框页面执行,上下文菜单动作不起作用。

内置 SNTP 设备

HMI 设备采用集成 SNTP,任何时候有可用的预定义服务器时,会同步内部实时时钟面板。

使用 HMI 设备的“系统设置”在本页 527配置该服务。

3 我的第一个项目

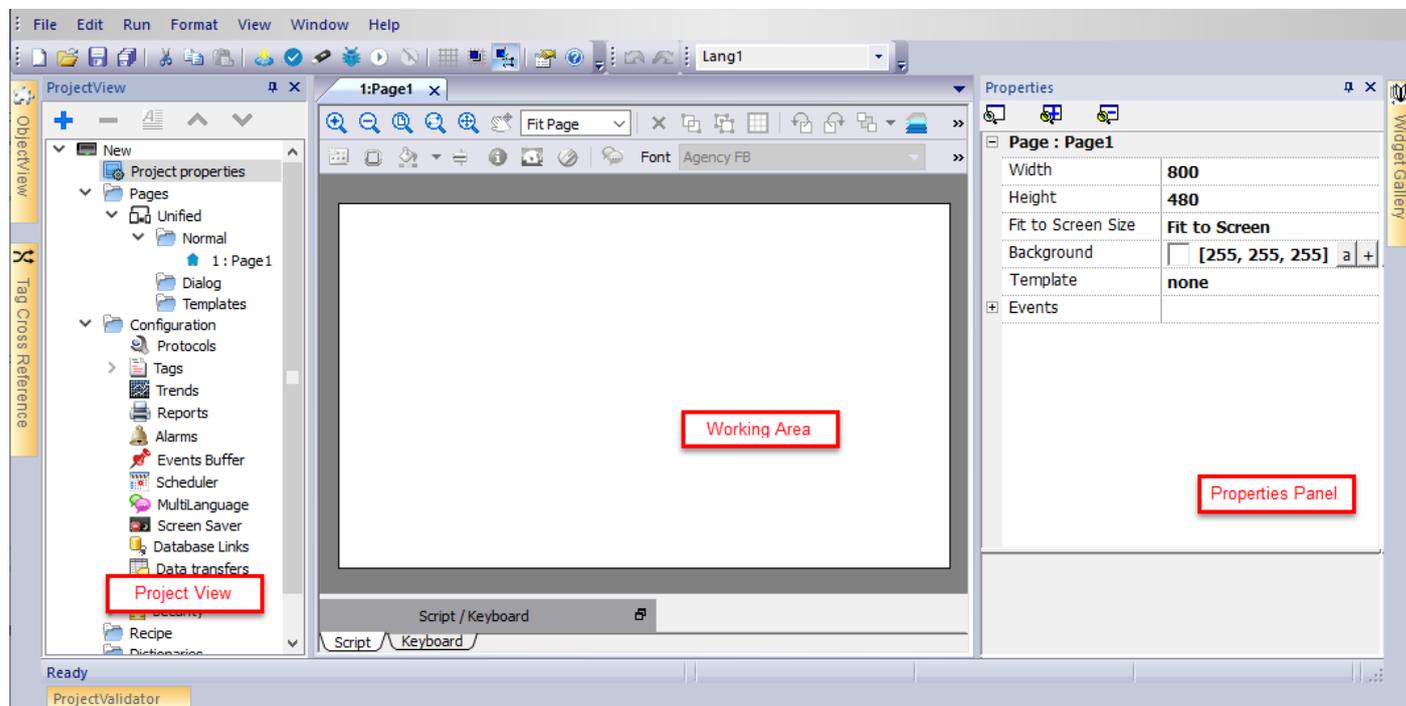
该部分描述如何创建一个简单的 LRH SW 项目。

工作空间	14
创建一个项目	15
设计一个页面	17
控件库	18
标签控件	20
数据字段控件	26
信息控件	29
将控件附在标签上	34
对话框页	35

工作空间

工作空间区域

LRH SW 工作空间分为以下主要区域：



区域	说明
项目视图	以等级项目树形显示的项目元素。
对象视图	页面组织的控件树形视图。
工作区域	页面可供编辑的区域。该区域顶端的选项卡显示所有打开的页面。
属性	所选对象的属性。
小控件库	图形对象和符号库
标签索引	给出的标签参照的位置列表。
项目验证程序	项目验证器用于列出与项目相关的警告消息的区域



注：工作空间布局在任何时候都可以更改，更改通过工作会话保存和维护。

复位工作空间布局

要恢复默认布局，请使用文件 > 复位和重启功能。

创建一个项目

路径: 文件 > 新项目

1. 在 **项目向导** 对话框中, 输入项目和存储位置的名称。
2. 单击 **下一步**: “HMI 设备选择”对话框显示。
3. 从可用型号列表中选择 **一个设备**。
4. 选择设备方向。
5. 选择要创建的项目模板
5. 单击 **完成** 完成向导。

纵向旋转特例

以下元素在纵向模式下不旋转。

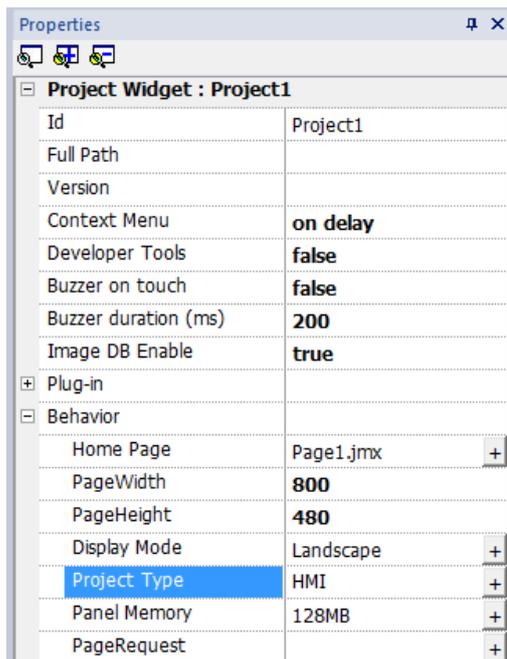
元素	说明
操作系统对话框	系统设置和系统对话框
上下文菜单 (ContextMenu) 和相关对话框	项目管理员、关于设置、日志、备份
视频	IP 摄像机、多媒体播放器
JavaScript	报警和打印功能
对话框页	对话框页的“标题”
计划任务	数据输入对话框。
程序安排	ShowMessage、LunchApplication、LunchBrowser
外部应用程序	PDF 阅读器



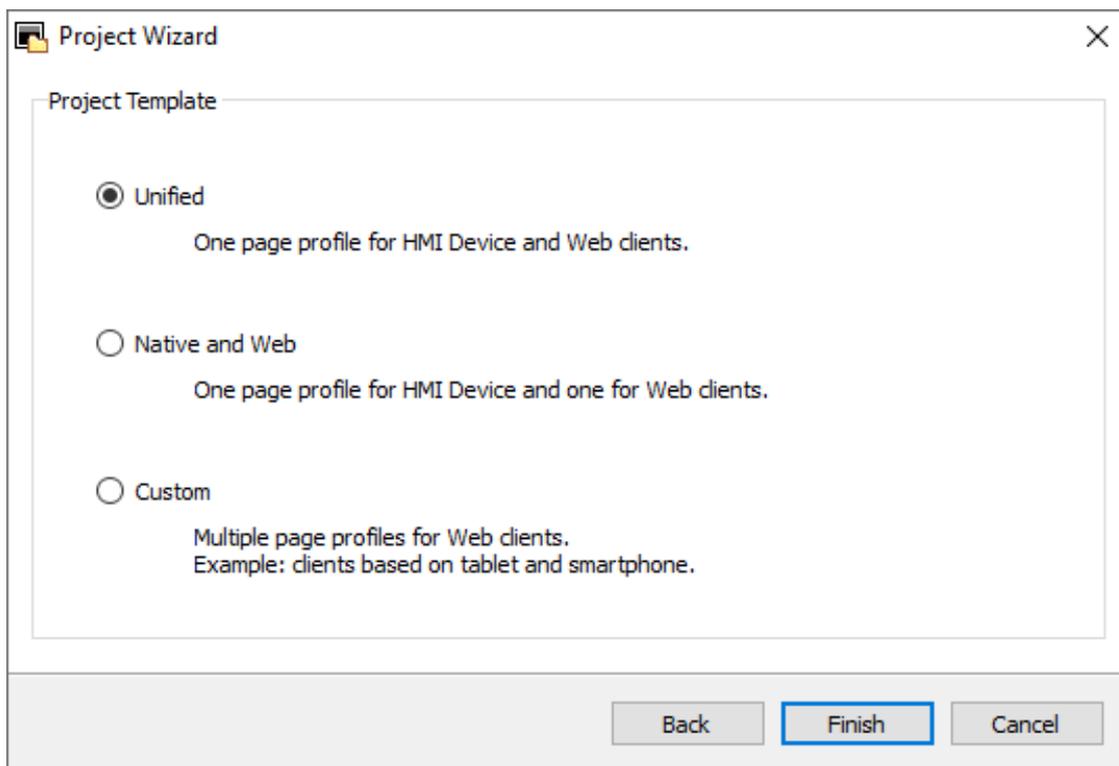
可以不受限制地从 BSP 中旋转基于 Linux 平台的 HMI 设备(请参阅 ["系统设置" 在本页 527](#)“Linux 设备”
在本页 528 中的“显示”选项卡)。

更改设备型号

一旦开发了项目, 您仍然可以从项目属性窗格更改设备型号。这不会改变控件的大小, 但是会在屏幕上改变位置。如果一些对象不能重新放置, 会出现警示。



项目模板



“项目模板”为项目页面建议并创建最常见的文件夹结构。您稍后随时可以修改结构。

建议的结构为：

元素	说明
统一	创建一个唯一的文件夹，其中将包含项目的所有页面。可以在 HMI 设备，远程客户端和 Web 客户端上使用相同的页面。如果要在所有平台上具有相同的页面，请使用此选项。
本机和网络	创建两个文件夹，一个文件夹包含将在 HMI 设备上使用的项目页面，另一个文件夹包含用于将在 Web 客户端上使用的页面。如果要在 Web 客户端上具有不同的页面，请使用此选项。
自定义	可以创建不同文件夹来包含要在 HMI 设备，Web 客户端，Table 客户端和 Smartphone 客户端上使用的页面。

复制、移动、重命名项目

LRH SW 项目文件夹包含项目的所有文件：要移动、复制或备份项目，请将项目文件夹移动或复制到想要的位置。

要重命名项目，请使用文件 > 项目另存为功能：该操作可能需要几分钟。



警告：请勿手动重命名项目文件夹。

设计一个页面

路径：项目视图 > 页面

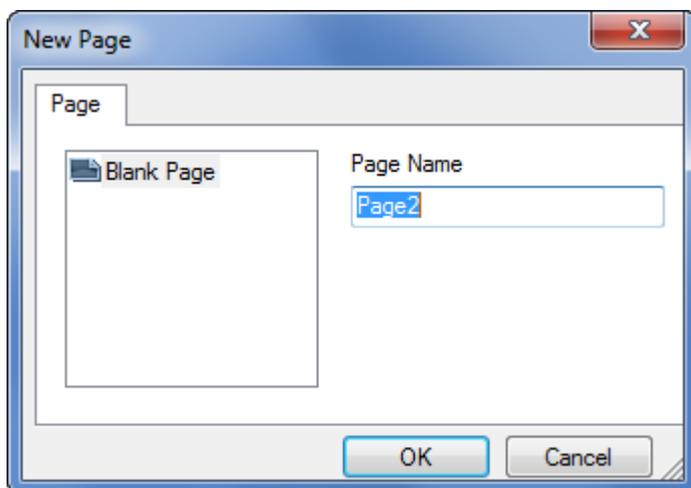
创建项目时，第 1 页自动添加并显示在页面编辑器中。

添加一个对象到页面

将对象从控件库拖放到页面上。

添加一个页面

1. 从项目树中右键单击页面节点并选择插入新页面。
2. 为新页面输入名称。



导入一个页面

当导入一个页面时，LRH SW 将会导入页面布局和页面控件，不会导入附在控件上的动作和数据链接。您可以在两个不同的项目中选择：

- 只导入页面和控件：在这种情况下，所有动作和数据链接需要定义。
- 导入关于动作和数据链接的页面：使用的标签必须项目中已有，使这些元素正常运行。



注：页面导入只能在使用同一软件版本的项目间进行。另存旧的项目为更新的版本，然后重试。

1. 从项目树中右键单击**页面**节点并选择**导入页面**。
2. 选择要从所需项目中导入的页面，然后单击**确定**：一条警告消息将显示。
3. 单击**是**可移除所有数据链接和动作。单击**否**可保留数据链接和动作参考。新项目中需要的标签。

页面组

您可以对相似页面分组以便更好地维护。页面分组不会影响运行时页面的外观。创建页面组：

1. 在**项目视图**中，右键单击**页面**节点并选择**创建组**：系统将添加一个新文件夹
2. 要将一个页面移动到组，请右键单击该页面并选择**组 > 组名称**。

控件库

路径：**视图 > 工具栏和连接窗口 > 控件库**

控件库中提供有构建应用程序所需的 HMI 对象。库被分成几个种类，每一种包含一个控件集。



添加一个控件到页面

1. 从**控件库**中选择控件。
2. 将其拖拽到页面。

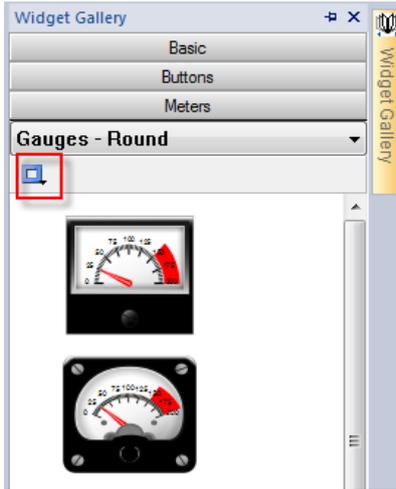
更改控件外观

所有控件的属性(属性窗格)都可以更改,有些控件以不同的样式显示。您可以在每个种类中点击按钮查看可用样式。

例如

要为圆形计量器设置控件样式:

1. 点击样式按钮,显示控件可用的样式。



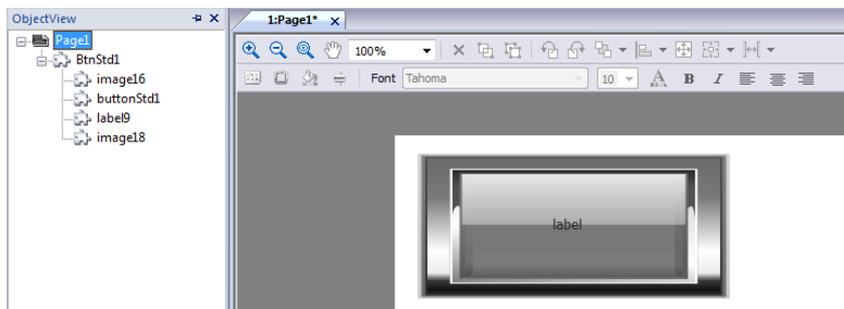
2. 从工具栏选择一个可用的样式:根据选择的控件不同,有不同的选项可用。



复杂的控件

一些控件有许多更小的控件构成。例如：一个按钮是一个由按钮控件和一个标签群组成的复杂控件。在控件被选中时，可以在**对象视图**中看到控件结构。

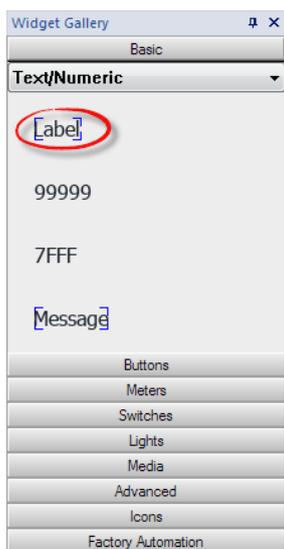
可以从**对象视图**中选择子控件(如按钮上的标签)，并在未取消整个控件的情况下对其修改。



标签控件

标签控件可以显示文本和标签值。

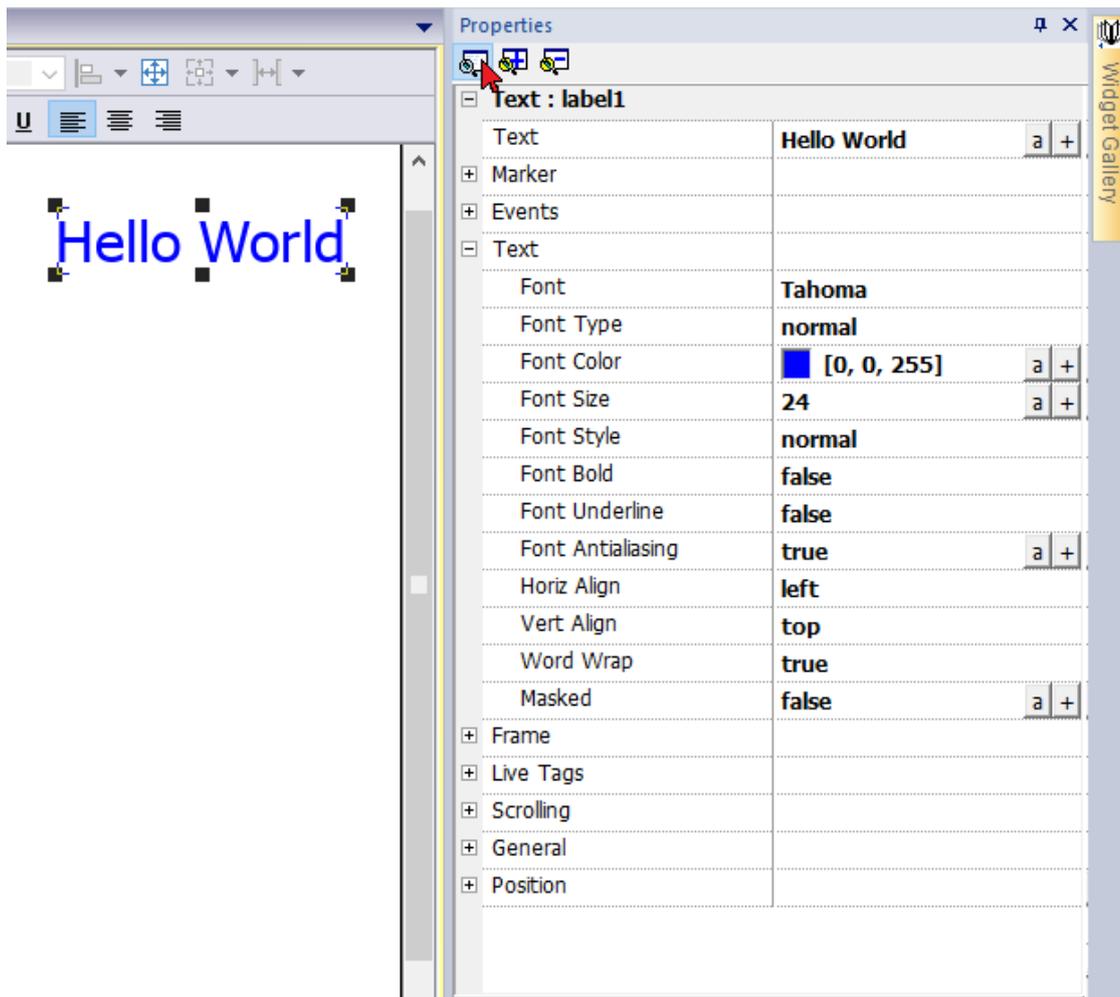
路径：**视图 > 工具栏和连接窗口 > 控件库**



标签属性

将控件拖放到页面内，然后选择控件打开控件的属性对话框。

请注意，只有选择“显示高级属性”按钮时，某些属性才可见。



属性	说明
文本	要显示的字符串。字符串可以是静态的，也可以从 TAG 检索。请参阅 "将控件附在标签上" 在本页 34
标记	启用控件周围的标记(仅在LRH SW内可见)
事件	控件内容更改时执行的动作。请参阅 "事件" 在本页 48
文本(文件夹)	文本属性
边框	用于启用和配置控件框架和/或背景颜色的参数
在线标签	<p>启用文本消息内部的标签值。请参阅 "在线标签" 在本页 24</p> <ul style="list-style-type: none"> - 启用在线标签 启用在线标签占位符 - 动态订阅 如果为 true, 仅从通信协议中检索可见的标签。如果为 false, 即使不可见, 所有标签也会保持不断更新。
滚动	用于启用和配置文本消息滚动的参数

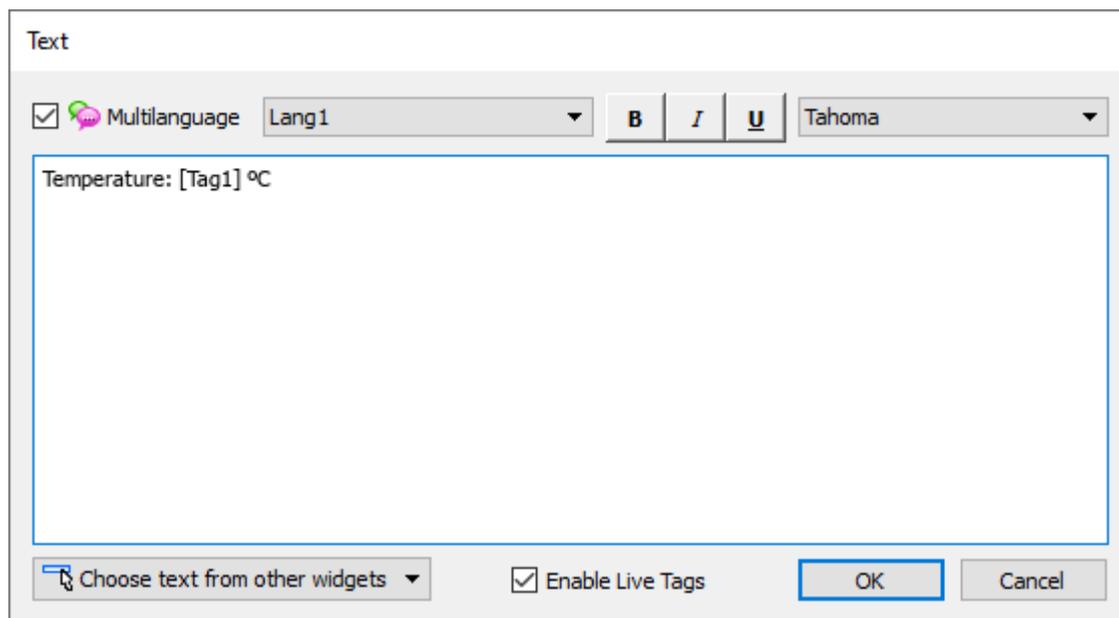
属性	说明
	<ul style="list-style-type: none"> • 无 • 曼 • 标准 • 快 • 自定义 <p>选择自定义模式时，可以定义以下参数：</p> <p>-滚动类型 每次超时，将滚动自定义数量的字符或像素的文本。</p> <ul style="list-style-type: none"> • 字符 • 像素 <p>-滚动延迟 标签有效开始滚动之后的超时时间 (mSec)</p> <p>-滚动计时器 定义每个滚动步骤的超时时间 (mSec)</p> <p>-滚动点或 -滚动字符 每个计时器超时滚动的像素数，或者 每个计时器超时滚动的字符数</p> <p>-滚动行为 只有一次 第一次完成后，文本滚动停止。</p> <p>LoopWithDelay 文本滚动在每个完整的循环后重新开始，等待延迟。</p> <p>Loop 文本滚动永不停止</p>

属性	说明
常规	<p>常规属性</p> <ul style="list-style-type: none"> - Id 控件标识符。您可以保留默认值或将其重命名为更合适的名称 - 可见 如果为 false, 控件不可见 - 不透明度 不透明度级别描述了透明度级别, 其中 1 完全不透明, 0.5 表示透明度为 50%, 0 为完全透明。 - 闪烁 - 文本将闪烁 - 锁定 如果为 True, 则无法选择控件并从 LRH SW 页面编辑器中将其移动 - 静态优化 标准 LRH SW 将决定要使用的最佳优化模式 静态 LRH SW 优化控件, 假设它永远不会被运行时间修改 动态 LRH SW 不会添加其他优化
位置	控件在显示屏上的位置。请参阅 "控件在显示屏上的位置" 在本页 25

一些属性具有几个按钮:

-  进入编辑模式:
您可以直接输入要使用的标签名称
-  附加至标签:
将打开从中选择要使用标签的对话框

双击标签控件将打开编辑对话框, 您可以在其中输入要显示的文本并设置主要文本属性。



在线标签

启用“启用在线标签”，将方括号之间的文本作为标签占位符进行管理，并将在运行时使用标签值进行呈现。

例如，文本标签“*Temperature: [Tag1] °C*”将呈现为：

温度: 18 °C

“18”是 Tag1 包含的值

Tags

- [TagName]
The tag value is read and continuously updated



Use '\ ' before '[' if you want to show the '[' in the description string, for example: `\[Tag\1\]` will display the string “[Tag1]”.

Use '\ ', even when the tag label contains square brackets. For example, to display the live tag value of tag “TAG]3” or “TAG[3]” use:

- TAG\]3 = [TAG\]3]
- TAG\[3] = [TAG\[3]]

Array Tags

To reference the entire array (all elements will be shown):

- [TagName]
All array elements will be displayed using a comma separate list.
- [TagName[-1]]
All array elements will be displayed using a comma separate list.

To reference an element of the array:

- [TagName.Index]
Example: [MyARRAY.5] will display the sixth element of the MyARRAY
- [TagName[TagIndex]]
Example: [TagIndex] will display the sixth element of the MyARRAY when TagIndex is 5

Data Formats

Placeholder characters can be used to control how to display the tag value (see "自定义格式" 在本页 28)

- [TagName|format("###")]

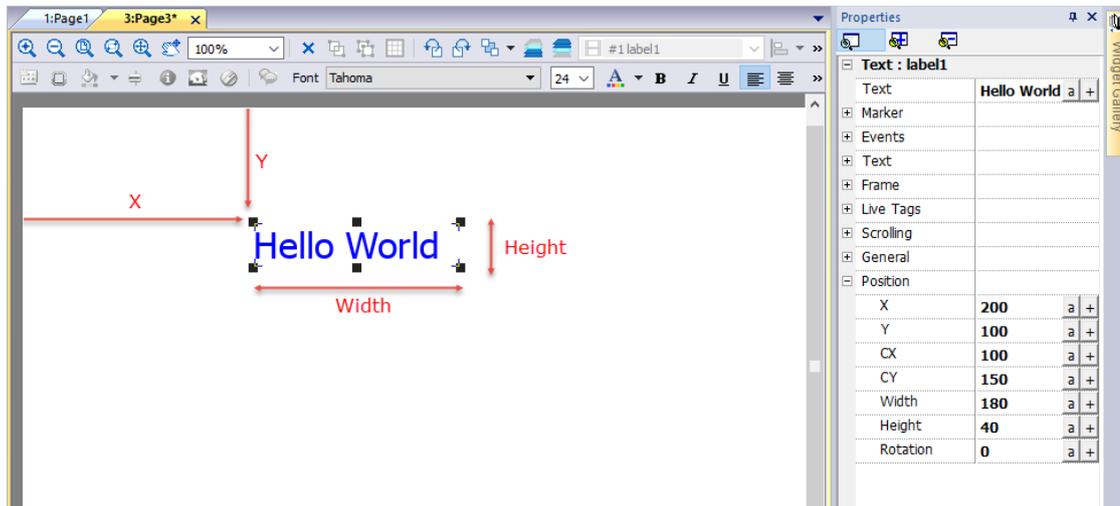
Example:

Live: [fCounter|format("#.00")] - Triggered: [!fCounter|format("#.00")]



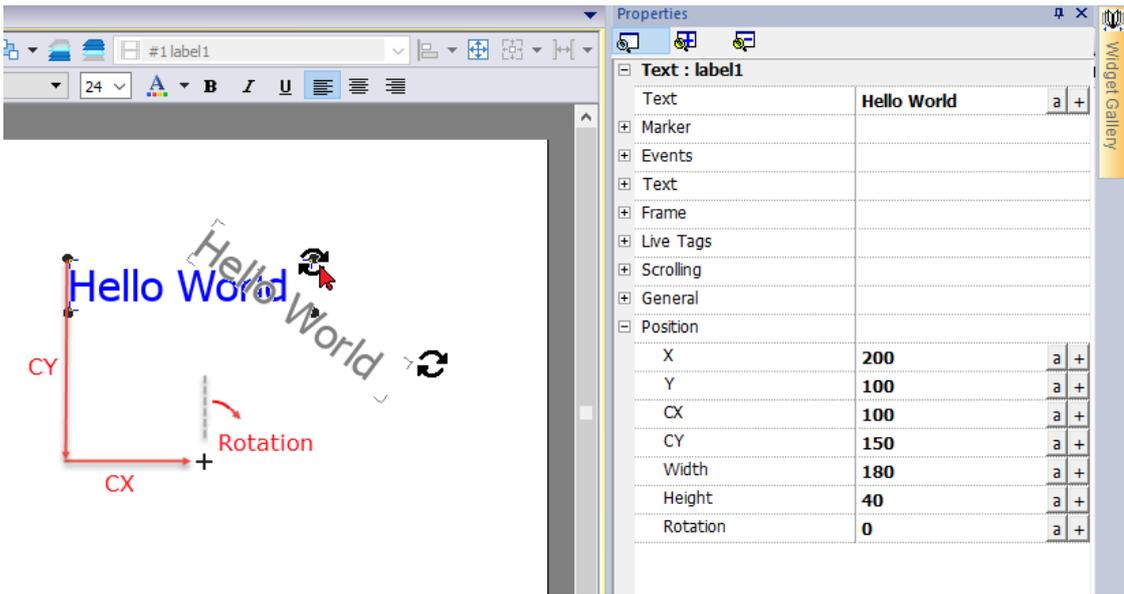
Note that by default, all tags are displayed as an integer. If you want to display a float number, you have to specify how to show the number adding the decimal digits.

控件在显示屏上的位置



旋转

要旋转控件，请单击控件两次。第一次单击后，标记将变为正方形，第二次单击后将变为圆形。现在，在圆形标记上单击鼠标，然后拖放以旋转控件。旋转中心由 CX 和 CY 参数定义。



请注意，所有“位置”属性都可以附加到标签，并且可以在运行时动态修改以移动控件。

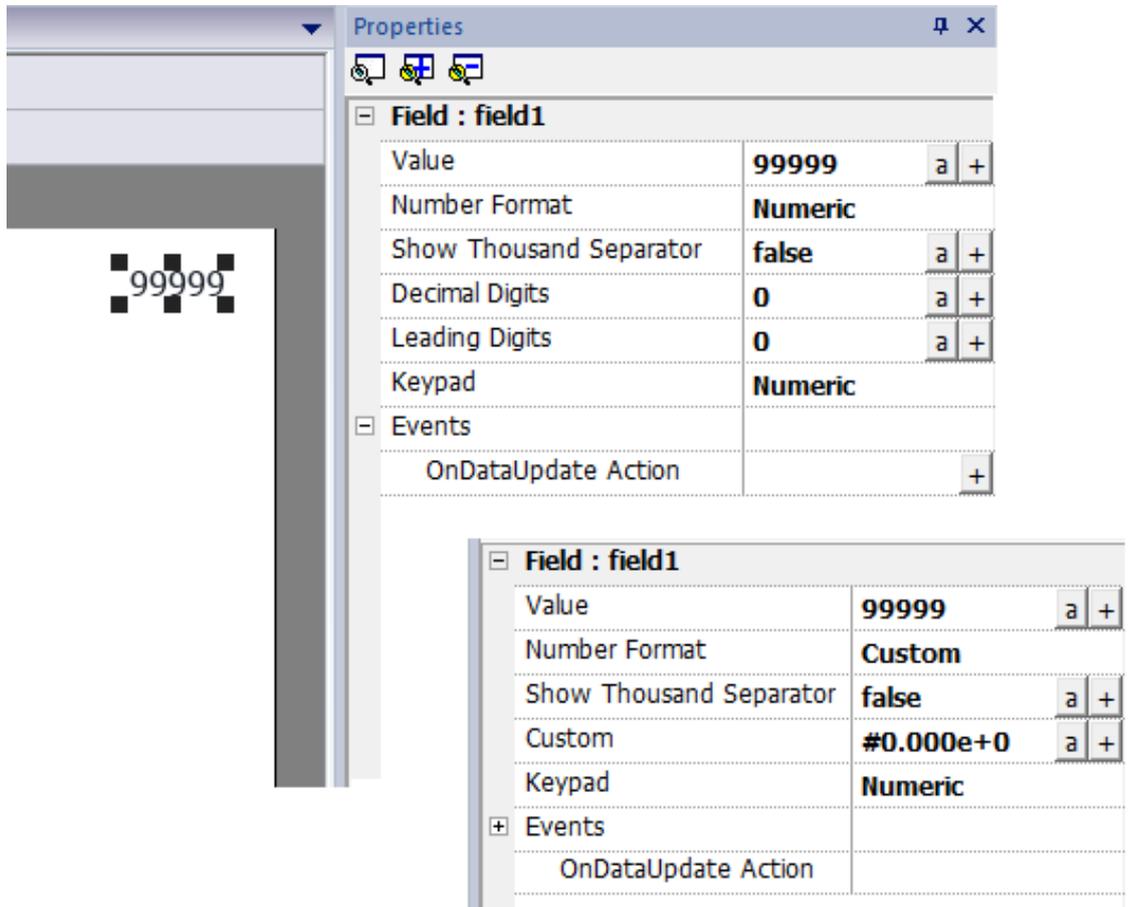
数据字段控件

最常见的控件是数据字段控件，它提供了显示标签值的可能性。(请参阅 ["标签编辑器"](#) 在本页 99)。

路径: [视图](#) > [工具栏和连接窗口](#) > [控件库](#)



字段属性



属性	说明
值	包含要显示信息的标签
数字格式	<p>显示格式</p> <ul style="list-style-type: none"> • 无 无限制(系统决定要使用的格式) • 数字 数字格式。小数位和前导位可用于更好地定义数字格式 • 十六进制 十六进制格式。前导位可用于更好地定义数字格式 • 十六进制 相同的“十六进制”格式，但使用大写字母 • 科学 科学格式。小数位可用于更好地定义数字格式 • 科学 相同的“科学”格式，但使用大写字母 • 自定义 使用额外的“自定义”参数更好地定义要使用的格式(参见下表)

属性	说明
显示千位分隔符	显示/隐藏千位分隔符
小数位	要显示的小数位数(在所有格式类型中都不可用)
前导位	要显示的前导位数(在所有格式类型中都不可用)
自定义	定义要使用的数字格式的字符串(仅当所选数字格式为“自定义”时可用)
键盘	要弹出的键盘类型以编辑选项卡值 <ul style="list-style-type: none"> • 无 • 字母、数字等 弹出预定义键盘或用户键盘(请参阅“键盘”在本页 347) • 滚轮 键盘将不显示。滚轮可用于递增/递减数值 • 程序安排 键盘将不显示。键盘宏可用于输入键(请参阅“键盘动作”在本页 173)
事件	
OnDataUpdate 操作	随时执行标签值更改的命令列表(请参阅“动作”在本页 167以了解可用命令)

 可以从全局项目属性中修改用作千位分隔符(点)的字符和用作十进制分隔符(逗号)的字符。请参阅“区域设置”在本页 85

一些属性具有几个按钮:

-  进入编辑模式:
您可以直接输入要使用的标签名称
-  附加至标签:
将打开从中选择要使用标签的对话框

自定义格式

在自定义属性中,允许的字符为“#”“.”“0”“h”“H”“e”“E”

使用占位符字符控制在小数位之前和之后显示的数字。如果要在数字中仅显示有效位数,则使用数字符号(#)。此符号不允许显示非有效零。如果要在数字位数比格式代码中指定的位数少时显示非有效零,则使用数字字符零(0)。

如果在小数点左侧的数字位数比格式代码中的占位符多,则会显示额外的数字。不过,如果在小数点右侧的数字位数比格式代码中的占位符多,则该数字被四舍五入到与占位符相同的小数位数。

例子

显示	为	占位符
123	0123	000#
1500	5DC	#H

显示	为	占位符
1500	5dc	#h
1500	05DC	000#H
123.456	123.46	#.##
123.456	000123.456000	00000#.000000
12,200,000	1.22E+07	#0.00E+00
12,200,000	12.2E+6	#0.0E+0

信息控件

信息控件可以显示标签值已编索的文本信息。

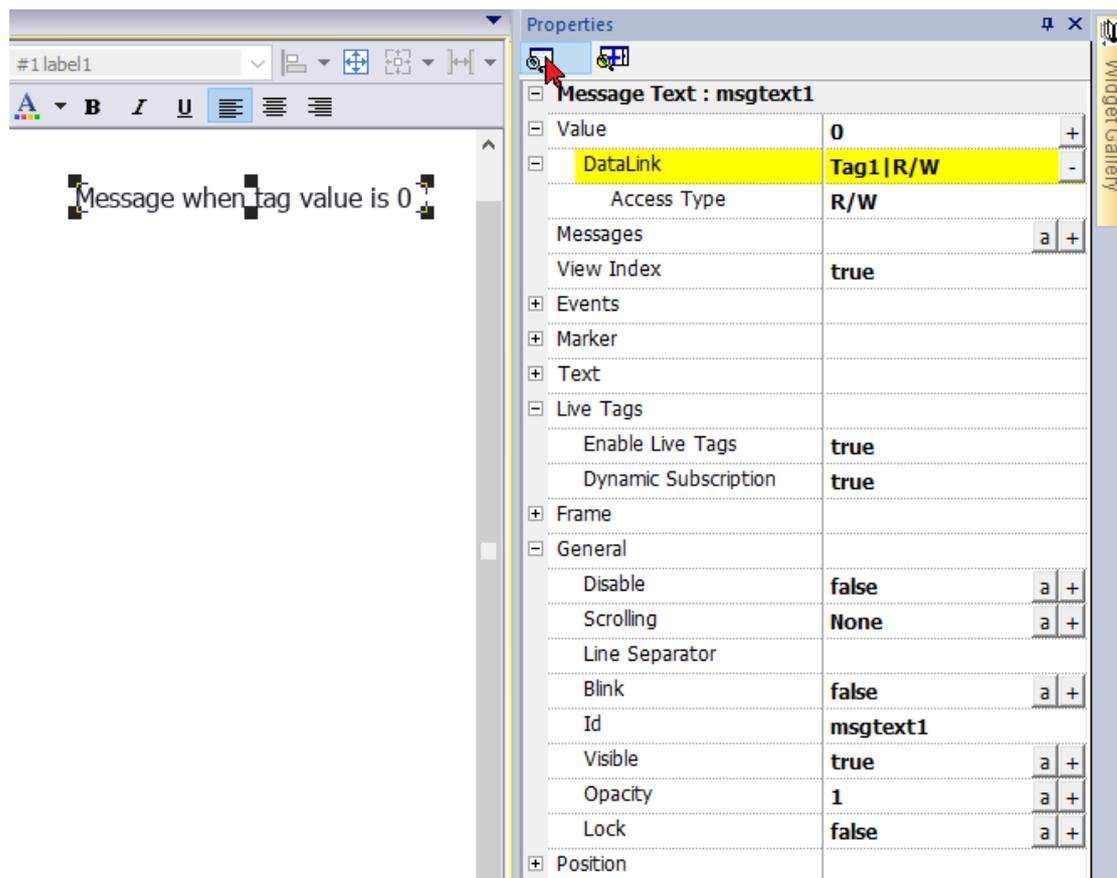
路径: **视图 > 工具栏和连接窗口 > 控件库**



信息属性

将控件拖放到页面内，然后选择控件打开控件的属性对话框。

请注意，只有选择“显示高级属性”按钮时，某些属性才可见。



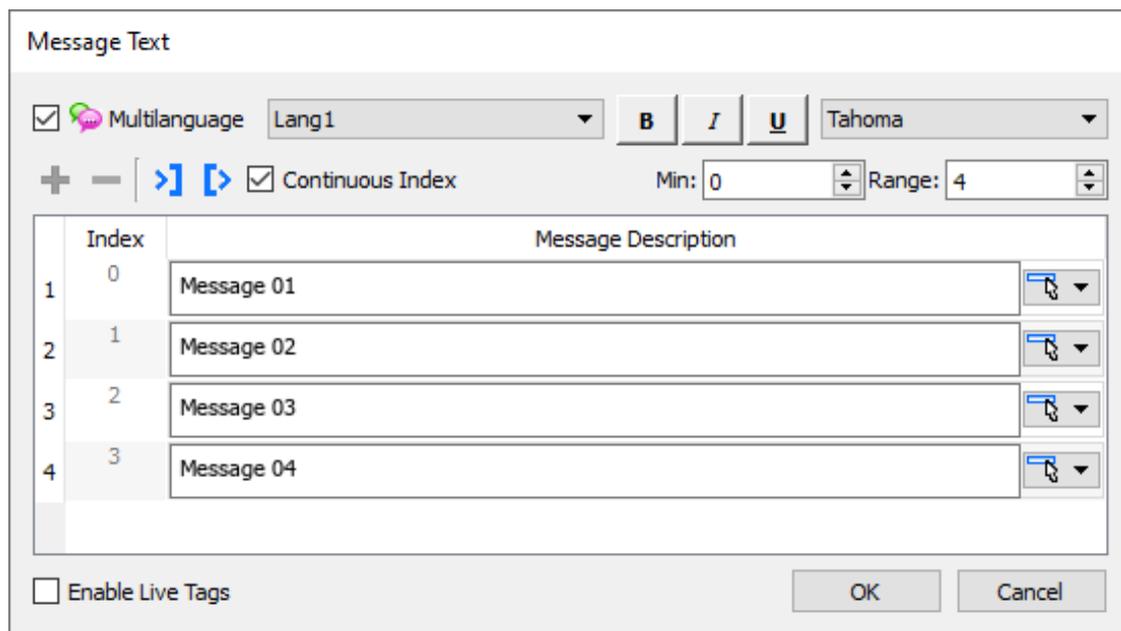
属性	说明
值	<p>标签名称, 用于动态选择要显示的消息。请参阅 "将控件附在标签上" 在本页 34</p> <p>当“访问类型”为 R/W 时, 可以通过单击消息来更改附加标签的值。消息因此被更新为与新索引值对齐。</p>
消息	要显示的消息列表。单击 + 打开一个对话框, 您可以在其中输入消息和相关的索引
视图索引	如果为 true, 则当控件是可编辑的(访问类型 = R/W)时, 选择对话框还将显示与每个消息关联的索引号。
事件	控件内容更改时执行的动作。请参阅 "事件" 在本页 48
标记	启用控件周围的标记(仅在 LRH SW 内可见)
文本	文本属性(字体、颜色、尺寸等)
在线标签	<p>启用文本消息内部的标签值。请参阅 "信息控件" 上一页</p> <ul style="list-style-type: none"> - 启用在线标签 启用在线标签占位符 - 动态订阅 如果为 true, 仅从通信协议中检索可见的标签。如果为 false, 即使不可见, 所有标签也会保持不断更新。

属性	说明
边框	用于启用和配置控件框架和/或背景颜色的参数
常规	<p>常规属性</p> <ul style="list-style-type: none"> - 禁用 禁用小工具上的用户输入 <li style="margin-left: 20px;"> 在此属性上附加标签可以启用/禁用在运行时修改值的可能性 - 滚动 用于启用和配置文本消息滚动的参数 <li style="margin-left: 20px;">启用后, 所有消息将链接在一起并以滚动模式显示。 - 行分隔符 在滚动模式下显示时, 在消息之间插入的字符 - 闪烁 - 文本将闪烁 - Id 控件标识符。您可以保留默认值或将其重命名为更合适的名称 - 可见 如果为 false, 控件不可见 - 不透明度 不透明度级别描述了透明度级别, 其中 1 完全不透明, 0.5 表示透明度为 50%, 0 为完全透明。 - 锁定 如果为 True, 则无法选择控件并从 LRH SW 页面编辑器中将其移动
位置	控件在显示屏上的位置。请参阅 "控件在显示屏上的位置" 在本页 25

一些属性具有几个按钮:

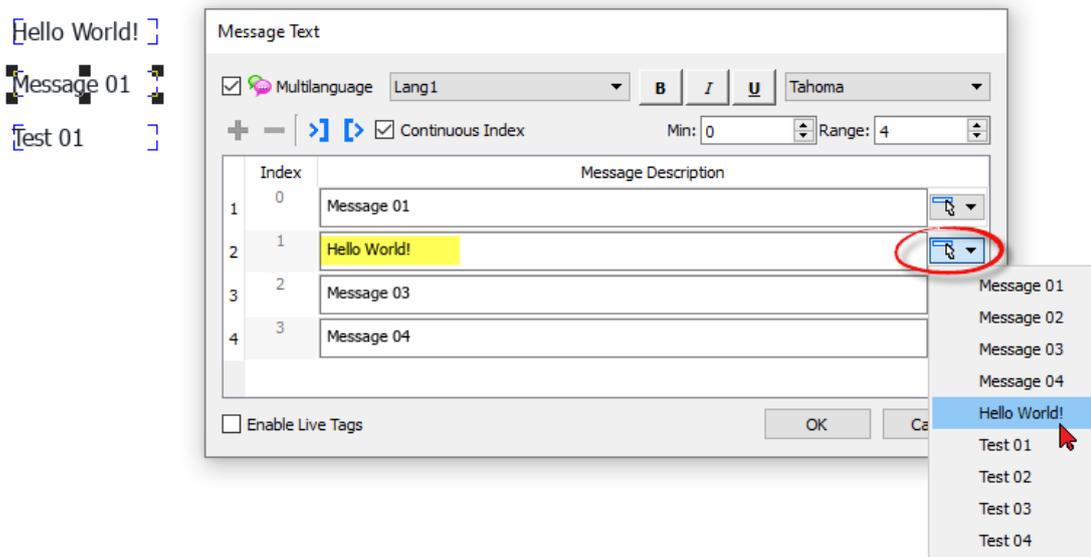
-  进入编辑模式:
您可以直接输入要使用的标签名称
-  附加至标签:
将打开从中选择要使用标签的对话框

双击标签控件将打开编辑对话框, 您可以在其中输入要显示的文本并设置主要文本属性。



选择文本按钮

“选择文本”按钮使您可以复制其他控件中已使用的文本



导入 / 导出

导入/导出按钮使您可以在 .xml 文件中导入或导出整个消息列表，可以使用外部工具进行编辑/修改。

 在可编辑的 .xml 文件中导出消息

 从 .xml 文件导入消息

 如果需要在不同的地方使用相同的消息控件，为了节省维护时间，您可以创建并复制自定义控件。



将控件附在标签上

要控制控件并通过实时数据驱动控件，可以对不同的数据源绑定特定的属性。例如，可以将仪表**值**属性绑定至探头温度标签，或者将**显示**属性绑定至配方数据

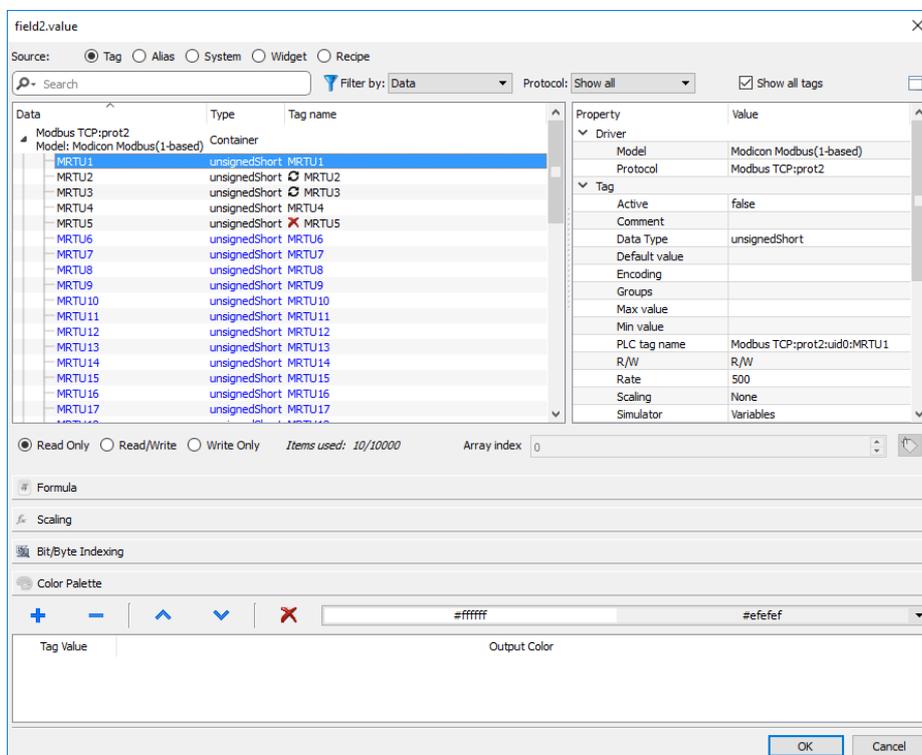
数据源

对象属性可以附属的元素：

数据源	说明
标签	标签编辑器中定义的标签
别名	编入索引的标签地址
系统	预定义的系统标签(查看" 系统变量(附加至) "在本页 125)
控件	连接到控件属性(如, 浮动块控件)
配方	配方管理上的数据(查看" 配方 "在本页 247)

在一个标签上附一个属性

1. 在**属性**窗格点击在 +。
2. 在**源**中选择数据源，在列表选择一个协议和标签。使用**搜索**框过滤标签。



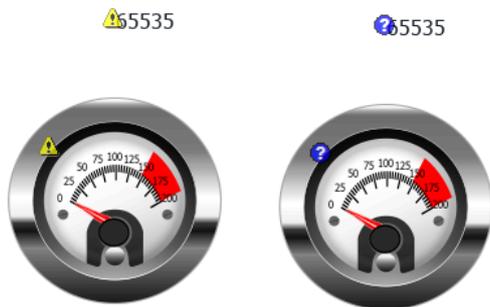
3. 设置存取类型 (如只读)。数组索引字段在所选的标签是一个数组时出现, 以辨别要使用的数组元素。通过一个额外的标签, 支持间接索引模式。
4. 点击 **OK** 确认。

当一个定义与词典中的标签定义不匹配时, 或者当它缺失时, 标签名称旁边的图标将突出显示。如果选中**显示所有标签**, 则即使未在应用程序中导入, 也会显示所有词典标签。双击将从词典中导入标签。

请参阅“**附属于**”参数”在本页 39 了解详细信息。

通信错误

靠近附着标签的控件可能出现两个图标。



- : 通信错误
- : 数据暂不可用(慢通信协议)

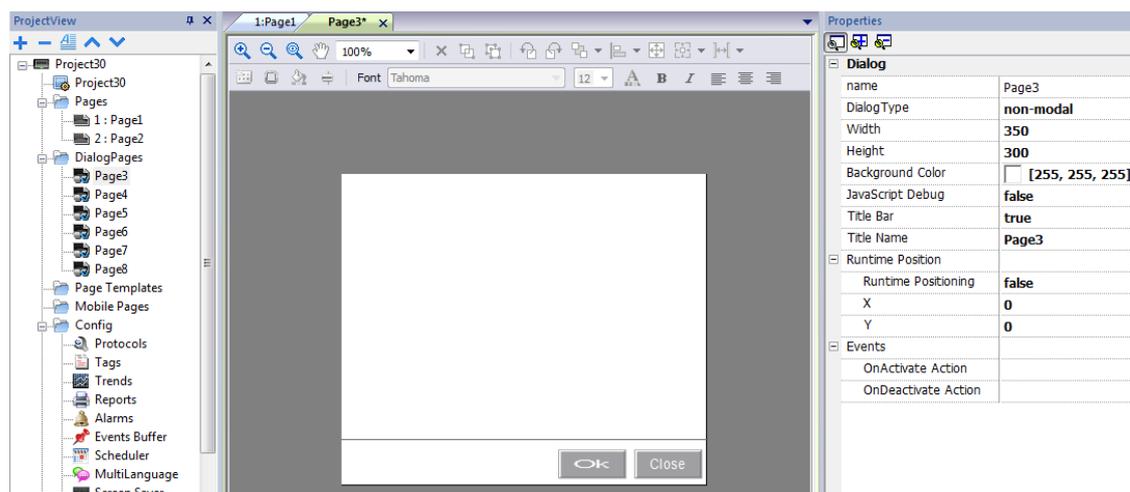
对话框页

路径: **项目视图 > 对话框**

按项目要求对话框页面在运行时在当前页顶端打开。这些被用来通知报警、错误或要求用户行动。

主要对话框属性

属性	说明
对话框类型	<p>modal = 用户不能在对话框关闭前返回主项目窗口/页面。</p> <p>non-modal = 在对话框显示在窗口顶端时, 用户可以继续使用主项目窗口(或其他非程序对话框)。</p>
标题栏	<p>true = 显示对话框标题</p> <p>false = 不显示对话框标题</p>
标题名称	对话框标题。仅当 标题栏 = true 时。
runtime 位置	<p>对话框固定位置</p> <p>false = 对话框将置于屏幕中间</p> <p>true = 对话框的左上角将置于 X 和 Y 位置</p>



对话框最大数

打开的对话框的最大数量在"功能规格说明与兼容性"在本页523中定义。

中。当打开的对话框达到最大数量时，最先打开的对话框被关闭以打开一个新的对话框。

4 编制程序概念

LRH SW 编程基于一些基本概念和行为。

数据类型	38
“附属”参数	39
公式	44
事件	48
控件定位	51
管理重叠控件	52
编组控件	53
更改多个控件属性	60
根据标签值更改填充颜色属性	61

数据类型

创建标签时，您必须指定其属性。数据类型特定于 LRH SW，而内存类型特定于所选协议。根据所选控制器地址所需的内部表示选择值。



注：数组类型使用相同的数据类型，后跟“[]”(例如：`boolean []`)

数据类型	内存空间	限制
boolean	1 位数据	0 ... 1
byte	8 位数据	-128 ... 127
short	16 位数据	-32768 ... 32767
int	32 位数据	-2.1e9 ... 2.1e9
int64	64 位数据	-9.2e18 ... 9.2e18
unsignedByte	8 位数据	0 ... 255
unsignedShort	16 位数据	0 ... 65535
unsignedInt	32 位数据	0 ... 4.2e9
uint64	64 位数据	0 ... 1.8e19
float	IEEE 单精度 32 位浮点类型	1.17e-38 ... 3.4e38
double	IEEE 双精度 64 位浮点类型	2.2e-308 ... 1.79e308
string	元素数组，包含通过所选编码定义的字符代码	

系统时间

HMI 设备内的系统时间格式为 Unix 时间(也称为新纪元时间)。Unix 时间是指从 Epoch(1970 年 1 月 1 日 00:00:00 UTC)开始所经过的秒数。

示例：

标签值	系统时间	ISO 8601
0	01/01/1970 – 01:00:00	1970-01-01T00:00:00+00:00
1	01/01/1970 – 1:00:01	1970-01-01T00:01:00+00:00
60	01/01/1970 – 1:01:00	1970-01-01T00:01:00+00:00

“附属于”参数

对象属性

在 LRH SW 中，页面对象的属性可以在编程时设置或者配置为动态。如需在编辑程序时改变属性，可使用页面工具栏或属性窗格。先选择对象查看显示的属性。

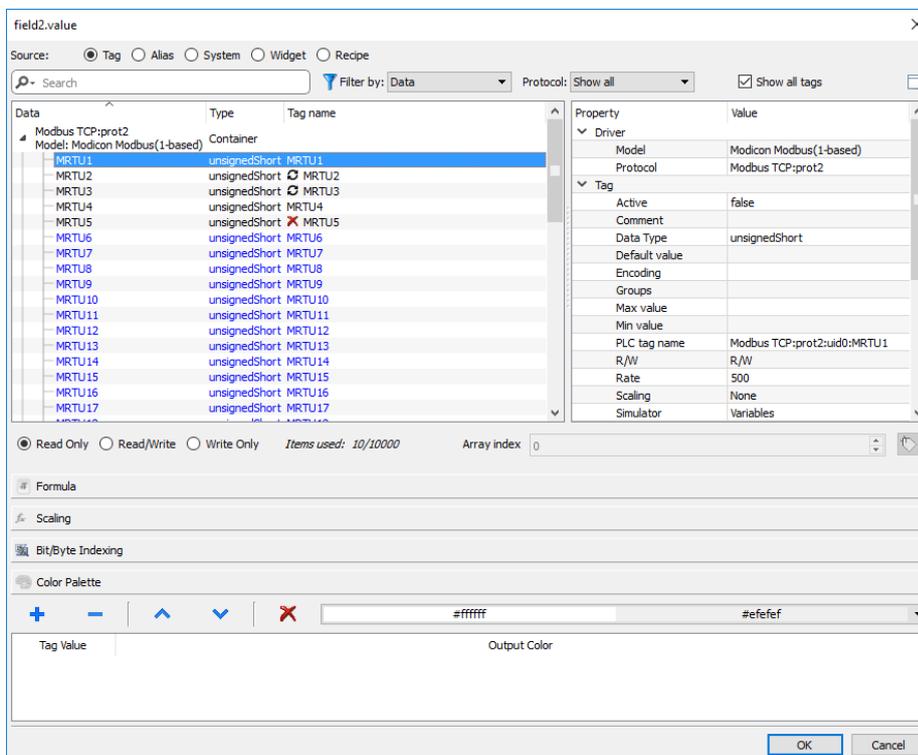


页面工具栏只显示了最常用的对象的属性，而属性窗格以基本或高级的角度显示了所有属性。

要动态地改变属性值，您可以将其附在标签或变量上。

在一个标签上附一个属性

1. 在属性窗格点击在 +。
2. 在源中选择数据源，在列表选择一个协议和标签。使用搜索框过滤标签。



3. 设置存取类型 (如只读)。数组索引字段在所选的标签是一个数组时出现,以辨别要使用的数组元素。通过一个额外的标签,支持间接索引模式。
4. 点击**OK**确认。

当一个定义与词典中的标签定义不匹配时,或者当它缺失时,标签名称旁边的图标将突出显示。如果选中**显示所有标签**,则即使未在应用程序中导入,也会显示所有词典标签。双击将从词典中导入标签。

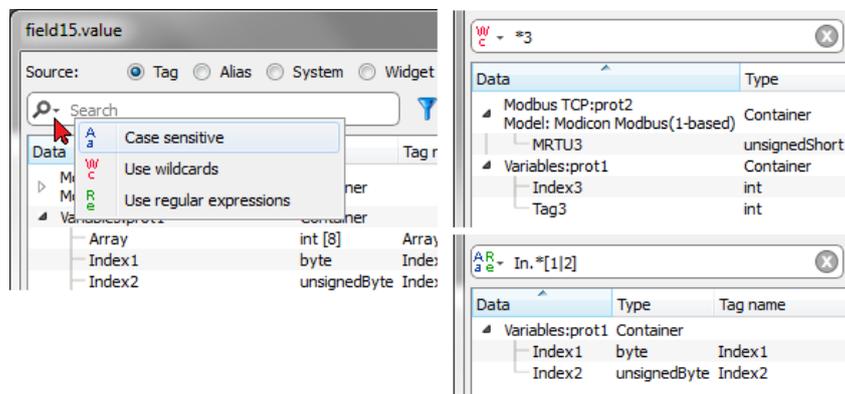
数据源

对象属性可以附属的元素:

数据源	说明
标签	标签编辑器中定义的标签
别名	编入索引的标签地址
系统	预定义的系统标签 (查看"系统变量(附加至)"在本页 125)
控件	连接到控件属性(如,浮动块控件)
配方	配方管理上的数据(查看"配方"在本页 247)

高级搜索

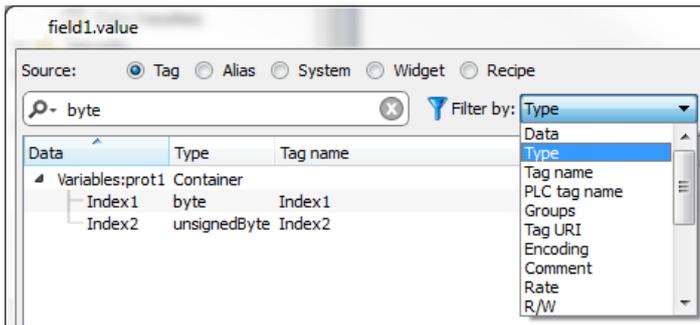
多重句法选项可以应用于搜索框:



主选项	功能
通配符	使用简单的通配符匹配来搜索。字符'?' :匹配单个任意字符。字符"*":匹配零个或多个任意字符。“[...]”:可以在方括号中被表示的字符组。
常规表达	描述字符模式。 有关正则表达式的其他详细信息,请参阅 https://en.wikipedia.org/wiki/Regular_expression 。

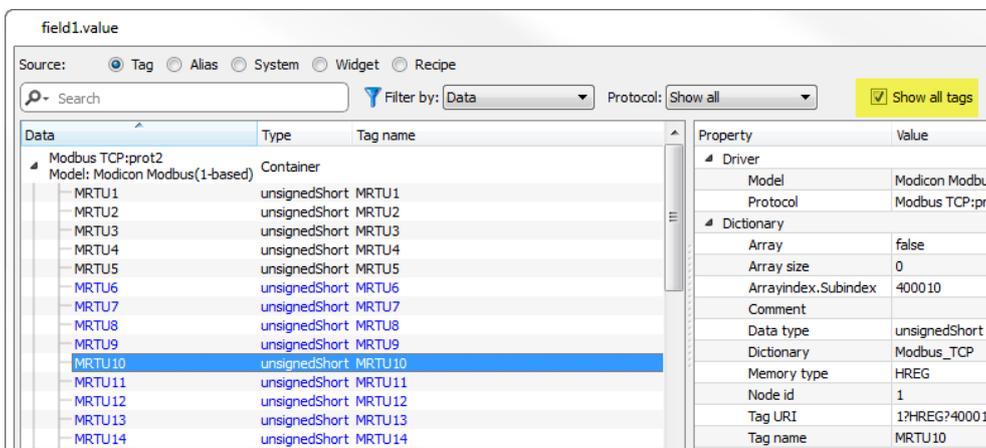
筛选标签

选择各种标签筛选标准:

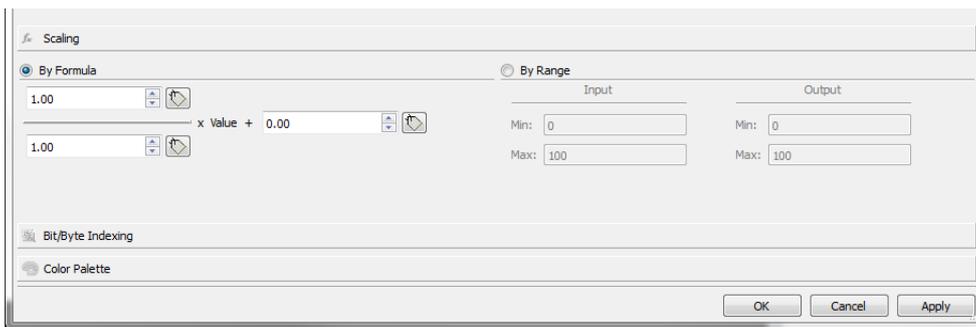


显示词典标签

选中**显示所有标签**时，属于一个词典但尚未导入的标签以蓝色显示。您可以选择并双击一个标签将其输入到项目中。



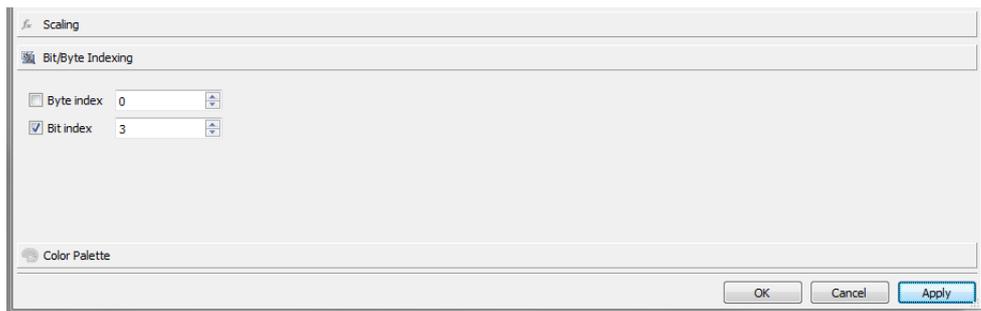
转换标签值



缩放选项卡可以转换标签值。在**按范围**部分中设置输入和输出范围：系统会自动计算缩放系数。

根据索引提取标签位/字节

允许根据指定的位或字节数从一个词提取单一位或字节内容

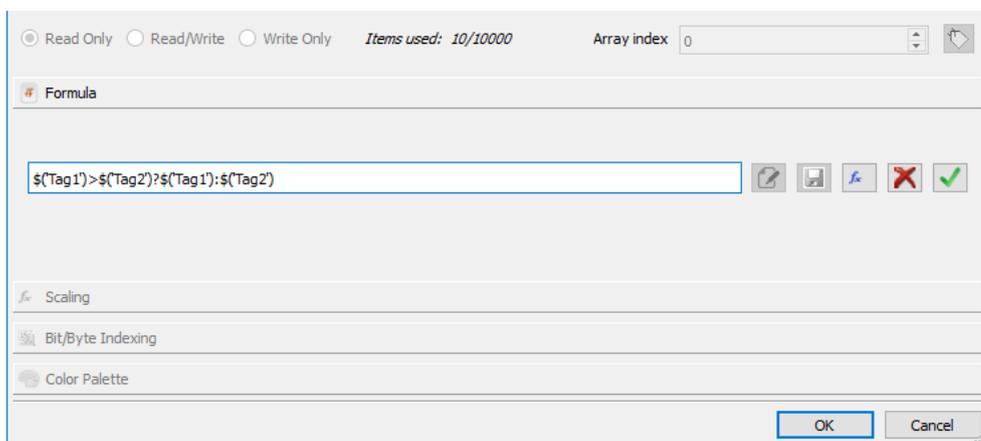


使用公式计算要使用的值

允许使用公式计算要使用的值。有关更多详细信息，请参阅“公式”在本页44。

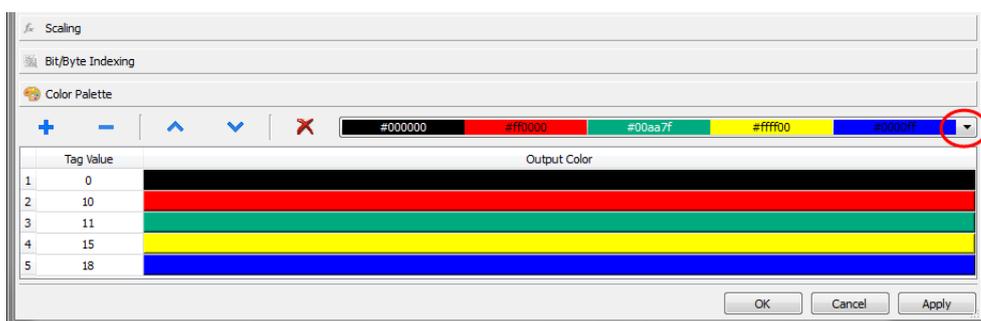


请注意使用数据链接将为只读的公式



用颜色绘制标签值

允许用颜色绘制数值或字符串标签值。例如，可以使用这一选项改变一个按钮的颜色。



部分	功能
	从工具栏添加/移除或将颜色列向上/下移动。标签值是可编辑的，您可以修改顺序值。
	最后设定的颜色组合自动保存。可以从颜色工具栏恢复。

标签值可以用逗号分隔的值范围，示例：

- 5、10-15、20
- A、AB、C



应用程序开发人员应正确定义所有项以覆盖所有可能的应用程序值；未定义已定义调色板中的值时，我们可能已获取意外颜色。

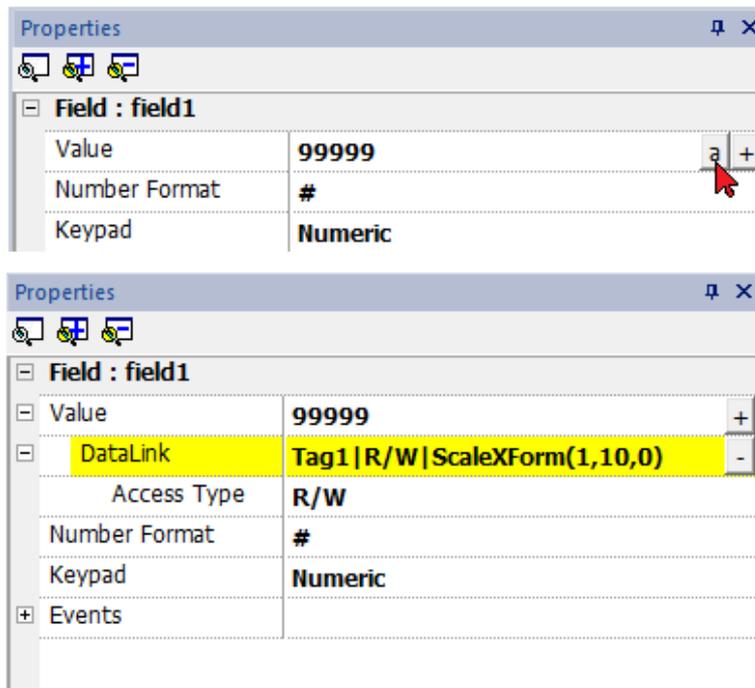


请注意，用颜色绘制的标签值将返回字符串数据类型（例如“#FF0000”）

数据链接序列化

可以使用上面的“附加到...”对话框手动输入或修改数据链接。

单击属性窗格中的某一按钮，然后输入用于描述数据链接的文本



数据链接格式如下：

```
Tagname [index] | [Attribute] | [XForm] | [XForm] | ...
```

或采用以下格式：

```
= <格式>
```

示例：

- arrayTag[2]
- Tag[0][index]
- Alarm triggered: _SysPropMgr
- Tag|R/W|ScaleXForm(1,10,0)
- Tag|R/W|ScaleXForm(1,10,0)|ByteIndexXForm(1)|ColorPaletteCustomXForm(0#00aa7f,1#ff0000)
- =\$('Tag1')>\$('Tag2')?\$('Tag1'):\$('Tag2')
- =\$Contains(\$('Tag4'),\$('Tag3'))
- =\$Pow(2,\$('Tag2'))

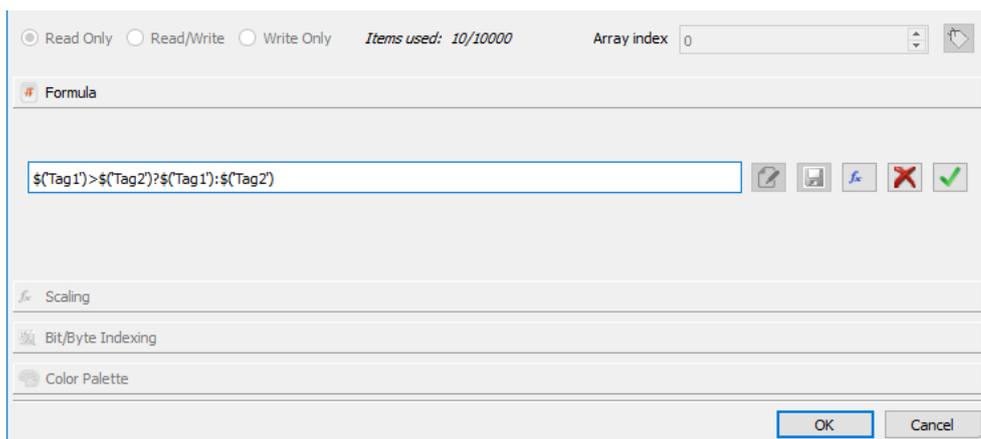
公式

公式由以下表达式组成：

- 运算符：可以是基本的数学运算、逻辑运算符、比较运算符或基本字符串运算符。
- 操作数：可以是文字(用作常量的数字和字符串)和对标签的引用。

支持圆括号作为优先级运算符。运算符 \$ 将用于调用函数，尤其用于引用标签(参见以下示例)。

附加到对话框允许使用公式计算要返回的值。



命令

-  进入编辑模式
-  将输入的公式保存在公式库中，以便在项目的其他地方重复使用相同的公式。
-  打开公式库以选择已定义的公式。
-  移除输入的公式
-  确认输入的公式

在编辑模式下，只需编辑公式并双击库中的标签或函数即可将其添加到公式中。

公式示例如下：

- \$('Tag1')+\$('Tag2')
- \$('Tag1')&\$('Tag2')
- \$('Tag1')>\$('Tag2')?\$('Tag1'):\$('Tag2')
- \$Pow(2,\$('Tag2'))
- \$Contains(\$('Tag1'),\$('Tag2'))

公式语法

基本运算	
“文本”	字符串文字
数字	数字文字, 例如 169857 或 13.547
String(...)	转换为字符串(注:没有 \$)
Number(...)	转换为数字(注:没有 \$)
\$FuncName(param1, param2, ...)	通用函数调用。(默认函数和用户函数)
\$('TagName')	标签或控件属性或配方等  请注意, 标签名称必须是字符串文字
\$('TagName')[index]	数组标签元素。  请注意, 标签名称必须是字符串文字
exp1 ? exp2 : exp3	三元表达式。如果 exp1 为 true, 则采用 exp2, 否则采用 exp3。这与使用 if/then/else 语句一样

数学运算符	
+	加法
-	减法
*	乘法
/	除法
%	模块

位运算符	
&	如果两位都为 1, 则将每个位设置为 1
	如果两位中的一位为 1, 则将每个位设置为 1

位运算符	
~	对所有位取反
^	如果仅两位中的一位为 1, 则将每个位设置为 1
<<	从右推入零并让最左位掉下, 向左移动
>>	将最左位拷贝从左推入并让最右位掉下, 向右移动
>>>	从左推入零并让最右位掉下, 向右移动

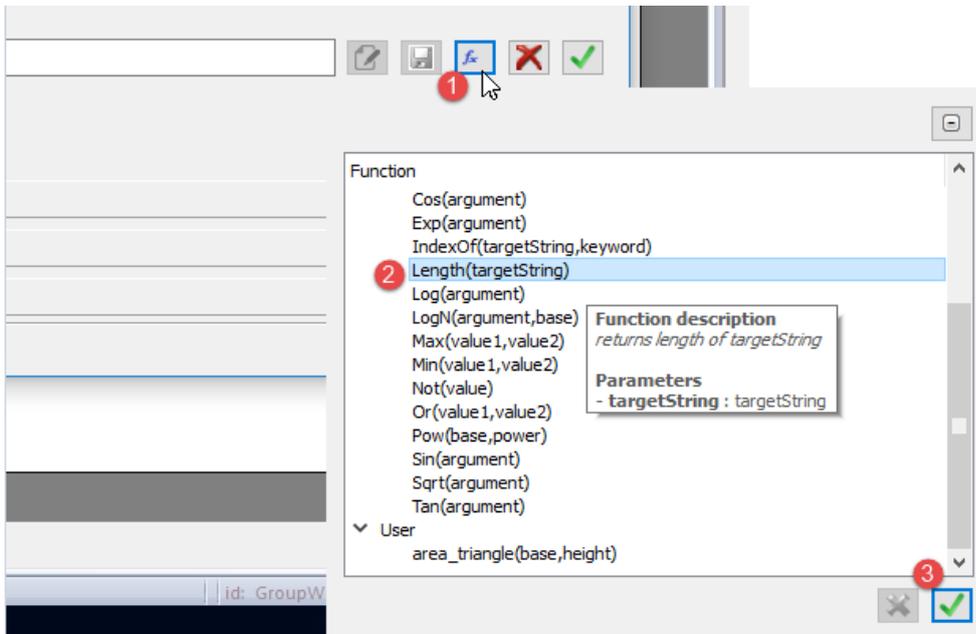
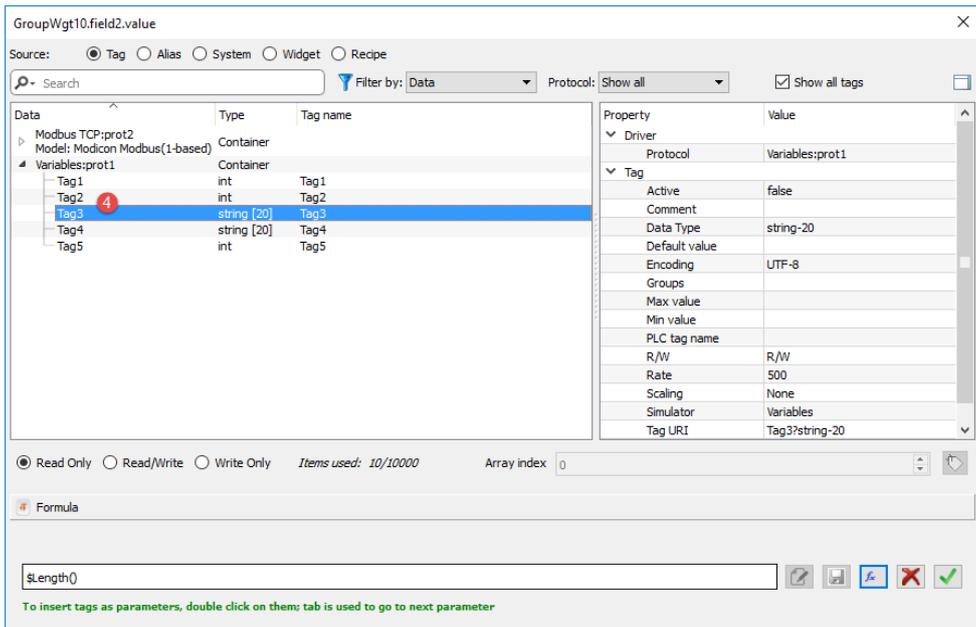
逻辑运算符	
&&	和
	或
!	非

比较运算符	
<	小于
<=	小于或等于
>	大于
>=	大于或等于
==	等于
!=	不等于

使用库中的预定义公式

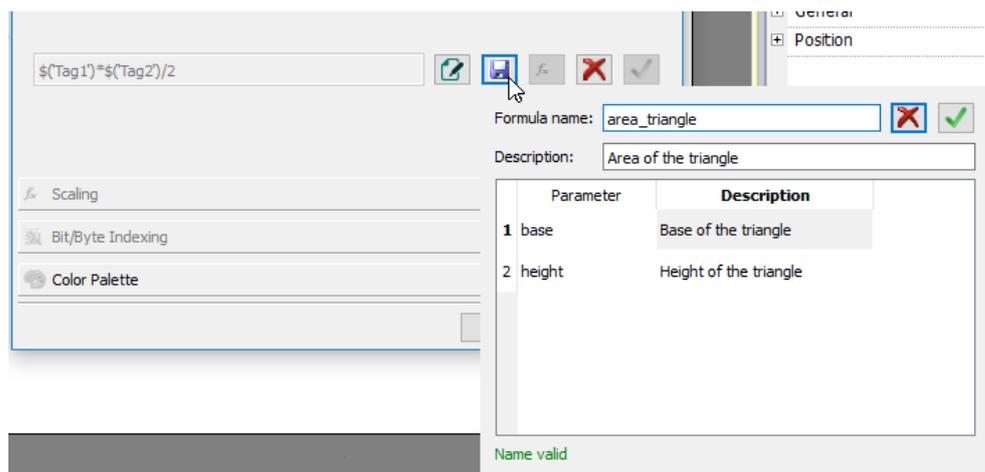
要插入项目或预定义公式, 请执行以下操作:

1. 打开公式库
2. 选择要使用的公式
3. 确认所选公式
4. 输入所选公式中所需的参数

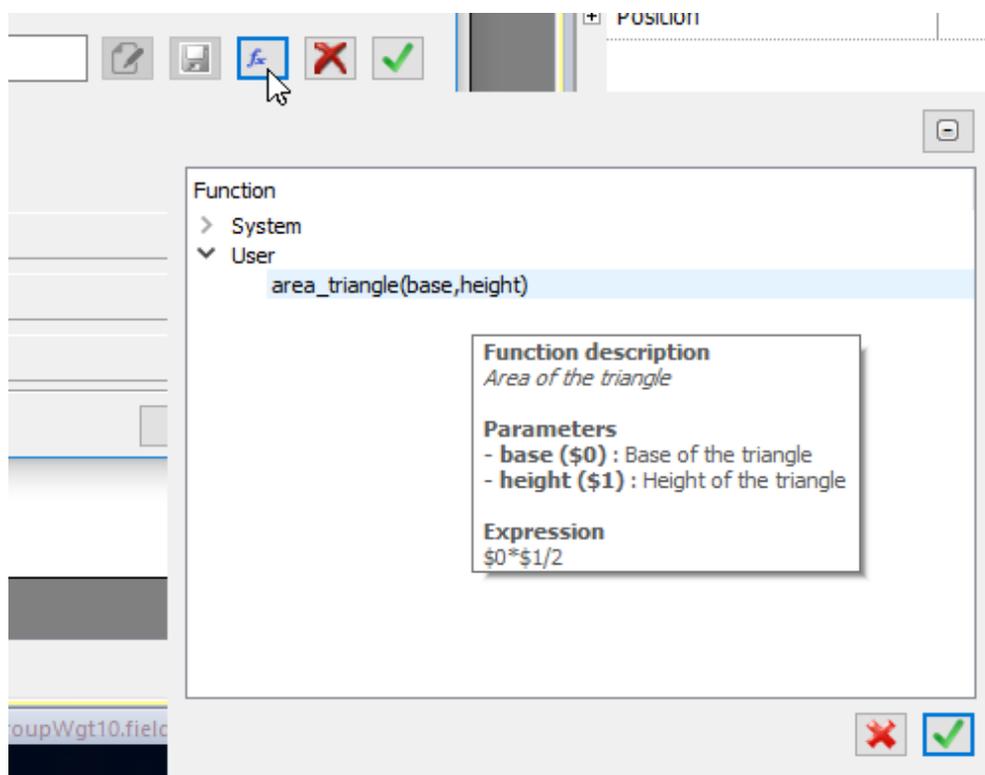


将用户公式添加到库

在输入新公式后，使用“保存”按钮可将新公式存储在项目文件夹中，以从公式库中可用。



如检索其他预定义公式一样，可从公式库中检索用户公式。



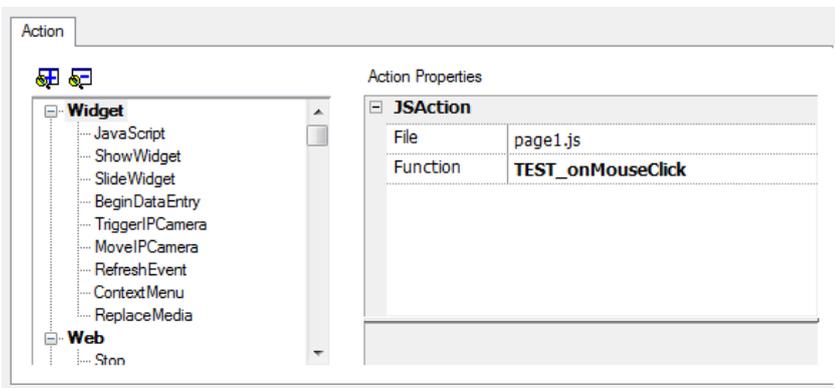
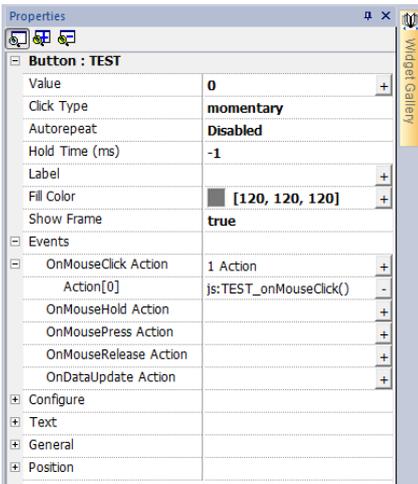
事件

事件用于在项目标准下启动动作，能够与以下相关联：

- 按键/触摸(点击、按压、释放)
- 鼠标滚轮
- 外部输入设备如键盘和鼠标(点击、按、保持、释放、滚动)
- 数据更改(数据更新)
- 页面切换(激活、失效)
- 报警
- 计划任务

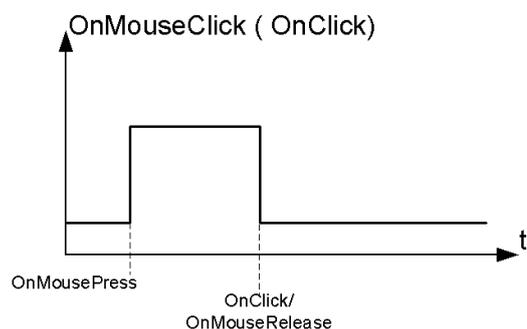
您可以向一个事件附加一个或多个动作，这样无论什么时候事件发生时，它们都可以被执行。

该例子显示了通过按一个按键激活一个 JavaScript 动作。



点击 / 鼠标点击

当按钮 / 键被快速按下并放开时会启动事件。



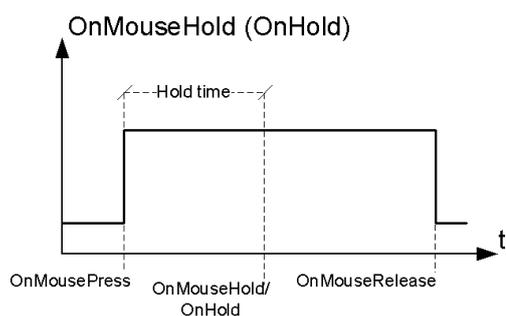
保持 / 鼠标保持

当按下按钮/键且保持按住的时间达到控件属性中设置的**保持时间**时触发事件。这一事件的动作程序只有在保持时间过期之后才会执行。

默认**保持时间**在项目属性中配置，但每一个按钮/键都可以重新定义。请参阅“[项目属性](#)”在本页 73。

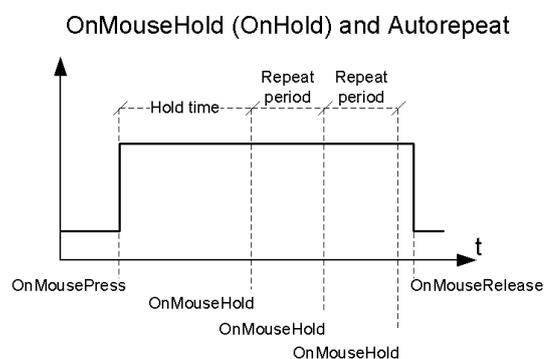


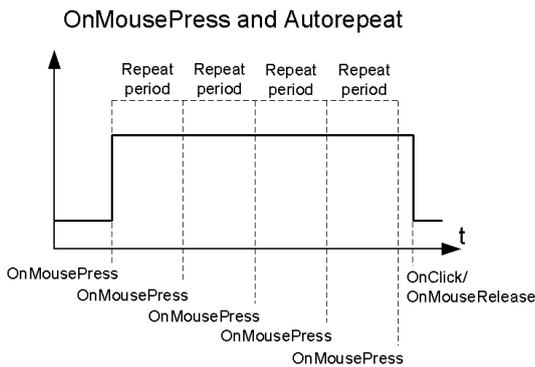
注：如果控件的**保持时间**设为 -1，则将使用项目的**保持时间**值。



自动重复

对按钮或键的按压或保持启动自动重复。**自动重复时间**在项目属性中指定，但每一个按钮/键都可以重新定义





滚轮

当滚轮 (例如 USB 鼠标滚轮) 值更改时启动事件。滚轮通常用于增加 / 减少文本框中或附在标签上的值。

激活

当页面加载时启动事件。事件在页面中的控件被初始化前启动。

OnDataUpdate

当标签值更改时启动事件。更新时间取决于协议完成更新过程需要的时间。例如，**数据更新时**事件可以触发或不触发，具体取决于在控件首次初始化之后或之前数据在协议中是否可用。尤其是页面更改通知更可能出现在慢协议和远程客户上。



注：执行**激活**期间读取的值可以与从后续**数据更新时**事件中获取的相同，因为**数据更新时**通知是非同步发送的。

控件定位

您可以使用两种方法在页面定位控件：

- 对齐网格
- 对齐对象

要显示网格，请在**视图**菜单上，单击**显示网格**。

对齐网格

路径：**视图** > **对齐网格**

当您移动或给一个目标改变大小时，即使网格不可见，其左上角将与网络中最近的交叉点对齐。

设置网格属性

路径：**视图** > **属性**

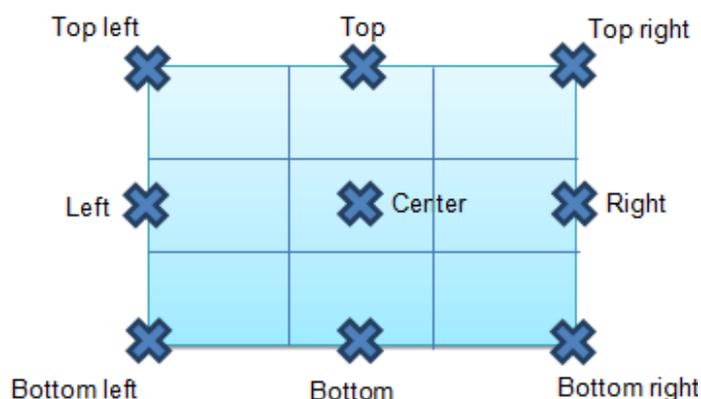
参数	说明
间距 X	在 X 轴上两条线/个点的像素间隔
间距 Y	在 Y 轴上两条线/个点的像素间隔
类型	网格类型(点或线)
颜色	网格颜色

对齐对象

路径: 视图 > 对齐对象

当您移动一个物体, 该功能帮助与页面上的其他对象对齐。

当您选择一个物体, 根据您按的区域: 上、左上、右、下、左下、右下、左、右、中, 以下热点之一会被选择作为移动点的来源。

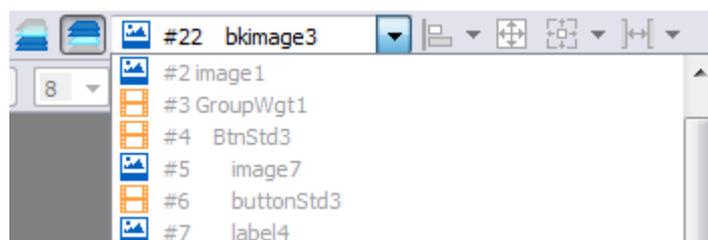


算法找到附近控件热点中匹配的热点, 与源移动点的x或y轴匹配。对于线控件, 源移动点是该线的终点。

管理重叠控件

当页面上一个或多个控件重叠时, 您可以管理其顺序, 这样一个会在另一个上面显示。

该页面控件的顺序显示在组合框中。具有更大的z序号的控件在低的z顺序的控件单元前面。图片图标识别静态物体, 影响框架图标识别动态物体。



重要: 正确的控件顺序对于运行系统性能至关重要, 因为重叠的动态控件可能使静态优化无效, 降低 HMI 应用程序的性能。

隐藏/显示 z 顺序控件

要在所选的控件上隐藏控件：

- 在工具栏上，单击  并选择一个控件：其上的所有控件均被隐藏

要在所选的控件下隐藏控件：

- 在工具栏上，单击  并选择一个控件：其下的所有控件均被隐藏

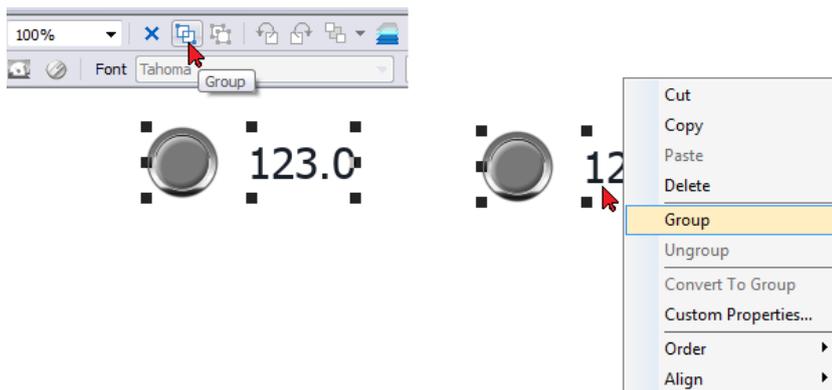
工具栏可以：

- 隐藏堆积在所选控件上和/或下的控件
- 使用以 z 顺序列举所有控件的组合框，对不同控件操作。

编组控件

要编组控件：

- 选择所有要编组的控件。
- 右击然后单击 **编组**。

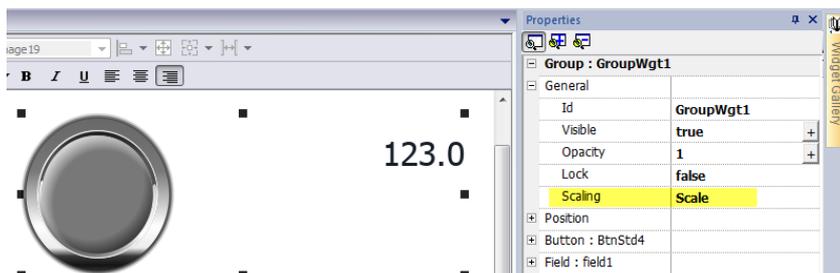


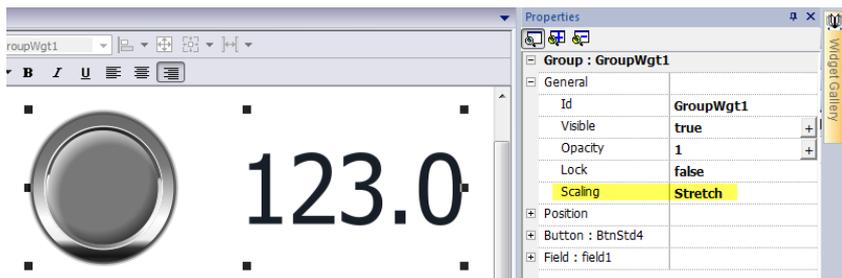
提示：双击以进入组编辑模式。在模式中，仅可编辑和选择组控件。所有其他控件为部分隐藏。

改变编组的控件的大小

您可以定义改变大小时对象如何反应。使用 **通用** 部分中的 **缩放** 属性：

- 缩放**：对象和文本不按比例改变大小
- 拉长**：对象和文本按比例改变大小





网格布局组

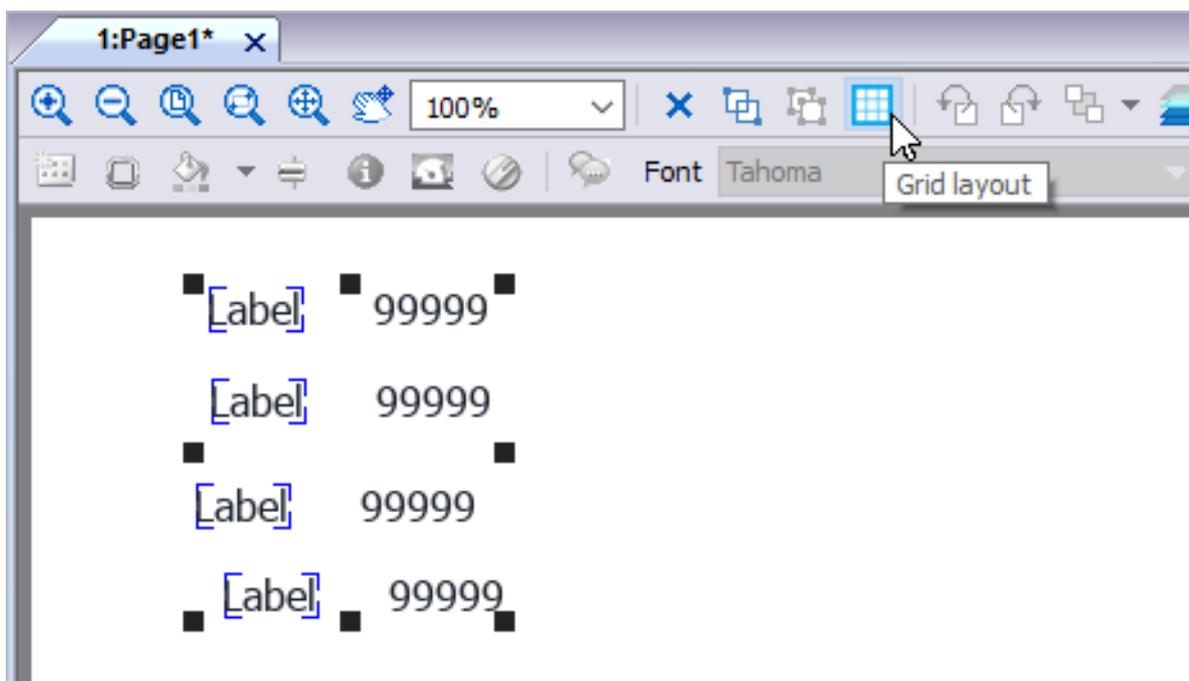
网格布局可配置组中控件的空间关系。

创建网格布局：

- 启用控件组的“网格布局”参数。

或者

- 选择将位于组内的控件，然后单击页面工具栏上的“网格布局”按钮。在启用“网格布局”属性时，系统将收集并对齐组内所选控件。



以下元素与网格布局相关联，且可进行配置：

- 网格属性
- 行、列属性
- 单元格属性

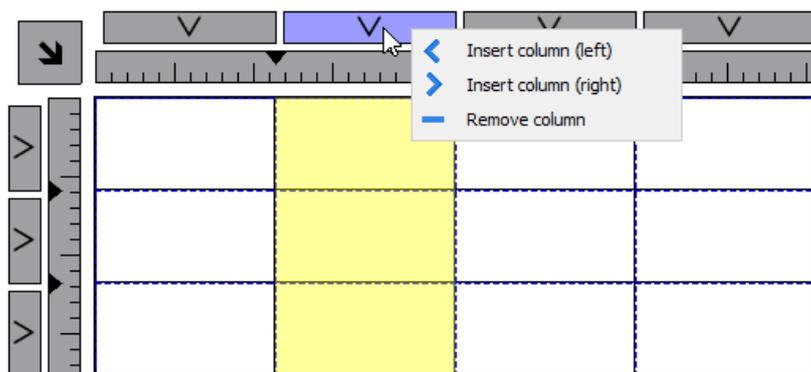
网格属性

参数	说明
启用	启用网格布局。 将在组中控件周围生成网格
行数 栏数	网格的行数和列数。  仅当单元格为空时，才能删除行和列。
水平上溢 垂直上溢	此参数定义网格太小不足以包含所有行和列时网格的行为。 <ul style="list-style-type: none"> • 隐藏 不显示与网格不符的行和列 • 可见 生成的网格不能小于包含所有已定义行和列所需的最小尺寸 • 滚动 网格太小不足以容纳所有已定义行和列时，可使用滚动条移动网格内容。
水平下溢	此参数定义网格尺寸超过为行和列定义的尺寸时网格的行

参数	说明
垂直下溢	为 <ul style="list-style-type: none"> • 阻止 生成的网格不能超过行和列的最大尺寸 • 左、中、右 - 上、中、下 定义单元格尺寸超过最大定义尺寸时控件的位置
滚动条颜色 滚动条图像 滚动条偏差 滚动条尺寸 滚动条自动隐藏	用于定义滚动条的外观和位置的参数
边距折叠	使用宽度更大的笔划参数折叠所有左右和上下边距。
外边距宽度 外边距颜色	外边距参数

添加或删除行或列

要添加或删除行或列，请双击网格，进入编辑模式，并右击列或行选择器以打开上下文菜单。



合并或拆分行或列

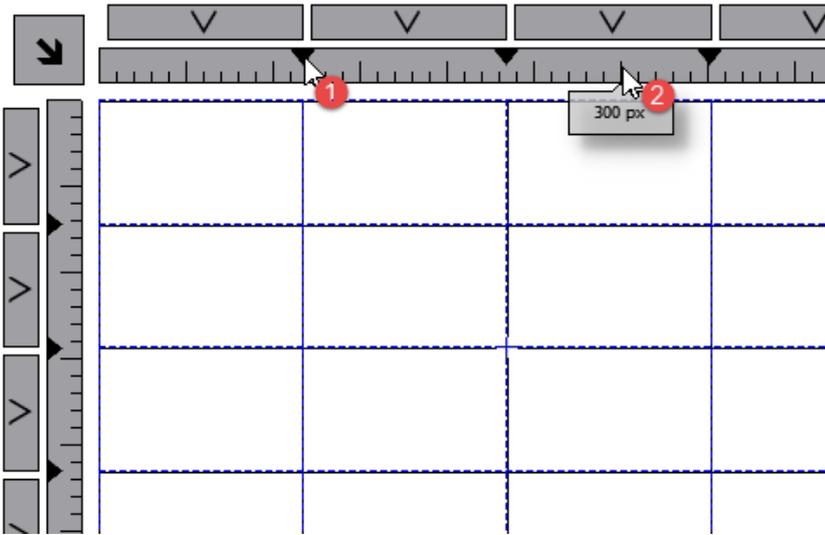
要合并或拆分行或列，请双击网格，进入编辑模式，然后让光标在功能区上移动：

- 双击黑色三角符号合并两个相邻行或列 (1)



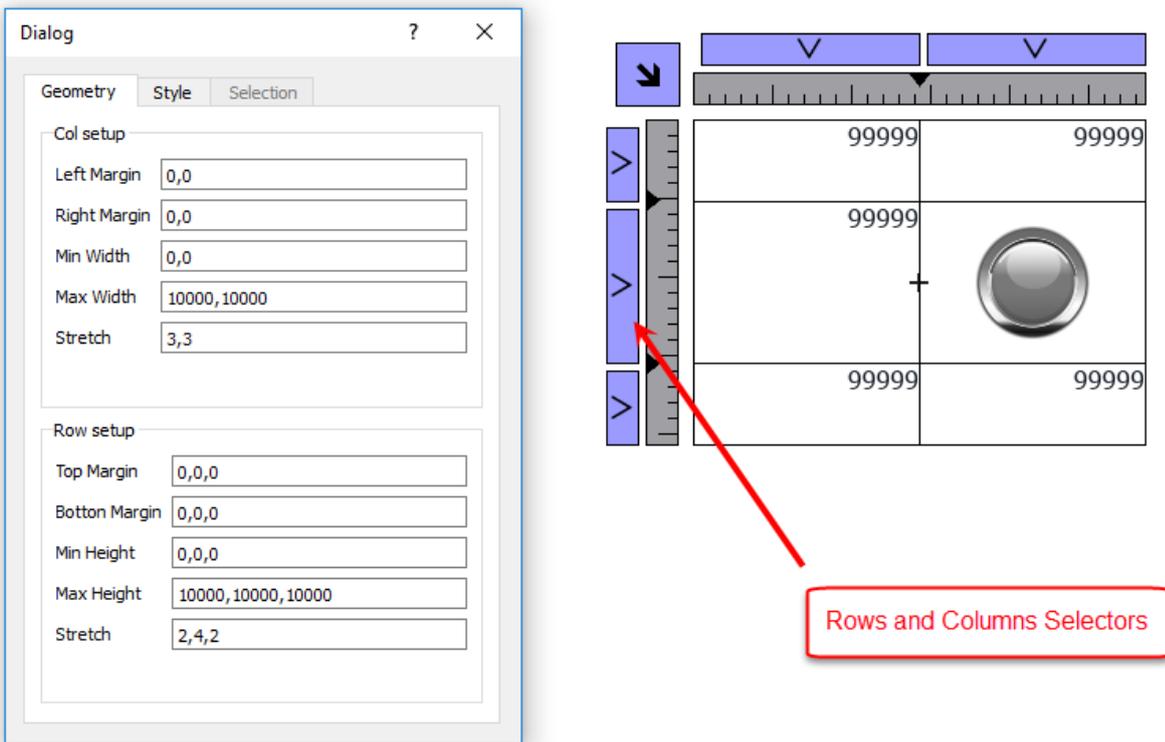
请注意，只能与空白行或空白列合并。

- 双击功能区拆分所选行或列 (2)



行、列属性

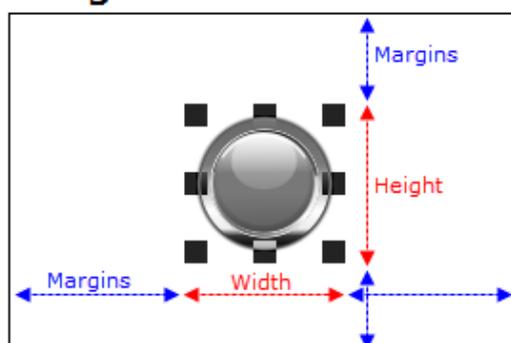
单击双击控件组时显示的行和列选择器后，行和列属性在弹出对话框中可见。



Stretch



Margins



几何结构参数

参数	说明
左边距 右边距	控件到单元格边界的距离
最小宽度 最大宽度	拉长单元格时，控件可呈现的最小/最大宽度
拉长	定义各列宽度之间的关系，如果拉长网格，将保持此关系
上边距 下边距	控件到单元格边界的距离
最小高度 最大高度	拉长单元格时，控件可呈现的最小/最大高度
拉长	定义各行高度之间的关系，如果拉长网格，将保持此关系

样式参数

参数	说明
左侧笔划宽度 右侧笔划宽度 顶部笔划宽度 底部笔划宽度	笔划宽度
左侧笔划颜色 右侧笔划颜色 顶部笔划颜色 底部笔划颜色	笔划颜色
背景颜色	行背景颜色



用逗号隔开的值列表与行和列相关。例如，第一个值表示行 0，第二个值表示行 1，依此类推。



颜色格式可以是 #rrggbb 或 #rrggbaa, 其中“aa”是 Alpha 值, 用于定义颜色的不透明度。

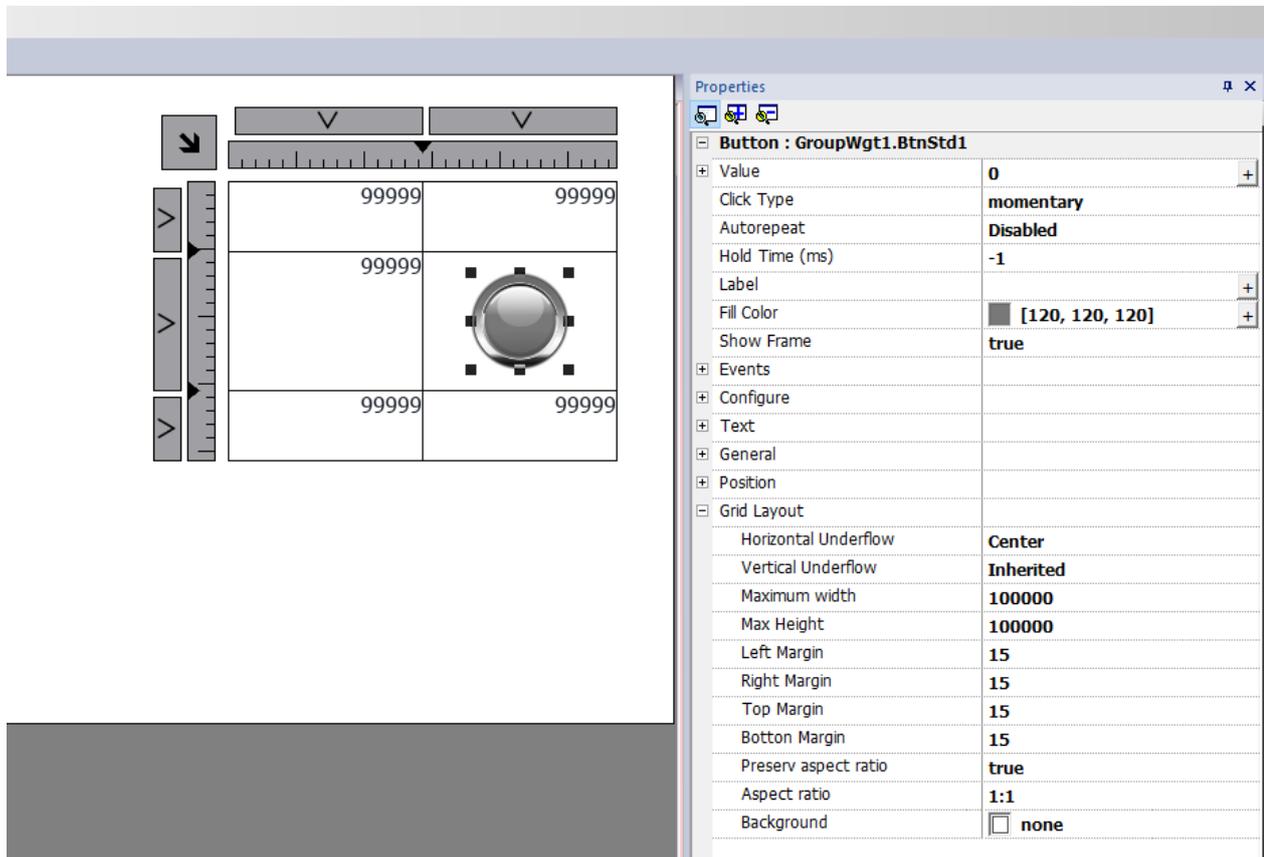
选择参数

仅当在表格控件内使用网格时, 选择参数才可用(请参阅"表格控件"在本页418了解详细信息)

参数	说明
前景颜色	选中该参数时, 行呈现的颜色
背景颜色	颜色列表与行模板相关。第一种颜色用于行模板 0, 第二种颜色用于行模板 1, 依此类推。
笔划颜色	

单元格属性

选中单元格时, 单个单元格的属性在属性面板中可见。选择单元格: 先双击控件组, 然后单击要选择的单元格。



参数	说明
水平下溢 垂直下溢	此参数定义单元格尺寸超过为控件定义的尺寸时控件的行为。 <ul style="list-style-type: none"> 继承 继承用于行或列的值 左、中、右 - 上、中、下 定义单元格尺寸超过最大定义尺寸时控件的位置
最大宽度 最大高度	覆盖全局网格参数
左边距 右边距 上边距 下边距	覆盖全局网格参数 将添加到总边距中的额外像素。
保留宽高比	保留控件的宽高比
宽高比	仅当“保留宽高比”为 true 时才可用
背景	单元格背景颜色

打印报表

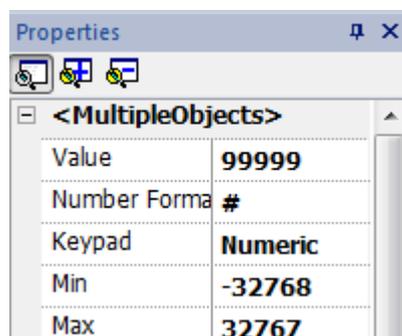
请注意，即使在打印报表库中，网格布局也可用。

更改多个控件属性

您可以同时设置同一类型的多个控件的属性。

要更改属性：

1. 选择控件
2. 从**属性**窗格中设置通用属性。
3. 当选择了多个控件时，“属性”窗格的标题更改为 **<多个对象>**：所有更改将会应用到选择的所有控件上。



注：不能同时对多个控件进行属性修改，因此必须单独修改。

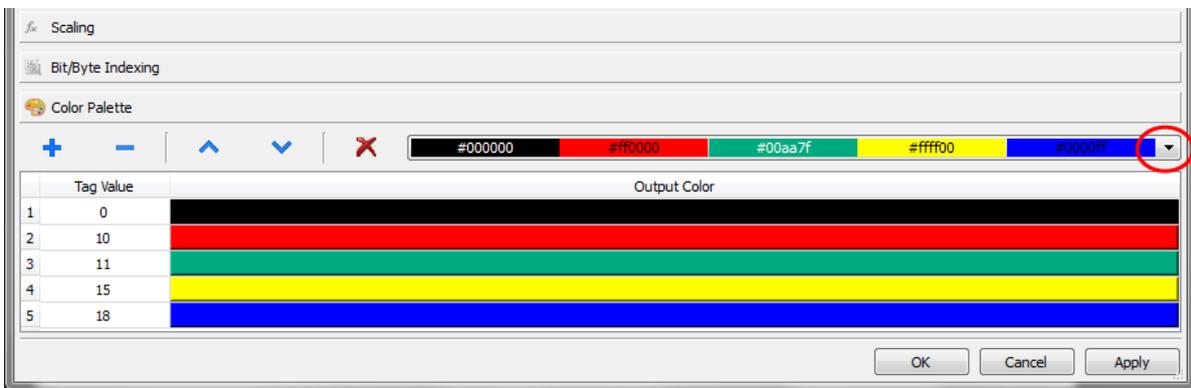
根据标签值更改填充颜色属性

LRH SW 允许根据标签值动态更改控件的颜色属性，方法有两种：

- 使用调色板
- 将颜色属性连接到字符串类型标签

使用调色板更改颜色属性

1. 创建您要因颜色管理而访问的标签(内部或 PLC)。标签可以是任何数据类型。基于该标签的值，颜色将更改。
2. 将该标签附加到一个对象(例如，一个按钮)的**填充颜色**属性中。
3. 在同一对话框中，选择**调色板**选项卡，根据标签值添加将对该对象使用的颜色。



注：系统保存最后使用的颜色表格，并且可以在工具栏上的“颜色列表”框中选择它们进行重复使用。

更改颜色属性，将颜色属性连接到字符串类型标签

1. 创建您要因颜色管理而访问的标签(内部或 PLC)。基于该标签的值，颜色将更改。标签必须是“字符串”类型，标签的 **Arraysize** 属性必须足够大以包含按此处说明被格式化的字符串。
2. 将该标签附加到一个对象(例如，一个按钮)的**填充颜色**属性中。
3. 在**字符串**标签中写入所需颜色的 RGB 颜色代码。使用这些格式之一：
 - **#XXYYZZ**，其中 XX、YY 和 ZZ 是以十六进制格式表示的所需颜色的 RGB 组件(范围 00–FF)。
 - **rgb(XXX,YYY,ZZZ)**，其中 XXX、YYY 和 ZZZ 是以十进制格式表示的所需颜色的 RGB 组件(范围 0–255)。



注：该功能可以应用在具有颜色属性的控件库中所有可用的对象上。运行时颜色的更改只对组成该对象的 SVG 的属性可用。该功能不能应用在其他图像格式上，例如 JPEG 或 BMP 文件。

5 页面

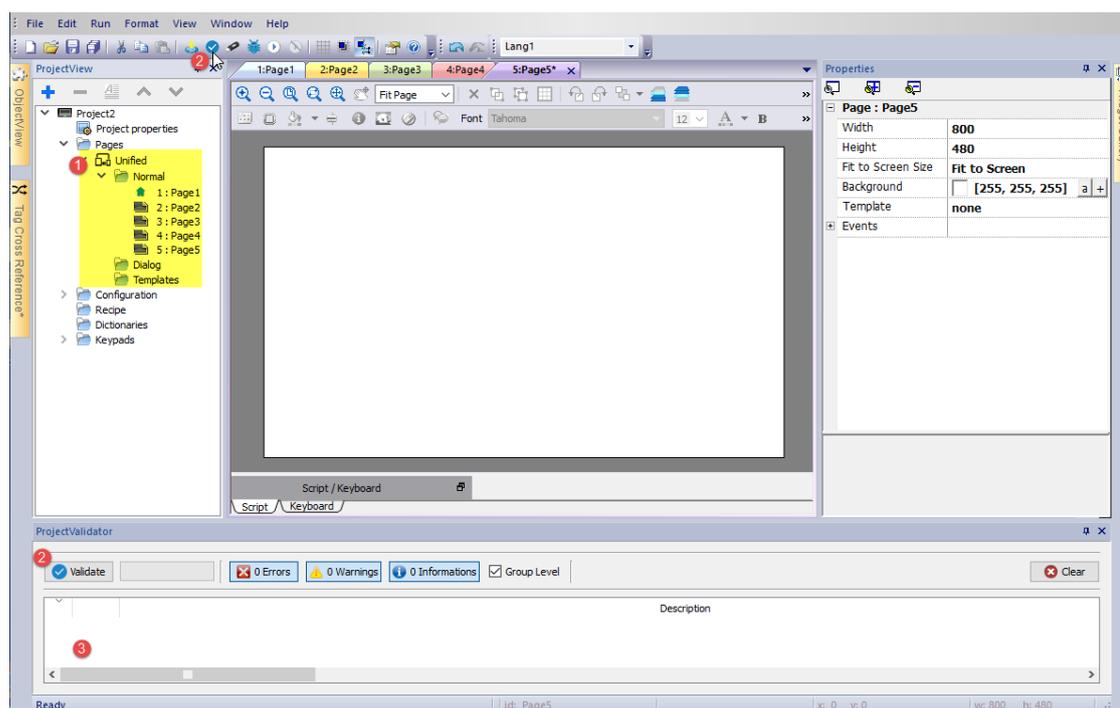
本节介绍页面的组织方式。您可以在所有客户端中显示相同的页面(默认模式)，也可以自定义页面来更好地适应每个不同的客户端。

统一页面	64
项目验证程序	64
区分页面	66

统一页面

从 LRH SW v4.0 开始，不再需要为 HMI 设备和 Web 客户端创建不同的页面。可以在 HMI 设备或 Web 客户端上以不同的方式呈现相同的页面。由于 Web 客户端可能不受支持某些属性或某些控件，因此某些页面可能会以不同的方式呈现。“项目验证程序”工具可用于检查某些页面是否包含将以不同的方式呈现到 Web 客户端中的控件。

1. 页面
2. 项目验证程序按钮
3. 项目验证程序输出信息

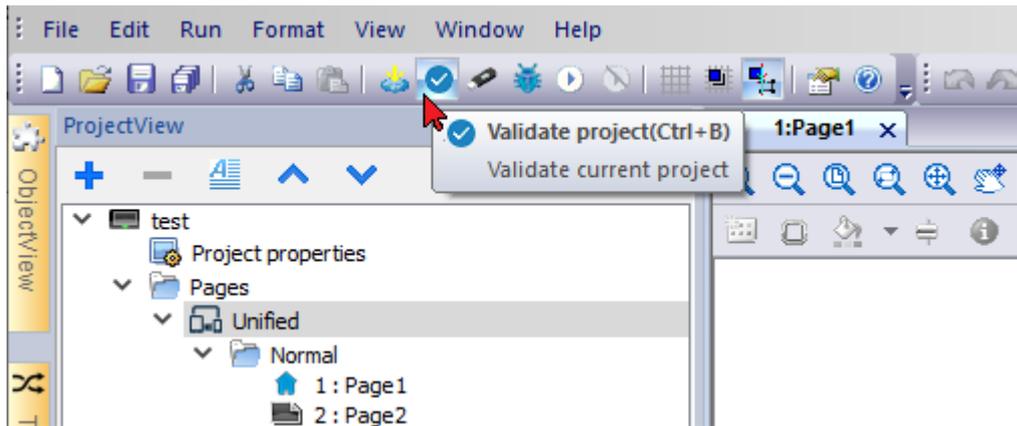


项目验证程序

“项目验证程序”工具检查并列出了将以不同方式呈现到 Web 客户端中的控件。用户可以双击项目验证程序报告中的每个警告消息，打开包含报告的控件页面以采取适当的措施。但是，用户操作不是强制性的，无论如何都可以下载项目，并且也不会对 Web 客户端对不受支持的属性进行管理。



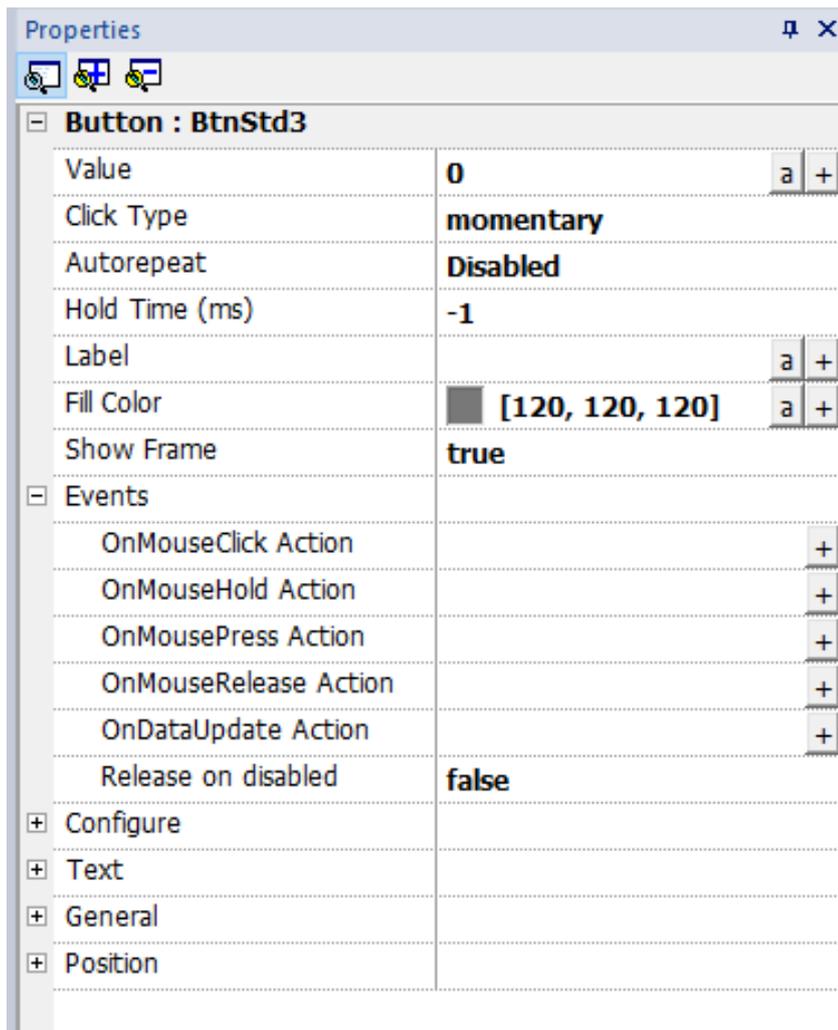
请注意，当前版本的项目验证器会检查小部件的 Web 兼容性。它不会检查整个项目（例如，缺少标签或 Javascript 错误）



例如

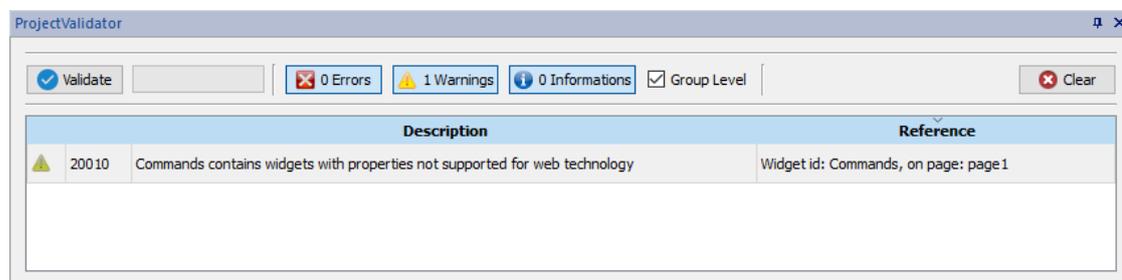
Web 客户端不受支持“Release on disabled”属性。

- 当属性设置为“False”时，HMI 设备和 Web 客户端将以相同的方式工作，并且项目验证程序将不报告任何消息。
- 当属性设置为“True”时，项目验证程序将报告警告消息。在项目将下载到 HMI 设备的情况下，Web 客户端不管理“Release on disabled”属性。

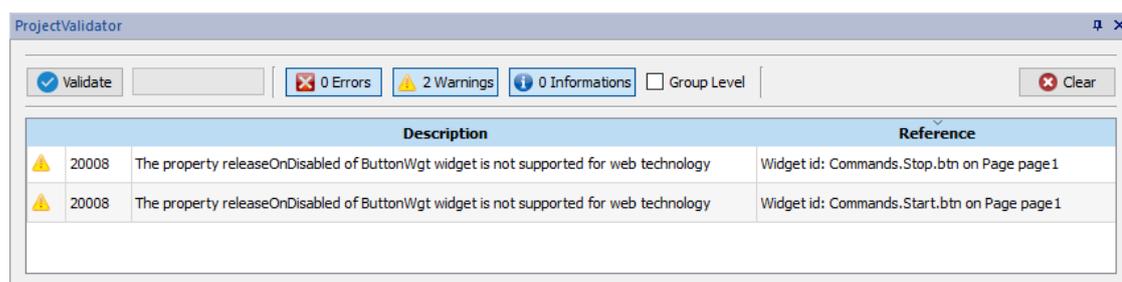


组级别

选中“组级别”后，项目验证程序将报告包含一个或多个具有不受支持属性的控件的组名称。双击选择编组控件。



如果未选中“组级别”，项目验证程序将报告不受支持的属性的列表。双击将选择具有不受支持属性的控件。



区分页面

如果项目需要具有用于 HMI 设备、Web 客户端、平板电脑客户端等的不同页面，则可以添加不同的文件夹以包含要在不同客户端上使用的页面。右键单击页面文件夹，添加页面的新类别。对于每个类别，您必须定义以下属性，其中“技术”、“用户代理”和“最小/最大”是筛选器参数，以定义属于该类别的 Web 客户端。

属性	说明
名称	类别名称
宽度, 高度	新建页面时使用的默认尺寸
技术	确定可以使用这些页面的客户端。它可以是以下各项的组合：

属性	说明						
	<ul style="list-style-type: none"> • 本地 HMI 设备 • 远程 LRH SW Client • Web 客户端 (PC、平板电脑、智能手机等) 						
用户代理	<p>是一个常规表达式，用于标识可以显示该类别页面的 Web 浏览器。Web 客户端的用户代理必须与此参数匹配。</p> <p>示例：</p> <table> <tr> <td>.*</td> <td>任何(所有网络客户端)</td> </tr> <tr> <td>Android</td> <td>仅 Android Web 客户端</td> </tr> <tr> <td>Android iPhone</td> <td>仅 Android 或 iPhone Web 客户端</td> </tr> </table>	.*	任何(所有网络客户端)	Android	仅 Android Web 客户端	Android iPhone	仅 Android 或 iPhone Web 客户端
.*	任何(所有网络客户端)						
Android	仅 Android Web 客户端						
Android iPhone	仅 Android 或 iPhone Web 客户端						
最小宽度 最小高度 最大宽度 最大高度	<p>定义显示该类别页面所需 Web 浏览器的显示尺寸。</p> <p>默认值 Min = 0 和 Max = -1，表示任何尺寸。</p>						



如果 Web 客户端的定义属于多个类别，请选择最相近类别中的可用 i 页面。

The screenshot displays the ProjectView window on the left and two Properties windows on the right. The ProjectView window shows a tree structure for 'Project1' with categories like Native, Web, Tablet, and Smartphone. The Properties windows show details for 'Page Category : Native' and 'Page Category : Tablet'.

Page Category : Native

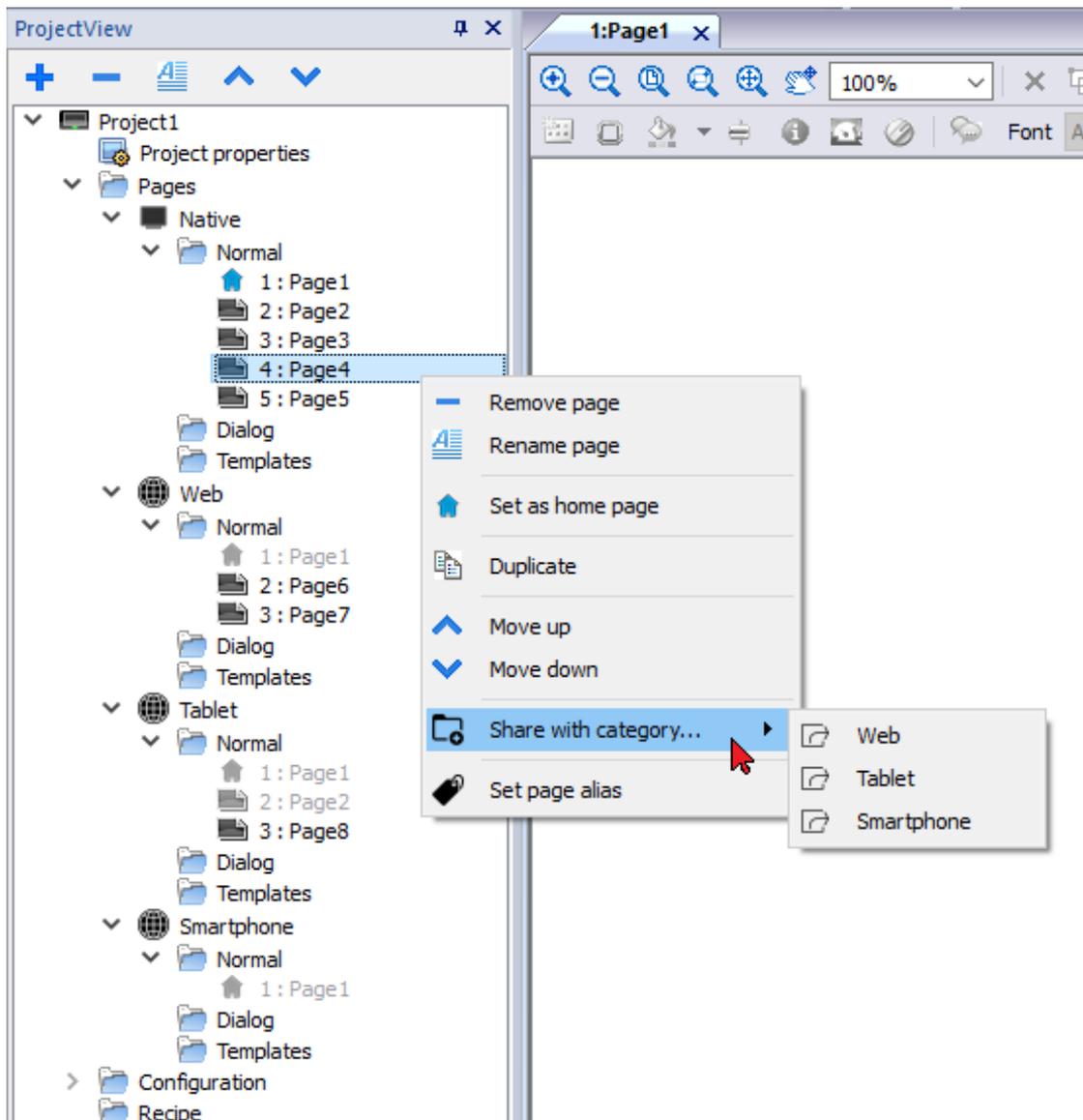
Name	Native
Width	1024
Height	768
Technology	HMI RemoteClient

Page Category : Tablet

Name	Tablet
Width	1024
Height	768
Technology	Web
Web Only	
User Agent	iPad RIM Tablet OS
Min Width	0
Min Height	0
Max Width	-1
Max Height	-1

共享页面

可以在类别之间共享页面。共享页面以灰色突出显示，并且可以在每个类别中以不同的方式打开。



主页

从页面的上下文菜单中可以定义类别的主页。主页定义了类别的浏览器中显示的第 1 页以及您的 Web 项目的起始点。您可以从主页访问的页面取决于项目中其他页面是如何链接的。

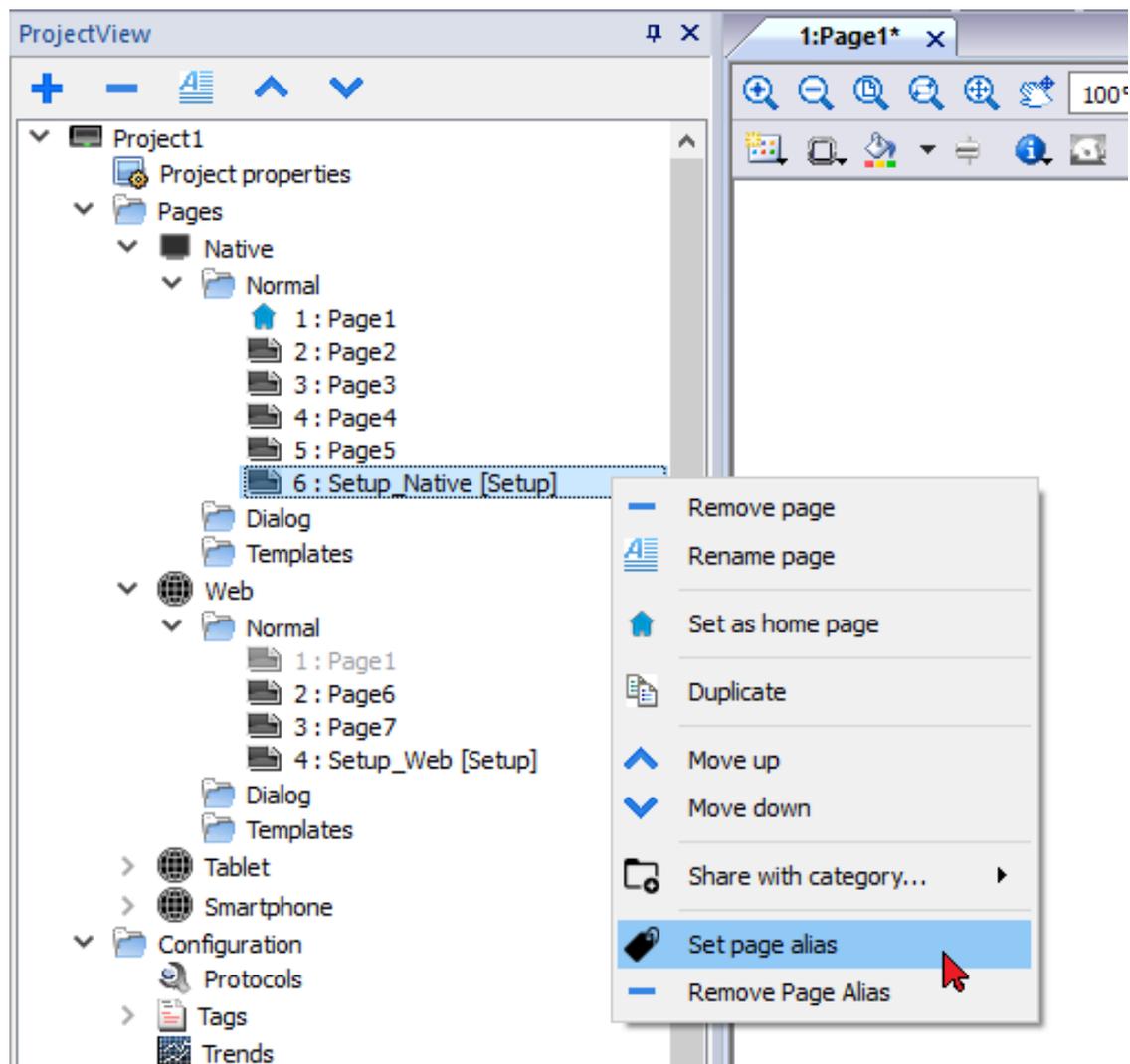
别名页面

使用类别之间的共享页面有助于别名页面参数加载适当的自定义页面。

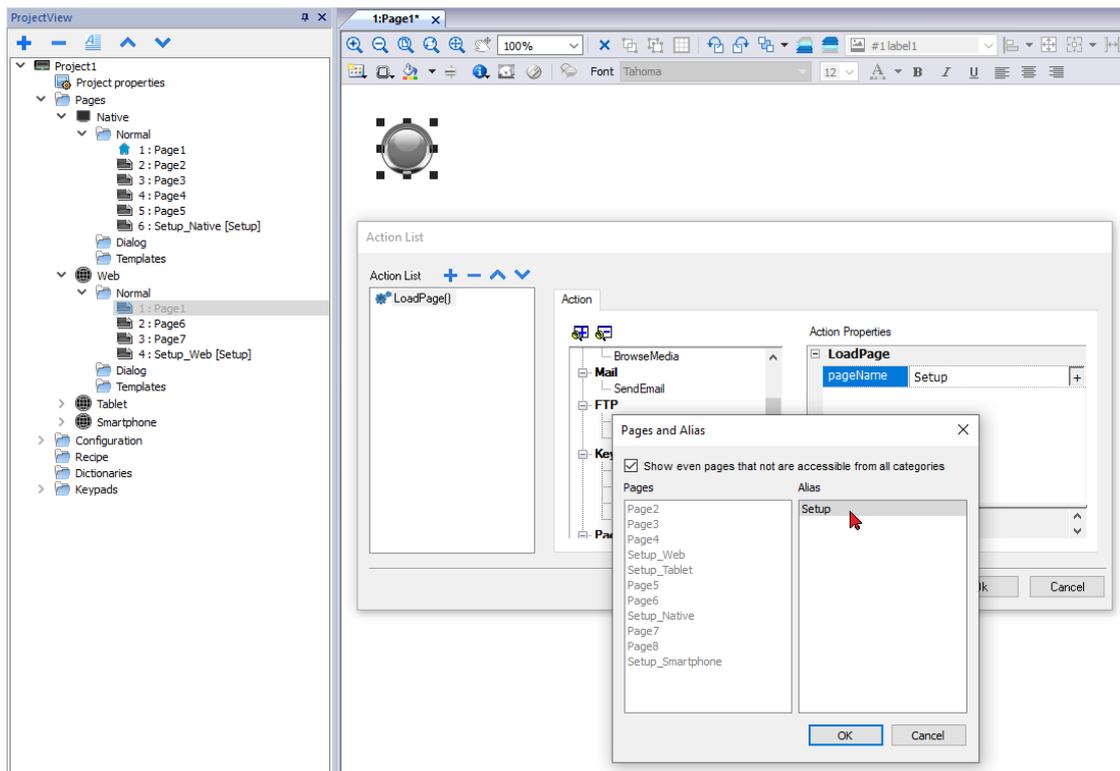
例如，您可以拥有所有类别共有的共享“Page1”。Page1 将同时显示在 HMI 设备和 Web 客户端上，但是您需要在该页面上添加一个宏以加载自定义安装页面。这意味着将在 HMI 设备上加载页面“Setup_Native”或在 Web 客户端上加载其他页面“Setup_Web”的宏。

要根据所使用的客户端加载不同的页面，可以在“Setup_Native”和“Setup_Web”页面上添加相同的别名，并在 LoadPage 宏中使用别名。

设置别名页面：



使用别名页面：

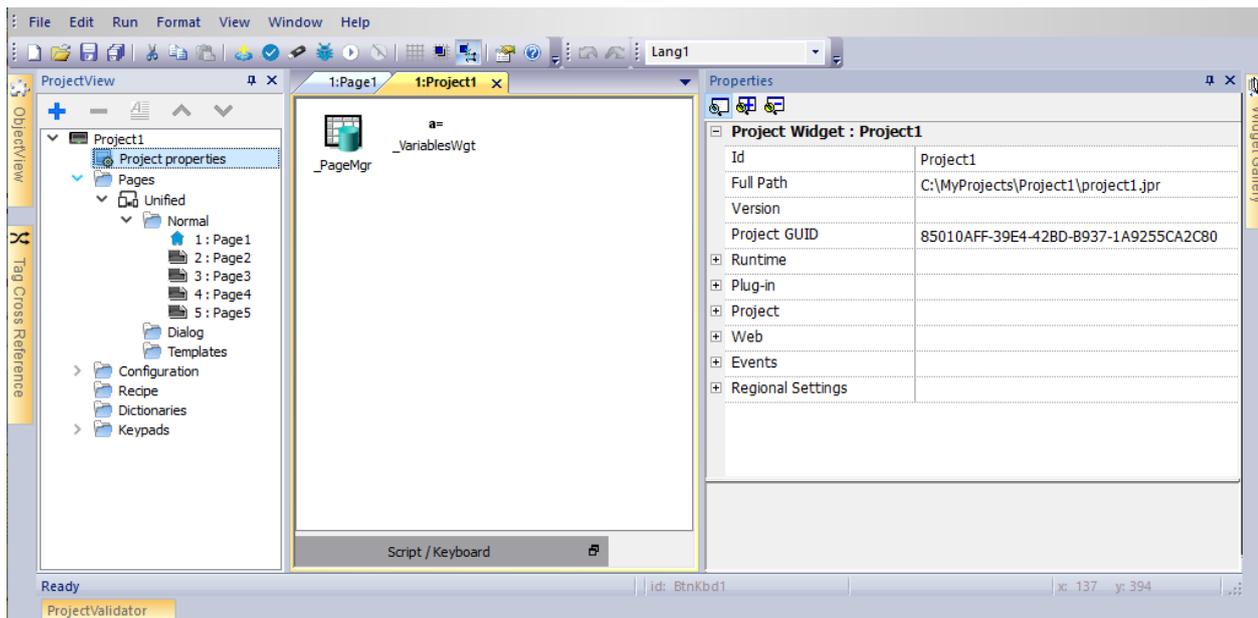


6 项目属性

项目属性包含项目的设置。

路径: **项目视图** > 双击 **项目属性** > **属性** 窗格

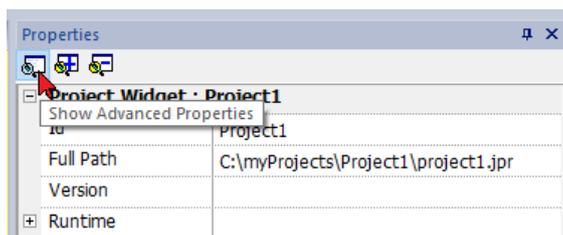
项目 **属性** 窗格包含项目级用户可配置数据的列表。



基本和高级属性

一些属性仅在高级模式下显示。要查看所有项目属性：

- 单击 **显示高级属性** 按钮以展开 **属性** 窗格中的属性视图。



可用属性

属性	说明
Id	项目名称(只读)
完整路径	项目路径(只读)
项目 GUID	项目唯一标识符(只读)

属性	说明
版本	用户可以使用“版本”字段来报告项目版本
+ Runtime	与应用程序 Runtime 相关的属性。请参阅 "Runtime" 向下 了解详细信息
+ 插件	可选模块。请参阅 "插件" 在本页 79 了解详细信息
+ 项目	与项目相关的属性。请参阅 "项目" 在本页 80 了解详细信息
+ Web	与 Web 接口相关的属性。请参阅 "网页" 在本页 83 了解详细信息
+ 事件	全局事件。请参阅 "事件" 在本页 85 了解详细信息
+ 区域设置	日期格式、列表分隔符、数字的千位和十进制符号的定义。请参阅 "区域设置" 在本页 85 了解详细信息

项目 Id、项目 GUI 和项目版本可从系统变量获得。请参阅 ["默认变量" 在本页 141](#) 了解详细信息。

Runtime

路径: 项目视图 > 双击项目属性 > 属性窗格

属性	说明
上下文菜单	<p>定义上下文菜单如何在 HMI 项目中出现。</p> <p>延迟 = 上下文菜单在触摸/按住屏幕的空白区域几秒钟后出现, 或通过上下文菜单动作显示</p> <p>动作 = 上下文菜单只通过上下文菜单动作显示。</p> <p>请参阅 "控件动作" 在本页 211 了解详细信息。</p>
开发者工具	启用/停用 runtime 调试应用工具集。
接触响铃	<p>触摸 HMI 设备屏幕上的控件时启用蜂鸣器。</p> <p>支持的控件:</p> <ul style="list-style-type: none"> • 按钮 • 热点 • 指针 • 域 • 外置键 • 组合框 • 表格 • 控制列表 <p> Windows CE 面板, 从 v1.76 ARM/2.79 MIPS 开始可用。</p>

属性	说明
响铃持续时间	默认 200 毫秒
键盘	当使用外置键盘时，在运行时启动键盘宏的使用。
JavaScript 调试	在运行时，对当前项目启用 JavaScript 调试。
允许 JS 远程调试程序	对当前项目启用 JavaScript 远程调试程序。  在 LRH SW Client 上不支持远程调试。
图像 DB 启用	激活 runtime 使用的引动程序，优化项目性能。  警告: 该属性只应在技术支持用于调试目的时被禁用，因为这可能在运行时降低性能。
FreeType 字体渲染	切换到 LRH SW 和 Runtime 使用的 FreeType 字体渲染。  使用 FreeType 的主要原因是我们需要在所有设备中使用相同引擎来避免不同的渲染，尤其是涉及静态优化的情况下。
通信图标延迟 (ms)	显示通信错误图标之前的延迟(默认为 0 mSec) 特殊值 -1 表示始终禁用
快速启动	当快速启动启用且用户界面在后台服务器之前启动时 <ul style="list-style-type: none"> • 默认:用户界面在准备使用后台服务器之后加载 • 快速 UI:用户界面在加载后台服务器之前加载

快速启动

当快速启动启用时，HMI 设备将在上电后尽快提供欢迎屏幕。在此模式下，在启动用户界面之前，只加载所需的最小功能。在加载用户界面之后，延迟加载协议、事件、趋势、报警和行动。

要设置两种标志：

- 在高级项目属性中可用的“快速启动”标志
- 在 BSP 系统设置工具“服务”页面中可用的“快速启动”标志(请参阅“系统设置”在本页 531)

当快速启动启用且用户界面在后台服务器之前启动时，JavaScript 事件 `project.onServerReady` 可用于获取服务器同步。

示例：

```
if (!project.serverIsReady) {
    // Set the callback to wait for server ready
    project.onServerReady = onServerReady;
} else {
    // Server is ready, call it now
```

```

    onServerReady();
}

function onServerReady()
{
    project.setTag("Tag1", 1);
    project.showMessage("Server is ready, tags can be used: " + project.getTag("Tag1"))
}
}

```



这是仅在 **Linux** 平台上可用的高级功能

开发人员工具

可以启用或停用的 **runtime** 调试功能集。

1. 在属性窗格中，将**开发人员工具**设为 **true**。
2. 下载项目。
3. 打开背景菜单。
4. 选择**开发人员工具**。

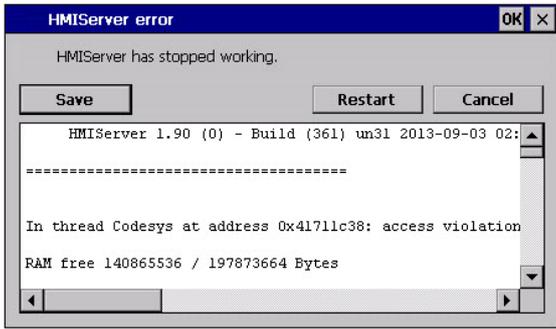
开发人员工具列表

工具	说明
显示/隐藏所有	显示包含关于设备状态信息的对话，如 CPU 负荷、存储使用、事件队列。
CPU 数据	显示 CPU 负荷的信息。请参阅 "CPU 数据" 对页 。
内存统计信息	先是关于系统 RAM 的信息。负数值说明空存储在减少。
事件队列	显示事件队列信息(大小、最大限度实现大小处理事件的数量，最后和最长的处理时间)。计时数据只对非 UI 队列可用。
Timelog 摘要	显示页面加载时间。
嵌入窗口	在 runtime 允许嵌入场景或将开发者工具窗口留作独立窗口(对话)。
复位队列统计	复位事件队列统计信息。
禁用监视狗	停用监测功能，防止系统在不动或服务失灵时重启。
忽略异常	停用失灵报表功能，异常并不保存在事故报表窗口。
剖析	计算加载/绘制活动页面的时间。请参阅 "剖析" 对页

监测

该功能使您能够停用监测。这样在 **Runtime** 失灵的情况下避免系统重启，并有时间保存事故记录或检查系统状态信息(可用内存、**CPU** 负载、事件队列大小等等)。

事故报表对话在系统冻结或崩溃的情况下自动显示，使用户可以保存事故记录文件。



重要:给技术支持保存这份文件。

CPU 数据

```
2014-04-25 23:02:48, up: 0:08:27, idle: 24 *
Period 2110 ms (overhead 69ms)
  Thread          ID Prio    ms kernel/  user
 *              59637774  3    697    0/   697
   Codesys      78839810  0     8     0/     8
Other threads < 5ms
RAM free 125833216 / 194211840 Bytes (diff: 0)
ImageDB size ~2MB, free 4MB / RMSIZE-76MB)
Page Preload 56MB free / RMSIZE-64MB)
Page Cache 80MB free / RMSIZE-40MB)
Storage free 45 / 92 MB

EvQueue  Size  MaxSize    EvtS    ms  max(ms)
EvtMgr   0      0          0      0      0
ActionMgr 0      1         61     22     189
AlmMgr   0      0          0      0      0
MODR     0      0        122     11     15
UI        0      11        270    --     --

Timelog is disabled!
(Tap-tap to change position)
```

在顶行, 当前机器时间和设备可用的总时间一同显示。

CPU 数据以 2000 毫秒的频率收集。收集和显现数据所需的实际周期和开销也被显示。实际周期与标准的 2000 毫秒相差越大, 系统符合越大。进程的 CPU 消耗被列出, 报表进程的名称(可行的话, 进程用一个*标记)、进程的 ID, 进程优先级以及在 2000 毫秒周期内 CPU 使用时间, 以用户和内核划分。

剖析

剖析使您可以检查加载/绘制活动页面的时间。剖析将在下一个页面加载开始, 仅对屏幕的页面的第一幅图片活跃(配置会被保存)。

```

2014-04-25 23:27:19, up: 0:32:58, idle: 36 *
Period 2053 ms (overhead 47ms)

Page "&alarms.jmox":
          START      dT (ms/ cpuMs)
Time parsing   : +    6    45/   45
Time unloading : +   54    6/    6
Time lst update : +  195    3/    0
Time gfx creation: +  198  300/  133
          OnLoad :    241/   94
Time rendering : +  535  390/  387
ImageDB cache 15 hit/0 miss(0 ms, cpu: 0 ms)

Page "TemplatePagel.jmox":
Time init/start : +   60  133/   86
Time lst update : +  195    2/    0
Time gfx creation: +  459   27/   27
          OnLoad :    9/    9
ImageDB cache 28 hit/0 miss(0 ms, cpu: 0 ms)

(Tap-tap to change position)

```

剖析选项	说明
启用时间记录	启用时间记录捕捉。可在“时间记录摘要”窗口中看到时间记录。
将时间记录保存至文件	保存一个报告，其中包含配置文件详细信息和将项目及其页面加载到 <code>timelog.txt</code> 文件所需的时间。该文件可以输出和共享以备进一步分析。  重要：该功能的执行可能降低页面更改性能。
覆盖加载时间 覆盖渲染时间	该视图可以显示单一控件所用时间，仅在渲染和加载步骤可用。该视图提供所用时间的位置的直接体验。红色区域代表时间最关键的区域。详细的控件时间可以通过工具提示窗口可视化。不在场景范围内时，一些箭头可以导航到这些区域，在这些区域停留，工具提示将会在显示区域概要
选择覆盖颜色	选择要使用的覆盖颜色

时间记录数据

数据	说明
时间解析	分析当前页面所用的时间。取决于页面复杂性/控件的数量。
时间 gfx 创建	图像渲染所用的时间。主要与 <i>加载方法</i> 相关。
渲染时间	渲染页面所用的时间。
卸载时间	如果当前页面依赖于另一个页面，卸载该页面所用的时间。

时间成对提供：时钟时间 /CPU 时间。时钟时间是这一部分要求的绝对时间，可能比要求的实际的 CPU 时间高，因为高优先级的进程也在运行(如实例协议)。开始时间栏指的是页面加载开始时间。它用于跟踪要求加载的页面的实际时间，因为部分时间仅指的是时间最关键的的功能，不包括通常占总时间很大部分的其他时间。

例如，加载一个页面要求的实际总时钟时间是渲染(最后一个步骤)开始时间+渲染的时钟时间。

FreeType 字体渲染

新项目默认使用 **FreeType** 字体引擎。使用旧版本 **LRH SW** 创建的项目会使用旧字体引擎，在项目转化后亦是如此，以避免任何向后兼容性问题。



任何可能的时候请转化为 **FreeType** 字体以获得更好的页面渲染。

一旦转换为新的字体渲染，保存项目并验证所有文本在所有项目页面中显示正确。

字体渲染事件

当转换为 **FreeType** 字体程序时，对于使用旧字体程序创建的项目，您可能遇到以下问题：

- 文本要求更大 / 更小像素渲染，因此会改变文本布局
- 控件会改变大小来适应文本
- 使用修边可以实现更好的渲染(修边是一种文本控件属性)

插件

您可以选择在该项目下载哪些软件模块到 **Runtime**。软件插件设计的目的是减少 **HMI** 设备的内存要求，**HMI** 设备存储量是有限的。

路径: 项目视图 > 双击 **项目属性** > **属性** 窗格

属性	说明
WebKit	浏览器控件需要的模块
TextEditor	TextEditor控件需要的模块
PDF 阅读器	PDF 阅读器
VNC 服务器	VNC 服务器



注: 并非所有软件插件模块都与所有 **HMI** 设备平台兼容。

一旦启用，软件插件成为 **runtime** 的一部分。使用以下步骤之一通过 **LRH SW** 来安装它：

- 安装 **Runtime**/更新 **Runtime**
- 更新包

要从 **Runtime** 上除去插件，使用系统模式中的以下功能之一：

- 格式化闪存
- 恢复出厂设置



重要: 系统不能自动检测 **HMI** 应用程序需要的软件插件，务必在项目属性中将其全选。

项目

这些属性限定了页面行为的各种元素。

路径: 项目视图 > 双击项目属性 > 属性窗格

属性	说明						
主页	运行时加载的第 1 页(如果启动了项目安全性,则在注册页面之后)。 当安全性启动后,您可以对每一组用户指定不同的主页 (HomePage)。在这种情况下,这一设置被忽略。请参阅"用户管理和密码"在本页 317 了解详细信息。						
页面宽度 页面高度	限定 HMI 页面的尺寸像素。默认的是创建此项目时选择的 HMI 设备型号的显示分辨率。						
显示模式	定义 HMI 设备方向。						
项目类型	定义项目的 HMI 设备类型。根据型号自动调整某些项目特征和属性。  警告:从 v2 起, LRH SW HMI Runtime 将检查所选项目类型与 HMI 设备型号是否匹配,并在所选类型不匹配时提供建议:“HMI 类型不匹配。转换项目并再次下载。”						
面板内存	可用内部面板内存的大小。						
页面要求 现有页面 同步选择	您可以从 PLC 之类的控制器中同步 LRH SW HMI Runtime 和 LRH SW Client 显示的页面。 附着的标签必须包含项目页面范围内的整数值,并且必须至少可作为读资源使用。  Web 浏览器支持,请查看 "网页" 在本页 83						
保持时间 自动重复时间	限定按钮和外置键盘持续时间和自动重复时间的值。  注:可以在控件属性表中针对每一个按钮和键对这些属性进行重新定义。						
在启动 时隐藏项目加载	隐藏时,开机画面将停留在屏幕上,直到应用程序已准备运行。						
目标缩放系数	当项目在 HMI 设备运行时的放大系数。 <table border="1" data-bbox="427 1668 1364 1960"> <tbody> <tr> <td>范围</td> <td>0.3–2.9</td> </tr> <tr> <td>缩放至屏幕</td> <td>-1 = 缩放至屏幕大小  缩放屏幕保持宽高比。确定缩放系数,即宽度与高度的比例,然后取最小值。</td> </tr> <tr> <td>默认值</td> <td>1 = 不放大</td> </tr> </tbody> </table>	范围	0.3–2.9	缩放至屏幕	-1 = 缩放至屏幕大小  缩放屏幕保持宽高比。确定缩放系数,即宽度与高度的比例,然后取最小值。	默认值	1 = 不放大
范围	0.3–2.9						
缩放至屏幕	-1 = 缩放至屏幕大小  缩放屏幕保持宽高比。确定缩放系数,即宽度与高度的比例,然后取最小值。						
默认值	1 = 不放大						

属性	说明
背景颜色选项	当所定义的网页比整个显示区域小时，页面未覆盖的区域将变成彩色(例如当页缩小时)
	无 旧模式，颜色为白色(默认)
	所选颜色 所用颜色
	页面背景 根据页面模板背景自动调整颜色
签名	要用于签名的算法 <ul style="list-style-type: none"> • sha256 • sha1
手势密码已启用	启用在可配置延迟后将手势事件传递给底层控件的可能性。用户必须按住手指，然后执行手势。
手势密码延迟 (ms)	启用时，在此延迟后将手势事件传递给底层控件(请参阅 "手势事件密码"在本页404 了解详细信息)
手势多点触摸	启用多点触摸手势 <ul style="list-style-type: none"> • false • true(默认)  此属性提供了禁用多点触摸手势的可能性。这可用于避免未设计用于管理多点触摸手势的旧项目问题。
访问已拒绝	当用户尝试使用从安全配置锁定至只读的控件(例如，字段或按钮)时，挂锁图标会显示几秒钟，以突出显示该控件不可访问。 <ul style="list-style-type: none"> • 无 • 显示图标
组合框视力模式	选择项目所有组合框控件的可视化模式(请参阅 "组合框控件"全屏"模式(支持图片)"在本页394 了解详细信息) <ul style="list-style-type: none"> • 上下文菜单 经典视图，具有下拉菜单 • 全屏 增强视图，具有可配置文本和图像，它们将在屏幕中间弹出，以便滚动和选择。

页面要求，现有页面和同步选择

可以在 HMI 显示的页面上显示 LRH SW HMI Runtime 的设备信息。您可以将 HMI 设备上显示的页面与 LRH SW Client 上的页面同步，或者从 PLC 之类的控制器控制 HMI 项目。

以下属性可以定制：

属性	说明
页面要求	HMI 设备和 LRH SW Client 将显示的页面。 附着的标签必须包含项目页面范围内的整数值，并且必须至少可作为读资源使用。
现有页面	HMI 设备或 LRH SW Client 或两者上都显示的页码。 附着的标签必须至少可作为写资源使用，并且必须为整数数据类型。
同步选择	将项目页面及包含的值同步到 现有页面 属性中。 选项可以是： <ul style="list-style-type: none"> • 禁用: 忽略页码值， • 本地: 页码显示在 HMI 上， • 远程: 页码值显示在 LRH SW Client 上。 • 本地 + 远程: 页码显示在 HMI 和 LRH SW Client 上，如果显示了不同的页面，则考虑最后加载的页面。

例如：从控制器/ PLC 强制更改 HMI 设备和 LRH SW Client 的页面

按如下设置属性：

页面要求	附在标签“A”上
现有页面	空白
同步选择	停用

设置标签“A”的值来显示 HMI 设备和 LRH SW Client 请求的页面。

例如：从控制器/ PLC 强制更改 HMI 设备和 LRH SW Client 的页面。读取 HMI 上加载的当前页面

按如下设置属性：

页面要求	附在标签“A”上
现有页面	作为读/写附着在标签“B”上
同步选择	本地

设置标签“A”的值来显示 HMI 设备和 LRH SW Client 请求的页面。标签“B”将包含设备当前显示的页面值。

例如：从控制器/ PLC 强制更改 HMI 设备和 LRH SW Client 的页面。读取 LRH SW Client 上加载的当前页面。

按如下设置属性：

页面要求	附在标签“A”上
现有页面	作为读/写附着在标签“B”上
同步选择	远程

设置标签“A”的值来显示 HMI 和 LRH SW Client 请求的页面。标签“B”将包含 LRH SW Client 当前显示的页面值。

例如：从控制器/PLC 强制更改 HMI 设备和 LRH SW Client 的页面。强制 LRH SW Client 页面与 HMI 设备同步(而不是反过来)。

按如下设置属性：

页面要求	作为读/写附在标签“A”上
现有页面	根据页面要求附到同一标签“A”
同步选择	本地

设置标签“A”的值来显示 HMI 和 LRH SW Client 请求的页面。更改 HMI 上的页面以显示 LRH SW Client 上的相同页面。

例如：从控制器/PLC 强制更改 HMI 设备和 LRH SW Client 的页面。强制 HMI 页面与 LRH SW Client 同步(而不是反过来)。

按如下设置属性：

页面要求	作为读/写附在标签“A”上
现有页面	根据页面要求附到同一标签“A”
同步选择	远程

更改标签“A”的值来显示 HMI 和 LRH SW Client 请求的页面。更改 LRH SW Client 上的页面以显示 HMI 上的相同页面。

例如：同步 HMI 设备和 LRH SW Client 上显示的页面

按如下设置属性：

页面要求	作为读/写附在标签“A”上
现有页面	根据页面要求附到同一标签“A”
同步选择	本地 + 远程

更改 HMI 设备上的页面，相同的页面将显示在 LRH SW Client 上，反之亦然。

网页

路径：项目视图 > 双击项目属性 > 属性 窗格

属性	说明	
Web 不活动超时	定义 LRH SW 客户端的超时时间。当客户不活跃时间超时，当前用户会被退出。	
	范围	1-86400 秒 (从 1 秒到 24 小时)
	默认值	600 秒
	值	0 = 已停用
Web 图标	收藏图标与网页相关联	
刷新时间	定义 Runtime 与 LRH SW 客户端之间的通信的刷新时间。	
	范围	500-10000 毫秒
	默认值	3000 毫秒
浏览器优化	true	启用 Web 引擎优化 (默认)
	false	禁用 Web 引擎优化 (对不支持 Web 引擎优化的旧浏览器很有用)
启用远程全局 JavaScript	定义在“项目属性”内定义的 JavaScript 代码 (通常从报警和计划事件中触发) 是否只能在本地 HMI 设备抑或是在远程客户端上运行。	
	无	将不在远程客户端上执行 (只在本地 HMI 设备中运行)
	客户端	将在 LRH SW Client 上执行
	网页	将在 Web 客户端上执行
	二者	将同时在 LRH SW Client 和 Web 客户端上执行
最大带宽 (Kbs)	服务器发送的最大数据限值 (对速度缓慢的旧浏览器很有用)。设为 0 以使用所有可用带宽 (默认)	
网络客户端连接模式	自动	客户端选择的连接模式 (默认)
	服务器推送事件	强制采用“服务器推送事件”模式
	长轮询	强制采用“长轮询”模式
WebPageRequest	<p>您可以从 PLC 之类的控制器中同步 LRH SW 客户端上显示的页面。</p> <p>LRH SW 客户端上将显示的页面。</p> <p>附着的标签必须包含项目页面范围内的整数值，并且必须至少可作为读资源使用。</p>	
Web 通信图标延迟 (ms)	显示通信错误图标之前的延迟 (默认为 0 mSec)	
	特殊值 -1 表示始终禁用	



project.getClientType() 可用于检索运行客户端类型。有关更多详细信息，请参阅 JavaScript 章节中的“

事件

路径:项目视图 > 双击项目属性 > 属性窗格

属性	说明
滚轮	<p>仅与带滚轮输入设备一起使用。通常滚轮是在没有外置键盘设备的情况下用于增加或减少标签值。</p> <p>将这一属性附加到滚轮事件的更改中，并使用 BiStep 之类的动作来增加/减少标签值。</p> <p> 只有在滚轮动作不会被加载的页面覆盖时才会执行项目的滚轮动作。</p>

区域设置

路径:项目视图 > 双击项目属性 > 属性窗格

属性	说明
短日期格式	用户在控件的日期格式中选择 SHORT-DATE 时要使用的日期格式
长日期格式	用户在控件的日期格式中选择 LONG-DATE 时要使用的日期格式
列表分隔符	列出要在转储文件中使用的分隔符。
小数点符号	在数字控件中使用的字符，用于将整数部分与小数部分分开(仅当用户配置控件以显示小数部分时才可见)
千位数符号	数字控件中用于分隔千位时要使用的字符(仅当用户将控件配置为显示千位字符时才可见)



您可以使用占位符自定义时间和日期格式(请参阅 ["时间和日期占位符"](#) 在本页 402)

7 HMI 模拟器

HMI 模拟器允许在将项目下载到 HMI 设备前对项目进行测试。当没有 HMI 设备可用时，它可以用来测试项目，加快进程并对活动进行调试。

HMI 模拟器支持：

- 在线模拟 — 与真实设备通信 (仅针对有网络的协议或 RS-232 通信)，
- 离线模拟 — 模拟标签状态

数据模拟方法在标签编辑器的 **模拟器** 列中设置。

数据模拟方法	88
模拟器设置	88
启动和停止模拟器	89

数据模拟方法

在标签编辑器的**模拟器**字段中设置标签模拟行为。

方法	说明
变量	数据存储在一个模拟器变量中。该变量保持标签的值，您可以读写该值。
锯齿	计数值从 偏移量 增至 幅度 + 偏移量 值， 周期 为 60..3600 秒。当计数器到达 幅度 + 偏移量 时，值复位为 偏移量 ，并且计数器重启。
正弦波	正弦波值产生，并被写入到标签值。可以为每个标签定义 最小、最大和周期 值。
三角波	可以为每个标签定义 最小、最大和周期 值。
方形波	可以为每个标签定义 最小、最大和周期 值。

请参阅["添加标签"](#)在本页 101 了解详细信息。

模拟器设置

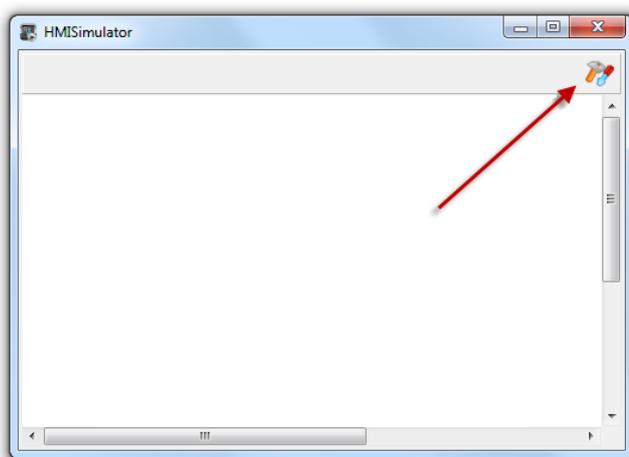
模拟器默认以模拟协议工作。也可以以真实协议工作(以太网或串行协议)



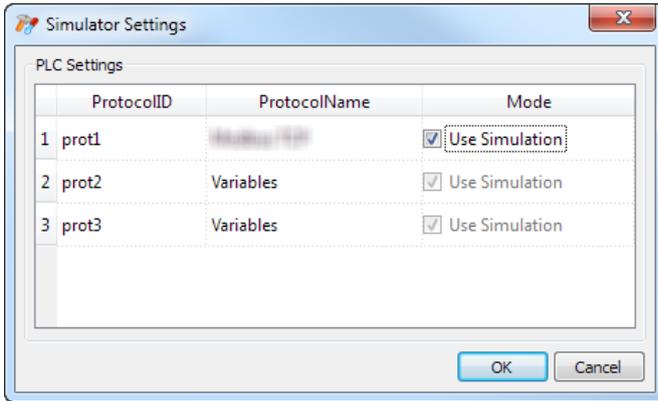
注:对于不支持与外置设备通信的协议,如变量协议,该选项始终被禁用。

更改模拟协议

1. 单击模拟器的**设置**图标。



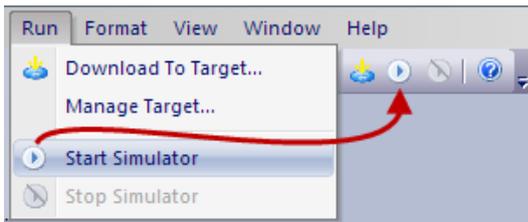
2. 选中**使用模拟**来使用模拟协议，否则真实协议将会被用于与外部设备通信。



启动和停止模拟器

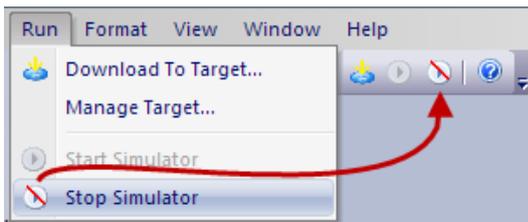
要启动模拟器：

1. 在运行菜单上，单击启动模拟器：模拟器按照服务器在 HMI 设备上运行的方式在计算机上运行。



要停止模拟器：

1. 在运行菜单上，单击**停止模拟器**或者在模拟页面上，双击**退出**按钮。



8 向 HMI 设备传输项目

要将 LRH SW 项目传输到目标 HMI 设备，您可以使用：

- 运行 > 下载至目标功能
- 运行 > 更新包功能(通过使用 USB 设备)

下载到 HMI 设备	92
更新包	94
Runtime 加载器	97
上传项目	98

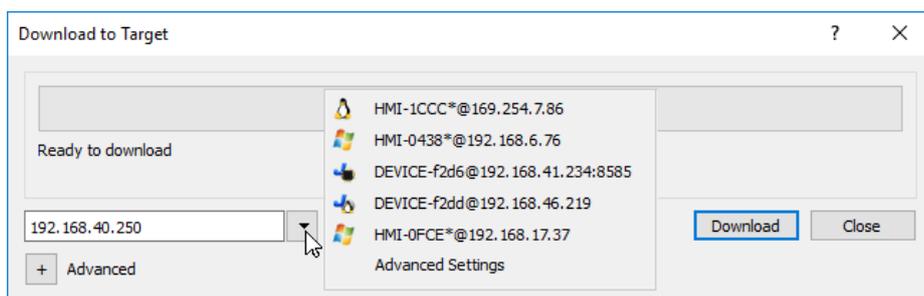
下载到 HMI 设备

路径: 运行 > 下载至目标

此功能通过以太网传输项目和 LRH SW HMI Runtime。

 注: HMI 设备必须有有效的 IP 地址。请参阅“HMI 设备基本设置”在本页 8, 了解如何分配 IP 地址的详细信息。

1. 点击发现按钮: 一系列被检测到的 IP 地址显示。
2. 选择 HMI 设备的 IP 地址。

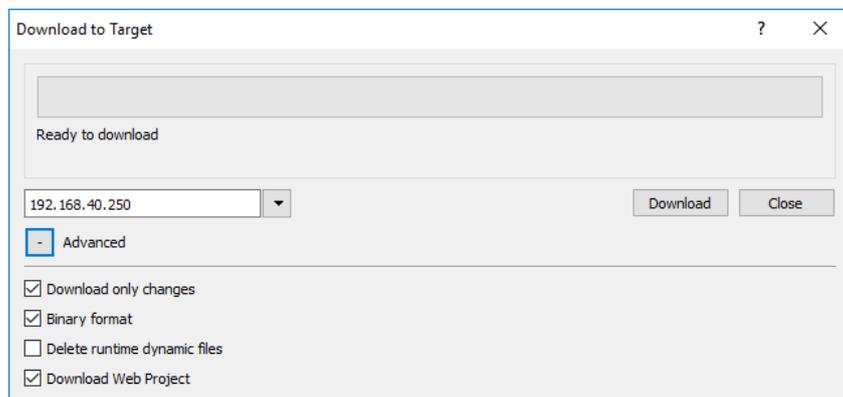


您甚至可以手动输入 IP 地址, 或者在可行时使用 DNS 服务器提供的主机名。可以使用 Bonjour 之类的服务工具通过设备主机名(例如 HMI-0d37.local)找到的 HMI 设备。Bonjour 是 Apple inc 的商标。

3. 单击**下载**: LRH SW 将 HMI 设备切换到配置模式, 并传输文件。

当下载操作完成时, HMI 设备自动切换回操作模式, 项目启动。

高级选项



选项	说明
仅下载更改	仅传输修改过的项目文件到 HMI 设备。
二进制格式	下载使用二进制格式的文件。
删除 runtime 动态文件	运行时修改的配方、用户或计划任务等将会被删除并被项目中定义的配置改写。  小心: 该操作不可逆, 删除的动态文件无法恢复。

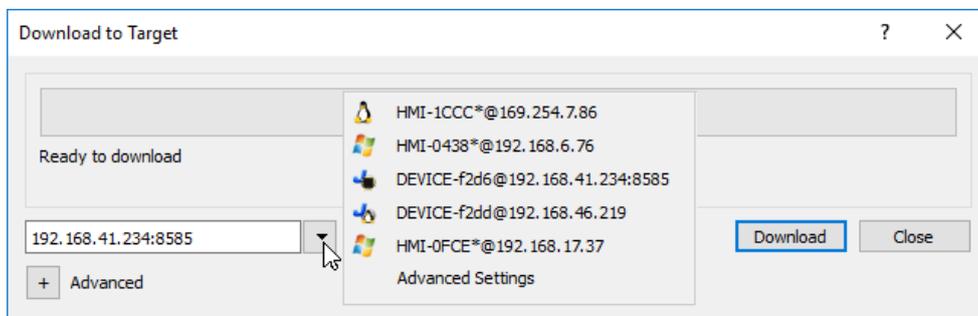
选项	说明
	 小心: 如果存储在外部设备(USB 或 SD 卡)上, 动态文件无法删除。
下载 Web 项目	将 LRH SW 页面下载到 HMI 设备。

传输项目时, LRH SW 使用 HTTP 和 FTP 组合连接:

- HTTP 连接发出命令切换传输模式或退出运行的项目,
- FTP 步骤传输文件到 HMI 设备上的闪存中。

高级设置

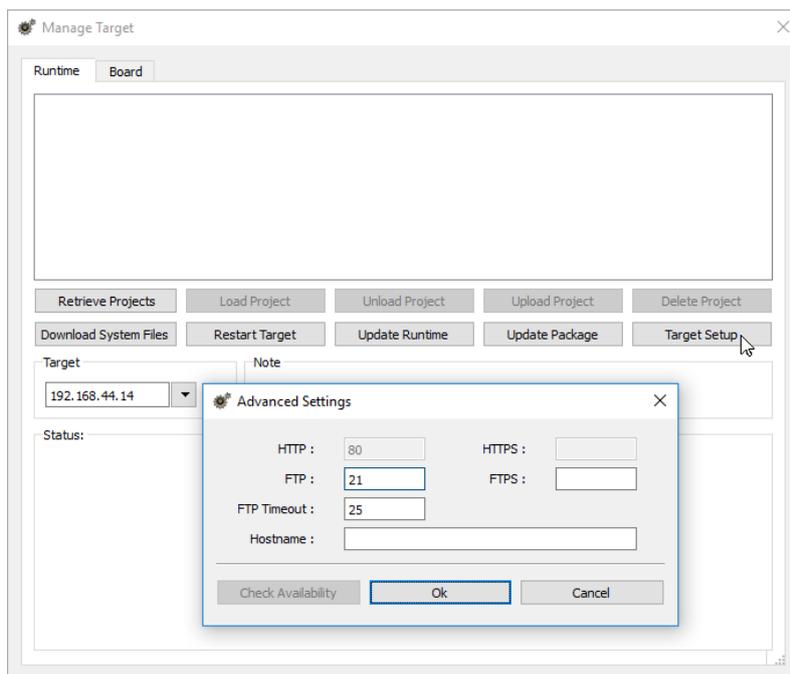
使用“高级设置”选项, 您可以定义要使用的端口, 但通常情况下, 您无需输入此信息, 因为 HMI 设备将提供用于面板内部的端口。



更改 HMI 设备连接设置

路径: 运行 > 管理目标

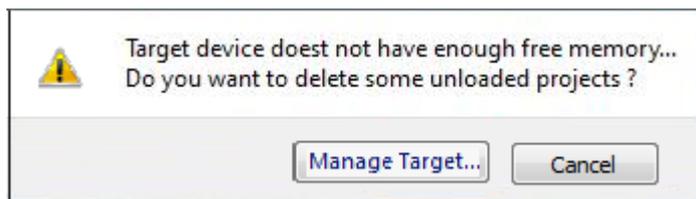
1. 单击目标设置: 高级设置对话框显示。HMI 设备的 HTTP 连接默认端口是端口 80。



2. 为 HMI 设备设置正确的 HTTP、FTP 或 HTTPS、FTPS 端口。(这些是系统用于连接 HMI 设备的端口, 当默认端口被其他服务或应用程序使用时或本地网络需要特殊设置时, 需要修改。)
3. 指定**主机名**以便在有多台设备可用的网络中轻松识别每一台设备。所有设备的默认主机名是“HMI”。
4. 单击**下载系统文件**。下一次下载 HMI 设备中的新的端口将被使用, 新的主机名将在下拉列表显示。

管理大项目

如欲成功下载, 项目大小至少要比可用内存小 **2MB**。如果不是的话, HMI 设备无空余闪存, 会显示警示信息。



要释放更多的存储空间:

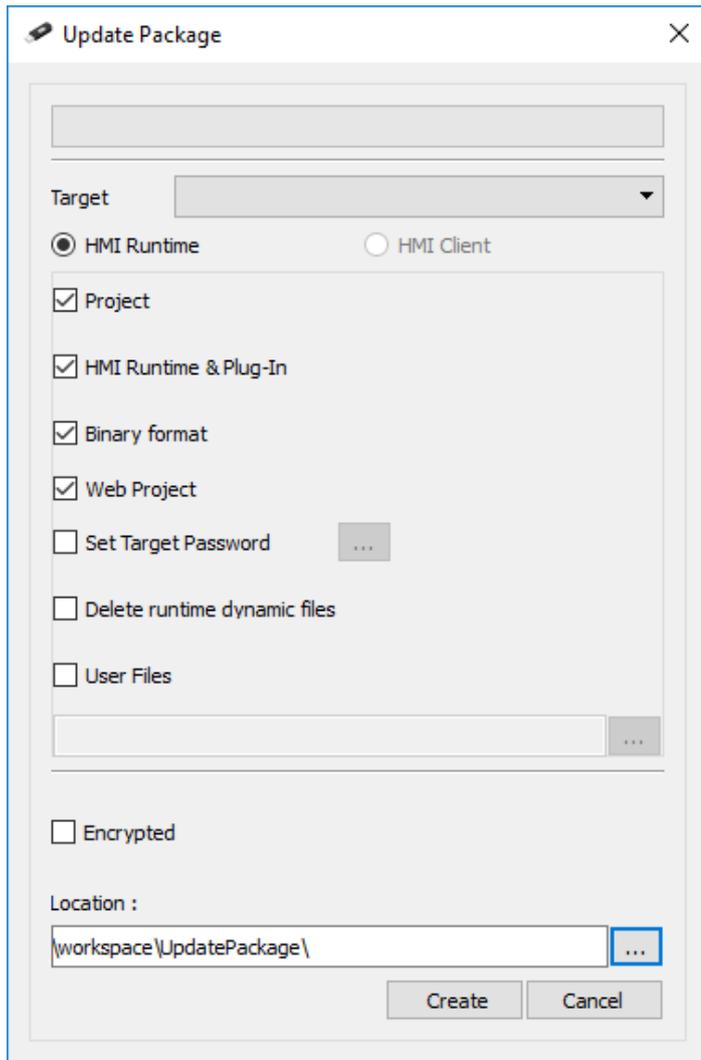
1. 单击**管理目标**。
2. 删除您不需要的项目, 腾出更多的空间。

更新包

更新包可创建 UpdatePackage.zip 文件以使用 U 盘安装或更新 HMI 设备中的应用程序。

创建一个更新包

路径: **运行 > 更新包**



选项	说明
目标	HMI 设备类型。如果项目已打开，会被自动选择。
应用程序选择器	选择要插入在 UpdatePackage.zip 中的应用程序 <ul style="list-style-type: none"> • HMI Runtime • HMI 客户端(仅在 Linux 设备上可用)
项目	向更新包添加打开的项目。
HMIRuntime& 插件程序	HMIRuntime 被添加到更新包。如果项目打开，所需的插件也被添加到更新包。
二进制格式	下载使用二进制格式的文件。
Web 项目	将 LRH SW 页面下载到 HMI 设备。
设置目标密码	设置密码来执行关键任务(例如:项目下载 / 上传、板管理) 请参阅"保护 HMI 设备使用"在本页 513。

选项	说明
删除 runtime 动态文件	当选中时，所有动态文件将被删除且 FRAM 会被清除。
用户文件	选择要复制到 HMI 设备的 QTHM 文件夹上的文件。最大 5MB
加密	启用对更新包的加密，这样只能通过 HMI Runtime 来解压。
位置	更新包的位置。



重要: 使用 HMI Runtime 应用程序创建包时，始终包括项目和 Runtime。如果需要在最新的 Runtime 版本使用旧的项目，首先转换项目。请参阅“[安装应用程序](#)”在本页 2 了解详细信息。

用户文件位置示例

计算机：

C:\Users\Username\Desktop\myFolder

- *subFolder1/file1*
- *subFolder1/file2*
- *file3*
- *file4*

Linux 设备：

/mnt/data/hmi/qthmi

- *subFolder1/file1*
- *subFolder1/file2*
- *file3*
- *file4*

加载一个更新包

路径: 从上下文菜单 > **更新**

1. 假设您已将包存储在 USB 驱动器的根文件夹中，请从计算机上拔下驱动器，将其插入 HMI 设备，手指按住屏幕几秒钟以显示上下文菜单并选择**更新**。
2. 系统将会检查USB驱动器根目录中的更新包是否存在，并请求确认进行更新。



3. 选择**自动选择最佳匹配**然后单击**下一步**:程序自动完成。或者使用浏览器按钮选择要使用的文件。

Runtime 加载器

HMI设备出厂时没有 runtime。

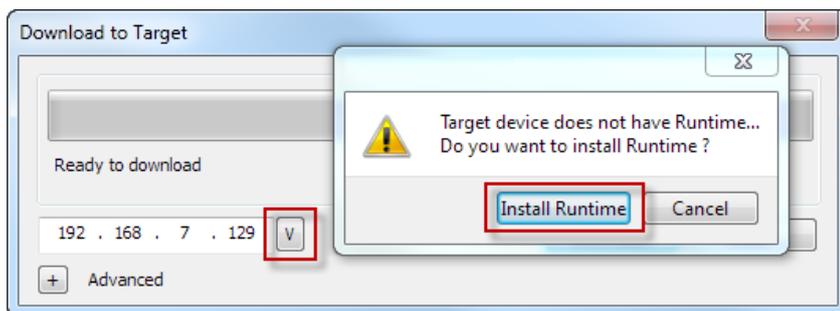
在您首次启动设备时，“Runtime 加载器”窗口显示(请参阅“系统设置”在本页 527 了解详细信息)。



Runtime 加载器的出现取决于设备操作系统，可能不是所有的设备都可用。旧版本的 HMI 设备可能不含 Runtime 加载器。如果需要进一步信息请联系技术支持。

从 LRH SW 中安装 Runtime

当您下载项目时，Runtime 如果需要会自动安装。



请参阅“向 HMI 设备传输项目”在本页 91 了解详细信息。

1. 单击**安装 Runtime**: 自动运行。

从 USB 驱动安装 Runtime。

1. 按照“更新包”在本页 94 中所述准备更新包
2. 将 USB 驱动器插入设备中，并按照针对此类设备的说明操作(请参阅“系统设置”在本页 527 了解详细信息)



注：旧版本的 HMI 设备可能不支持自动安装 Runtime。联系技术支持以获得更多信息。

上传项目

路径：运行 > 管理目标

您可以将项目从 Runtime 复制到运行 LRH SW 的计算机上。

1. 在 **Runtime** 选项卡中，从 **目标** 下拉列表中选择设备 IP 地址。

The screenshot shows a software interface with a 'Target' dropdown menu. The selected target is '192.168.40.28'. A dropdown menu is open, showing a list of targets: 'winxp-client1@192.168.42.30', 'HMI@192.168.40.28', 'HMI@192.168.41.1', 'HMI@192.168.42.20', 'HMI@192.168.41.171', 'HMI*@192.168.6.7', and 'Advanced Settings'. There is also a 'Note' field and a 'Status:' label visible.

2. 单击**检索项目**：所有可用项目的列表将显示。
3. 选择要上传的项目
4. 单击**上传项目**



可以用密码保护上传过程。请参阅“保护 HMI 设备使用”在本页 513 了解详细信息。

5. 如有必要，输入密码。上传过程开始。

项目副本被保存在：

`C:\Users\username\Documents\LRH SW\workspace\Uploaded\RuntimeIPAddress\workspace\ProjectName`



注：如果上传操作失败，请检查运行 LRH SW 的计算机上的防火墙设置。

9 标签编辑器

标签是一个友好的名称，用于标识设备的内存位置。可以通过通信协议从外部设备读取或写入标签。

在标签编辑器中，您可以配置协议和要使用的标签列表。

通信协议	100
添加标签	101
导出标签	104
导入标签	104
标签查找和重命名	108
标签查找和替换	110

通信协议

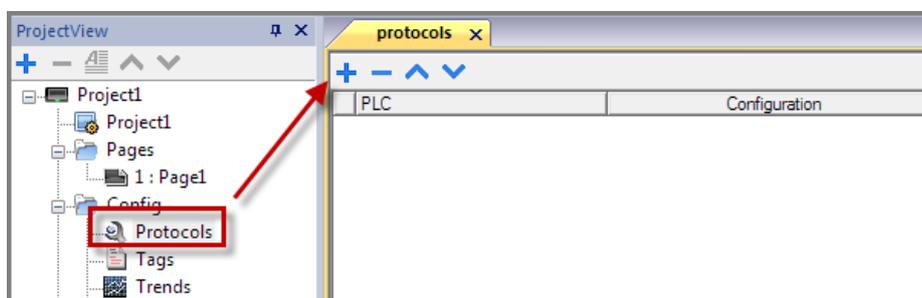
路径: 项目视图 > 配置 > 协议

设备通信驱动程序在**协议编辑器**中配置。可以添加功能限制表中指定的最大数量协议。变量和系统变量不看作协议。

 注: 您可以在相同的物理以太网端口运行不同的以太网协议, 但是不能使用同样的串口运行不同的串行协议。一些串行协议支持访问多个控制器, 但该选项在协议内部设置, 该协议仍然视为一项协议。

添加一项协议

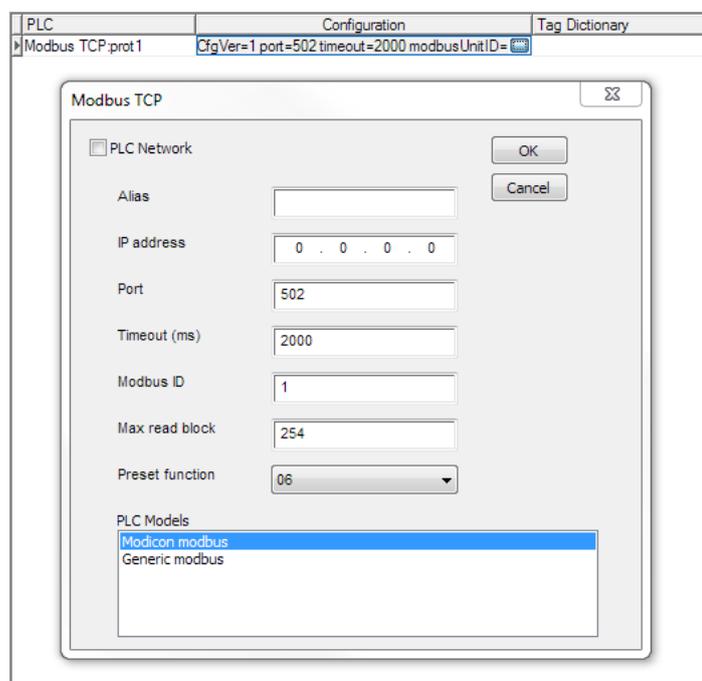
1. 单击 **+**。



2. 从 **PLC** 列表中选择协议并输入要求的值。

更改协议设置

要更改配置参数, 请单击 **配置** 列中的“浏览”按钮。



协议参数

单击**显示高级属性**图标以查看所有参数。

参数	说明
字典	为协议导入的标签 请参阅 "导入标签" 在本页 104 了解详细信息。
启用离线算法离线重试超时	请参阅 "自动脱机节点检测" 在本页 286 了解详细信息。
版本	在 LRH SW 中可用的协议版本，针对所选的 HMI 设备。

添加标签

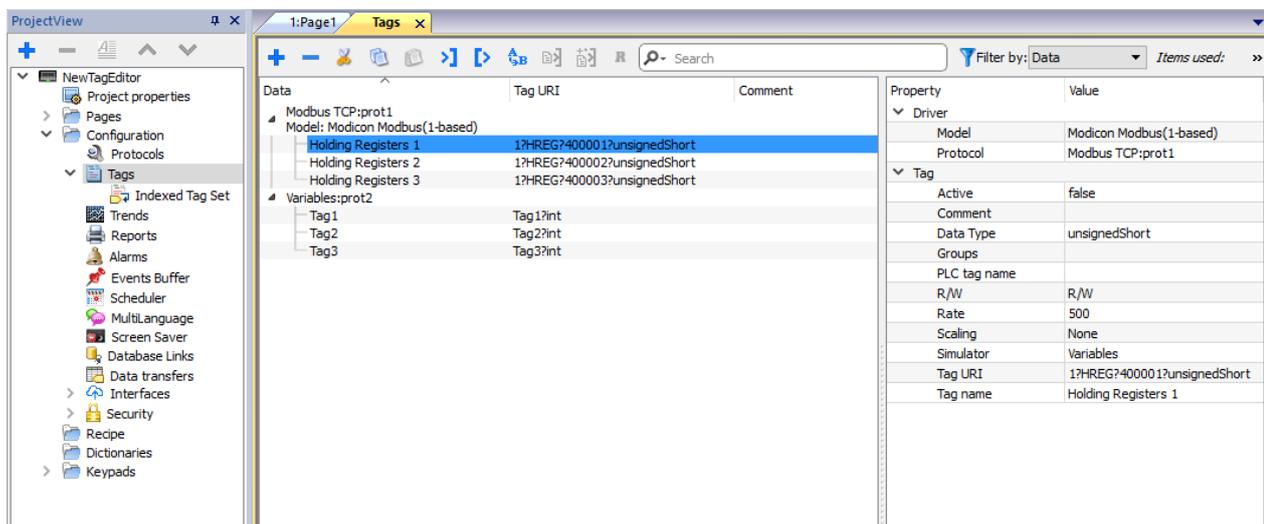
LRH SW 使用标签名称访问所有设备数据。设备上所有域和参考位置需要被指定一个标签名以在 HMI 项目中被使用。

标签编辑器可以用来创建和管理标签。定义标签后，可以将其附在控件属性中在项目中使用。

请参阅["附属"参数"](#)在本页 39 了解详细信息。

标签编辑器

路径: **项目视图 > 标签**



添加一个标签

1. 单击 **+** 输入要求的数据。
2. 从通讯协议地址对话框中选择地址
3. 如果要更改某些内容(例如标签名称), 请点击属性对话框中的字段



请注意, 如果选择了标签, 则添加标签命令 **+** 将使用所选标签的属性创建一个新标签。

标签属性

某些属性取决于所使用的协议。查看具体协议文件了解详情。

属性	说明
激活	<p>更新模式。</p> <ul style="list-style-type: none"> • false = 仅当 HMI 设备要求时从控制器中读取标签。 • true = 连续读取标签，即使显示的页面未要求。 <p> 重要：将该值保留为 false 以获得更高的通信性能。</p>
说明	标签描述
编码	字符串数据类型为编码类型 (UTF-8、Latin1、UTF-2 和 UTF-16)
组	与标签关联的组名称
PLC 标签名称	原始 PLC 标签名称，用于匹配用于 HMI 应用程序的标签和从 PLC 导出的标签。
R/W	<p>R/W 标签属性 (R/W、R 或 W)。</p> <p> 注：只写标签的内容始终被写，不能被读。当通信不在运行状态时，这些标签的内容可能在程序中不可用。</p>
率	<p>标签刷新时间。默认：500ms。</p> <p>当刷新率设置为“手动”时，HMI 设备将不会在后台从远程设备自动读取标签。只有通过“ForceReadTag(强制读取标签)”动作明确要求或在 JavaScript getTag() 中使用“forceRefresh”选项才能读取新标记并刷到数据库中。</p> <p> 警告：标签刷新率为最快刷新率。实际刷新率取决于：通信类型(串行、现场总线、Ethernet)、协议、交换的数据数量。</p>
缩放	<p>在数据库存储前应用在标签上的转换。</p> <ul style="list-style-type: none"> • 按公式 = 定义为线形转化。 • 按范围 = 定义为范围转化。 • 固定点 = 固定点缩放
模拟器	模拟期间标签的状态。一些简介可用。
标签地址	<p>控制器存储地址。</p> <p>要编辑，点击栏右侧获取对话框，您可以在这里输入地址信息。</p>
标签名称	<p>项目级别唯一的标签名称。识别 Runtime 标签数据库信息的主键码。</p> <p> 警告：不允许复制标签名称。</p>



请注意，允许在标签编辑器中选择多个标签并全部更改为同一属性(例如，将 10 个标签的刷新时间更改为 500，而无需一一更改各个标签中的刷新时间)。

管理标签名称

标签名称在项目层级必须唯一。如果同一个符号文件的相同的标签必须用于两个不同的控制器，使用别名 (Alias) 功能，对导入的标签添加字首，使其在项目层级唯一。



注：不是所有协议都支持“别名”功能。

管理标签群组

每一页上使用的标签被识别为一个群组的一部分，这样通信协议对连接的控制器的请求可以更快地处理：只有在显示的页面上的标签会从控制器获得。

缩放

使用标签缩放功能，可以调整标签值的大小，使之从 HMI 应用程序可见。

有可能进行配置

- 使用“按公式”或“按范围”模式进行线性转换
- 固定点转换

通常，HMI 内部使用的数据类型与从 PLC 设备继承的数据类型相同。使用转换时，请考虑是否可以更改 HMI 的数据类型而不会损失精度。

例如

如果您的 PLC 使用固定点上的整数管理带有两位十进制数字的值，则可以将缩放转换配置为下图，即将从 PLC 读取的值除以 100，并存储在 float 数据类型中。例如。PLC 整数 12345 将成为 HMI 设备内部的浮点值 123.45。

Fixed Point

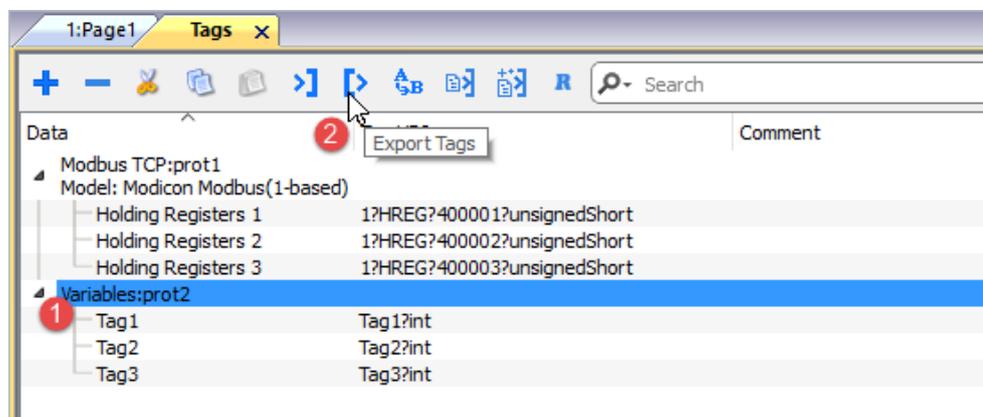
Number of decimal digits: 

Converted:

HMI data type:

导出标签

路径: 项目视图 > 标签



1. 为您想导出的标签选择协议。
2. 单击**导出标签**按钮: 所选协议的所有标签配置均将导出至一个 .xml 文件中。

您可以使用第三方工具编辑生成的 .xml 文件(例如, Microsoft Excel), 然后重新导入修改的文件(请参阅"导入标签" 向下了解详细信息)。

导入标签

简介

一些协议的标签以逗号分隔存储在文件中(.csv 或其他格式)。

导入包括两步：

1. 导入标签定义到词典
2. 从词典导入标签到项目

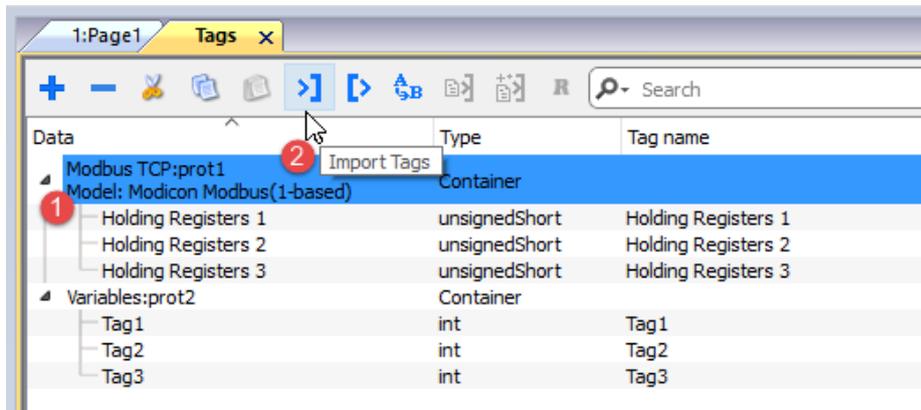


警告：导入时，标记名称中的特殊字符(例如“&”字符)可能会导致通信错误，将用下划线“_”字符替换。请参阅“Unicode 支持限制”在本页 296

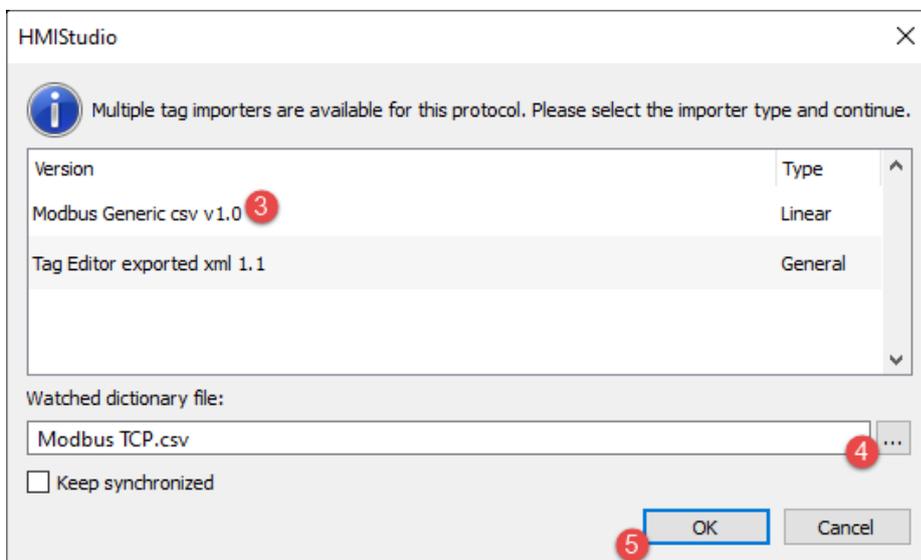
导入标签

要从外部文件导入标签：

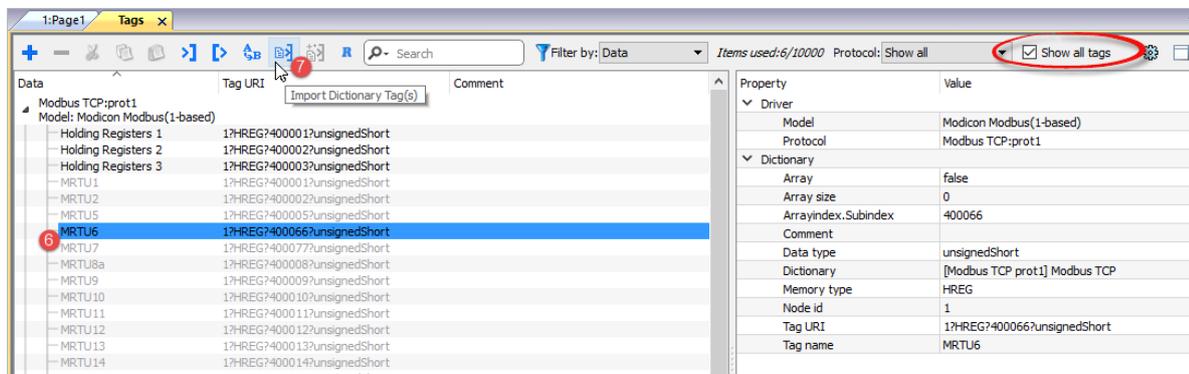
1. 在**项目视图**、**标签**中，从筛选器列表中选择协议。



2. 单击导入标签按钮：出现选择导入器类型的对话框。支持的导入器列表取决于所选协议。

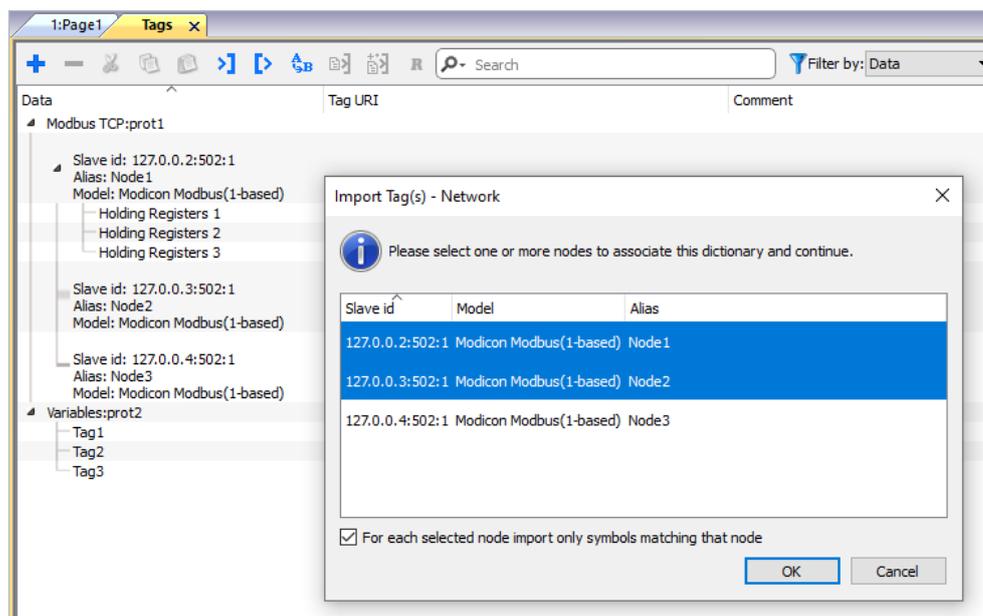


3. 选择要使用的导入器类型
4. 选择字典文件
5. 按 **OK** 键将词典文件附加到项目文件。词典中可用但未导入到项目中的标签为灰色，并且仅当选中“显示所有标签”复选框时可见。



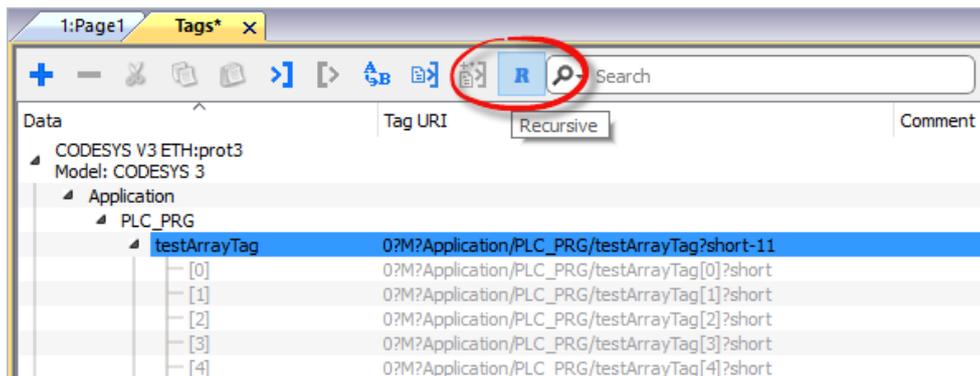
6. 要导入标签，请选择一个或多个标签或一个节点(仅用于分层视图)
7. 单击导入标签按钮：标签将导入到项目中并以黑色列出。

当项目被配置为使用协议网络时，您还必须选择要在其中导入标签的协议节点。您可以在多个协议上导入相同的标签。当标签文件包含节点信息时，您可以选择使用这些信息来过滤标签并只导入与所选节点匹配的标签。



递归

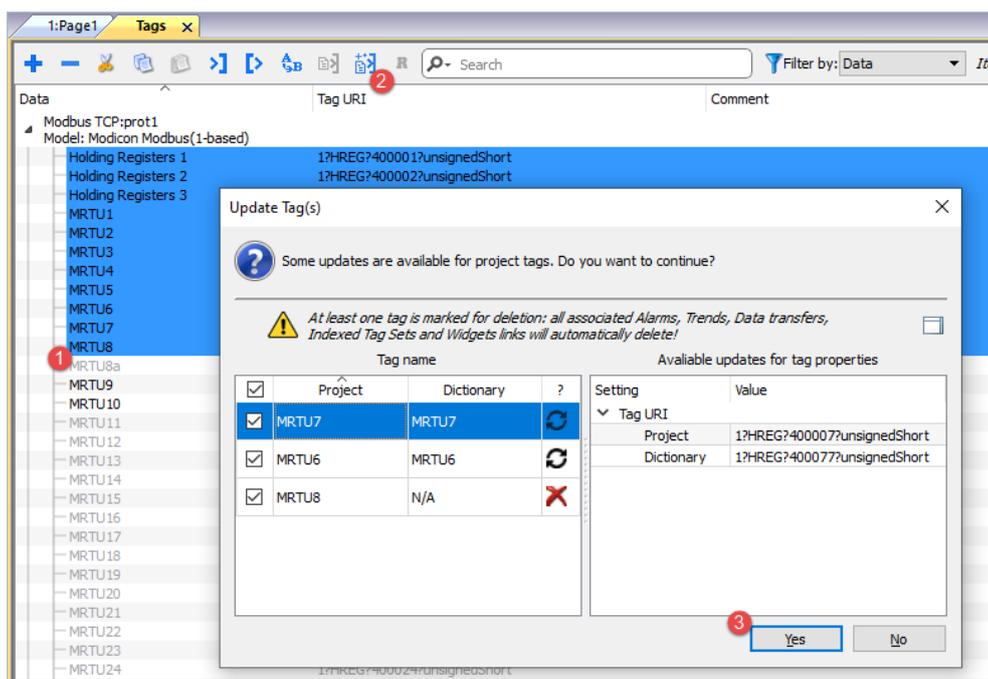
递归是一个切换按钮。如果选择此选项，在导入数组标签时，所有数组元素也会导入到单独的标签中。



更新导入的标签

检查字典文件并更新导入的标签：

1. 选择您要检查的标签
2. 如果发现了某些更改，将启用更新图标，单击该图标，将显示带有发现差异列表的“更新标签”对话框。
3. 取消选中您不希望更新的标签，然后单击“OK”确认



这些标签需要更新。显示项目和字典之间的差异列表。



这些标签在库中不再可用。如果更新，这些标签将从项目中移除。

保持同步

如果您要在没有用户干预的情况下自动检查 LRH SW 并更新文件词典中的标签，请选中“保持同步”复选框。

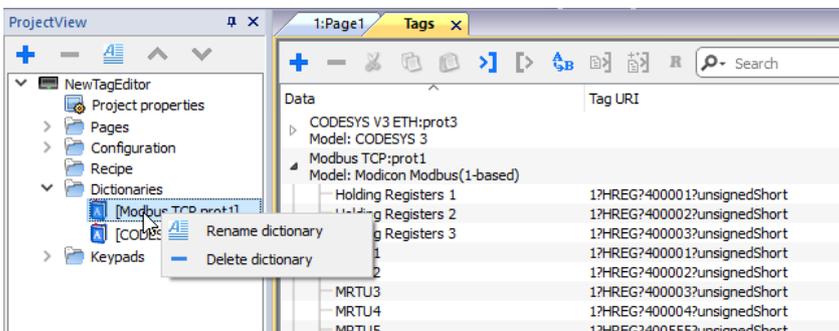


字典

路径: 项目视图 > 词典

词典是为特定协议从标签编辑器中导入的标签列表。根据协议类型的不同, 标签以线形或层级视图显示。

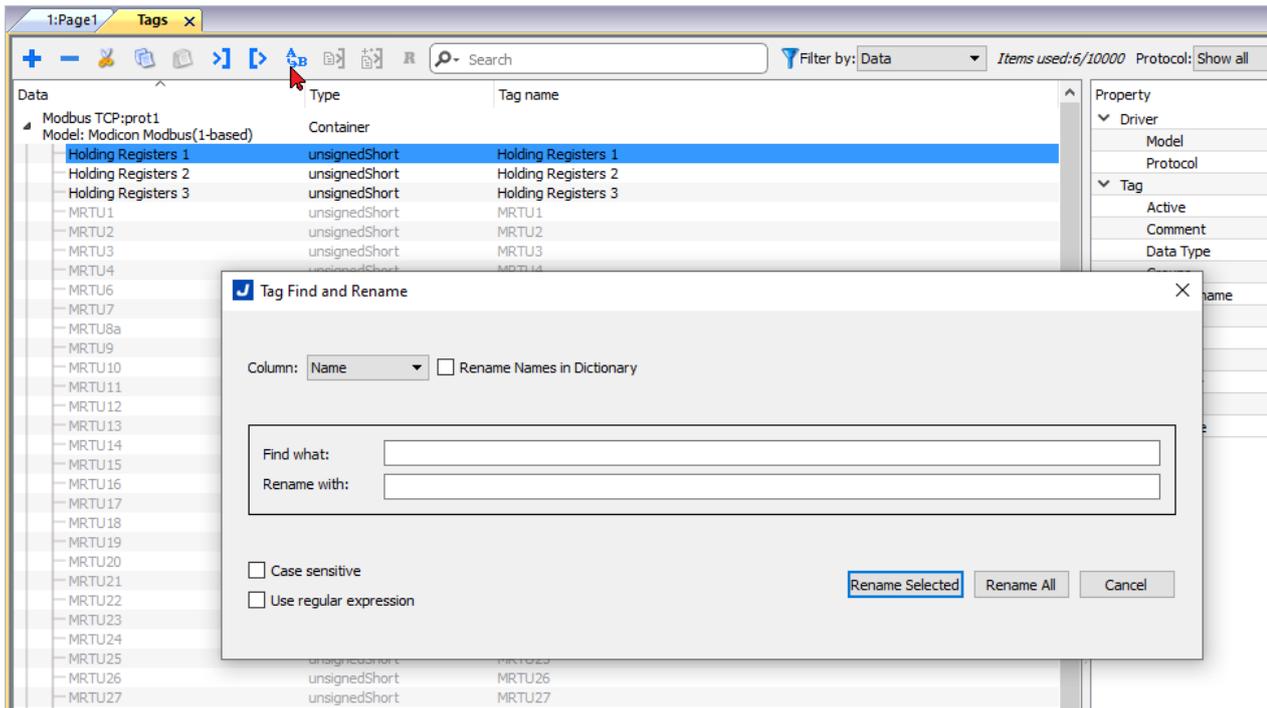
要删除字典, 请右键单击字典名称。



标签查找和重命名

标签查找和重命名功能将对整个项目中的标签引用执行重命名。

请注意, 此功能不仅可用于重命名标签, 还可在标签编辑器的每列中更改值



参数

栏	选择要使用查找和重命名操作修改的列
重命名词典中的名称	<p>标签的名称重命名将扩展到对内部词典标签的名称执行重命名。必须将该词典替换为包含重命名标签的另一个词典时，此参数有用</p> <p> 此参数仅在所选列为“名称”时才可用</p>
查找项	要搜索的字符串
重命名为	要替换的字符串
区分大小写	考虑大写和小写字母
使用正则表达式	<p>在搜索/替换模式中启用正则表达式</p> <p>有关正则表达式的其他详细信息，请参阅 https://en.wikipedia.org/wiki/Regular_expression。</p> <p> 启动正则表达式时，“查找项”参数将不提供预定义值，而仅提供自由文本处理。</p>

重命名所选项

仅对所选标签执行重命名

全部重命名

对整个标签数据库执行重命名



**Java 脚本和自定义控件中使用的引用未更新。
此命令不支持撤销**

正则表达式示例

使用上图中的标签列表。

如果要为所有标签添加前缀，无需使用正则表达式：

查找项：	MRTU(.*)
重命名为	PLC01_MRTU

但是如果添加后缀，需要使用正则表达式：

查找项：	MRTU(.*)
重命名为	MRTU\1_PLC01

其中

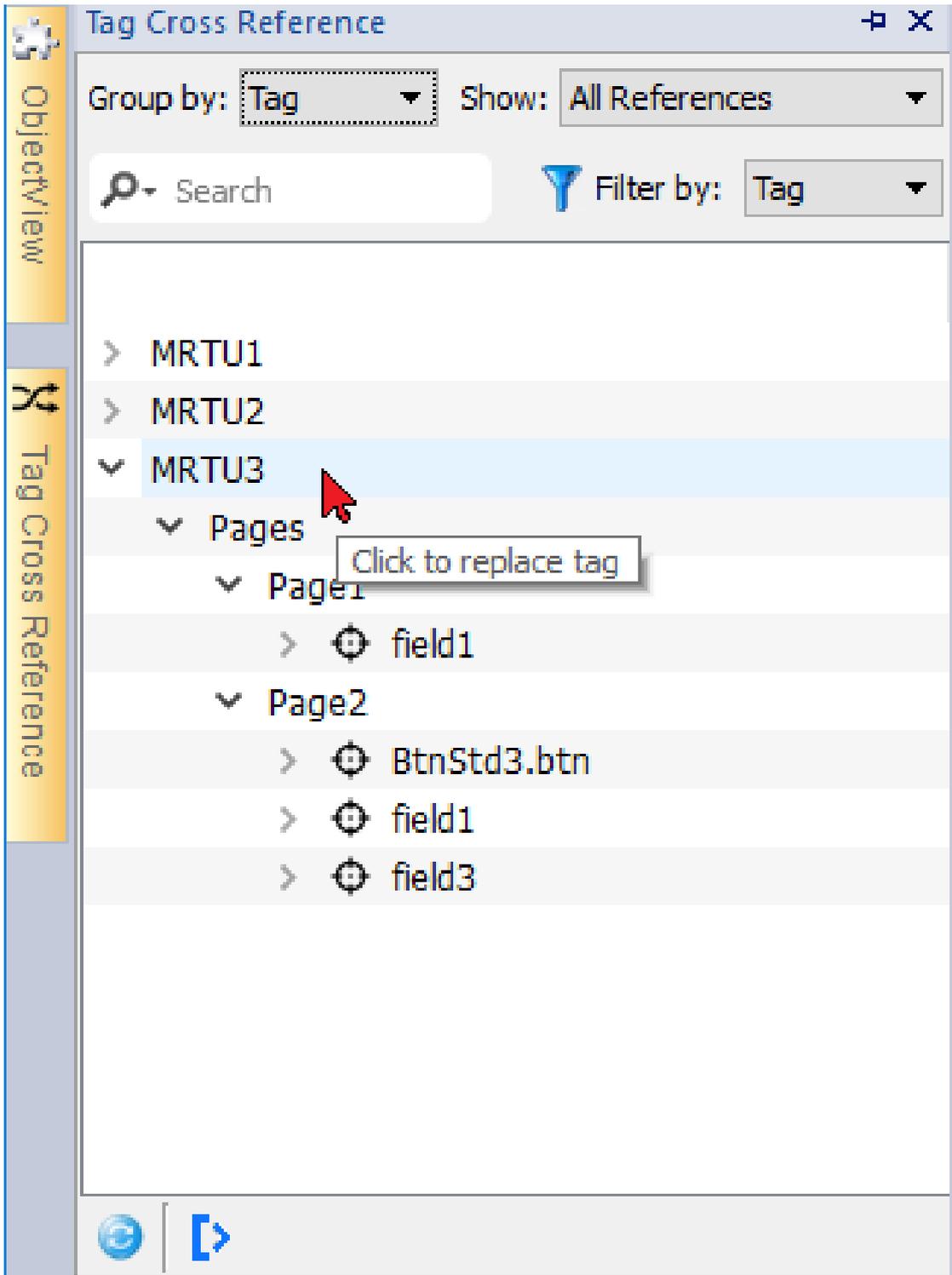
- (.*)** 指字符的任何序列
- \1** 是在搜索字符串中找到并由 (...) 包围的字符的第一个序列副本

标签查找和替换

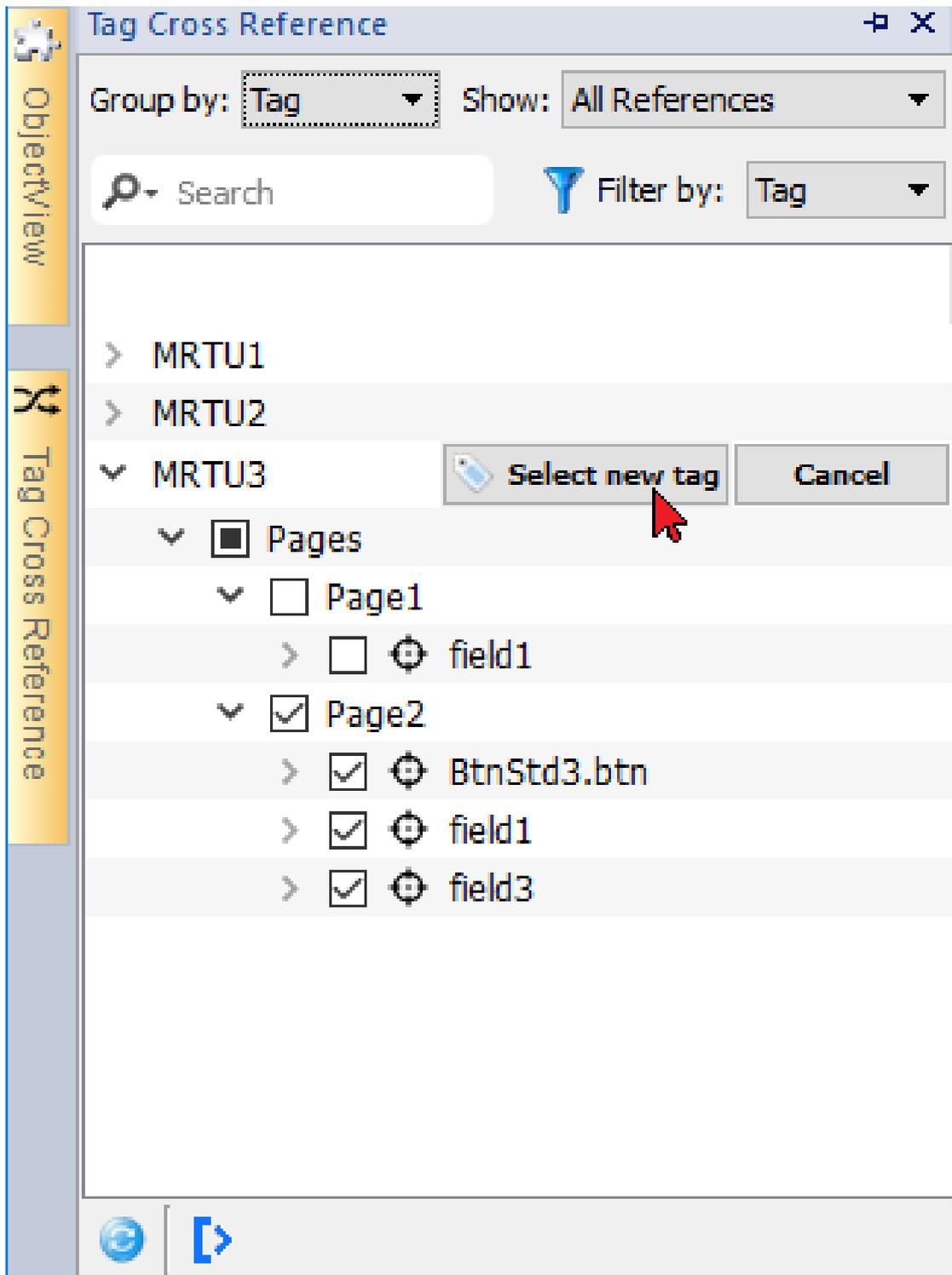
使用此功能，您可以搜索项目中出现的所有标签并将其替换为另一个标签。

 标签替换仅适用于以黑色显示的协议标签，不适用于系统变量、别名和配方标签。(不同的颜色的含义，请参阅 ["打开标签索引窗格" 在本页 120](#))

从“标签交叉引用”视图中，单击要替换的标签



使用复选框选择要应用替换的位置，
然后单击“选择新标签”按钮以替换所选对象的数据链接或按“取消”以中止操作。



可能未列出 **Java** 脚本和自定义控件中使用的引用。
此命令不支持撤销

10 索引寻址

索引寻址使您可以根据一个标签的值选择一个标签集。这是非常有用的，例如，要使用相同的图表使不同来源的数据可视化，用户要做的就是从列表选择要监控的源。

创建一个索引地址集	114
使用页面中的索引标签集	117

创建一个索引地址集

场景

在这种场景下，环境数据从四个房间收集，每个房间都配备了温度、压强和湿度感应器。可用数据如下：

Room Number	温度	压力	湿度
1	1号房间-温度	1号房间-压强	1号房间-湿度
2	2号房间-温度	2号房间-压强	2号房间-湿度
3	3号房间-温度	3号房间-压强	3号房间-湿度
4	4号房间-温度	4号房间-压强	4号房间-湿度

使用索引地址功能，您可以用一个单一的表格形式整理 HMI 设备中的所有数据。

三个不同感应器的数据可以在一个页面显示，房间号用做选择器(组合框)来选择正确的标签集。

Room 1

Temperature (°C)	21
Pressure	1
Umidity (%)	75

如何创建一个索引标签集

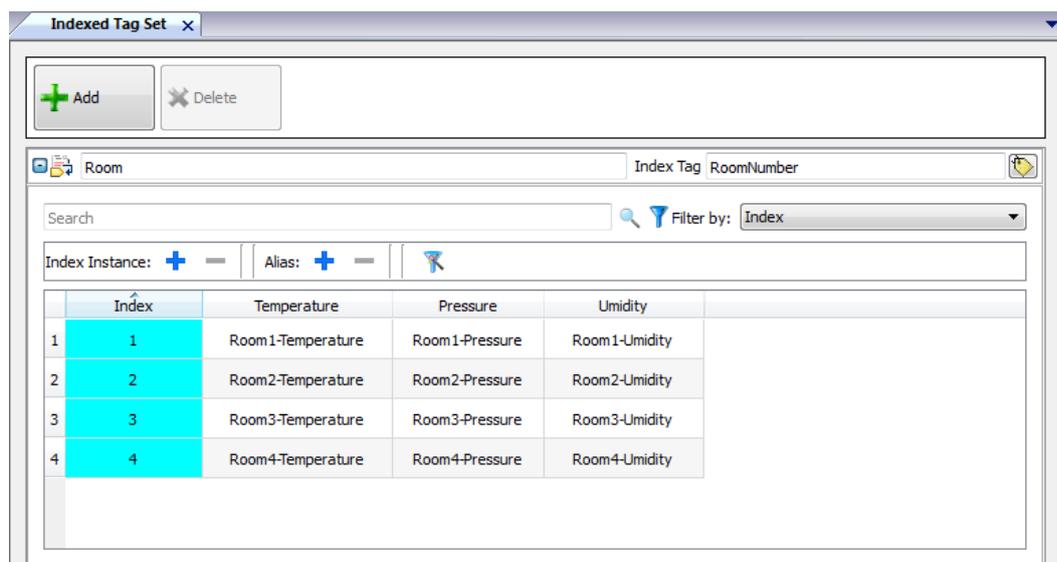
路径: 项目视图 > 标签

要实现这一点，您需要创建一个索引标签集。

1. 在标签编辑器中，定义协议和标签。定义为每个数据索引的标签，在这个例子中，您必须为每个房间的每个感应器创建一个标签。

Name	Group	Driver	Address
Room1-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400001 unsignedShort
Room1-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400002 unsignedShort
Room1-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400003 unsignedShort
Room2-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400004 unsignedShort
Room2-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400005 unsignedShort
Room2-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400006 unsignedShort
Room3-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400007 unsignedShort
Room3-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400008 unsignedShort
Room3-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400009 unsignedShort
Room4-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400010 unsignedShort
Room4-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400011 unsignedShort
Room4-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400012 unsignedShort

2. 创建一个标签，用做索引标签。在该示例中，您创建一个“RoomNumber”标签，可以是使用变量协议的 unsignedInt 类型。
3. 从项目视图中，选择配置 > 标签，双击索引标签集：“索引标签集”编辑器显示。
4. 点击 + 添加一个索引标签集。在示例中，您将称之为“Room”。
5. 选择标签“RoomNumber”用做房间编号选择器。
6. 为每组数据创建一个索引实例。在该例子中，每个房间一个。
7. 为每种数据类型创建一个别名，并相应重命名表格各列。在该例子中命名为“Temperature”、“Pressure”和“Humidity”。
8. 双击每个小格，与正确的标签关联。



注：“索引标签”数据类型可以是数字、字符串或任何简单的数据类型。

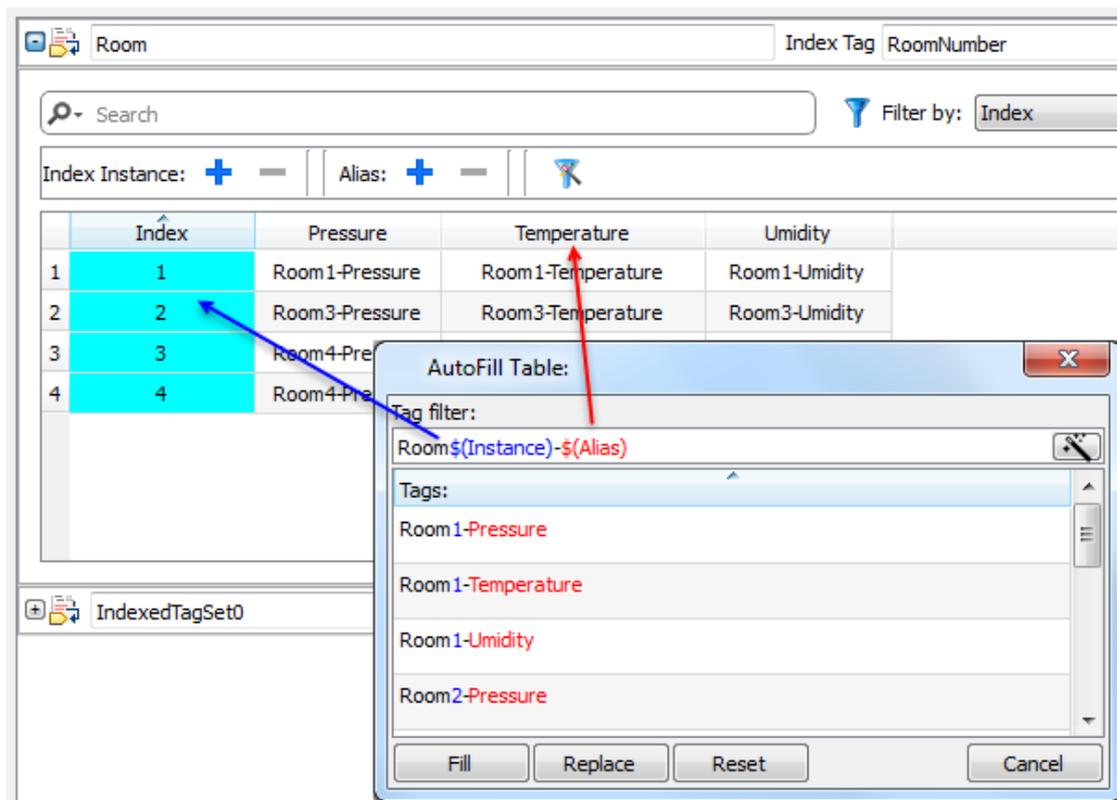


注：要引用数组数据类型，请使用数组索引 = -1

自动填入功能

索引标签集表格可能变得非常复杂，填入表格可能出错。启动自动填入功能，确保别名 (Alias) 输入正确。

单击 启用自动填充功能：自动填充表格显示。



该功能使用常规表达用标签填入表格，尽量与筛选器匹配，关键词\$(示例)将被指定的索引值 (Index) 代替，关键词\$(别名) 将被指定的别名 (Alias) 标签代替。

有关正则表达式的其他详细信息，请参阅 https://en.wikipedia.org/wiki/Regular_expression。

自动填入的例子

“Room\$(Instance)-\$(Alias)”将匹配所有标签名称：

Room1-Temperature,

Room1-Pressure,

Room1-Humidity,

Room2-Temperature,

...

“Room0*\$(Instance)-\$(Alias)”将匹配所有标签名称：

Room1-Temperature,

Room01-Pressure,

Room001-Humidity,

Room2-Temperature,

Room02-Pressure,

Room002-Humidity,

...

自动填入表格元素

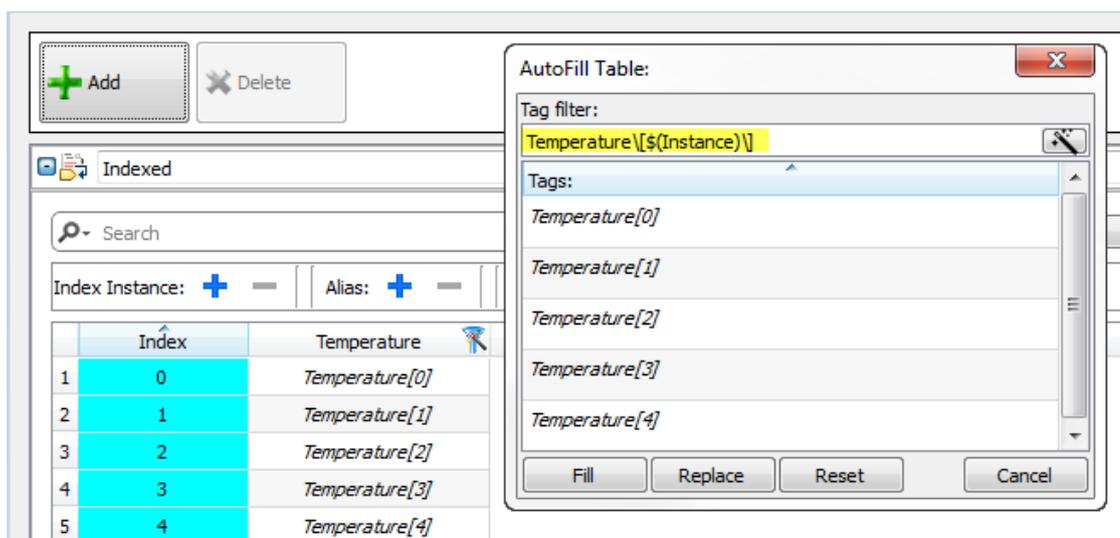
元素	说明
填充	使用设置的筛选器在标签表格中填入缺少的条目(如果有的话)。例如,当新的示例或新的别名 (Alias) 被添加,您可以使用该选项填入新的条目。
更换	用自动填入表格提供的条目代替表格中的条目。
复位	复位标签筛选器为空白,无自动填入。
	为项目建议一个可用的筛选器表达。



注:筛选器另存为项目优先项,可以对整个表格或一列设置。一旦筛选器为一列进行设置,表格筛选器将被忽略。因此您可以有选择地更改筛选器,以处理特定的别名 (Alias)。



注:要引用一个数组的元素,请使用 \ 字符(数组标签通过斜体字区分)。



使用页面中的索引标签集

创建索引标签集后,可以使用它为 HMI 设备创建页面,如本例中所示。

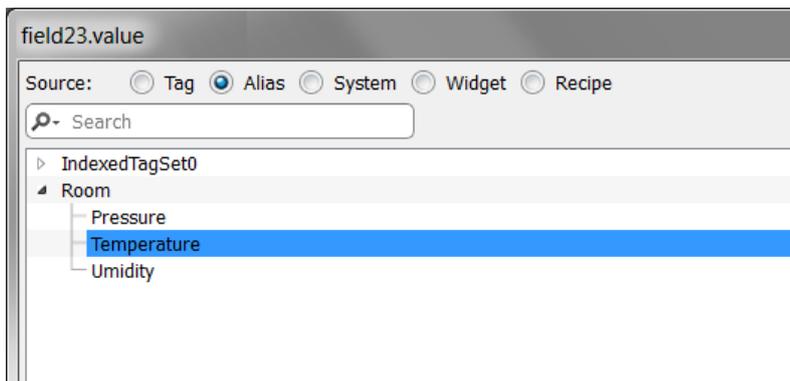
Room 1	
Temperature (°C)	21
Pressure	1
Umidity (%)	75

要创建该页面：

1. 创建一个页面并添加一个组合框、三个标签和三个数字栏位。
2. 为房间号创建的索引标签用于组合框，本例中为“RoomNumber”。这将会是房间号的选择器。
3. 为组合框创建一个列表。在该例子中，使用如下列表。

索引	字符串列表
0	Room Number
1	Room 1
2	Room 2
3	Room 3
4	Room 4

4. 为每个数字字段值附上相应的别名变量 (**Room > Temperature, Room > Humidity, Room > Pressure**)



11 标签索引

标签交叉索引窗格显示根据位置和用途组织的、在当前项目中使用的标签名称列表。

在这个窗格您可以：

- 验证每个标签使用的位置(报警、页面、配方、计划任务、趋势等等)
- 识别不可用的标签参照(标签编辑器中未被定义的标签参照)
- 识别未在项目中使用的标签



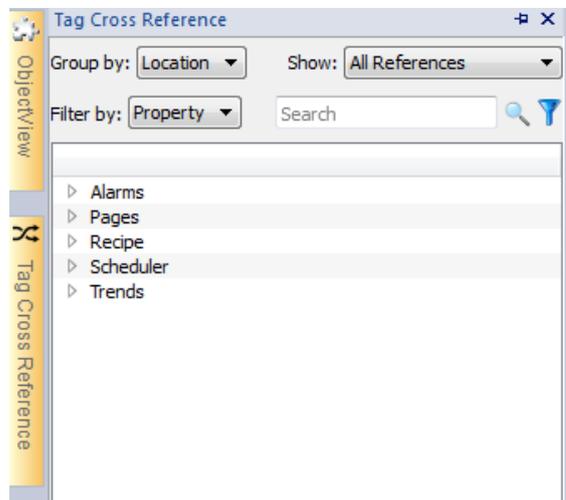
注：“标签交叉索引”窗格未列出 JavaScript 代码中使用的全部标签。

在标签索引窗格更新数据	123
-------------------	-----

打开标签索引窗格

路径: 视图 > 工具栏和连接窗口 > 标签交叉索引

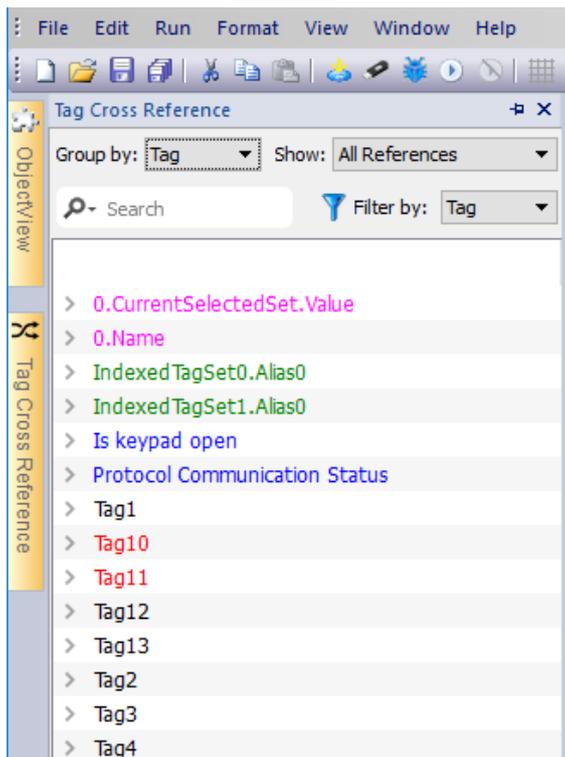
单击**标签交叉索引**选项卡以打开“标签交叉索引”窗格。



色彩的含义

黑	协议标签
洋红色	配方标签
蓝色	系统变量标签
深绿色	别名标签
红色	无效标签

示例:



在标签索引窗格操作

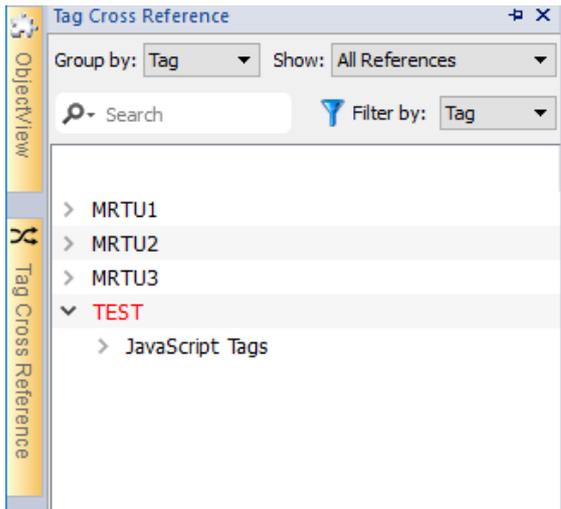
标签索引窗格提供一组标准功能。

元素	功能
分组方式	根据位置(报警、页面、趋势等)或标签名称分组标签
显示	过滤标签并显示： <ul style="list-style-type: none"> • 所有参考:所有标签 • 无效标签参考:标签编辑器中未定义的标签。 • 未用标签:在标签编辑器中定义但未在项目中使用的标签。
搜索栏	应用过滤器显示一定数量的标签
筛选方式	根据位置、标签或属性筛选标签。

对列出的标签导航，找到它们在项目中什么位置被使用。

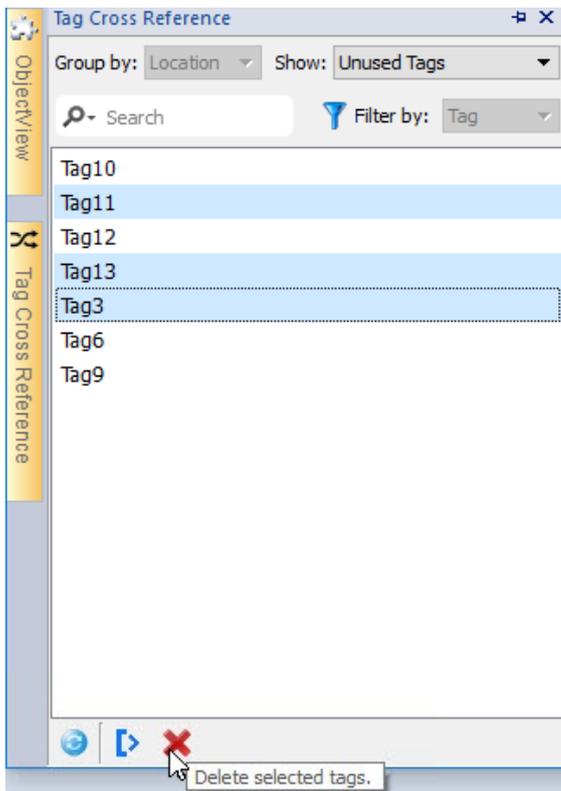
双击一个标签，打开编辑器或该标签被使用的页面。

无效标签参考将以红色列出：



删除未用标签

从未用标签视图中选择一个或多个标签并从标签编辑器中删除。要选择多个标签，请单击多个标签并使用 SHIFT 或 CTRL 键。



请注意，最终可能找不到 **JavaScript** 中引用的标签(取决于如何写入代码)。即使在自定义控件中使用标签引用不可行，但从“标签交叉引用”引擎中可能检测不到在自定义控件中引用的标签。

在标签索引窗格更新数据

手动更新

默认情况下，显示在标签索引窗格中的信息必须手动更新。要实现这一点，请单击刷新按钮 。当需要刷新时会出现一个警示标志。

自动更新

路径: **视图 > 属性**

从 LRH SW 属性页面启用“标签交叉索引”窗格自动更新。



选中自动更新选项。

导出数据

在标签索引窗格显示的数据可以导入到 .csv 文件中。

数据根据在窗格中的分组情况在导出的文件中组织。

分组方式:	文件格式
位置	源, 源磁盘, 控件-ID, 属性, 标签
标签	标签, 源, 源磁盘, 控件-ID, 属性



注: 导出操作中使用的分隔符取决于计算机的区域设置。

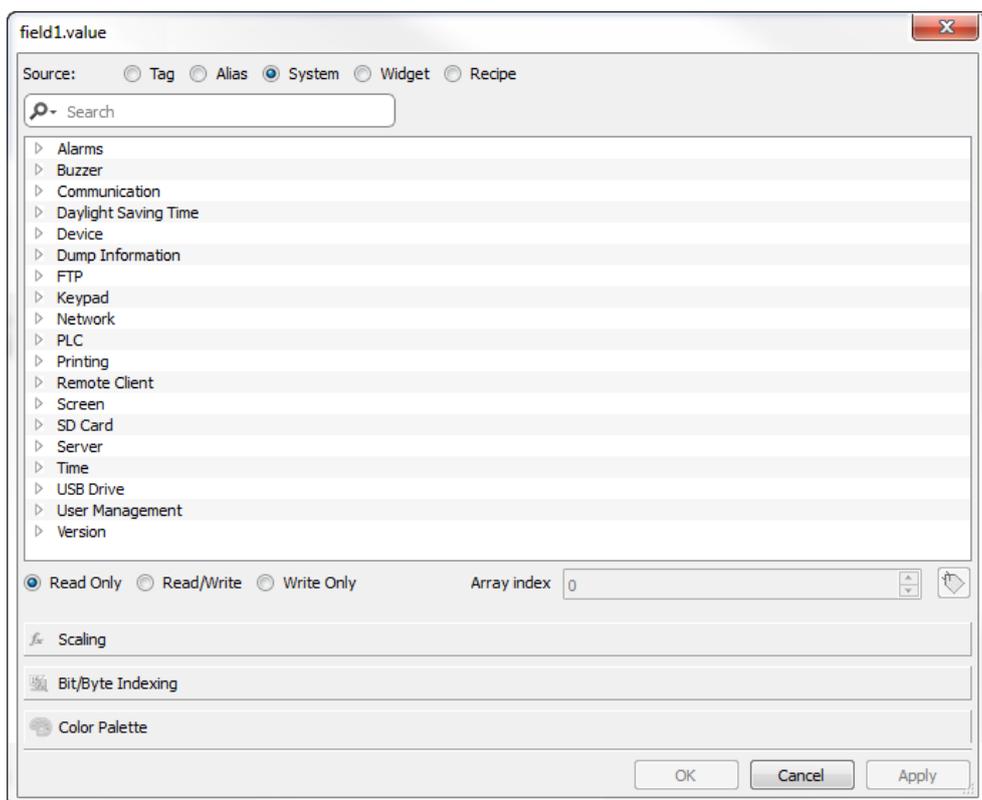
12 系统变量(附加至)

路径:源 > 附加到

系统变量是特殊的包含关于 HMI Runtime 信息的标签。



注:系统变量还可以作为协议编辑器中的标准协议。当必须在系统变量和设备标签之间传输数据,或为系统变量选择自定义刷新率时,使用系统变量作为协议。



报警变量	127
响铃变量	127
通信变量	128
夏令时间变量	128
设备变量	129
转储信息变量	130
FTP客户端变量	131
键盘变量	131
网络变量	131
打印变量	132

远程客户变量	133
版本变量	134
屏幕变量	134
SD 卡变量	134
服务器变量	135
时间变量	135
触摸屏变量	135
USB 驱动变量	136
用户管理变量	137

报警变量

要求的型号的报警数量。

变量	说明	数据类型
未确认的报警	未确定报警时为真 (未启动未确定 <>0) 或 (未确定启动 <>0)	boolean 只读
启动的报警	启动至少一个报警时为真 (确认启动 <>0) 或 (已启动 <>0) 或 (未确认启动 <>0)	boolean 只读
丢失的报警事件数量	报警超过事件队列在 <i>engineconfig.xml</i> 文件中已定义队列长度。	int 只读
未确认启动的数量	报警条件不再运行;报警已确认	int 只读
未确认未启动的数量	报警条件不再运行;等待确认	int 只读
确认启动的数量	报警条件在运行状态;报警已确认	int 只读
启动报警的数量	报警处于运行状态;不需要确认	int 只读
未确认启动的数量	报警条件运行;等待确认	int 只读



注:出于兼容性原因,旧名称仍然有效,但不建议使用。

响铃变量

调整蜂鸣器行为。

变量	说明	数据类型
蜂鸣器设置	<p>0 = 已禁用 1 = 已启用(在出现任何触摸屏幕事件时,蜂鸣器发出声音) 2 = 蜂鸣器状态由蜂鸣器控制系统变量或由"项目属性"在本页 73内的触摸响铃属性控制</p> <p> 触摸屏幕响铃(设置 = 1)在 Linux 平台上不可用。请参阅可作为替代的“触摸响铃”属性。</p>	int
蜂鸣器控制	<p>0 = 蜂鸣器关闭 1 = 蜂鸣器打开 2 = 蜂鸣器闪烁</p>	int
蜂鸣器关闭的时间	当选择闪烁时,以毫秒表示关闭的持续时间。默认 = 1000。范围:100–5000。	int
蜂鸣器打开的时间	当选择闪烁时,以毫秒表示打开的持续时间。默认 = 1000。范围:100–5000。	int

通信变量

HMI设备和控制器的通信状态。

变量	说明	数据类型
协议通信状态	<p>总结通信协议的状态。</p> <p>0 = 无协议运行,协议驱动器可能未正确下载到 HMI 设备。 1 = 协议已加载并启动,无通信错误。 2 = 至少一个通信协议报告错误。</p>	int 只读
协议错误信息	<p>具有错误源的通信错误。</p> <p>For example: “[xxxx]”中“xxxx”是协议缩写,是错误源。 在多个错误源的情况下,多个缩写词出现。当无错误报表时,出现空白。</p>	ASCII 字符串 只读
协议错误数量	上一次复位以来的通信错误的数量。使用“复位协议错误计数”动作复位值,请参阅“ 系统动作 ”在 本页 190 。	int 只读

夏令时间变量

系统时钟上的信息。变量包含当地时间信息。标准时间(太阳时)和夏令时间(DST)可用。



注:所有变量均是只读的;您不能将其用于更新系统时钟。

变量	说明
标准偏移量	当标准时间设定时, 偏移量以分钟计算, 以 GMT 为准(例如: $-8 \times 60 = -480$ 分钟)。
标准周	标准时间开始的周(例如: 第一 = 1)。
标准月	标准时间开始的月。范围: 0–11。(例如: 十一月 = 10)。
标准日	标准时间开始的星期(例如: 星期天 = 0)。
标准时	标准时间开始的小时(例如: 02 = 2)。
标准分钟	标准时间开始的分钟(例如: 00 = 0)。
DST 偏移量	当 DLS 时间设定时, 偏移量以分钟计算, 以 GMT 为准。
DST 周	DLS 时间开始的周
DST 月	DLS 时间开始的月。范围: 0–11。
DST 日	DLS 时间开始的星期
DST 小时	DLS 时间开始的小时
DST 分钟	DLS 时间开始的分钟

设备变量

设备设置和操作状态信息。

变量	说明	数据类型
可用的系统存储	剩余可用 RAM 存储, 以字节为单位。	uint64 只读
背光时间	从设备生产起, 显示背光的激活小时数。	unsignedInt 只读
显示屏明亮度	返回并调整明亮度。 即使设为 0, 背光仍然打开, 背光时间计数器数值增加。 范围: 0–255	int
外部超时	超过无操作时间后显示屏背光自动关闭。当用户接触屏幕时, 背光自动打开。 0 = 打开背光(打开显示屏) 1..n = 超时, 以秒计算, 用于关闭背光(屏幕保护程序计时器)  上, 超时限值累计为分钟倍数(60, 120, 180 等)。	int

变量	说明	数据类型
	在 "HMI 设备地址" 在本页 525 寻找您的设备平台。	
闪存空余空间	内部闪存剩余空间	uint64 只读
制造商代码	辨别 HMI 型号的内部代码	unsignedInt 只读
系统字体列表	系统字体列表	string 只读
系统模式	Runtime 操作状态 1 = 正在启动 2 = 配置模式 3 = 操作模式 4 = 重启 5 = 关闭	int
系统开启时间	设备自生产起的系统通电时间(小时)。	unsignedInt 只读

转储信息变量

趋势和事件缓存复制到外部存储(USB 或 SD 卡)的过程状态。

变量	说明	数据类型
转储错误信息	在执行转储操作期间,如果出现任何错误,则将返回错误消息	string 只读
存储存档状态	0 = 初始默认状态 1 = 操作已触发 2 = 操作成功完成 3 = 操作出错	int 只读
存储配方状态	0 = 初始默认状态 1 = 操作已触发 2 = 操作成功完成 3 = 操作出错	int 只读
存储趋势状态	0 = 初始默认状态 1 = 操作已触发 2 = 操作成功完成 3 = 操作出错	int 只读

变量	说明	数据类型
复位配方状态	0 = 初始默认状态 1 = 操作已触发 2 = 操作成功完成 3 = 操作出错	int 只读
恢复配方状态	0 = 初始默认状态 1 = 操作已触发 2 = 操作成功完成 3 = 操作出错	int 只读

FTP客户端变量

在使用 FTP 动作时，系统将更新 FTP 客户端变量。

变量	说明	数据类型
FTP 当前命令	最近的 FTP 命令	string 只读
FTP 错误消息	最近的 FTP 错误消息	string 只读
FTP 进度	下载/上传进度 (0/100%)	short 只读
FTP 状态	最近的 FTP 命令的状态： <ul style="list-style-type: none"> • 0 = 闲置 • 1 = 活动 • 2 = 完成 • 3 = 错误 	short 只读

键盘变量

键盘状态。

变量	说明	数据类型
键盘是否打开	0 = 无键盘打开 1 = 键盘打开	int 只读

网络变量

设备网络参数。

变量	说明	数据类型
适配器参数	这是一种可以用来读取或更新网络适配器参数的 JSON 字符串	string
网关	设备主网络界面的网关地址	string 只读
IP 地址	设备主网络界面的 IP 地址	string 只读
Mac ID	设备主网络界面的 MAC ID	string 只读
状态	包含写入适配器参数内所需的最后操作的结果。在每次写稿操作完成后更新。 <ul style="list-style-type: none"> 空字符串代表无错误 最后错误说明 	string 只读
子网掩码	设备主网络界面的子网掩码	string 只读

打印变量

打印功能方面的信息。

变量	说明	数据类型
完成比率	当前打印任务完成的比率。 范围：0–100	只读
当前磁盘用途	存储 PDF 报表的文件夹大小，以字节为单位。 如果闪存已选为 <i>Spool</i> 多媒体类型，此值将与 <code>reportspool</code> 对应。	只读
当前任务	任务在进行的报表的名称 当前任务如下：图形报表 <ul style="list-style-type: none"> 图形报表的 [报表名称] 文本报表的 [文本首行] 	只读
当前 RAM 用途	用于处理当前任务的 RAM 的大小，以字节为单位	只读
磁盘定额	存储 PDF 报表的文件夹最大字节量	只读
图表任务队列大小	打印队列中可用的图表任务数量	只读
上一个错误消息	上一个返回错误的描述	string 只读

变量	说明	数据类型
RAM 定额	用于生成报表的 RAM 的最大大小, 以字节为单位	只读
状态	打印系统状态。 值: <ul style="list-style-type: none"> • idle • error • paused • printing 	string 只读
文本任务队列大小	打印队列中可用的文本任务数量	只读

远程客户变量

在远程客户端上, 以下系统变量可用于了解服务器(HMI 设备)是否可访问。

变量	说明	数据类型
连接状态	<p>0 = 客户端无法访问服务器客户端。与服务器连接中断。</p> <p>1 = 客户端可以访问服务器。与服务器连接处于活动状态。</p> <p> 这只是一个客户端侧变量。在 LRH SW HMI Runtime 或 LRH SW PC Runtime 上, 该变量始终为 0。</p>	int (32 bit) 只读

以下系统变量与传输文件到远程 HMI 设备相关联。

变量	说明	数据类型
从 HMI 下载时的错误消息	错误描述	ASCII 字符串 只读
HMI 下载进度百分比	下载进程 (0→100)	只读
HMI 下载状态	<p>0 = 闲置, 动作未使用或已完成</p> <p>1 = 正在下载文件</p> <p>2 = 错误</p>	int (32 bit) 只读
上传至 HMI 时的错误消息	错误描述	ASCII 字符串 只读

变量	说明	数据类型
上传至HMI的进度百分比	上传进程(0→100)	只读
上传至 HMI 状态	0 = 闲置, 动作未使用或已完成 1 = 正在上传文件 2 = 错误	int (32 bit) 只读

版本变量

操作系统和 Runtime 版本。

变量	说明	数据类型
主操作系统版本	主操作系统版本。	string
运行时版本	运行时版本。	string

屏幕变量

屏幕状态

变量	说明
Time remaining to unlock	解锁屏幕剩余时间(请参阅 锁定屏幕 动作, " 页面动作 " 在本页 178)
X Screen resolution	以像素显示横向屏幕大小
Y Screen resolution	以像素显示纵向屏幕大小

SD 卡变量

外置 SD 卡的信息

变量	说明	数据类型
SD 卡剩余空间	卡上的可用字节空间	uint64 只读
SD 卡名称	SD 卡名称	string 只读
SD 卡大小	卡槽中卡的字节大小	uint64 只读
SD 卡状态	0 = SD 卡已拔出 1 = SD 卡已插入	int 只读

服务器变量

服务器状态。



重要:所有变量均与服务器相关,而非 LRH SW Client。

变量	说明	数据类型
当前页面	当前页面的名称	string
当前项目	当前项目的名称	string
操作模式时间	设备启动操作模式如系统日期格式(毫秒)后经过的秒数。	uint64
项目加载时间	项目加载到 LRH SW HMI Runtime 上的日期,采用系统日期格式(毫秒)。	uint64

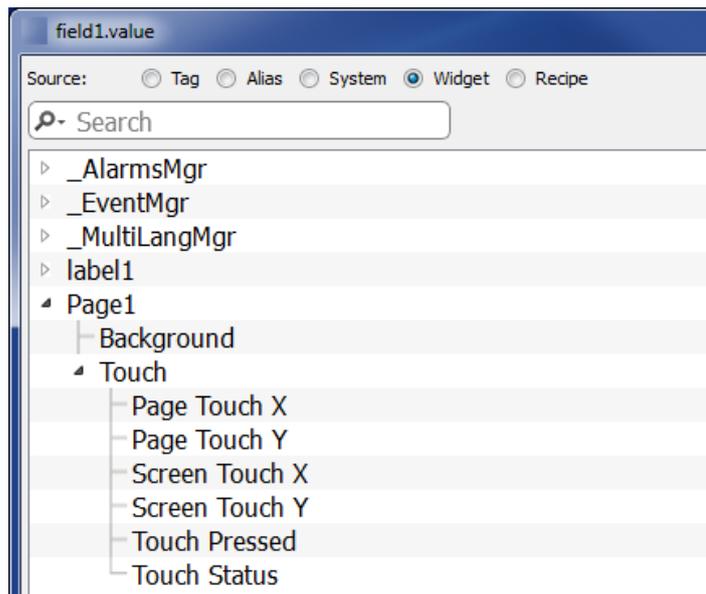
时间变量

The Time data type requires a special data conversion.

变量	说明	数据类型
月中日期	范围:1-31	int
星期几	范围:0=星期日,依此类推,6=星期六	int
小时	范围:0-23	int
分钟	范围:0-59	int
月	范围:1-12	int
次要	范围:0-59	int
系统时间	与 UTC 时间相同。也可设为此变量的日期/时间。	unsignedInt
年	当前年份	int

触摸屏变量

触摸屏上的指针状态和位置。它们是活动页面的属性,可以在**控件**部分中选择。



注：页面大小可以与 HMI 设备显示大小不同。

变量	说明	Java Script
页面触摸 X 页面触摸 Y	页面相关的指针位置	page.primaryTouch.x page.primaryTouch.y
屏幕触摸 X 屏幕触摸 Y	触摸屏相关的指针位置	page.primaryTouch.screenX page.primaryTouch.screenY
触摸按	0 = 未触按屏幕 1 = 已触按屏幕	page.primaryTouch.pressed
触摸状态	由触摸屏更改。此变量将 屏幕触摸 X 、 屏幕触摸 Y 和 触摸按 值串联起来(例如,“924,129,0”)。 该变量的主要用途是在有更改(x、y或点击)时,使用数据更新功能启动一个事件。	page.primaryTouchStatus

USB 驱动变量

连接到设备的外部 USB 驱动器的信息。

变量	说明	数据类型
USB 驱动器的剩余空间	可用字节空间	uint64 只读
USB 驱动名称	USB 设备的名称	string 只读
USB 驱动器的大小	USB 端口插入的设备的字节大小	uint64 只读
USB 驱动器的状态	0 = USB 驱动器已拔出 1 = USB 驱动器已插入	int 只读

用户管理变量

用户和群组的信息。

变量	说明	数据类型
客户用户名	登录到系统变量显示的客户端上的用户名。	string 只读
用户群组名称	当前登录用户的群组	string 只读
本客户端 ID	仅适用于 LRH SW Client。连接至同一服务器(例如:Runtime)的本地和远程客户端获得唯一的 ID。	short 只读
无活跃的远程用户	连接到服务器的 LRH SW Client 的数量	short 只读

JavaScript

这些变量可作为 `_SysPropMgr` 对象的属性从 JavaScript 访问。

示例:

```
var sysVar = project.getWidget( "_SysPropMgr" ); var UserName = sysVar.getProperty( "This Client User-Name" ); var UserGroup = sysVar.getProperty( "This Client Group-Name" ); var clientId = sysVar.getProperty( "This Client ID" ); var numClients = sysVar.getProperty( "No of Remote-Clients Alive" );
```


13 系统变量(协议)

“系统变量”通信驱动程序允许创建可指向系统信息的标签。



系统变量通信驱动程序不用作物理协议。
请参阅主手册**功能限制表**中的“物理协议数量”部分。

协议编辑器设置	140
标签导入	140
默认变量	141
保留内存变量	154
服务变量	160
PLCM09 变量	163

协议编辑器设置

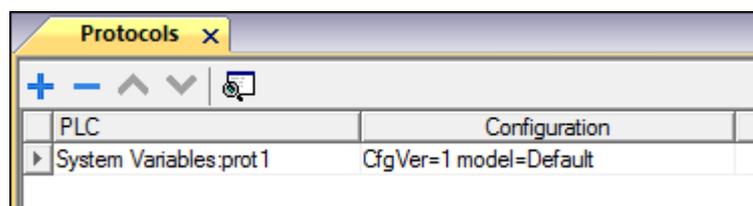
添加一项协议

配置协议：

1. 在**配置**节点中双击**协议**。
2. 要添加一个驱动程序，点击**+**：新的行被添加。
3. 从**PLC**列表选择协议。

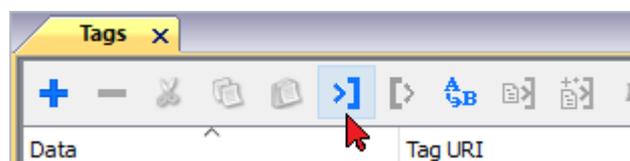
显示协议配置对话框。

从“PLC 模式”列表中，选择特定的系统变量类型。



标签导入

在标签编辑器中选择驱动程序，然后单击**导入标签**按钮启动导入程序。

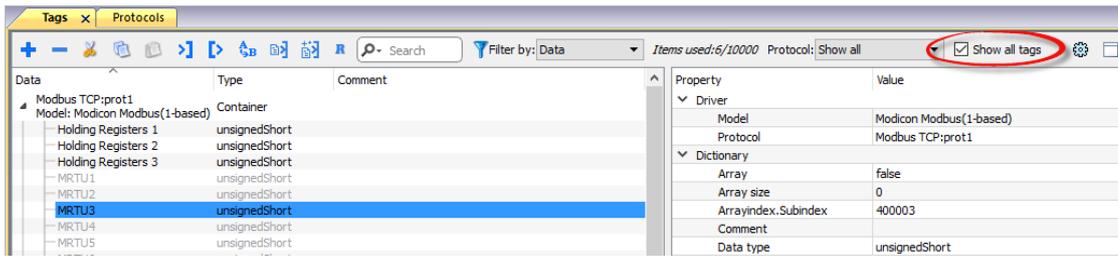


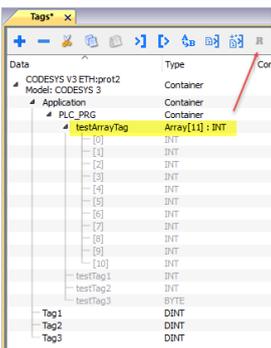
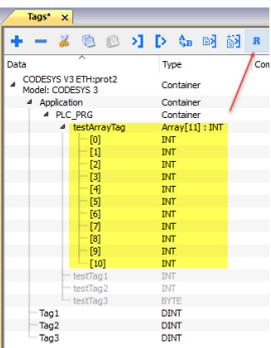
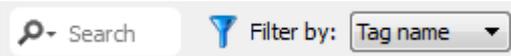
系统将需要通过相应按钮从标签编辑器中导出的通用 XML 文件。



选择导入程序后，找到符号文件并点击**打开**。

词典中可用但未导入到项目中的标签为灰色，并且仅当选中“显示所有标签”复选框时可见。



工具栏项	说明
	<p>导入标签。</p> <p>选择要导入的标签并点击此图标可将标签词典中的标签添加到项目</p>
	<p>更新标签。</p> <p>点击此图标可在导入新词典时更新项目中的标签。</p>
	<p>选中此框可导入标签的所有子元素。</p> <p>选中和未选中的结果示例：</p> <div style="display: flex; justify-content: space-around;">   </div>
	<p>根据选择的筛选器组合框项搜索词典中的标签。</p>

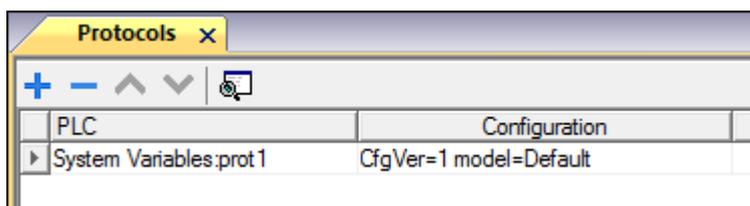
默认变量

“系统变量 - 默认”协议允许创建可指向以下相关 HMI 系统变量的标签：

- [报警](#)
- [响铃](#)
- [通信](#)
- [数据库](#)
- [夏令时间](#)
- [设备](#)
- [转储信息](#)
- [网络](#)
- [屏幕](#)
- [SD卡](#)
- [服务器](#)
- [时间](#)
- [USB 驱动器](#)
- [版本](#)
- [虚拟通信开关](#)

协议编辑器设置

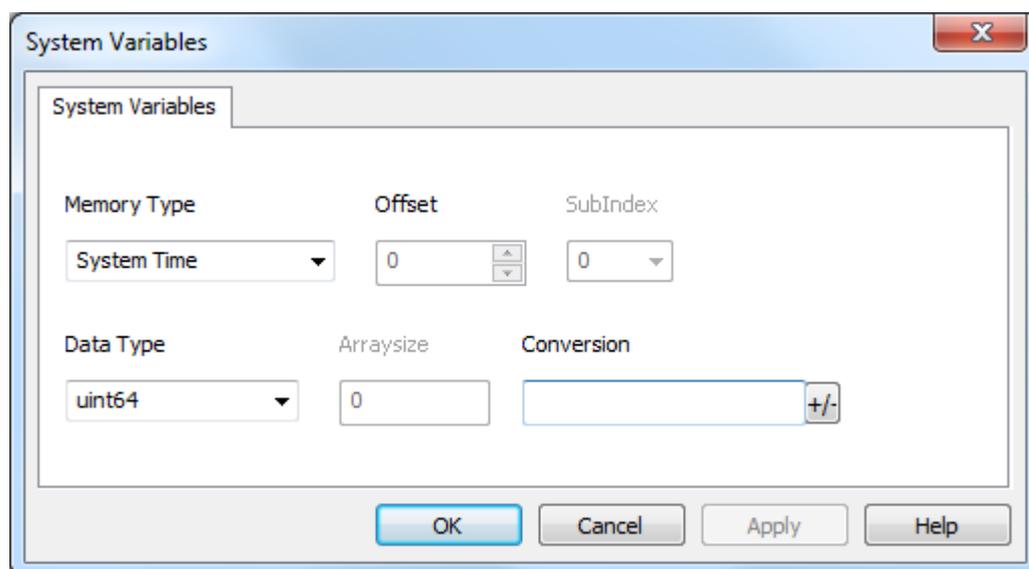
从“协议编辑器”对话框的“PLC 模式”列表中，选择“默认”。



标签编辑器设置

路径: **项目视图 > 配置 > 双击 标签**

1. 要添加标签，请单击 **+**：系统将添加新的一行。
2. 从 **驱动程序** 列表中选择 **系统变量**：“标签定义”对话框将显示。



元素	说明		
内存类型	表示标签所引用的系统变量。 以下部分显示按类别分组的可用系统变量的完整列表。		
	报警变量		
	变量名称	说明	数据类型
	未确认的报警	未确定报警时为真 (未启动未确定 <>0) 或 (未确定启动 <>0)	boolean 只读
	启动的报警	启动至少一个报警时为真 (确认启动 <>0) 或 (已启动 <>0) 或 (未确认启动 <>0)	boolean 只读
	丢失的报警事件数量	报警超过事件队列在 engineconfig.xml 文件中已定义队列长度。	int 只读
	未确认启动的数量	报警条件不再运行;报警已确认	int 只读
	未确认未启动的数量	报警条件不再运行;等待确认	int 只读
	确认启动的数量	报警条件在运行状态;报警已确认	int 只读
	启动报警的数量	报警处于运行状态;不需要确认	int 只读
未确认启动的数量	报警条件运行;等待确认	int 只读	

元素	说明	
	响铃变量	
	变量名称	说明
	蜂鸣器设置	<p>0 = 已禁用</p> <p>1 = 已启用(在出现任何触摸屏幕事件时,蜂鸣器发出声音)</p> <p>2 = 蜂鸣器状态由蜂鸣器控制系统变量或由主手册中“项目属性”内的触摸响铃属性控制</p> <p> 触摸屏幕响铃(设置 = 1)在 Linux 平台上不可用。请参阅可作为替代的“触摸响铃”属性。</p>
	蜂鸣器控制	<p>0 = 蜂鸣器关闭</p> <p>1 = 蜂鸣器打开</p> <p>2 = 蜂鸣器闪烁</p>
	蜂鸣器关闭的时间	当选择闪烁时,以毫秒表示关闭的持续时间。默认 = 1000。范围: 100–5000
	蜂鸣器打开的时间	当选择闪烁时,以毫秒表示打开的持续时间。默认 = 1000。范围: 100–5000
	通信变量	
	变量名称	说明
	协议通信状态	<p>总结通信协议的状态。</p> <p>0 = 无协议运行,协议驱动器可能未正确下载到 HMI 设备</p> <p>1 = 协议已加载并启动,无通信错误</p> <p>2 = 至少一个通信协议报告错误</p>
	协议错误信息	<p>具有错误源的通信错误。</p> <p>For example: “[xxxx]”中“xxxx”是协议缩写,是错误源。</p> <p>在多个错误源的情况下,多个缩写词出现。当无错误报表时,出现空白。</p>

元素	说明	
	通信变量	
	变量名称	说明
	协议错误数量	上一次复位以来的通信错误的数量。使用“复位协议错误计数”动作复位值，请参阅主手册中的“系统动作”
		int 只读
	数据库变量	
	变量名称	说明
	数据库链接错误信息	上一次检测的错误描述
		string 只读
	数据库链接状态。	0 = 未定义(未初始化) 1 = 连接(就绪) 2 = 未连接(不可用) 3 = 传输正在进行中 4 = 错误
		int 只读
	数据库链接错误计算	错误计数器。每次错误后会增加
		int 只读
	 每个数据库变量是一个数组，其中索引选择数据库关联连接(范围 1-10)变量仅在执行任一数据库连接器动作时更新	
	夏令时间变量	
	变量名称	说明
	标准偏移量	当标准时间设定时，偏移量以分钟计算，以 GMT 为准(例如：-8x60 = -480 分钟)
		int 只读
	标准周	标准时间开始的周(例如：第一 = 1)
		int 只读
	标准月	标准时间开始的月。范围：0-11。(例如：十一月 = 10)
		int 只读
	标准日	标准时间开始的星期(例如：星期天 = 0)
		int

元素	说明	
	夏令时间变量	
	变量名称	说明
		数据类型
		只读
	标准时	标准时间开始的小时(例如:02 = 2)
		int 只读
	标准分钟	标准时间开始的分钟(例如:00 = 0)
		int 只读
	DST 偏移量	当 DLS 时间设定时, 偏移量以分钟计算, 以 GMT 为准。
		int 只读
	DST 周	DLS 时间开始的周
		int 只读
	DST 月	DLS 时间开始的月。范围:0-11
		int 只读
	DST 日	DLS 时间开始的星期
		int 只读
	DST 小时	DLS 时间开始的小时
		int 只读
	DST 分钟	DLS 时间开始的分钟
		int 只读
	 所有变量均是只读的;它们不能用于更新系统时钟。	
	设备变量	
	变量名称	说明
		数据类型
	可用的系统存储	剩余可用 RAM 存储, 以字节为单位
		uint64 只读
	背光时间	从设备生产起, 显示屏背光的激活小时数
		unsignedInt 只读

元素	说明	
	设备变量	
	变量名称	说明
	电池 LED	<p>启用/停用低电量 LED 指示器(可用时)</p> <p>0 = 已禁用</p> <p>1 = 已启用</p> <p> Linux 平台上不可用(在 "HMI 设备地址" 在本页 525 寻找您的设备平台)</p>
	外部超时	<p>超过无操作时间后显示屏背光自动关闭。当用户接触屏幕时,背光自动打开</p> <p>-1 = 关闭背光并停用触摸(关闭显示屏)。背光时间计数器停止。</p> <p> 需要 BSP v1.0.324 或更高版本。</p> <p>-2 = 关闭背光但不停用触摸。如果按下触摸,事件未传达至应用程序,但屏幕保护程序退出且背光打开。</p> <p> 需要 BSP v1.0.324 或更高版本。</p> <p>0 = 打开背光(打开显示屏)</p> <p>1..n = 超时,以秒计算,用于关闭背光(屏幕保护程序计时器)</p> <p> 上,超时限值累计为分钟倍数(60, 120, 180 等)。</p> <p>在 "HMI 设备地址" 在本页 525 寻找您的设备平台。</p>
	闪存空余空间	<p>内部闪存剩余空间</p> <p>uint64 只读</p>
	制造商代码	<p>识别 HMI 的代号</p> <p>short 只读</p>
	系统 RAM 用途	<p>HMI 使用的当前 RAM 内存,以字节表示</p> <p>uint64 只读</p>

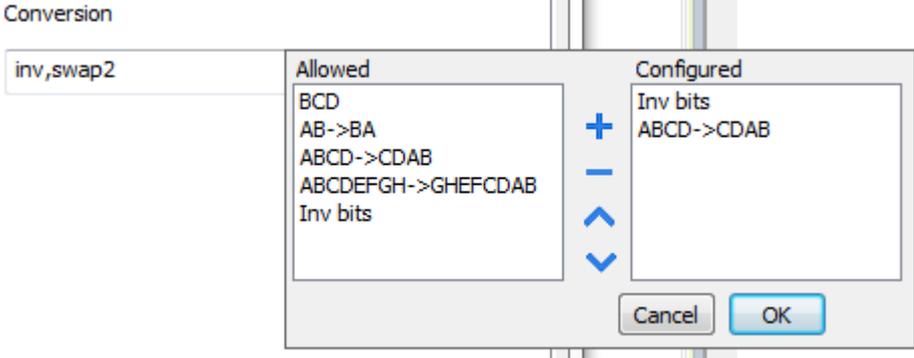
元素	说明	
	设备变量	
	变量名称	说明
	系统字体列表	系统字体列表 string 只读
	系统模式	Runtime 操作状态 1 = 正在启动 2 = 配置模式 3 = 操作模式 4 = 重启 5 = 关闭 int
	系统开启时间	设备自生产起的系统通电时间(小时) unsignedInt 只读
	转储信息变量	
	变量名称	说明
	转储错误信息	在执行转储操作期间,如果出现任何错误,则将返回错误消息 string 只读
	存储存档状态	0 = 初始默认状态 1 = 操作已触发 2 = 操作成功完成 3 = 操作出错 int 只读
	存储配方状态	0 = 初始默认状态 1 = 操作已触发 2 = 操作成功完成 3 = 操作出错 int 只读
	存储趋势状态	0 = 初始默认状态 1 = 操作已触发 2 = 操作成功完成 3 = 操作出错 int 只读

元素	说明	
	转储信息变量	
	变量名称	说明
	复位配方状态	0 = 初始默认状态 1 = 操作已触发 2 = 操作成功完成 3 = 操作出错
	数据类型	int 只读
	恢复配方状态	0 = 初始默认状态 1 = 操作已触发 2 = 操作成功完成 3 = 操作出错
	数据类型	int 只读
	网络变量 0	
	变量名称	说明
	网关	HMI 主网络界面的网关地址
	数据类型	string 只读
	IP 地址	HMI 主网络界面的 IP 地址
	数据类型	string 只读
	Mac ID	HMI 主网络界面的 MAC ID
	数据类型	string 只读
	网络适配器参数	可以用来读取或更新网络适配器参数的 JSON 字符串
	网络状态	包含写入适配器参数内所需的所需的最后操作的结果。在每次写稿操作完成后更新。 <ul style="list-style-type: none"> • 空字符串代表无错误 • 最后错误说明
	数据类型	string 只读
	子网掩码	HMI 主网络界面的子网掩码
	数据类型	string 只读

元素	说明	
	屏幕变量	
	变量名称	说明
	X Screen resolution	以像素显示横向屏幕大小
		数据类型 int 只读
	Y Screen resolution	以像素显示纵向屏幕大小
		数据类型 int 只读
	SD 卡变量	
	变量名称	说明
	SD 卡剩余空间	卡上的可用字节空间
		数据类型 uint64 只读
	SD 卡名称	SD 卡名称
		数据类型 string 只读
	SD 卡大小	卡槽中卡的字节大小
		数据类型 uint64 只读
	SD 卡状态	0 = SD 卡已拔出 1 = SD 卡已插入
		数据类型 int 只读
	服务器变量	
	变量名称	说明
	页面名称	当前页面的名称
		数据类型 string 只读
	当前项目	当前项目的名称
		数据类型 string 只读
	项目加载时间	项目加载到 LRH SW HMI Runtime 上的日期, 采用系统日期格式(毫秒)
		数据类型 uint64 只读
	上次操作模式开始时间	设备启动操作模式后经过的秒数
		数据类型 uint64 只读
	 所有变量均与服务器相关, 而非 LRH SW Client。	

元素	说明	
	时间变量	
	变量名称	说明
	月中日期	范围: 1-31 int
	星期几	范围: 0 = 星期日, 依此类推, 6 = 星期六 int
	小时	范围: 0-23 int
	分钟	范围: 0-59 int
	月	范围: 1-12 int
	次要	范围: 0-59 int
	系统时间	与 UTC 时间相同。也可设为此变量的日期/时间 unsignedInt
	年	当前年份 int
	 以 UTC 格式表示的系统时间	
	USB 驱动器变量	
	变量名称	说明
	USB 驱动器剩余空间	可用字节空间 uint64 只读
	USB 驱动名称	USB 设备的名称 string 只读
	USB 驱动器的大小	USB 端口插入的设备的字节大小 uint64 只读
	USB 驱动器的状态	0 = USB 驱动器已拔出 1 = USB 驱动器已插入 int 只读

元素	说明	
	版本变量	
	变量名称	说明
	主操作系统版本	主操作系统版本 string 只读
	Runtime 版本	Runtime 版本 string 只读
	项目名称	项目名称 string 只读
	与运行版本不一致	与运行版本不一致 string
	项目 GUID	项目 GUID(唯一标识符) string 只读
	虚拟通信开关变量	
	变量名称	说明
	VCS 状态	提供 VCS 服务的状态。 0 = 已启用服务 1 = 客户端以交错模式连接 2 = 客户端以独占模式连接 3 = 已禁用服务(默认) unsignedByte 只读
	VCS 禁用	提供 VCS 服务的手动覆盖。 0 = 已启用 VCS 服务 1 = 已禁用 VCS 服务(默认) boolean
	VCS 端口	通过 VCS 服务提供 HMI 上的当前侦听 TCP 端口 unsignedShort
数据类型	每个系统变量具有特定的数据类型，如上表中所述。 下表显示用于系统变量的任何数据类型详细信息。	

元素	说明																								
	<table border="1"> <thead> <tr> <th>数据类型</th> <th>内存空间</th> <th>限制</th> </tr> </thead> <tbody> <tr> <td>short</td> <td>16 位数据</td> <td>-32768 ...32767</td> </tr> <tr> <td>int</td> <td>32 位数据</td> <td>-2.1e9 ...2.1e9</td> </tr> <tr> <td>unsignedByte</td> <td>8 位数据</td> <td>0 ...255</td> </tr> <tr> <td>unsignedShort</td> <td>16 位数据</td> <td>0 ...65535</td> </tr> <tr> <td>unsignedInt</td> <td>32 位数据</td> <td>0 ...4.2e9</td> </tr> <tr> <td>uint64</td> <td>64 位数据</td> <td>0 ...1.8e19</td> </tr> <tr> <td>string</td> <td colspan="2">元素数组, 包含通过所选编码定义的字符代码</td> </tr> </tbody> </table>	数据类型	内存空间	限制	short	16 位数据	-32768 ...32767	int	32 位数据	-2.1e9 ...2.1e9	unsignedByte	8 位数据	0 ...255	unsignedShort	16 位数据	0 ...65535	unsignedInt	32 位数据	0 ...4.2e9	uint64	64 位数据	0 ...1.8e19	string	元素数组, 包含通过所选编码定义的字符代码	
数据类型	内存空间	限制																							
short	16 位数据	-32768 ...32767																							
int	32 位数据	-2.1e9 ...2.1e9																							
unsignedByte	8 位数据	0 ...255																							
unsignedShort	16 位数据	0 ...65535																							
unsignedInt	32 位数据	0 ...4.2e9																							
uint64	64 位数据	0 ...1.8e19																							
string	元素数组, 包含通过所选编码定义的字符代码																								
数组大小	<p>如果是字符串标签, 此属性代表字符串标签中字节的最大数量。</p> <p>注: 如果在标签编辑器中将编码属性设为 UTF-8 或 Latin1, 则字节数将与字符串字符数对应。</p> <p>如果编码属性设为 UCS-2BE、UCS-2LE、UTF-16BE 或 UTF-16LE, 则一个字符需要 2 个字节。</p>																								
转换	<p>应用于标签的转换。</p>  <p>根据选择的数据类型, 允许列表将显示一个或多个转换类型。</p> <table border="1"> <thead> <tr> <th>值</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>Inv bits</td> <td> <p>inv: 对标签的所有位取反。</p> <p>示例:</p> <p>1001 → 0110(二进制格式)</p> <p>9 → 6(十进制格式)</p> </td> </tr> <tr> <td>Negate(取反)</td> <td> <p>neg: 设置标签值求反。</p> <p>示例:</p> <p>25.36 → -25.36</p> </td> </tr> </tbody> </table>	值	说明	Inv bits	<p>inv: 对标签的所有位取反。</p> <p>示例:</p> <p>1001 → 0110(二进制格式)</p> <p>9 → 6(十进制格式)</p>	Negate(取反)	<p>neg: 设置标签值求反。</p> <p>示例:</p> <p>25.36 → -25.36</p>																		
值	说明																								
Inv bits	<p>inv: 对标签的所有位取反。</p> <p>示例:</p> <p>1001 → 0110(二进制格式)</p> <p>9 → 6(十进制格式)</p>																								
Negate(取反)	<p>neg: 设置标签值求反。</p> <p>示例:</p> <p>25.36 → -25.36</p>																								

元素	说明	
	值	说明
	AB → BA	swapnibbles: 用一个字节交换半字节。 示例: 15D4 → 514D(十六进制格式) 5588 → 20813(十进制格式)
	ABCD → CDAB	swap2: 用一个字来交换字节。 示例: 9ACC → CC9A(十六进制格式) 39628 → 52378(十进制格式)
	ABCDEFGH → GHEFCDAB	swap4: 以双字交换字节。 示例: 32FCFF54 → 54FFFC32(十六进制格式) 855441236 → 1426062386(十进制格式)
	ABC...NOP → OPM...DAB	swap8: 用长字交换字节。 示例: 142.366 → -893553517.588905(十进制格式) 0 10000000110 000111001011101101100100010110100001110010101100000 1 → 1 10000011100 101010100001010001011011011001011011000010011110 1 (二进制格式)
	BCD	bcd: 在两个半字节中分隔字节, 将它们读为十进制(从0到9) 示例: 23 → 17(十进制格式) 0001 0111 = 23 0001 = 1(第一个半字节) 0111 = 7(第二个半字节)

选择转换并点击+。所选项目将被添加到**已配置**列表中。

如果配置了多个转换, 将按顺序应用这些转换(从**已配置**列表自上而下)。

使用箭头按钮可对配置的转换进行排序。

保留内存变量

“系统变量 - 保留内存”协议允许创建可指向当 HMI 关闭时保存其内容的内存区域的标签。

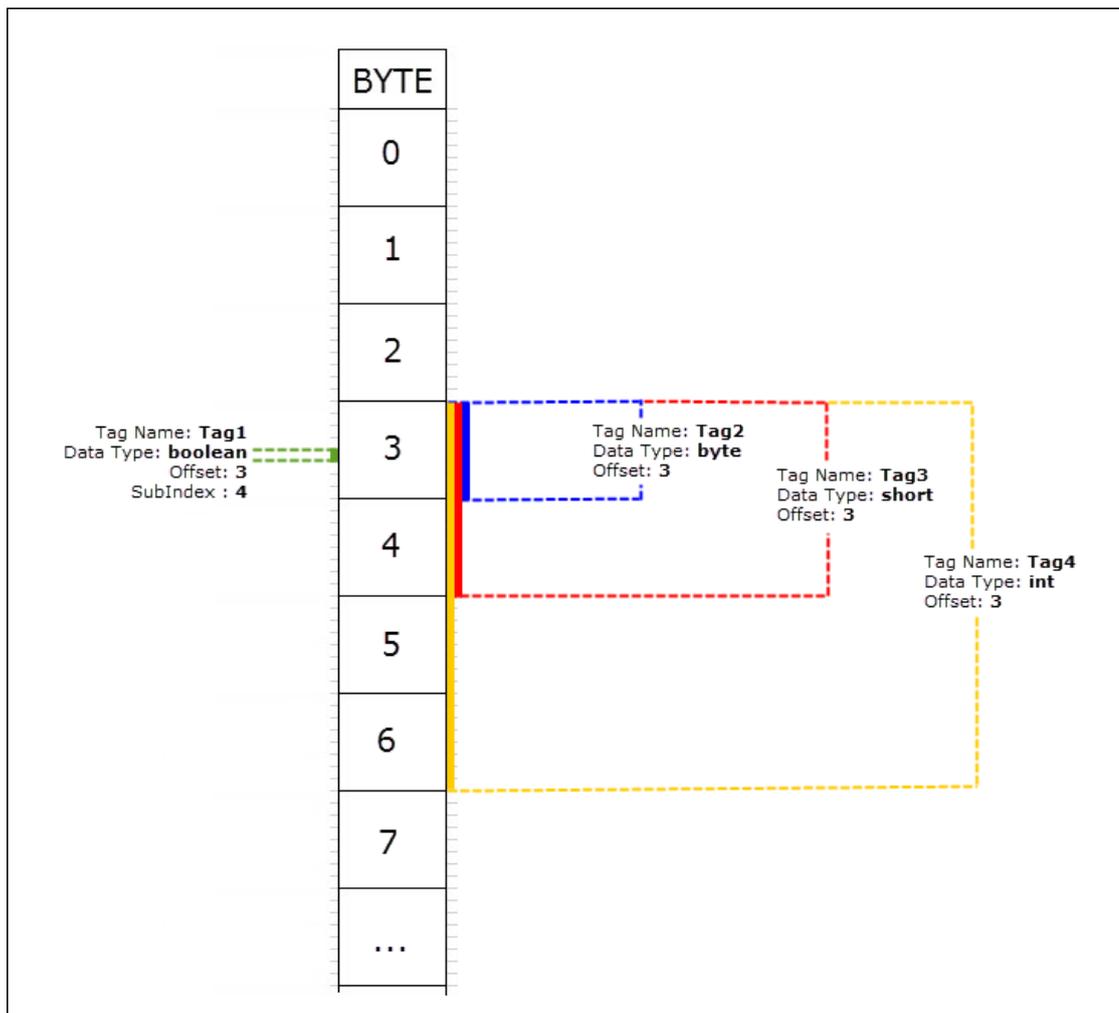
对保留内存的物理支持基于 FRAM 技术。



重要:并非所有 HMI 设备均包括 FRAM 内存。如果 FRAM 内存不可用,可使用用户内存存储(闪存、硬盘驱动器)支持持久性。闪存技术在最大数量的写入操作方面存在限制。使用闪存作为具有频繁写入操作的保留内存的存储介质,可能会损坏内存组件。检查 HMI 设备数据,以确定 FRAM 内存的可用性。



重要:保留内存是以字节组织的 16 KB 平坦内存区域并可通过偏移访问。请参阅以下方案。

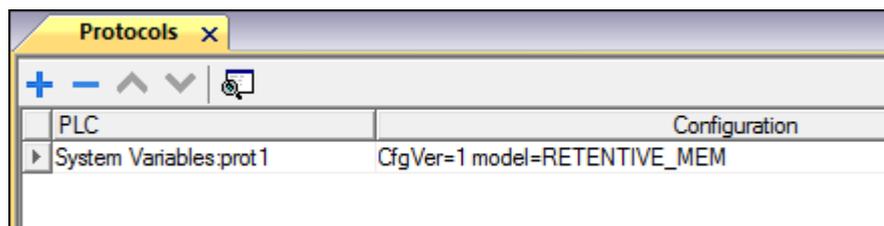


保留内存与配方存储

当保留数据保存在 FRAM 中时,配方数据保存在闪存中(LRH SW PC Runtime除外)。闪存不适用于大量写入操作,而 FRAM 几乎支持无限数量的写入操作,并且在需要频繁写入操作时应作为首选。

协议编辑器设置

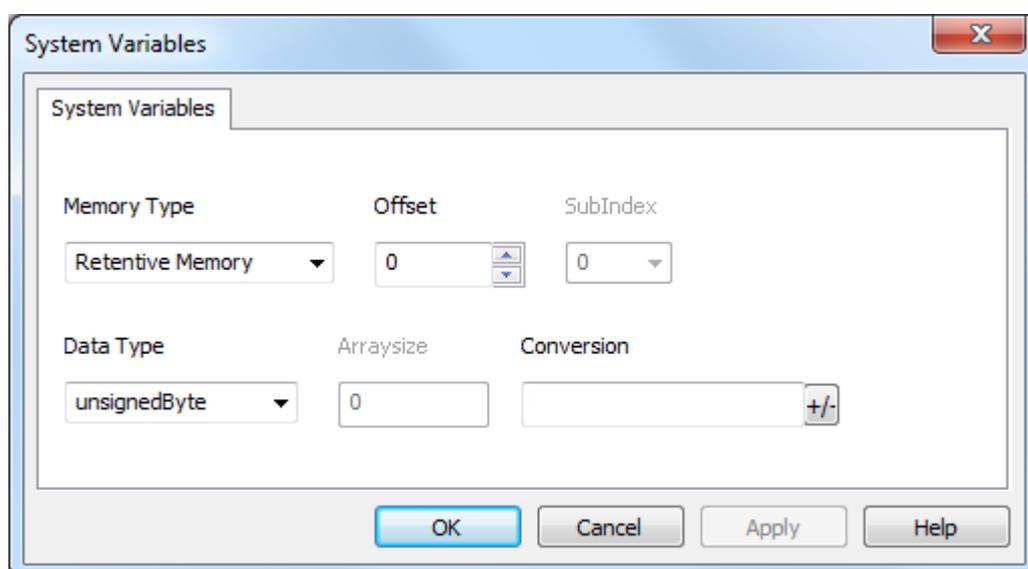
从“协议编辑器”对话框的“PLC 模式”列表中,选择“保持性存储”。



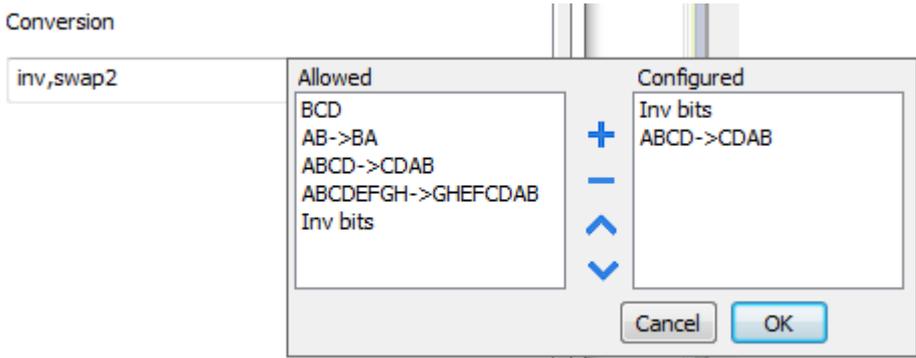
标签编辑器设置

路径: 项目视图 > 配置 > 双击标签

1. 要添加标签, 请单击 **+**: 系统将添加新的一行。
2. 从驱动程序列表中选择**系统变量**: “标签定义”对话框将显示。



元素	说明															
内存类型	固定为保留内存															
偏移	标签所在的偏移地址。范围: 0–16383															
子索引	此参数允许基于所选数据类型的资源偏移选择。															
数据类型	<table border="1"> <thead> <tr> <th>数据类型</th> <th>内存空间</th> <th>限制</th> </tr> </thead> <tbody> <tr> <td>boolean</td> <td>1 位数据</td> <td>0 ... 1</td> </tr> <tr> <td>byte</td> <td>8 位数据</td> <td>-128 ... 127</td> </tr> <tr> <td>short</td> <td>16 位数据</td> <td>-32768 ... 32767</td> </tr> <tr> <td>int</td> <td>32 位数据</td> <td>-2.1e9 ... 2.1e9</td> </tr> </tbody> </table>	数据类型	内存空间	限制	boolean	1 位数据	0 ... 1	byte	8 位数据	-128 ... 127	short	16 位数据	-32768 ... 32767	int	32 位数据	-2.1e9 ... 2.1e9
数据类型	内存空间	限制														
boolean	1 位数据	0 ... 1														
byte	8 位数据	-128 ... 127														
short	16 位数据	-32768 ... 32767														
int	32 位数据	-2.1e9 ... 2.1e9														

元素	说明																														
	<table border="1"> <thead> <tr> <th>数据类型</th> <th>内存空间</th> <th>限制</th> </tr> </thead> <tbody> <tr> <td>int64</td> <td>64 位数据</td> <td>-9.2e18 ...9.2e18</td> </tr> <tr> <td>unsignedByte</td> <td>8 位数据</td> <td>0 ...255</td> </tr> <tr> <td>unsignedShort</td> <td>16 位数据</td> <td>0 ...65535</td> </tr> <tr> <td>unsignedInt</td> <td>32 位数据</td> <td>0 ...4.2e9</td> </tr> <tr> <td>uint64</td> <td>64 位数据</td> <td>0 ...1.8e19</td> </tr> <tr> <td>float</td> <td>IEEE 单精度 32 位浮点类型</td> <td>1.17e-38 ...3.4e38</td> </tr> <tr> <td>double</td> <td>IEEE 双精度 64 位浮点类型</td> <td>2.2e-308 ...1.79e308</td> </tr> <tr> <td>string</td> <td colspan="2">元素数组, 包含通过所选编码定义的字符代码</td> </tr> <tr> <td>binary</td> <td colspan="2">任意二进制数据</td> </tr> </tbody> </table> <p> 注:要定义数组,请选择一种数据类型格式,后跟方括号,如“byte[]”、“short[]”等</p>	数据类型	内存空间	限制	int64	64 位数据	-9.2e18 ...9.2e18	unsignedByte	8 位数据	0 ...255	unsignedShort	16 位数据	0 ...65535	unsignedInt	32 位数据	0 ...4.2e9	uint64	64 位数据	0 ...1.8e19	float	IEEE 单精度 32 位浮点类型	1.17e-38 ...3.4e38	double	IEEE 双精度 64 位浮点类型	2.2e-308 ...1.79e308	string	元素数组, 包含通过所选编码定义的字符代码		binary	任意二进制数据	
数据类型	内存空间	限制																													
int64	64 位数据	-9.2e18 ...9.2e18																													
unsignedByte	8 位数据	0 ...255																													
unsignedShort	16 位数据	0 ...65535																													
unsignedInt	32 位数据	0 ...4.2e9																													
uint64	64 位数据	0 ...1.8e19																													
float	IEEE 单精度 32 位浮点类型	1.17e-38 ...3.4e38																													
double	IEEE 双精度 64 位浮点类型	2.2e-308 ...1.79e308																													
string	元素数组, 包含通过所选编码定义的字符代码																														
binary	任意二进制数据																														
数组大小	<ul style="list-style-type: none"> 如果是数组标签,此属性代表数组元素的数量。 如果是字符串标签,此属性代表字符串标签中字节的最大数量。 <p>注:如果在标签编辑器中将编码属性设为 UTF-8 或 Latin1,则字节数将与字符串字符数对应。 如果编码属性设为 UCS-2BE、UCS-2LE、UTF-16BE 或 UTF-16LE,则一个字符需要 2 个字节。</p>																														
转换	<p>应用于标签的转换。</p>  <p>根据选择的数据类型,允许列表将显示一个或多个转换类型。</p>																														

元素	说明	
	值	说明
	Inv bits	inv: 对标签的所有位取反。 示例: 1001 → 0110(二进制格式) 9 → 6(十进制格式)
	Negate(取反)	neg: 设置标签值求反。 示例: 25.36 → -25.36
	AB → BA	swapnibbles: 用一个字节交换半字节。 示例: 15D4 → 514D(十六进制格式) 5588 → 20813(十进制格式)
	ABCD → CDAB	swap2: 用一个字来交换字节。 示例: 9ACC → CC9A(十六进制格式) 39628 → 52378(十进制格式)
	ABCDEFGH → GHEFC DAB	swap4: 以双字交换字节。 示例: 32FCFF54 → 54FFFC32(十六进制格式) 855441236 → 1426062386(十进制格式)
	ABC...NOP → OPM...DAB	swap8: 用长字交换字节。 示例: 142.366 → -893553517.588905(十进制格式) 0 10000000110 000111001011101101100100010110100001110010101100001 → 1 10000011100 1010101000010100010110110110110010110110000100111101 (二进制格式)
	BCD	bcd: 在两个半字节中分隔字节, 将它们读为十进制(从0到9) 示例: 23 → 17(十进制格式) 0001 0111 = 23 0001 = 1(第一个半字节) 0111 = 7(第二个半字节)

选择转换并点击+。所选项目将被添加到**已配置**列表中。

元素	说明
	如果配置了多个转换, 将按顺序应用这些转换(从 已配置 列表自上而下)。使用箭头按钮可对配置的转换进行排序。

清理保留内存

使用**清除保留内存**动作清除保留内存的内容。



提示: 使用此动作可将内存内容随时设置为已知状态。

请参阅主手册中的**动作 > 标签动作**部分, 了解更多详细信息。



用于此动作的 JavaScript 接口为:
`project.clearRetentiveMemory();`

项目下载时保留保留内存

将项目文件下载到 HMI 或修改活动项目时, 通常将删除保留内存的内容。

如需在项目下载或更新时保留保留数据的内容, 请选择 HMI 设备设置选项卡中的**项目更新时保持保留数据**选项。

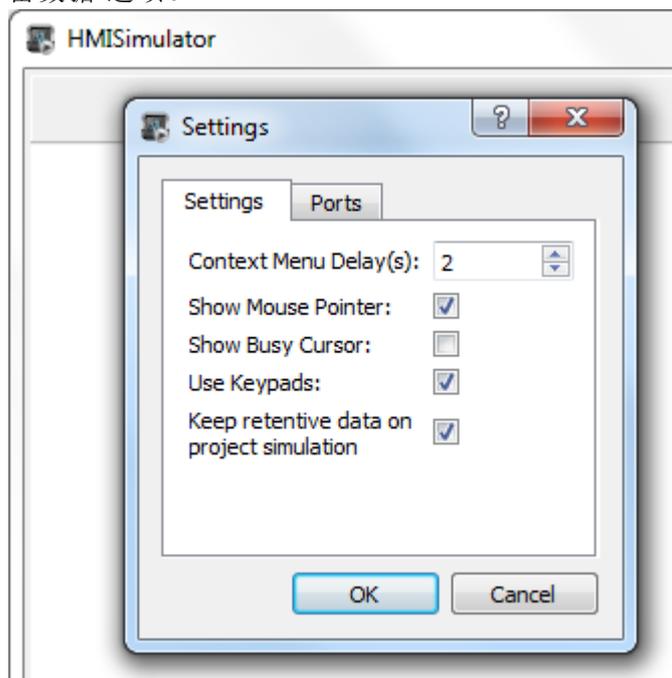


如果从下载至目标窗口中选择**删除 runtime 动态文件**选项, 此设置将被忽略。



在模拟器中保留保留内存

LRH SW 模拟器支持保留内存。要在项目模拟期间启用保留内存，请使用上下文菜单中的“项目模拟时保持保留数据”选项。



服务变量

服务变量可读取状态并向 VNC 服务器传递命令。

协议编辑器设置

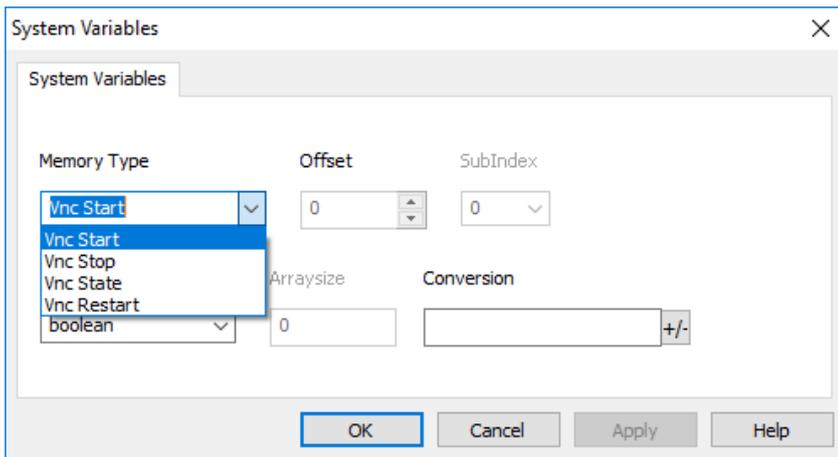
路径: 项目视图 > 协议

1. 单击 **+** 添加 **系统变量:系统变量** 对话框显示。
2. 从 **PLC 模式** 列表中选择 **服务**。

标签编辑器设置

路径: **项目视图 > 配置 > 双击 标签**

1. 要添加标签, 请单击 **+**: 系统将添加新的一行。
2. 从 **驱动程序** 列表中选择 **系统变量**: “标签定义”对话框将显示。

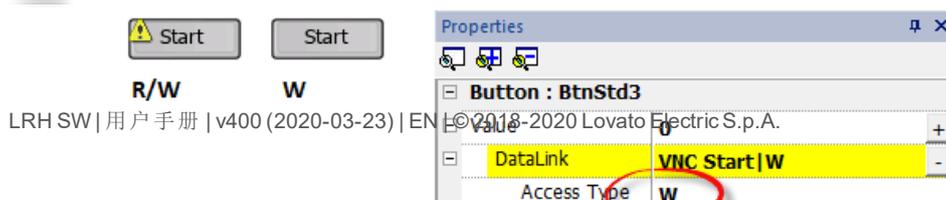


只有配备 BSP 1.0.344 版或更高版本的 Linux 设备才支持 VNC 状态变量。(请参阅 ["HMI 设备地址"](#) 在本页 525)

元素	说明	数据类型
VNC 启动	在此标签中写入 1 以强制 VNC 服务器启动。  这是一个只写变量, 当您随时重新写入时将执行的命令。	boolean 只写
VNC 停止	在此标签中写入 1 以强制 VNC 服务器停止。  这是一个只写变量, 当您随时重新写入时将执行的命令。	boolean 只写
VNC 重新启动	在此标签中写入 1 以强制 VNC 服务器重新启动。  这是一个只写变量, 当您随时重新写入时将执行的命令。	boolean 只写
VNC 状态	VNC 服务器状态 0 闲置 10 运行中 -1 错误	int 只读



无法读取只写变量。确保不要使用 R/W 访问模式以免显示读取错误图标。





PLCM09 变量

PLCM09 设备是一个具有 LED 和数字 I/O 的无线调制解调器。相关系统变量的行为取决于如何在“系统设置”中配置该模块(请参阅“[PLCM09 插入式无线调制解调器](#)”了解更多信息)

协议编辑器设置

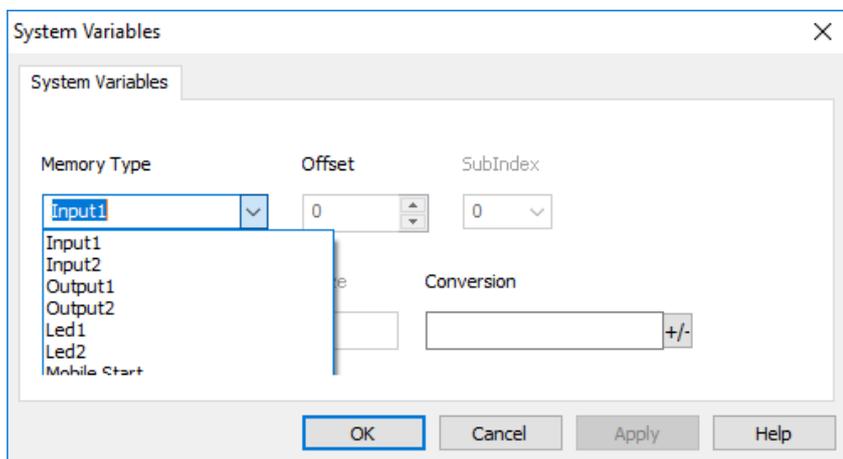
路径: **项目视图** > **协议**

1. 单击 **+** 添加 **系统变量**: 系统变量对话框显示。
2. 从 **PLC 模式** 列表中选择 PLCM09。

标签编辑器设置

路径: **项目视图** > **配置** > 双击 **标签**

1. 要添加标签, 请单击 **+**: 系统将添加新的一行。
2. 从 **驱动程序** 列表中选择 **系统变量**: “标签定义”对话框将显示。



元素	说明	数据类型
Input1	输入信号值	boolean
Input2		只读
Output1	输入信号值	boolean
Output2	配置为“用户控制”时, 输出变量为只读/只写(请参阅“ PLCM09 插入式无线调制解调器 ”了解更多信息)。在直接从 PLCM09 模块控制输出信号的其他配置中, 系统变量为只读。	读写
Led1	LED 状态值	unsignedByte
Led2	<ul style="list-style-type: none"> • 0 = 关 • 1 = 开 • 2 = 闪烁 配置为“用户控制”时, LED 变量为只读/只写(请参阅“ PLCM09 插入式无线调制解调器 ”了解更多信息)。在直接从 PLCM09 模块控制 LED 状态的其他配置中, 系统变量为只读。	读写

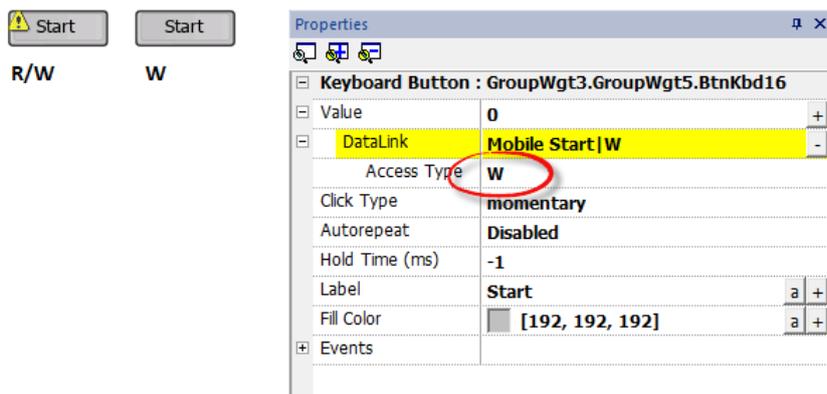
元素	说明	数据类型
手机启动	<p>在此标签中写入 1 以强制无线模式启动。</p> <p> 这是一个只写变量，当您随时重新写入时将执行的命令。</p>	<p>boolean</p> <p>只写</p>
手机停止	<p>在此标签中写入 1 以强制无线模式停止。</p> <p> 这是一个只写变量，当您随时重新写入时将执行的命令。</p>	<p>boolean</p> <p>只写</p>
手机重启	<p>在此标签中写入 1 以强制无线模式重启。</p> <p> 这是一个只写变量，当您随时重新写入时将执行的命令。</p>	<p>boolean</p> <p>只写</p>
手机状态	<p>手机连接状态</p> <ul style="list-style-type: none"> 0 闲置 1 启动中 10 运行中 100 连接中 200 已连接 300 正在停止 -1 通用错误 -10 系统错误 -100 未找到调制解调器 -101 调制解调器繁忙 -110 调制解调器通信 -120 调制解调器超时 -130 调制解调器错误 -200 SIM 缺失 -300 需要 PIN -301 需要新 PIN 	<p>int</p> <p>只读</p>

元素	说明	数据类型
	-310 PIN 错误 -320 需要 PUK -330 PUK 错误 -400 漫游受阻 -500 用户信息错误	
手机信号	手机信号质量 (0-100)	byte 只读
手机运营商	手机运营商名称(例如, 'Vodafone')	字符串 [8] 只读
手机访问技术	手机访问技术 -1 不适用 0 GSM (2G) 2 UTRAN (2G) 3 GSM W/EGPRS (2G) 4 UTRAN W/HSDPA (3G) 5 UTRAN W/HSUPA (3G) 6 UTRAN W/HSDPA 和 HSUPA (3G)	int 只读
手机注册状态	手机注册状态 -1 不适用 0 未注册。无线调制解调器当前未搜索要注册的新运营商。 1 已在家庭网络上注册。 2 未注册。无线调制解调器当前正搜索要注册的新运营商。 3 注册被拒。 4 未知 5 漫游时注册	int 只读
手机 RX/TX	号码或接收/传输的字节数	unsignedInt[2]

元素	说明	数据类型
		只读
手机启动时间(秒)	启动手机连接时(几秒钟后)	unsignedInt 只读



无法读取只写变量。确保不要使用 R/W 访问模式以免显示读取错误图标。



JavaScript(手机连接状态)

当“protocolSysVar”为协议实例代码(例如“prot1”, “prot2”等)时,也可从下列 JavaScript 界面检索手机连接状态

```
Mobile_State = tagMgr.invokeProtocolCommand(protoSysVar,"get_mobile_state", ""); //获取调制解调器状态
```

14 动作

动作是用于与系统交互的功能，通常在事件发起的时候执行。

事件可以由各种控件发起，如按压和释放一个按钮。并不是所有动作对一个对象的所有事件都有效。

在属性面板(页面编辑器)事件部分，动作与各控件相关联。

报警动作	168
数据库动作	169
事件动作	172
多语言动作	172
键盘动作	173
多媒体播放器动作	175
FTP 动作	175
页面动作	178
打印动作	183
配方动作	185
远程客户动作	189
系统动作	190
标签动作	202
趋势动作	204
文本编辑器动作	208
用户管理动作	208
控件动作	211

报警动作

主要用于确认或复位报警。

选择所有报警

选择所有报警。

参数	说明
模式	<p>切换 反转选择状态。</p> <p> 切勿选择未触发或无待定确认或复位请求的报警。</p> <p>选择 选择所有已触发或具有确认或复位请求待定状态的报警</p> <p>取消选择 取消选择所有报警</p>

选择报警

选择特定报警。

参数	说明
报警 ID	报警 ID
选择标志	<p>TRUE 选择报警。</p> <p> 不选择未触发或无待定确认或复位请求的报警。</p> <p>FALSE 取消选择报警。</p>

确认报警

确认特定报警或所有所选报警。

参数	说明
报警 ID	<p>特定报警 ID</p> <p>选中 所有选中的报警</p>

确认所选报警。

复位报警

复位特定报警或未触发和确认的所有所选报警。

参数	说明
报警 ID	特定报警 ID 选中 所有选中的报警

启用报警

启用或停用特定报警或所有所选报警。

停用的报警将不会生成报警事件。

参数	说明
报警 ID	特定报警 ID 选中 所有选中的报警
选择标志	TRUE 启用报警。 FALSE 停用报警。

数据库动作

DBInit



重要: 这一动作只在空白数据库使用一次。这不是 HMI 设备任何时候启动要调用的初始化命令。

按项目要求创建表格。如果数据库已经包含所需的表格，这一动作便不需要。

Action Properties

DBInit	
Link Name	myRemoteDB
Custom SQL query	
Link Name Database link name	

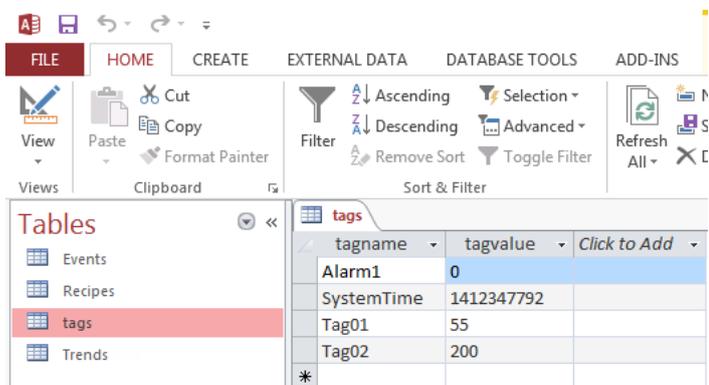
实用自定义 **SQL 查询** 参数来限定要创建的页面。留出空白来生成默认表格名称



提示:在您的项目的设置页添加该命令,仅在首次安装应用时获得授权的人可以使用。

JavaScript 界面

```
project.dbInit (dbLinkName, sqlCustomQuery);
```



DB写入标签, DB读取标签

将选择的标签的值转移到远程数据库,或从远程数据库转移所选的标签的值。

Action Properties

DBWriteTags	
Link Name	myRemoteDB
Custom SQL query	
Tag names	Tag01;Tag02
Link Name Database link name	

Action Properties

DBReadTags	
Link Name	myRemoteDB
Custom SQL query	
Tag names	Tag01;Tag02
Link Name Database link name	

JavaScript 界面

```
project.dbWriteTags (dbLinkName, sqlCustomQuery, Tags);
```

```
project.dbReadTags (dbLinkName, sqlCustomQuery, Tags);
```

DB写入组, DB读取组

在HMI设备和数据库之间传输标签组

Action Properties

DBWriteGroups	
Link Name	myRemoteDB
Custom SQL query	
Group names	Group1
Link Name Database link name	

Action Properties

DBReadGroups	
Link Name	myRemoteDB
Custom SQL query	
Group names	Group1
Link Name Database link name	

JavaScript 界面

```
project.dbWriteGroups(dbLinkName, sqlCustomQuery, Groups);
```

```
project.dbReadGroups(dbLinkName, sqlCustomQuery, Groups);
```

DB写入趋势

将在所选的时间范围示例的最近数据的值插入到远程数据库趋势表格中。

Action Properties

DBWriteTrends	
Link Name	myRemoteDB
Custom SQL query	
Trend names	Trend1
Duration	10 min

Link Name
Database link name

JavaScript 界面

```
project.dbWriteTrends(dbLinkName, sqlCustomQuery, trendName, durationIndex)
```

DB写入事件

将在所选的时间范围最近事件的值插入到远程数据库趋势表格中。

Action Properties

DBWriteEvents	
Link Name	myRemoteDB
Custom SQL query	
Buffer	AlarmBuffer1
Duration	1 hour

Buffer
Select Event buffer

Action Properties

DBWriteEvents	
Link Name	myRemoteDB
Custom SQL query	
Buffer	AuditTrail
Duration	1 hour

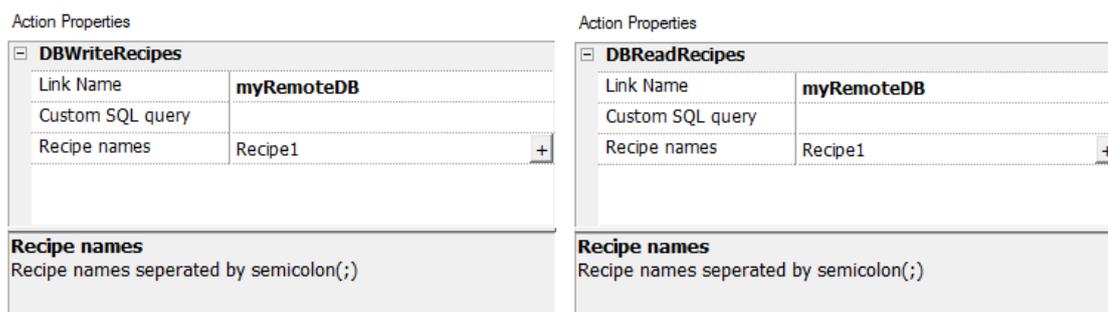
Buffer
Select Event buffer

JavaScript 界面

```
project.dbWriteEvents(dbLinkName, sqlCustomQuery, archiveName, durationIndex)
```

DB写入配方, DB读取配方

将配方数据传输到或从远程数据块传输。



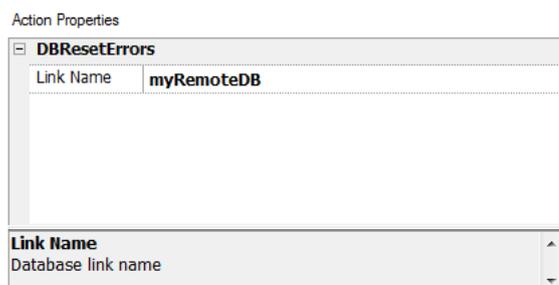
JavaScript 界面

```
project.dbWriteRecipes(dbLinkName, sqlCustomQuery, recipeNames)
```

```
project.dbReadRecipes(dbLinkName, sqlCustomQuery, recipeNames)
```

DBResetErrors

复位所选的数据库链接的三个状态变量。第 1 页的“数据库变量”



JavaScript 界面

```
project.dbResetErrors(dbLinkName)
```

事件动作

报警历史控件用来使事件/报警在桌面视图向前向后滚动(事件缓存控件)。

向后滚动事件

使事件/报警在桌面视图下向后滚动(事件缓存控件)。

向前滚动事件

使事件/报警在桌面视图下向前滚动(事件缓存控件)。

多语言动作

选择应用程序语言。

设置语言

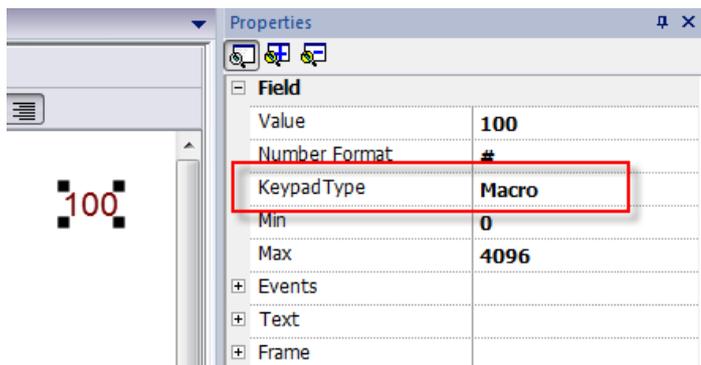
设置使用的语言。选择的语言在运行时将会被运用到所有可应用的控件上。

键盘动作

改变键盘的使用。

发送键

向数字控件发送一个字符。数字控件的**键盘输入**属性必须设定为 **Macro**。

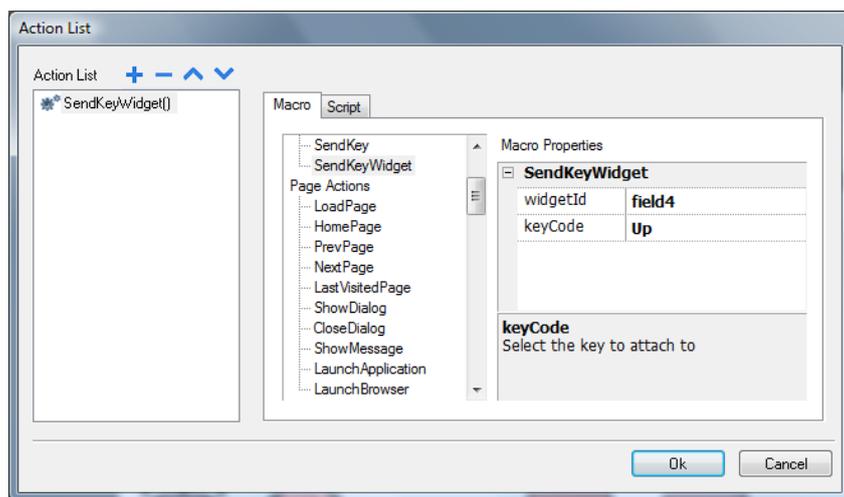
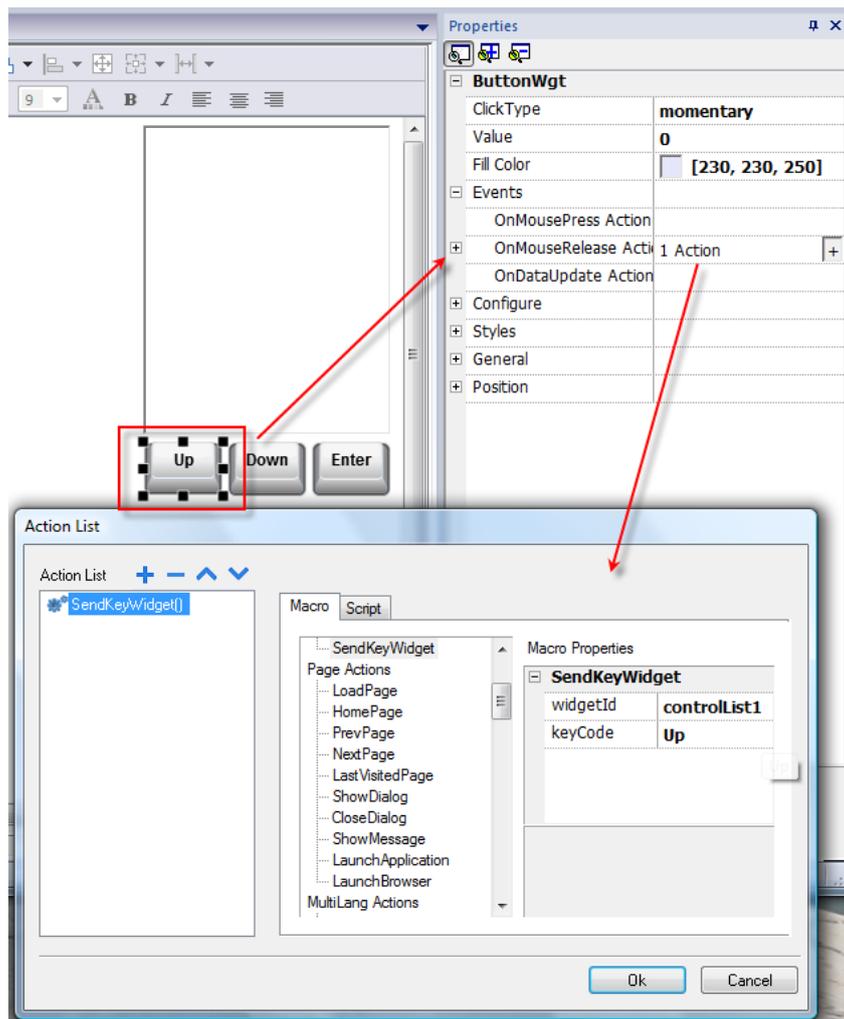


发送键控件

向特定的控件发送一个字符。

例如

上下键使用与**控制列表控件**相关联的**发送键控件**动作。



显示键盘

显示默认操作系统触摸键盘。

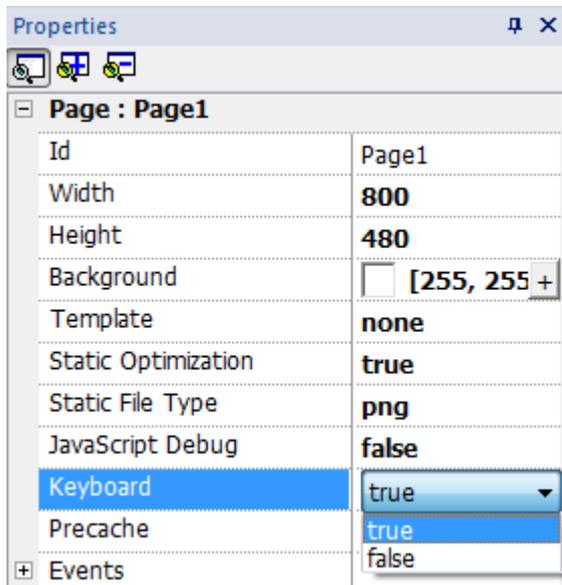


注：可能不能被所有操作系统支持。

键盘

使用外置键盘时启用/停用该动作的使用。动作执行可以在项目和页面上启用/停用。

效果与使用项目和页面属性键盘等同。



多媒体播放器动作

运行时与多媒体播放器相互作用。

动作	说明
播放多媒体	开始播放视频。
停止多媒体	关闭视频
重新载入多媒体	从开始处重新开始视频。
暂停多媒体	暂停视频
浏览多媒体	选择视频播放。

FTP 动作

用于上传文件到远程 FTP 服务器或从远程 FTP 服务器下载文件。

ftpGET

从远程 FTP 服务器下载文件。

参数	说明
FtpConfig	配置 FTP 参数
FtpRemoteFileName	要进行上传的远程 FTP 服务器上的文件名(源)
FtpLocalFileName	本地 HMI 设备上的文件名(目标)

ftpPUT

向远程 FTP 服务器上传文件。

参数	说明
FtpConfig	配置 FTP 参数
FtpLocalFileName	本地 HMI 设备上的文件名(源)
FtpRemoteFileName	要进行上传的远程 FTP 服务器上的文件名(目标)



文件名可以包含通配符。

当被传输时，系统变量和正在进行的操作状态一同更新(请参阅"[FTP客户端变量](#)"在本页 131了解详细信息)。

FTP 服务器配置

要配置 FTP 参数，请在 **FtpConfig** 设置中输入以下信息：

参数	说明
FTP 地址	FTP 服务器 IP 地址
服务器端口	FTP 连接端口(默认 = 21)。
验证	选择要使用的 FTP 身份验证： <ul style="list-style-type: none"> • 标准(需要用户名和密码) • 匿名
用户名	远程 FTP 帐户的用户名
密码	远程 FTP 帐户的密码

单击 **+** 可添加更多 FTP 服务器配置。



提示：可以动态更改 LRH SW HMI Runtime 中的服务器参数，但需要使用标签。

FTP JavaScript 界面

ftpConfig

```
ftpCONFIG(IPAddress, Port, Authentication, UserName, Password)
```

设置用于 FTP 调用的 FTP 参数

参数	说明
IPAddress	FTP 服务器的 IP 地址。
端口	FTP 连接端口(默认 = 21)。
验证	选择要使用的 FTP 身份验证： <ul style="list-style-type: none"> • 标准(需要用户名和密码) • 匿名
用户名	远程 FTP 帐户的用户名
密码	远程 FTP 帐户的密码

ftpGET

```
ftpGET (remoteFileName, localFileName, [callback])
```

从远程 FTP 服务器下载文件。

参数	说明
remoteFileName	要进行上传的远程 FTP 服务器上的文件名(源)
localFileName	本地 HMI 设备上的文件名(目标)
callback	在 FTP 传输结束时将要调用的函数

ftpPUT

```
ftpPUT (remoteFileName, localFileName, [callback])
```

向远程 FTP 服务器上传文件。

参数	说明
remoteFileName	要进行上传的远程 FTP 服务器上的文件名(源)
localFileName	本地 HMI 设备上的文件名(目标)
callback	在 FTP 传输结束时将要调用的函数

示例：

```
project.ftpCONFIG("192.168.0.200", "21", "true", "admin", "admin");
```

```
project.ftpGET( "data.txt",
               "\\USBMemory\\data.txt",
               function(ftpStatus) {fnFtpGetFinished(ftpStatus);} );

function fnFtpGetFinished(ftpStatus) {
    alert(ftpStatus);
}
```

页面动作

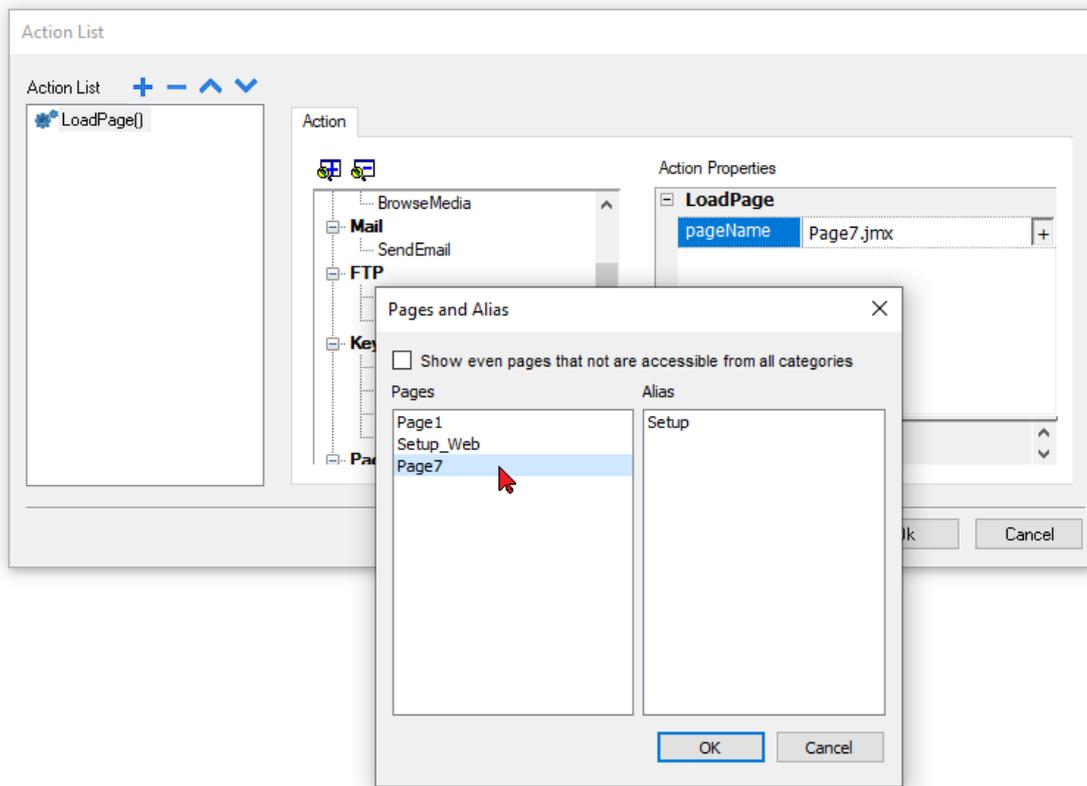
页面导航。页面动作可以在下列事件时使用：

- OnMouseClicked
- OnMouseRelease
- OnMouseHold
- 激活
- 失效
- 报警
- 计划任务

加载页面

前往项目所选页面。

从 LRH SW v4.0 开始，除了页面之外，您还可以使用别名(请参阅第 1 页的 ["别名页面" 在本页 69](#))



主页

前往主页。

您可以在 **项目控件** 的行为部分中设置主页，请参阅["项目" 在本页 80](#)

前一页

前往前一页。

下一页

前往下一页。

上一次访问的页面

前往上一个显示的页面。

显示对话框

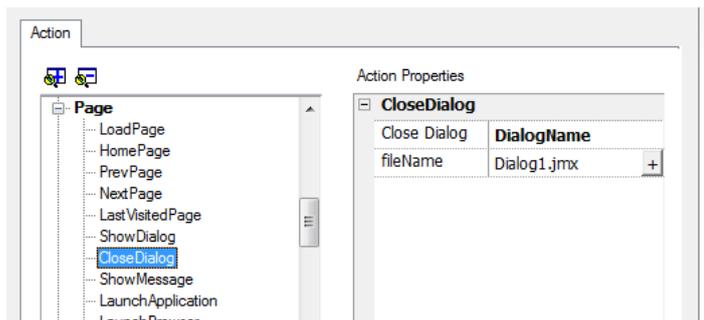
打开项目中定义的对话框页。

关闭对话框

关闭对话框页。



注：这一动作只在对话框页可用。



关闭对话框选项

选项	说明
所有	关闭所有打开的对话框
已选	仅关闭活跃对话框
对话框名称	关闭指定为 fileName 属性的对话框

JavaScript 界面

`project.closeDialog(DialogID);`

对话框ID:

所有	关闭所有打开的对话框
已选	仅关闭活跃对话框
DialogName.jmx 或 AliasName	关闭指定为 fileName 参数的对话框

例子

例如	行为
<code>project.closeDialog("All");</code>	所有打开的对话框都被关闭
<code>project.closeDialog("Selected");</code>	已选对话框被关闭
<code>project.closeDialog("Dialog1.jmx")</code>	所有对话框 1 的情况都关闭

函数 `project.closeDialog();`(无参数) 与 `project.closeDialog("Selected");` 工作方式相同。

显示信息

显示弹出的信息。输入要显示的信息文本。

启动应用程序

启动外部应用程序。

参数	说明
应用名称	有扩展名的可执行名词(例如运行记事本的“notepad.exe”)
路径	应用程序路径。
变量	应用程序特定变量(例如用于打开文档“Manual.pdf”的 <code>\flash\qthmi\Manual.pdf</code>)
单个实例	在单个事例或多个事例中启动的应用程序的变量。 当选择可单个事例时,系统首先验证程序是否已经在运行;如果是的话,程序会被调到后台,如果不是,程序将会被启动。
FlushRuntimeCache	运行应用程序之前,先清除所有 runtime 缓存以获得尽可能更多的可用 RAM。



注:带空格的参数必须加引号(例如“\Storage Card\Manual.pdf”)

示例:

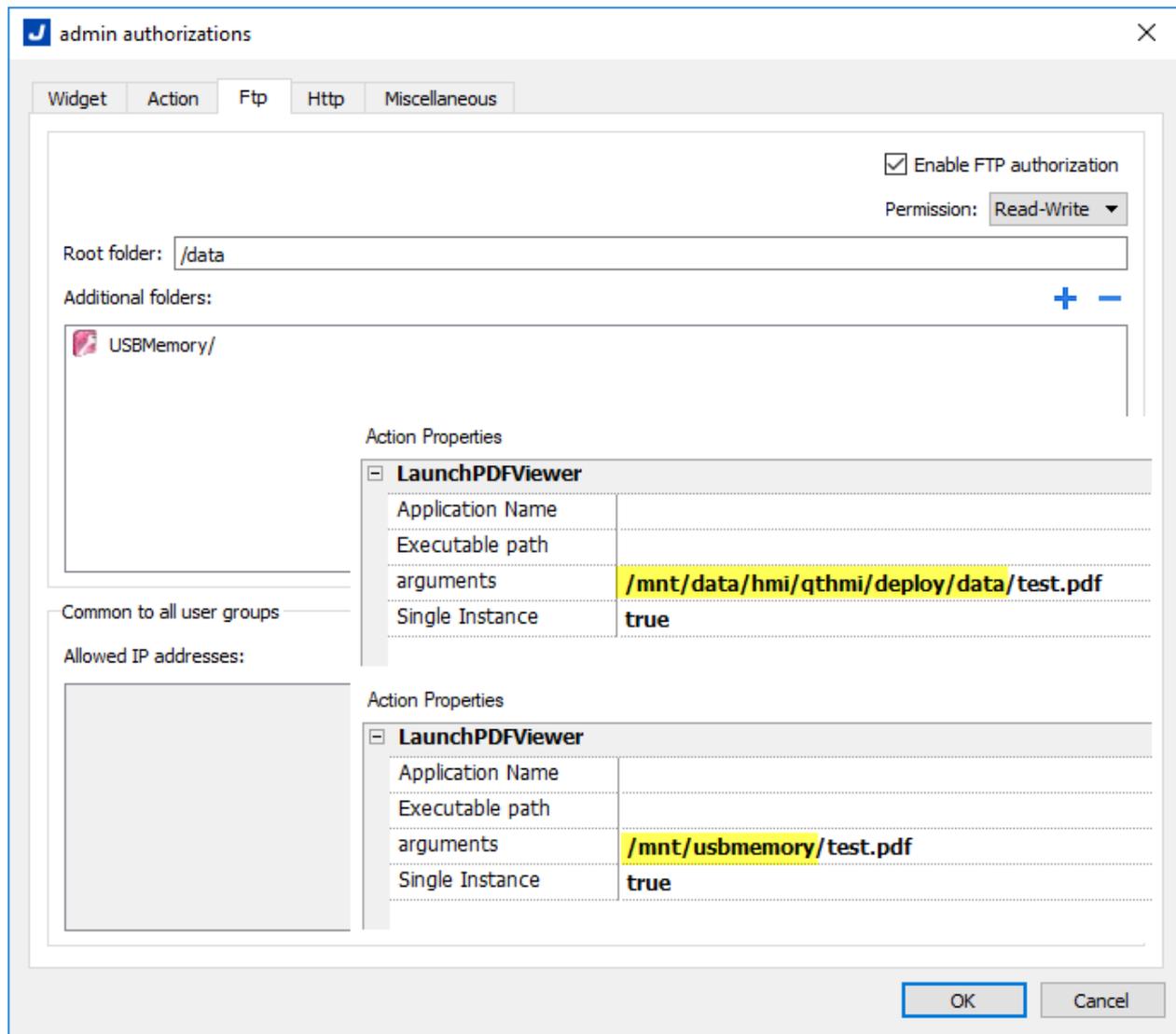
LaunchApplication	
Application Name	<code>\Windows\cmd.exe</code>
Executable path	
arguments	<code>/c "\Flash\New Folder\test.bat" Par1 Par2</code>
Single Instance	true

启动 PDF 阅读器

开启 PDF 阅读器。

请注意,路径名的语法字段采用本机 OS 格式(请参阅“HMI 设备地址”在本页 525)。

在 Linux 设备上,HMI 应用程序安装路径为 `/mnt/data/hmi/qthmi/deploy/`,且路径名的语法使用斜杠字符。



-Hide-open-button (仅在 Linux 设备上可用)

使用此选项，将从 PDF 工具栏中删除用于打开其他文件的图标，以将导航限制为已打开并通过命令行传递的 PDF 文件。

Action Properties

LaunchPDFViewer	
Application Name	
Executable path	
arguments	/mnt/usbmemory/test.pdf -hide-open-button
Single Instance	true

启动更新

从外部设备更新项目和 Runtime。

使用 **路径** 参数指定将包含更新包文件的文件夹。如果您喜欢在调用宏时手动在 HMI 设备上选择文件，请将路径参数留空。

执行 `LaunchUpdater` 宏时，HMI 设备上将显示以下对话框



JavaScript 界面

`project.launchUpdater(strPath)`

例子

```
project.launchUpdater("\\USBMemory")
```

锁定屏幕

暂时锁定触摸屏。允许清洁触摸屏。

系统变量 **解锁剩余时间** 显示解锁剩余时间。请参阅 ["屏幕变量"](#) 在本页 134

LoadProject

卸载当前项目并加载 HMI 设备中的选定项目。

必须使用相对路径来指定项目名称，如下例所示：

Action Properties	
LoadProject	
projectName	workspace/project2/project2.jpr

最近访问的项目

卸载当前项目并返回至之前的项目

打印动作

管理打印任务

打印图形报表

打印一份表格报表。

参数	说明
reportName	给报表命名
silent	False = 允许在运行时设置打印机属性
文件名	文件名(仅适用于 PDF 报表) 支持的占位符： <ul style="list-style-type: none"> • %n = 报表名称 • %p = 项目名称 • %y = 年, %M = 月, %d = 日 • %h = 小时, %m = 分钟, %s = 秒。
文件夹路径	文件夹名(仅适用于 PDF 报表) 请注意, 路径名的语法字段采用本机 OS 格式(请参阅 "HMI 设备地址" 在本页 525)。 <ul style="list-style-type: none"> • 在 Linux 设备上 USB 设备路径为"/mnt/usbmemory" "测试文件夹"将位于"/mnt/data/hmi/qthmi/deploy/testFolder"内
已签名	当输出为 PDF 文件时, 使用面板上的 x.509 生成一个已签名的文件。  在 Linux 设备上需要 BSP v1.0.507 或更高版本  在项目属性参数中定义用于签名的算法 请参阅 "项目" 在本页 80 了解可用算法 另请参阅： <ul style="list-style-type: none"> • "已签名 PDF 文件" 在本页 313

清空打印序列

清除当前打印队列。当在执行任务时执行此动作, 队列会在当前任务完成后清除。

暂停打印

使当前打印队列保持。当在执行任务时执行此动作, 队列会在当前任务完成后暂停。

继续打印

重启先前保留的队列。

取消打印

停止执行当前任务, 将其从队列清除。如果队列还有任务, 那么取消后下一个任务启动。

配方动作

用于安排配方管理。

下载配方

从HMI设备闪存中复制配方数据到控制器(例如, PLC, 局部变量, 取决于协议)。

参数	说明
配方名称	下载的配方的名称
配方集	要复制的配方集的数量。 curSet = 下载当前选择的配方集

上传配方

从控制器上保存配方数据(如:PLC、局部变量, 取决于协议)到设备闪存中。

参数	说明
配方名称	上传的配方的名称
配方集	要复制的配方集的数量。 curSet = 下载当前选择的配方集

写入当前配方集

设置选择的配方集为当前的配方集。

参数	说明
配方名称	要设定为当前配方的配方名称
配方集	要设定为当前配方集的配方集

下载当前配方

下载当前配方集数据到控制器。

未要求参数。

上传当前配方

上传控制器数据集到当前配方集。

未要求参数

复位配方

对配方数据恢复出厂设置。原始配方数据将重写上传的配方
选择您要复位为出厂数据的配方。

转储配方数据

转储配方数据到内部或外部存储。数据以 .csv 格式被保存。

参数	说明
配方名称	要转储的配方名称
文件路径	<p>目标文件夹</p> <ul style="list-style-type: none"> • 内存 = \闪存\QTHMI\工作空间\转储 • USB驱动器 = \USB存储 • SD卡 = \存储卡 • 公网 = \\<主机名或 IP>\共享路径 • 专用网络 = \\<用户名>:<密码>@<主机名或 IP>\sharePath <p> 注:支持的外部存储器格式为 FAT 或 FAT32(不支持 NTFS 格式)。</p> <p> 注:只有配备 BSP 1.0.25 及更高版本的 Linux 设备才支持专用网。</p>
文件名	<p>指定文件名的标签。</p> <p>支持以下通配符</p> <ul style="list-style-type: none"> • %r = 配方名称 • %d = 数据集名称 <p>示例: %r_%d</p>
日期时间词头文件名	true = 保存的文件的名称将以日期和时间作为词头 (例如 D2012_01_01_T10_10_recipe1.csv)
时间格式	<p>时间格式:</p> <ul style="list-style-type: none"> • Local = 输出的时间值是 HMI 设备上的时间。 • Global = 输出的时间值以 UTC 格式。

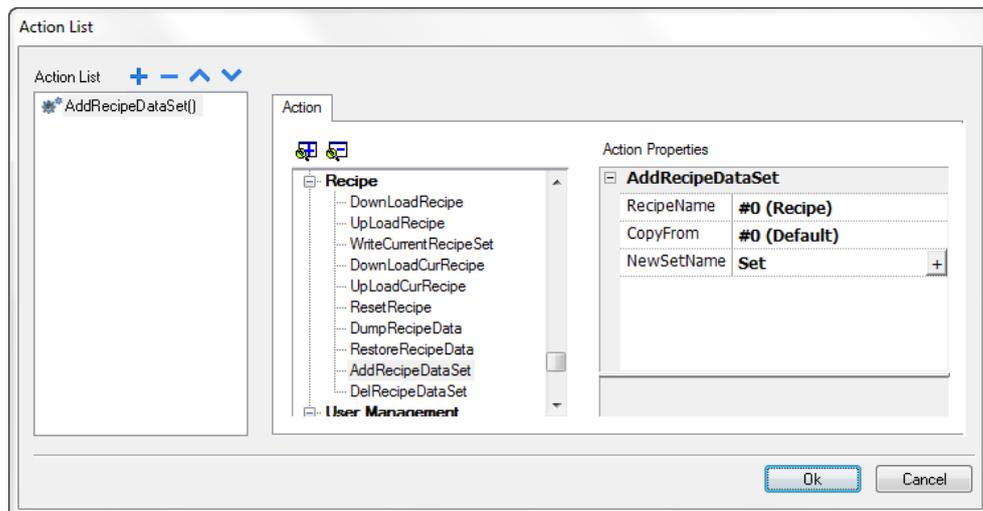
还原配方数据

复位先前保存的配方数据。

参数	说明
配方名称	要还原的配方： <ul style="list-style-type: none"> • 所有配方 所有配方数据均将更换为从外部文件中读取的数据 • CurrentRecipe 只有当前所选的配方数据将更换为从外部文件中读取的数据
RecipeDataSet	仅当配方名称= CurrentRecipe 时才适用。 选择要还原的数据集： <ul style="list-style-type: none"> • AllRecipeDataSet 所有数据集均将被还原 • 当前集 只有当前所选的数据集的数据集将被还原
还原类型	仅当 RecipeDataSet=AllRecipeDataSet 时才适用。 此参数定义要还原的文件中的数据集数量与 HMI 设备中的数据集数量不符时采取的行为 <ul style="list-style-type: none"> • 更换 设备中的所有数据集均将被删除并更换为 csv 文件中的数据集 • Match 仅更换设备中具有相同数据集 ID 的数据集 • 匹配和添加 更换设备中具有相同数据集 ID 的数据集，并添加在 csv 文件中找到的附加数据集 (注：不会从设备中删除位于设备中但不位于 csv 文件中的数据集)
文件路径	源文件夹 <ul style="list-style-type: none"> • 内存 = \闪存\QTHMI\工作空间\转储 • USB驱动器 = \USB存储 • SD卡 = \存储卡 • 公网 = \\<主机名或 IP>\共享路径 • 专用网络 = \\<用户名>:<密码>@<主机名或 IP>\sharePath <p> 注：支持的外部存储器格式为 FAT 或 FAT32(不支持 NTFS 格式)。</p> <p> 注：只有配备 BSP 1.0.25 及更高版本的 Linux 设备才支持专用网。</p>
文件名	附着的标签，运行时从该标签读取文件名。
BrowseForFile	true = 显示“打开”对话框以浏览要阅读的文件。 false = 不显示对话框，

添加配方数据组

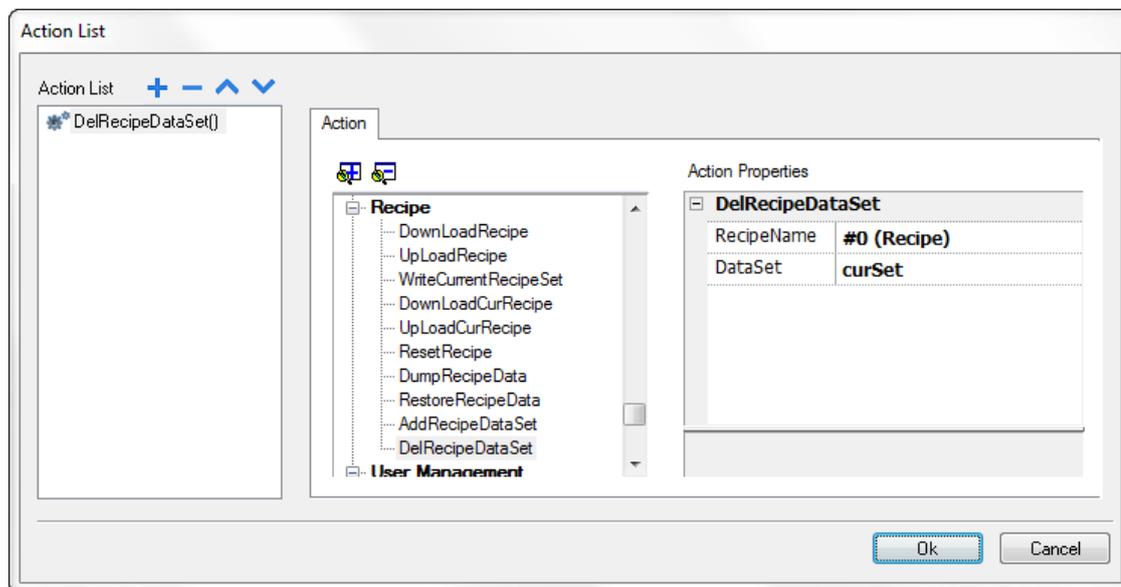
将新数据集添加至所选配方。新数据集附加至已经定义的数据集末尾。



参数	说明
配方名称	添加数据集的配方。
CopyFrom	从中复制参数值以初始化新数据集的数据集
NewSetName	新数据集名称。 这里可以使用标签引用。

删除配方数据组

从所选配方中删除数据集。删除数据集将重新安排接下来的数据集位置编号。



参数	说明
配方名称	将删除数据集的配方。
数据集	将删除的数据集。

远程客户动作

用于上传文件到远程 HMI 设备或从远程 HMI 设备下载文件。这些动作只能远程 LRH SW Client 使用，通过 FTP 访问远程文件。



重要: 启用 FTP 支持，并为用于传输文件的文件夹提供所有需要的用户权限。

上传至 HMI

打开一个文件开放对话框选择要上传到远程 HMI 设备的文件。

参数	说明
目标	文件上传的 HMI 设备上的目标路径
筛选器	文件的扩展名用逗号分隔显示(例如, *.txt)

从 HMI 下载

打开一个文件开放对话框选择要从远程 HMI 设备下载的文件。



注: 只有与设定的过滤器匹配的文件被显示，被下载。

参数	说明
源	文件下载的 HMI 设备上的源路径
筛选器	文件的扩展名用逗号分隔显示(例如, *.txt)

JavaScript 界面

```
boolean project.uploadToHMI (dirPath, strFilter);
```

```
boolean project.downloadFromHMI (dirPath, strFilter);
```

参数	说明
dirPath	文件下载/上传的 HMI 设备上的源路径
串过滤	文件的扩展名用逗号分隔显示(例如, *.txt)

返回值:

True	成功传输
False	传输失败



注:当被传输时,系统变量和正在进行的操作状态一同更新。

系统动作

用于管理系统属性。

重启

重启 runtime。

保存趋势

将历史趋势数据保存到外部驱动器(USB驱动器或SD卡)。

参数	说明
TrendName	要存储的历史趋势名称
FolderPath	<p>目标文件夹:</p> <ul style="list-style-type: none"> • 内存 = \闪存\QTHM\工作空间\转储 • USB驱动器 = \USB存储 • SD卡 = \存储卡 • 公网 = \\<主机名或 IP>\共享路径 • 专用网络 = \\<用户名>:<密码>@<主机名或 IP>\sharePath <p> 注:支持的外部存储器格式为 FAT 或 FAT32(不支持 NTFS 格式)。</p> <p> 注:只有配备 BSP 1.0.25 及更高版本的 Linux 设备才支持专用网。</p>
FileFormat	<p>Binary = 缓存以二进制格式转储(.dat 文件和 .inf 文件)。这些文件都要求用外置应用以.csv 格式转换数据。</p> <p>Compatibility CSV = 缓存以兼容版本 1.xx 的 .csv 文件格式转储到指定位置</p> <p>Compact CSV = 缓存使用更新格式以 .csv 文件转储至指定位置</p> <p>参见 "导出趋势缓存数据" 在本页 260</p>
DateTimePrefix	true = 保存的文件的名称将以日期和时间作为词头(例如 D2012_01_01_T10_10_Trend1.csv)

参数	说明
时间格式	时间格式： <ul style="list-style-type: none"> • Local = 输出的时间值是 HMI 设备上的时间。 • Global = 输出的时间值以 UTC 格式。
文件名	当 <code>DateTimePrefixFileName=true</code> 时启用 支持以下通配符 <ul style="list-style-type: none"> • %n = 趋势名称 • %y = 年 • %M = 月 • %d = 日 • %h = 小时 • %m = 分钟 • %s = 秒 示例： <code>\\%n\%y%M%d\%h%m%s</code>

仅在所选文件格式为**紧凑 CSV**时才可用的附加参数



当“选择字段”和“选择曲线”参数都为空时，.csv 文件将以旧的“紧凑 CSV”而非列选择格式转储。另请参阅“导出趋势缓存数据”在本页 260

参数	说明
选择 字段	选择转储文件内要导出的列。 可用列如下： <ul style="list-style-type: none"> • 日期和时间 • 日期 • 时间 • 值 • Quality 请注意，“附加到标签”可用于定义在运行时从 HMI 应用程序导出的列。该标签必须包含一个字符串，其中包含要导出并用逗号分隔的字段列表。 示例： <ul style="list-style-type: none"> • ""(空字符串 = 所有可用字段) • "DateTime,Value,Quality" • "Date,Time,Value"
选择 曲线	选择转储文件内要导出的曲线 请注意，“附加到标签”可用于定义在运行时从 HMI 应用程序导出的曲线。该标签必须包含一个字符串，其中包含要导出并用逗号分隔的曲线名称列表。

参数	说明	
	示例： <ul style="list-style-type: none"> 空字符串或“所有曲线”将导出所有数据集 "Name1,Name2,Name3" "Name1,Name3" 	
日期格式	选择日期和时间格式 使用“附加到标签”可以在运行时通过字符串定义日期格式	
	日期占位符	
	d	不带前导零的天数(1至31)
	dd	带前导零的天数(01至31)
	ddd	缩写的本地化日期名称(例如,“周一”至“周日”)
	dddd	较长的本地化日期名称(例如,“周一”至“周日”)。
	M	不带前导零的月份数(1-12)
	MM	带前导零的月份数(01-12)
	MMM	缩写的本地化月份名称(例如,“一月”至“十二月”)。
	MMMM	较长的本地化月份名称(例如,“一月”至“十二月”)。
	yy	以两位数表示的年份(00-99)
	yyyy	以四位数表示的年份
	时间占位符	
	h	不带前导零的小时数(如果显示上午/下午,0至23或1至12)
	hh	带前导零的小时数(如果显示上午/下午,00至23或01至12)
	m	不带前导零的分钟数(0至59)
	mm	带前导零的分钟数(00至59)
	s	不带前导零的秒数(0至59)
	ss	带前导零的秒数(00至59)
	zzz	带前导零的毫秒数

参数	说明								
	<table border="1"> <thead> <tr> <th colspan="2">时间占位符</th> </tr> </thead> <tbody> <tr> <td>Z</td> <td>毫秒</td> </tr> <tr> <td>AP</td> <td>使用上午/下午显示。AP 将替换为“上午”或“下午”。</td> </tr> <tr> <td>AP</td> <td>使用上午/下午显示。AP 将替换为“上午”或“下午”。</td> </tr> </tbody> </table>	时间占位符		Z	毫秒	AP	使用上午/下午显示。AP 将替换为“上午”或“下午”。	AP	使用上午/下午显示。AP 将替换为“上午”或“下午”。
时间占位符									
Z	毫秒								
AP	使用上午/下午显示。AP 将替换为“上午”或“下午”。								
AP	使用上午/下午显示。AP 将替换为“上午”或“下午”。								
语言	选择要使用的语言。								



注：保存趋势 (DumpTrend) 动作的执行将自动强制暂时保存在 RAM 存储器上的数据盘。查看“历史趋势控件”在本页 263 了解如何保存样本数据到磁盘。



注：连接到 USB 端口的外部驱动器必须有 FAT 或 FAT32 格式。NTFS 格式不支持。



警告：在一个文件夹内可以创建的最大文件数存在限制。具体限制取决于不同因素，不能简单地计算，您可以认为 999 是在文件夹内可以创建的最大文件数。

将二进制转储文件转化为 .csv 格式

TrendBufferReader.exe 工具存储在 LRH SW 安装文件夹的 *Utils* 文件夹中。

使用以下句法：

```
TrendBufferReader -r Trend1 Trend1.csv 1
```

其中：

Trend1 = 没有因为保存而产生扩展名的趋势缓存名称(原始文件名是 trend1.dat)

Trend1.csv = 输出文件的名称。



警告：TrendBufferReader.exe 是一个旧的实用工具，不适用于新的多标签缓冲区。不建议使用该实用工具。由于现在可以直接转储 .csv 格式的趋势缓存数据，因此该实用工具无需更多维护。

.csv 文件结构

形成的 .csv 文件有五栏

栏	说明
数据类型	样本标签的数据类型： 0 = empty 1 = boolean 2 = byte 3 = short 4 = int 5 = unsignedByte 6 = unsignedShort 7 = unsignedInt 8 = float 9 = double
值	样本的值
时间戳 (UTC)	时间戳以UTC格式
样本时间 (ms)	样本间隔时间以毫秒表示
Quality	标签值质量。根据 OPC DA 标准编码的信息以及存储在字节数据中以(8位)三位域形式的信息；Quality、Sub 状态和 Limit 状态。 八质量位按如下排列：QQSSSSL。需要所有单一域的完整和详细描述，请查看 OPC DA 官方文件。

常用质量值

HMI 获取引擎返回的最常用的质量值是：

质量编码	Quality	说明
0	BAD	该值不好但没有具体原因
4	BAD	配置的具体服务器问题。例如：标签已从配置文件 (tags.xml) 被删除。
8	BAD	此时可能没有可用的值，例如：数据来源没有提供值。
12	BAD	检测到设备失败
16	BAD	设备反应之前已过期。
24	BAD	通信失败
28	BAD	没有找到更高或更低的绑定值的趋势界面特定标签的数据。

质量编码	Quality	说明
32	BAD	未收集数据(例如:存档未激活)。 趋势界面特定记号。 该值用于指示暂时的离线状态(针对任何取样停止的情况)。
64	UNCERTAIN	无具体原因。
65	UNCERTAIN	无具体原因。 该值在某个更低的限制下被固定。
66	UNCERTAIN	无具体原因。 该值在某个更高的限制下被固定。
67	UNCERTAIN	无具体原因。 该值是恒量,不能移动。
84	UNCERTAIN	在其限定的限制以外的返回值。 此时 限制 字段指示哪个限制被超过了,但这个值可以移动到这个范围外。
85	UNCERTAIN	在其限定的限制以外的返回值。 此时 限制 字段指示哪个限制被超过了,但这个值可以移动到这个范围外。 该值在某个更低的限制下被固定。
86	UNCERTAIN	在其限定的限制以外的返回值。 此时 限制 字段指示哪个限制被超过了,但这个值可以移动到这个范围外。 该值在某个更高的限制下被固定
87	UNCERTAIN	在其限定的限制以外的返回值。 此时 限制 字段指示哪个限制被超过了,但这个值可以移动到这个范围外。 该值是恒量,不能移动。
192	GOOD	-

删除趋势

删除保存的趋势数据。

定义您要删除记录的**趋势名称**。

保存事件存档

保存历史报警记录和日志跟踪数据存储到外部驱动器,如 **USB 闪存**或 **SD 卡**。

参数	说明
EventArchive	存储数据的缓存名称
FolderPath	<p>目标文件夹</p> <ul style="list-style-type: none"> • 内存 = \闪存\QTHMI\工作空间\转储 • USB驱动器 = \USB存储 • SD卡 = \存储卡 • 公网 = \\<主机名或 IP>\共享路径 • 专用网络 = \\<用户名>:<密码>@<主机名或 IP>\sharePath <p> 注:支持的外部存储器格式为 FAT 或 FAT32(不支持 NTFS 格式)。</p> <p> 注:只有配备 BSP 1.0.25 及更高版本的 Linux 设备才支持专用网。</p>
DumpConfigFile	转储存档的描述文件
DumpAsCSV	<p>true = 缓存作为 .csv 文件存储到指定位置</p> <p>false = 缓存以二进制格式转储(.dat 文件和 .inf 文件)。这些文件都要求用外置应用以.csv格式转换数据。</p>
DateTimePrefix	true = 保存的文件的名称将以日期和时间作为词头(例如 D2012_01_01_T10_10_alarmBuffer1.csv)
timeSpec	<p>时间格式:</p> <ul style="list-style-type: none"> • Local = 输出的时间值是 HMI 设备上的时间。 • Global = 输出的时间值以 UTC 格式。
csv 列	<p>选择要转储到 .csv 文件中的列。</p> <p> 仅当 EventArchive 为报警缓存时才适用</p>

参数	说明
文件名	<p>支持以下通配符</p> <ul style="list-style-type: none"> • %n = 事件存档名称 • %y = 年 • %M = 月 • %d = 日 • %h = 小时 • %m = 分钟 • %s = 秒 <p>示例: \%n\%y%M%d\%h%m%s</p> <p> 仅当 <code>DateTimePrefixFileName=true</code> 时才适用</p>
语言	<p>选择要使用的语言。</p> <p> 仅当 <code>EventArchive</code> 为报警缓存时才适用</p>

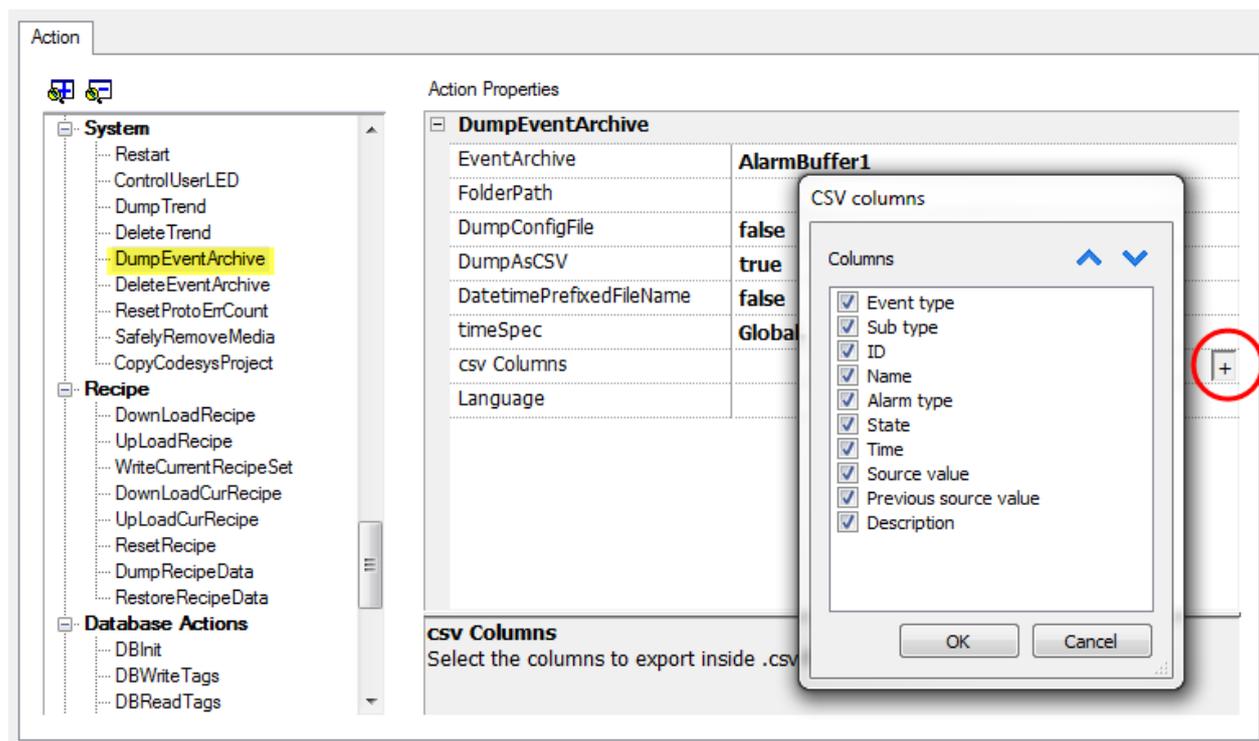
以 CSV 格式转储

`DumpAsCSV = true`

对于报警缓存，附加“csv 列”参数可以选择要在 .csv 文件中导出的列



注：只对报警缓存可用。



以二进制格式转储

DumpAsCSV = false

导出采用二进制格式的事件缓存且 **DumpConfigFile** 设为 true(推荐设置)时, 有两个文件夹:

- **data**, 包含数据文件,
- **config**, 包含 .csv 转换配置文件。

一旦这两个文件夹从 USB 驱动复制到计算机磁盘中, 文件夹结构会是:

\config\

alarms.xml

eventconfig.xml

\data\

AlarmBuffer1.dat

AlarmBuffer1.inf

\

AlarmBufferReader.exe

将存储的文件转化为 .csv 格式

AlarmBufferReader.exe 工具存储在 LRH SW 安装文件夹的 *Utils* 文件夹中。

使用以下句法:

```
AlarmBufferReader AlarmBuffer1 FILE ./AlarmBuffer1.csv
```

其中:

- AlarmBuffer1 = 保存的没有扩展名的 .dat 的名称
- AlarmBuffer1.csv = 输出文件的名称。

应用程序 AuditTrailBufferReader.exe 对检测尝试缓存可用。

使用以下句法：

```
AuditTrailBufferReader AuditTrail FILE ./AuditTrail.csv
```

其中：

- AuditTrail = 保存的没有扩展名的 .dat 的名称以及
- AuditTrail1.csv = 输出文件的名称。

删除事件存档

删除保存的事件缓存记录数据。

指定要从事件记录中删除的事件缓存的名称。

复位原始错误计数

复位协议错误计数系统变量。

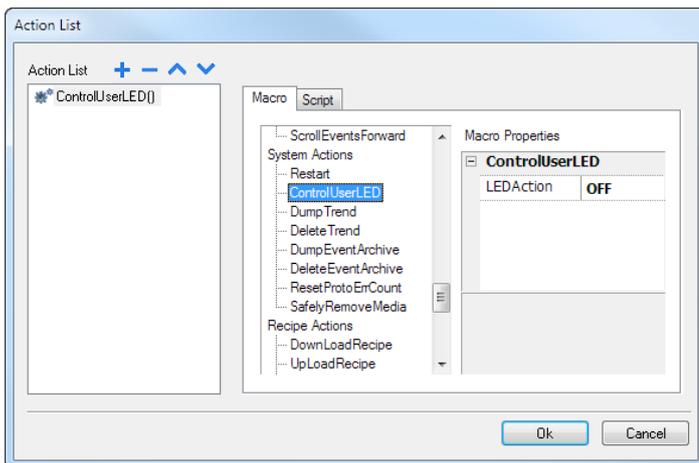
请参阅"系统变量(附加至)"在本页 125 了解详细信息。

安全移除多媒体

提供从 HMI 安全的移除 SD 卡或 USB 驱动。

控制用户 LED

设置用户 LED 行为。



Linux 平台上不可用(在 "HMI 设备地址" 在本页 525 寻找您的设备平台)

保存事件存档

将位于日志中的记录导出到签名文件。该文件签名将确保报告中的记录不会更改。

参数	说明
EventArchive	存储数据的缓存名称
FolderPath	<p>目标文件夹</p> <ul style="list-style-type: none"> • 内存 = \闪存\QTHMI\工作空间\转储 • USB驱动器 = \USB存储 • SD卡 = \存储卡 • 公共网 = \\<主机名或 IP>\共享路径 • 专用网络 = \\<用户名>:<密码>@<主机名或 IP>\sharePath <p> 注: 支持的外部存储器格式为 FAT 或 FAT32(不支持 NTFS 格式)。</p> <p> 注: 只有配备 BSP 1.0.25 及更高版本的 Linux 设备才支持专用网。</p>
文件名	<p>支持以下通配符</p> <ul style="list-style-type: none"> • %n = 事件存档名称 • %y = 年 • %M = 月 • %d = 日 • %h = 小时 • %m = 分钟 • %s = 秒 <p>示例: \%n\%y%M%d\%h%m%s</p>
格式化	<p>输出文件的格式</p> <ul style="list-style-type: none"> • CSV
已签名	<p>生成文件签名。</p> <p> 在 Linux 设备上需要 BSP v1.0.239 或更高版本</p> <p> 在项目属性参数中定义用于签名的算法 请参阅“项目”在本页 80 了解可用算法</p> <p>另请参阅:</p> <ul style="list-style-type: none"> • “已签名 CSV 文件”在本页 311
时间格式	<p>时间格式:</p> <ul style="list-style-type: none"> • Local = 输出的时间值是 HMI 设备上的时间。 • Global = 输出的时间值以 UTC 格式。

参数	说明
PeriodMode	定义要导出的时间窗口 <ul style="list-style-type: none"> • 所有事件 • 今天 • 昨天 • 上周 • 上月 • 本周 • 本月 • 自定义 将显示附加参数“periodFrom”和“periodTo”
单独的日期和时间	使用两个单独的日期和时间列
日期格式	选择日期和时间格式

签名文件

当“签名文件”参数为 **true** 时，除 **filename.csv** 格式之外还将添加两个文件：

- **filename.csv.sign**
该文件签名将确保 **filename.csv** 文件中的记录尚未更改
- **ssl-HMI.crt**
验证报告真实性所需的 HMI 设备的证书副本。

Name	Date modified	Type	Size
 AuditTrail-1413.csv	28/03/2018 16:13	Microsoft Excel Comma Separated Values File	1 KB
 AuditTrail-1413.csv.sign	28/03/2018 16:13	SIGN File	1 KB
 ssl-HMI.crt	28/03/2018 16:16	Security Certificate	2 KB

有关该证书以及如何验证签名文件的更多信息，请参阅“[X.509 证书](#)”在本页 309。

有关导出信息的更多信息，请参阅“[导出日志到 .csv 文件](#)”在本页 335。

日志消息

将消息添加到日志缓冲区中。

通过该宏，开发人员可以决定跟踪日志中的一些事件(例如，按下按钮、激活页面时)。通过附加到标签，支持在运行时定义要记录的消息。

参数	说明
EventArchive	添加消息所在的日志缓冲区的名称
消息	要在日志缓冲区中添加的消息

删除旧文件

该宏可删除给定天数的旧文件。

在 PC 中，没有路径使用限制。在面板中，允许在动态媒体和数据分区中使用（以及 Linux 中的）



开发人员负责配置该应用程序以免删除系统文件。

参数	说明
FolderPath	搜索要删除的文件所在的文件夹
FileTypes	要删除的文件列表，用逗号分隔。支持通配符 示例：*.png,*.jpg
OlderDays	最小天数(无更改)

标签动作

与标签交互。

数据传输

相互之间交换数据：

- 两个控制器，
- 在一个控制器中注册，
- 从系统变量到控制器，
- 从控制器到系统变量

各种标签类型包括一个控制器标签、一个系统变量、一个配方标签和控件属性。

切换位

切换标签位的值。

字节索引让您能够选择要切换的位：切换要读-修改-写操作；读取的值被颠倒，然后被写入到标签。

设置位

设置选定的位为“1”。

字节索引使您可以选择标签内的字节位置。

复位位

复位选择的位为“0”

字节索引使您可以选择标签内的字节位置。

写入标签

向控制器写入常数。指定标签名称和值。

步骤标签

增加或减少标签值。

参数	说明
标签名称	要增加/减少的标签的名称
步骤	步骤值
步进不得超出限制	启动步骤限制
步骤限制	步骤限制的值, 如果可用的话。

BiStep

此动作类似于步骤标签动作, 但是方向递增/递减由滚轮转动自动选择。顺时针转动滚动时, 标签值将增加。逆时针转动滚动时, 标签值将减小。

属性	说明
标签名称	在其上执行 BiStep 标签动作的标签的名称
步骤	在每次滚轮转动时要添加到标签/从标签中减去的值(取决于事件步骤属性)
事件步骤	此属性允许选择是否在每个滚轮步骤或每个转动事件中添加/减去步骤值。 false = 在每个转动事件中将步骤值添加到标签/从标签中减去步骤值。 示例: 在单个事件中转动执行 5 个滚轮步骤的滚轮, 标签将增加/减小 1。 true = 在每个滚轮步骤中将步骤值添加到标签/从标签中减去步骤值。 示例: 在单次转动中转动执行 5 个滚轮步骤的滚轮, 标签将增加/减小 5。
步进不得超出限制	如果为 true, 启用下限和上限(代表标签因 BiStep 标签动作可假设的下限值和上限值)
下限	如果“步进不得超出限制”为 true, 此属性代表标签因 BiStep 标签动作可假设的下限值
上限	如果“步进不得超出限制”为 true, 此属性代表标签因 BiStep 标签动作可假设的上限值



仅在 OnWell 动作中可用

激活组

强制标签组更新。

如果在标签编辑器中设定为激活的, 标签不论是在当前页使用还是连续使用都被更新。该动作强制一个组的所有标签连续更新。

解除激活组

解除激活标签组, 即停止强制更新标签组。

使能节点

启用/停用离线节点管理动作。对停用节点不进行任何通信。

参数	说明
协议 ID	所选协议的唯一标识符
节点 ID	所选协议的节点标识符。可以附在一个标签上。
启用	节点通信状态： False = 禁用 True = 启用 附加至标签后，标签 = 0 意味着 False

BACnet 清除优先级

请参阅“Communication Drivers”文件夹中的 BACnet 手册，了解 BACnet 动作的详细描述。

BACnet 清除所有优先级

请参阅“Communication Drivers”文件夹中的 BACnet 手册，了解 BACnet 动作的详细描述。

BACnet 设置优先级

请参阅“Communication Drivers”文件夹中的 BACnet 手册，了解 BACnet 动作的详细描述。

清空保持型存储区

设置为 0 时，清除保留内存的内容。

强制读取标签

从远程控制器强制刷新指定标签。

趋势动作

用于实时数据趋势和历史趋势控件。

刷新趋势

刷新趋势窗口

这可以在任何趋势/图表控件中使用。指定控件作为该动作的参数。

向左滚动趋势

向左滚动趋势窗口，滚动页面持续时间的十分之一 (1/10)。



注：实时趋势使用 **暂停趋势** 动作暂停趋势，否则窗口将会连接切换到当前值。

向右滚动趋势

向右滚动**趋势**窗口，滚动页面持续时间的十分之一 (1/10)。



注：实时趋势使用**暂停趋势**动作暂停趋势，否则窗口将会连接切换到当前值。

趋势向左翻页

滚动**趋势**窗口一页。例如，如果页面大小是 10 分钟，则使用**趋势向左翻页**动作向左滚动趋势 10 分钟。

趋势向右翻页

滚动**趋势**窗口一页。例如，如果页面大小是 10 分钟，则使用**趋势向右翻页**动作向右滚动趋势 10 分钟。

趋势页面时间宽度

设置**趋势**窗口的页面持续时间。

设定趋势的名称和页面的持续时间。



注：您可以在运行时使用组合框控件设置页面持续时间。

放大趋势

减少页面持续时间。

缩小趋势

增加页面持续时间。

复位趋势缩放

复位缩放水平回到原始缩放水平。

放大趋势 Y 轴

缩短 Y 轴。

缩小趋势 Y 轴

扩展 Y 轴。

复位趋势缩放 Y 轴

将 Y 轴缩放水平复位回原始缩放水平。

暂停趋势

停止在**趋势**窗口中绘制趋势曲线。

当与实时趋势同时被使用时，在曲线到达图表右边的边缘时作图停止。该动作不会停止趋势记录。

恢复趋势

如果暂停了，重新开始趋势作图。

显示趋势光标

显示X轴上指定的点上曲线的值。

这会激活趋势光标。指针(纵线)会在趋势控件中显示。

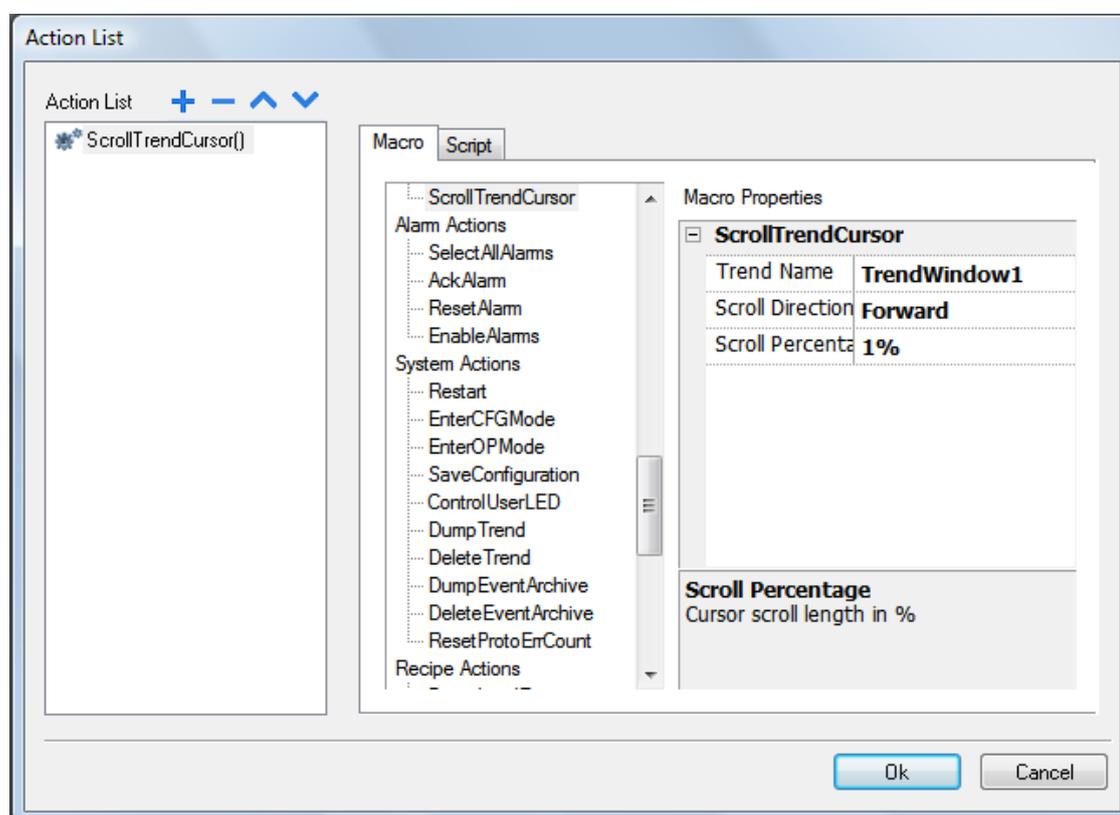
当图表指针启动时，趋势滚动停止。

滚动指针动作使图表指针在曲线上移动，或在整个**趋势**窗口上移动。

滚动趋势光标

向后或向前滚动趋势光标。

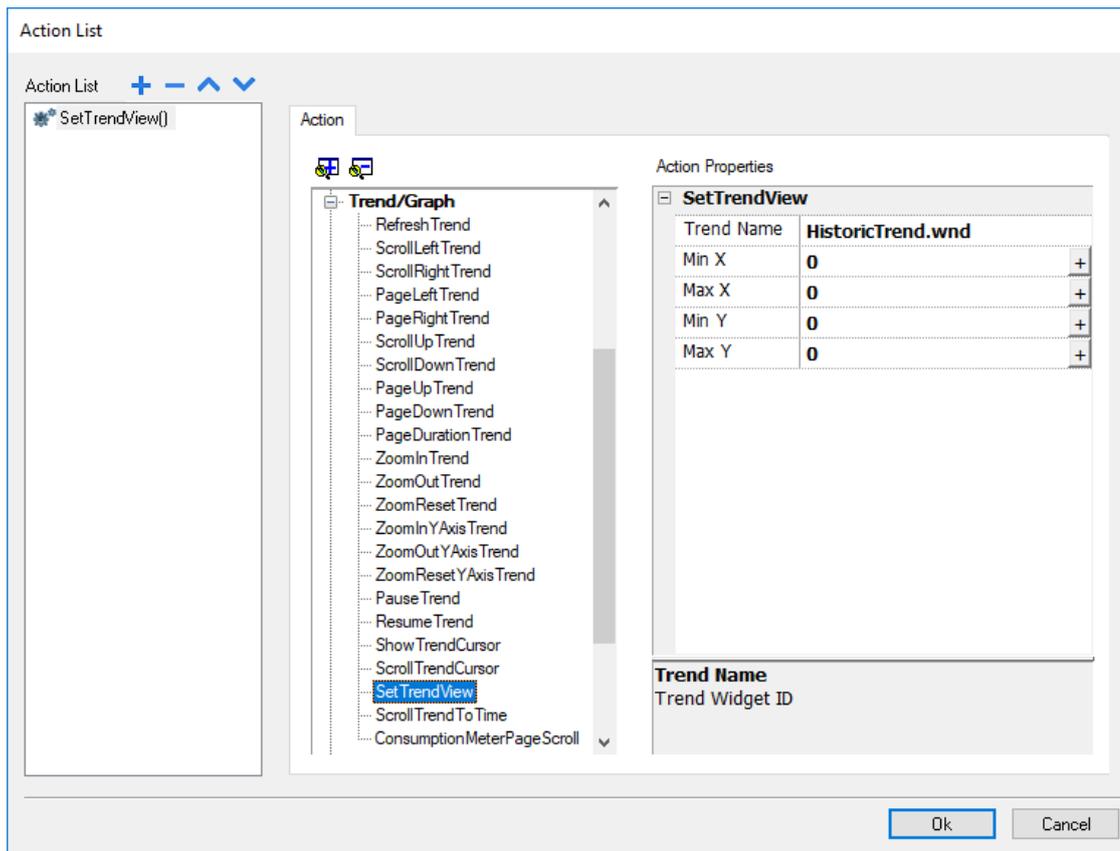
Y指针值将显示指针的点的趋势值。滚动的比率可以设定为1%或10%。百分比根据趋势窗口持续时间计算。



设置趋势视图

使用此宏可更改趋势视图的轴范围。

当 $\text{Min X} = 0$ 和 $\text{Max X} = 0$ 时，将使用在控件属性内定义的静态值。对Y轴同样如此。



滚动趋势到时间

将**趋势**窗口滚动到一个特定的时间点。

当您在特定事件出现需要滚动趋势窗口到一个特定位置时，使用该动作。

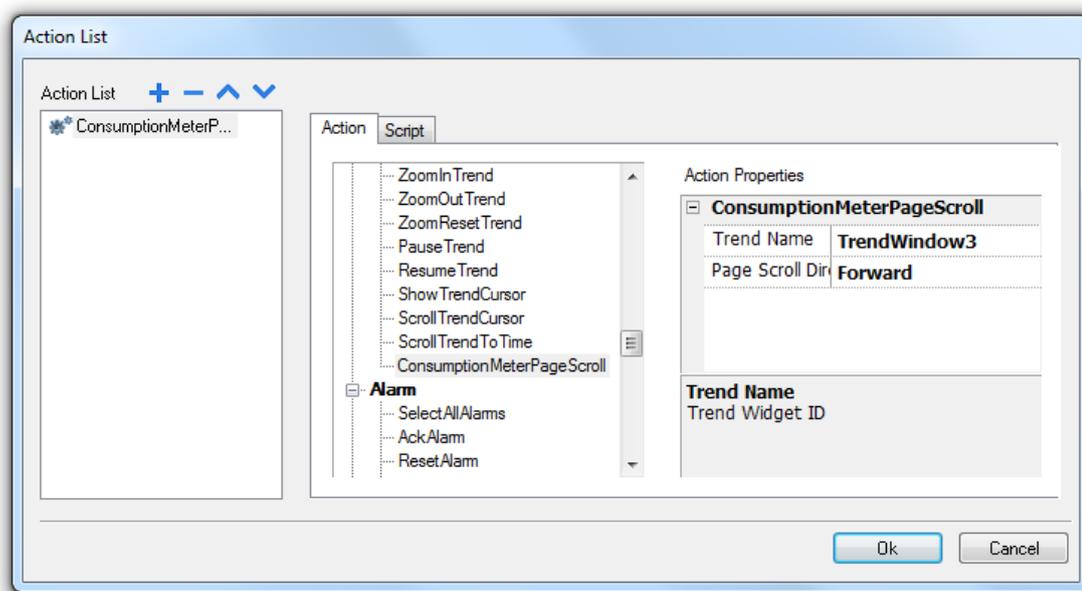
例如

1. 为一个事件(例如，报警)配置一个动作，执行将系统时间的数据传输到标签。
2. 选择该标签作为**滚动趋势到时间**参数:趋势窗口将会在事件被触发时成为中心。

滚动耗量表页面

耗量表控件向后或向前滚动页面。

参数	说明
趋势名称	趋势控件 ID(例如 TrendWindow3)
页面滚动方向	页面滚动方向(向前/向后)



文本编辑器动作

用于与文本编辑器控件交互的宏。

请参考"[TextEditor 控件](#)"在本页 434 了解详细信息

用户管理动作

用户管理和安全设置。

注销

注销当前用户。默认用户随后会自动登录。如果未配置默认用户，登录窗口将显示。

切换用户

在两个用户之间切换而不注销登录的用户：用户登录对话框出现。用户可以单击**返回**返回到先前登录的用户。

User name:

Password:

Show password

服务器继续以先前登录的用户运行，直到下一个用户登录。系统中始终有一个登录的用户。

更改密码

更改用户密码: 出现对话框

未要求参数。

重设密码

为当前用户恢复原始密码和该项目中指定的设置。

未要求参数。

添加用户

为具有可以管理其他用户属性设置的用户保留。

在运行时添加一个用户: 出现一个对话框。

User name:	<input type="text" value="user3"/>	
Password:	<input type="password" value="*****"/>	<input type="checkbox"/> Show password
Group:	<input type="text" value="admin"/>	
Comments:	<input type="text"/>	

Password must contain number:	<input type="checkbox"/>
Password must contain special character:	<input type="checkbox"/>
User must change his initial password:	<input type="checkbox"/>
Enable logoff time:	<input type="checkbox"/>
Inactivity logoff time:	<input type="text" value="0"/> min

删除用户

为具有可以管理其他用户属性设置的用户保留。

在运行时删除一个用户: 出现一个对话框。

未要求参数。

User name:	<input type="text" value="admin"/>
Group:	<input type="text" value="admin"/>

编辑用户

为具有可以管理其他用户属性设置的用户保留。

编辑用户设置。

User name:

Password: Show password

Group:

Comments:

Password must contain number:

Password must contain special character:

User must change his initial password:

Enable logoff time:

Inactivity logoff time: min

删除UM动态文件

删除动态用户管理文件。运行时对用户设置进行的更改被擦除。原始设置从项目信息中恢复。

未要求参数。

导出用户

将用户设置导出至加密 .xml 文件 (*usermgnt_user.xml*) 中，需要时可还原。

为导出文件指定目标文件夹。



重要: 用户文件已加密，无法编辑。



注: 支持的格式是 FAT 或 FAT32。NTFS 格式不支持。

导入用户

从先前保存的 .xml 导出文件 (*usermgnt_user.xml*) 中导入用户设置。

为导入文件指定源文件夹。



注: 支持的格式是 FAT 或 FAT32。NTFS 格式不支持。

控件动作

显示控件

显示或隐藏页面控件。

属性	说明
控件	要显示/隐藏的控件

滑动控件

显示控件、或控件组的滑动效果。



注: 控件或控件组可以在项目页面的可视范围之外, 通过滑动进入或离开视野。

属性	说明
控件	滑动的控件
方向	滑动方向
速度	滑动控件的转换速度
X 距离	以像素为单位在X坐标上移动距离
Y 距离	以像素为单位在Y坐标上移动距离
滑动限制	启用/停用x,y坐标上的控件移动限制
X 限制	x 坐标上的滑动限制位置
Y 限制	y 坐标上的滑动限制位置
Toggle Visibility	在每一次滑动结尾显示/隐藏控件
图片	在滑动过程中显示的图片

开始数据输入

显示键盘并开始和数据栏位输入数据, 且部接触控件本身。通过使用条形码扫描仪, 该动作可以用来激活数据输入。

Java Script 接口

```
project.beginDataEntry(wgtName [, pageName])
```

参数	说明
wgtNameWidget	控件名称
pageName	数据输入活动页面。可选参数。对于在非程序活动的对话框内选择数据栏位有用。

启动IP摄像头

从IP摄像机捕捉图片。只有在包含IP摄像机控件的页面有效。

移动IP摄像头

向支持命令的摄像机发送远程命令。查看"[IP 摄像机控件](#)"在[本页 407](#)。确保IP摄像机支持移动命令。

刷新事件

刷新[报警历史](#)控件的事件缓存。请参阅"[报警历史控件](#)"在[本页 239](#)了解详细信息。

上下文菜单

显示上下文菜单

如果项目控件的上下文菜单属性已设为**延迟**，则上下文菜单在接触屏幕背景区域几秒钟后才会出现。参见"[项目属性](#)"在[本页 73](#)

替换多媒体

用USB/SD卡中的新文件替换已有的多媒体文件。可以用来替换多媒体播放器控件中的视频文件或项目图片。



注：新的多媒体文件必须与要替换的文件有相同的名称和格式。

参数	说明
Media Type	要更新的文件类型
设备	提供新的多媒体文件的设备
sourcePath	新的多媒体文件存储的文件夹(例如, "\USB 存储")
图片调整大小	将新的图片的大小改变到要被替换的图片的尺寸。对视频文件不可用。
静默	自动替换多媒体。按照默认, 显示一个对话框, 让用户指定文件位置。

Java Script 接口

```
void replaceMedia(var sourcePath, var bSilent, var Device, var nMediaType, var bResize)
```

```
project.replaceMedia("Images", true, "\USBMemory", 1, true);
```

滚动表

向前或向后滚动表中的行

参数	说明
表格控件	表格控件名称
方向	要跳转的行的数量, 正数时向前跳转, 负数时向后跳转。

Java Script 接口

```
page.getWidget(TableWgt).scrollTo(Direction);
```

ShiftTableDataSrcColumns

数据表的列向左或向右移动。请注意，重新映射将应用于数据源控件。

参数	说明
数据源控件	数据源控件 ID
列移动	数据源控件列(向左或右,取决于签名)的移动量为
固定左列	自定义列数(表格左侧)可以在移动期间保持固定
重新映射筛选器	表控件筛选器(如果已定义)连接到数据源控件列。默认情况下,此列不会通过移动动作重新映射,但可以强制为 true

Java Script 接口

```
var ColumnOrder = [0,1,2,3,4,5,6,7,8,9,10];
var json = {_c:ColumnOrder};

page.getWidget("TableDataSrcWgt").remapColumns(json);
```

ResetTableDataSrcColumns

恢复原始的列顺序(请参阅“ShiftTableDataSrcColumns”宏)

SetTableSortingColumn

选中某列以及用于对表格的行进行排序的标准。

参数	说明
表格控件	表格排序
数据源列	用于对表格排序的列
模式排序	可以是:升序、降序或切换
按类型排序	可以是:字母或数字

Java Script 接口

```
var column = "Column1"; // Colum name (TableDataSource)
var mode = 0; //0=Ascending, 1=Descending
var type = 1; //0=Aphabetic, 1=Numeric

var sorting_rule_1 = { _c : column, _m : mode, _t : type };
var json = [ sorting_rule_1 ]
page.getWidget("TableWgt").setSortingRules(json);
```


15 LRH SW Client

LRH SW Client 是一个独立的应用程序，提供对 LRH SW HMI Runtime 的远程访问，且包含在 LRH SW 中。LRH SW Client 使用与 HMI 设备中的 Runtime 相同的图表渲染系统，一个指定的 LRH SW HMI Runtime 作为其获得实时数据的服务器。

LRH SW Client 充当远程客户端与服务器通信，与通过通信协议维护和更新的标签值共享本地可视化信息。



HMI项目包含指示当前显示在 HMI上的页面的属性，可以强制 HMI 切换到特定页面。您可以使用这些属性使显示在 HMI 设备和 LRH SW Client 上的页面同步，或者使用这些属性通过 PLC 控制 HMI 设备。请参阅“项目”在本页 80 了解详细信息。



要避免意外行为：

- 确保使用同一版本的 LRH SW HMI Runtime
- 使用“FreeType 字体渲染”，确保在 HMI 客户端和 HMI 设备上使用相同的字体渲染引擎(请参阅“Runtime”在本页 74)

PC 上的客户端应用程序	216
HMI 上的客户端应用程序	217
设置和时区选项	218

PC 上的客户端应用程序

要在 PC 上运行 LRH SW Client 应用程序，请执行以下操作：

1. 从开始菜单中 > **LRH SW > LRH SW Client**: 客户端在类似浏览器的窗口中打开。
2. 在地址栏输入服务器 / 设备 IP 地址 (例如: <http://192.168.1.12>) : LRH SW Client 将会连接到服务器，并且设备上运行的同一图表应用程序将加载到客户端窗口中。

客户端应用程序工具栏



元素	说明
HMI 服务器地址	远程 HMI 设备的 IP 地址 (例如 192.168.0.1:80)
连接状态	网络请求状态。数据交换时显示红色。
从缓存重新加载	重新加载项目
书签	书签优先页并加载这些页。
设置	打开设置对话框

重新加载选项

选项	说明
F5	从高速缓存重新加载项目
Shift + F5	下载项目到客户端

将文件传输到远程 HMI 设备

您可以上传文件到远程 HMI 设备或从远程 HMI 设备下载文件。这些动作只能远程 LRH SW Client 使用，并通过 FTP 访问远程文件。

请参阅“远程客户动作”在本页 189 和“远程客户变量”在本页 133。



重要: 启用 FTP 支持，并为用于传输文件的文件夹提供所有需要的用户权限。

工作区

项目文件从设备上传，存储在 LRH SW Client 的以下高速缓存文件夹中。

`%appdata%\Lovato\[build number]\client\cache`

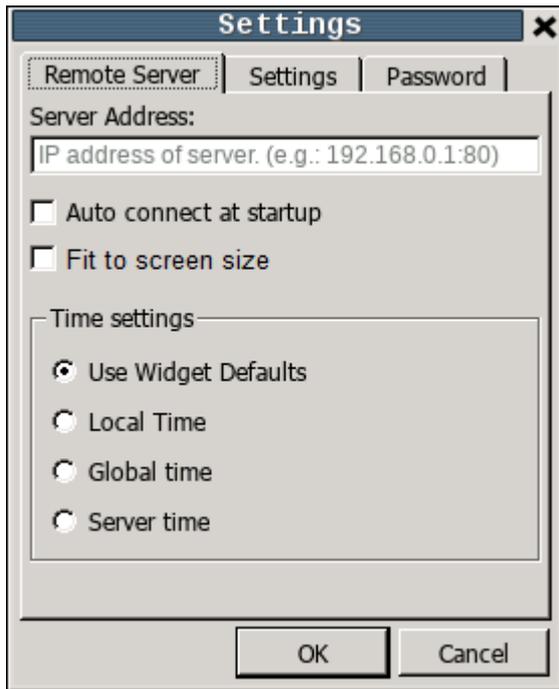
其中：

`[build number]` = 以“build number”命名的文件夹，例如 01.90.00.608。

HMI 上的客户端应用程序

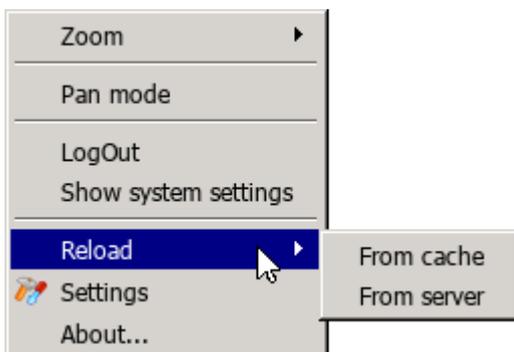
要在 Linux HMI 设备上运行 LRH SW Client 应用程序，请执行以下操作：

1. 从 **运行 > 更新包** 菜单中，创建更新包并在 HMI 设备中安装 HMI 客户端应用程序(请参阅“更新包”在本页 94 了解更多信息)
2. 在启动 HMI 设备时可用的“设置”对话框中输入服务器/设备 IP 地址(例如：`http://192.168.1.12`)：HMI 客户端将会连接服务器，与在设备上运行的相同的图表应用程序将在客户端窗口被加载。



上下文菜单

通过右键单击可用的上下文菜单，将显示以下命令：

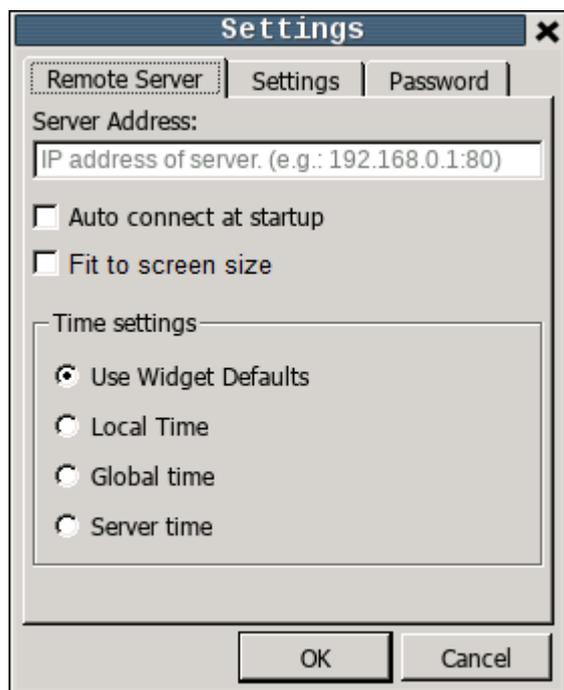


选项	说明
缩放	运行时选择视图尺寸 <ul style="list-style-type: none"> • 放大 • 缩小 • 缩放 100%
窗格模式	放大后启用/停用平移模式
注销	注销当前用户。
显示系统设置	允许显示 HMI 设置并管理系统组件。请参阅 "系统设置" 在本页 527 了解详细信息。
重新加载	重新加载远程项目 <ul style="list-style-type: none"> • 从缓存 • 从服务器
设置	打开 HMI 客户端设置。请参阅 "设置和时区选项" 向下了解详细信息  可以用密码保护
关于	显示有关 HMI 客户端版本的信息。

设置和时区选项

在设置对话框中，您可以配置客户端设置，决定如何显示项目时间戳信息。

远程服务器



连接设置

参数	说明
服务器地址	远程 HMI 设备的 IP 地址(例如 192.168.0.1:80)
启动时自动连接	当面板启动时,使用服务器地址尝试自动连接到远程服务器。
缩放至屏幕大小	将视图调整为屏幕大小

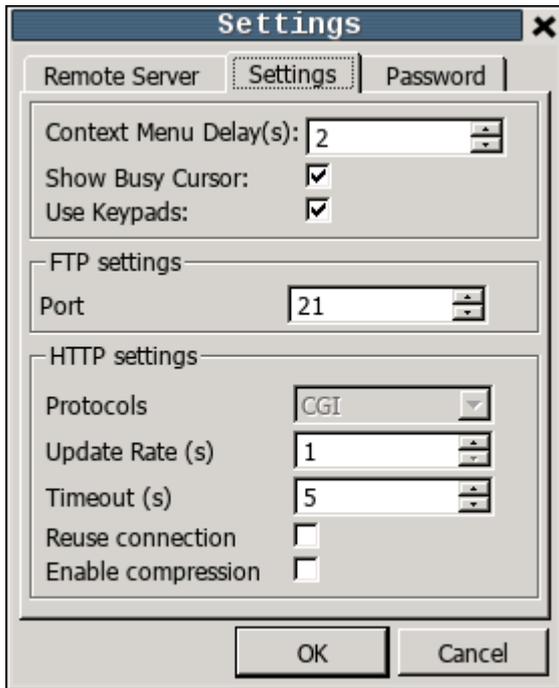
时间设置

参数	说明
使用控件默认值	根据控件设置显示时间信息。
本地时间	将项目所有时间戳上的时间转化为客户端安装的计算机的本地时间。
全球时间	将项目中所有时间戳转化为 UTC 格式。
服务器时间	将项目中所有时间戳转化为与 HMI 设备 / 服务器的时间以显示相同的时间。



重要: 确保您设置了 HMI RTC 正确时区和 DST 选项。

设置



接口设置

参数	说明
上下文菜单延迟	上下文菜单激活延迟。范围：1–60 秒。
显示忙碌光标	当系统忙碌时显示一个沙漏
使用键盘	当用户触及数字输入字段时显示键盘。 当外接 USB 键盘连接到设备时设置为禁用。

FTP 设置

参数	说明
端口	FTP 通信端口

HTTP 设置

参数	说明
协议	LRH SW Client 用于与 HMI 设备进行通信的通信协议。
更新率	从服务器同步数据的轮询频率。默认 = 1 s。
超时	请求的最长等待时间由 LRH SW Client 重复。默认 = 5 s。
重复使用连接	启动对多个 HTTP 请求重新使用相同的 TCP 连接，减少网络通信量。

参数	说明
	 注:当启用时,如果代理服务器没有立即终止过去的请求,造成连接接口饱和,该选项可能造成恢复时间很长。这是 3G 连接通常出现的情况。
启用压缩	压缩数据减少下载次数。默认 = 已停用。  注意:启用该选项可能引起 CPU 过度消耗。
时间设置	由客户端使用来修改控件时间戳信息。

密码



通过此对话框,您可以更改 admin 用户的 HMI 设备内部密码(默认密码为“admin”)。
 密码保护在 PC 版本的 HMI 客户端上不可用

16 使用集成的FTP服务器

LRH SW HMI Runtime 系统使用集成的 FTP 服务器。

使用标准 FTP 客户端应用程序连接到 HMI 设备 FTP 服务器。FTP服务器默认在标准的端口 21 响应。



重要:该服务器一次只支持一个连接;如果使用多连接 FTP 客户端,请在客户端程序上禁用此功能,或将每次会话中连接的最大值设为 1。

FTP 设置

FTP 默认凭据

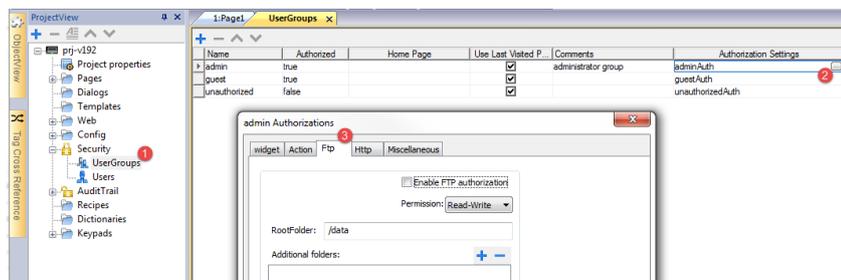
当用户管理 / 安全性被停用,使用以下凭据以进行输入连接:

用户名	admin
密码	admin

更改FTP设置

路径:项目视图 > 安全性 > 用户组 > 授权设置

您可以在管理员授权对话框的 FTP 选项卡中更改 FTP 权限和帐户信息。



请参阅["配置组和授权"](#)在本页318了解详细信息。

17 VNC 远程访问

VNC 是一个远程控制软件，使您能够使用本地鼠标和键盘远程查看和控制 HMI 应用程序。

远程访问对管理和技术支持特别有用。要使用它您需要：

- 启动 HMI 设备上的一个服务器
- 在远程设备上安装一个查看器

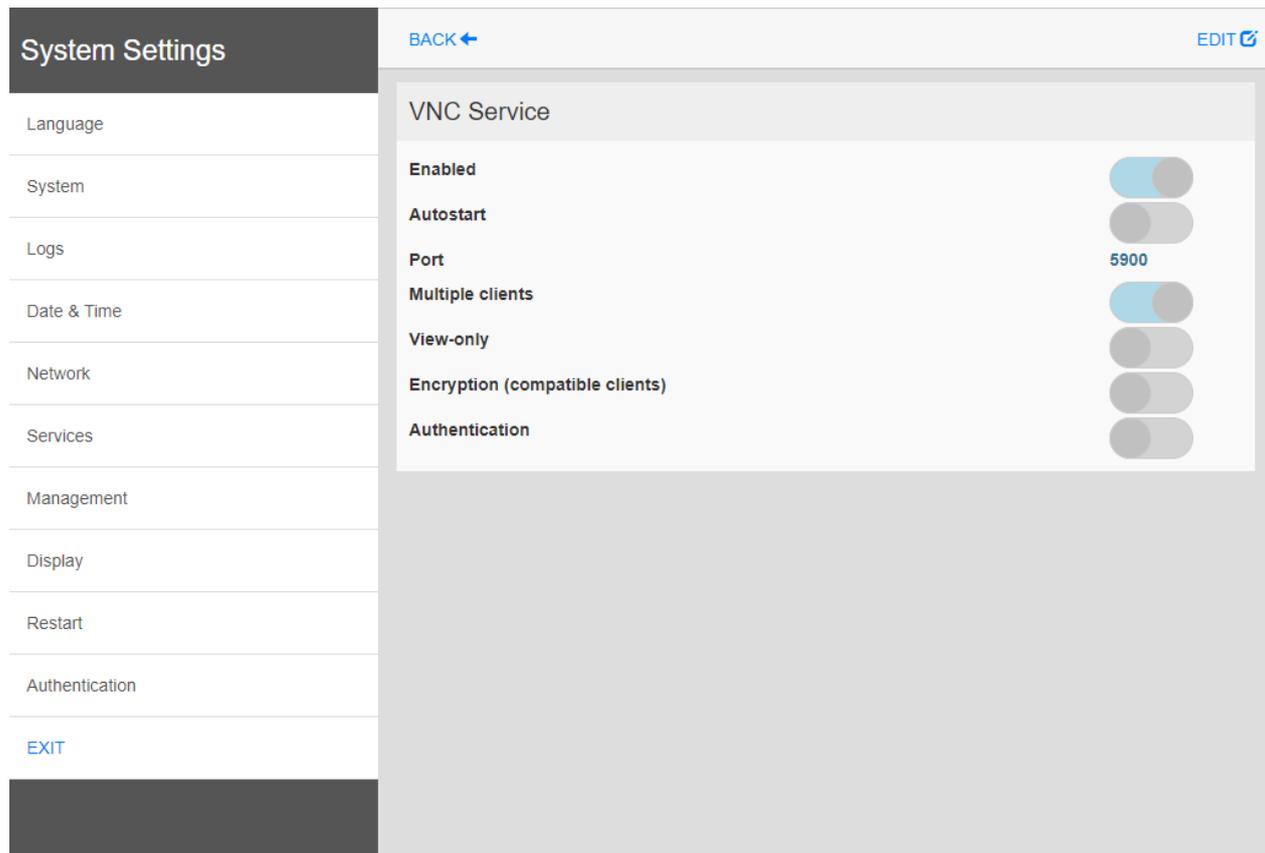
在 Linux 设备上启动 VNC 服务器	226
启动 VNC 查看器	226

在 Linux 设备上启动 VNC 服务器

VNC 服务器是嵌入到 BSP 中的一种服务，可以从设备“系统设置”的“服务”选项卡中激活。请参阅“系统设置”在本页 531 了解详细信息。



使用后应禁用 VNC，不建议自动启动。



启动 VNC 查看器

LRH SW 不提供 VNC 查看器。

许多兼容的 VNC 查看器可以免费下载(例如 TightVNC)。

18 报警

报警处理系统设计的目的是通过弹出信息显示警示信息，暗示处于控制下的系统中的任何异常情况或故障。无论什么时候位改变，或标签的值超过报警配置中设定的界限，信息便会显示。也可以编辑特定的动作程序，当报警启动时执行。



重要:无默认动作与任何报警相关联。

您可以先定义报警在 HMI 设备上如何显示，是否需要用户确认，以及如何将其记录到事件列表中。

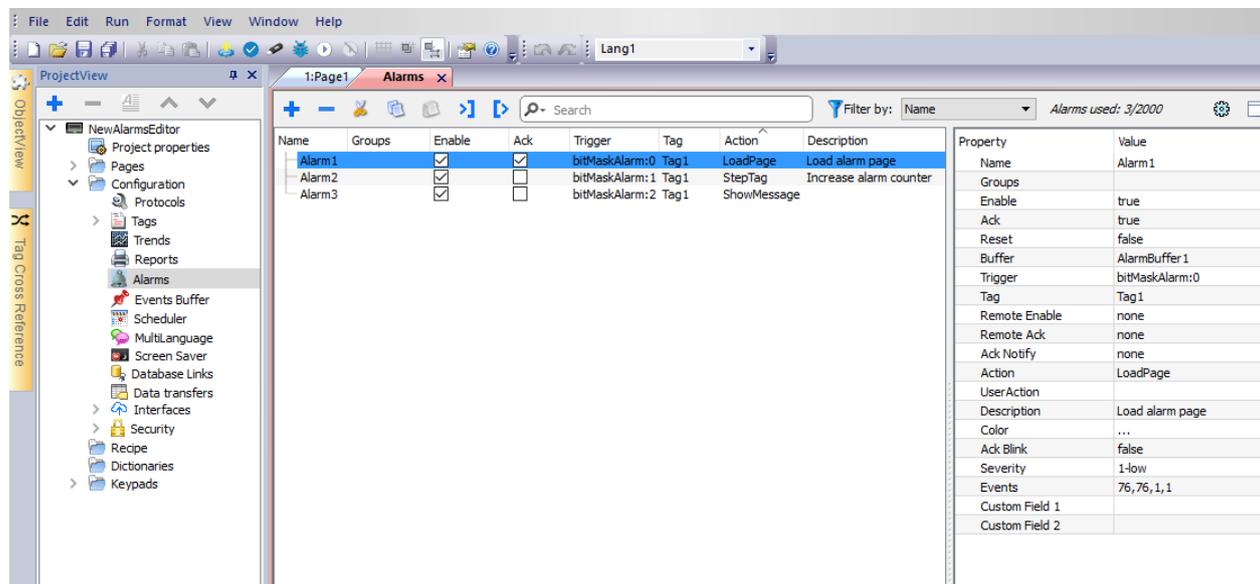
报警在报警配置编辑器中配置，因此，对于此项目的所有页面都可用。一个报警控件可以在大小合适的情况下一次显示一个以上报警。您可以在事件中启动打开或关闭报警窗。

对报警的操作与对其他任何事件的操作是一样的。您可能不希望在触发报警时显示对话框，并且可以将其同其他任何可用的动作关联。

报警编辑器	228
远程报警确认	231
报警状态机	231
设置 事件	232
激活报警控件	234
报警历史控件	239
其他报警控件	239
运行时管理报警	240
运行时启用/停用报警	240
显示实时报警数据	241
输出报警缓存到 .csv 文件	243
导出报警配置	243

报警编辑器

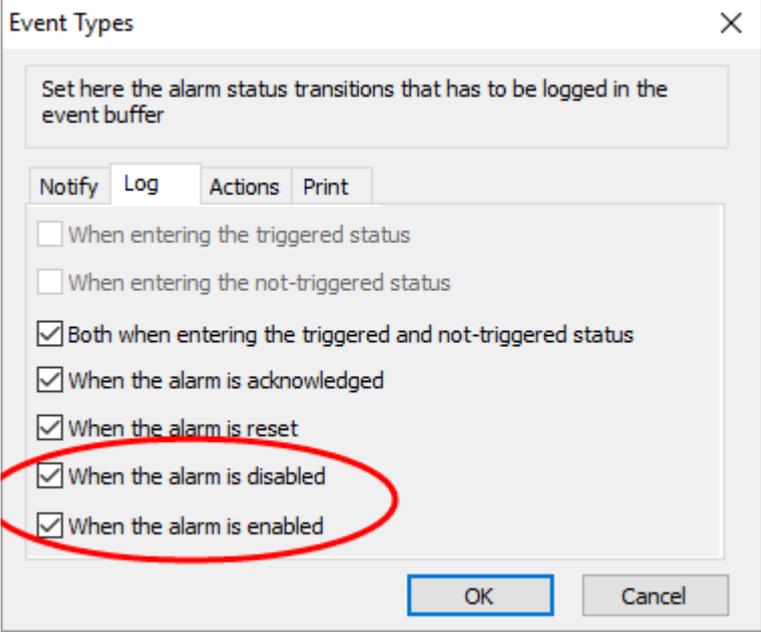
路径: 项目视图 > 配置 > 双击报警



添加一个报警

单击 **+** 添加报警。

参数	说明
名称	报警名称
组	与报警相关的群组。它们可用于控件显示筛选器。
启用	<p>启用/禁用报警。</p>  <p>也可在运行时启用或禁用报警(请参阅"运行时启用/停用报警"在本页 240 了解详细信息)。</p>
确认	启用/禁用报警确认, 如果选择, 在触发报警将其从 活动报警 控件中删除后, 操作者必须确认报警。
复位	与 确认 选项一起使用, 如果选择, 在操作者按下报警控件中的 复位 按钮之前, 已确认报警将留在报警列表中, 标记为 未触发已确认 。
缓存	报警历史保存在缓存文件中。
触发器	<p>触发条件取决于报警类型:</p> <ul style="list-style-type: none"> • limitAlarm: 在标签值超过了限制时触发的报警。如果该值达到限制, 报警不会启动。 • valueAlarm: 在标签值等于配置值时触发的报警

参数	说明
	<ul style="list-style-type: none"> • bitMaskAlarm: 位“与”运算符使用与报警相应的标签值比较位掩码的每一个位。如果两个位都打开了,报警设置为真。您可以指定标签中一个或多个位的位置(从 0 开始)。位的位置必须是十进位的格式;如果指定了更多的位,每个位置必须用一个 "," 隔开。 • deviationAlarm: 在标签值偏离设定点的比例超过了设定的偏离量时触发的报警。 $Value_{now} - SetPoint > \left(\frac{deviation}{100} \times SetPoint \right)$ • protAlarm:BACN: 报警从远程 BACnet 设备中触发。  仅当协议配置包含 BACnet 设备时,该触发模式才可用。请参阅“Communication Drivers”文件夹中的 BACnet 手册,了解 BACnet 报警事件的详细描述。
标签	<p>标签的值超过设置的限制时会启动报警。</p> <p>如果选择了 bitMaskAlarm 作为触发器,报警会引用此标签的值,或引用位的状态。</p>
远程启用	<p>PLC 的标签启用/停用报警。</p> <ul style="list-style-type: none"> • 如更改报警控件中的启用状态,则会更改标签值 • 无法读取标签(如通信错误)时,则停用报警 • 报警停用后,无法更新与报警相关的标签。 <p> 提示:在启用记录报警启用标签时很有用</p> 
远程确认	<p>由 PLC 的标签确认报警。标签的值从 0 变为非 0 视为确认请求。</p> <p>如果未要求远程确认,留为空白。</p> <p>请参阅“远程报警确认”在本页 231 了解详细信息。</p>

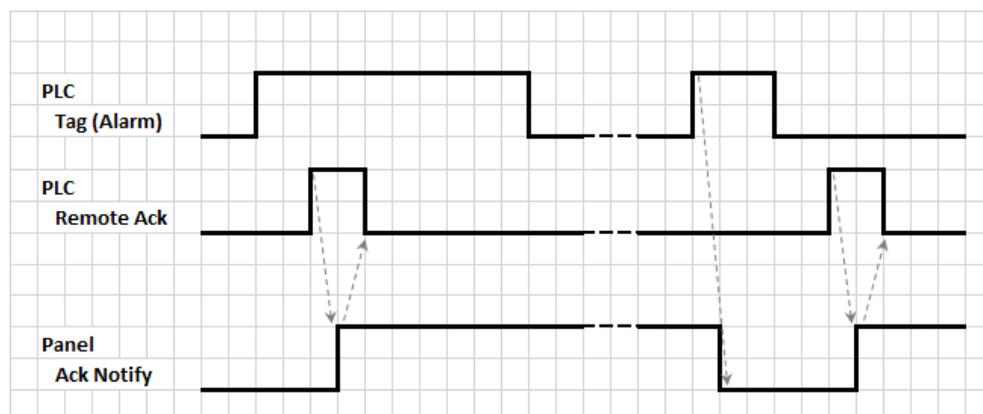
参数	说明
确认通知	当报警从设备或从 PLC 确认时,要进行通知的 HMI 设备所用的标签。 0=当报警启动时设定该值 1=当报警确认时设定该值。
动作	当报警启动时执行的动作。可以在 事件 列中指定附加条件。 请参阅 "设置 事件" 在本页 232 了解详细信息。
用户动作	用户按下激活报警控件中的动作按钮后,则执行动作。 请参阅 "激活报警控件" 在本页 234 了解详细信息。
说明	报警说明。该文本支持多语言特征,可以是静态和动态部分的结合,动态部分包括一个或多个标签值。 请参阅 "显示实时报警数据" 在本页 241 了解详细信息。
自定义字段 #	它是一个额外的报警描述,可用于显示报警控件中的额外信息。例如,可以是用于显示与报警相关的图片的索引。
颜色	以报警状态为基础的报警列前景色和背景色。
确认闪烁	触发的报警一直闪烁。如果选择,报警列一直闪烁,直到确认为止。仅在选择 确认 后才有效。
严重性	报警的严重性。如果同时触发了多个报警,动作会根据严重性设置执行。 0=不重要 1=低 2=标准以下 3=标准 4=标准以上 5=高 6=危险
事件	报警的通知、记录或打印情况。 请参阅 "设置 事件" 在本页 232 了解详细信息。

远程报警确认

设置**远程确认**参数时，可以从 PLC 设备确认报警，将标签值设为非零值。确认状态通过**确认通知**标签通知到 PLC 设备。

报警确认过程

远程确认标签由 PLC 设置/复位以请求确认，而**确认通知**由 HMI 设备设置/复位以通知执行确认操作。



1. 检测到报警情况时，HMI 设备将**确认通知**设为 0，系统将执行所有相关动作。
2. 报警(由 HMI 设备或以远程方式)确认后，**确认通知**设为 1。
3. 由控制器将**远程确认**设为 1 来确认报警，或者如果 HMI 设备发出报警已确认的通知，将其复位为 0(**确认通知** = 1)



警告:当报警被触发时，需要通过连接的设备更新/传递一些信号。我们假设确认是一个操作员推送的一个信号，不是控制器设备自动释放的。这里接受信号通信所需的时间。

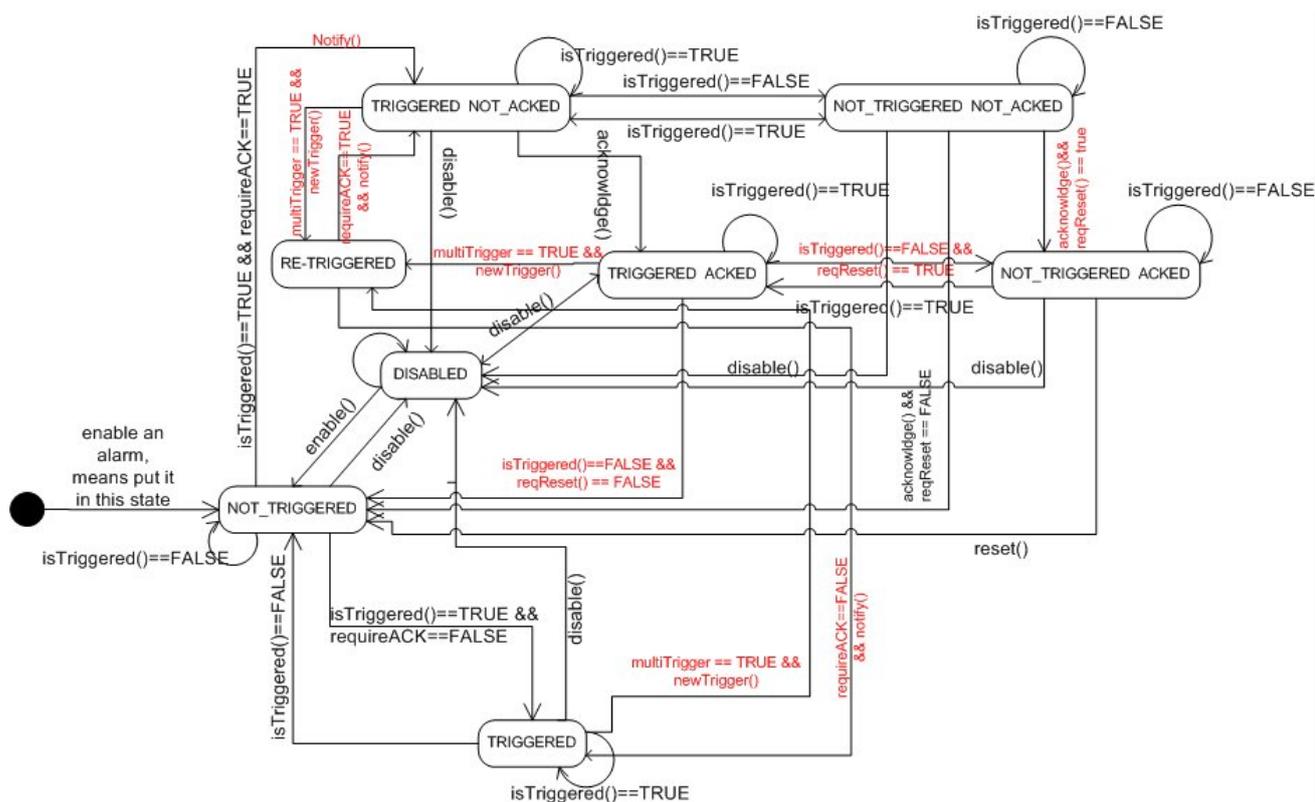


提示:对**远程确认**和**确认通知**使用相同的标签可以将更多设备连接到同一控制器并确认来自 HMI 设备的任何报警。

报警状态机

runtime 执行此图示中描述的报警状态机。

状态间的状态和过渡根据选择的选项和需要的行为描述。



设置事件

路径:项目视图 > 配置 > 报警 > 事件属性

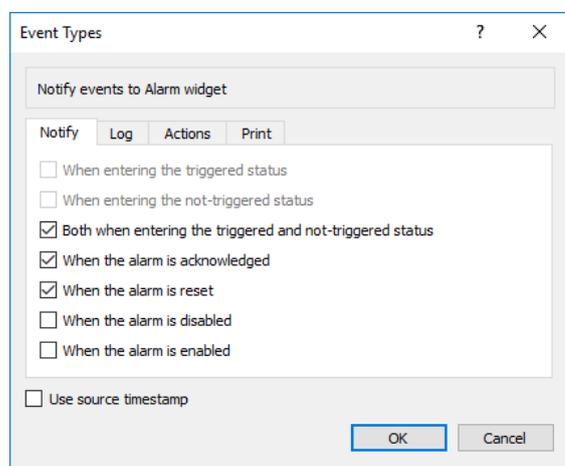
事件通过使用报警编辑器限定。

请参阅"报警编辑器"在本页228了解详细信息。

通知事件

路径:项目视图 > 配置 > 报警 > 事件属性 > 通知选项卡

设置报警在报警控件中被发出的条件。



您在此指定报警集中的可用默认报警行为，决定在什么情况下报警状态改变使控件更新。



小心：只按特定应用程序的要求进行调整，其他的设置保留为默认。

记录事件

路径：项目视图 > 配置 > 报警 > 事件属性 > 日志选项卡

设置在报警历史缓存中保存特定事件的条件。

Event Types

Log event to buffer

Notify Log Actions Print

When entering the triggered status

When entering the not-triggered status

Both when entering the triggered and not-triggered status

When the alarm is acknowledged

When the alarm is reset

When the alarm is disabled

When the alarm is enabled

Use source timestamp

OK Cancel

报警历史记录在事件缓存中。

执行动作

路径：项目视图 > 配置 > 报警 > 事件属性 > 动作选项卡

设置报警执行特定动作的条件。

Event Types

Trigger actions on event

Notify Log Actions Print

When entering the triggered status

When entering the not-triggered status

Both when entering the triggered and not-triggered status

When the alarm is acknowledged

When the alarm is reset

When the alarm is disabled

When the alarm is enabled

Use source timestamp

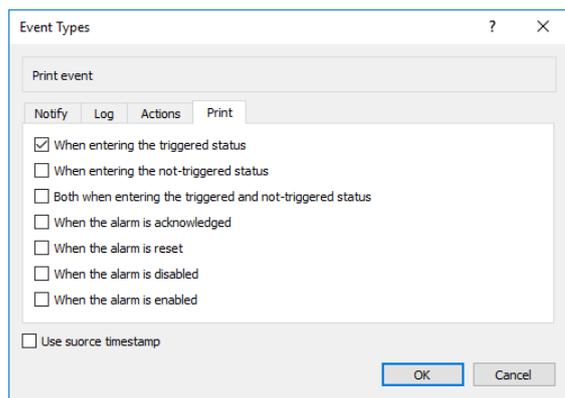
OK Cancel

默认条件下，只有在报警启动时才会执行动作；其他的报警状态也可以设置执行的动作。

打印事件

路径：项目视图 > 配置 > 报警 > 事件属性 > 打印选项卡

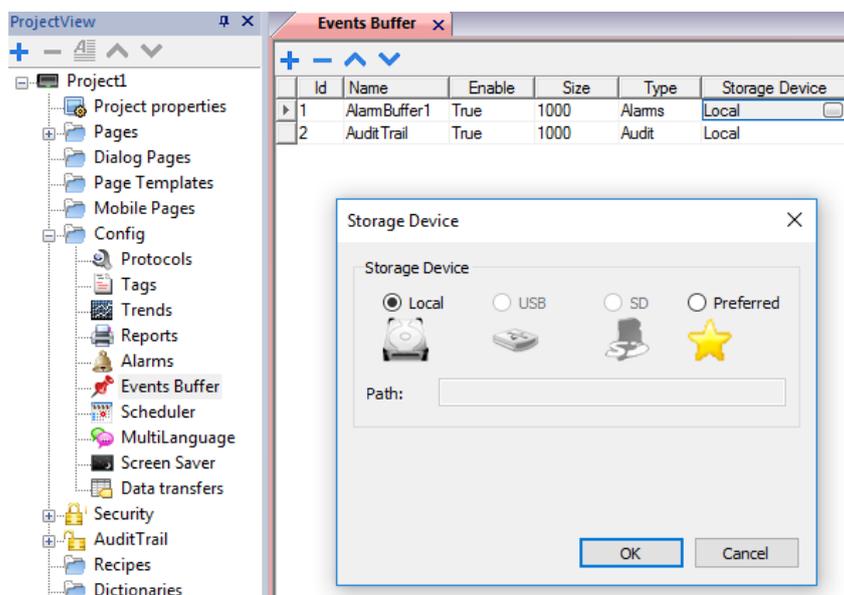
设置打印特定事件的条件



设置存储设备

路径: 项目视图 > 配置 > 事件缓存 > 存储设备选项卡

1. 打开**存储设备**对话框。
2. 选择事件数据存储的设备。



除实时保存的报警数据外，其他数据每五分钟自动保存一次。

使用源时间戳

当 HMI 设备检测到事件时，使用时间戳存储事件。选中“使用源时间戳”时，使用远程设备收到的时间戳存储事件。



仅可用于支持此功能 (OPC UA 客户端 和 BACnet) 的设备协议

激活报警控件

您可以在页面中插入**活动报警**控件以显示报警，并认可、复位或启用/禁用报警。

Active Alarms

Select	Name	State	Value	Time	Description	Severity	Enable

Filter :

报警筛选器

路径: **活动报警控件** > **属性窗格** > **筛选器**

定义筛选器只显示兼容的报警。筛选器基于报警栏，意思是您可以根据名称、严重性、描述等筛选报警。

筛选器 1 是默认筛选器。它由组合框 **筛选器 1** 管理，并且有两个选项：**显示所有报警** 和 **隐藏未启动报警**，选择后者时，将只显示活动报警。

默认情况下，筛选器 2 无配置，可以进行自定义。

筛选器的表达式使用 AWK 语言，这些表达式应用于“报警”控件所选 **筛选器** 列中包含的数据。

Alarms List	
Columns	
Sorting	false
Sort Column	Severity
Text	
Filter	
Filter Column	State
Filter 1	Hide Not Triggered
DataLink	itemData:Combo2
Filter Column	Select
Filter 2	

设置筛选器

路径: **活动报警控件** > **属性窗格** > **筛选器**

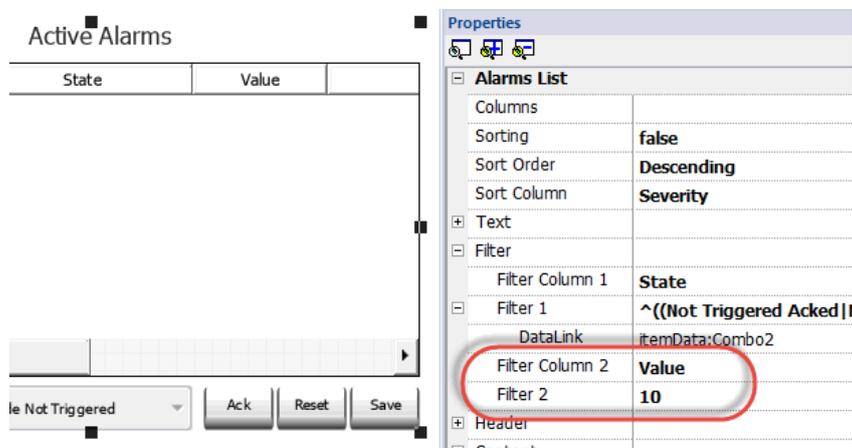
设置两个可用筛选器之一：

1. 选择 **筛选器列 1** 并选择筛选要使用的值(例如：“名称”、“状态”、“时间”、“组”)
2. 在 **数据链接** 上附上一个组合框控件。使用 **Shift+** 左击选择组合框。
3. 在 **属性窗格** 中，选择列表属性并打开对话框自定义组合框的值
4. 在“组合框配置”对话框中，指定 **字符串列表** 和正则表达式以筛选值。

有关正则表达式的其他详细信息，请参阅 https://en.wikipedia.org/wiki/Regular_expression。

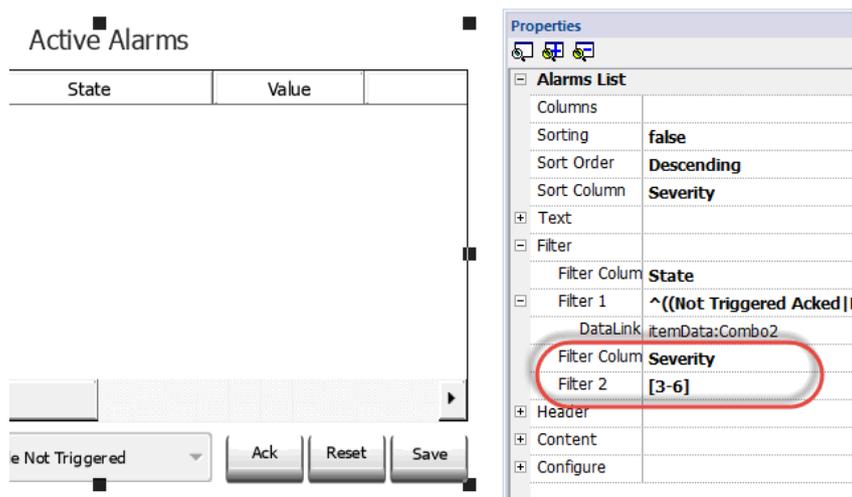
筛选器举例 1

您想显示所有值为 10 并与筛选器 1 匹配的报警。那么属性设置如下：**筛选器列 2 = 值**，**筛选器 2 = 10**



筛选器举例 2

您想显示所有匹配严重性值为 3 到 6(正常到危险)的报警。那么属性设置如下：**筛选器列 2 = 严重性**，**筛选器 2 = [3-6]**



筛选器举例 3

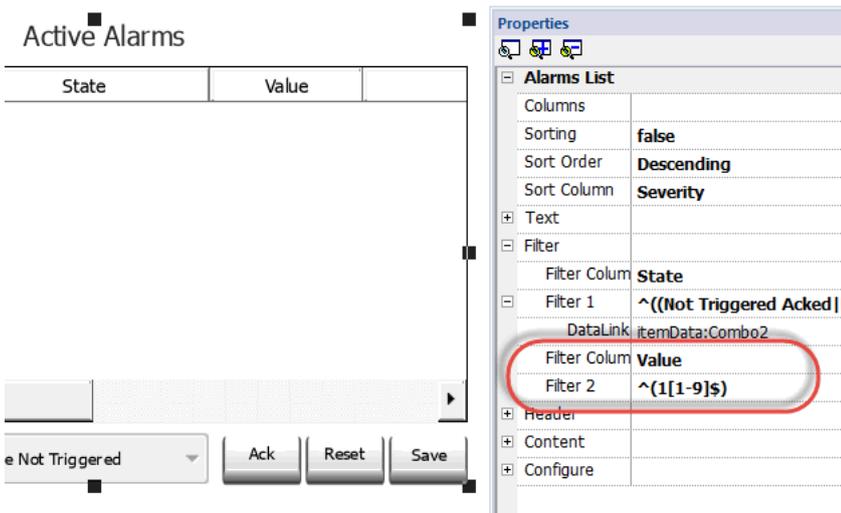
您想显示所有匹配值为 11 到 19 的报警。那么属性设置如下：**筛选器列 2 = 严重性**，**筛选器 2 = ^(1[1-9])\$**

意思：

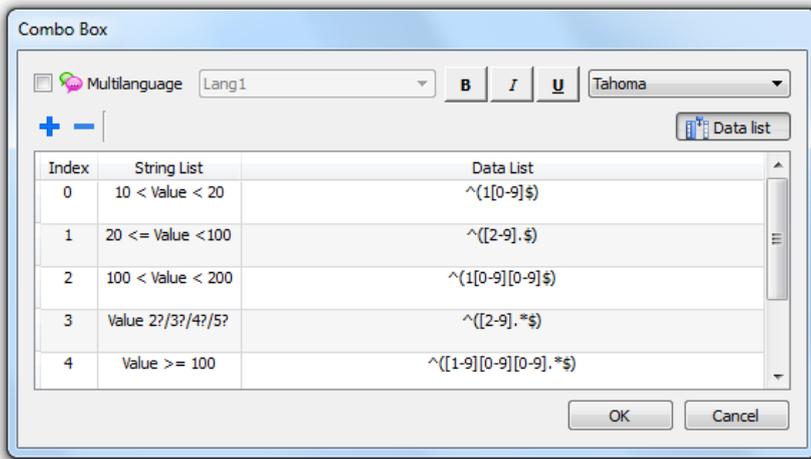
^ = 匹配必须从字符串的起点开始

1[1-9] = 第一个字符必须是 1，第二个字符必须在 1 和 9 之间。

\$ = 结束比较



筛选器表达示例



筛选方式	串表	数据列表
状态	隐藏未启动报警	^((未启动确认 未启动未确认 启动).*\$)
值	10 < 值 < 20	^(1[0-9]\$)
值	20 <= 值 < 100	^([2-9].\$)
值	100 < 值 < 200	^(1[0-9][0-9]\$)
值	值 2?/3?/4?/5?	^([2-9].*\$)
值	值 >= 100	^([1-9][0-9][0-9].*\$)
值	值 >= 20	^([2-9].*\$[1-9][0-9][0-9].*\$)

分类报警

路径: 活动报警控件 > 属性窗格 > 排序

分类功能使您在运行时在报警控件中通过点击栏目标题对报警分类。



注：在报警编辑器中设定了此处显示的严重性值。

动作

当与报警关联的“用户动作”(请参阅“报警编辑器”在本页 228 了解详细信息)包含有效动作时，“动作”图标将显示。按下该图标后，将执行配置动作。

Active Alarms

Action	Name	State	Time
	Alarm1	Not Triggered	03/08/2016 11:07:43 AM
	Alarm2	Triggered	03/08/2016 11:07:55 AM
	Alarm3	Not Triggered	03/08/2016 11:07:43 AM

Check/Uncheck All Filter : Show All Ack Reset Save



警告：如果您使用的是较旧版本的转换项目，则必须从控件库选择新版本替代旧的“活动报警”控件



注：可在“活动报警”控件的“列”属性中修改图片

Table Column Editor

Columns + - ^ v

- Action
- Select
- Enable
- Name
- Groups
- State
- Value
- Time
- Description
- Severity

Col 0 Info

Header	Action
Value	aUserAction
Width	100
Type	Image
Visible	true
Image path	images\action.png

OK Cancel

Alarms List : ActiveAlarms

Columns	
Sorting	false
Sort Order	Descending
Sort Column	Severity
Text	
Filter	
Header	
Content	
Configure	
General	
Position	

启用/禁用报警

在运行时，报警控件可用于启用或禁用报警。

保存在报警控件的启用列中所做的更改。这一动作与报警控件中的保存按钮一起使用。

Time	Enable
04-01-2012 12:05:00	<input checked="" type="checkbox"/>
04-01-2012 12:05:00	<input checked="" type="checkbox"/>
04-01-2012 12:05:00	<input checked="" type="checkbox"/>

Ack Reset Save

报警历史控件

如果设置了报警配置编辑器中的**缓存**属性，则记录并显示报警列表。

Alarms History

From : 09/24/13 - 16:04:49 Duration : 1 Min Refresh

To : 09/24/13 - 16:04:49

Name	State	Value	Time	Description	Event Type
					

Backward Forward

将控件附在缓存中

路径:报警历史控件 > 属性窗格 > 缓存 > 事件缓存

在属性窗格 > 事件中, 事件缓存可从中获取报警列表

其他报警控件

除了“活跃警报”和“警报历史”这两个主要的控件之外, 库里还包含其他一些警报控件, 其外观略有不同, 但基本与这两个主要控件相似。您可以自由选择和使用外观上能更好地满足您需求的控件。



请注意, 即使在打印报表库中, 某些控件也可用。

一些控件基于新的表格结构。对于上述控件，除了公开的属性之外，您还可以选择内部表，并使用表功能根据需求和喜好来修改控件(请参阅“表格控件”在本页418了解更多详情)。

打印历史警报列表

打印库包含基于表结构的历史警报控件，可用于生成警报报表。可以绘制表格并扩大至整个页面。如果要打印的行数大于一页，则将使用其他页面来打印警报表格。

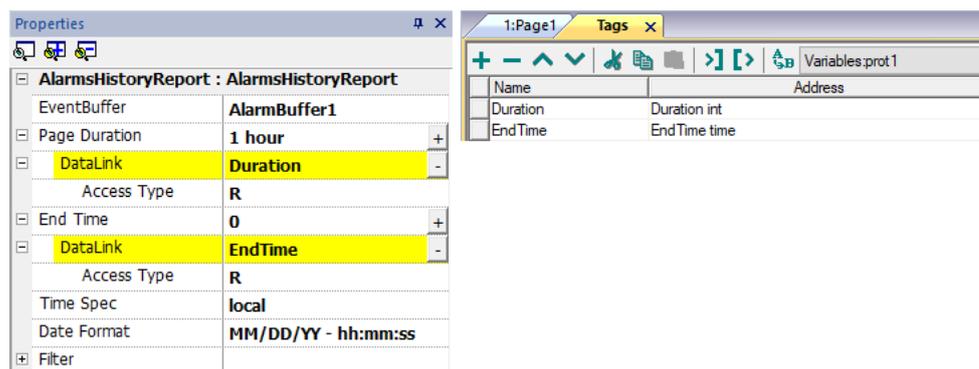
Timestamp	Name	State	Value	Description
Label	Label	Label	Label	Label



使用标签功能的添加附件，可在运行时使用标签定义要打印的历史警报列表的某些属性：

- 显示时长
- 结束时间

“页面持续时间”和“结束时间”定义要打印的警报缓冲区部分。



运行时管理报警

当报警启动，报警将会在活动报警控件中显示，您可以对其确认和复位。您可以使用一些过滤器过滤显示的报警，例如，您可以隐藏未启动的报警或显示所有报警。

请参阅“激活报警控件”在本页234了解详细信息。

 **重要提示：“活动报警”控件不会自动显示。您必须添加指定的动作，当报警启动时，这个动作将打开包含报警控件的页面。**

运行时启用/停用报警

您可以在运行时启用或停用报警。

要启用报警，请在报警控件中选中**启用**选项。

停用报警未启动，因此在运行时不会显示。

Select	Id	Source Value	State	Date	Time	Enable
<input type="checkbox"/>	Alarm1	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm2	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm3	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm4	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm5	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm6	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm7	23	Not Triggered Not Acked	25-01-2011	16:59:32	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm8	23	Not Triggered Not Acked	25-01-2011	16:59:32	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm9	23	Not Triggered Not Acked	25-01-2011	16:59:32	<input checked="" type="checkbox"/>

Check/Uncheck All Filter: Show All Ack Reset Save



注:报警甚至可通过 PLC 配置为启用/禁用。
请参阅[报警配置编辑器](#)了解详细信息。

显示实时报警数据

路径:项目视图 > 配置 > 双击报警

在“活跃报警”控件和“报警历史记录”控件中,都可以设置报警说明或自定义字段,以显示在线标签的数据。

Id	Name	Enable	Ack	Reset	Tag	Buffer	Trigger	Action	Description
1	Alarm 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tag1	AlarmBuffer1	bitMaskAlarm:	ShowDialog	Alarm 1 Tag Value is [Tag1]
2	Alarm2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 2 Tag Value is [Tag2]
3	Alarm3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 3 Tag Value is [Tag3]

要显示标签值,请在说明中设置一个占位符,在方括号中输入标签名称,例如“[Tag1]”。运行时,在“活跃报警”控件的说明列中,标签的当前值将显示。

在线标签占位符

Tags

- [TagName]
The tag value is read and continuously updated



Use '\' before '[' if you want to show the '[' in the description string, for example: `\[Tag\1\]` will display the string “[Tag1]”.

Use '\', even when the tag label contains square brackets. For example, to display the live tag value of tag “TAG]3” or “TAG[3]” use:

- TAG\]3 = [TAG]3
- TAG\[3] = [TAG[3]]

Array Tags

To reference the entire array (all elements will be shown):

- [TagName]
All array elements will be displayed using a comma separate list.
- [TagName[-1]]
All array elements will be displayed using a comma separate list.

To reference an element of the array:

- [TagName.Index]
Example: [MyARRAY.5] will display the sixth element of the MyARRAY
- [TagName[TagIndex]]
Example: [TagIndex] will display the sixth element of the MyARRAY when TagIndex is 5

Data Formats

Placeholder characters can be used to control how to display the tag value (see "自定义格式" 在本页 28)

- [TagName|format("####")]

Example:

Live: [fCounter|format("#.00")] - Triggered: [!fCounter|format("#.00")]



Note that by default, all tags are displayed as an integer. If you want to display a float number, you have to specify how to show the number adding the decimal digits.

冻结在线标签值

在线标签将被读取并不断更新。如果要在触发报警的那一刻冻结标签值，请使用感叹号作为标签名称前缀：

- [TagName]
当报警启动时，标签值会被读取并连续更新
- [!TagName]
当报警启动时，标签值会被读取并冻结

报警控件的例子

Select	Name	State	Value	Time	
<input type="checkbox"/>	Alarm1	Triggered Not Acked	1	30/09/2019 12:56:19	Live Counter: 44
<input type="checkbox"/>	Alarm2	Triggered Not Acked	1	30/09/2019 12:56:21	Triggered Counter: 11
<input type="checkbox"/>	Alarm3	Triggered Not Acked	1	30/09/2019 12:56:24	Live: 44 - Triggered: 14
<input type="checkbox"/>	Alarm4	Triggered	1	30/09/2019 12:56:35	Live: 0 - Triggered: 0
<input type="checkbox"/>	Alarm5	Triggered	1	30/09/2019 12:56:17	Live: 0.44 - Triggered: 0.07

Filter :



在历史报警控件中或者在 .csv 文件中，实时标签值为报告状态更改时采用的值(针对两种类型的占位符)

说明字段的长度限制

根据使用的标签大小，可用于每次报警说明中的实时标签数。使用太多标签时，LRH SW 将检查并显示警告消息。



使用底层算法计算的字节总数必须小于或等于 50。

4 个字节 + (标签大小 + 2) + (标签大小 + 2) + (标签大小 + 2) + ... ≤ 50

示例：

Alarm Description:

Tag1=[TagInt], Tag2=[TagBool], Tag3=[TagStr8]

Fixed	4	
Tag1	6	4 (sizeof-INT) +2
Tag2	3	1 (sizeof-BOOL) +2
Tag3	10	8 (sizeof-STR8) +2
Total:	23	

当使用数组时，例如 Tag1 作为 8 个整数的数组：

- **[Tag1]** 或 **[Tag1[-1]]**
显示整个数组，所需的字节数计算为 $4(\text{size-INT}) \times 8(\text{数组元素}) + 2 = 34$ 字节
- **[Tag1[Index]]**
显示数组的一个元素，所需的字节数计算为 $4(\text{size-INT}) \times 1(\text{数组元素}) + 2 = 6$ 字节。在这种情况下，如果在运行时索引假定值为 -1，则某些值可能会丢失

输出报警缓存到 .csv 文件

要导出包含历史报警列表的事件缓存，请使用 **保存事件存档** 动作。

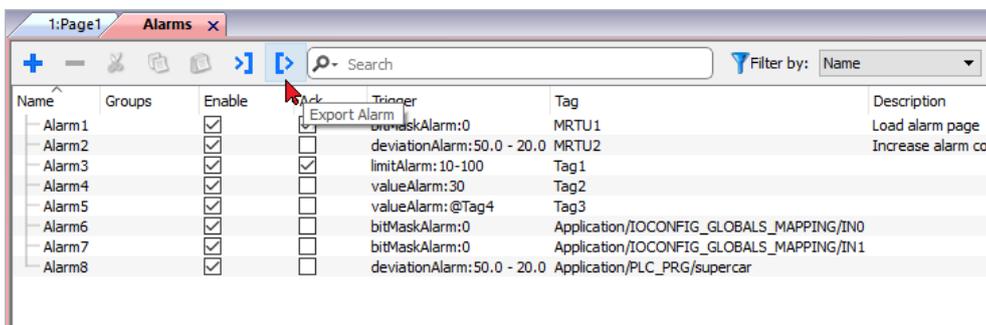
请参阅“**系统动作**”在本页 190 了解详细信息。



注：报警描述中显示的标签值也将包含在缓存中。当报警启动且值被记录和包括在描述中时，标签被取样。

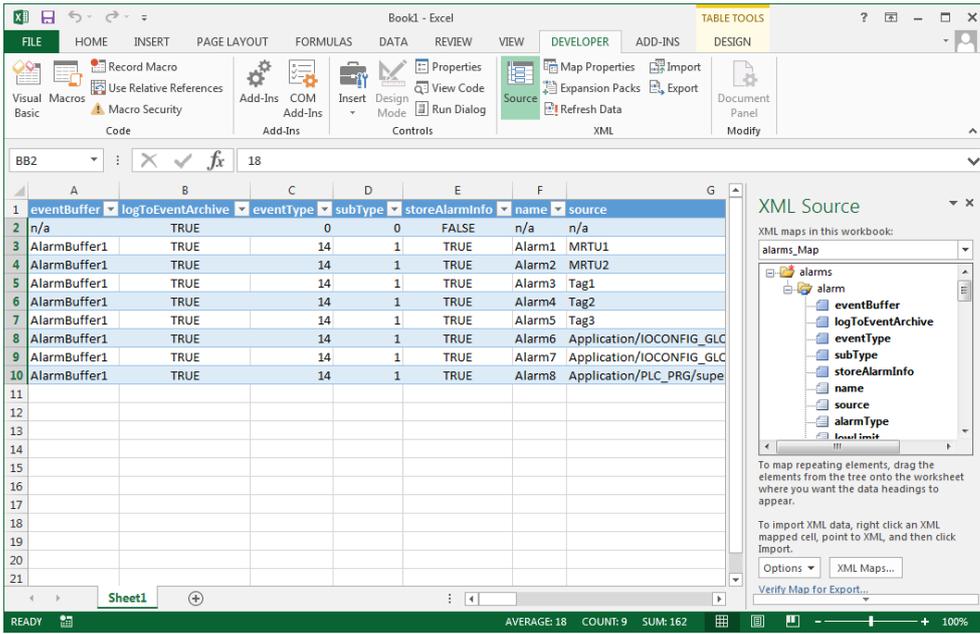
导出报警配置

路径：**项目视图 > 配置 > 双击报警**



单击 **导出报警** 按钮：报警配置表格导出至 .xml 文件中。

您可以使用第三方工具编辑最后的 .xml 文件(例如，Microsoft Excel)。

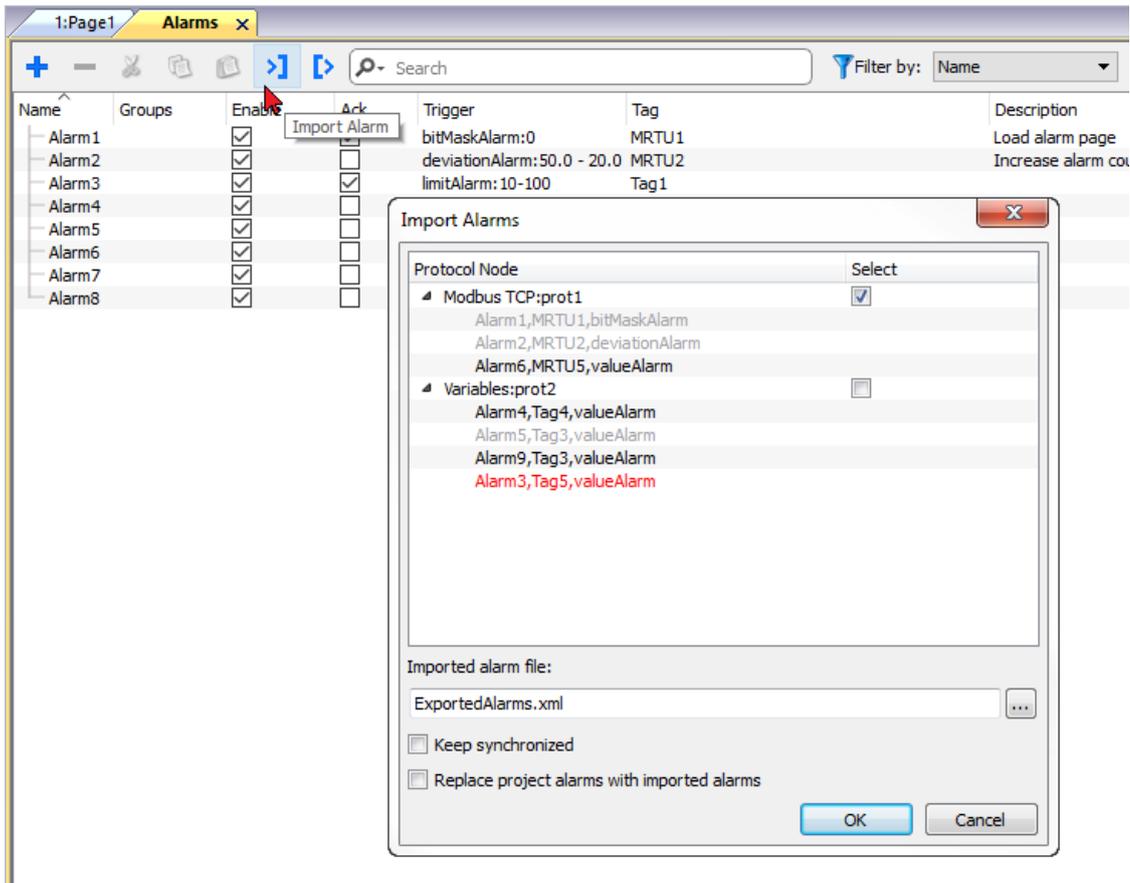


警告: bitMask 值以十六进制格式报告为 $2^{\text{BitPosition}}$ 。

alarmType	lowLimit	highLimit	value	bitMask	deviation	setPoint	enableTag
n/a	0	1000	0	1	50	20	n/a
bitMaskAlarm				1			
bitMaskAlarm				2			
bitMaskAlarm				4			
bitMaskAlarm				8			
bitMaskAlarm				10			
bitMaskAlarm				20			
bitMaskAlarm				40			
bitMaskAlarm				80			
bitMaskAlarm				100			
bitMaskAlarm				200			
bitMaskAlarm				400			
bitMaskAlarm				800			
bitMaskAlarm				1000			
bitMaskAlarm				2000			

导入报警配置

路径: 项目视图 > 配置 > 双击报警



1. 单击**导入报警**按钮并选择可从中导入报警配置的 .xml 文件：**导入报警**对话框显示。
2. 选择要导入的报警组并单击**确定**确认。

在**导入报警**对话框中用不同的颜色来突出显示不同的状态

颜色	说明
黑	这是一个新的报警，将被导入
红色	找不到此报警，将移除(仅当选中的“将项目报警替换为导入报警”复选框时)
蓝色	该报警已被修改，将被更新。
灰	该报警已经是项目的一部分，将被跳过。

自动同步

在**导入报警**对话框中选中**保持同步**选项，启用报警配置文件自动同步。

无论什么时候报警配置中发生更改，该文件将以静态模式被自动更新。



提示：当报警文件由不同的工具(例如，PLC 编程软件)以及 LRH SW 管理时，启用此功能。

19 配方

配方是以组的形式满足特定应用要求的标签值集合。

例如，如果要控制早上、下午和晚上的房间变量(温度和湿度)。您将创建三个组(早上、下午和晚上)，在其中设置相应标签值。

配方的每一个元素与一个标签相关联，可以集中索引以备更有效的使用。此功能允许您扩展具有有限内存的控制器的功能。

您可以使用配方控件将控制器数据添加至页面。配方数据包含所有控制器数据项；但数据不再从控制器直接读取，而是从 HMI 设备上的相关配方元素读取。

配方数据在 LRH SW 工作空间中配置；用户可以为数据记录的每个元素指定默认值。在 LRH SW HMI Runtime 中，数据可以编辑并保存到新的数据文件中，因此对配方数据的任何更改均可存储到磁盘中。使用单独的数据文件时，LRH SW HMI Runtime 可确保修改的配方值在更新不同项目的过程中被保留下来。换句话说，后续项目更新不会影响用户在 LRH SW HMI Runtime 中修改的配方数据。

请参阅“配方动作”在本页 185，了解如何复位配方数据的详细信息。



注：配方数据可以存储在闪存、USB 驱动器或 SD 卡上。

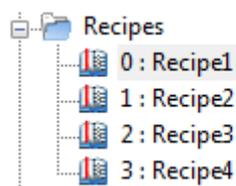
管理配方	247
配置配方控件	250
配方状态	251
上传 / 下载一个配方	252
备份和恢复配方数据	253

管理配方

创建配方

为您的项目创建配方：

1. 在**项目视图**中，右键单击**配方**然后选择**插入配方**：系统将添加一个空配方。使用配方编辑器创建并配置配方。



配方编辑器

路径：**项目视图 > 配方 > 双击配方名称**

index	Element Name	Tag	Fill Tank 1	Fill Tank 3	Fill Tank 5	Fill Tank 7	Fill Tank 1	Empty Tank	Empty Tank	Empty Tank 75_	Em
0	Home Valve	Recipe_HomeV:1	1	1	1	1	0	0	0	0	0
1	Truck Valve	Recipe_TruckV:0	0	0	0	0	1	1	1	1	1
2	Fill Flow Meter	Recipe_FillFlow:15	35	50	75	100	75	50	25		15
3	Empty Flow Meter	Recipe_EmptyFl:0	0	0	0	0	25	50	75		85
4	Chemical1	Recipe_Chemic:0	0	0	0	0	0	0	0		0
5	Chemical2	Recipe_Chemic:0	0	0	0	0	0	0	0		0

配置配方属性

在各个配方的属性窗格中，设置以下参数：

参数	说明
配方名称	配方名称
组数量	每个配方的数量组。每个组有不同的可配置名称。

Properties

Recipe : _RecipeMgr

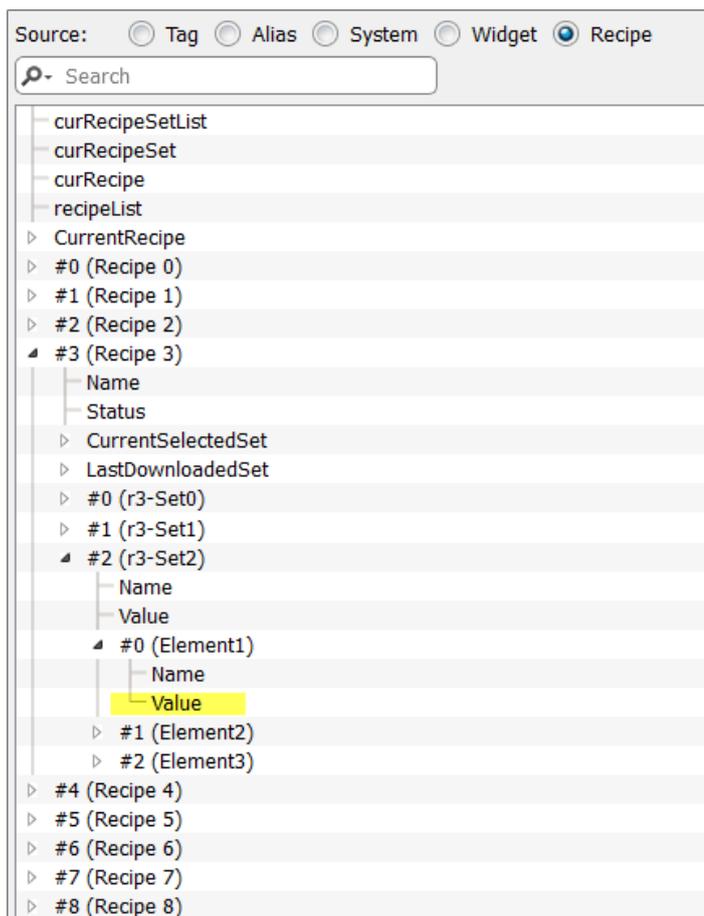
Recipe Name	Recipe1
Number of sets	10
Set 0	Fill Tank 15_
Set 1	Fill Tank 35_
Set 2	Fill Tank 50_
Set 3	Fill Tank 75_
Set 4	Fill Tank 100_
Set 5	Empty Tank 25_
Set 6	Empty Tank 50_
Set 7	Empty Tank 75_
Set 8	Empty Tank 90_
Set 9	Empty Tank 100_

添加一个配方

1. 单击 + 添加该配方的元素。
2. 将标签链接到每个配方元素。

定义配方字段

使用一个数字控件在页面创建一个配方域，并选择配方的来源，将其附在配方项上。



在**附加到**对话框中，您可以选择所有不同的配方变量，例如：

- 当前配方 > 当前所选配方组 > 元素 > 值
- 所选配方 > 所选集 0 > 元素 > 值
- 配方列表

当数字控件按照读/写定义，默认的配方数据可以在运行时被编辑。这些新的值当作修改的配方数据保存在单独的文件中。



注：因使用 JavaScript API 函数，所以可通过名称或位置引用配方元素和配方集。为避免名称和索引之间出现模糊，配方元素和数据集的名称必须包括至少一个字母数字字符。

存储配方数据

在配方编辑器中，单击存储类型图标  选择配方数据的存储位置：**存储设备**对话框显示。



对于 USB 驱动和 SD 卡存储，您可以提供文件夹位置。

 **警告:**保存项目且将其存储在项目的数据子文件夹中时，系统自动创建配方配置文件。要使用外部存储设备，您需要将该文件夹复制到外部设备中。请注意，您有责任管理外部设备内的数据文件夹。更新项目时，即使使用“删除动态文件”选项，也无法删除动态文件。

 **重要:**您可以添加子文件夹，但不得重命名“**data**”子文件夹。

导入/导出配方

如要导入/导出您的项目的配方配置：

在项目视图中，右键单击**配方**然后选择**插入配方**或**导出配方**。

以下是导入支持的格式：

- 逗号分隔值 (.csv)
- Unicode 文本 (.txt)

 **注:**当您导入使用 Microsoft® Excel® 修改的文件时，请使用 Unicode 文本文件格式。

配置配方控件

您可以在**控件库**中选择两个可用配方控件中的一个：

- **配方集:**您可以选择要上传或下载的配方集。请参阅["上传 / 下载一个配方"](#)在本页 252
- **配方菜单:**为某个项目创建了多个配方时，使用该控件管理所有配方并选择各个所需的配方集。

Recipe Set

Recipe Set

Download

Upload

Recipe Menu

Recipe

Recipe Set

Download

Upload

配置配方集控件

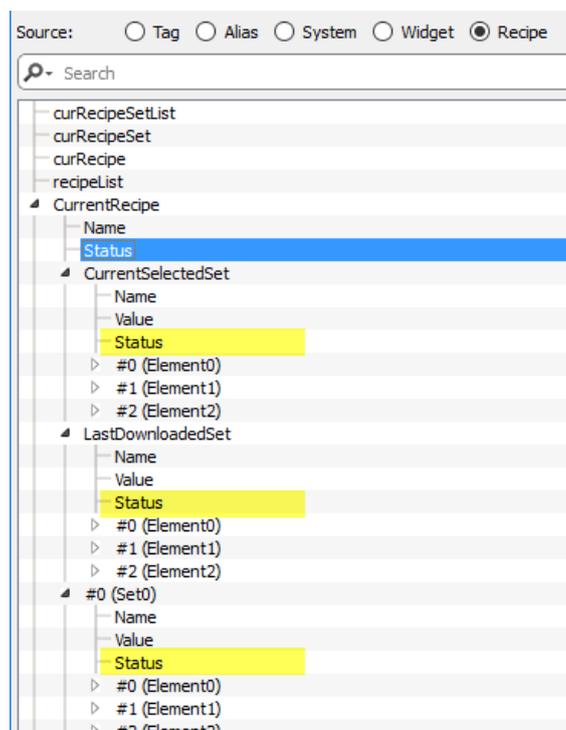
在各个配方集控件的属性窗格中，设置以下参数：

参数	说明
配方名称	配方名称

配方状态

每个配方包括两种状态参数

- 配方状态(下图中以蓝色显示)
提供有关上次下载或上传操作的信息
- 数据集状态(下图中以黄色显示)
提供修改的数据集信息



配方状态

每次上传或下载配方后或者修改配方集后，**配方状态**参数都包含一个含操作结果的值。

代码	功能	说明
0	设置修改	所选集被更改
1	下载被启动	下载请求被启动
2	下载完成	下载动作完成
3	下载错误	下载过程中的错误(例如, 位置集、位置配方、控制器未就绪、标签写入失败等)
4	上传被启动	上传请求被启动
5	上传完成	上传动作完成
6	上传错误	上传过程中出错-与下载相同
7	综合错误	综合错误(例如, 数据不可用)

数据集状态

每个数据集的状态表示它已更改。不要忘记下载配方以同步 PLC 时，此信息可能有用。两种下载或上传操作将使**数据集状态**重置为 0。

代码	功能	说明
0	已同步	用户已使用数据集值同步 PLC
1	已修改	用户已修改数据集的一些值



注:在设备启动或配方重置/恢复之后,所有状态值将重置为 0。

上传 / 下载一个配方

上传一个配方

使用配方控件和**上传配方**、上传当前配方动作通过以下方式之一将配方上传到 HMI 设备:

- 将该动作附加到按钮或开关事件上(请参阅"**“附属于”参数**"在本页 39了解详细信息)
- 在报警动作列表中配置该动作(请参阅"**报警动作**"在本页 168了解详细信息)
- 在计划任务动作列表中配置该动作(请参阅"**运行时计划任务事件**"在本页 304了解详细信息)

下载一个配方

使用配方控件和**下载配方**、**下载当前配方**动作下载 HMI 设备中的配方。请参阅"**配方动作**"在本页 185

备份和恢复配方数据

储存在HMI设备上的配方数据可以导出以进行备份和后期恢复。这可使用**转储配方数据**或**还原配方数据**动作来完成。

请参阅"**配方动作**"在本页**185**了解详细信息。

20 趋势

趋势使您可以根据特定取样条件对指定的标签进行取样和记录。趋势功能包括获取趋势和显示趋势。

取得趋势参数在趋势编辑器中设置，这样可以保存数据。保存的数据可使用趋势控件以图表格式显示出来。

数据记录	256
导出趋势缓存数据	260
实时趋势控件	262
历史趋势控件	263
散布图控件	265
趋势控件手势	266
表格趋势控件	272

数据记录

数据可以记录和存储至 HMI 内存。数据记录允许您同时向缓冲区存储一组标签的值。数据记录可以由定时器或指定标签发起。记录的数据可以导出到 .csv 文件或使用历史趋势控件显示。记录的数据可以本地保存在 USB 设备上或 SD 卡上，或任何可用自定义网络文件夹上。



警告:对包含大量文件的可移动存储设备(USB 闪存驱动器、SD 内存卡)执行操作可能导致系统性能下降。



警告:SD 内存卡内的最大文件数量取决于格式化类型(例如 FAT32 最多支持 65536 个文件;FAT 最多支持 513 个文件)。



警告:闪存卡支持有限数量的写入操作。我们建议仅使用质量好的存储卡;如果您的应用频繁使用存储卡,请考虑定期替换存储卡。



警告:如果回移数据/时间,系统将从趋势缓存中删除具有无效日期/时间的样本。当系统检测到的无效的数据/时间时(例如电池电量低),会弹出信息显示最后采用的用户和日期/时间,以避免丢失数据。

存储基于趋势缓冲区。趋势缓冲区为一个 FIFO 队列:缓冲区满时,放弃最旧的值,除非您配置趋势创建缓冲区备份副本。

添加趋势缓存

路径:项目视图 > 配置 > 双击趋势

1. 单击**添加**来添加新的缓存。
2. 单击每个趋势缓存旁边的 **+** 显示所有配置参数。

Name	Title	Tag	Format	Comment
1 Name1	Temperature ([UnitaDIMisura])	Tag1	Custom	
2 Name2	PLC Value	Tag2	Numeric	
3 Name3	String	Tag3		

工具条元素	说明
添加趋势	添加将从 HMI 设备采样的趋势。
添加 PLC 趋势	添加将从外部设备，而不是从 HMI 设备进行管理和采样的趋势。您需要支持此功能的设备才能进行使用 (例如 BACnet 设备)。参数取决于使用的设备，请参阅所选设备的手册。
删除趋势	删除选择的趋势。
设置	<p>提供自定义趋势转储标题和趋势表窗口小部件中显示的标签的功能。</p> <ul style="list-style-type: none"> • 时间戳 • 日期 • 时间 • Quality <p>可以使用以下占位符：</p> <ul style="list-style-type: none"> • " \n " (空格+ \n +空格) 可用于将标签分成两行或更多行 • [TagName] (括在方括号中的标签名称) 可用于显示标签值
总内存空间	定义的趋势缓冲区使用的内存。

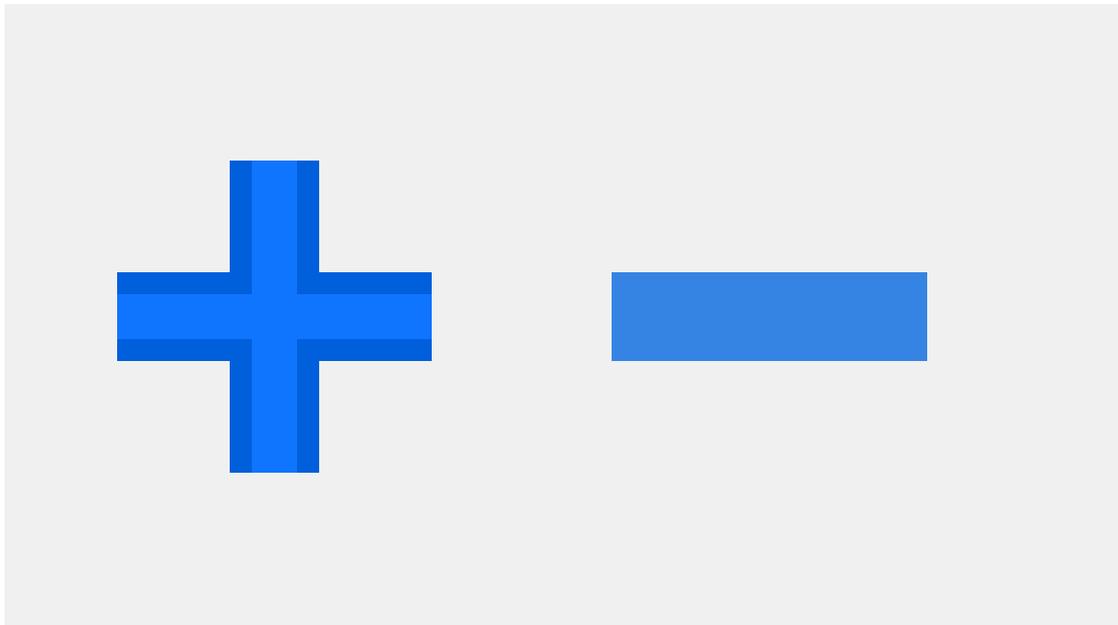
趋势标题	说明
趋势名称	趋势集合的名称(同时采样的一组标记)
激活	<p>启用时，趋势在系统启动时以默认方式运行。</p> <p> 注：不能在运行时激活趋势。</p>
源	趋势采样的标签列表。

趋势元素	说明
样本数量	趋势缓冲区大小(请参阅"样本数量"在本页 260 了解更多详情)
采样时间	<p>采样间隔。</p> <p> 注意，除了常量以外，还可以使用标签来定义、更改运行时的采样时间范围。当采样时间为 0 或为负时，采样将暂停。</p>
时间	<p>采样时间的时间单位。可能是 1 秒(默认)或 1/10 秒</p> <p> 请注意，提高采样率可能会影响 HMI 设备的整体性能。</p>
时间戳	<p>选中后，将使用远程设备提供的时间戳存储样本。</p> <p>仅可用：</p>

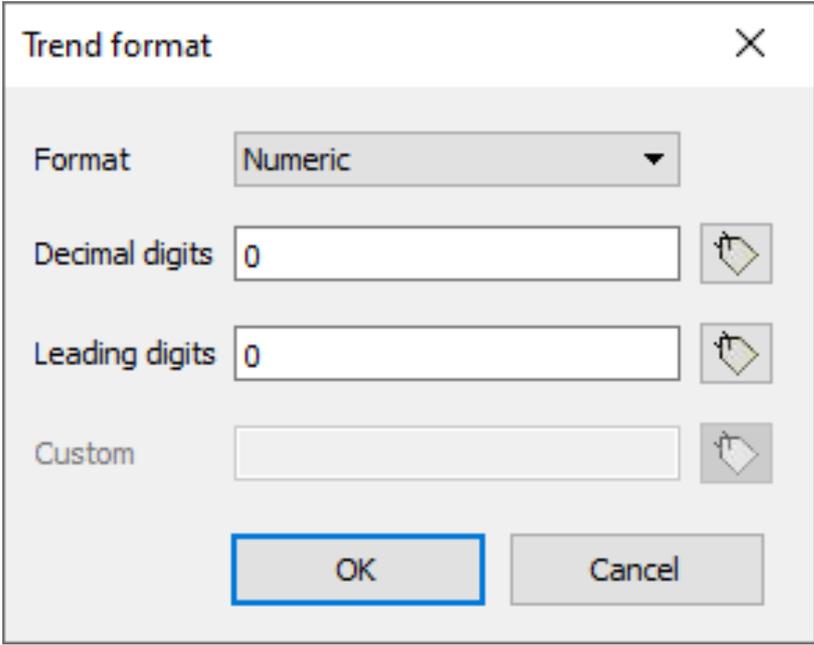
趋势元素	说明												
	<ul style="list-style-type: none"> 于支持此功能(OPC UA 客户端 和 BACnet)的设备协议 当趋势缓冲区配置有单个标签时 												
触发器	<p>触发标签。</p> <p>如果使用触发器，当标签值更改时，将收集样本。</p> <p> 注:触发器和源可以指向同一个标签。</p>												
存储设备	趋势缓存数据将保存的地方。												
备份存档	<p>启用存满后保存副本选项后，将先创建缓存数据备份副本，然后用更新的数据覆盖。</p> <table border="1"> <tr> <td>.csv</td> <td>备份使用文本 CSV 格式的数据。</td> </tr> <tr> <td>路径</td> <td> <p>趋势缓存数据将复制的地方。</p> <p>支持以下通配符</p> <ul style="list-style-type: none"> • %n = 趋势名称 • %y = 年 • %M = 月 • %d = 日 • %h = 小时 • %m = 分钟 • %s = 秒 </td> </tr> <tr> <td>选择字段</td> <td> <p>转储文件中的字段</p> <p> 请注意，您可以使用字符串标签来定义要在运行时转储的字段。</p> </td> </tr> <tr> <td>选择曲线</td> <td> <p>转储文件中的曲线</p> <p> 请注意，您可以使用字符串标签来定义要在运行时转储的字段。</p> </td> </tr> <tr> <td>时间规格</td> <td> <p>样本时间戳</p> <ul style="list-style-type: none"> • 本地 项目运行时，使用 HMI 设备的时间 • Global 采用全球时间 (GMT) </td> </tr> <tr> <td>数据格</td> <td>时间和日期格式。可使用占位符(请参阅 "时间和日期占位符" 在本</td> </tr> </table>	.csv	备份使用文本 CSV 格式的数据。	路径	<p>趋势缓存数据将复制的地方。</p> <p>支持以下通配符</p> <ul style="list-style-type: none"> • %n = 趋势名称 • %y = 年 • %M = 月 • %d = 日 • %h = 小时 • %m = 分钟 • %s = 秒 	选择字段	<p>转储文件中的字段</p> <p> 请注意，您可以使用字符串标签来定义要在运行时转储的字段。</p>	选择曲线	<p>转储文件中的曲线</p> <p> 请注意，您可以使用字符串标签来定义要在运行时转储的字段。</p>	时间规格	<p>样本时间戳</p> <ul style="list-style-type: none"> • 本地 项目运行时，使用 HMI 设备的时间 • Global 采用全球时间 (GMT) 	数据格	时间和日期格式。可使用占位符(请参阅 "时间和日期占位符" 在本
.csv	备份使用文本 CSV 格式的数据。												
路径	<p>趋势缓存数据将复制的地方。</p> <p>支持以下通配符</p> <ul style="list-style-type: none"> • %n = 趋势名称 • %y = 年 • %M = 月 • %d = 日 • %h = 小时 • %m = 分钟 • %s = 秒 												
选择字段	<p>转储文件中的字段</p> <p> 请注意，您可以使用字符串标签来定义要在运行时转储的字段。</p>												
选择曲线	<p>转储文件中的曲线</p> <p> 请注意，您可以使用字符串标签来定义要在运行时转储的字段。</p>												
时间规格	<p>样本时间戳</p> <ul style="list-style-type: none"> • 本地 项目运行时，使用 HMI 设备的时间 • Global 采用全球时间 (GMT) 												
数据格	时间和日期格式。可使用占位符(请参阅 "时间和日期占位符" 在本												

趋势元素	说明	
	式	页 402)
	语言	要使用的语言
采样滤波器/触发滤波器	<p>如有必要，此参数允许指定动态筛选器。</p> <p>基于时间进行采样时，将偏移应用于采样标签值。如果新值超过指定限制，则新值视为有效并可存储，否则新记录将保留之前保存的值。</p> <p>进行采样时，将偏移应用于触发器标签值。如果触发器标签值更改超过指定的限制，则提取并存储新样本，否则无法进行采样。</p>	

使用添加/删除按钮，添加标签以进行采样



样本	说明
名称	趋势名称
标题	<p>必须出现在趋势表或趋势转储中的标题。</p> <ul style="list-style-type: none"> 占位符 "<code>\n</code>" (空格+<code>\n</code>+空格) 可用于将标签分成两行或更多行 占位符 <code>[TagName]</code> (括在方括号中的标签名称) 可用于显示标签值 <p>示例：</p> <p>当标签 <code>UnitaDiMisura = "°C"</code> 时，“温度 (<code>[UnitaDiMisura]</code>)”将显示为“温度 (°C)”。</p>
标签	可以采样的标签
格式化	显示要使用的格式。请注意，甚至可以使用自定义格式(请参阅 "自定义格式" 在本页 28)。

样本	说明
	
注释	您可以在此写入任何注释

样本数量

您可以拥有的样本数量取决于为趋势缓冲区保留的内存大小以及每个样本的大小。

$$\text{可用样本数} = \text{可用内存} / \text{样本大小}$$

每个样本的大小取决于使用多少标签，并且可以使用以下公式计算：

$$\text{样本数量} = \text{TAGS} * 9 + 11$$

您可以自由地将整个可用内存用于唯一的趋势缓冲区，也可以将可用内存划分为多个趋势。

另请参阅：

- "功能限制表" 在本页 524 的趋势限制
- "HMI 设备地址" 在本页 525 上的趋势缓冲区的保留内存

导出趋势缓存数据

使用 **保存趋势** 动作将趋势缓存数据导出到一个 .csv 文件中。请参阅 "保存趋势" 在本页 190 了解宏参数详细信息。

导出的 .csv 文件可能具有 "保存趋势" 宏参数中定义的不同格式。不同的格式维护主要出于兼容性原因。

FileFormat: 兼容 **CSV**

	A	B	C	D	E	F	G	H	I	J	K
1	Type	Value	Time Stamp	Refresh Time	Quality	Type	Value	Quality	Type	Value	Quality
2	4	0	2015-09-18T14:42:22.000Z	1000	192	8	0.00E+00	192	3	0	192
3	4	0	2015-09-18T14:42:23.000Z	1000	192	8	0.00E+00	192	3	0	192
4	4	0	2015-09-18T14:42:24.000Z	1000	192	8	0.00E+00	192	3	0	192
5	4	40	2015-09-18T14:42:25.000Z	1000	192	8	0.00E+00	192	3	0	192
6	4	40	2015-09-18T14:42:26.000Z	1000	192	8	0.00E+00	192	3	0	192
7	4	40	2015-09-18T14:42:27.000Z	1000	192	8	0.00E+00	192	3	0	192
8	4	40	2015-09-18T14:42:28.000Z	1000	192	8	5.00E+01	192	3	0	192
9	4	40	2015-09-18T14:42:29.000Z	1000	192	8	5.00E+01	192	3	0	192
10	4	40	2015-09-18T14:42:30.000Z	1000	192	8	5.00E+01	192	3	0	192

FileFormat: 紧凑 CSV

	A	B	C	D	E	F	G
1	Timestamp	Tag1	4 Tag2	8 Tag3	3		
2		Value	Quality	Value	Quality	Value	Quality
3	2015-09-18T14:42:22.000Z	0	192	0.00E+00	192	0	192
4	2015-09-18T14:42:23.000Z	0	192	0.00E+00	192	0	192
5	2015-09-18T14:42:24.000Z	0	192	0.00E+00	192	0	192
6	2015-09-18T14:42:25.000Z	40	192	0.00E+00	192	0	192
7	2015-09-18T14:42:26.000Z	40	192	0.00E+00	192	0	192
8	2015-09-18T14:42:27.000Z	40	192	0.00E+00	192	0	192
9	2015-09-18T14:42:28.000Z	40	192	5.00E+01	192	0	192
10	2015-09-18T14:42:29.000Z	40	192	5.00E+01	192	0	192



注: 标题的第一行包含标签名称和标签数据类型

FileFormat: 具有列选择的紧凑 CSV

	A	B	C	D	E	F	G	H
1	Date	Time	Name1(int)	Quality	Name2(int)	Quality	Name3(boolean)	Quality
2	17/04/2018	07:24:29	0	192	10	192	0	192
3	17/04/2018	07:24:30	1	192	11	192	1	192
4	17/04/2018	07:24:31	2	192	12	192	0	192
5	17/04/2018	07:24:32	3	192	13	192	1	192
6	17/04/2018	07:24:33	4	192	14	192	0	192
7	17/04/2018	07:24:34	5	192	15	192	1	192
8	17/04/2018	07:24:35	6	192	16	192	0	192
9	17/04/2018	07:24:36	7	192	17	192	1	192
10	17/04/2018	07:24:37	8	192	18	192	0	192



保存趋势缓冲区所需的时间取决于缓冲区里的显示的样本数量、记忆类型以及 HMI 设备类型。

最坏情况下的示例

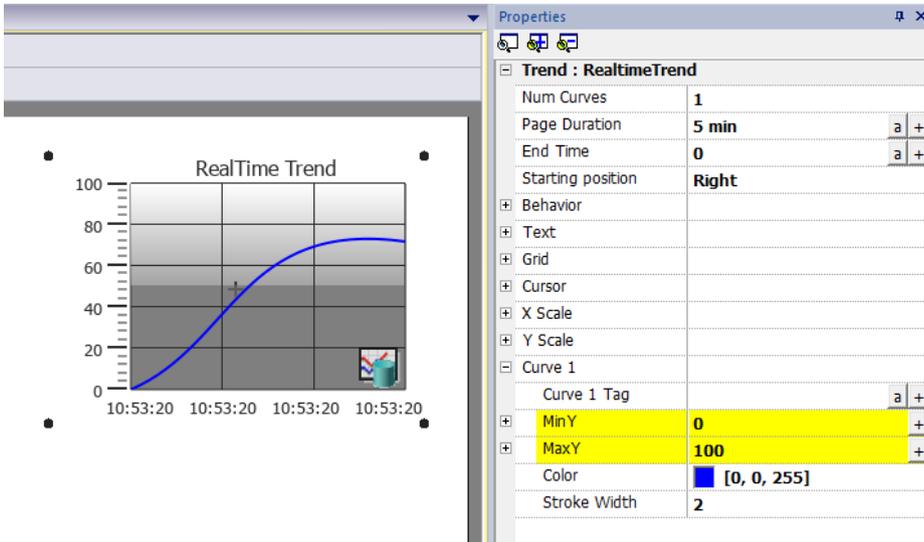
HMI 类型	缓冲区大小	样本	时间
Linux	50 Mb	1.807.800 个样本 (2 个标签)	4 分钟

实时趋势控件

实时趋势控件可以用来显示标签值的更改。数据不保存在趋势缓存中，不能重新获取以便后期分析。

显示实时趋势：

1. 将**实时趋势**控件从控件库拖放到页面中。



2. 将您要取样的标签附加到**曲线 n 值**。数据始终按照时间绘制。

实时趋势控件属性

属性	说明
曲线数量	要显示的趋势曲线数量
显示时长	要显示的时间窗口
结束时间	时间窗口的结束时间 可用于滚动时间窗口的参数。当为 0 时，结束时间为当前系统时间。
起始位置	当打开页面时，指定在何处开始绘制曲线(可以向左、向中或向右)。
行为	定义： <ul style="list-style-type: none"> • Y 轴的最小值/最大值 • 在轴上绘制的票证 • 背景图片
文本	趋势标题和字体属性(字体大小、标记等)
网格	网格显示的属性(颜色)
光标	光标指针显示的属性(启用和颜色)
X 刻度	X 刻度显示的属性

属性	说明
Y 刻度	Y 刻度显示的属性
曲线“n”	会在趋势控件中显示的标签。



可以使用**附加到**对话框中的 X 表格换算标签值。请参阅“**附属于**”参数 在本页 39 了解详细信息。

历史趋势控件

可以使用“历史趋势控件”分析从数据记录器收集和存储的数据。

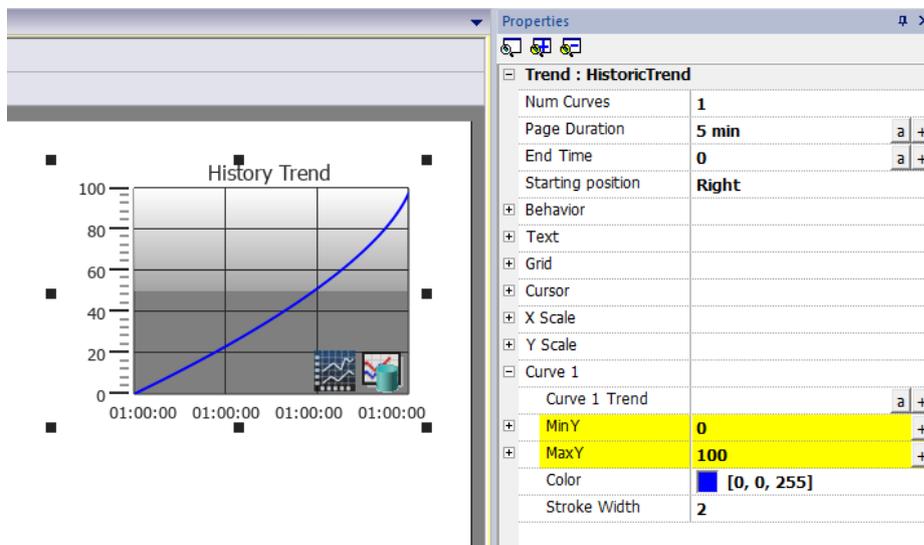
这个过程包括两步：

- 首先创建一个趋势缓冲区，在特定时间点收集指定标签的数据，
- 然后配置历史趋势文件，在图表格式中显示收集的数据。

请参阅“**数据记录**”在本页 256 了解如何创建趋势缓存的详细信息

要显示历史趋势：

1. 将**历史趋势**控件从控件库拖放到页面中。



2. 将您要绘制的趋势缓冲区附加到**曲线 n 值**。数据始终按照时间绘制。

历史趋势控件属性

属性	说明
曲线数量	要显示的趋势曲线数量
显示时长	要显示的时间窗口
结束时间	时间窗口的结束时间

属性	说明
	可用于滚动时间窗口的参数。当为 0 时，结束时间为当前系统时间。
起始位置	当打开页面时，指定在何处开始绘制曲线(可以向左、向中或向右)。
行为	定义： <ul style="list-style-type: none"> • Y轴的最小值/最大值 • 在轴上绘制的票证 • 背景图片
文本	趋势标题和字体属性(字体大小、标记等)
网格	网格显示的属性(颜色)
光标	光标指针显示的属性(启用和颜色)
X 刻度	X 刻度显示的属性
Y 刻度	Y 刻度显示的属性
曲线“n”	缓冲区包含要在趋势控件中绘制的标签值。



可以使用**附加到**对话框中的 X 表格换算标签值。请参阅“**附属于**”参数”在本页 39 了解详细信息。

打印历史趋势控件

可从打印报表库查找并使用历史趋势控件。

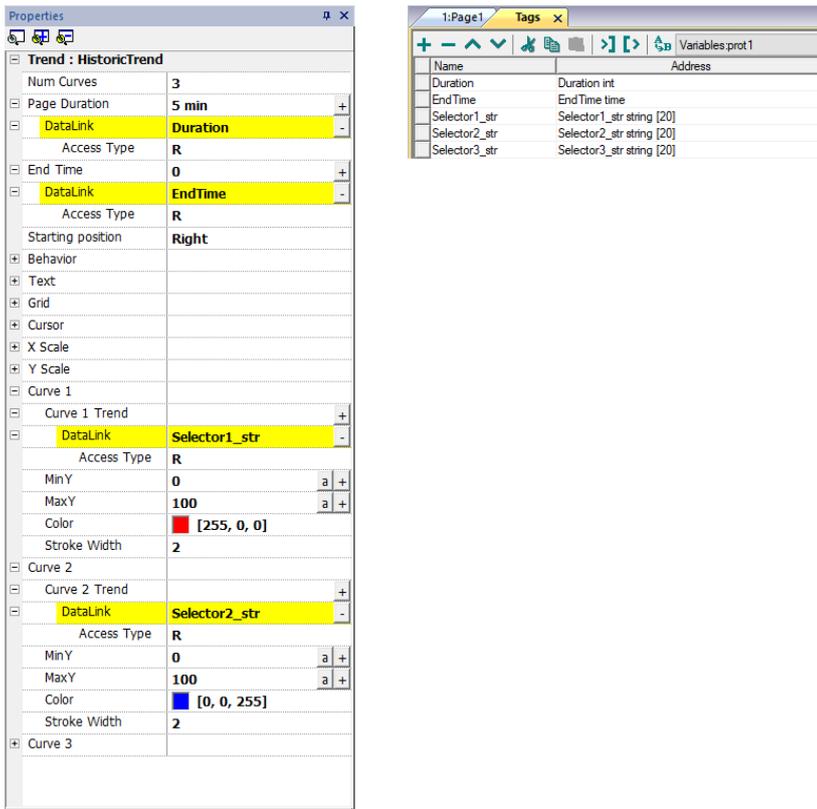
使用标签功能的添加附件，可在运行时使用标签定义要打印的历史趋势的某些属性：

- 显示时长
- 结束时间
- 曲线名称

“页面持续时间”和“结束时间”定义要打印的趋势缓冲区部分。“曲线名称”可用于选择要显示的曲线。清空字符串意味着没有曲线可显示。

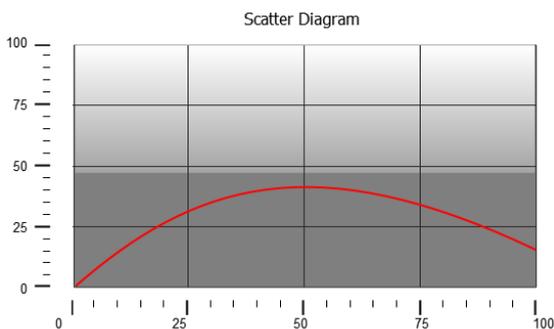


设置趋势视图() 和 滚动趋势时间()显示宏，且对报表打印没有影响。



散布图控件

散布图是使用笛卡尔坐标数据显示两个变量值的图表类型。数据显示为点的集合，每个点有一个变量值确定横轴的位置和另一个变量值确定在纵轴的位置。因此通常被称作 **XY图**。



散布图曲线通过点的线形插值获得。要创建新的散布图：

1. 将**散布图**控件添加到页面中。
2. 选择要显示的曲线数量：各条曲线被命名为图 1、图 2，依此类推
3. 自定义图的常规属性，如 **X 最小**、**X 最大**、**网格** 详细信息。
4. 通过设置**最大样本**参数对每一条曲线的样本/值最大数量进行定义。

在此处设置要在图表上显示的最大值，从数组的第一个元素开始。

For example: Tag1[20] 和最大样本 = 10 将会显示 Tag1 数组前 10 个元素。

5. 定义每条曲线要显示的数组类型的两个标签(**X 标签**和**Y 标签**)。

当数组标签更改时,您可以使用**刷新趋势**动作强制刷新。



注:散布图只支持**刷新趋势**动作。

打印散布图控件

可从打印报表库查找并使用散布图控件。请注意,使用标签功能的添加附件,可在运行时使用标签定义散布图的某些属性。

趋势控件手势



请注意,一些属性只有在属性窗格以高级视图显示时可用。

范围外的值或无效值

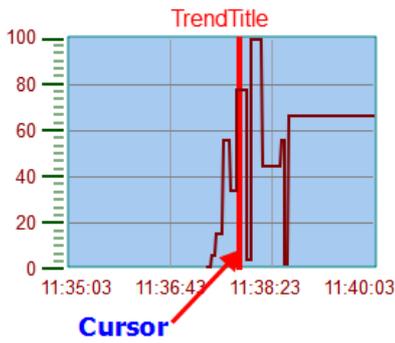
当趋势值超过对该趋势控件设置的限制,会显示一行虚线。当标签值不可用时,例如控制器设备断开,无发出曲线。



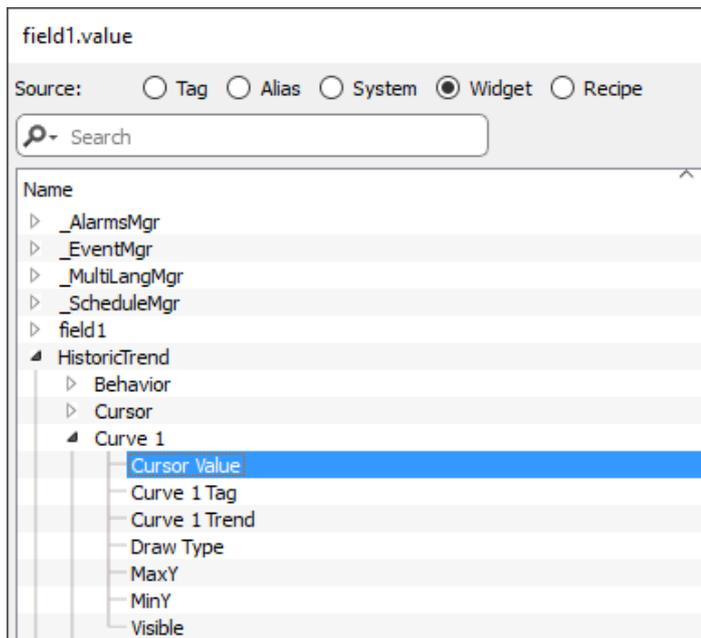
显示趋势值(光标)

趋势指针在某一特定点显示趋势值。

使用动作**显示趋势光标**和**滚动趋势光标**来启用趋势光标并将其移到要求的点上,及时获得曲线上具体点的值。

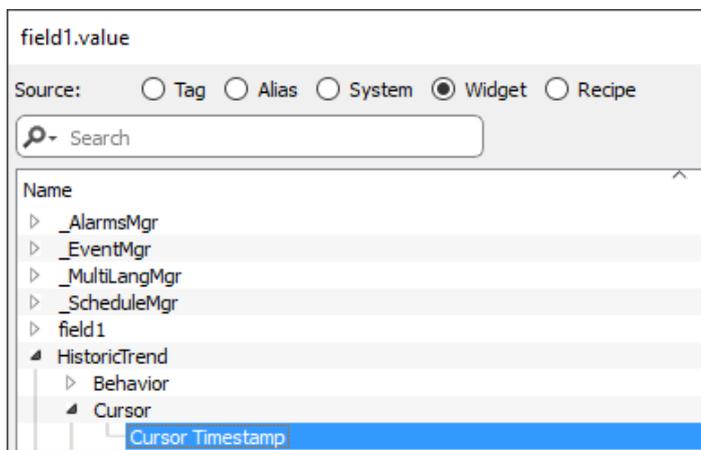


要在页面上显示趋势光标的值，请定义一个数值字段，并将其附加到**光标值**控件标签上。



在该示例中，指针Y轴的值被显示。

要在光标位置显示趋势时间戳，请使用“时间/数据”控件(在“基本”->“控件”类别中可用)，并将控件的值附加到趋势控件的光标时间戳属性。



在运行时修改趋势控件属性

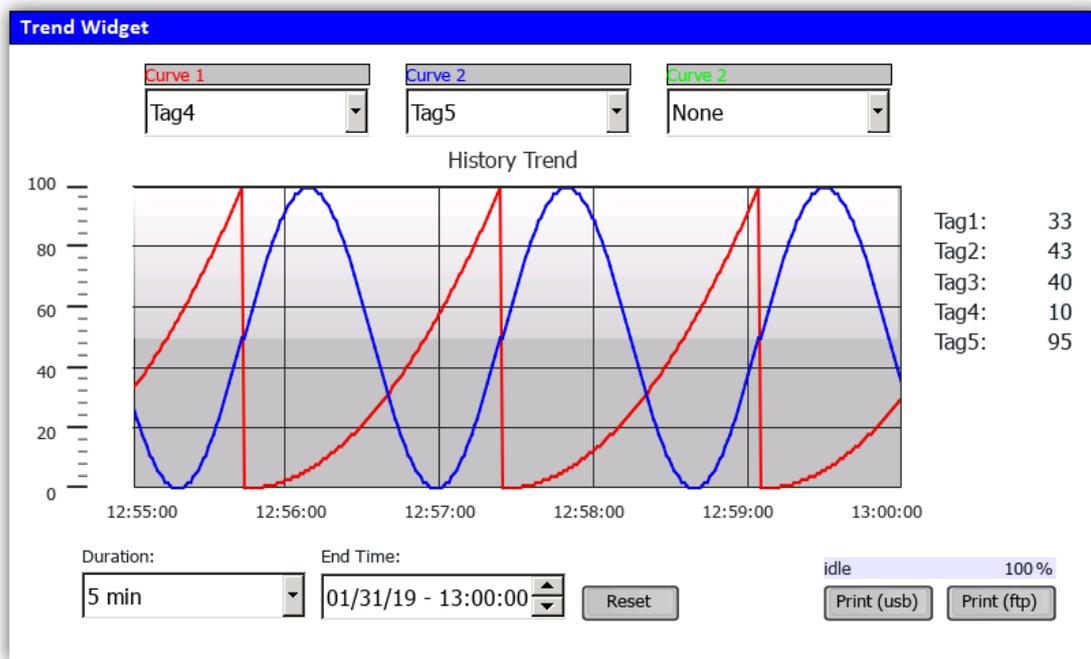
使用标签功能的添加附件，可在运行时使用标签修改趋势控件的某些属性。

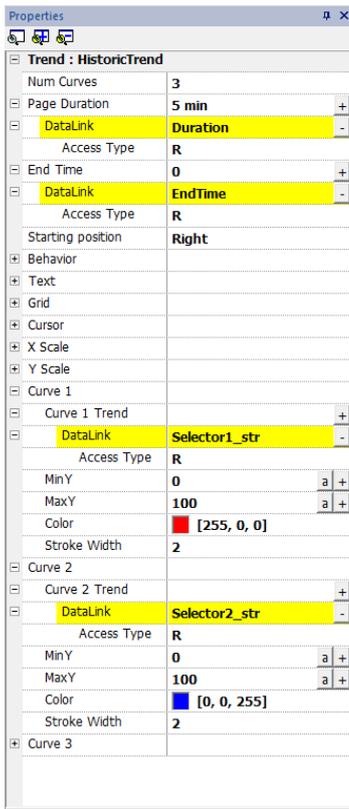
示例 1

使用：

- 显示时长
- 结束时间
- 曲线名称

可以从运行时应用程序修改缩放系数、窗口周期和要绘制的曲线。

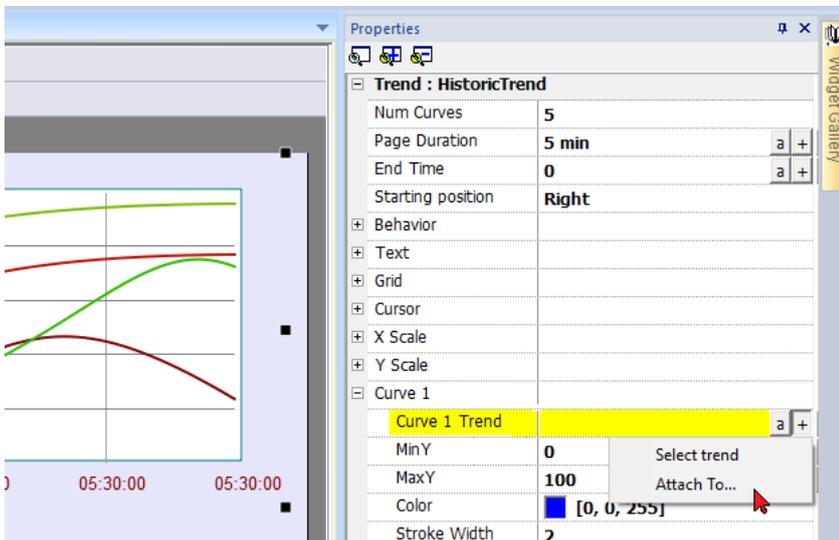


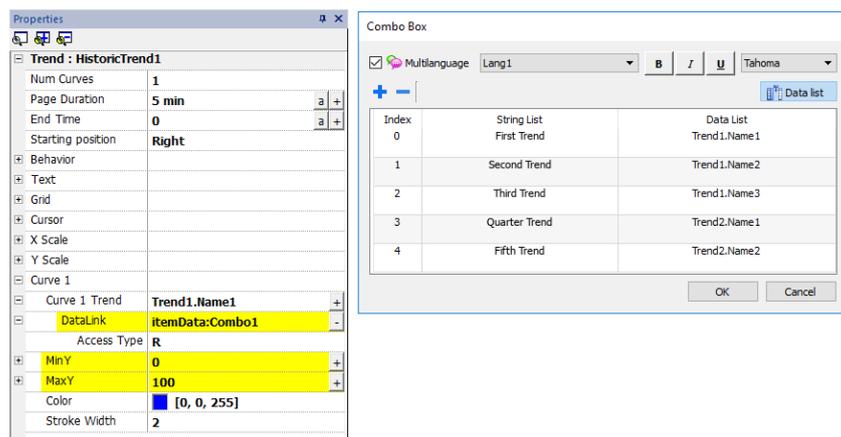


Name	Address
Duration	Duration int
EndTime	EndTime time
Selector1_str	Selector1_str string [20]
Selector2_str	Selector2_str string [20]
Selector3_str	Selector3_str string [20]

示例 2

曲线属性可以附加到组合框，以选择要绘制的曲线





趋势控件手势

趋势控件支持手势命令：

手势	说明
平移	触摸控件以在控制区内滚动曲线
缩放	使用两个手指捏住曲线并执行变焦操作。



警告：仅多点触摸 HMI 设备可缩放



注：为了支持 Y 轴上的手势，趋势控件的最小/最大属性必须链接至行为参数的最小值/最大值(默认用于新趋势)。

Properties		↑	×
Trend : RealtimeTrend			
Num Curves	1		
Page Duration	5 min		+
Y Page Size	100		+
Starting position	Right		
Behavior			
Min Y	0		+
Max Y	100		+
X Labels	4		+
Y Labels	6		+
Background Image	true		
Text			
Grid			
Cursor			
X Scale			
Y Scale			
Min	0		+
DataLink	y0:RealtimeTrend.wnd		-
Access Type	R		
Max	100		+
DataLink	y1:RealtimeTrend.wnd		-
Access Type	R		

申请样本

申请样本属性可以针对每条曲线设置，指示控件从趋势缓存一次读出的最大样本数量。



提示：通常您不需要修改默认值。在趋势控件刷新时将其调整为微调性能。

色带

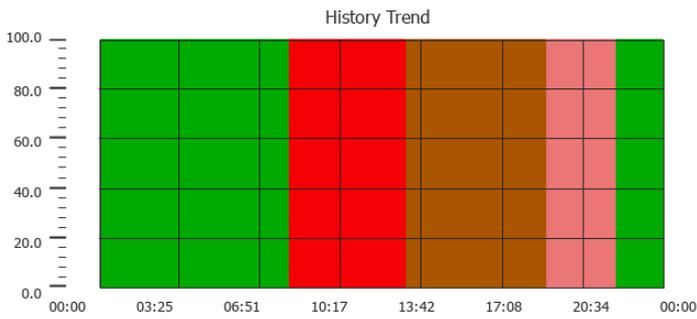
使用颜色带配置对图表背景定制，例如让特定的日期或小时(周末、晚上等)突出。

1. 在属性窗格的色带属性中，单击 **+**:配置色带窗口显示。
2. 单击 **+**添加您需要的颜色。
3. 选择多个单元，点击色带指定颜色到所选单元范围中。



注:该功能只使用趋势控件中的本地时间，而不是全球时间选项。

日历色带示例



表格趋势控件

路径:控件库 > 基本 > 趋势/图形

显示控件中的趋势缓冲区内容

Trend Table

From: 06/20/18 - 08:34:44
To: 06/20/18 - 12:34:44

Duration: 4 Hours

Refresh

TimeStamp	Name1	Name2	Name3	Name4	Name5
06/20/18 - 12:34:31	0	0	0	0	0
06/20/18 - 12:34:32	1	2	3	4	4
06/20/18 - 12:34:33	2	2	6	8	8
06/20/18 - 12:34:34	3	2	9	12	12
06/20/18 - 12:34:35	4	2	12	16	16
06/20/18 - 12:34:36	5	2	15	20	20
06/20/18 - 12:34:37	6	2	18	24	24

Backward

Forward

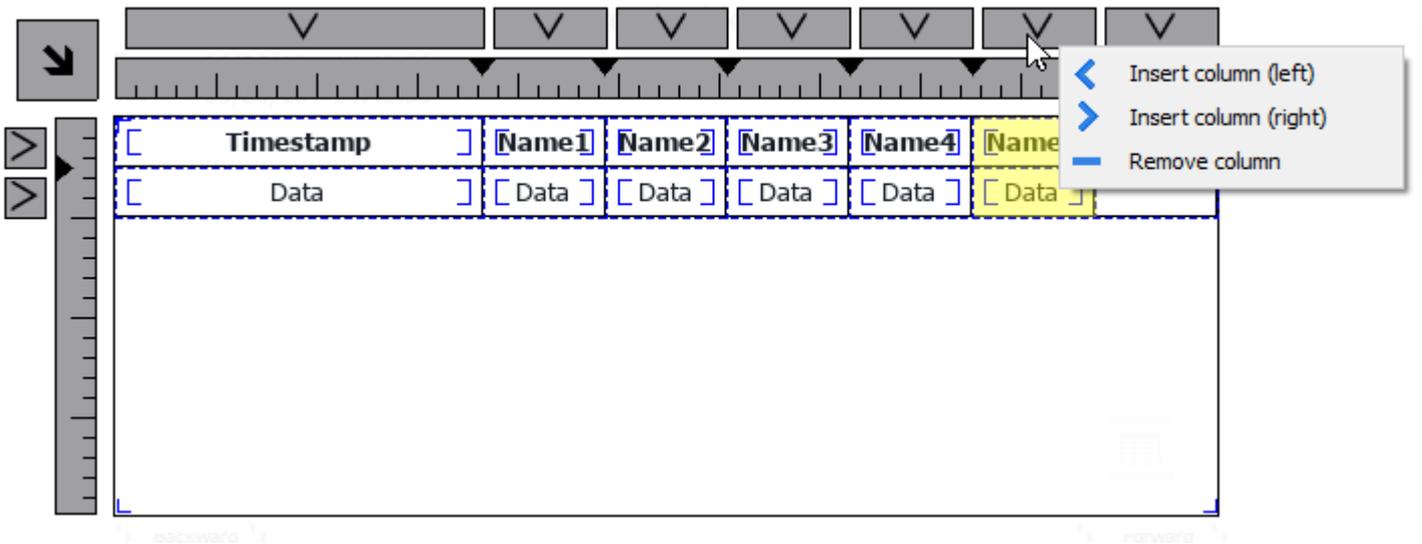
按钮:

- 刷新
从内部缓冲区中检索趋势数据并刷新表格视图
- 向后/向前
向前或向后移动按持续时间参数指定的显示窗口

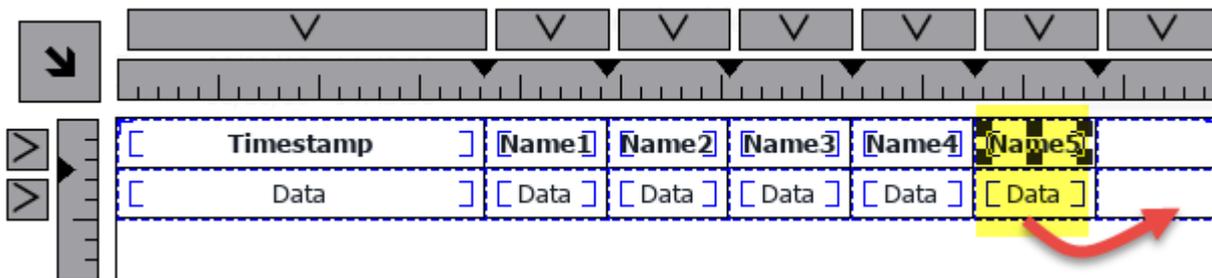
参数	说明
TrendName	从中检索样本的趋势缓冲区(请参阅"数据记录"在本页 256)
标题	标题标签 LRH SW编辑器内的可见标签只是占位符,将显示的实际标签由趋势配置定义(请参阅"数据记录"在本页 256)
显示时长	要显示的时间窗口
结束时间	时间窗口的结束时间 可用于滚动时间窗口的参数。当为 0 时,结束时间为当前系统时间。
时间规格	时间格式: <ul style="list-style-type: none"> • Local = 显示 HMI 设备的时间值。 • Global = 显示使用 UTC 格式的时间值。
日期格式	选择日期和时间格式
表格布局	定义滚动条的特点并允许删除表格的标题

添加或移除趋势列

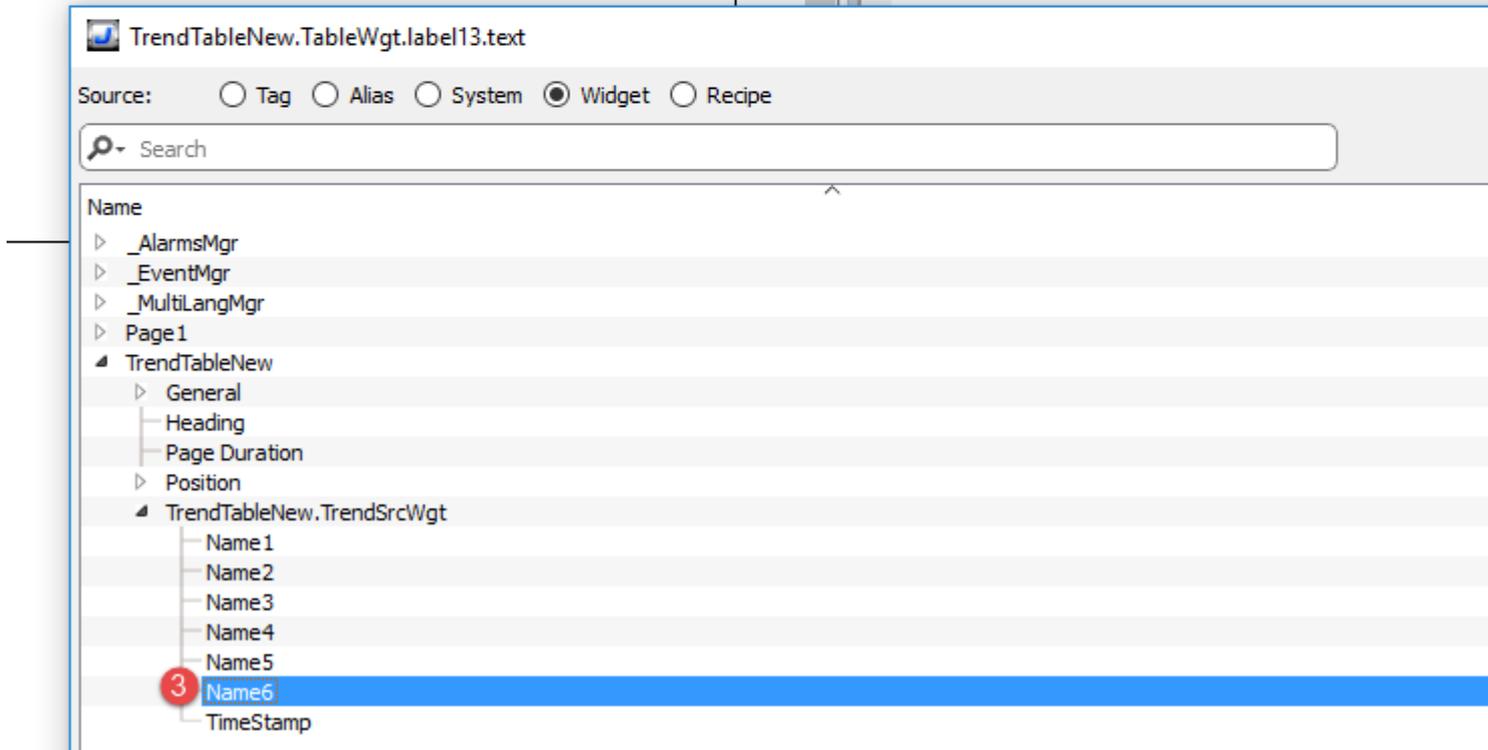
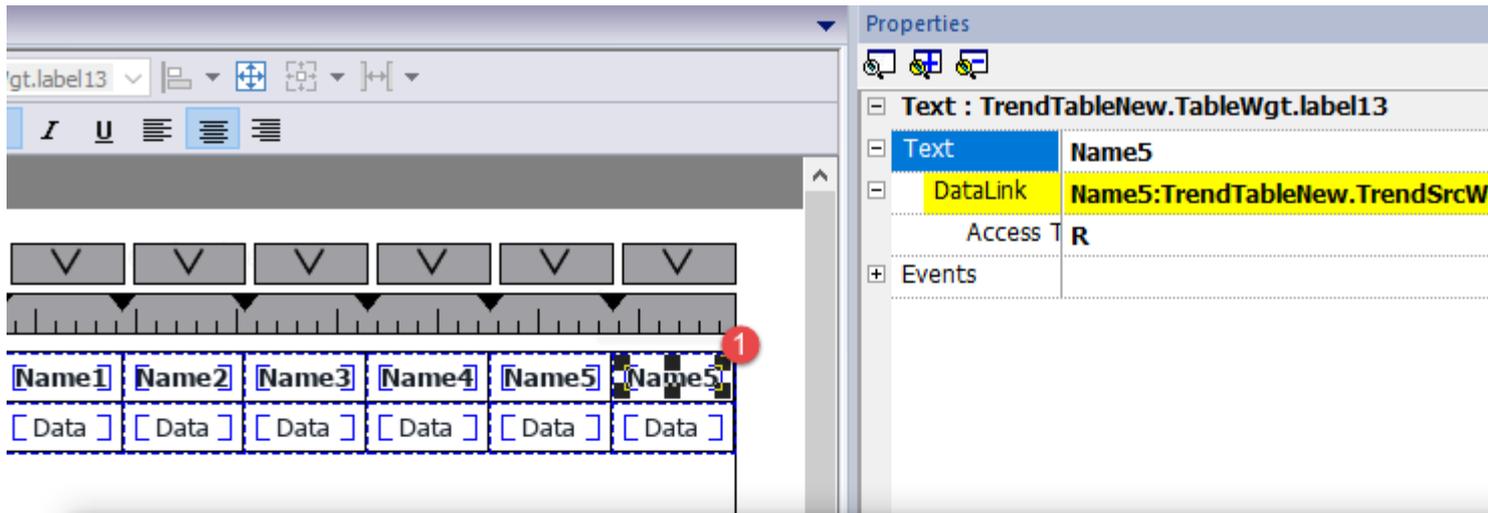
要添加或移除列,请双击网格以进入编辑模式并右击列选择器以打开从中插入或移除列的上下文菜单。

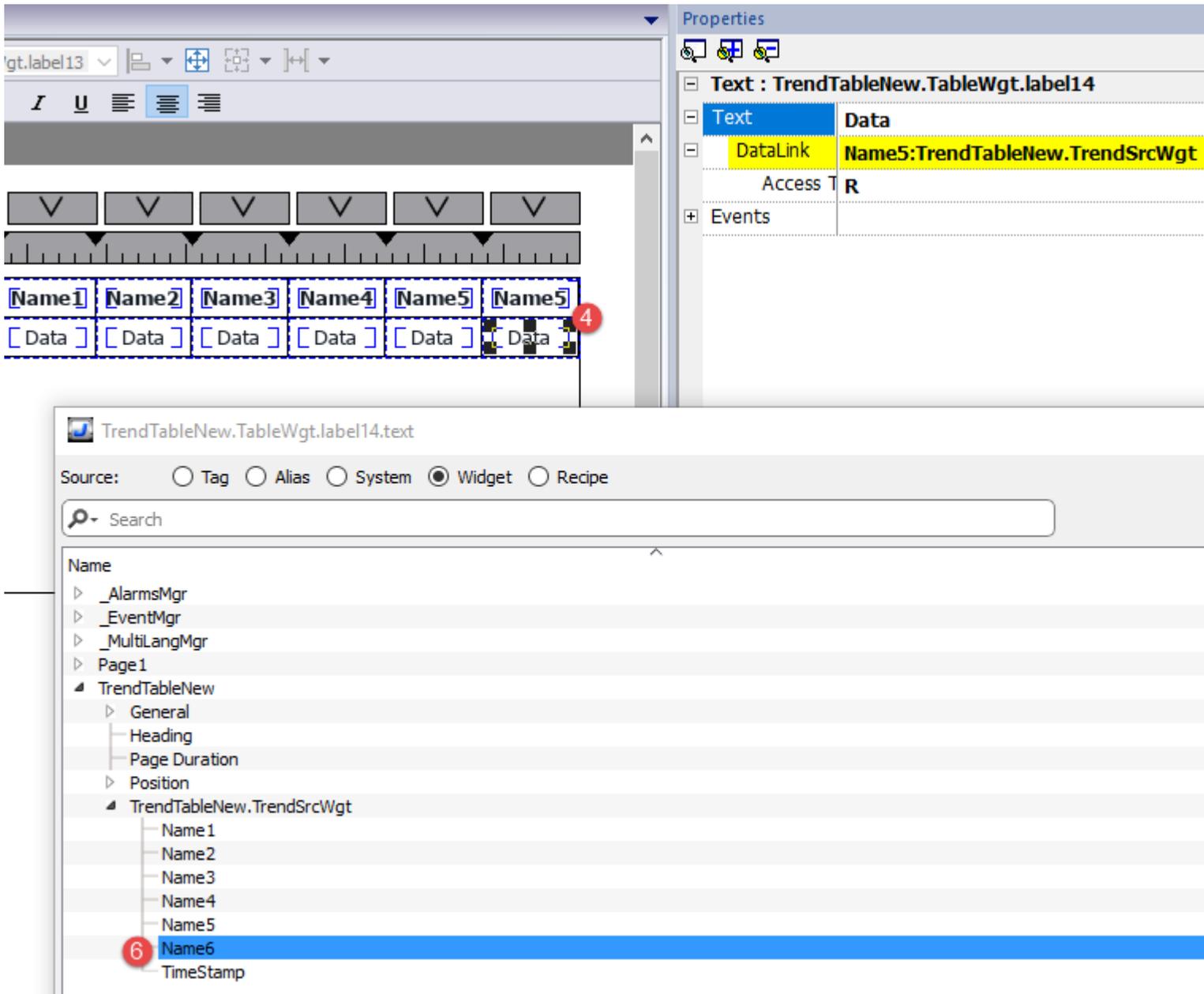


从另一列复制粘贴字段



然后使用属性面板选择要添加至新列的趋势元素





打印趋势表格

可从打印报表库查找并使用没有按钮的趋势表格控件。可以绘制表格并扩大至整个页面。如果要打印的行数大于一页，则将使用其他页面来打印趋势表格。请参阅“功能限制表”在本页 524，了解可打印的最大行数

使用标签功能的添加附件，可在运行时使用标签定义要打印的历史趋势的某些属性：

- 显示时长
- 结束时间
- 曲线名称

“页面持续时间”和“结束时间”定义要打印的趋势缓冲区部分。“曲线名称”可用于选择要显示的曲线。清空字符串意味着没有曲线可显示。



设置趋势视图() 和 滚动趋势时间()显示宏, 且对报表打印没有影响。

Properties

Trend : HistoricTrend

Num Curves 3

Page Duration 5 min

DataLink Duration

Access Type R

End Time 0

DataLink EndTime

Access Type R

Starting position Right

Behavior

Text

Grid

Cursor

X Scale

Y Scale

Curve 1

Curve 1 Trend

DataLink Selector1_str

Access Type R

MinY 0

MaxY 100

Color [255, 0, 0]

Stroke Width 2

Curve 2

Curve 2 Trend

DataLink Selector2_str

Access Type R

MinY 0

MaxY 100

Color [0, 0, 255]

Stroke Width 2

Curve 3

1:Page1 Tags

Name	Address
Duration	Duration int
EndTime	EndTime time
Selector1_str	Selector1_str string [20]
Selector2_str	Selector2_str string [20]
Selector3_str	Selector3_str string [20]

21 数据传输

数据传输可以从一个设备向另一个设备传输变量数据。使用 HMI 设备的该功能可以当作两个设备间的网关操作，即使设备使用的不是相同的通信协议。

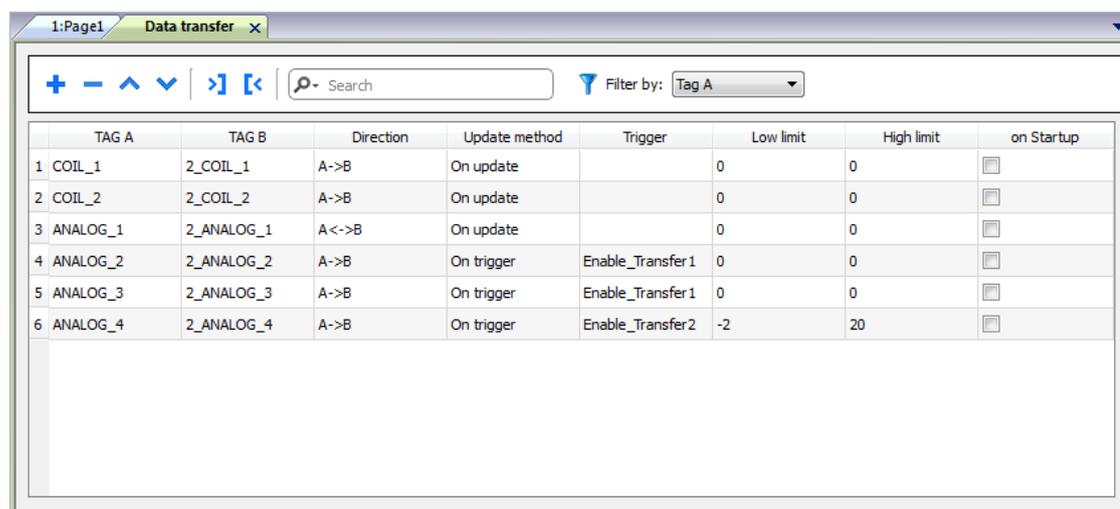
数据传输编辑器	280
导出数据到 .csv 文件	282
数据传输限制和建议	282

数据传输编辑器

路径: 项目视图 > 配置 > 双击 **数据传输**

使用数据传输编辑器映射传输规则。

数据传输编辑器中的每条线定义了两个标签之间的映射规则。如果您需要不同的方向、升级方法或触发器，可以定义更多的映射规则。



	TAG A	TAG B	Direction	Update method	Trigger	Low limit	High limit	on Startup
1	COIL_1	2_COIL_1	A->B	On update		0	0	<input type="checkbox"/>
2	COIL_2	2_COIL_2	A->B	On update		0	0	<input type="checkbox"/>
3	ANALOG_1	2_ANALOG_1	A<->B	On update		0	0	<input type="checkbox"/>
4	ANALOG_2	2_ANALOG_2	A->B	On trigger	Enable_Transfer 1	0	0	<input type="checkbox"/>
5	ANALOG_3	2_ANALOG_3	A->B	On trigger	Enable_Transfer 1	0	0	<input type="checkbox"/>
6	ANALOG_4	2_ANALOG_4	A->B	On trigger	Enable_Transfer 2	-2	20	<input type="checkbox"/>

要添加新规则，请单击 **+**：系统将添加新的标签行。

数据传输工具栏

参数	说明
导入 / 导出	从一个 .csv 文件导入数据传输设置或将数据传输设置导出到 .csv 文件中。
搜索	仅显示包含搜索关键词的行。
筛选方式	仅显示匹配过滤和搜索域的行。

数据传输参数

参数	说明
TAG A / TAG B	要映射经过 HMI 设备交换的标签对。
方向	传输方向。 A->B 和 B->A : 单向传输，始终从一个标签复制值，并按指定的方向将其发送到另一个标签。 A<->B : 双向传输，值在两个标签间相互传输。
更新方法	触发时 : 当标签的值触发器标签的变化高于或低于设置的限制值时，数据传输发生。触发器标签更新时，根据先前的值重新计算容差。  注: 该方法仅适用于单向传输 (A->B 或 B->A)。

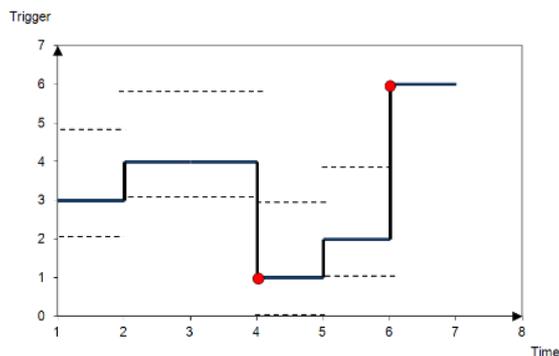
参数	说明
	<p>更新时:在源标签的值改变时,数据传输发生。</p> <p> 注:该方法适用于单向和双向传输(A->B、B->A和A<->B)。</p> <p> 注:Runtime根据标签编辑器比率参数循环监测源标签变化(使用触发器时的触发标签,或使用更新时的传输标签)。如果源标签的比率设置为500 ms(默认),系统每500 ms检查一次更新。</p> <p> 注:如果源标签的更改比比率快,则可能检测不到。</p>
触发器, 上限, 下限	<p>启动数据传输过程的标签。当该标签更改的值在上限和下限的界限之外时,数据传输启动。触发器标签更新时,根据先前的值重新计算容差。如果更改在限制范围内,则不会有动作。</p> <p>该机制使得数据传输仅在参照值发生重大变化时发生。</p> <p>下限小于或等于零。</p> <p> 注:如果下限和上限都设为“0”,则只要触发器标签值更改,数据传输就会发生。</p>
启动时	<p>选中时,强制执行数据传输:</p> <ul style="list-style-type: none"> • 在HMI启动时执行(如果源标签有效) • 在出现通信错误后执行(关联设备节点返回活动状态时) <p>请参阅“对象”在本页465了解有关性质的详细信息。</p> <p> 重要:数据传输在启动时执行可能对HMI设备启动时间有很大的影响。仅在需要时启动该选项。</p>

限制设置的例子

上限 = 1,9

下限 = -0,9

• = 触发数据传输的点



导出数据到 .csv 文件

配置数据传输的信息可以导出到一个 .csv 文件。

.csv 文件中数据传输设置的例子

A	B	C	D	E	F	G	H	I	J
COIL_1	2_COIL_1	A->B	On update		0	0	data1	true	1
COIL_2	2_COIL_2	A->B	On update		0	0	data2	true	1
ANALOG_1	2_ANALOG_1	A<->B	On update		0	0	data3	true	1
ANALOG_2	2_ANALOG_2	A->B	On trigger	Enable_Transfer1	0	0	data4	true	1
ANALOG_3	2_ANALOG_3	B->A	On trigger	Enable_Transfer1	0	0	data5	true	1
ANALOG_4	2_ANALOG_4	A->B	On trigger	Enable_Transfer2	-10	20	data6	true	1

栏	说明
A 到 G	与数据传输编辑器数据相同
H	自动与每一行相关的唯一识别符。  重要: 在您编辑 .csv 文件并添加任一额外行时, 确保在此列中输入唯一标识符。
I 和 J	保留以供将来使用。



导入/导出使用 Windows 定义的分隔符。

数据传输限制和建议

正确的数据传输规则对于 HMI 设备良好的性能很关键。要保证操作和性能的可靠性, 记住以下规则。

关于启动方法

触发时方法只允许单向传输, (A->B 或 B->A)

应首选基于**触发时**模式的数据传输, 因为这样您可以强制传输并只监控触发器标签, 而不是传输中涉及的所有标签。

关于更新方法

更新时方法允许仅当源值改变时根据方向设置更改值。

通过使用**更新时**方法, 您强制系统持续读取定义的所有源标签, 检查是否有需要传输的更改。每个标签更新率的默认值是 500 ms, 可以用标签编辑器修改。

性能观察

数据传输性能取决于：

- 定义的数据传输数
- 最终同时传输的数据量
- 被监控的PLC变量更改的频率



重要：在项目进行中始终测试操作性能。



重要：如果设置不当，数据传输任务可能导致涉及的标签产生循环的情况。辨认和避免此类情况。



提示：根据整个项目的性能，使用计划任务校准更新率。



提示：使用数组类型标签优化数据传输并减少工作量。



提示：减少数据传输数以减少页面更改时间和启动时间。

22 离线节点管理

当与HMI设备通信的一个控制器离线了，系统通信性能最终会下降。

离线节点管理功能会识别离线控制器，并将其从通信中移除直到再次连接。

另外，如果您知道安装的某一控制器将离线一段时间，您可以手动将其停用以使系统性能最优化。



注：该功能不受所有通信协议支持。检查协议文件了解是否支持。

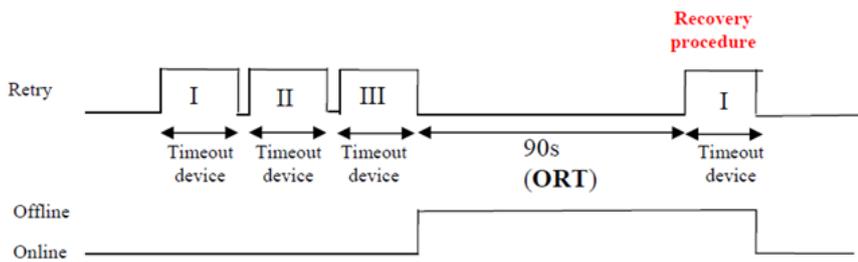
离线节点管理过程	286
手动离线节点管理过程	286
手动离线配置	286
自动脱机节点检测	286

离线节点管理过程

该过程的步骤包括：

- 系统通常使用特定的设备通信。当设备对通信请求没有回复时，系统会在宣布设备离线前重复两次该请求。
- 当设备离线时，系统将向设备发出通信请求，中间间隔更长，间隔称作离线重试超时。如果设备回应其中一个请求，系统会宣布连接，并重新开始正常通信。

该图表显示三次通信超时尝试，离线及恢复的过程。



手动离线节点管理过程

离线节点管理可以手动完成。当特定设备在线并且正常通信时，您可以：

- 使用一个步骤说明设备离线：系统会停止与设备通信。
- 使用一个步骤说明设备连接：系统会重新开始与设备正常通信。

手动离线配置

当您知道与 HMI 设备通信的一些设备将保持离线一段时间时，您可以使用**使能节点**动作将其从通信中排除。



警告：如果相同的项目下载到设备上，所有禁用的设备节点仍将被禁用，反之，如果下载了不同的项目，则所有禁用的设备将会被重新启用。数据包更新也是一样。



提示：要使此功能更动态，您可以选择不指定特定的**节点 ID**，而将其附加到标签值上，或附加到创建的内部变量，以标识可能在网络中安装的不同设备。



注：使用动作**使能节点**强制设备节点回到在线状态时，通信会立刻启动。

自动脱机节点检测

设备不应答通信请求时停用。HMI 设备停止向此设备发送请求。**3 秒**后，HMI 设备发送一条命令检查设备是否可用，如果可用则重启通信，否则下一个超时间隔再次停用。

可以在协议编辑器中修改默认设置。

PLC	Configuration	Dictionaries	Enable Offline Algorithm	Offline Retry Timeout (s)
Modbus RTU.prot1	CfgVer=1 defNodeId=1 timeout=2000	None available...	<input checked="" type="checkbox"/>	3
System Variables.prot2	CfgVer=1 model=Default	None available...	<input type="checkbox"/>	Not applicable



注：并非所有协议都支持此功能。

参数	说明
启用离线算法	启用协议离线管理
离线重试超时	停用设备后重试循环间隔，以秒为单位。范围：1-86.400 秒(24 小时)。

23 多语言

多语言功能是为创建同时包含多重语言文本的HMI应用程序设计。

多语言功能使用编码页面支持来处理不同的语言。编码页面(或文本文件)是在每种语言内使用的字母形状集合。

多语言功能可用于定义项目中的语言和字符集。LRH SW 也扩展 Windows 系统提供的 TrueType 字体，以提供与不同字符集相关的字体外观。

LRH SW 还让您能够为每种支持的语言提供字符串。

LRH SW 还允许您更改显示语言，从而可以在设计阶段了解页面观感。



在 Windows 10 中，默认未预先安装语言字体。要使用一种语言(例如，印度泰卢固语)，首先需要在 PC 中添加该语言(“设置”->“区域和语言”->“添加语言”->“选择泰卢固语”)，然后 Windows 10 将下载所需字体并安装在 PC 中。完成上述步骤后，打开 LRH SW，然后选择泰卢固文书写系统。这次，正确设置了字体，可以在 LRH SW HMI Runtime 中使用。

简体中文的支持字体

对于简体中文，以下字体是支持的：

字体名称	字体文件
仿宋	simfang.ttf
Arial Unicode MS	ARIALUNI.TTF
楷体	simkai.ttf
微软雅黑	msyh.ttf
NSImSun	simsun.ttc
SimHei	simhei.ttf
Simsun	simsun.ttc

繁体中文的支持字体

对于繁体中文，以下字体是支持的：

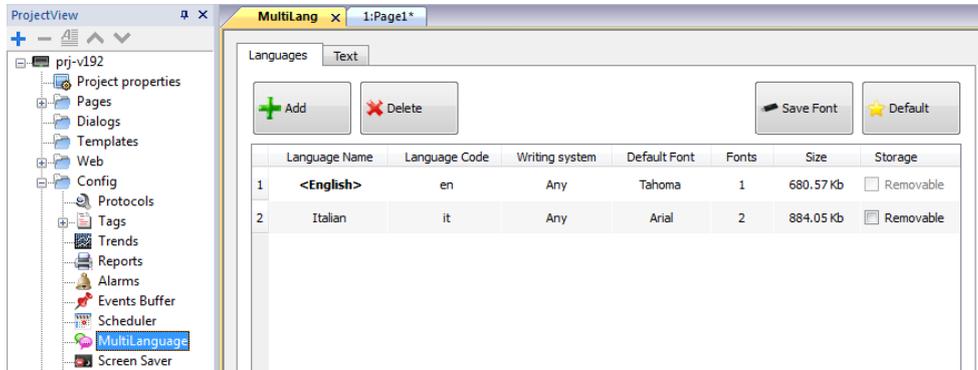
字体名称	字体文件
DFKai-SB	kaiu.ttf
Microsoft Sheng Hai	msjh.ttf
Arial Unicode MS	ARIALUNI.TTF
MingLiU	mingliu.ttc

字体名称	字体文件
PMingLiU	mingliu.ttc
MingLiU_HKSCS	mingliu.ttc

多语言编辑器	291
更改语言	292
多语言控件	292
导出/导入多语言字符串	294
在运行时更改语言	296
Unicode 支持限制	296

多语言编辑器

路径: 项目视图 > 配置 > 双击多语言



语言设置

参数	说明
语言名称	识别项目中语言的名称
语言代码	ISO 639 语言代码标识符, 在从外部 xml 文件导入资源时用于匹配语言项目。
写入系统	选择与语言使用的字体
默认字体	项目控件的默认字体。  注: 在您选择新字体时, 系统会提示您替换已创建控件中使用的字体。
字体	与所选语言相关的字体数量。
大小	用来存储字体文件的存储器。
空间	文件字体的位置是可移动的外部存储器。  提示: 将大字体文件存储在可移动内存中, 释放 HMI 设备的内存需求。

添加语言

1. 在语言选项卡中, 单击 **+**: 表中将添加一行。
2. 进入所有语言设置。
3. 单击 **默认** 设置所选语言为 Runtime 启动时的默认语言。
4. 单击 **保存字体** 将您标记为可移动的字体复制到外部存储器。

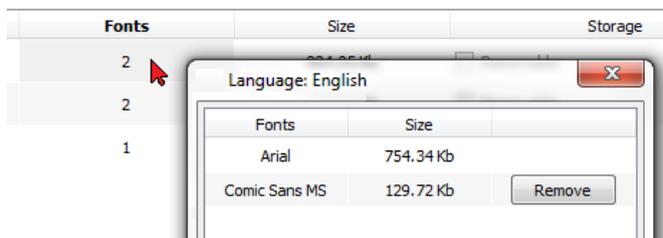


重要: 配置为存储在可移动存储器上的字体文件必须提供给最终用户以在 HMI 设备上完成字体安装。

移除字体

如需移除不再需要的字体：

1. 点击多语言编辑器中的字体编号：出现一个已用字体列表对话框。



2. 选择要移除的字体，然后单击**移除**：被移除字体将替换为默认字体。

更改语言

在页面设计过程中更改语言

在页面设计过程中，可以用组合框更改语言。如果未出现文本，请选中多语言编辑器中的**文本**选项卡，并插入缺失字符串。

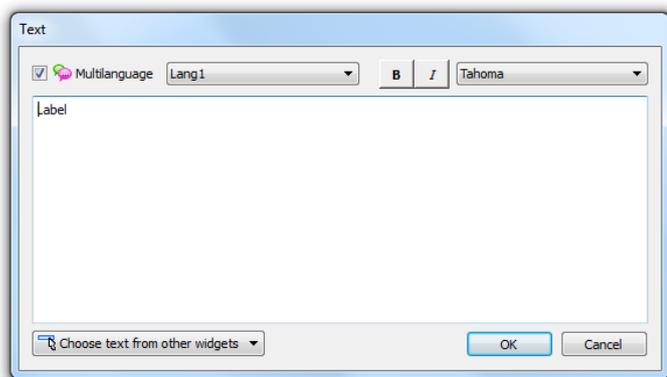


多语言控件

多语言支持按钮、静态文本、信息、报警描述和弹出的信息等目标。

标签控件的多语言

双击页面上的文本控件打开**文本**对话框。



启用/停用多语言功能，编辑所选语言的文本并选择字体。

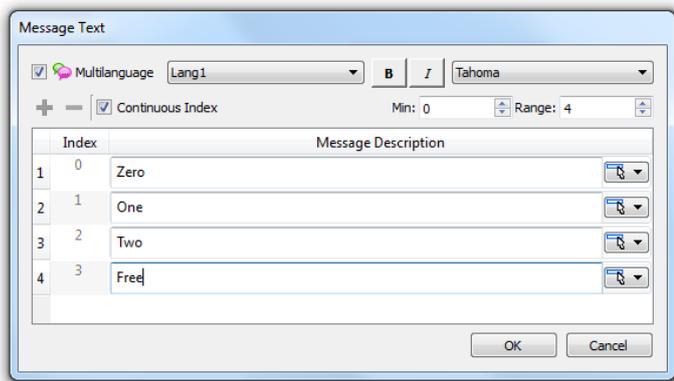


注：在此处对控件设置的粗体、斜体和颜色属性对所有语言适用。

参数	说明
多语言	对该控件启用/停用多语言功能。
从其他控件选择文本	点击按钮浏览项目中已有的信息串，为控件选择文本。

信息控件的多语言

双击页面上的消息控件打开**消息文本**对话框。

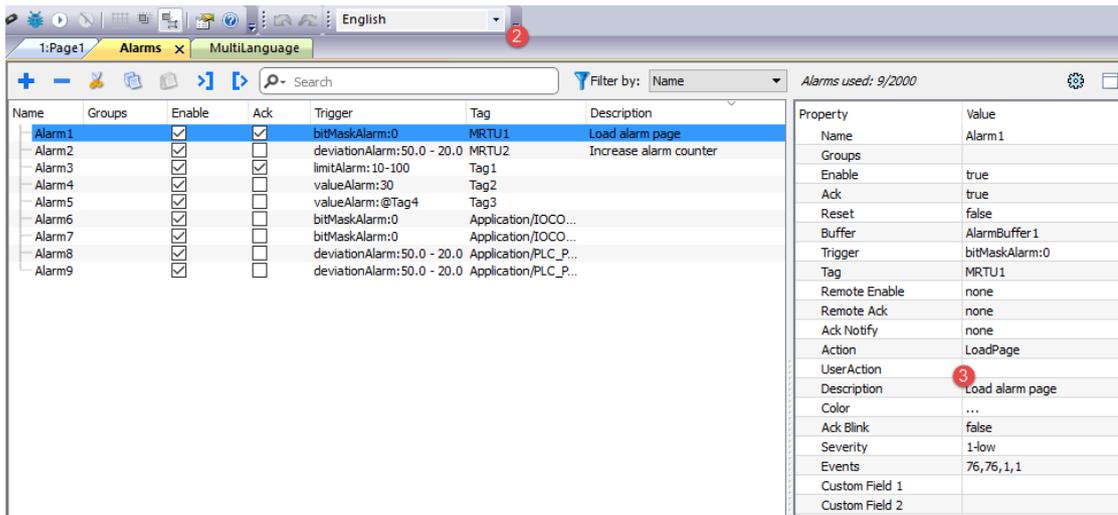


参数	说明
多语言	对该控件启用/停用多语言功能。
连续索引	控件索引被设置为连续的数字(例如 3, 4,5,6)
最小	索引的开始数值
范围	信息的数量
从其他控件选择文本	点击按钮浏览项目中已有的信息串，为控件选择文本。

报警信息的多语言

要对报警信息添加多语言字符串：

1. 打开报警编辑器。
2. 使用语言组合框选择语言。
3. 在**说明**列中输入报警文本。

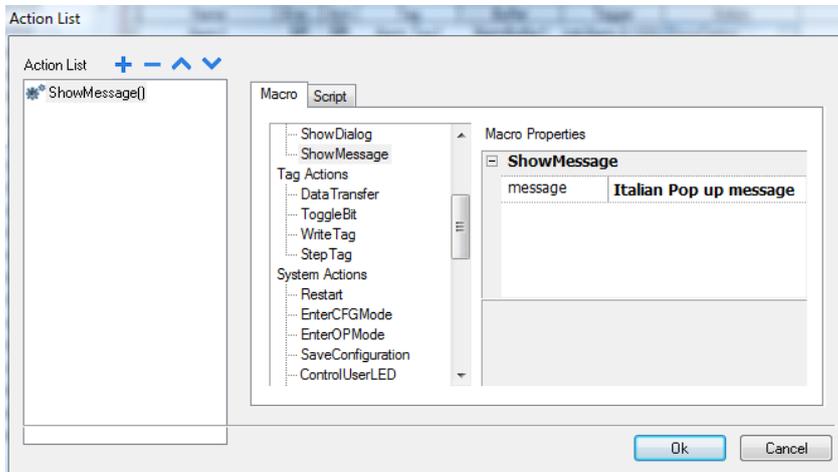


提示:报警控件显示的报警状态的文本标签可被翻译或通过多语言文本编辑器个性化。

弹出信息的多语言

要对弹出信息添加多语言:

1. 从语言组合框中选择一种语言。
2. 添加页面动作**显示信息**并用所选语言输入文本。



导出/导入多语言字符串

将项目译成多语言的最简单的方法是将所有文本导出到一个.csv文件,将结果文件翻译出来,然后将翻译的文本导入到项目中。



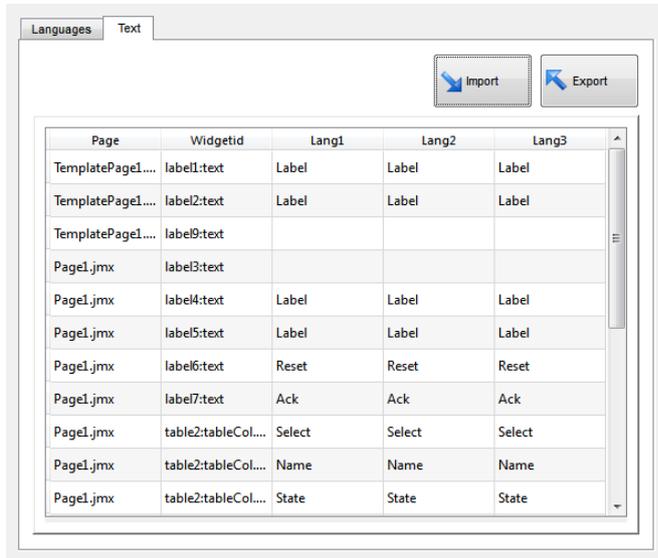
重要:LRH SW 导出的 .csv 文件以 **Unicode** 编码,要将其编辑,您需要一项支持 **Unicode** 编码 .csv 文件的特殊工具。

导出和重新导入字符串

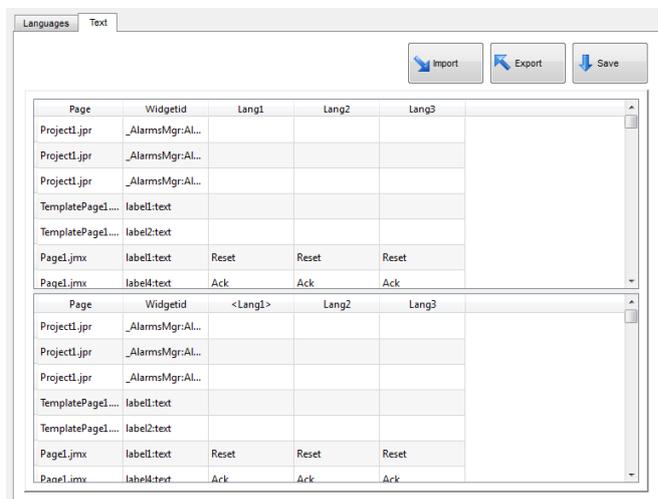
路径: 项目视图 > 配置 > 双击多语言

导出和重新导入多语言字符串:

1. 在文本选项卡中, 单击**导出**: 所有多语言字符串均导出至一个 .csv 文件中。



重要: 设置在导出文件前要在项目中使用的所有语言。这将确保导出的文件包含所有栏和语言定义。



2. 字符串经转换后, 单击**导入**, 将其重新导入到项目中: 字符串将导入, 同时与每个控件的控件 ID 和页码匹配。
3. 单击**保存**, 保存新的控件数据。



注: 要更改导出文件中使用的分隔符, 请更改计算机的区域设置。导入时, 分隔符信息从文件重新获取, 如果未找到, 会使用默认的字符“;”。

导入系统规定参数

以下是导入支持的格式：

- 逗号分隔值 (.csv)
- Unicode 文本 (.txt)



注：当您导入使用 Microsoft® Excel® 修改的文件时，请使用 Unicode 文本文件格式。

在运行时更改语言

对一个动作改变语言

在项目下载后，HMI 运行时将开始使用设为默认的语言。您可以使用 **设置语言** 动作更改语言。请参阅“[多语言动作](#)”在本页 172。



注：语言更改后，它也将在今后的会话中使用。

可从 JavaScript API 获取有效的语言代码。有关更多详细信息，请参阅“[curlLangCode](#)”在本页 483。

丢失的字体

当您更改语言时，如果要求的字体在设备存储中不可用，会弹出消息让您插入包含丢失的字体的存储卡。在操作完成后，您可以移除存储卡。



Unicode 支持限制

LRH SW 与 Unicode 文本兼容。但是，对于与其他平台的兼容性问题，Unicode 仅对属性子集支持。

区域	属性	可接受字符集	保留字符/字符串
协议编辑器	别名	ASCII [32..126]	(空格), ; : . < * > '
标签编辑器	名称	ASCII [32..126]	. \ / * ? : > < " & # % ; =
	编组	ASCII [32..126]	<新> \ / * ? : > < " & # % ;
	注释	Unicode	

区域	属性	可接受字符集	保留字符/字符串
趋势	名称	ASCII [32..126]	\/*?:>< "&#%;
打印报表	名称	ASCII [32..126]	\/*?:>< "&#%;
报警	名称	ASCII [36..126]	\/*?:>< "&#%;
	说明	Unicode	[]-对实时标签,\扩展 seq[和\
事件	缓冲区名称	ASCII [32..126]	\/*?:>< "&#%;
计划任务	名称	ASCII [32..126]	\/*?:>< "&#%;
语言	语言名称	ASCII [32..126]	\/*?:>< "&#%;
	控件中的文本	Unicode	-
	导入的文件中的文本	Unicode	-
用户组	组名称	a-z A-Z _	管理员, 访客, 未授权的
	注释	Unicode	-
用户	名称	ASCII [32..126]	\/*?:>< "&#%;
	密码	Unicode	-
	注释	Unicode	-
配方	名称	ASCII [32..126]	\/*?:.>< "&%;,
	集名称	ASCII [32..126]	\/*?:.>< "&%;,
	元素名称	ASCII [32..126]	\/*?:.>< "&%;,
常规	项目名称	A-Z,a-z,0-9,-,_	"PUBLIC", "readme", "index.html"
	页面名称	A-Z,a-z,0-9,-,_	-
	对话框页面名称	A-Z,a-z,0-9,-,_	-
	模版页面名称	A-Z,a-z,0-9,-,_	-
	键盘名称	A-Z,a-z,0-9,-,_	-
	文件(图像/视频/等)	A-Z,a-z,0-9,-,_	-

区域	属性	可接受字符集	保留字符/字符串
	控件ID	A-Z,a-z,0-9,-,_,	-
Runtime	PLC 通信	UTF-8, Latin1, UCS-2BE, UCS-2LE, UTF-16BE, UTF-16LE	-

24 计划任务

LRH SW 提供能够以设定的间隔或按一定的时间执行特定动作的计划任务。

创建一项计划通常需要两步：

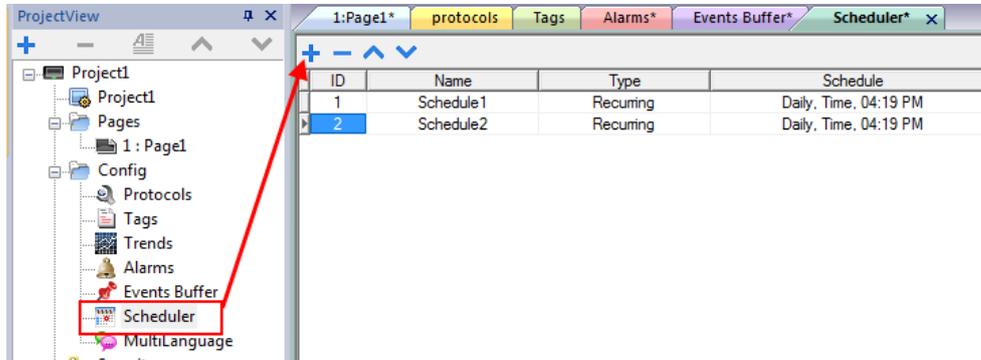
1. 您可以创建一项计划事件发生时要执行的一系列动作的计划。在计划任务编辑器中操作
2. 创建用户界面，允许直接用户更改每项计划的设置。通过向项目页面添加**计划任务**控件并将其配置为适应用户计划需求实现这一点。

创建一个调度程序	300
高分辨率调度表	300
循环调度	300
为调度表配置地址	302
配置计划任务控件	303
运行时计划任务事件	304

创建一个调度程序

路径: 项目视图 > 配置 > 双击计划任务

- 单击 + 添加计划。



计划参数

参数	说明
ID	自动对计划分配唯一的代码
名称	计划名称
类型	计划类型 <ul style="list-style-type: none"> • 循环, 请参阅"循环调度" 向下了解详细信息。 • 高分辨率, 请参阅"高分辨率调度表" 向下了解详细信息
计划	计划任务设置和选项。请参阅"循环调度" 向下了解详细信息。
动作	在计划时间执行的动作
优先级	事件的优先级。如果两个计划同时出现, 具有高优先级的事件将先被执行。

高分辨率调度表

高分辨率计划用于执行需要以指定的间隔重复的动作。执行动作之间的间隔在**计划**列中设置(以毫秒为单位)。

 注: 不能在运行时更改此类计划的设置。如果需要在运行时更改动作时间设置, 请选择**循环**计划, 并将**类型**设为**每种**。请参阅"循环调度" 向下了解详细信息。

循环调度

循环计划用于在特定的时间点执行动作。在运行时可以修改设置。

循环计划任务参数

参数	说明
类型	计划动作的频率
模式	每一种计划任务类型要求的具体设置
条件	<p>boolean 标签 (true/false) 在计时器启动时激活指定的动作。如果标签 = true, 动作将被执行。按照默认, 当计时器启动时执行动作。</p> <p> 注: 只显示附加到 Boolean 数据类型的标签。</p>
动作	<p>在计划时间执行的动作</p> <p> 重要: 在运行时不能修改动作和计划参数。</p>
日期	执行计划动作的日期
时间/偏移	<p>该域显示以下之一:</p> <p>时间 = 执行计划动作的时间</p> <p>偏移 = 根据所选模式出现延迟或提前。</p>
位置	计算日出/日落时间的参考位置。
工作日	执行计划动作的工作日
启动	启动时执行计划
启用计划	启用/停用计划
仅在启动时执行	仅在启动时执行一次计划

计划类型选项

选项	说明
按日期	动作在指定日期和时间执行。
每日	动作在每日的指定时间执行。
每隔	以指定的间隔执行动作(范围:1秒-1天)
每小时	动作在每小时的指定分钟执行。
每月	动作在每月的指定日期和时间执行。
每周	动作在每周的指定工作日和时间执行。
每年	动作在每年的指定日期和时间执行。

计划模式选项

选项	说明
时间	取决于计划类型。您可以指定日期/时间/周数据。
随机 10	动作在设置时间前或后间隔的10分钟内执行。 例如,如果设置时间是 10:30,动作在 10:20 和 10:40 之间的任何时间执行。
随机 20	动作在设置时间前或后间隔的20分钟内执行。 例如,如果设置时间是 10:30,动作在 10:10 和 10:50 之间的任何时间执行。
日出+	动作在日出后指定的延迟时间执行。延迟设置为分钟/小时,日出时间以具体位置为准。
日出-	动作在日出前指定的时间提前执行。提前设置为分钟/小时,日出时间以具体位置为准。
日落+	动作在日落后指定的延迟时间执行。延迟设置为分钟/小时,日落时间以具体位置为准。
日落-	动作在日落前指定的时间提前执行。提前设置为分钟/小时,日落时间以具体位置为准。

请参阅["为调度表配置地址"](#)向下,了解日出和日落设置的详细信息。



注:模式选项并不适用于所有计划类型。

为调度表配置地址

计划的动作可以配置为在特定的时间执行,日出和/或日落时间。要做到这一点,您需要根据 UTC 信息指定正确的位置。系统将自动计算日出和日落的时间。

默认只有几个位置可用。如果您的地址未列出,您可以在 `Target_Location.xml` 文件输入纬度、经度和 UTC 信息添加位置。



重要:每一个平台都有各自的 `Target_Location.xml` 文件。

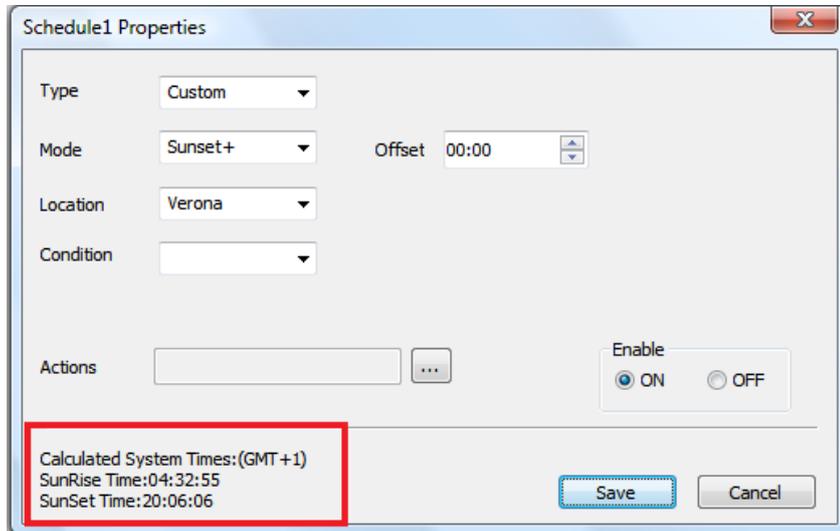
位置文件的路径

应用	位置文件的路径
LRH SW	<code>Lovato Electric\LRH SW\languages\shared\studio\config\Target_Location.xml</code>
HMI 设备	<code>Lovato Electric\LRH SW\runtime\<HW Platform>\config\Target_Location.xml</code>
模拟器	<code>Lovato Electric\LRH SW\simulator\config\Target_Location.xml</code>

例如,维罗纳市(意大利)的信息如下所示:

```
<file city="Verona" latitude="45.44" longitude="10.99" utc="1"/>
```

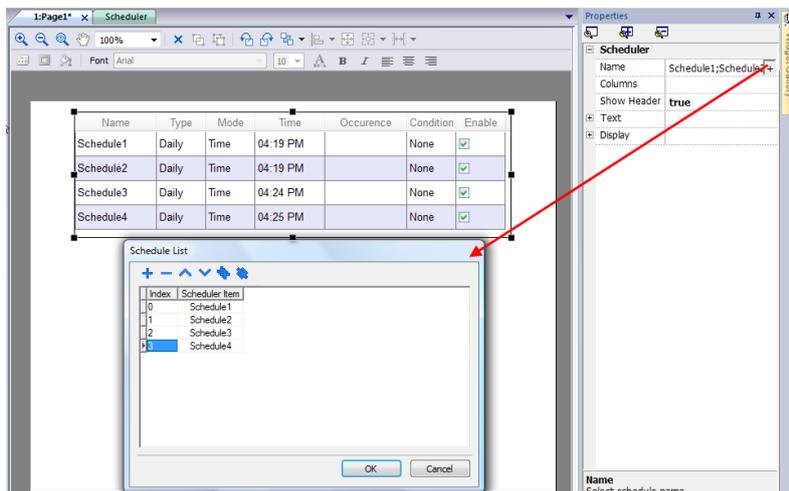
位置信息和日出日落时间一同显示在对话框中。



配置计划任务控件

在一个页面显示计划任务数据：

1. 将**计划任务**控件从控件库拖放到页面中。
2. 在**属性**窗格中，单击**名称**参数后面的 **+**：**计划列表**对话框显示。
3. 添加所有您想显示在该页的计划。



4. 在**属性**窗格中，自定义所有设置。

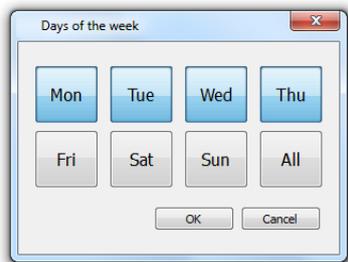
计划任务设置

参数	说明
名称	要现实的计划
列	要显示的栏及其特点
显示标题	显示/隐藏栏标题
时间规格	在运行时显示的时间
文本	文本使用的字体
显示	表格风格

运行时计划任务事件

在运行时您可以修改以下计划参数。

Name	Type	Mode	Time	Occurence	Condition	Enable
Schedule1	By Date	Time	11:01	JUN 20, 2013	None	<input checked="" type="checkbox"/>
Schedule3	Monthly	Sunrise+	11:01	Day : 3	None	<input checked="" type="checkbox"/>
Schedule4	Weekly	Rando...	16:19	MTWTFSS	None	<input checked="" type="checkbox"/>
Schedule5	Yearly	Time	01:00			
Schedule6	Custom	Time	01:16			



参数	说明
发生	计划类型和执行时间的信息
条件	用于动作执行的条件
启用	不删除计划的情况下启用/停用计划动作的执行。

请参阅["循环调度"](#)在本页300了解计划参数的详细信息。

25 21 CFR 第 11 部分合规性

LRH SW 包括一组功能，以响应 FDA 21 CFR 第 11 部分中规定的要求。该标准旨在为安全处理工业应用中的电子记录和电子签名提供解决方案。

该表列出了本条例规定的所有要求并报告 LRH SW 中可供遵守的功能。



在应用程序开发期间，FDA 21 CFR 第 11 部分合规性可选，且应用程序开发人员负责适当配置该应用程序。

章节	说明	LRH SW 合规水平 (v2.8)
11.10(a)	(a) 验证系统以确保准确性、可靠性、一致预期性能并可识别无效或更改的记录。	<p>可使用 x.509 证书对由 LRH SW 生成的报告进行签名。验证报告签名所需的证书(包括公共密钥)将与该报告一起导出。</p> <p>引用：</p> <ul style="list-style-type: none"> • "保存事件存档" 在本页 199 • "打印图形报表" 在本页 184
11.10(b)	能够以用户可读取和电子格式生成准确完整的记录副本，适用于机构检查、审查和复制。如对机构是否能够对电子记录执行审查和复制有任何疑问，相关人员应联系该机构。	<p>应用程序开发人员可以选择将其更改跟踪到日志的资源(过程值、报警等)。所选资源的每次更改将连同执行该更改的操作员的名称一起记录。日志报告可以导出到 .csv 或 .pdf 文件。</p> <p>引用：</p> <ul style="list-style-type: none"> • "启用/停用日志" 在本页 330 • "导出日志到 .csv 文件" 在本页 335 • "保存事件存档" 在本页 199 • "打印日志表格" 在本页 335 • "打印图形报表" 在本页 184
11.10(c)	保护记录，使其能够在整个记录保存期内准确、随时检索。	<p>可开发应用程序，以便在预定义时间间隔内(例如，在一天结束时)或当循环缓冲区已满时将已签名报告自生成到外部存储器或网络文件夹中。用户负责在保留期间保存这些记录。</p> <p>引用：</p> <ul style="list-style-type: none"> • "保存事件存档" 在本页 199 • "打印图形报表" 在本页 184 • "计划任务" 在本页 299
11.10(d)	限制系统访问经授权的个人。	<p>应用程序开发人员负责对该应用程序进行适当的安全配置。</p> <p>引用：</p>

章节	说明	LRH SW 合规水平 (v2.8)
		<ul style="list-style-type: none"> • "用户管理和密码" 在本页 317
11.10(e)	使用安全、由计算机生成、加盖时间戳的日志来独立记录用于创建、修改或删除电子记录的操作员条目和动作的日期和时间。记录更改不应掩盖之前记录的信息。该日志文档至少应保留一段时间直至主题电子记录所需的时间为止，并应可供机构审查和复制。	<p>日志记录使用循环缓冲区进行存储(为确保设备不会耗尽内存)。操作员无法修改日志。每个记录包含序列号，以便轻松检查所有记录是否存在。可开发该应用程序，以便定期(例如，在每天结束时)保存/导出数据副本；操作员负责将报告副本存储在安全位置。</p> <p>引用：</p> <ul style="list-style-type: none"> • "导出日志到 . csv 文件" 在本页 335 • "保存事件存档" 在本页 199 • "打印日志表格" 在本页 335 • "打印图形报表" 在本页 184 • "计划任务" 在本页 299
11.10(f)	适当时，使用操作系统检查来执行允许的步骤和事件排序。	宏或 JavaScript 可用于配置应用程序中的命令序列。
11.10(g)	使用权限检查，以确保只有经授权的个人才能使用该系统、以电子方式签署记录、访问操作或计算机系统输入或输出设备、更改记录或执行当前操作。	<p>HMI 应用程序可配置为</p> <ul style="list-style-type: none"> • 仅当用户使用其自己的密码登录后才可访问 • 根据登录到系统的用户，对象可配置为可用或不可用 • 资源可配置为在修改之前需要密码确认 <p>引用：</p> <ul style="list-style-type: none"> • "用户管理和密码" 在本页 317 • "电子签名" 在本页 331
11.10(h)	适当时，使用设备(例如终端)检查确定数据输入源或操作指令的有效性。	<p>资源可配置为仅从所选用户组中可访问。可在“用户管理”设置中配置允许的 IP 地址列表。</p> <p>引用：</p> <ul style="list-style-type: none"> • "修改访问权" 在本页 319
11.10(i)	确定开发、维护或使用电子记录/电子签名系统的人员接受执行其指定任务的教育、培训并具备相关经验。	应用程序开发人员负责定义并将适当的用户权限分配给在 HMI 设备上具有访问权限的每个用户
11.10(j)	制定和遵守追究个人在其电子签名下采取行动的责任的书面政策，以防伪造记录和签名。	应用程序开发人员负责建立适当的程序。
11.10(k)	对系统文档使用适当的控制，包括： (1)对文档的分发、访问和使用进行适当控制，以便进行系统操作和维护。	应用程序开发人员负责建立适当的程序。

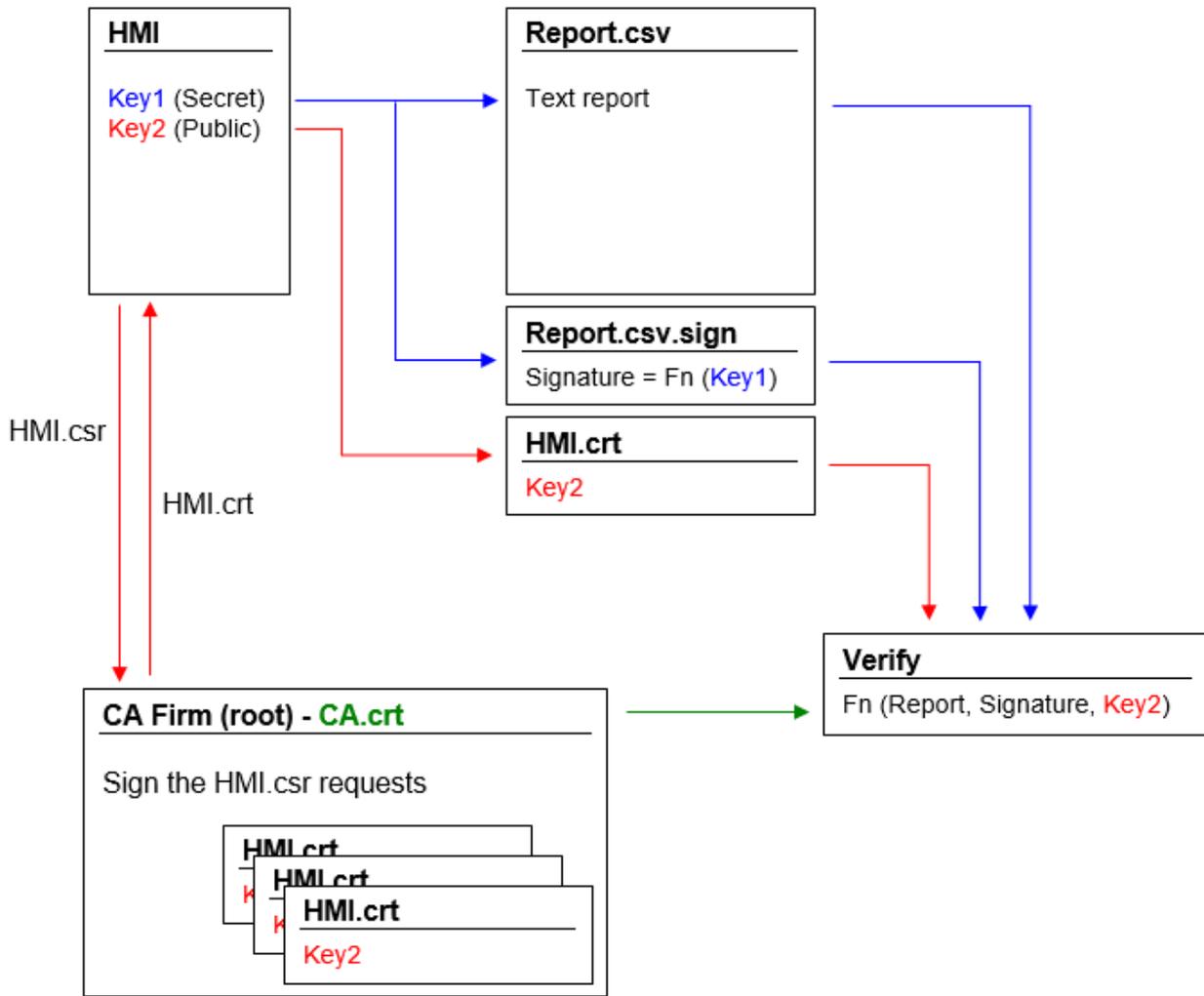
章节	说明	LRH SW 合规水平 (v2.8)
	(2) 修改和更改控制程序, 以维护记录系统文档时序开发和修改的日志。	
11.30	适当时, 使用开放式系统创建、修改、维护或传输电子记录的人员应从创建到收到记录时采用旨在确保电子记录的真实性和完整性的程序和控制措施。这类程序和控制措施应包括 11.10 中指明的程序和控制措施以及文档加密等其他措施, 并使用适当的数字签名标准以在必要时确保记录的真实性和完整性。	LRH SW 设计用于在密闭式系统中操作。
11.50(a)	已签名电子记录应包含与签名有关的信息, 其中应明确说明下列所有内容: (1) 签名人的打印名称; (2) 执行签名时的日期和时间; 和 (3) 与签名相关的含义(如审查、批准、责任或身份)。	所有记录将被添加到具有已登录用户时间戳和用户 ID 的日志中。 引用: <ul style="list-style-type: none"> • "导出日志到 . csv 文件" 在本页 335 • "表格日志控件" 在本页 334
11.50(b)	本部分 (a)(1)、(a)(2) 和 (a)(3) 段中指明的项目, 须接受与电子纪录相同的控制并包括在该电子纪录的任何用户可读取形式(例如电子显示或打印输出)内。	
11.70	在电子记录上执行的电子签名和手写签名应链接到其各自的电子记录, 以确保签名不能被删除、复制或以其他方式传输, 从而以普通手段伪造电子记录。	应用程序开发人员负责避免使用允许导入/导出用户密码的宏。
11.100(a)	每个电子签名应为一入独有, 不得由任何其他人重复使用或重新分配给任何其他人。	系统将确保无法定义具有相同 ID 的两个用户。用户须避免将同一用户 ID 移除并重新分配给其他用户。
11.100(b)	在组织创建、分配、认证或以其他方式批准个人的电子签名或该电子签名的任何元素之前, 该组织应验证个人身份。	用户责任。
11.100(c)	使用电子签名的人员应在使用之前或使用向机构证明, 在 1997 年 8 月 20 日或之后在其系统中使用的电子签名, 旨在成为具有法律约束力的传统手写签名的等效物。 (1) 证书应以书面形式提交区域办事处 (HFC-100)(5600 Fishers Lane, Rockville, MD 20857), 并具有传统的手写签名。 (2) 使用电子签名的人员应代理请求提供额外的证明或证词, 证明某一特定电子签名是具有法律约束力的签名人手写签名的等效物。	用户责任。

章节	说明	LRH SW 合规水平 (v2.8)
11.200(a)	<p>(a) 不基于生物特征电子签名应:</p> <p>(1) 使用至少两个不同的标识组件, 例如标识码和密码。</p> <p>(i) 当个人在单次、连续受控系统访问期间执行一系列签名时, 应使用所有电子签名组件执行第一次签名; 随后的签名应使用至少一个只能由个人执行和设计仅供个人使用的电子签名组件执行。</p> <p>(ii) 当个人执行未在单次、连续受控系统访问期间执行的一个或多个签名时, 应使用所有电子签名组件执行每个签名。</p> <p>(2) 只供其真正所有者使用; 及</p> <p>(3) 进行管理和执行, 以确保在除其真正所有者之外的任何人企图使用个人的电子签名时需要两个或以上个人的合作。</p>	<p>LRH SW 安全功能基于用户名/密码组合。</p> <p>用户必须输入名称和密码以访问系统。可以将关键动作配置为在启动执行之前再次输入密码。</p> <p>引用:</p> <ul style="list-style-type: none"> • "用户管理和密码" 在本页 317 • "电子签名" 在本页 331 <p>每个用户不得泄露自己的密码。首次使用时, 可强制重新定义由管理器定义用于首次访问的密码。</p> <p>引用:</p> <ul style="list-style-type: none"> • "配置用户" 在本页 324
11.200(b)	基于生物特征电子签名应可确保除其真正所有者之外的任何人不得使用这些签名。	LRH SW 不支持生物特征。
11.300(a)	保持每个组合标识码和密码的唯一性, 这样没有两个人具有相同的标识码和密码组合。	无法定义具有相同用户 ID 的两个用户
11.300(b)	确保定期检查、召回或修订识别代码和密码发布(例如, 包括密码使用期等事件)。	<p>系统可以配置为强制每个用户在可配置天数后定义新密码和不同的密码</p> <p>引用:</p> <ul style="list-style-type: none"> • "配置用户" 在本页 324
11.300(c)	遵循损失管理程序, 以电子方式解除对丢失、被盗、缺失或其他可能受到损害的标记、卡片和其他具有或生成标识码或密码信息的设备的授权, 并使用适当、严格的控制措施进行临时或永久替换。	<p>用户可以随时更改其密码。管理员可以重新定义每个用户的密码, 并强制他们在第一次登录时重新定义。</p> <p>引用:</p> <ul style="list-style-type: none"> • "用户管理动作" 在本页 208 • "配置用户" 在本页 324
11.300(d)	使用交易保障措施, 防止未经授权使用密码和(或)标识码, 并在适当时立即、紧急地检测到任何未经授权使用密码的企图并向系统安全单位和组织管理部门报告。	记录尝试失败会被记录到日志中。
11.300(e)	对具有或生成标识码或密码信息的设备(如标记或卡片)进行初始和定期测试, 以确保它们正常工作, 并且没有以未经授权的方式进行更改。	用户确保采取适当的措施。

X.509 证书

为确保由 HMI 设备生成的报告的真实性和完整性，LRH SW HMI Runtime 可以生成带有签名文件的报告，以验证生成报告的真实性和完整性。

LRH SW HMI Runtime 使用非对称加密密钥对文件进行签名和 x.509 标准管理公共密钥证书。下图显示架构。



公共密钥可由保证其真实性的证书颁发机构 (CA) 进行签名。

workflow

- 每个 HMI 设备包含两个密钥：
 - 密钥 1 为私钥，用于对由 HMI 设备生成的报告进行签名。该密钥安全存储在 HMI 设备中。
 - 密钥 2 为公共密钥，任何人都可以使用该密钥验证由 HMI 设备生成的报告的真实性和完整性。
- 宏“保存事件存档”或“打印图形报表”可用于生成签名报告(请参阅“保存事件存档”在本页 199 或“打印图形报表”在本页 184 了解更多详细信息)
- 对于 .csv 文件，您可以使用公钥和已签名文件来验证报告的真实性和完整性，并且未被篡改。(请参阅“已签名 CSV 文件”对页)
- 对于 .pdf 文件，您可以使用 PDF 阅读器来验证报告的真实性和完整性，并且未被篡改。(请参阅“已签名 PDF 文件”在本页 313)

内部 x.509 证书文件

每个 HMI 设备已经具有一个自签名证书。您可以自由使用该证书，要求证书颁发机构对其进行签名，使用您喜欢的信息创建一个新证书，或者上载和使用自己的证书。所有操作均可从设备的“系统设置”进行(请参阅“系统设置”在本页 527 中的“x.509 证书”部分)。



请注意，您无法从 HMI 设备检索私钥。但是，您可以提供包含私钥和公钥的证书。

使用自签名证书

要使用自签名证书，您无需执行任何操作。只需使用生成签名报告的宏即可。即使将通过宏提供证书，也可以使用“系统设置”来检索证书的副本(以确保证书的独创性)。

使用从证书颁发机构签名的证书

要使用来自证书颁发机构的已签名 HMI 证书，必须从“系统设置”面板下载证书签名申请文件。发送并要求证书颁发机构对证书进行签名(通常为付费操作)，然后将已签名证书上传到 HMI 设备。



检索“证书签名请求”文件并发送给证书颁发机构后，请确保不要重新生成新证书，否则与发送给证书颁发机构的证书关联的内部私钥将丢失。

使用您自己的证书

如果您拥有自己的证书并且想要进行使用，则可以从“系统设置”面板将其上传到 HMI 设备中。请注意，您必须同时提供私钥和公钥。



当证书包含私钥时，当前的私钥将替换为证书中找到的密钥，并且此操作无法恢复。

具有公钥和私钥的证书示例(证书编码为 base64)。

```

ssl-certificate.crt x
1 -----BEGIN CERTIFICATE-----
2 MIIDBDCCAewCCQDeBYW7PYwJsDANBqkqhkiG9w0BAQsFADBEMQswCQYDVQQGEwJJ
3 VDEPMA0GA1UEBwwGVmVyb25hMRMwEQYDVQQKDApUZXRhbnR0T2ZmaWNlMQ8wDQYDVQQD
4 DAZITUktMDQwHhcNMTcwNjI2MDgwOTQ1WhcNMTgwNjI2MDgwOTQ1WjBEMQswCQYD
5 VQOGZwJjVDEPMA0GA1UEBwwGVmVyb25hMRMwEQYDVQQKDApUZXRhbnR0T2ZmaWNlMQ8w
6 DQYDVQQDDAZITUktMDQwGggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCdC
7 Nlp2kswcbLh4IxS6eeCgQ4EAUHCrpaZ5YPfQ/un9/s0tejaa3S13Pcqv3JqddJM8
8 mJE2aPF/+HhAEhtC+rv57TbgullUQJdoQpfoGChofpULforXZt2BfdWNx67plNoa
9 YM3E1aNtAKIW2o6S9HGEvlkf09XFLGkFgeMgC59+SejggucNT0m99m6fNa5910I7
10 UDJFINKC3bxtONj+WiL/iEZYkHXacaN9q06fx+2NfmiSsXGPNmSys5mocqo89tMa
11 TjyeF7jYpDccCpJ9pY4xRjRpcIkDCM7PabVoG/ascSMUUEXPE2R0W4UJ6bPAyGd6
12 QLKCCq0BUi6/eUj0pnanAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAMLfIEXQOEjS
13 OpwVxzNxXmL/A6PLU5BK1hVYhb7ofb2Z37zN69vCn8ESglAFYK7QhkhJu3zAD+jH
14 fYBVKVdxfd3HS8EmcDWxpC6F21fgqsSqepMRTbKbsSa053a7JsXtwnHVNfG6EBZV
15 8tqS1Gc4RwtJeVZJelUdmWSBD4Fc7asFeBCKqLrHJinz7buj3I4fLcyscTaMTBI9
16 fsE7poEpWvKc7NWtKYZglGG3AG6xONu3sEahcJ5k+UVdh/QQdAiCt3vG+JJ/owYU
17 sd30WI24pNzG/GUH9MbJyvI4ftA8IvEhGxHvi3xt7s1JnvYQDaghOEDhdtGvilOr
18 nJZ2FZOBCEI=
19 -----END CERTIFICATE-----
20 -----BEGIN RSA PRIVATE KEY-----
21 MIIIEpAIBAAKCAQEA3D2adpLMHGy4eCMUunngoEOBAFBwkaWmeWD30P7p/f7NLXo2
22 mt0otz3Kr9yanXSTPjIRGwJxf/h4QBIbQvq7+e024LtZVECXaEKX6BgoaH6VC36K
23 l2bdgX3Vjceud6TaGmDNxJWjBQCifTqOkvRxxL5ZH9PVxSxpBYHjIAuffkno4ILg
24 jU9Jvf2unzWuf2dCO1AyRSD2At28bTjY/loi/4hGWJB12nGjfatOn8ftjX5okrFx
25 j55ksrOZqHKqPPbTgk48nhe42KQ3HAqSfaWOMUY0aXCJAWjOz2mlaBv2rHEjFFO1
26 zxNkdFuFCemzMoA+kCygqgtAVIuv3lI9KZ2pwIDAQABAoIBAGnamsuqrwDu5hGh
27 02H8GHUpvd/3ytIISujHyvgkwTf+FoTI32y9uMe0pUy5/3y2v9v9/qm3P3djafJq
28 gb5Fprxx4dJPKJZaYi2U7U585lesmVqoHneCk/GeGlyH4zWlwo2xgNgBkkhgaIoR
29 zz0m0bachVz+SCD6wUJpbMOW0FBw54oPL0XS/gD+76S9ET7xmQZAS5xV/w8Khnt
30 PtjPFT58GKqVIC9cMrrBrkuQQPrNrDaJMPsQDxrFp7POQm4+GivrUJ0FA9Vt46
31 C5QhXqVps/BODo3mjeOcj2b/FqsvG7WCc5PWOAcCqStmDxl+DQZOIVFSTrE4kdpq
32 mNn/80kCgYEA88Xfmqg0ta83lpe9b6U0BaLvvs1gxgXmCmkyvK7Ru+iKyPUMzxB+
33 BjGWeeiZuigmIhXfFu3eBs5xOgDrUxf9j55sJAFamljG4LTyun378RnOdA87ff1q
34 rpF4oPKVfTrfXXz2keIq0eX2tD6Lsn3+MJWYqpefovxyJA3kPgcGv0CgYEA50H0
35 HQififZ22nApGpF/jJpU7hBLC45cSXvE2MX2I3rd3ptGwzKR0/l2ks1bvQutqRln
36 slyEF+c9LCz6g7FYhJoewChLqCVfeZ9GxBzHeJ1oxZwmxDX18L4vmEDphwlcV8b3
37 ExHqU1MGuINHGelPIR1LKEsbTQU+OVHuNv443MCgYEA7rMKYh1lC6bYCsjowSMG
38 TqKembX84cqyl+zstp+EVbi99Usm0Lc4f/4cd6EQrp1Twbqi6YPgDdAmRQLTalp
39 e3FIOFVub4aQr0XgDEcC5bI8W57yxUrZJLjjYs5HHQoB4Dw5m0TOMFnS+enoxs3i
40 kly3Nowjz+fRCYFWN8kLVE0CgYEA43CLLk7ZcW9XKa2cNB00PE1g8A4YMJJfk2nl
41 zKjNj1F9ujyO2NV4RYOsI+RSsFe3ARDJcS6xP200Tc8ixrh57VhCnAxFdGblQpFy
42 oNqJGkf9zjPoMJsqykjSOHTG+CctqaqmPxxkkLScbIW4PPSn/U6KDPNHpVNOuQeO
43 hXHak58CgYBLW1719vgYhUiSWc9Gd3mCSxpAb6y8RcyTgqF76K8v4MallPqFkEtD
44 0BaFt1A+PtMLk2ODTRH4XU18oc9eV+7VDFkPJ8T0A2VwjzjMgNAd+vKlM4nOEBTt
45 UhegY0k8yLxS12vuYiVnHvKBIOF/G2ckwrxj09KVE+SA45Ex0Px5qQ==
46 -----END RSA PRIVATE KEY-----

```



您可以在每个 HMI 设备中导入相同的证书文件，使所有 MHI 设备具有唯一的公共证书文件。

已签名 CSV 文件

可以使用 HMI 设备中随附的 x.509 证书对使用 SaveEventArchive 宏以 CSV 格式生成的报告进行签名。签名可确保签名后没有人篡改文档的内容。

另请参阅：

- SaveEventArchive 参数 ("保存事件存档" 在本页 199)
- 如何向 Linux 设备提供 x.509 证书 ("X.509 证书" 在本页 542)
- 如何向 WinCE 设备提供 x.509 证书 (第 1 页上的“系统设置”)

必要时使用 Signed = True, 除 [ReportName].csv 外, SaveEventArchive 宏还将生成其他两个文件：

- [ReportName].csv.sign
- ssl-[CertificateName].crt

其中, [ReportName].csv.sign 是报告的签名, 而 ssl-[CertificateName].crt 是 HMI 设备的 x.509 证书的副本。请注意, 从 HMI 设备的系统设置中也可以检索 HMI 设备的证书。

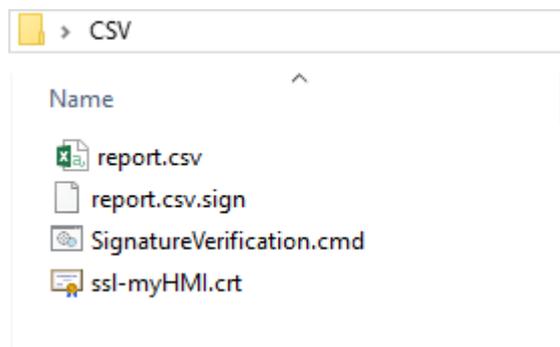
如何使用公共 OpenSSL 库验证报告的签名

要验证报告内容未被他人篡改, 您需要

- 确保 ssl-[CertificateName].crt 来自 HMI 设备
- 使用工具来验证签名 (例如, OpenSSL-Win32)

参考：<https://www.openssl.org/>

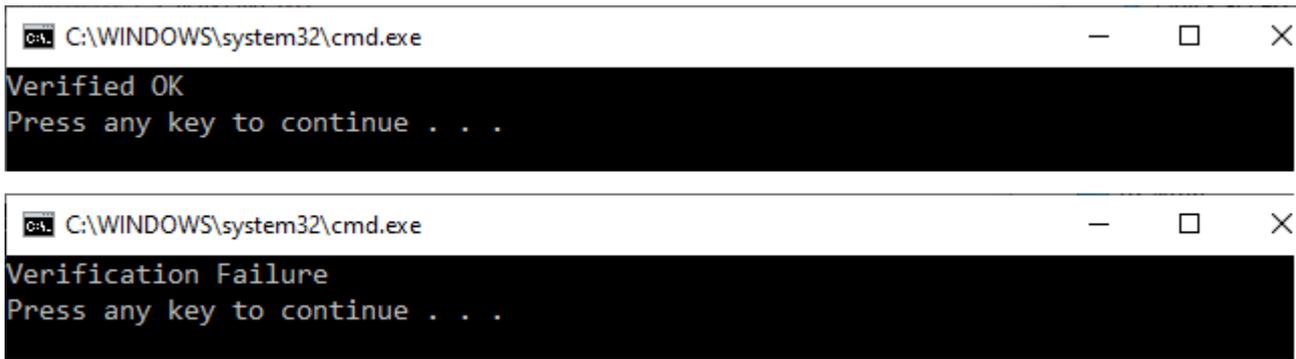
要验证从 HMI 设备生成的.csv报告是否未被篡改, 您可以安装公共 OpenSSL 库, 将从宏生成的所有文件复制到同一文件夹中, 并使用以下批处理文件



文件：*SignatureVerification.cmd*

```
@echo off set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe" set
FileToCheck=Report.csv set hmiCertificate=ssl-myHMI.crt rem Extract public key from
the certificate %OpenSSL% x509 -in %hmiCertificate% -pubkey -noout > publicKey.pem rem
Verify Signature %OpenSSL% dgst -sha256 -verify publicKey.pem -signature
%FileToCheck%.sign %FileToCheck% rem Remove public key del publicKey.pem pause
```

下图显示了批处理文件的可能输出



在 Linux 设备上需要 **BSP v1.0.239** 或更高版本

已签名 PDF 文件

可以使用 HMI 设备中随附的 x.509 证书对使用 PrintGraphicReport 宏以 PDF 格式生成的报告进行签名。签名可确保签名后没有人篡改文档的内容。

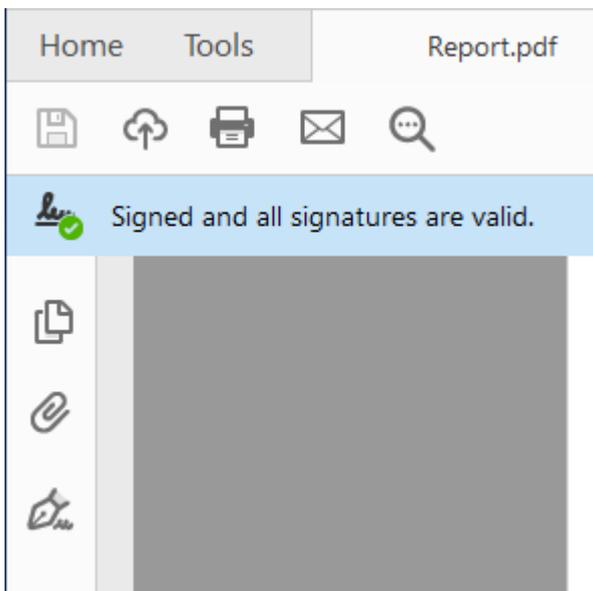
另请参阅：

- PrintGraphicReport 参数(“打印图形报表”在本页 184)
- 如何向 Linux 设备提供 x.509 证书(“X.509 证书”在本页 542)
- 如何向 WinCE 设备提供 x.509 证书(第 1 页上的“系统设置”)

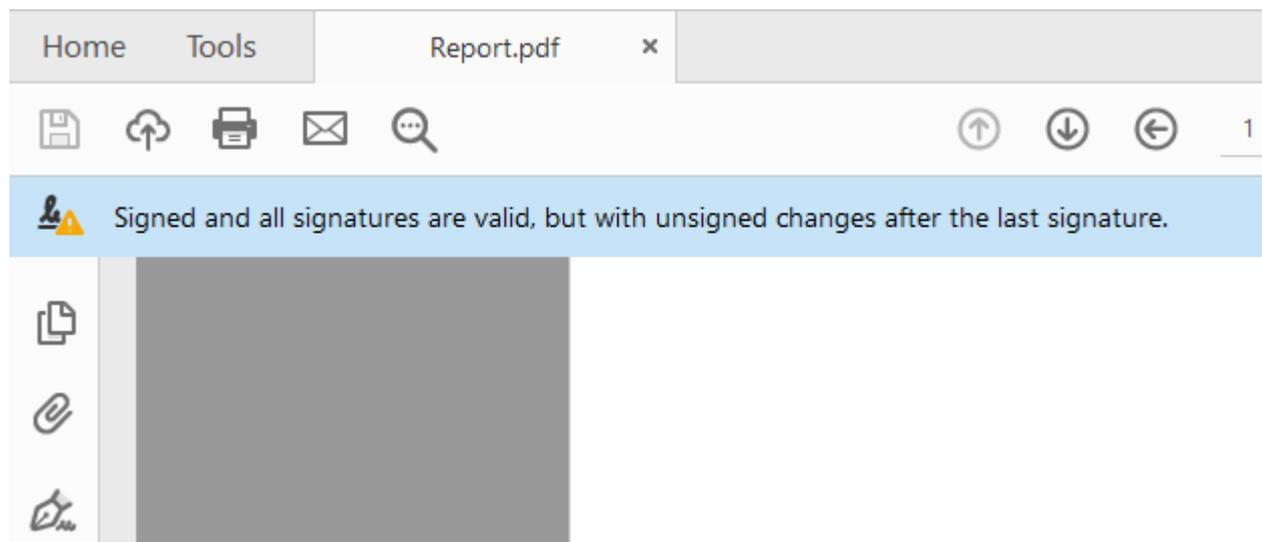
当您打开文件时，PDF 阅读器尝试确定签名是否有效，然后查看用于对文档签名的证书。

由证书颁发机构签名的 x.509 证书

如果您已将有效的 x.509 证书上传到操作员面板，该证书由证书颁发机构签名，则在打开生成的 PDF 文件时，您将获得一条提醒消息显示该文档有效。



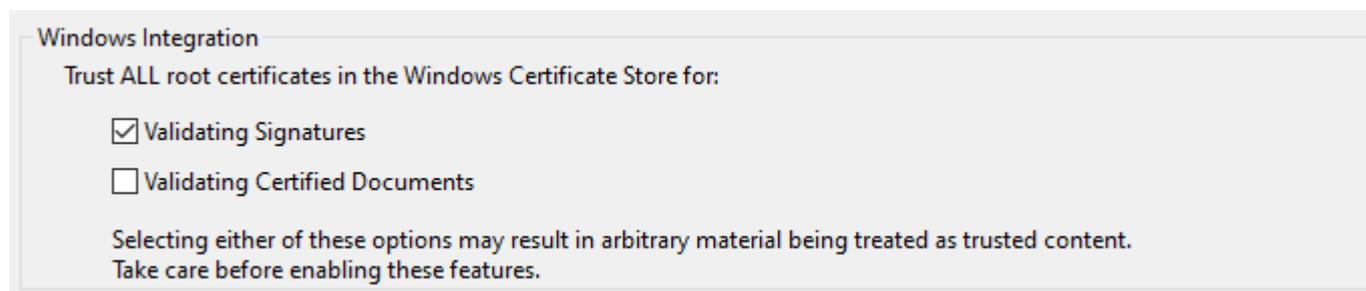
如果文档已被修改，将以不同的消息突出显示。



证书信任和真实性

签名证书的可信任度取决于证书的颁发者。如果您告诉 PDF 阅读器信任该特定证书的颁发者，则它将信任该证书。默认情况下，Adobe Reader 仅信任 Adobe 或其合作伙伴之一颁发的证书。这意味着如果证书不是由这些机构之一颁发的，将显示警告。Microsoft Windows 还使用证书来验证软件供应商和内容提供商。您可以将 Adobe Reader 配置为信任 Adobe 伙伴之外的发行者。

如果要启用 PDF 阅读器甚至使用 Microsoft Windows 证书，请检查 PDF 阅读器的内部偏好

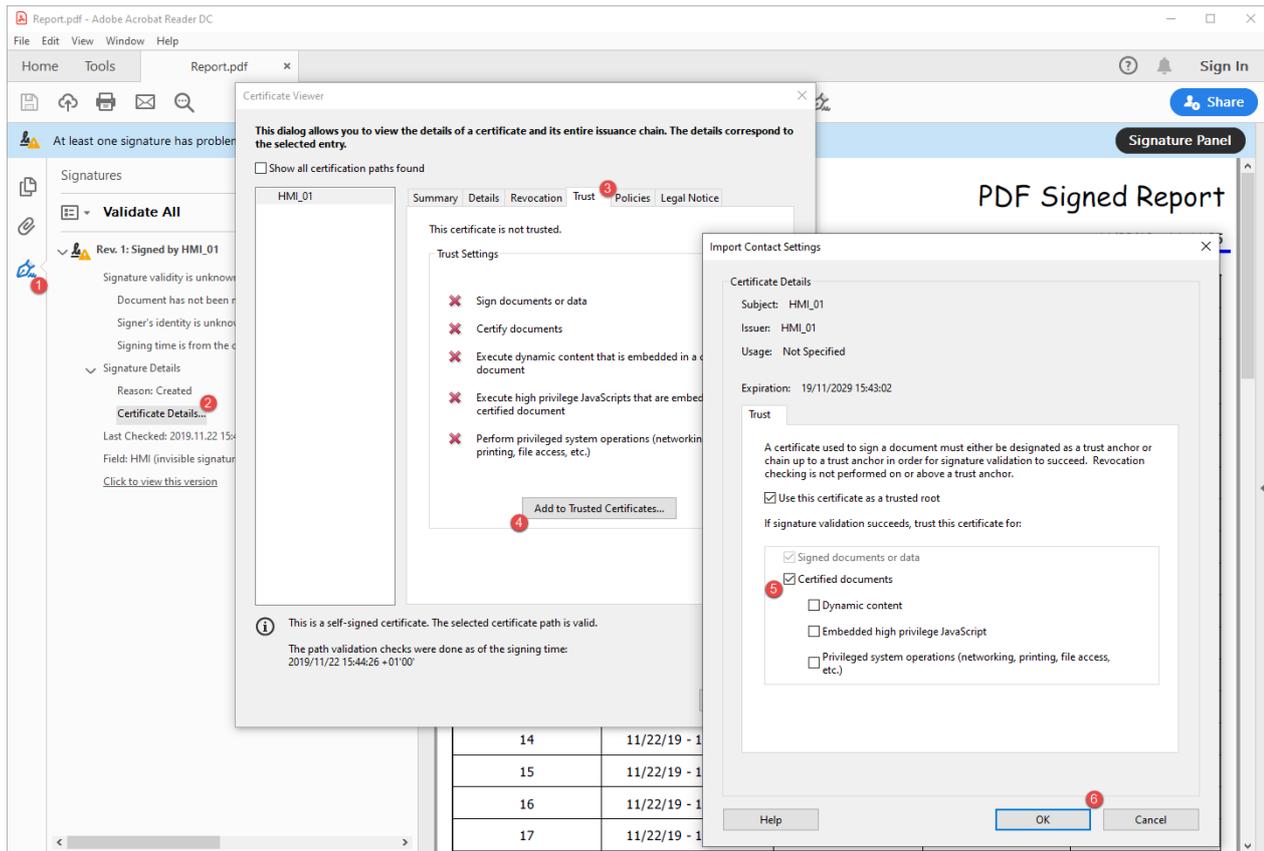


X.509 自签名证书

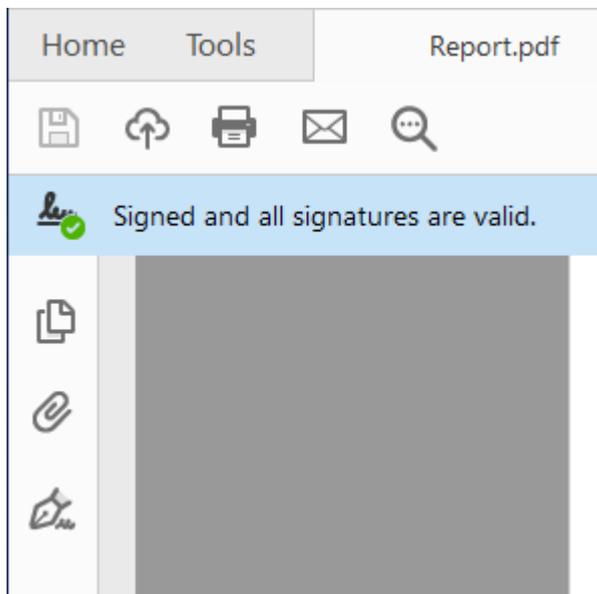
自签名证书是未经由证书颁发机构 (CA) 签名的证书。

这意味着 PDF 阅读器可以确认文件已签名并且未被篡改，但是无法确认签名 (别名证书) 是真实的。用户必须注意核实证书是否真实 (例如，确保文件确实是由面板制作的)，并向 PDF 阅读器确认文件中包含的证书有效且可视为对下次报告是有效的。

手动确认证书真实性的步骤：



现在，如果您关闭并重新打开 PDF 文档，您将获得有效的签名。此外，因为证书真实性的信息已存储在 PDF 阅读器的设置中，所以从同一 HMI 设备产生的所有其他文档也将显示正确的签名。



在 Linux 设备上需要 BSP v1.0.507 或更高版本

符合应用程序

CFR11 符合应用程序的研发建议

用户管理宏

可以由任何用户使用的用户管理宏

- 登录
- 注销
- 切换用户
- 更改密码

只能由管理员使用的用户管理宏

- 重设密码
- 添加用户
- 编辑用户
- 导出用户，

不得在 CFR 21 第 11 部分合规性应用程序中使用的已弃用的宏

- 导入用户
- 删除用户
- 删除 UM 动态文件，

26 用户管理和密码

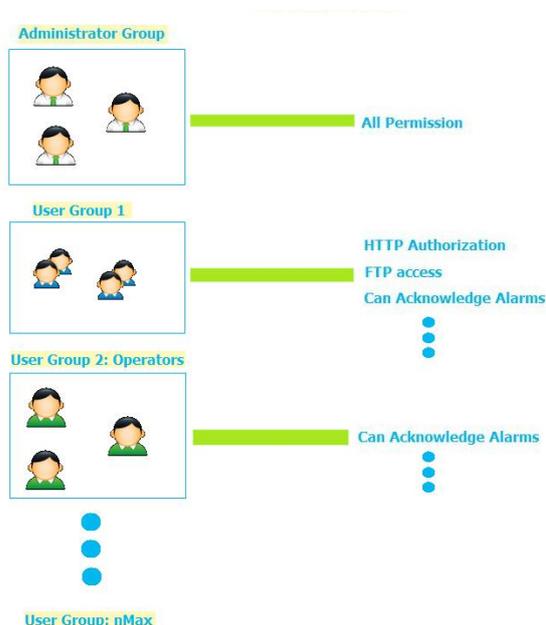
您可以通过设置用户、用户组和对每组指定授权限制各种控件可操作的访问。

每一个用户必须是且仅是一个群组的成员。每一个组有特定的授权和许可。

授权和许可分为两种：

- 控件许可：隐藏、只读、完全访问
- 动作许可：允许或不允许。

通过组织许可和群组，您可以限定一个项目的安全性。



启用/停用安全管理	318
配置组和授权	318
修改访问权	319
从页面视图指定控件许可	324
配置用户	324
默认用户	326
运行时管理用户	326
强制远程登录	327

启用/停用安全管理

路径: 项目视图 > 右键单击安全性 > 启用

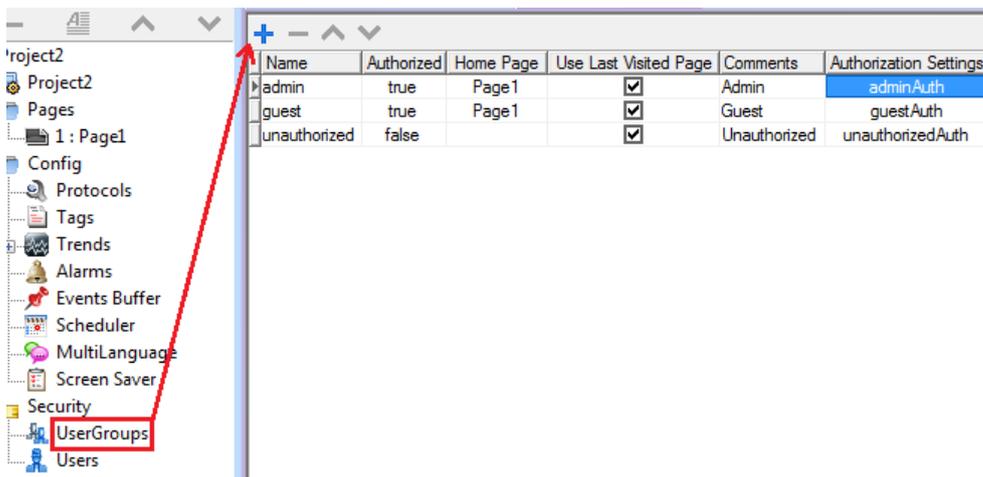
挂锁符号指示该功能是否被启用或已停用。



重要: 安全设置只有在安全功能启用时有效。

配置组和授权

路径: 项目视图 > 安全性 > 双击用户组



默认提供三个预定义组 (**admin**、**guest** 和 **unauthorized**): 这些不能删除或重命名。但是, 您可以修改授权和其他设置。

添加一个用户组

单击 **+** 添加用户组。

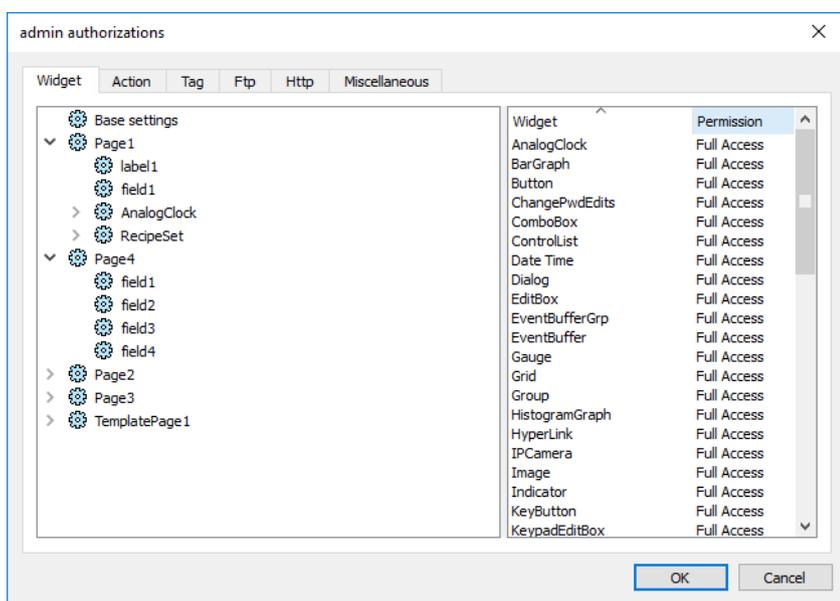
参数	说明
名称	用户组的名称
已授权	授权许可
主页	当属于该组的用户登录时, 显示的页面
使用上次访问的页面	当被选择时, 上一个用户最近显示的页面将会在属于改组的用户登录时被显示

参数	说明
注释	对改组的任何注释或描述
授权设置	打开管理员授权对话框设置存取许可。 请参阅 "修改访问权" 向下了解详细信息。

修改访问权

路径: [项目视图](#) > [安全性](#) > [双击用户组](#) > [授权设置列](#)

点击按钮: 显示一个带有控件和动作列表的对话框。您可以为列表中的每一项修改访问权。



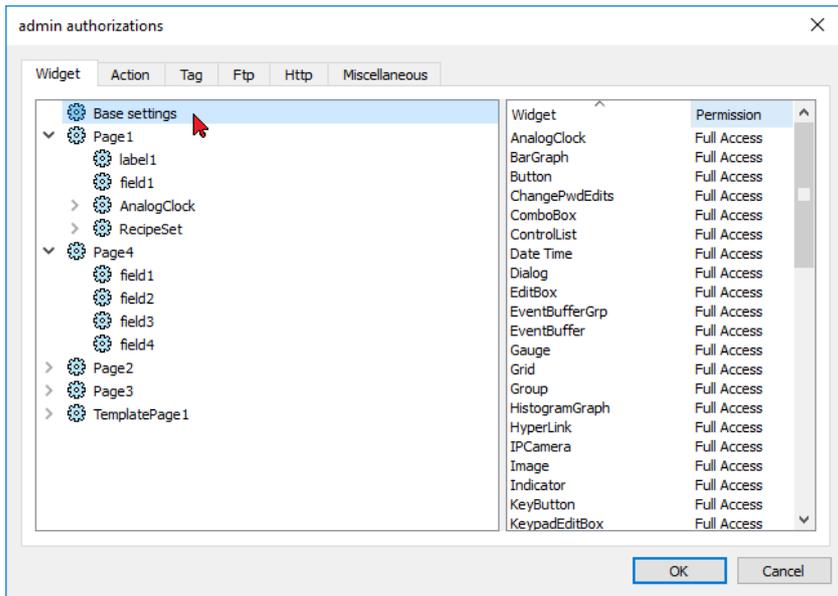
控件权限

在**控件**选项卡中,可以为项目中使用的所有控件定义项目级、页面级或控件级的控件访问选项。较低级权限(例如控件级)覆盖较高级(即页面和项目级)。

使用**基本设置**设置项目级默认权限。

可用设置:

- **完全访问权**, 启用对控件的读/写访问
- **只读**, 启用对控件的只读访问
- **隐藏**, 对所选组隐藏控件



更改控件权限

要更改项目页面中单个控件的访问权，请在右侧窗格中导航至页面中的该控件，并自定义访问选项。否则所有控件采用项目或页面级权限。

例如，如果控件的页面权限在项目级别设为**只读**，则所有相同控件的权限均为**只读**。从树结构中选择页面内的控件时，权限实际设为**使用基本设置**。您可以在此页面中更改此设置和仅修改此控件的访问权。

访问优先级

控件权限考虑以下优先级：

权限级	优先级
项目级 - 基本设置	低
页面级	中
控件级	高

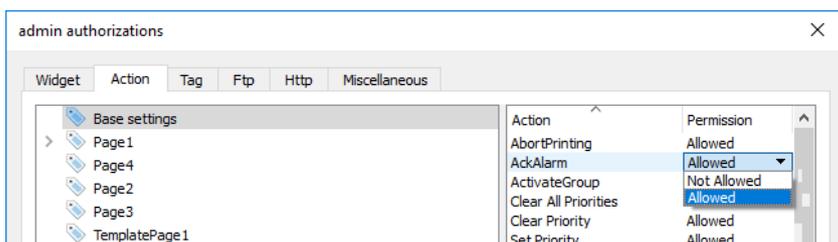
这允许您直接从页面视图指定动作或控件例外。

例如，如果您在项目级将控件权限设置为只读，在页面级设置为完全访问，则页面级设置将优先。

可以直接从项目页面修改访问权。请参阅"[从页面视图指定控件许可](#)"在本页324了解详细信息。

动作权限

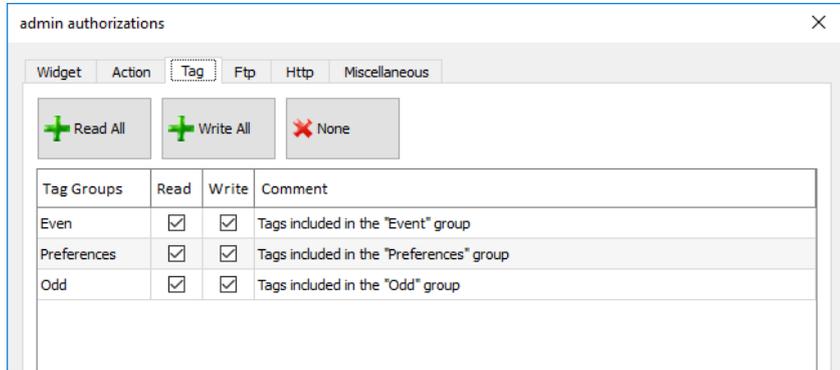
在**动作**选项卡中，可以定义项目级、页面级或控件级的动作授权。动作可以是**允许或不允许**。



可以直接从项目页面修改动作权限。请参阅"从页面视图指定控件许可"在本页 324 了解详细信息。

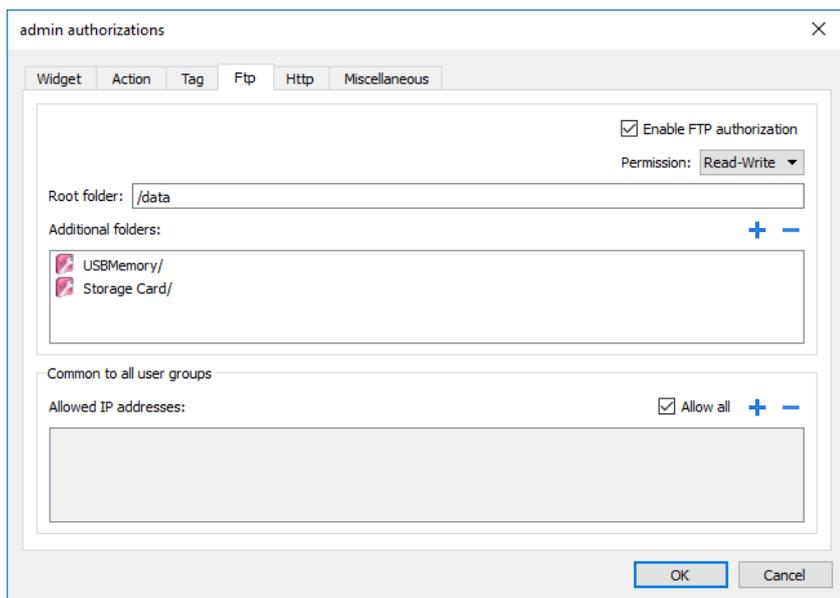
标签权限

您可以定义每组标签的读/写访问权



FTP 授权

在 **FTP** 选项卡中，可以为 FTP 服务器设置特定授权。



元素	说明
启用 FTP 授权	为特定组启用 FTP 功能
权限	权限类型： <ul style="list-style-type: none"> • 只读 • 读写
Root 文件夹	用于 FTP 访问 root 的文件夹。这是相对路径。

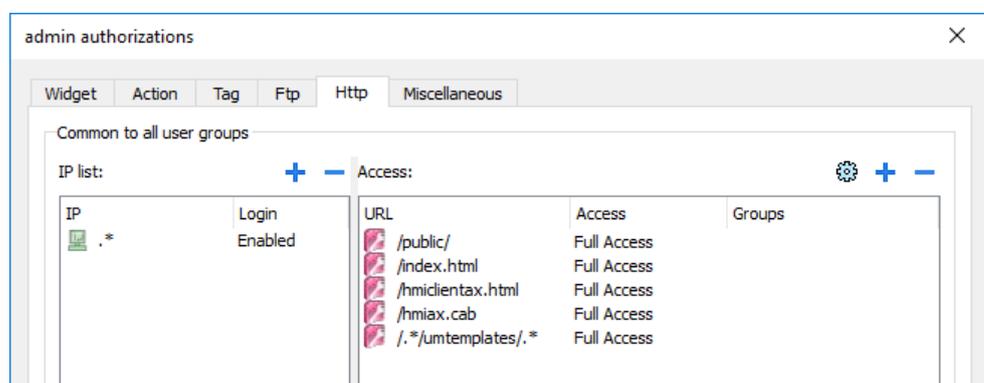
元素	说明
其他文件夹	用作 FTP 访问 root 的其他文件夹(例如在 USB 驱动器或 SD 上)
允许的 IP 地址	FTP 连接可接受的 IP 地址列表。  此设置对所有用户组通用。

HTTP 授权

在 **HTTP** 选项卡中, 设置对集成在 LRH SW HMI Runtime 中的 Web 服务器的 HTTP 访问限制。

通配符可用于确定 IP 地址的范围。

例如, 下面的两个规则规定只能通过本地网络上范围为 192.168.*.* 的所有 IP 地址访问 HMI 设备, 只有 IP 地址 192.168.1.20 无需输入登录名即可访问设备。



元素	说明
IP 列表	授权访问 HTTP 服务器的 IP 地址。  默认情况下, 任何 IP 地址 (IP=.*, Login=Enabled) 均需登录。
登录	禁用时, 无需用户名和密码。
访问限制	限制访问的资源列表

这些设置的效果取决于是否选中 **强制远程登录** 选项。请参阅 "[强制远程登录](#)" 在本页 327 了解详细信息。

强制远程登录	工作空间默认访问	访问限制
-	完全	-
禁用	完全	可用于阻止访问某些文件/文件夹或要求授权
启用	无访问权	可用于打开文件/文件夹访问权



重要: 此设置对所有用户组通用。

添加 HTTP 配置

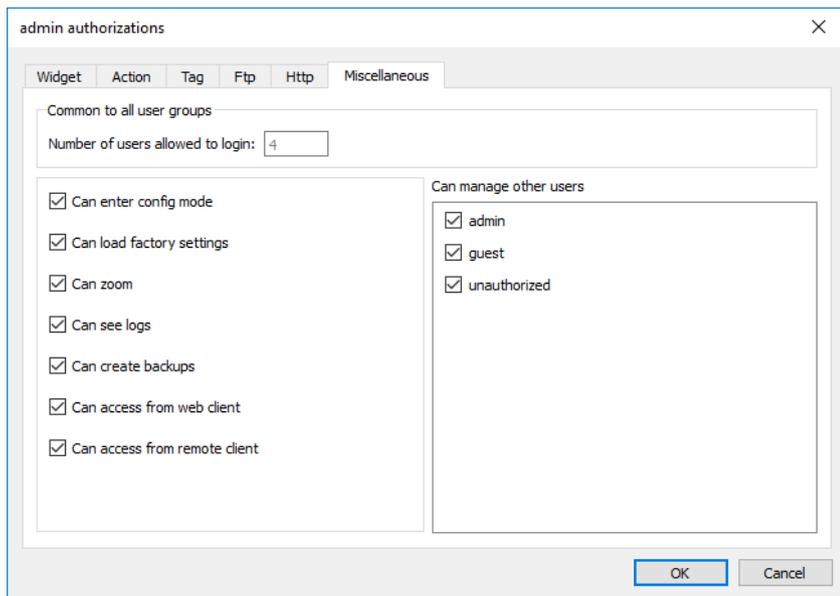
要添加和配置新访问，请单击 **+**: **访问限制** 对话框显示。

要恢复默认配置，请单击 **设置默认访问限制** 图标。默认配置允许访问以下内容：

- PUBLIC 文件夹和 Index.html

其他设置

在 **其他** 选项卡中，可以定义各种授权设置。

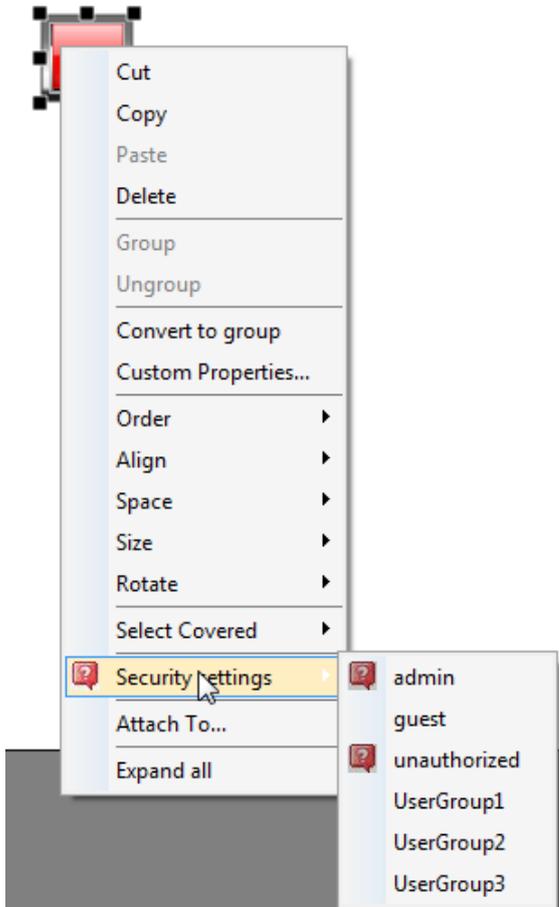


选项	说明
可以进入配置模式	启用从 Runtime 切换为配置模式。通常用于维护。
可以加载出厂设置	恢复出厂设置。
可以缩放	允许运行时在上下文菜单中放大/缩小
可以查看日志	允许用户在运行时查看日志
可以创建备份	允许用户备份项目。
可从 Web 客户端访问	允许从 Web 客户端联接
可从远程客户端访问	允许从 LRH SW Client 联接
可以管理其他用户	在运行时给予超级用户权限，以管理所选组。允许添加、删除和修改用户权限。
允许登录用户数	可同时连接到 HMI Runtime 的最大用户数量。  此设置对所有用户组通用。

从页面视图指定控件许可

您可以直接从项目页面在单一控件上对不同的用户组指定不同的安全级别。

1. 右键单击控件，选择**安全设置**。
2. 选择组：改组授权的对话框被显示。
3. 设置存取该控件的安全属性。



请参阅["修改访问权"](#)在本页319了解详细信息。

配置用户

路径：[项目视图](#) > [安全性](#) > [双击用户](#)

在“用户”编辑器中，单击 **+** 添加用户：表中将添加一行。

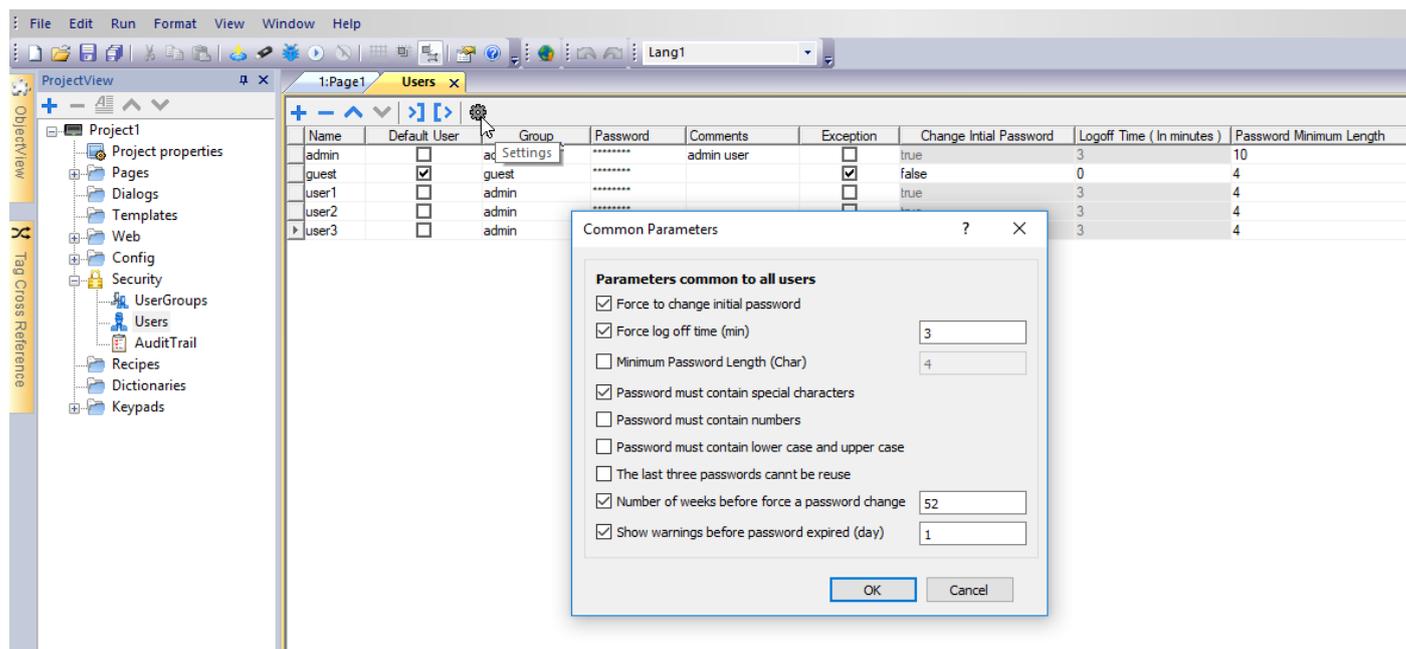
Name	Default User	Group	Password	Comments	Exception	Change Initial Password	Logoff Time (In minutes)	Password Minimum Length
admin	<input type="checkbox"/>	admin	*****	admin user	<input type="checkbox"/>	false	0	4
guest	<input checked="" type="checkbox"/>	guest	*****		<input checked="" type="checkbox"/>	false	0	4
user1	<input type="checkbox"/>	admin	*****		<input type="checkbox"/>	false	0	4
user2	<input type="checkbox"/>	admin	*****		<input type="checkbox"/>	false	0	4
user3	<input type="checkbox"/>	admin	*****		<input type="checkbox"/>	false	0	4

参数	说明
名称	用户名
默认用户	当系统启动时或另一个用户退出之后，该用户自动登录。只可以设置一个默认用户
编组	用户组
密码	用户密码。请注意，出于安全原因，切勿显示密码  密码已加密，即使对专业技术人员而言也不能检索
注释	进一步的用户描述
异常	允许更改从“用户设置”参数中强制设置的值
更改初始密码	该用户在首次登录时被强制更改密码
注销时间(分钟)	非活跃时间之后该用户注销。设置为 0 以停用
密码最小长度	密码最小长度
必须包含特殊字符	密码必须包含至少一个特殊字符
必须包含数字	密码必须包含至少一个数字
必须包含小写和大写字母	密码必须包含小写和大写字母
密码不能重复使用	新密码必须不同于最后 3 个使用的密码
密码使用期(周数)	在强制密码更改之前的周数(1/52 周)
警告(天数)	在密码过期(1/30 天)之前显示警告消息

用户设置

从“设置”命令中，可以定义将对所有用户通用的参数值。

不强制已选中“异常”标志的用户使用通用参数。



默认用户

您只能在项目中定义一个默认用户。这是系统启动、当前登录用户退出或超时退出时默认用户自动登录。

使用不同用户登录 LRH SW HMI Runtime，使用以下动作之一：

- 切换用户
- 注销

请参阅“用户管理动作”在本页208了解详细信息。

运行时管理用户

如果有默认用户，默认用户在 HMI Runtime 启动时自动登录。如果未配置默认用户，系统要求输入用户名和密码。请参阅“用户管理动作”在本页208，了解用户可执行的动作的详细信息。

移除用户数据

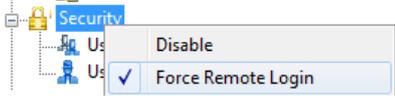
运行时修改的所有用户信息被保存在指定文件中。运行时要移除这些动态文件和所有对用户配置应用的更改，您可以：

- 在 HMIRuntime: 执行删除 UM 动态文件(DeleteUMDynamicFile)动作
- 对于 LRH SW: 请在“下载”对话框中选择**删除动态文件**。

 注：在 Studio 中对用户管理进行任何修改时，需要删除用户管理动态文件，以便应用新的用户管理设置。

强制远程登录

路径: **项目视图** > 右键单击 **安全性** > **强制远程登录**



选中此选项，通过 LRH SW Client 进行远程访问时强制用户登录。如果未选中，远程访问将使用与本地访问相同的保护级别。



重要: 此功能仅在启用用户管理时有效。



警告: 当您有默认用户但同时想保护远程访问时使用此选项。

请参阅"启用/停用安全管理"在本页 318 了解详细信息。

当该功能启用时仍可以访问的文件 / 文件夹有：

- PUBLIC 文件夹和 Index.html。

请参阅"修改访问权"在本页 319 了解 HTTP 访问限制的详细信息。

27 日志

日志是以时间为顺序的记录序列。每一项记录包含已执行的动作和执行动作的用户的信息。

该功能提供事件的时间戳进行过程跟踪和用户识别。

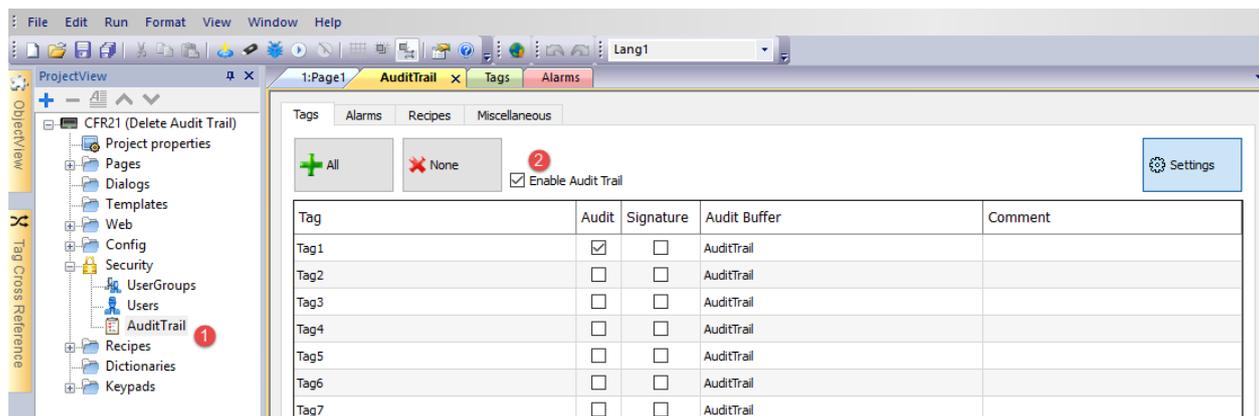
启用/停用日志	330
电子签名	331
表格日志控件	334
导出日志到 .csv 文件	335
配置日志事件	336

启用/停用日志

路径: 项目视图 > 安全性 > 双击日志

可从“启用日志”复选框中启用日志记录

启用时, 对所选资源所作的更改将被记录到具有时间戳、执行操作的用户名和有关已修改资源的一些更多信息(例如标签的新值和之前值)的日志缓冲区



从日志编辑器的主选项卡(标签、报警、配方和其他)中, 可在可用资源的列表视图之间切换。

参数	说明
日志	启用所选资源跟踪
签名	在允许用户修改资源之前, 需要输入用户密码(请参阅 "电子签名"对页 了解更多信息)
日志缓冲区	存储相关日志事件的内部缓冲区(请参阅 "配置日志事件"在本页 336 了解更多信息)
注释	可用于开发人员的注释空间

标签

- 跟踪标签值何时更改。

报警

- 跟踪用户何时确认或重置报警事件

配方

- 跟踪用户何时下载或上传配方

其他资源

- 用户登录详细信息
跟踪用户何时登录、注销或更改密码
- 用户管理动作
跟踪何时添加、移除用户或修改用户属性
- 系统动作
跟踪系统动作(HMI 设备重启、上电、备份、更新、下载、进入“系统设置”、打开“项目管理器”)
- FTP 动作
跟踪 ftpGET、ftpPUT、OpenTextEditor、SaveTextEditor
- 缓冲区动作
跟踪有关报警、日志或趋势缓冲区的转储和删除动作

日志消息宏

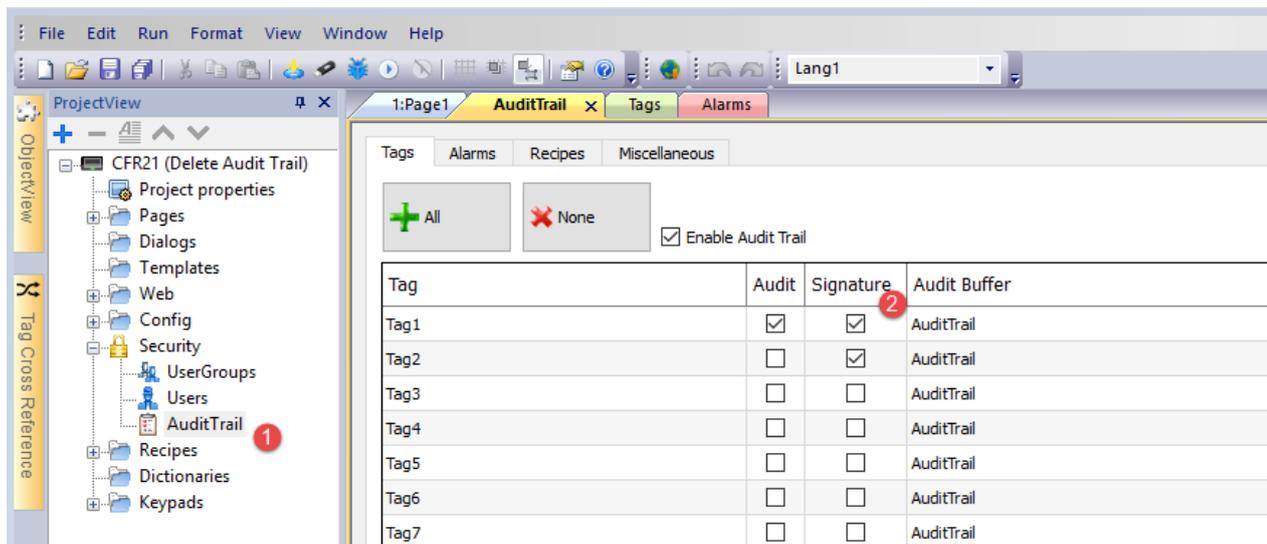
此外，日志消息宏可以定义记录到日志缓冲区的额外事件。

请参阅“[日志消息](#)”在本页 201 了解更多详细信息。

电子签名

对于“日志”编辑器内列出的各资源，可以配置 LRH SW HMI Runtime 以便在更改之前需要密码确认。如果日志已启用，用户可以选择添加将在“跟踪日志”中记录的注释。

路径：[项目视图](#) > [安全性](#) > [双击日志](#)



在允许用户修改资源之前，需要输入用户密码

Confirm your password

Password:

Comment:

This is a short comment that explains why I am doing this change

Ok

Cancel



在接下来的 10 秒内发布的命令将不再需要引入的密码。
有效时间可以在“设置”对话框中进行修改。

Signature	Audit Buffer	Comment
<input checked="" type="checkbox"/>	AuditTrail	
<input checked="" type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	

Settings

The Settings dialog box is open, showing a field for 'Signature password validity (Sec)' with the value '10'. A red arrow points from the 'Settings' button in the top right of the main interface to the dialog box.



表格日志控件

路径: 控件库 > 基本 > 日志表格

显示控件中的日志内容

Audit View

From : 29/03/18 - 14:08:25

Duration : 1 Hour

Refresh

To : 29/03/18 - 15:08:25

Filter on column: Username

#	Timestamp	Username	Operation	Information
1	29/03/18 - 15:07:35	SYSTEM_IDAL	SYSTEM_POWERON	
2	29/03/18 - 15:07:35	admin	LOGIN	1
3	29/03/18 - 15:07:38	admin	WRITE_TAG	Tag1;0;1
4	29/03/18 - 15:08:00	admin	WRITE_TAG	Tag1;1;0
5	29/03/18 - 15:08:03	admin	ACK_ALARM	Alarm1
6	29/03/18 - 15:08:07	admin	RESET_ALARM	Alarm1
7	29/03/18 - 15:08:24	SYSTEM_IDAL	RECIPE_WRITE_TAG	Tag1;0;1

Backward

Forward

按钮:

- 刷新
从内部缓冲区中检索趋势数据并刷新表格视图
- 向后/向前
向前或向后移动按持续时间参数指定的显示窗口

筛选器:

使用组合框选择搜索所在的列, 并选择在右侧输入要搜索的字符串的文本。

参数	说明
日志缓冲区	从中检索事件列表的事件缓冲区(请参阅"配置日志事件"在本页 336)
标题	标题标签
默认持续时间	要显示的时间窗口的初始值
时间规格	时间格式: <ul style="list-style-type: none"> • Local = 显示 HMI 设备的时间值。 • Global = 显示使用 UTC 格式的时间值。
日期格式	选择日期和时间格式

参数	说明
筛选器列表	要在筛选列选择中显示的标签
表格布局	定义滚动条的特点并允许删除表格的标题

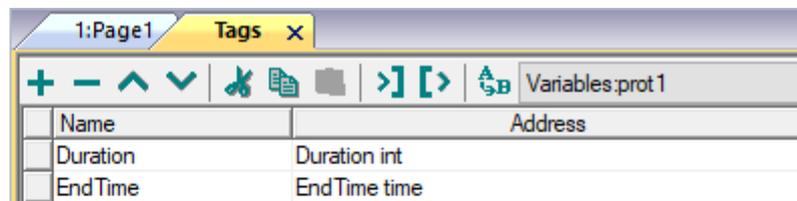
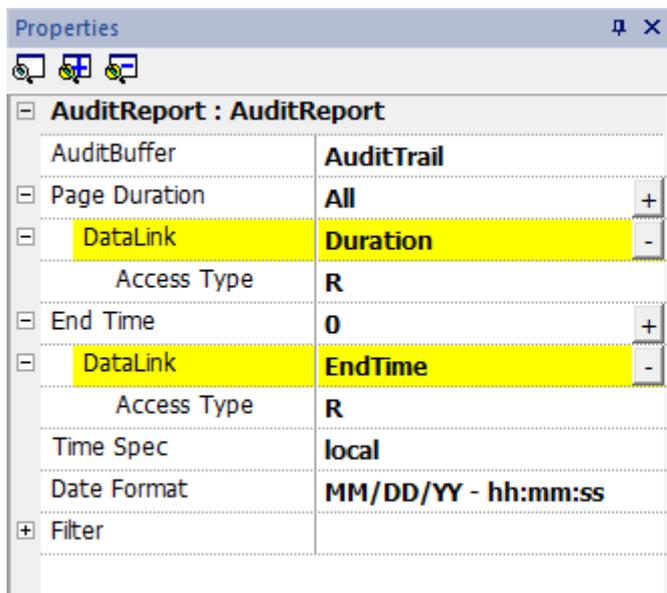
打印日志表格

可从打印报表库查找并使用没有按钮的日志表格控件。可以绘制表格并扩大至整个页面。如果要打印的行数大于一页，则将使用其他页面来打印日志表格。

使用标签功能的添加附件，可在运行时使用标签定义要打印的历史趋势的某些属性：

- 显示时长
- 结束时间

“页面持续时间”和“结束时间”定义要打印的日志缓冲区部分。



导出日志到 .csv 文件

可使用 **保存事件存档** 动作将日志中记录的数据在 csv 文件中导出。请参阅 **“保存事件存档”** 在本页 199 了解详细信息。

文件结构

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Record ID	Date	Time	User ID	Interface	Action	Status	Data				
3	1	27/03/2018	14:22:06	SYSTEM_IDAL	SYSTEM_IDAL	SYSTEM_POWERON	S_OK					
4	2	27/03/2018	14:22:06	admin	LOCAL	LOGIN	S_OK	1				
5	3	27/03/2018	14:22:08	admin	LOCAL	WRITE_TAG	S_OK	Tag1	0	1		
6	4	27/03/2018	14:22:09	admin	LOCAL	WRITE_TAG	S_OK	Tag2	0	1		
7	5	27/03/2018	14:22:26	admin	LOCAL	WRITE_TAG	S_OK	Tag2	1	5	This is a test	
8	6	27/03/2018	14:22:50	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag1	1	1		
9	7	27/03/2018	14:22:50	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag2	5	3		
10	8	27/03/2018	14:22:50	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag3	0	5		
11	9	27/03/2018	14:22:50	admin	LOCAL	DOWNLOAD_RECIPE	S_OK	Recipe0	set-00			
12	10	27/03/2018	14:22:54	admin	LOCAL	ACK_ALARM	S_OK	Alarm1				
13	11	27/03/2018	14:22:58	admin	LOCAL	RESET_ALARM	E_FAIL	Alarm2				
14	12	27/03/2018	14:23:02	admin	LOCAL	DUMP_AUDIT_BUFFER	S_NEEDNOT_NOTIFY	AuditTrail				
15												
16												
17	Record ID	Date	Time	User ID	Interface	Action	Status	Data				
18	13	27/03/2018	14:23:24	admin	LOCAL	DELETE_AUDIT_BUFFER	S_OK	AuditTrail				
19	14	27/03/2018	14:23:26	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag1	1	2		
20	15	27/03/2018	14:23:26	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag2	3	4		
21	16	27/03/2018	14:23:26	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag3	5	6		
22	17	27/03/2018	14:23:26	admin	LOCAL	DOWNLOAD_RECIPE	S_OK	Recipe0	set-01			
23	18	27/03/2018	14:23:27	user1	CGI	LOGIN	S_OK	192.168.49.242				
24	19	27/03/2018	14:23:27	user1	CGI	WRITE_TAG	S_OK	Tag1	6	55		
25	20	27/03/2018	14:24:28	admin	LOCAL	DUMP_AUDIT_BUFFER	S_NEEDNOT_NOTIFY	AuditTrail				
26												

导出的数据文件具有以下内容

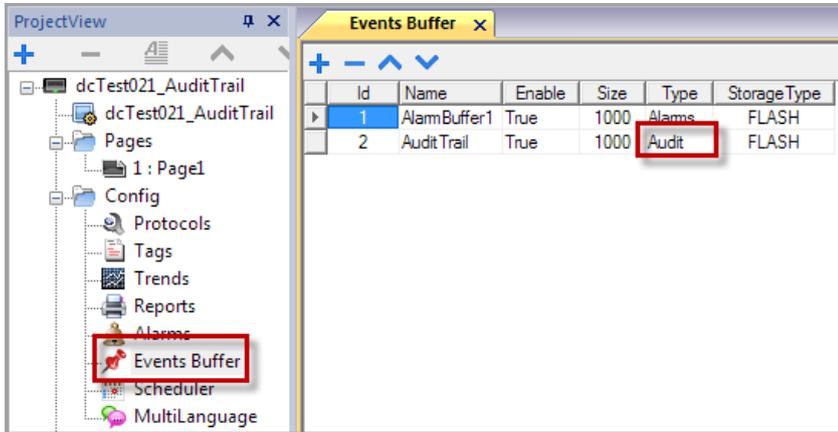
记录 ID	每个记录使用累进数字进行存储，以便轻松识别缺失的记录或确认这些记录未丢失。请注意，删除缓冲区时，累进数字未重置为 0。
日期、时间	事件时间戳。时间可以设置为存储动作的当地时间或世界时间。
用户 ID	执行操作的用户
接口	本地：当动作在 HMI 设备中执行时 CGI：当动作由远程客户端执行时。 SYSTEM_IDAL：当动作从 LRH SW HMI Runtime 应用程序执行时
动作	动作执行。
状态	执行动作的结果 <ul style="list-style-type: none"> • S_OK 动作正确执行 • E_FAIL 动作未执行 • S_NEEDNOT_NOTIFY 动作触发(将非同步执行)
信息	与执行动作相关的额外信息。

配置日志事件

您可以有一个以上的日志记录集。您需要配置指定的事件缓存。

创建事件缓存

路径: 项目视图 > 配置 > 双击事件缓存



1. 在事件缓存编辑器中，单击 **+**：表中将添加一行。
2. 为**类型**选择日志。
3. 配置缓存参数。

参数	说明
Id	缓存识别号码
名称	缓存名称
启用	启用/停用记录
大小	记录文件的大小。数据每 5 分钟自动保存到磁盘。
类型	记录的事件的类型： <ul style="list-style-type: none"> • 报警 • 日志 • 通用
存储设备	日志数据将保存在该设备上。

28 报表

报表是在被事件启动时需要被打印的信息集。当计划的事件被启动，打印在后台开始。

您可以在报表编辑器中配置报表、内容、启动条件和输出打印机。

并非所有控件都能在报表中使用。配置报表时，LRH SW 提供可用于报表的专用控件集。

报表的格式可以使用预设的页面布局模版进行定制设置。



注：LRH SW Client 不支持报表打印。

添加一个报表	340
配置图形报表	340
启动打印事件	341
默认打印机	342

添加一个报表

路径: **项目视图** > **配置** > 双击 **报表**

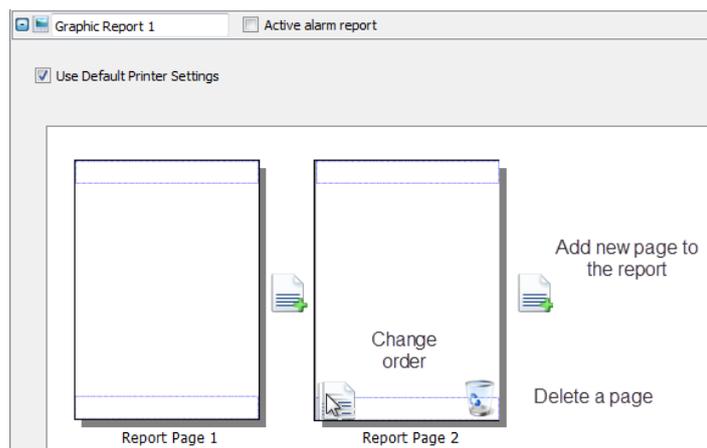
在**报表编辑器**中,单击**图形报表**:表格中将添加新的一行。

报表类型

报表类型	说明
图形报表	包含图表元素,可能包括像屏幕截图或报警等复杂控件。  重要: 每台打印机要求特定的打印机驱动。请参阅" 配置图形报表 "向下获取受支持的打印机的列表。

配置图形报表

使用**报表编辑器**配置图形报表。



添加一个报表页

单击 **+** 向报表布局中添加新页面。

当鼠标在一个页面上时,两个图标被显示,允许您将页面重新排序或删除页面。

修改报表页内容

1. 双击页面编辑其内容:**图形报表编辑器**出现。
每一页被分为:页眉、页脚和页面主体。
2. 在您要编辑的区域双击:编辑的区域以白色显示,其他的区域呈灰色。

控件集根据背景变化,仅显示您在编辑的区域可用的控件。

报表可用的控件

可用于图形报表的控件：

控件	功能
页码	自动编入页码
截屏	截取当前显示在 HMI 设备上的屏幕页面。报表页自动改变大小以适应 HMI 设备页面。  注：打印全屏，包括所有打开的对话框。
报警	事件缓存的全部内容(默认缓存是报警缓存 1)。
文本	类似文本数字控件

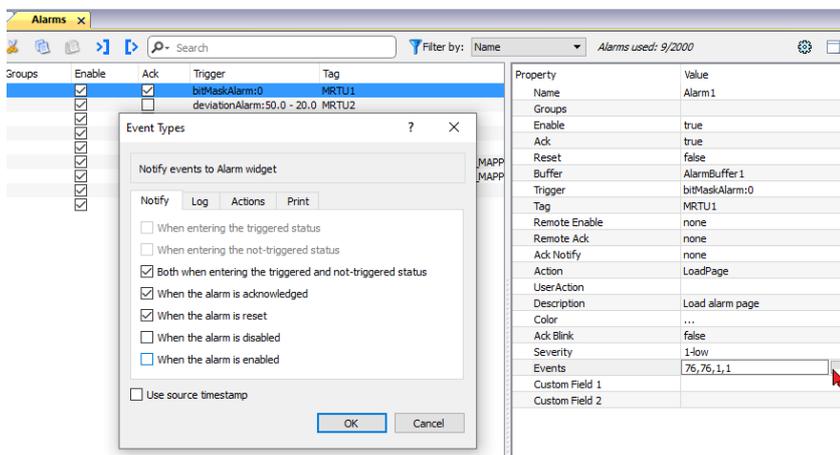
启动打印事件

报表打印可以由事件启动。

配置报警打印

路径：**项目视图 > 配置 > 双击报警**

1. 在“报警”编辑器中，从**事件属性**中打开**事件类型**对话框。
2. 在**打印**选项卡中，选择要触发打印的所有条件。

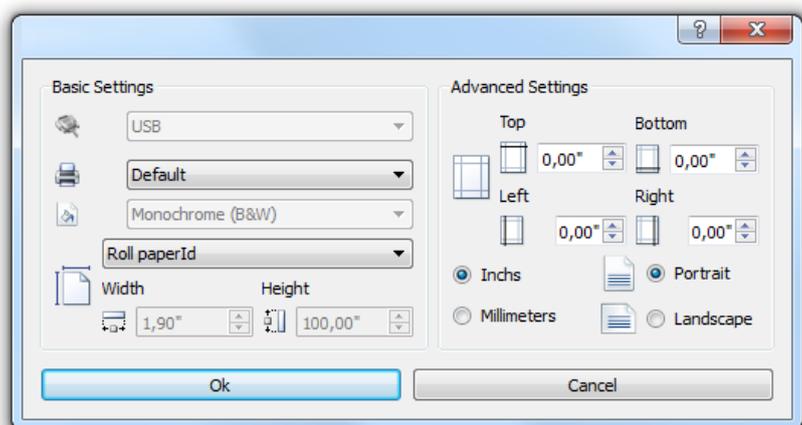


重要：一个项目中只有一个报表可以设为活动报警表，它可以是文本报表或图形报表。

运行时调整打印机设置

图形报表打印也可使用**打印图形报表**动作启动。

将动作属性**静默**设为 **false** 以显示弹出对话框。



默认打印机

打印机设置

您可以对所有的图形报表设置默认的打印机。每个报表可以配置为使用默认的打印机或其他可用的打印机。单击**打印机设置**按钮以设置打印机参数。

对于 PDF 打印机，也可定义其中的文件是使用**打印文件位置**保存的文件夹。

29 屏幕保护程序

屏幕保护程序可以在 HMI 设备不被使用时用来显示幻灯片。幻灯片在以下事件都没有出现且持续一定时间的情况下启动：

- 触摸屏幕
- 鼠标移动
- 外接键盘按键
- 活动对话框

触摸显示屏或检测到鼠标移动或按下外部键盘上的键或启动对话框时，如果屏幕保护程序处于活动状态，则将其禁用。

启动屏幕保护功能

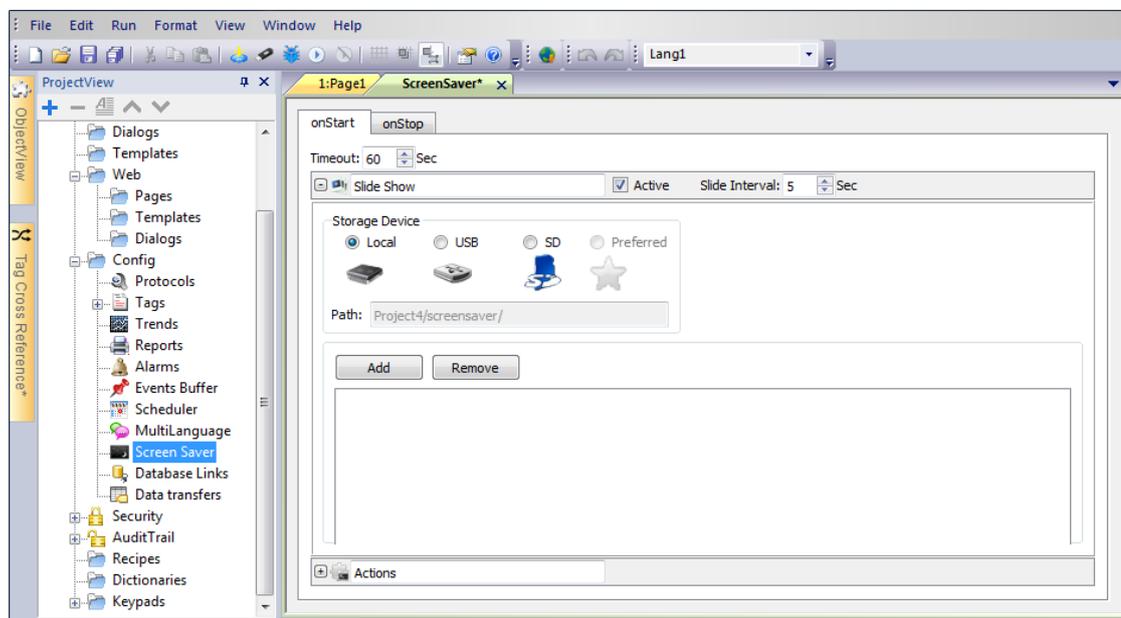
路径：*项目视图* > *配置* > *双击屏幕保护程序* > *启用*



重要：您必须在配置前启用屏幕保护程序。

配置屏幕保护程序

路径：*项目视图* > *配置* > *双击屏幕保护程序*



幻灯片参数

参数	说明
超时	屏幕保护程序启动的时间
幻灯片时间间隔	幻灯片之间的间隔
存储设备	<p>幻灯片使用的图片的位置。</p> <p>本地存储的图像保存在 <code>workspace\projectname\screensaver</code> 中，当项目已下载时，可以下载到 HMI 设备中。</p> <p>存储在 USB 或 SD 设备上的图片保存在设备本身的一个屏幕保护文件夹中。</p> <p> 重要：仅支持 JPEG 和 PNG 格式的图像。</p>

将动作关联到屏幕保护程序

动作可以由屏幕保护程序启动和/或停止来驱动。

- 单击 **启动时** 选项卡中 **动作** 旁边的 **+**，配置屏幕保护程序启动时要执行的动作。
- 单击 **停止时** 选项卡中 **动作** 旁边的 **+**，配置屏幕保护程序停止时要执行的动作。

30 Runtime 和项目备份/恢复

您可以将 HMI 设备中的所有内容，包括

- LRH SW HMI Runtime
- HMI 应用程序项目

备份到一个外部存储器中。备份可以用于恢复 HMI 设备中的内容或复制到新的 HMI 设备中。

备份功能只对启用该功能的登录用户可用。请参阅“修改访问权”在本页 319 了解详细信息。



注：LRH SW Client 中不支持备份。

备份功能

备份功能自动执行以下步骤：

1. 退出当前项目，解锁当前使用的文件。
2. 将 \QTHMI 文件夹(包含 LRH SW HMI Runtime、项目以及配方、报警、趋势之类的动态文件)的内容存档在一个 .zip 文件(标准或已加密)。
3. 重置 HMI 设备(重新加载项目)。

开始备份程序：

1. 在 LRH SW HMI Runtime 中右键单击打开上下文菜单。
2. 选择**备份:备份**对话框显示。



3. 选择存储备份文件的路径。



注：备份过程不包括存储在 USB 和 SD 卡中的文件。动态数据如存储在这些设备上的配方、趋势、事件将不被包含在备份中。

恢复功能

可以对 HMI 设备执行备份包恢复操作

- 从“上下文菜单”中(请参阅“更新包”在本页 94 了解详细信息)
- 从“系统设置”中(请参阅“系统设置”在本页 527 了解详细信息)

31 键盘

默认情况下，在 LRH SW 中提供多个键盘，以供数字输入之用。

字母键盘可以与字符串数据类型关联使用



数字键盘可以与数字数据类型关联使用



日历键盘可以与日期数据类型关联使用



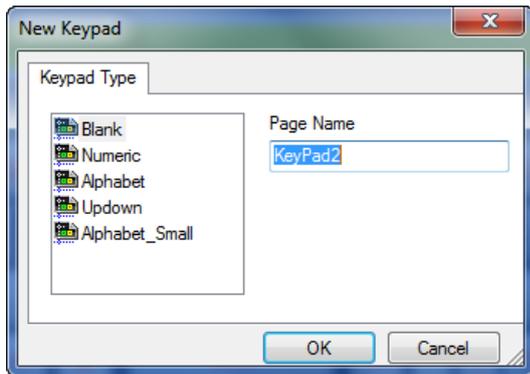
创建和使用自定义键盘	349
删除或重命名自定义键盘	351
键盘类型	351

创建和使用自定义键盘

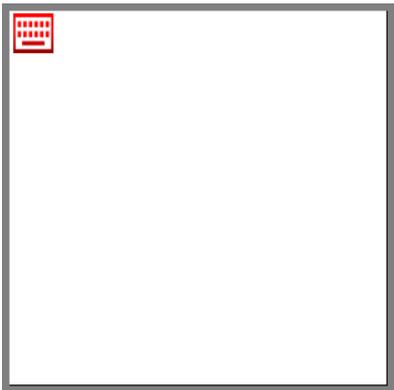
可以创建一个新的键盘或自定义已有的键盘。

创建一个键盘

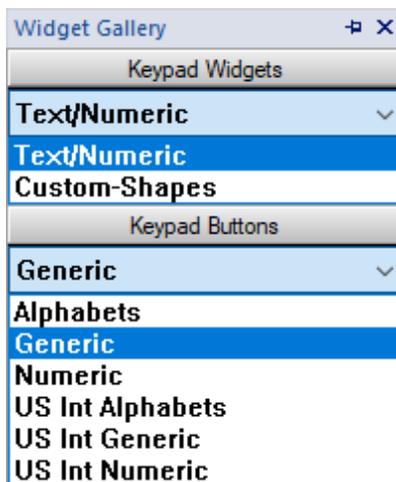
1. 在项目视图中，右键单击**键盘**并选择**插入键盘:新建键盘**对话框显示。



2. 从中选择一个可用键盘，或选择**空白**从头创建一个键盘。在这种情况下，将显示一个空白的键盘。



3. 使用控件库中的**键盘控件**和**键盘按钮**创建自定义键盘。

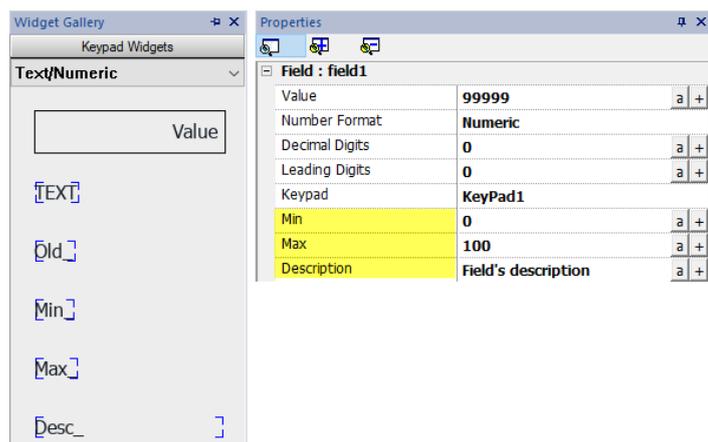


如示例所示，您创建的键盘将保存在项目文件夹中。



文本/数字控件

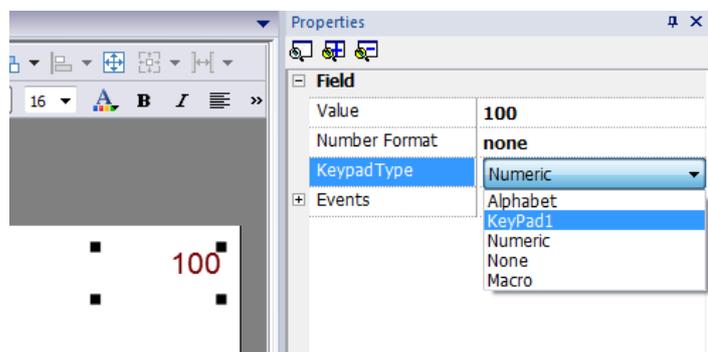
文本/数字文件夹包含一些用于键盘开发的特定控件。



数据源	说明
文本	简单文本标签
Old_	当前值
值	新值
Min_	在当前正在编辑的字段属性定义最小值。
Max_	在当前正在编辑的字段属性定义最大值。
Desc_	在当前正在编辑的字段属性定义描述。

将自定义键盘附到区域

自定义键盘随后可用**键盘**属性指向的任何字段，如本示例所示。



自定义键盘的提示和技巧

默认情况下，任何数字控件(读/写数字域)被指定数字键盘。

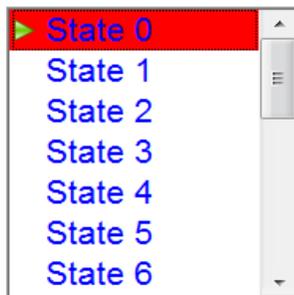
如果您想将自定义版的数字键盘应用到所有数字控件中，按如下添加：

1. 创建一个新的键盘并选择**数字**作为**键盘**类型。这将是数字键盘原始设置的备份。
2. 自定义默认数字键盘并保存。自定义版本的数字键盘现在将按照默认在项目中被指定。

请参阅["删除或重命名自定义键盘"](#)向下，了解如何重命名自定义键盘的详细信息。

键盘的上下箭头

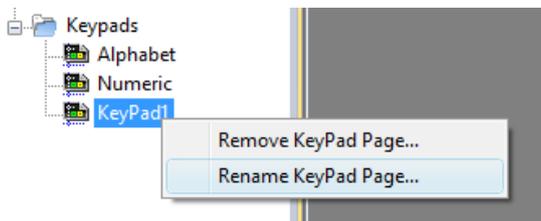
该类键盘尤其对需要该功能将光标在控件中上下移动有用。这里有一个使用**控制列表**控件的示例。请参阅["控制列表控件"](#)在本页 397 了解详细信息。



删除或重命名自定义键盘

在项目视图中，右键单击自定义键盘，选择以下选项之一：

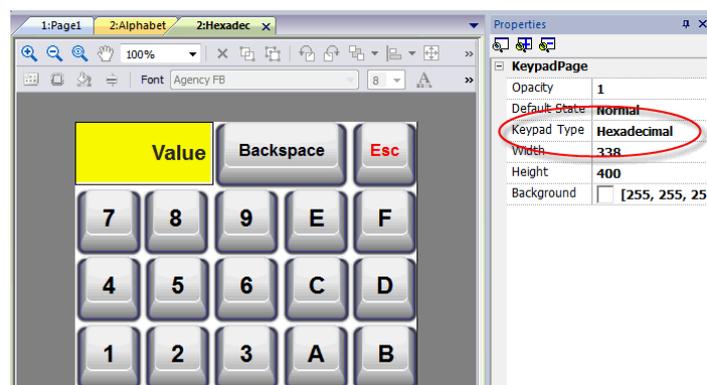
- **删除键盘页**可将键盘从项目中删除
- **重命名键盘页**可重命名键盘。



键盘类型

路径：[项目视图](#) > [键盘](#) > [双击键盘](#) > [属性](#)

设置键盘的**键盘类型**参数，定义数据条目的类型。



键盘类型	说明
自动	默认设置
十进制	只接受数字键。输入 10，如果所附字段是数字或 ASCII，键盘返回 10 且它显示为“10”，如果所附字段是十六进制，则显示为“A”。
十六进制	只接受十六进制键。输入 10，如果所附字段是数字或 ASCII，键盘返回 16 且它显示为“16”，如果所附字段是十六进制，则显示为“10”。
Ascii	所有键被启用。输入 1A，如果所附字段是数字，键盘返回 1A 但它显示为“1”，如果所附字段是 ASCII 或十六进制，则显示为“1A”。

键盘位置

键盘的**Runtime 定位**属性可用来定义屏幕上键盘的显示位置。

选项	说明
自动	最佳位置根据数据条目的要求选择。
绝对	输入 X、Y 坐标来识别确切的位置。
左上	预设的屏幕位置
左中	
左下	
中上	
中中	
中下	
右上	
右中	
右下	

如果您不想拖拽移动键盘位置，请选中**锁定键盘位置**选项。

32 外置键盘

LRH SW HMI Runtime 可以通过 USB 连接外接键盘。

键盘可以用于：

- 数据输入(默认)
- 执行特定按键对应的动作

例如右箭头键单击事件可以映射加载页面动作。

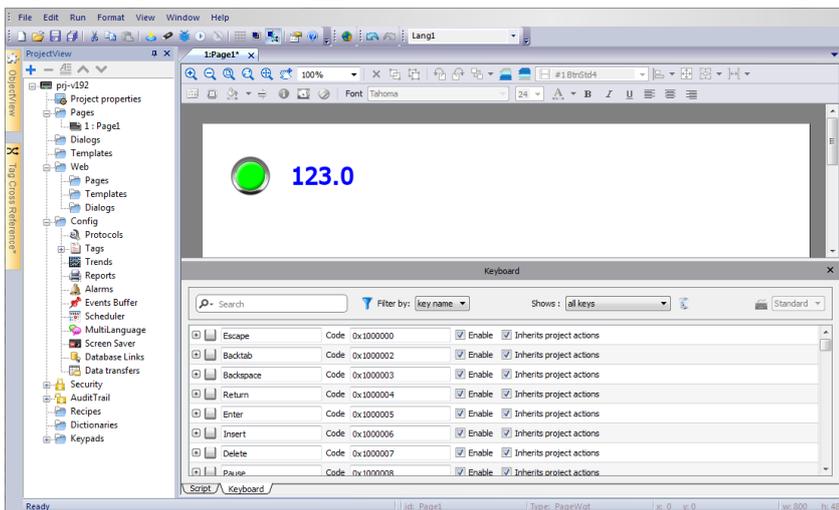
键盘可以在项目级别编辑，这样这些设置对所有页面都有效。页面中，您可以选择哪些键的设置要使用项目的设置，哪些键的设置要在具体的页面自定义。

打开外接键盘

1. 在页面编辑器中，点击工作区底部的**键盘/脚本**右侧的图标：键盘/脚本编辑器将显示。
2. 选择**键盘**选项卡。



键盘编辑器中的每一行对应一个键。



会显示每一个键的如下信息：

元素	说明
标签	键名称
代码	键代码
启用	键启用状态
继承项目动作	定义该键是否沿用项目级别编辑的动作

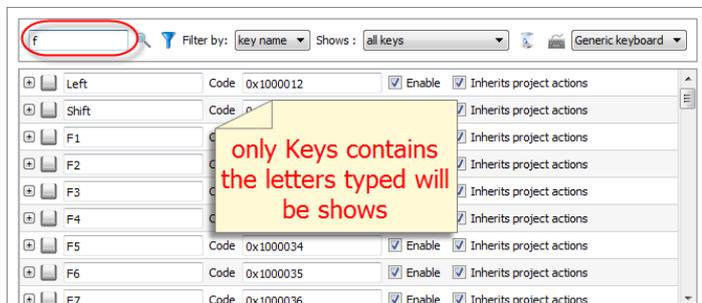
此处为可能的配置：

启用	继承项目动作	编辑器外观	LRH SW HMI Runtime 行为
已核实	未核实	动作列表显示页面动作(如果列表空白,则显示空白)	仅仅执行页面动作(如果有的话)。
已核实	已核实	动作列表仅显示项目动作,不能编辑	仅仅执行配置项目动作(如果有的话)。
未核实	已核实	沿用项目动作验证框,所有动作列表被停用。动作列表仅显示项目动作。	无页面或项目动作将被执行。
未核实	未核实	沿用项目动作验证框,所有动作列表被停用。动作列表仅显示项目动作。	无页面或项目动作将被执行。

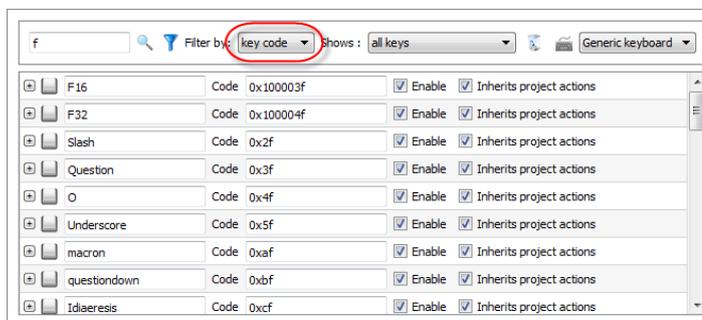
搜索并过滤	357
显示	357
删除动作关联	357
键盘布局	358
启用/停用键盘	358
将动作与按键关联	358

搜索并过滤

要显示一组经过筛选的键，请在**筛选条件**中选择**键名称**并在搜索栏中输入一个字母：只有名称中包含该字母的键会在键盘编辑器中显示。



或者在**筛选条件**中选择**键代码**并在搜索栏中输入一个字母：只有代码中包含该字母的键会在键盘编辑器中显示。



显示

您可以简单地选择将要列在键盘编辑器窗口的键。要显示数量有限的一组按键，请在**显示**中选择一个选项。

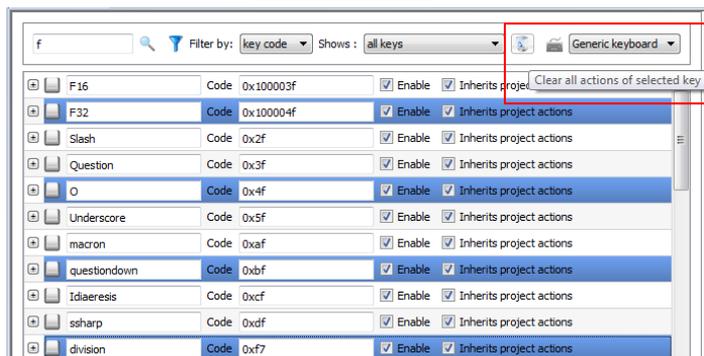
选项	说明
所有键	所有在键盘布局中可用的键被列出
修改键	只有在页面级别与动作相关的键被列出
修改项目中的键	只有在项目级别与动作相关的键被列出

删除动作关联

要删除按键与动作之间的所有创建的关联：

1. 选择您想删除的关联按键。
2. 单击**清除所选按键的所有动作**按钮。

如果是在页面级别操作，页面动作将被删除，如果是在项目级别操作，项目动作将被删除。

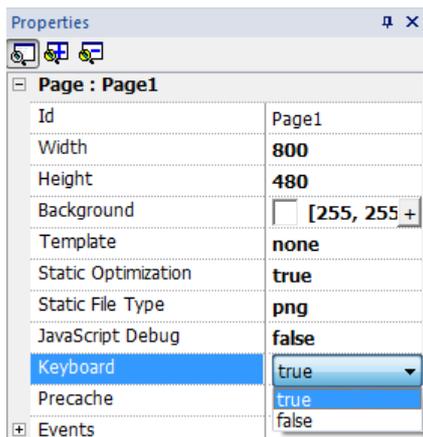


键盘布局

从**键盘布局**组合框中选择键盘的布局。**通用键盘**是指通用国际键盘布局。

启用/停用键盘

您可以在项目级别和页面级别启用/停用键盘动作。要启用键盘动作，请在**属性**窗格中将**键盘宏**设为 **true**。

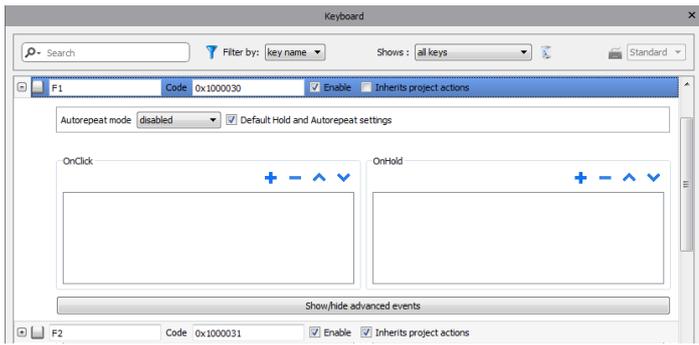


您可以在运行时使用键盘宏动作启用/停用键盘动作。请参阅["键盘动作"](#)在本页 173 了解详细信息。

将动作与按键关联

在键盘编辑器上将动作与按键相关联。

1. 单击您要编程的键旁边的 **+**: 键配置字段出现。



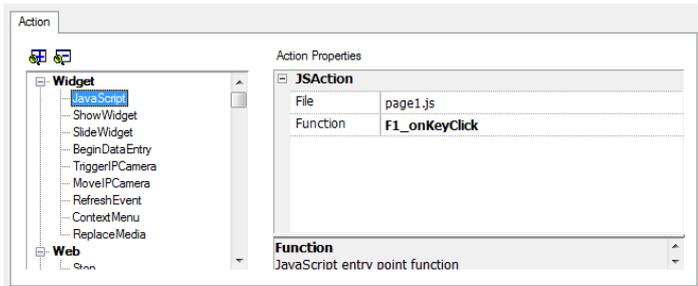
1. 单击 + 添加动作。

您可以将动作同时关联至**单击**事件和**长按**事件。

请参阅"事件"在本页48了解详细信息。



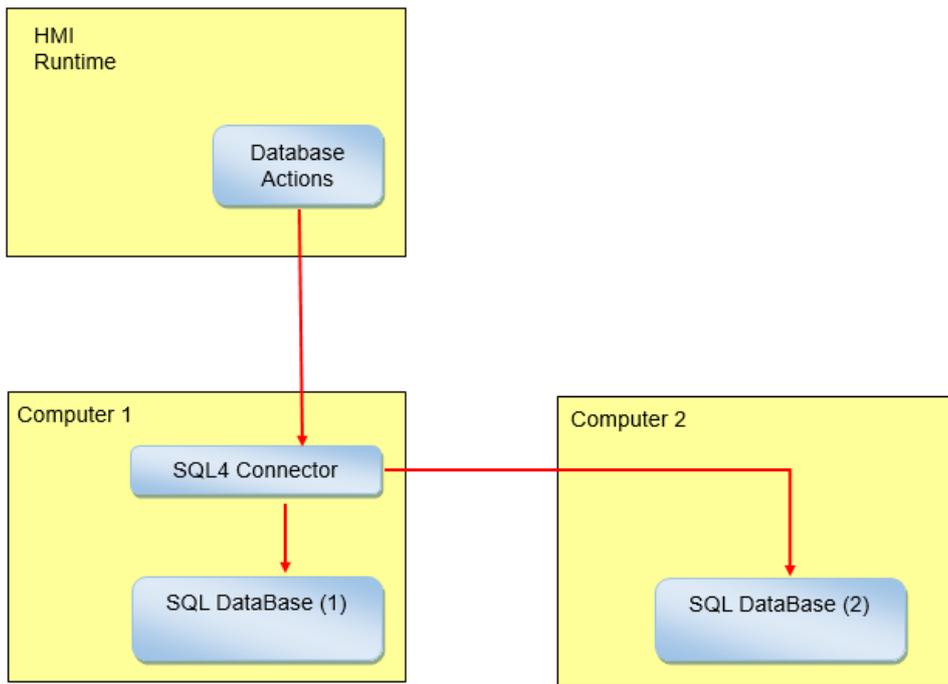
注:注: JavaScript 代码可以与按键事件关联。



33 将数据存储到外部数据库

LRH SW 能够连接到 SQL4 自动化连接器，这是一种工业应用软件解决方案。其将 HMI、PLC 和机械手控制与 SQL 数据库直接连接。HMI 通过连接器直接访问 SQL 数据库，并且可以通过使用 SQL 命令(结构化查询语言)插入、更改和删除表中的数据项查询数据。

数据库地址通过 ODBC 进行通信。因此，所有 SQL 数据库可集成为一体，支持 ODBC 接口。SQL 语法需要适应给定的数据库，例如 MS SQL 服务器、MySQL、MS Office Access、SQLite、Oracle、PostgreSQL 等



如需将数据存储在外数据库：

1. 在有数据库的计算机上或 HMI 设备和数据库之间的计算机上安装 SQL4 自动操作工具。
2. 配置该 SQL4 自动操作工具。
3. 创建一个使用指定的 DB 动作对外部数据库进行存取的项目。

安装 SQL4	362
配置 SQL4	362
配置 HMI 项目	364
用 JavaScript 传输数据	364
数据库表格	366
自定义表格	367
连接限制	367

安装 SQL4

下载最新版本的 SQL4 并将其安装在计算机上。请访问 www.sql4automation.com 了解详细信息并进行下载。
该文件中描述的步骤，参考 SQL4 自动控制连接器版本 3.3.2.0

配置 SQL4



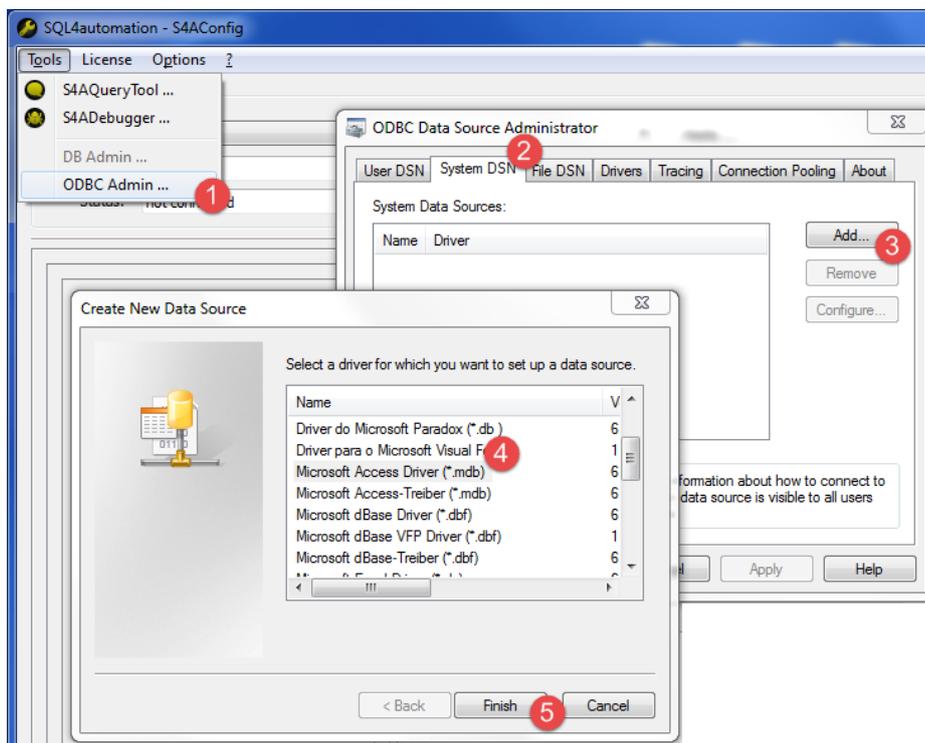
重要: 请参阅 SQL4 用户手册，了解详细的配置说明。

如下是对如何访问 MS 办公数据库 (MS Access) 的粗略描述。

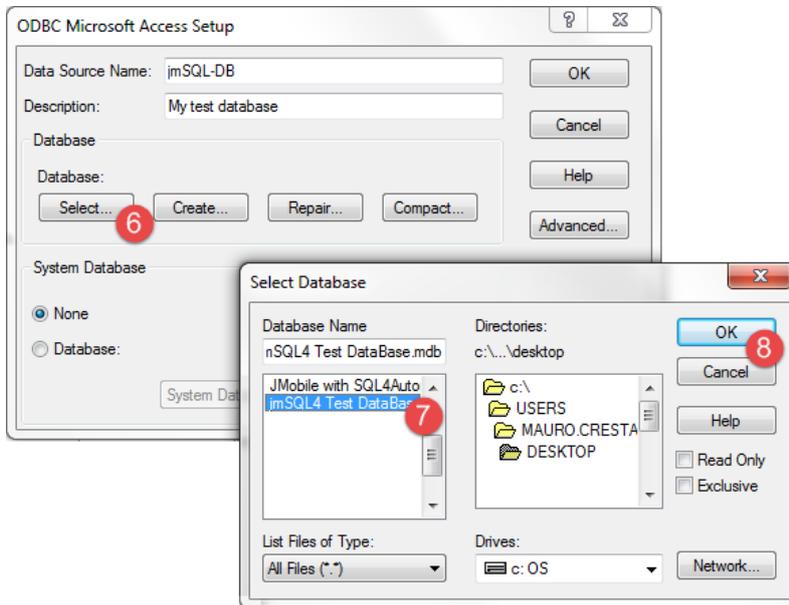
您必须在计算机上安装 MS 办公软件套件，使用 Microsoft Access 创建一个空白数据库。

启动 SQL4，按照步骤配置您的 SQL4 自动控制连接器：

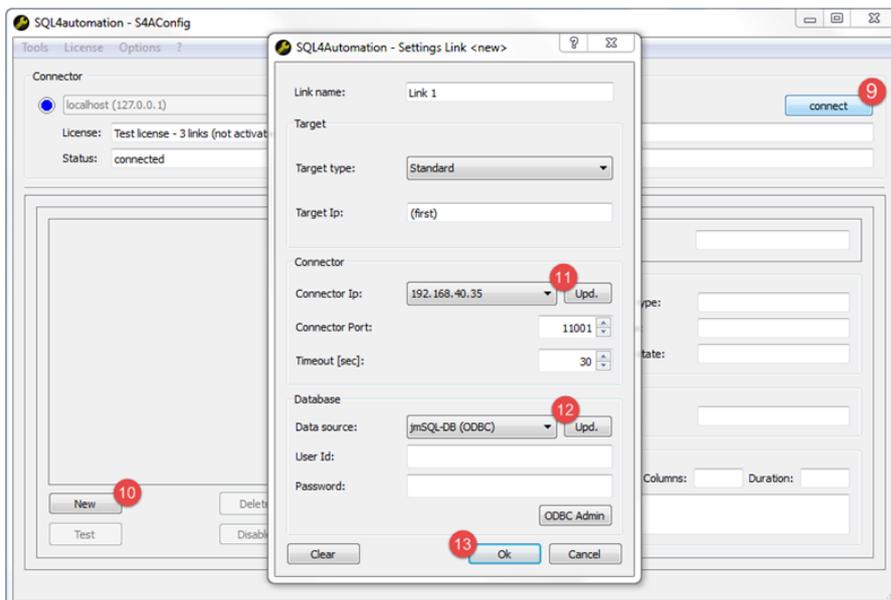
1. 选择 **ODBC Admin: ODBC 数据源管理员** 对话框显示。
2. 选择 **系统 DSN** 选项卡。
3. 单击 **添加: 创建新数据源** 对话框显示。
4. 选择 **Microsoft Access 驱动**
5. 单击 **完成** 确认。



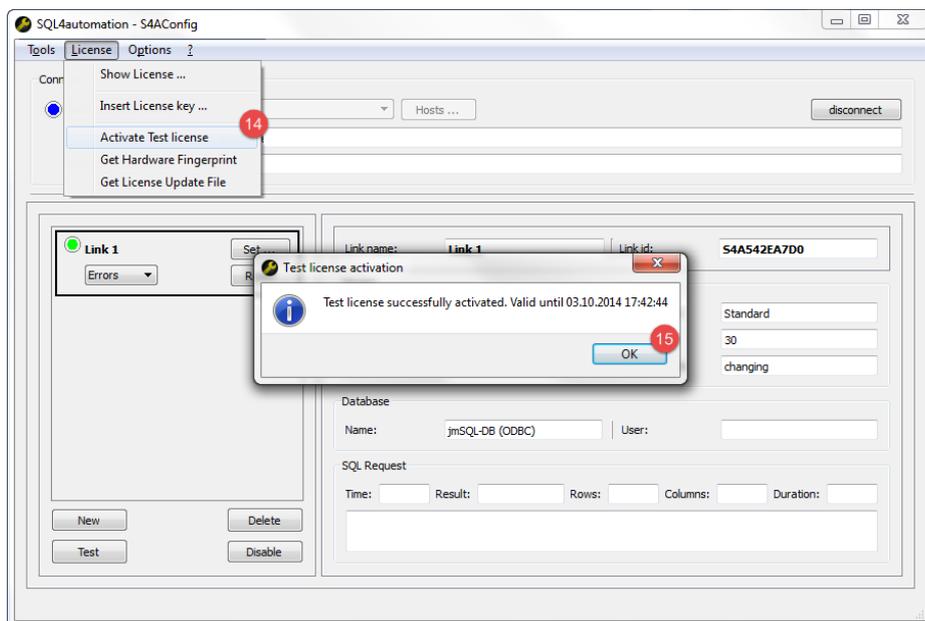
6. 输入**数据源名称**和**描述**，然后单击**选择:选择数据库**对话框显示。
7. 选择您的Access数据库。
8. 单击**确定**确认。



9. 如果是首次连接，请单击**新**选择您的数据源
10. 选择您的计算机的**IP地址**。这将是您的HMI设备使用的连接IP地址。
11. 选择数据源。
12. 单击**确定**确认。
13. 单击**连接**。



14. 选择许可证 > 激活测试许可证: 如果链接 1 LED 变为绿色, 则表明过程已成功完成。
15. 单击确定确认。

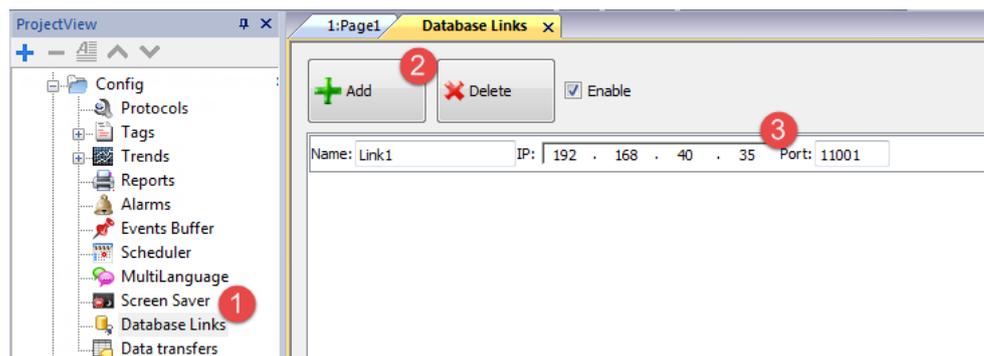


配置 HMI 项目

路径: 项目视图 > 配置 > 双击数据库链接

要保存项目数据到外部数据库, 您需要创建有具体数据库的连接。

1. 在数据库链接编辑器中, 选择启用以启用功能。
2. 单击添加来创建新的链接。
3. 在有 SQL4 自动控制联结器的计算机输入 IP 地址。



重要: 此处的链接名称不一定与 SQL4 自动化连接器中定义的相同。但该名称将在使用远程数据库的所有动作中被使用。

用 JavaScript 传输数据

一些用于从 HMI 设备传输数据到远程数据库的动作可以用作 JavaScript 的宏, 如下示例所示。

数据库连接状态通过系统变量标签可用。请参阅第 1 页上的“数据库变量”。

错误状态可以用动作复位。请参阅“数据库动作”在本页 169

```
function myButton1_onMouseClicked(me, eventInfo) {
    var CustomSQL = ' ' ;
    var DatabaseLink ='Link1';
    project.dbInit(DatabaseLink, CustomSQL);
};

function myButton2_onMouseClicked(me, eventInfo) {
    var CustomSQL = ' ' ;
    var DatabaseLink ='Link1';
    var Tags ='Alarm1;SystemTime;Tag01;Tag02;';
    project.dbReadTags(DatabaseLink, CustomSQL, Tags);
};

function myButton3_onMouseClicked(me, eventInfo) {
    var CustomSQL = ' ' ;
    var DatabaseLink ='Link1';
    var Tags ='Alarm1;SystemTime;Tag01;Tag02;';
    project.dbWriteTags(DatabaseLink, CustomSQL, Tags);
};
```

dbQuery

```
project.dbQuery(databaseLink, customSQL, dbCallback);
```

使用该查询，您可以执行 SQL 查询。

参数	说明
databaseLink	链接到数据库使用
customSQL	SQL 查询字符串
dbCallback()	当查询数据就绪时将被调用的功能

dbCallBack

```
project.dbCallBack(dbStatus, dbResponse);
```

参数	说明
dbStatus	0: 未发现错误
dbResponse	查询回应。表格栏名称后是它的行:

参数	说明
	在示例中： 标签名称 - 标签值 Tag09 - 103 Tag10 - 302

Script

```

1
2 function JSI_onMouseClicked(me, eventInfo) {
3
4     var customSQL = "SELECT Tagname, Tagvalue FROM Tags WHERE Tagname='Tag09' OR Tagname='Tag10' ORDER BY Tagname"
5     var databaseLink = "Link1";
6     project.dbQuery(databaseLink, customSQL, dbCallback)
7 };
8
9
10 function dbCallback(dbStatus, dbResponse){
11
12     alert("SQL Answer = " + dbResponse + "\ndbStatus =" + dbStatus);
13 };
14
15

```

数据库表格

这里是数据库动作使用的数据库表格的结构。



注：这些表格可以用 **DBInit** 动作在空白数据库上生成。

表格：标签

FieldName	文本 (255)	主键码
标签值	文本 (255)	

表格：趋势

Id	Long Integer	主键码
TrendName	文本 (255)	
样本时间	文本 (255)	
趋势值	文本 (255)	
Quality	文本 (255)	
刷新时间	文本 (255)	

表格：配方

配方	文本 (255)	主键码
设置名称	文本 (255)	主键码
元素名称	文本 (255)	主键码
设置值	文本 (255)	

表格：事件

Id	Long Integer	主键码
事件名称	文本 (255)	
样本时间	文本 (255)	
事件类型	文本 (255)	
事件附属时间	文本 (255)	
事件值	文本 (255)	

自定义表格

从 DB 动作发出的 SQL 查询在项目文件 `config\dbconnector.xml` 中列出。

修改该文件中定义的命令来自定义 DB 动作中发出的 SQL 字符串，然后访问一个不同的结构化数据库。

例如

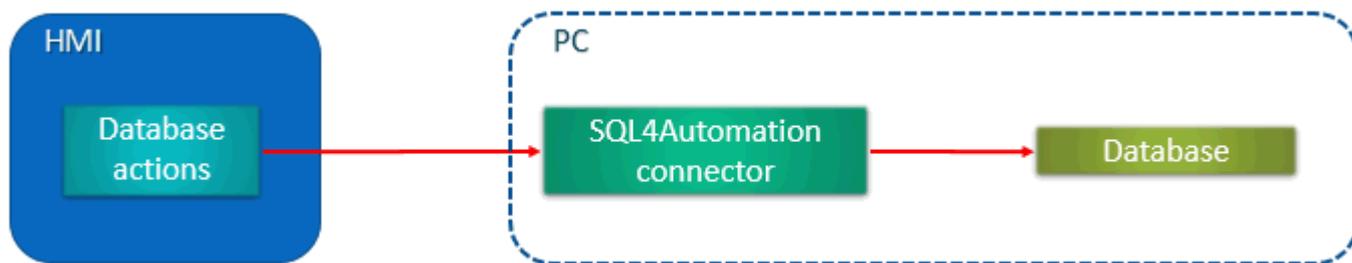
```
CREATE TABLE myTagsTable (tagname VARCHAR(255) PRIMARY KEY, tagvalue VARCHAR(255))
UPDATE myTagsTable SET Tagvalue= '%_JMV' WHERE Tagname= '%_JMT'
INSERT INTO myTagsTable (Tagname, Tagvalue) Values ('%_JMT', '%_JMV')
```

其中 "%_JMV" 将被替换为标签值，"%_JMT" 被替换为标签名称。

连接限制

SQL4 使用 USB 加密狗，具有众多预定义连接的许可证。在 SQL4 自动化连接器中，连接被称为“链接”。连接数量视您购买的许可证而定。

SQL4 自动化连接器可安装在运行该数据库的同一计算机/服务器上：



或不同的计算机/服务器上：



34 OPC UA 服务器

路径: 项目视图 > 配置 > 接口 > 双击 **OPC UA**

根据 OPC UA 标准使用 OPC UA 服务器发布数据。

参数	说明
启用 OPC UA 服务器	用于激活 OPC UA 服务器的主标志。 HMI 设备中定义的数据值通过 OPC UA 服务器发布。

功能

参数	说明
启用报警	激活实时报警数据的发布(活动报警)。
启用历史报警	激活历史报警数据的发布。
启用趋势	激活趋势数据发布。
标签组	仅属于所选组的标签将对 OPC UA 服务器可用。
报警组	仅属于所选组的报警将对 OPC UA 服务器可用。

网络

参数	说明
节点名称	输入节点名称或留空以使用主机名。
端口	OPC UA 服务器的端口号。  提议的默认端口号可能与 OPC UA 客户端使用的端口不同。

验证

选择 OPC UA 服务器的验证选项。



OPC UA 客户端将负责从可用选项中选择最合适的选项，根据其功能使用。

用户身份验证

参数	说明
匿名	支持匿名客户端。
用户/密码	使用接受的用户名进行验证。  任何有效用户可以不受限制地访问 OPC UA 服务器(请参阅"配置用户"在本页 324)。

使用 x.509 证书

OPC UA 提供了使用数字证书的安全通信通道。可配置的端到端安全级别可用，确保了每条消息的加密、机密性和完整性。启用时，服务器将验证客户端证书，反之亦然。



OPC UA 客户端将负责从可用选项中选择最合适的选项，根据其功能使用。

安全模式	说明
无	允许无证书连接。  不建议用于公共网络。
签名	OPC UA 客户端必须提供自己的证书:允许通过已签名消息进行通信。
签名并加密	OPC UA 客户端必须提供自己的证书:允许通过已签名和已加密消息进行通信。

安全策略	说明
Basic128Rsa15	接受的加密级别(仅当“安全模式”处于活动状态时可用)。
Basic256	接受的加密级别(仅当“安全模式”处于活动状态时可用)。
Basic256Sha256	接受的加密级别(仅当“安全模式”处于活动状态时可用)。

参数	说明
自动信任任何新客户端	接受由 OPC UA 客户端提供的所有证书。
受信任的证书	仅接受提供所列证书之一的 OPC UA 客户端。  要将新证书添加至受信任的证书列表中，必须具有由 OPC UA 客户端设备所有者支持的证书文件。同时接受二进制和 ASCII 证书文件格式。

全局发现服务器

OPC UA 服务器与 GDS 推送型号兼容。这意味着，您可以使用远程 GDS 工具进行集中证书管理。



为了能够成功连接到 OPC UA 服务器，必须检索 GDS 工具的证书并将其添加至 OPC UA 服务器的受信任证书列表。

例如

当 OPC UA 客户端尝试与 OPC UA 服务器连接时，服务器将检查客户端证书是否在自己的受信任证书列表中可用。如果找不到，通信将被拒绝且证书会存储在不可靠的证书列表中。

使用 GDS 工具，您可以连接到 OPC UA 服务器，检查可用证书并定义每个证书的信任或不信任状态。

证书文件

HMI 设备将证书存储在以下文件夹的子文件夹中：`"/workspace/<ApplicationName>/config/pkiserver"`

- 自己
自己的证书和私人密钥
- 受信任
受信任的自签名证书和 CA 证书
- 被拒
被拒证书
- 签发者
验证信任链所需的可信中间(不直接信任)CA

服务器标识

参数	说明
制造商名称	<p>用户可读取产品制造商名称。</p> <p> OPC UA 客户端可以从以下标签中检索其信息： <code>ServerName Objects Server ServerStatus BuildInfo ManufacturerName</code></p>
产品名	<p>在服务器中运行时用户可读取产品名。</p> <p> OPC UA 客户端可以从以下标签中检索其信息： <code>ServerName Objects Server ServerStatus BuildInfo ProductName</code></p>

证书参数

服务器证书可以自动生成或通过添加现有证书文件生成。

自动生成自签名证书

启用自动生成的证书时，证书将在用户每次更改证书参数后重新生成。该证书也可替换为任何显式设置的证书。

证书参数

每个证书必须包含应识别证书及其限制的信息。如果您已选择使用自生成证书，请在自生成证书中输入所需信息。否则，将从您提供的证书中读取参数。

参数	说明
服务器名称 (通用名称)	证书名称(例如,设备名称)。
组织	组织名称
单位	组织单位  此字段可能有助于区分组织内部的不同部门。
位置	位置字段表示组织所在的城市。
状态	州或省字段指定组织所在的物理位置。  不应缩写州或省字段内容。例如, "CA" 不是有效的州名称。“加州”是正确的州名称。
国家	X.509 命名方案标准需要一个 2 字符国家/地区代码。  美国的国家/地区代码是 US ; 意大利的国家/地区代码是 IT 。

参数	说明
产品 URI	服务器的全球性唯一标识。 示例: "urn:NodeName:CompanyName:ServerName"
DNS 名称 IP 地址	安装此 OPC UA 服务器的设备的 DNS 名称或 IP 地址。 单个证书中可以包含多个 DNS 名称和/或 IP 地址。  仅当运行 OPC UA 服务器 的 IP 地址 包含在此列表中, 该证书才有效。
有效性	从创建日期开始的证书有效期
密钥长度	RSA 加密算法使用的密钥长度

用于生成证书脚本

如果要提供自己的证书, 请注意证书必须包括 OPC UA 标准需要的“主题备选名称 (SAN)”参数。

以下示例说明如何使用公共 OpenSSL-Win32 库生成证书(参考: <https://www.openssl.org/>)

```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"
set NodeName=HMI-Server
set IPAddress=192.168.44.165

rem Generate an RSA key
%OpenSSL% genrsa -out server-key.pem 2048
```

```
rem Creating Certificate Signing Requests
    %OpenSSL% req -new -key server-key.pem -out server.csr -subj
"/ST=Italy/C=IT/L=Verona/O=CompanyName/OU=R&D Team/CN=OPCUAServer@%NodeName%

rem Creating Certificate (.pem)
    echo subjectAltName=URI:urn:%NodeName%:CompanyName:OPCUAServer,IP:%IPAddress% >
san.txt
    echo
keyUsage=digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment,keyCertSign
>> san.txt
    echo extendedKeyUsage=critical,serverAuth,clientAuth >> san.txt
    echo authorityKeyIdentifier=keyid,issuer >> san.txt
    echo basicConstraints=CA:TRUE >> san.txt
    %OpenSSL% x509 -req -days 3650 -in server.csr -signkey server-key.pem -out
server.crt -extfile san.txt

rem Convert Certificate (.der)
    %OpenSSL% x509 -in server.crt -outform der -out server.der

rem Not necessary files
    del san.txt

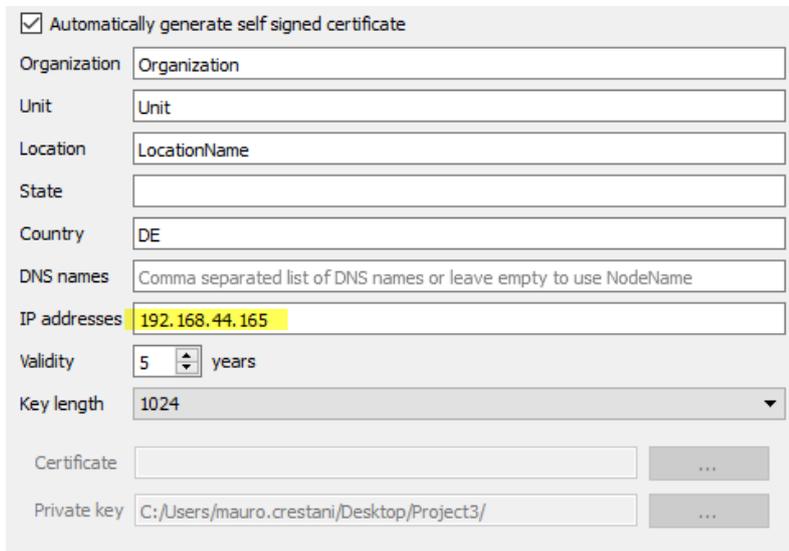
pause
```

使用自签名证书

本章是一个逐步式示例，解释如何配置两个 HMI 设备使用自签名证书进行通信

OPC UA 服务器

1. 创建一个简单的项目，包括一些标签
2. 打开 OPC UA 对话框并启用 OPC UA 服务器。确保启用标签组(例如，选择“所有”)
3. 在“IP 地址字段”中输入运行 OPC UA 服务器的 HMI 设备的 IP 地址



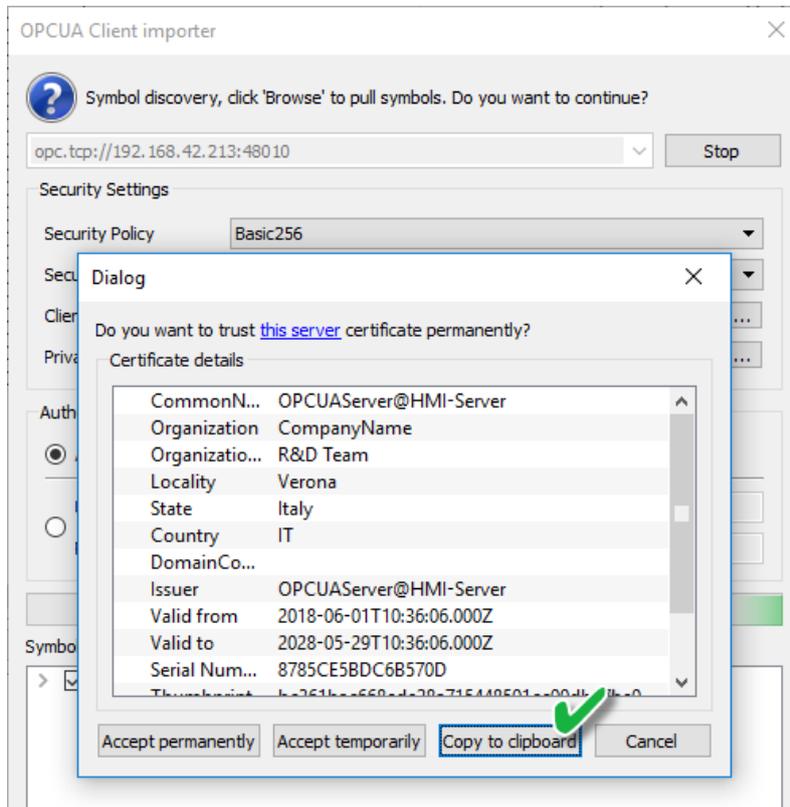
The screenshot shows a configuration window for generating a self-signed certificate. It includes the following fields and options:

- Automatically generate self signed certificate
- Organization: Organization
- Unit: Unit
- Location: LocationName
- State: (empty)
- Country: DE
- DNS names: Comma separated list of DNS names or leave empty to use NodeName
- IP addresses: 192.168.44.165
- Validity: 5 years
- Key length: 1024
- Certificate: (empty) ...
- Private key: C:/Users/mauro.crestani/Desktop/Project3/ ...

4. 将项目下载至 HMI 设备

OPC UA Client

5. 创建一个简单的项目
6. 添加 OPC UA 客户端协议。输入远程 OPC UA 服务器的 IP 地址及其端口号 (48010)。证书参数留空。
7. 打开标签编辑器并导入标签。选择“OPC UA 发现”模式
8. 选择将证书复制到如下图所示的剪贴板。然后，关闭此对话框并返回协议配置对话框，将证书粘贴到“服务器证书”字段中。



9. 重复第 7 步，接受服务器 OPC UA 证书，导入一些标签。请注意，您可以永久或临时接受证书。如果您永久接受证书，证书副本将保存在计算机中以供日后使用，而不再弹出对话框询问您确认。



证书文件将被复制到以下文件夹中：

`%AppData%\Roaming\Lovato\studio\OPCUA\pk\trusted\certs`

10. 再次打开协议对话框。选择安全策略 = Basic256 和安全模式 = 签名并加密
11. 将项目下载至 HMI 设备

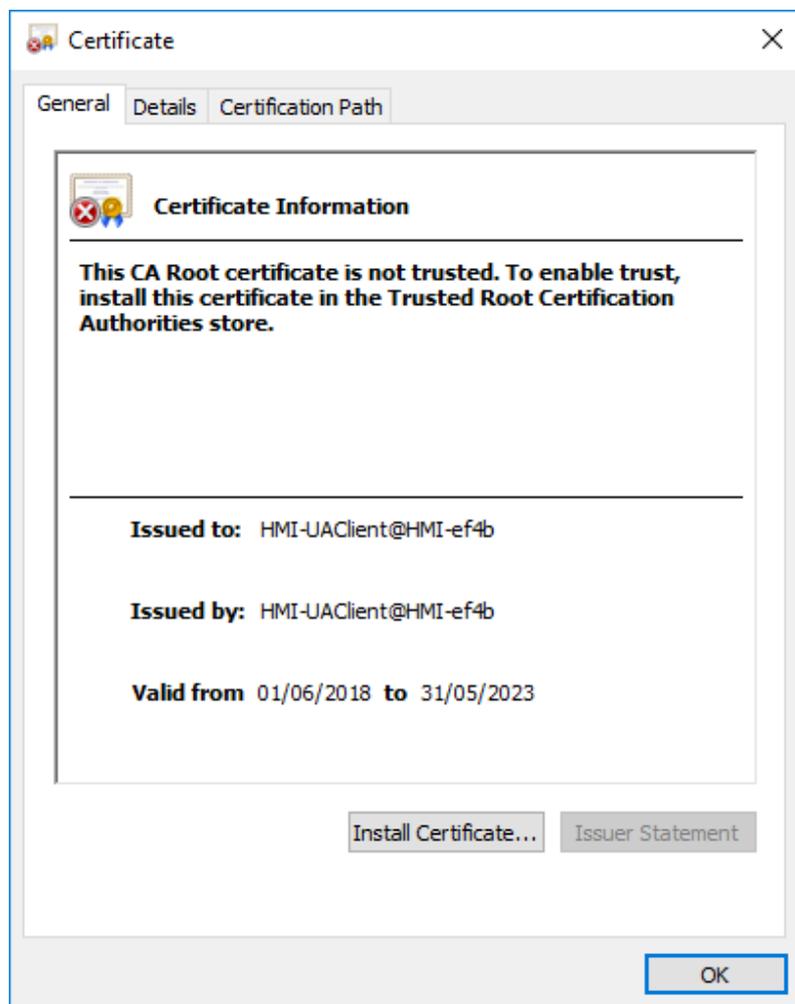
由于我们将 OPC UA 客户端中的“客户端证书”字段留空，因此 OPC UA 客户端协议已生成自己的证书并将其发送至 OPC UA 服务器，但由于服务器不知道此证书，所以它拒绝连接请求。现在我们必须告诉服务器信任这些证书。为此，有不同可用的方式。

使用 FTP 客户端使被拒证书可信：

1. 使用 FTP 客户端连接到 OPC UA 设备
2. 在证书文件夹中查找并将被拒证书从被拒文件夹移至受信任的文件夹。

/workspace/<YourProjectName>/config/pkiserver/rejected
 /workspace/<YourProjectName>/config/pkiserver/trusted/certs

您可以双击证书文件将其打开并查看证书参数，以确保您正在验证的证书



现在，通信将启动

使用 **GDS** 工具使被拒证书可信：

1. 打开 **GDS** 工具并导出其证书
2. 打开项目并将 **GDS** 工具证书添加至受信任的证书列表
3. 将更新项目下载至 **HMI** 设备

现在您可以使用 **HMI** 设备中的工具管理证书。

使用外部证书

本章是一个逐步式示例，解释如何配置两个 **HMI** 设备使用外部证书进行通信。

生成证书

您可以使用本手册中指定的脚本生成自己的证书副本，一个用于 OPC UA 服务器，另一个用于 OPC UA 客户端。

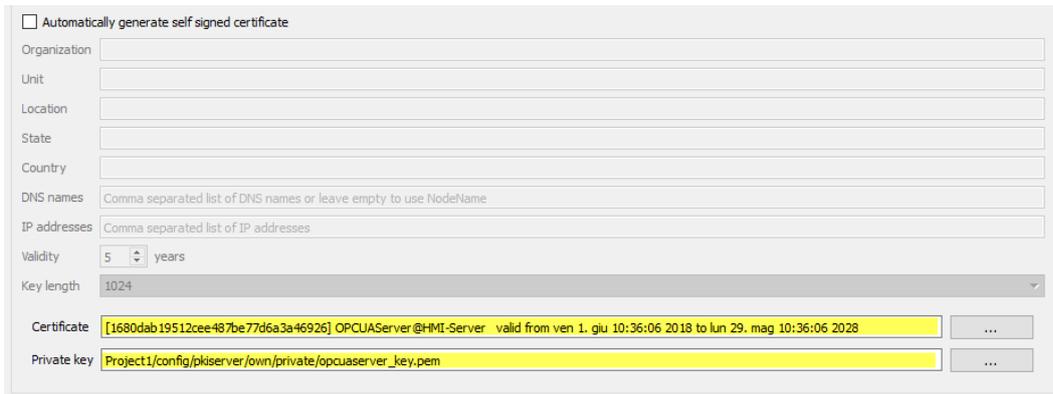
1. 安装 OpenSSL-Win32 库 (参考: <https://www.openssl.org/>)
2. 使用脚本 ("用于生成证书的脚本" 在本页 372) 生成 OPC UA 服务器证书。确保在运行脚本之前，使用运行 OPC UA 服务器的 HMI 设备的 IP 地址设置 IP 地址变量。
3. 在 OPC UA 客户端协议手册中查找示例脚本，为 OPC UA 客户端协议生成证书

OPC UA 服务器

4. 创建一个简单并使用一些标签的项目
5. 打开 OPC UA 对话框并启用 OPC UA 服务器。确保启用标签组 (例如，选择“所有”)
6. 将 client.der 证书添加至受信任的证书区域，以使 OPC UA 客户端与 OPC UA 服务器通信



7. 移除“自动生成自签名证书”上的复选标记并添加服务器证书 (server.der) 和服务器私人密钥 (server-key.pem)



8. 将项目下载至 HMI 设备

OPC UA Client

9. 创建一个简单的项目
10. 添加 OPC UA 客户端协议。
11. 输入远程 OPC UA 服务器的 IP 地址及其端口号 (48010)。
12. 打开 ASCII 版本的服务器证书 (server.crt), 移除所有换行符, 然后在服务器证书字段中复制粘贴证书的 ASCII 字符。
13. 为客户端证书 (client.crt) 和客户端私人密钥 (client-key.pem) 重复相同的步骤
14. 选择安全策略 Basic256 和安全模式 = 签名并加密
15. 打开标签编辑器并导入标签。选择“OPC UA 发现”模式
16. 接受 OPC UA 服务器证书, 导入一些证书。请注意, 您可以永久或临时接受证书。如果您永久接受证书, 证书副本将保存在计算机中以供日后使用, 而不询问您确认
17. 将项目下载至 HMI 设备

请注意, OPC UA 客户端正在从使用指定证书的 OPC UA 服务器中检索数据。

报警映射

根据以下规则将报警状态映射至 OPC UA 状态:

OPC UA 报警状态	LRH SW 报警状态
Opcua.Alarm.Active	当报警状态启动时为真 <ul style="list-style-type: none"> • TRIGGERED • TRIGGERED_NOT_ACKED • TRIGGERED_ACKED
Opcua.Alarm.Acked	无需报警确认时为真 <ul style="list-style-type: none"> • TRIGGERED_ACKED • NOT_TRIGGERED_ACKED • NOT_TRIGGERED
Opcua.Alarm.Retain	报警待定时为真 <ul style="list-style-type: none"> • TRIGGERED • TRIGGERED_NOT_ACKED • TRIGGERED_ACKED • NOT_TRIGGERED_ACKED, 但需要 RESET
Opcua.Alarm.Confirmed	报警返回时为真(未启动、确认和重置) <ul style="list-style-type: none"> • NOT_TRIGGERED <div style="display: flex; align-items: center;">  <p>仅当报警配置为需要 RESET 时, 此信息才可用。</p> </div>

35 MQTT 接口

路径: 项目视图 > 配置 > 接口 > 双击 **MQTT**

根据 MQTT 标准使用 MQTT 接口发布数据。

请注意, 标签或报警必须在组内定义, 才能通过 MQTT 协议传输。

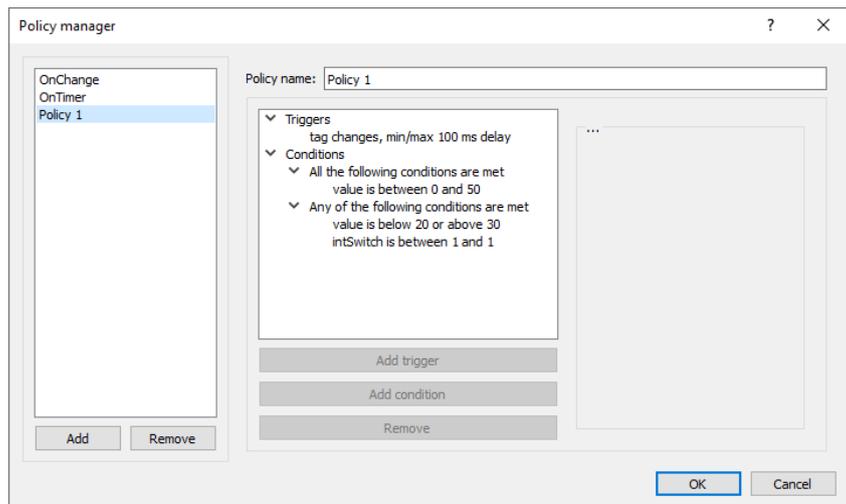
参数	说明
启用 MQTT 接口	用于激活 MQTT 服务的主标志。 选定的标签组将发布到 MQTT 代理。
启用报警	选定的报警组将发布到 MQTT 代理。 只要报警状态发生更改, 就会发布报警。

标签配置

参数	说明
启用	启用组内列出的标签传送
标签组	满足指定的警察条件时, 将传送标签列表。
QoS	要使用的 QoS <ol style="list-style-type: none">0. 最多交付一次(即付即忘), 这意味着无需确认1. 至少交付一次, 这意味着需要确认2. 仅交付一次, 这意味着完成了 4 步握手
保留	此标志定义消息是否由代理保存为指定主题的最后一个已知值。当新客户订阅主题时, 他们会收到该主题上保留的最后一条消息。
策略	定义用于决定何时发布标签值的标准。如果为空, 则使用表顶部定义的“默认推送策略”。

管理推送策略

策略由触发条件和几个(可选)条件组成, 必须进行验证才能传输标签值。



触发器

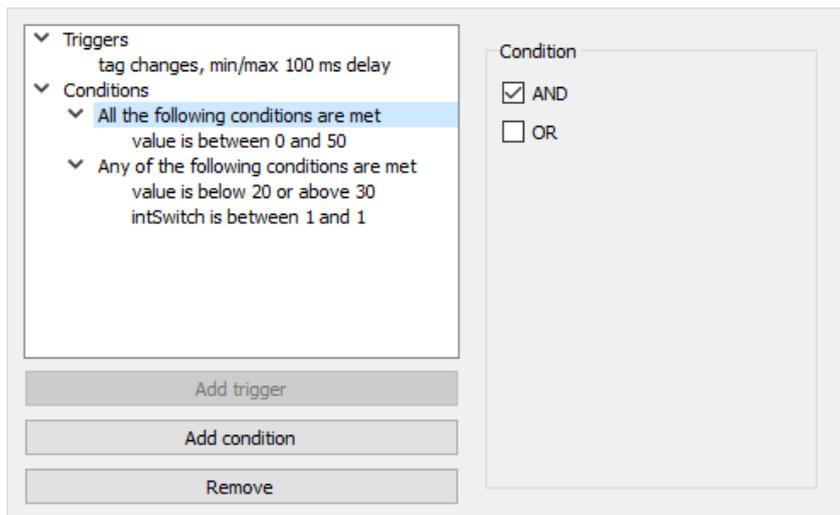
参数	说明
计时器	<p>即使值未更改，发布也会连续执行。</p> <ul style="list-style-type: none"> 间隔(毫秒) 周期性发布时间
On change	<p>标签值改变时执行发布。</p> <ul style="list-style-type: none"> 最小间隔(毫秒) 值检查间隔 Deadband 必须找到与先前发布的差异才能触发新发布。 使用百分比 死区值以百分比表示 标签名称 激活发布时要检查的标签。如果为空，则使用要发布的标签。

条件

条件包含条件文件夹。每个文件夹可以有两种类型：

- 满足以下所有条件(与)
- 满足以下任何条件(或)

必须验证所有文件夹，使触发器请求传输。当所有包含的条件都为 **true** 时，将验证类型“满足所有以下条件”的文件夹，而当至少一个包含的条件为 **true** 时，将验证“满足以下条件的任何一个”类型的文件夹。



主题和负载

支持的主题有五种：

主题	说明
出生	这是一个特殊主题，仅在 HMI 设备启动时发布一次。
将	这是一个特殊主题，在启动 HMI 设备时会发布，但会从 MQTT 代理中隐藏并存储。如果它检测到客户端异常断开连接，它将从 MQTT Broker 发布。
数据 (Pub)	主题用于遵循与标签组关联的传输策略发布标签的值。
数据 (子)	主题用于订阅标签。负载是用于识别接收到的标签的值的模板。
报警	用于发布报警的主题

对于每个主题，负载都定义了关联值的结构。请注意，在主题和有效负载定义中可以使用占位符。

占位符	说明
\$(clientId)	MQTT 客户端 ID
\$(protocolName)	与标签关联的协议名称
\$(tagGroup)	标签所属组的名称
\$(tagName)	标签名称
\$(alarmGroup)	报警所属的组名

占位符	说明
<code>\${alarmName}</code>	报警名称
<code>\${timestamp}</code>	当前时间
<code>\${value}</code>	标签的最新已知值
<code>\${activeValue}</code>	报警活跃时, 标签的值
<code>\${quality}</code>	标签的质量(如, 可靠性)
<code>\${activeTimestamp}</code>	最新一次事件的时间戳引发报警
<code>\${inactiveTimestamp}</code>	最新一次事件的时间戳停止报警条件
<code>\${ackTimestamp}</code>	操作员确认报警后的时间戳
<code>\${description}</code>	报警说明
<code>\${customField1}</code>	报警自定义字段 1
<code>\${customField2}</code>	报警自定义字段 2
<code>\${state}</code>	报警状态
<code>\${severity}</code>	报警严重性
<code>\${lowLimit}</code>	报警“下值”
<code>\${highLimit}</code>	报警“上值”
<code>\${[0]}</code>	如果在报警说明中可用, 则第一个在线标签的值, [1]第二个依此类推。
<code>\${[Tag1]}</code>	如果在报警说明中可用, 则“Tag1”在线标签的值

MQTT 代理设置

代理选择和配置参数。请注意, 某些参数取决于已选择的代理。

参数	说明
代理	代理使用 <ul style="list-style-type: none"> • 通用 MQTT 代理 • Azure • 亚马逊AWS • IBM BlueMix • Murano
待处理信息上限	未与代理连接时, 可以在 RAM 中队列的消息数。MQTT 服务器恢复可达时, 将释放队列消息。
最大缓冲区大小	在 MQTT 服务器确认接收之前, 文件系统内为以 QoS 标记大于 0 的主题存储的空间, 以便在断电时不会丢失。

通用 MQTT 代理

参数	说明
代理地址	MQTT 服务器的名称或 IP 地址
端口	MQTT 服务器端口。通常，默认的 TCP/IP 端口为 1883，或者使用基于 SSL 的 MQTT 时端口为 8883。
客户端 ID	客户端标识符是连接到 MQTT 代理的每个 MQTT 客户端的标识符。您可以写入自己喜欢的内容，但每个代理必须是唯一的。代理用它来识别客户端和客户端的当前状态。如果您不需要代理保留状态，则在 MQTT 3.1.1(当前标准)中，也可以发送一个空的客户端 ID，可导致连接没有任何状态。在这种情况下，必须选中“Use clean session”复选框，否则，连接将被拒绝。
用户名 密码	如果 MQTT 代理配置要求使用有效的用户名和密码进行客户端身份验证
保持活跃时间(秒)	没有数据流时，向服务器发送 PING 请求之前的时间间隔(有助于了解客户端和服务器是否仍处于活动状态并且可访问)。
使用 clean session	当 clean session 设置为 false 时，代理将为客户端创建一个持久会话。所有信息和消息都将保留，直到客户端下次请求 clean session 为止。如果将 clean session 标志设置为 false，并且代理已经有可用于客户端的会话，则代理将使用现有会话并将先前排队的消息传递给客户端。
使用旧版	“使用旧版”标志使客户端符合 MQTT 规范 3.1

启用 TLS

如果将 MQTT 服务器配置为通过 TLS 连接工作，则 HMI 设备必须向服务器提供其自己的证书。即使不是强制性的，每个客户端也应该有自己的证书(但是可以将相同的证书部署到所有客户端)。

要求服务器和客户端证书均由同一机构签署。

参数	说明
启用 TLS	启用 TLS 加密
CA 证书	在 Mosquitto 代理上签署服务器证书的 CA 公共证书
客户端证书	HMI 设备的公共证书。必须从 CA 证书上签署
客户端私钥	与客户端证书关联的私钥
TLS 版本	要使用的 TLS 版本(必须与 MQTT 代理中使用的加密级别保持一致) <ul style="list-style-type: none"> • tlsv 1 • tlsv 1.1 • tlsv 1.2
不安全	改选项禁用服务器证书上的服务器主机名称验证当测试初始化服务器配置时可选用该选项，但是它可能导致恶意第三方通过 DNS 诱骗模拟您的服务器。仅在测试中使用该选项。

参数	说明
QoS	用于报警、出生和 Last Will 信息的优质服务器值
保留	用于报警、出生和 Last Will 信息的保留值

MQTT 代理示例

以下示例阐述如何配置应用程序，使之与 MQTT 服务器通信。在该示例中，我们配置 HMI 设备采用证书与开放源 MQTT 代理 (<https://mosquitto.org>)通信。采用公共开放 SSL-Win32 库 (<https://www.openssl.org>)创建证书。

生成证书

以下脚本文件将创建几个服务器和客户端证书，公共证书机构将签署服务器并验证证书的权威性。

- ca.crt
- server.crt, server.key
- client.crt, client.ket

文件: *CreateCertificates.cmd*

```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"

rem Generate self signed CA certificate (Certificate Authority)
%OpenSSL% req -nodes -batch -new -x509 -days 1000 -keyout ca.key -subj
"/CN=Broker/O=company.com" -out ca.crt

rem Generate MQTT Server private key
%OpenSSL% genrsa -out server.key 2048

rem Generate MQTT Server certificate signed request
%OpenSSL% req -batch -new -key server.key -subj "/CN=localhost/O=company.com" -out
server.csr

rem Sign the MQTT Server certificate
%OpenSSL% x509 -req -days 1000 -in server.csr -CA ca.crt -CAkey ca.key -
CAcreateserial -out server.crt

rem Generate HMI Client private key
%OpenSSL% genrsa -out client.key 2048

rem Generate HMI Client Server certificate signed request
%OpenSSL% req -batch -new -key client.key -subj "/CN=client/O=company.com" -out
client.csr

rem Sign the HMI Client certificate
%OpenSSL% x509 -req -days 1000 -in client.csr -CA ca.crt -CAkey ca.key -
CAcreateserial -out client.crt
```

```
rem Remove unnecessary files
del *.rnd *.srl *.csr

pause
```

请注意，服务器主机名是 `localhost(/ CN = localhost)`，这意味着如果在代理地址参数中无法写入“localhost”域，则无法使用安全连接。仅当 MQTT 服务器和 HMI 设备都在同一设备上运行时，才可以使用“localhost”域；否则，为了能够访问 MQTT 服务器，必须使用 IP 地址和“insecure”标志。

MQTT 代理配置

服务器证书 (`server.crt`, `server.key`) 和授权证书 (`ca.crt`) 必须放置在 MQTT 文件夹的子文件夹中，例如在“certs”子文件夹中。

必须将“`mosquitto.conf`”文件配置为使用 TLS 支持

```
# ===== # Default listener
# ===== ...(omiss) ...# 用于默认倾听器的端口。#port 1883 port 8883 # -----
----- # Certificate based SSL/TLS support # -----
----- ...(omiss) ...#cafile #capath cafile
certs/ca.crt certfile certs/server.crt keyfile certs/server.key tls_version tlsv1.2
```

MQTT 代理可从 dos 命令窗口开始使用下列命令：

```
mosquitto -v -c mosquitto.conf
```

MQTT 客户端

为了进行测试，启动带有所有主题订阅的 MQTT 客户端可能会很有用，这样您就可以看到将与 HMI 设备交换的消息。由于我们正在使用 TLS 通信，因此我们必须提供客户端证书。我们可以在 `certs-client` 子文件夹内复制 `client.crt`, `client.key` 和授权证书 `ca.crt`。

因此，激活 MQTT 客户端的命令是：

```
mosquitto_sub --cafile certs-client\ca.crt --cert certs-client\client.crt --key certs-client\client.key -p 8883 -t /#
```

HMI 设备

要配置 HMI 设备，我们必须提供：

- 使用运行 MQTT 服务器的 IP 地址设置代理地址参数
- 将端口地址设置为 8883
- 加载授权证书、客户端证书和客户端密钥文件
- 将 TLS 版本设置为版本 1.2, 与 MQTT 服务器设置保持一致
- 由于可能是您使用 IP 地址引用 MQTT 服务器, 而该 IP 地址与服务器证书声明的域不同, 因此必须设置“不安全”标志

要执行第一个测试, 您可以保留主题和负载的默认值, 并配置要转移到 MQTT 代理的报警组和标签组。

MQTT Interface

Features

Enable MQTT interface

Enable alarms

Alarm groups: MQTT

Tags configuration

Default push policy: OnChange

Manage push policies

	Enable	Tag Group	QoS	Retain	Policy
1	<input type="checkbox"/>	All	0	<input type="checkbox"/>	
2	<input checked="" type="checkbox"/>	Counter	0	<input checked="" type="checkbox"/>	
3	<input checked="" type="checkbox"/>	MQTT	0	<input checked="" type="checkbox"/>	

36 特殊控件

为特殊目的设计的控件称作特殊控件，包括控制列表、日期和时间控件、变量控件等等。

BACnet 控件	388
浏览器控件	388
Canvas 控件	389
组合框控件	392
耗量表控件	396
控制列表控件	397
日期时间控件	401
手势控件	402
IP 摄像机控件	407
Javascript 功能块控件	410
多媒体播放器控件	412
多状态图像控件	414
多状态图像多层控件	415
网络适配器控件	417
RSS Feed 控件	417
滚动 RSS Feed 控件	418
表格控件	418
TextEditor 控件	434
变量控件	436

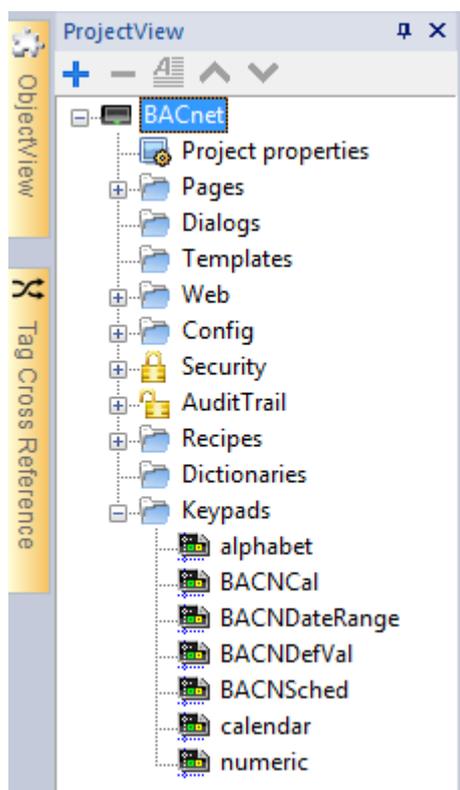
BACnet 控件

路径: 控件库 > BACnet

BACnet 控件是特殊控件，支持与本机 BACnet 对象交互。

- BACnet Calendar
- BACnet 计划任务
- BACnet 有效期

这些控件使用特殊键盘，在您使用控件时，它们将被添加到键盘文件夹中。通常情况下，您无需留意这些键盘，除非您要对其进行自定义。



请参阅“Communication Drivers”文件夹中的 BACnet 手册，了解 BACnet 特殊控件的详细描述。



Web 页面不支持 BACnet 控件。

浏览器控件

路径: 控件库 > Media > Web 控件

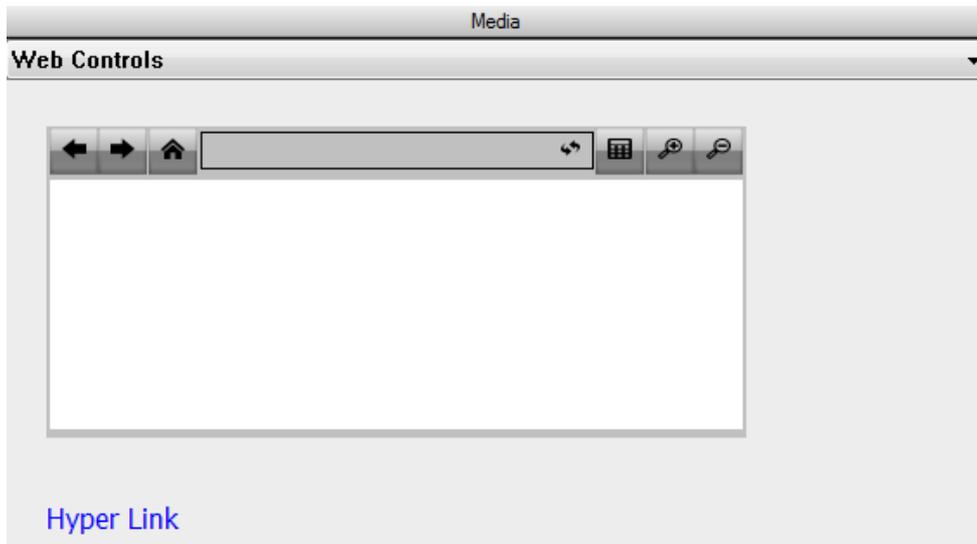
使用该控件将网页嵌入 HMI 设备页面中。这是一个基于 WebKit 引擎的与 HTML5 兼容的浏览器控件。



注: WebKit 库仅在需要时作为插件(请参阅“插件”在本页 79 了解详细信息)下载到 LRH SW HMI Runtime。



重要:基于 MIPS 的设备不支持此控件。



参数	说明
主页	当控件在页面显示时默认打开的 URL。
缩放至合适大小	按照视图区域大小，自动调整内容。
超时	页面加载超时秒数。
清除历史记录	加载时自动清除历史记录
滚动	显示/隐藏滚动条
显示进程光标	显示/隐藏加载光标

这样您可以在项目中不需要该控件时节省大约 3MB 的空间。

超链接 控件可用于创建页面超链接。一旦被点击，这些链接会通知浏览器控件，会有一个特殊的网页被加载。



重要:不支持 HTTP 协议。

Canvas 控件

路径: 控件库 > 基本 > 通用 Canvas

可以使用 Canvas 控件通过 JavaScript 脚本绘制图形。



注: JavaScript 方法同样适用于 HTML5 <canvas> 标签

参数	说明
Canvas 宽度 Canvas 高度	<p>Canvas 尺寸。</p> <p>请注意，这不是控件尺寸。例如，Canvas 尺寸可以是 500x500 像素，而控件尺寸可以是 100x100 像素。绘制提示参数将定义如何拉长 Canvas 尺寸来适应控件尺寸。</p>
绘制提示	<p>定义如何调整 Canvas 使其在控件尺寸内</p> <ul style="list-style-type: none"> • 剪裁 不应用转换，不缩放坐标系统，而是剪裁绘图，使其位于控件边界矩形内。 • 调整为合适大小 使其适应控件尺寸，保留 Canvas 模型的宽高比。 • 拉长 使其适应控件尺寸，忽略 Canvas 模型的宽高比。 <p>例如使用的 Canvas 尺寸比控件尺寸大：</p> <p>Canvas size: 400x400 Widget size: 100x200</p>
设计时间预览	<p>LRH SW 内的 Canvas 预览</p> <p> 请注意，JavaScript 代码使用的数据在 LRH SW 中不可用，仅在 HMI 设备中可用</p>
自动清除背景	<p>绘制 Canvas 前自动清除背景。停用时，已绘制的项将保留，无需从头重绘所有内容。</p>
OnDraw 动作	<p>在绘制页面时执行 OnDraw 事件。此事件必须与用于绘制 Canvas 图形的 JavaScript 代码相连接。</p>
OnMousePress 动作 OnMouseRelease 动作 OnMouseDown 动作	<p>鼠标事件</p>

Canvas 函数

// Painter Save/Restore

- void save(); // calls painter save
- void restore(); // calls painter restore

// Scale/Transform

- void scale(qreal x, qreal y);
- void rotate(qreal angle);
- void translate(qreal x, qreal y);
- void transform(qreal m11, qreal m12, qreal m21, qreal m22, qreal dx, qreal dy);
- void setTransform(qreal m11, qreal m12, qreal m21, qreal m22, qreal dx, qreal dy);

// Gradient

- CanvasGradient createLinearGradient(qreal x0, qreal y0, qreal x1, qreal y1);
- CanvasGradient createRadialGradient(qreal x0, qreal y0, qreal r0, qreal x1, qreal y1, qreal r1);

// Rectangle Functions

- void clearRect(qreal x, qreal y, qreal w, qreal h);
- void fillRect(qreal x, qreal y, qreal w, qreal h);
- void strokeRect(qreal x, qreal y, qreal w, qreal h);
- void rect(qreal x, qreal y, qreal w, qreal h);

// Path

- void beginPath();
- void closePath();
- void moveTo(qreal x, qreal y);
- void lineTo(qreal x, qreal y);
- void quadraticCurveTo(qreal cpx, qreal cpy, qreal x, qreal y);
- void bezierCurveTo(qreal cp1x, qreal cp1y, qreal cp2x, qreal cp2y, qreal x, qreal y);

// Drawing Text

- void fillText(const QString &text, qreal x, qreal y);

// Arc

- void arcTo(qreal x1, qreal y1, qreal x2, qreal y2, qreal radius);
- void arc(qreal x, qreal y, qreal radius, qreal startAngle, qreal endAngle, bool anticlockwise);

// Fill/Stroke

- void fill();
- void stroke();
- void clip();
- bool isPointInPath(qreal x, qreal y) const;

// Image manipulation (Draw QImageWgt using target and source rect)

- void drawImage(QObject *pObjImage, qreal sx, qreal sy, qreal sw, qreal sh, qreal dx, qreal dy, qreal dw, qreal dh);
- void drawImage(QObject *pObjImage, qreal dx, qreal dy);
- void drawImage(QObject *pObjImage, qreal dx, qreal dy, qreal dw, qreal dh);
- void drawImage(const QVariant& image, int width, int height, const QString& format, qreal sx, qreal sy, qreal sw, qreal sh, qreal dx, qreal dy, qreal dw, qreal dh);

// Pixel manipulation

- ImageData createlImageData(double sw, double sh); //Empty Image
- ImageData createlImageData(ImageData fromImage); //from another Image
- ImageData createlImageData(ArrayBuffer value); //From arraybuffer
- void putImageData(ImageData imgData, double dx, double dy);
- void putImageData(ImageData imagedata, double dx, double dy, double dirtyX, double dirtyY, double dirtyWidth, double dirtyHeight);
- ImageData getImageData(qreal sx, qreal sy, qreal sw, qreal sh);

Canvas JavaScript 示例

Canvas 最初是空白的。为显示内容，脚本需要先访问渲染上下文并在其上绘制：

```
var ctx = me.context2d;
```

然后您可以使用 **Canvas** 方法，如下例所示

```
function GenericCanvasWgt1_onDraw(me, eventInfo)
{
    var ctx = me.context2d;
    ctx.fillStyle = 'red';
    ctx.fillRect(0,0,250,250);
    ctx.fillStyle = 'green';
    ctx.fillRect(250,0,250,250);
    ctx.fillStyle = 'blue';
    ctx.fillRect(0,250,250,250);
    ctx.fillStyle = 'black';
    ctx.fillRect(250,250,250,250);
}

function GenericCanvasWgt1_onMouseDown(me, eventInfo)
{
    alert("X = " + eventInfo.posX + "\nY = " + eventInfo.posY );
}
```

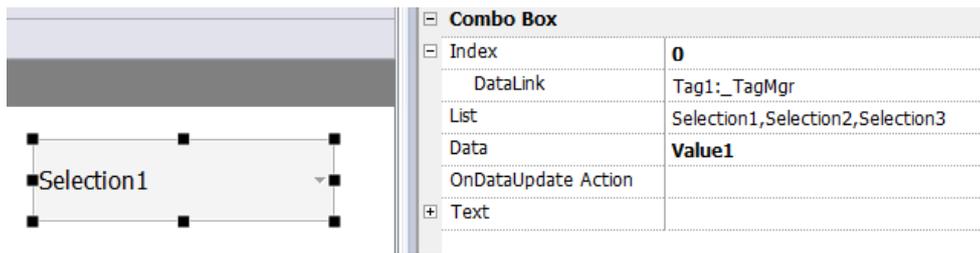
更新函数可用于动态重绘 **Canvas** 控件

```
function BtnStd1_btn_onMouseClicked(me, eventInfo)
{
    var myCanvasWidget = page.getWidget("GenericCanvasWgt1");
    myCanvasWidget.update();
}
```

组合框控件

路径: **控件库 > 基本 > 控件**

使用该控件作为选择器控件或筛选表格中的行，仅显示组合框中选择的值。



参数	说明
索引	所选项索引
列表/字符串行表	组合框中的项目字符串。  注:该字段支持多语言。
数据/数据列表	返回控件的数据字段中数据列表列(字符串)中的值。  提示:使用该参数返回基于组合框中所选项的自定义值。
文本	显示文本。

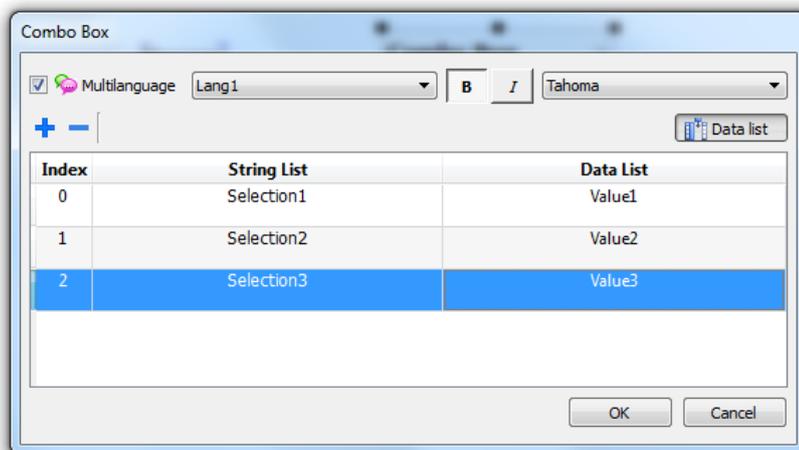
数据列表

数据列表与“listData”属性关联,可以使用 JavaScript 代码进行动态修改。

```
//读取数据列表
var comboWgt = page.getWidget(" Combo1");
var listData = comboWgt.getProperty(" listData")

//写入数据列表
var comboWgt = page.getWidget(" Combo1");
comboWgt.setProperty(" listData", " NewData1,NewData2,NewData3");
```

附加数据与附加索引标签



在很多项目中，您可能需要将**索引**或**数据**等字段附加到标签上，以了解组合框中所选项的值。使用：

- **索引**：显示所选项的索引(整数)(0...n)。
- **数据**：显示在“数据列表”列中指定的数据值(字符串)。

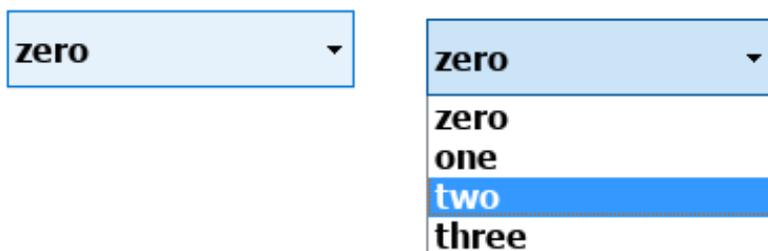
组合框控件“全屏”模式(支持图片)

从“项目属性”在本页73中，可以将组合框的外观和行为从“上下文”模式切换到“全屏”模式

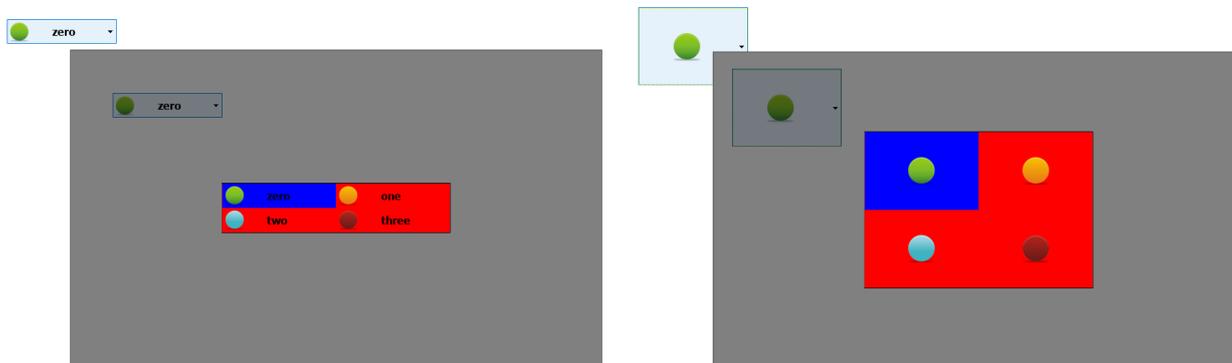
路径：项目视图 > 双击项目属性 > 项目窗格 > 项目 > 组合框视图模式

参数	说明
组合框视图模式	<p>选择项目所有组合框控件的可视化模式</p> <p>上下文菜单 经典视图，具有下拉菜单</p> <p>全屏 增强视图，具有可配置文本和图像，它们将在屏幕中间弹出，以便滚动和选择。</p>

上下文视图示例

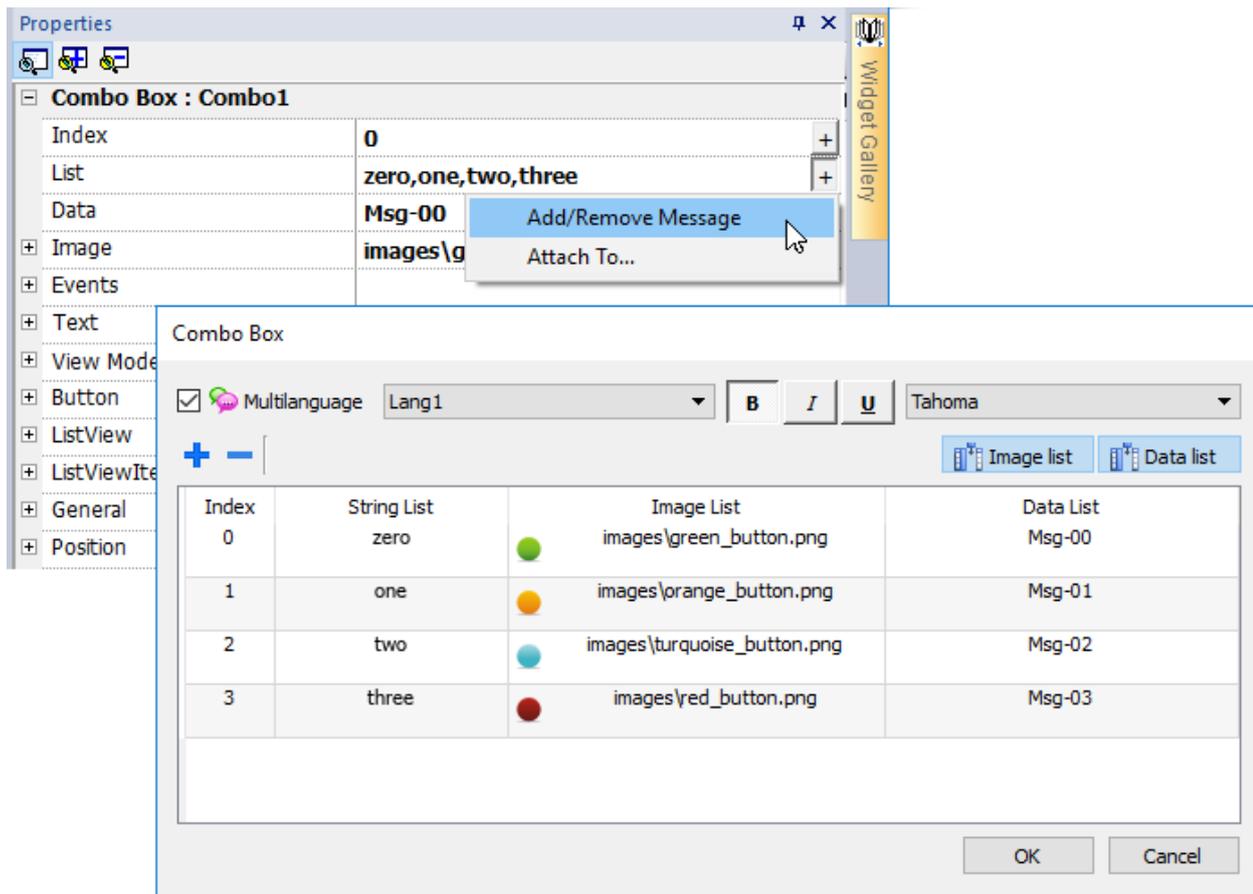


全屏视图示例



在全屏模式下可获得的更多参数

系统将在**组合框 > 列表参数**中附加“图片列表”列：



注：一些属性仅在高级模式下显示。

参数	说明
图片	返回所选图片的文件名，位于附加标签内
按钮	定义组合框的外观 <ul style="list-style-type: none"> 显示背景 = true 显示组合框按钮 显示背景 = false 仅显示图像或文本
列表视图	处于编辑模式的组合框布局参数
列表视图项	定义将位于组合框内的项目类型 图像模式： <ul style="list-style-type: none"> 仅文本 仅图片 文本和图片

耗量表控件

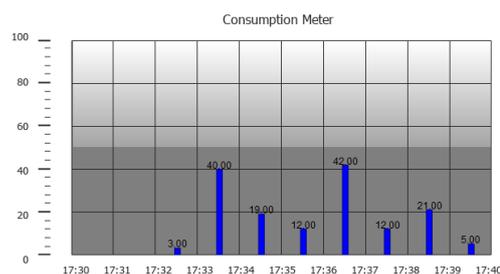
路径: 控件库 > 基本 > 趋势/图形

使用该控件监控持续增加的资源。系统读取资源的值，并计算增加的时间范围，然后增加显示在类似趋势窗口的条形图中。

可以用不同的颜色根据时间框在图表中显示。



提示: 使用该控件计算系统的能耗。



参数	说明
值	监控的资源
图持续时间 图形持续时间单位	窗口中显示的时间周期
柱时间 柱持续时间单位	图表中每条代表的时间周期
时间周期	指定一种特定的颜色来突出在指定时间周期内监控资源的增量。(最低决定=1小时)。
颜色 柱宽度	条形图颜色和宽度
条形图值	显示/隐藏各条形图的值
耗量表	要在图表中显示的标签数量。

示例: 如何监控能量消耗

在下面的例子中，设计了一个控件，监控每周和每天的能量消耗。

1. 将一个标签附在物理变量上进行监控。在示例中，对于总的能量消耗(标签 KWh)。该标签包含增长量，表明从能量消耗开始每小时消耗了多少 KM。
2. 添加一个趋势，将其链接到要监控的标签，标签 KWh。
3. 将耗量表控件添加到页面中。
4. 将耗电表的值属性添加到您在第 2 步创建的趋势中。
5. 将图形持续时间/单位设为 1 周：这将提供一个每周能耗表。
6. 将柱形图持续时间/单位设为 1 天，这是能耗计算的时间范围。
7. 在耗量表中设置要显示在条形图中的标签数量，在本例中，7 表示是一个每周图表。
8. 从时间段属性中，打开配置时间周期对话框：为各个条形图中标签 KWh 的不同值设置不同的颜色。



提示：要为表格中的单元指定颜色，请选择该单元，单击需要的颜色，或者将该条的索引值 (1、2、3) 输入该单元。

9. 按需要添加尽可能多的颜色条，在该示例中为 3 个颜色条。
10. 在每周图表中对每小时添加一个条，该示例中，使用了一个红色的条(E1)表明最高能量消耗的日期/周的时间范围。



注：如果需要，您可以对每个颜色条应用一个缩放系数。

该结果是一个条形图表消耗量，以 KM/h 为单位小时每日能量消耗，颜色表明不同能量消耗。每一个条形的高度代表考虑的时间范围的能量消耗，该示例中是 1 天。

使用滚动仪表页面 (ConsumptionMeterPageScroll) 动作，将条形图前后滚动，使用刷新趋势 (RefreshTrend) 动作来刷新条形图表，因为数据不会自动刷新。



重要：当前“耗量表”控件不支持其他趋势动作。

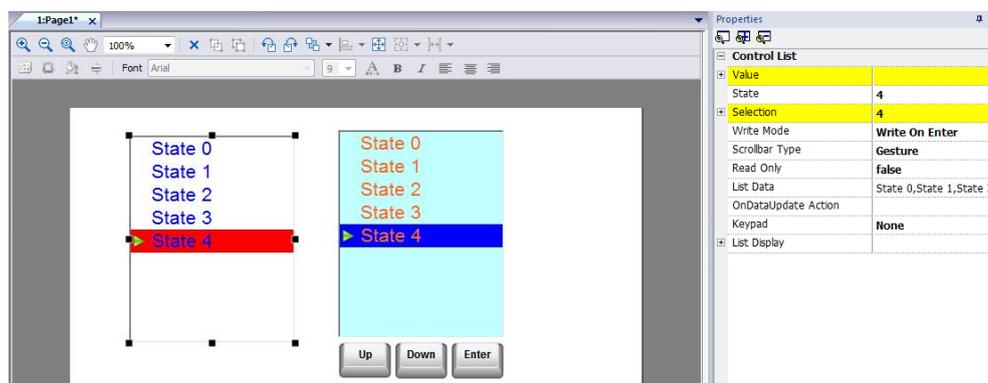
控制列表控件

路径：**控件库 > 高级 > 控制列表**

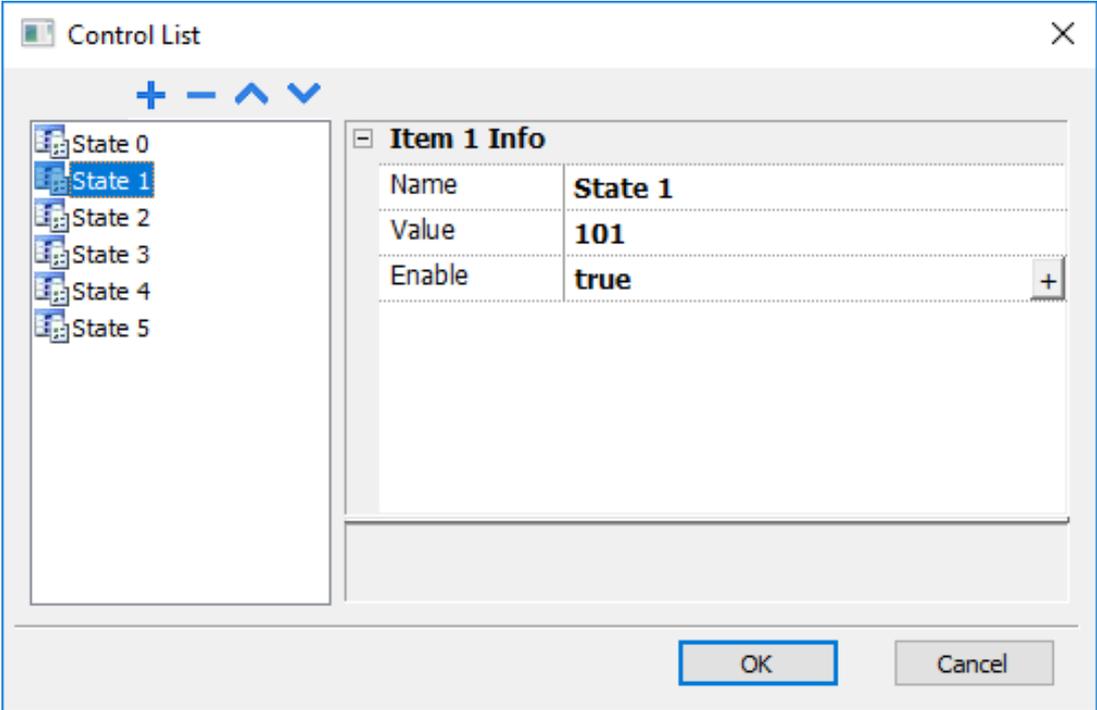
使用这些控件代表与特定过程相关的状态，并控制相应过程。

有两种控制列表可用：

- 组控制列表，包括导航按钮
- 基本控制列表，没有导航按钮。



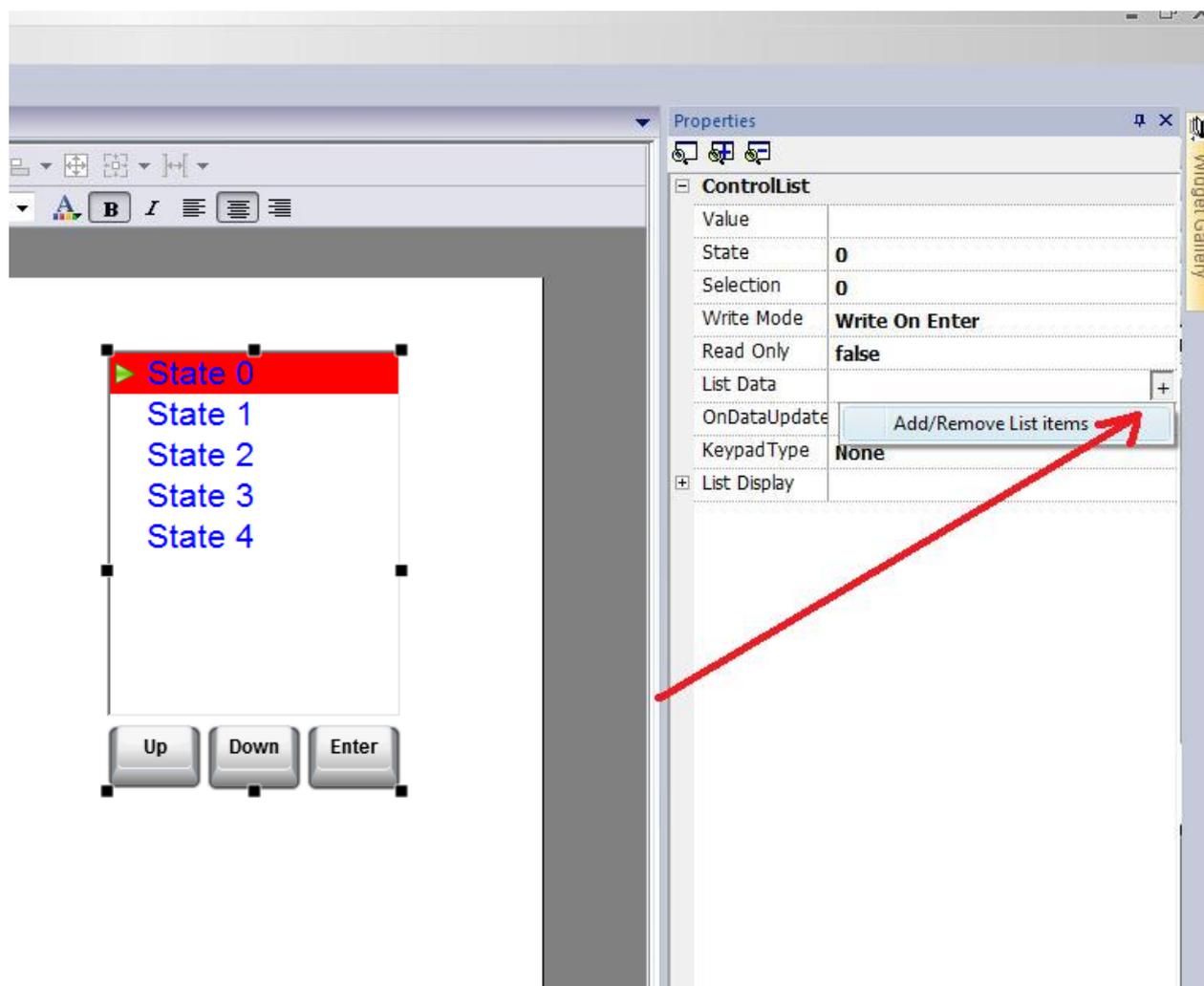
参数	说明
值	与控件的状态相对应的值。如果在值属性附加标签，则在加载控件时，状态将与标签值对齐。
状态	控件状态。控件以不同的背景颜色突出显示与其状态相关的项(请参阅控件属性中的“状态颜色”)。
选择	状态选择。所选项将在列表的左侧凸显一个小三角形。
写入模式	选择状态更新模式 <ul style="list-style-type: none"> • 选择时写入:状态将自动更新，与光标位置对齐。 • 键入时写入:仅当用户按下 Enter 键时，状态才会随光标位置更新
滚动条类型	选择表格的滚动模式 <ul style="list-style-type: none"> • 手势:平移手势可用于平滑滚动表格。 • 滚动条:使用滚动条滚动表格

参数	说明
只读	定义该列表是否仅是指示器。
列表数据	<p>状态项列表。每个项都有一个状态名称，一个相应的值和一个标志，该标志使项显示在控件内。</p> 

定义状态

从**列表数据**属性添加/删除状态，即列表中的项目。

一个状态可以分配一个值。当您在选择时写入相关项(如果处于**选择时写入**模式)或通过按 **Enter** 键选择并确认项目(如果处于**按 Enter 键时写入**模式)，则会将指定给该状态的值写入到与“控制列表”控件**值**相链接的标签。



通过 JavaScript 代码管理列表数据项

可以在运行时使用 `setProperty("listData", <NewControlList>)` 从 JavaScript 代码修改数据项列表。以下示例显示了如何修改项的列表

```
function SetItemsList_btn_onMouseClicked(me, eventInfo)
{
    var NewControlList = [{"OFF",100,true}, {"ON",101,true}, {"MAN",102,true}, {"AUTO",103,true}];
    var ControlListWgt = page.getWidget("controlListBtn.controlList");
    ControlListWgt.setProperty("listData", NewControlList);
}
```

其中

- `NewControlList` 是带有项说明的数组
- `controlListBtn.controlList` 是要修改的控制列表控件的 ID

而 `getProperty("listData")` 仅返回用逗号分隔的仅包含名称的字符串。

```
function Read_btn_onMouseClicked(me, eventInfo)
{
    var ControlListWgt = page.getWidget("controlListBtn.controlList");
    var ListData = ControlListWgt.getProperty("listData");
}

```

此处 ListData 的结果将为:"OFF,ON,MAN,AUTO"

状态

getProperty("state") 可用于检索 S 状态值。下列是 JavaScript 代码的示例

```
function controlListBtn_onDataUpdate(me, eventInfo)
{
    var ControlListWgt = page.getWidget("controlListBtn.controlList");
    var State = ControlListWgt.getProperty("state");
    project.setTag("State", State);
    return false;
}

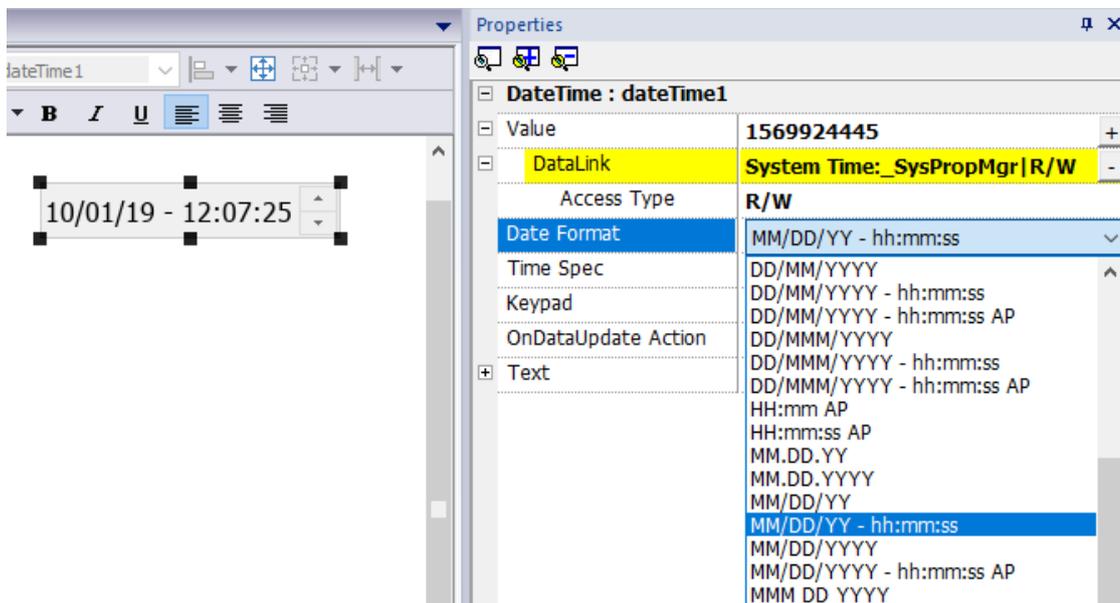
```

日期时间控件

路径: **控件库 > 基本 > 控件**

使用该控件显示和编辑当前日期和时间。

属性窗格中提供有不同的格式来表示日期和时间。



时间选项

对于时间格式属性, 选择该控件运行时显示的时间样式。

选项	说明
本地	显示当地时间，项目所运行的HMI设备的时间
全球	显示全球时间 (GMT)
服务器	显示 HMI 设备服务器这边处理的时间信息

时间和日期占位符

您可以使用占位符自由定义时间和日期格式

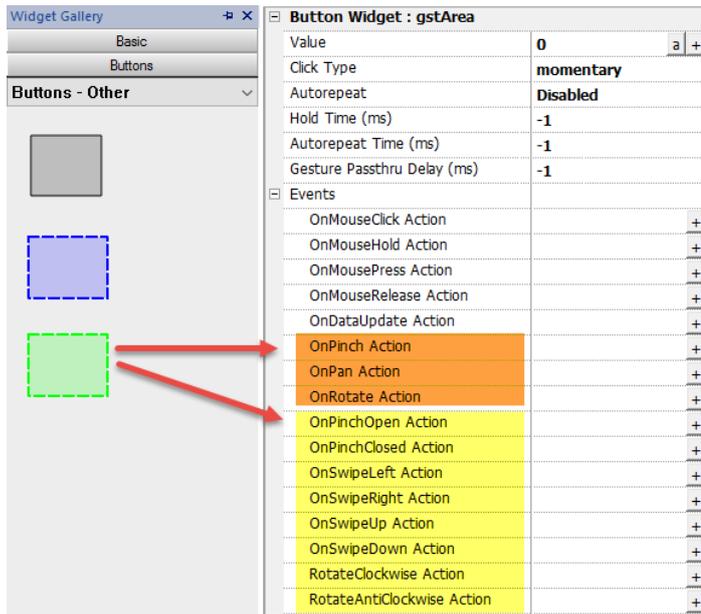
日期	说明
d	不带前导零的天数(1至31)
dd	带前导零的天数(01至31)
ddd	缩写的本地化日期名称(例如，“周一”至“周日”)
dddd	较长的本地化日期名称(例如，“周一”至“周日”)
M	不带前导零的月份数(1-12)
MM	带前导零的月份数(01-12)
MMM	缩写的本地化月份名称(例如，“一月”至“十二月”)
MMMM	较长的本地化月份名称(例如，“一月”至“十二月”)
yy	以两位数表示的年份(00-99)
yyyy	以四位数表示的年份

时间	说明
h	不带前导零的小时数(如果显示上午/下午，0至23或1至12)
hh	带前导零的小时数(如果显示上午/下午，00至23或01至12)
m	不带前导零的分钟数(0至59)
mm	带前导零的分钟数(00至59)
s	不带前导零的秒数(0至59)
ss	带前导零的秒数(00至59)
AP 或 A	使用上午/下午显示。A/AP 将替换为“上午”或“下午”
ap 或 a	使用上午/下午显示。a/ap 将替换为“上午”或“下午”

手势控件

路径: 控件库 > 按钮 > 其他

手势控件是一个热键，通过其可生成手势事件。



手势事件	说明
OnSwipeLeft OnSwipeRight	检测到滑动手势时会释放事件
OnPinchOpen OnPinchClose	检测到缩放手势时会释放事件
顺时针旋转 逆时针旋转	检测到旋转手势时会释放事件
OnPan OnPinch OnRotate	<p>在手势期间释放的一系列事件。</p> <p>尽管开发人员可以通过 JavaScript 代码管理其喜欢的手势事件，但是仅 JavaScript 可用于服务这些事件。</p> <p> 警告：仅多点触摸 HMI 设备可生成 OnPinch 和 OnRotate 事件</p>

OnPan

boolean onGesturePan(me, eventInfo)

本事件会在按下该区域内一点时并检测到线性移动时发生。

参数	说明
me	启动事件的对象。
eventInfo	<p>ID = 手势 ID; 用于识别不同手势。</p> <p>running = 除用于通知手势完成的最后事件外均为 true。</p> <p>dx = 从初始触摸位置以屏幕像素单位 X 轴移动总和。</p> <p>dy = 从初始触摸位置以屏幕像素单位 Y 轴移动总和。</p>

OnPinch

boolean onGesturePinch(me, eventInfo)

本事件会在按下该区域内两点时并检测到线性移动时发生。

参数	说明
me	对象启动事件
eventInfo	<p>ID = 手势 ID; 用于识别不同手势。</p> <p>running = 除用于通知手势完成的最后事件外均为 true。</p> <p>dx = 从初始触摸位置以屏幕像素单位 X 轴移动总和。它代表手指之间的距离变化。正值代表正在增大的距离; 负值代表正在减少的距离。该量可用于控制缩放值。</p> <p>dy = 从初始触摸位置以屏幕像素单位 Y 轴移动总和(见 dx)。</p>

OnRotate

boolean onGestureRotate(me, eventInfo)

本事件会在按下该区域内两点时并检测到旋转移动时发生。

参数	说明
me	对象启动事件
eventInfo	<p>ID = 手势 ID; 用于识别不同手势。</p> <p>running = 除用于通知手势完成的最后事件外均为 true。</p> <p>drot = 自上一个事件起已添加多少度 (0/360)。</p> <p>trot = 整个移动的总度数 (0/360)。</p> <p>整数表示顺时针旋转, 负数表示逆时针旋转。</p>

手势事件密码

要使用手势对象中覆盖的控件(例如, 按钮或滑块), 必须按住该控件 **200 mSec** 以便将该控件移至底层对象。可通过高级属性视图中可用的“手势密码延迟”参数修改将命令发送到底层对象所需等待的时间。

参数	说明
手势密码已启用	<p>启用在可配置延迟后将手势事件传递给底层控件的可能性。用户必须按住手指，然后执行手势。</p> <p>default = 使用项目属性中定义的值。请参阅"项目"在本页 80</p> <p>true = 手势密码已启用</p> <p>false = 手势密码已停用</p>
手势密码延迟 (ms)	<p>将命令发送到底层对象所需等待的时间</p> <p>0/500 mSec</p> <p>-1 使用项目属性中定义的延迟。请参阅"项目"在本页 80</p>

使用与 JavaScript 相关联的手势事件的示例

以下是一些使用与 JavaScript 代码相关联的手势事件来识别手势和编程请求动作的示例

滑动手势

如何识别“滑动”手势，以更换应用页面。

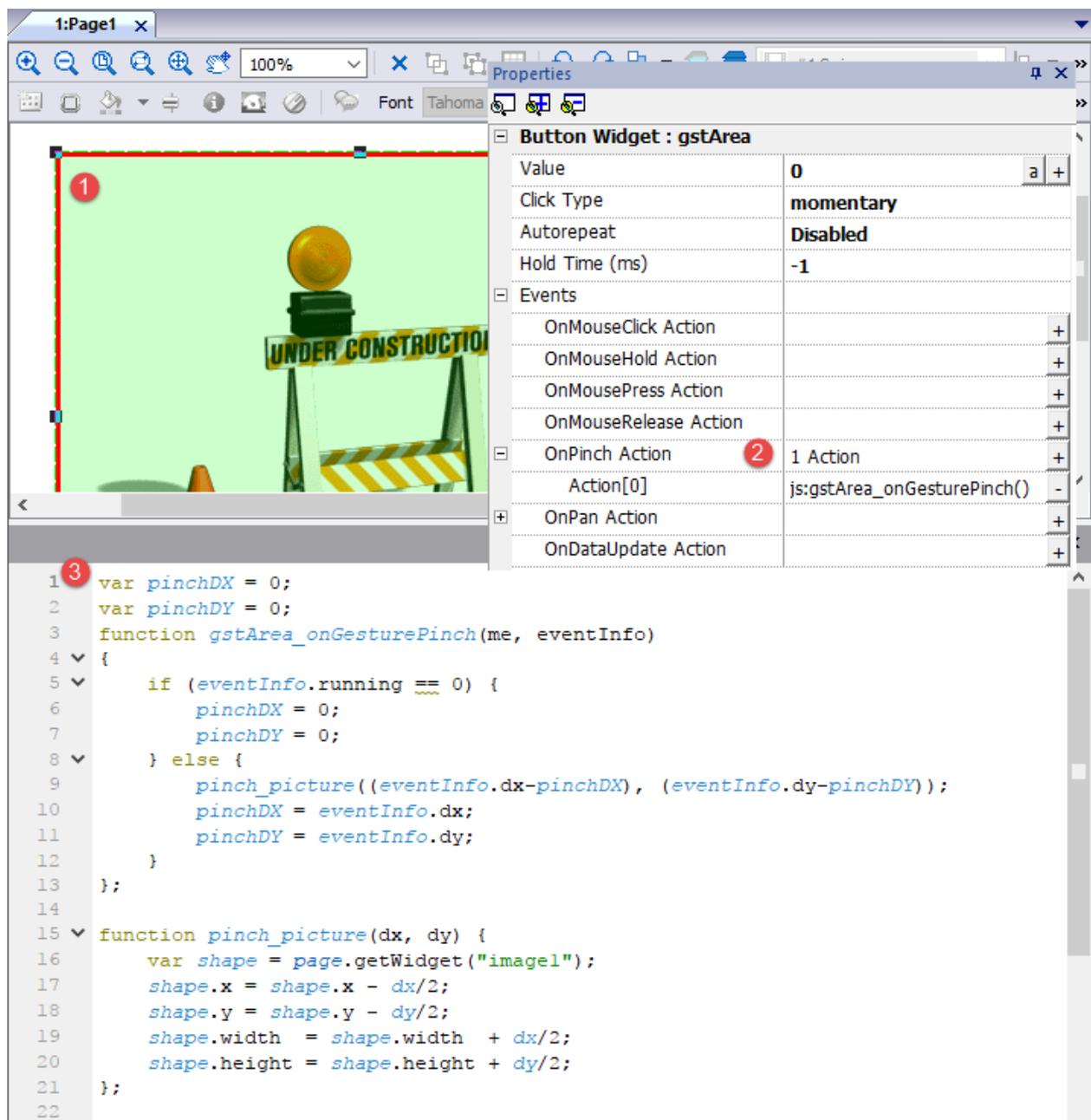
1. 将手势控件放入页面中
2. 配置 OnPan 动作以触发 JavaScript 函数
3. 编写可识别和管理滑动手势的 JavaScript 代码

The screenshot displays a development environment with a visual editor on the left and a properties panel on the right. In the visual editor, a green rectangular widget is placed on a page, with a red circle '1' indicating its position. Below the editor, a JavaScript code block is shown with a red circle '3' next to the function definition. The code defines a function `Swipe_onGesturePan` that checks the direction of a pan gesture and calls `project.nextPage()` or `project.prevPage()` accordingly. The properties panel on the right shows the configuration for the `Button Widget : Swipe`. It includes fields for `Value` (0), `Click Type` (momentary), `Autorepeat` (Disabled), `Hold Time (ms)` (-1), and `Autorepeat Time (ms)` (-1). Under the `Events` section, the `OnPan Action` is configured with a red circle '2' next to it, and its `Action[0]` is set to `js:Swipe_onGesturePan()`.

缩放手势

如何识别“缩放”手势，以调整图片大小。

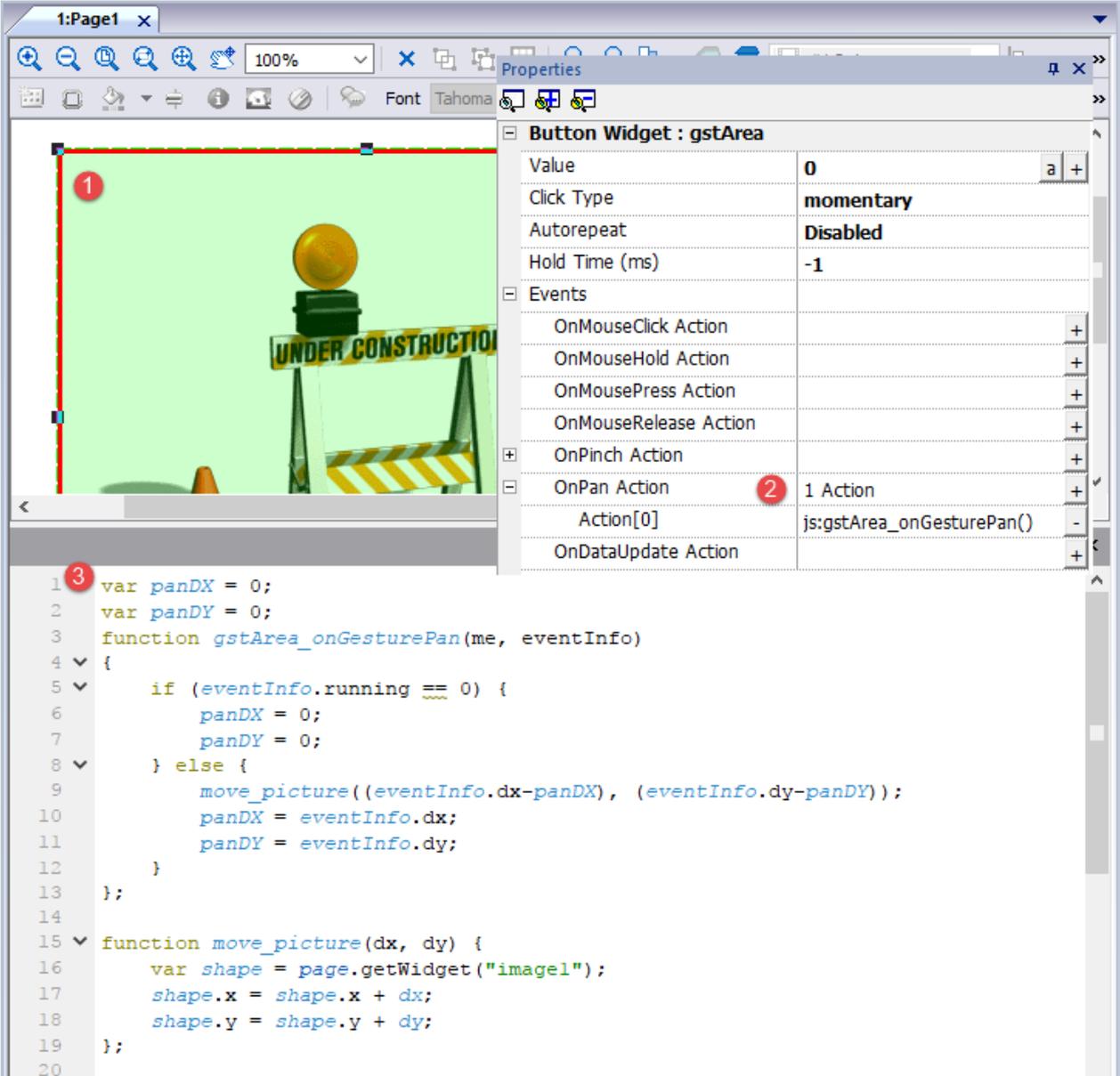
1. 将手势控件放入图片上方的页面中
2. 配置 OnPinch 动作以触发 JavaScript 函数
3. 编写可识别和管理缩放手势的 JavaScript 代码



平移手势

如何识别“平移”手势，以移动图片。

1. 将手势控件放入图片上方的页面中
2. 配置 OnPan 动作以触发 JavaScript 函数
3. 编写可识别和管理平移手势的 JavaScript 代码



Button Widget : gstArea

Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Events	
OnClick Action	+
OnMouseHold Action	+
OnMousePress Action	+
OnMouseRelease Action	+
OnPinch Action	+
OnPan Action	1 Action
Action[0]	js:gstArea_onGesturePan()
OnDataUpdate Action	+

```

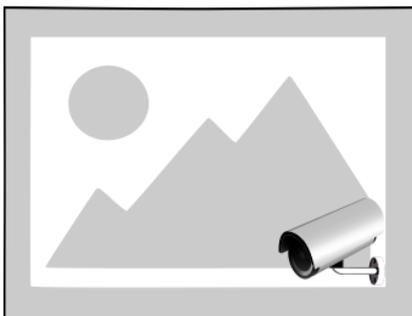
1 3 var panDX = 0;
2  var panDY = 0;
3  function gstArea_onGesturePan(me, eventInfo)
4  {
5    if (eventInfo.running == 0) {
6      panDX = 0;
7      panDY = 0;
8    } else {
9      move_picture((eventInfo.dx-panDX), (eventInfo.dy-panDY));
10     panDX = eventInfo.dx;
11     panDY = eventInfo.dy;
12   }
13 };
14
15 function move_picture(dx, dy) {
16   var shape = page.getWidget("image1");
17   shape.x = shape.x + dx;
18   shape.y = shape.y + dy;
19 };
20

```

IP 摄像机控件

路径: 控件库 > Media > IP 摄像机

使用这些控件显示用 IP 摄像机或视频流捕捉的图像。



参数	说明
摄像机 URL	使用 JPEG 格式的 IP 摄像机的 URL。
刷新率	一秒内 JPEG 图像的数量。最大=1 fps
用户名	允许访问摄像机的用户名 当访问摄像机受密码保护时，查看此参数。
密码	访问摄像机的密码。
MJPEG 相机 URL	MJPEG 流的 URL(例如，http://192.168.0.1/video.cgi)

当使用该控件传输 HTTP MJPEG 时，**摄像机 URL** 和 **刷新率** 将被忽略。

传输的性能不固定，取决于很多因素，如：照片大小、照片压缩程度，HMI设备的CPU，IP摄像机的质量。根据这些因素，可达到 25fps。

您可以添加多个 IP 摄像机控件，但这会降低每个控件的帧速。

支持的 IP 摄像机

以下 IP 摄像机目前已经过测试：

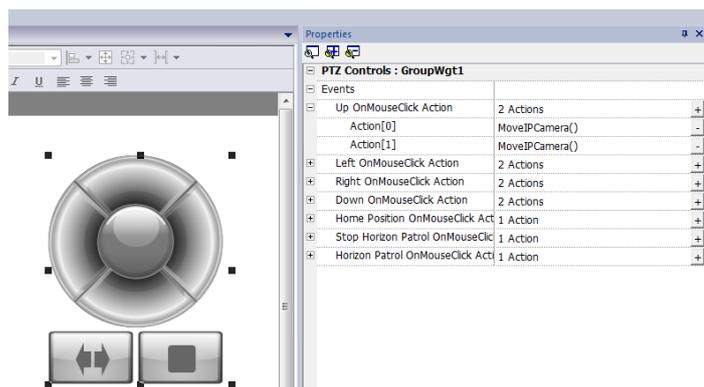
IP 摄像机	协议	URL
Apexis APM-J901-Z-WS PTZ IP 摄像机	MJPEG	http://{ip_address}/videostream.cgi
	HTTP	http://{ip_address}/snapshot.cgi
AXIS M3027-PVE 网络摄像机	MJPEG	http://{ip_address}/axis-cgi/mjpg/video.cgi
	HTTP	http://{ip_address}/axis-cgi/jpg/image.cgi
DAHUA DH-IPC-HD2100P-080B 1.3mp 户外防破坏	HTTP	http://{ip_address}:9988/onvif/media_service/snapshot
D-Link DCS-5605 PTZ	MJPEG	http://{ip_address}/video/mjpg.cgi
D-Link DCS-900W IP 摄像机	MJPEG	http://{ip_address}/video.cgi
D-Link DCS-932L	MJPEG	http://{ip_address}/video.cgi
Edimax IC-7100P PTZ	MJPEG	http://{ip_address}/mjpg/video.mjpg
	HTTP	http://{ip_address}/picture.jpg
Foscam FI8916W	MJPEG	http://{ip_address}/videostream.cgi
	HTTP	http://{ip_address}/snapshot.cgi
Foscam FI9803 EP	MJPEG	http://{ip_address}:88/cgi-bin/CGIStream.cgi?cmd=GetMJStream&usr={user}&pwd={pass}

IP 摄像机	协议	URL
		注： <ul style="list-style-type: none"> • 根据 IP 摄像头设置，端口 88 可能不同 • {user} = IP 摄像机设置中定义的用户名 • {pass} = IP 摄像机设置中定义的密码
Hamlet HNIPCAM IP 摄像机	MJPEG HTTP	http://{ip_address}/video.cgi http://{ip_address}/image.jpg
MOXA VPort 254 (耐用 4 通道 MJPEG/MPEG4 工业视频编码器)	MJPEG HTTP	http://{ip_address}/moxa-cgi/mjpeg.cgi http://{ip_address}/moxa-cgi/getSnapShot.cgi?chindex=1
NVS30 网络视频服务器	MJPEG HTTP	http://{ip_address}:8070/video.mjpeg http://{ip_address}/jpg/image.jpg
Panasonic WV-系列网络摄像机	MJPEG	http://{ip_address}/cgi-bin/mjpeg
Ubiquiti UniFi 视频摄像机	HTTP	http://{ip_address}:7080/images/snapshot/camera/{camera_guid}?force=true 注： <ul style="list-style-type: none"> • {camera_guid} 可在 IP 摄像机网页中找到 • 根据 IP 摄像头设置，端口 7080 可能不同
Zavio F3210 2MP 昼夜两用紧凑 IP 摄像机	MJPEG HTTP	http://{ip_address}/stream?uri=video.pro3 http://{ip_address}/cgi-bin/view/image?pro_0 注： <ul style="list-style-type: none"> • MJPEG 视频流可以配置为选择 IP 摄像机设置中的“视频配置文件 3”和 640x480 分辨率。

PTZ 控制控件

PTZ (平面-倾斜-移动) 摄像机是能够远程控制方向和大小的摄像机。

“PTZ 控制”控件使用 MoveIPCamera 动作向 PTZ IP 摄像机发送 HTTP/cgi 命令。



参数	说明
摄像机 URL	IP 摄像机的 URL
用户名	允许访问摄像机的用户名 当访问摄像机受密码保护时，查看此参数。
密码	访问摄像机的密码。
命令	向 PTZ 控制器发送的命令 (例如 <code>decoder_control.cgi?command=0</code>)

鉴定方法

鉴定方法自动由控件连接的摄像机网络服务器设定。支持的鉴定方法：

- 基本
- NTLM 版本 1
- Digest-MD5

Web 浏览器

在 Web 浏览器上，仅支持“基本身份验证”模式。使用时，支持身份验证的 IP 摄像机会显示一个弹出对话框，以便输入登录名和密码。

Chrome 和 Firefox 支持控件，我们在使用当前版本的 Edge 浏览器时发现了问题。

Javascript 功能块控件

路径：**控件库 > 基本 > JSFunctionBlock**

Javascript 功能块是包含 Javascript 逻辑的控件，在标签值发生变化时执行。

参数	说明
value1 ... value16	触发 OnDataUpdate 动作的对象。
OnDataUpdate	会在检测到相关值变化时执行的动作



注:该控件仅在 LRH SW 中呈现,不在 HMI 设备中呈现。

示例:

用于检查三个选择器的组合,其 Javascript 代码为

The screenshot shows a development environment with three rotary switches labeled 'one', 'two', and 'three'. Below them is a widget named 'JSFuncBlockWgt'. The Properties window for this widget shows three 'DataLink' properties connected to 'NeedleWgt.value:Knob1', 'NeedleWgt.value:Knob2', and 'NeedleWgt.value:Knob3'. The script editor shows the following code:

```

1
2 function JSFuncBlockWgt_onDataUpdate
3 {
4 var vUNLOCK = page.getWidget("unlock")
5
6 // Accept the incoming new value
7 me[eventInfo.attrName] = eventInfo.newValue;
8
9 // Check the unlock code
10 if ((me.value1=="3") && (me.value2=="3") && (me.value3=="3")) {
11     vUNLOCK.setProperty("value", "Unlock!");
12 } else {
13     vUNLOCK.setProperty("value", me.value1+"-"+me.value2+"-"+me.value3);
14 };
15
16 return false;
17 };
18

```

```

1
2 function JSFuncBlockWgt_onDataUpdate(me, eventInfo)
3 {
4 var vUNLOCK = page.getWidget("unlock")
5
6 // Accept the incoming new value
7 me[eventInfo.attrName] = eventInfo.newValue;
8
9 // Check the unlock code
10 if ((me.value1=="3") && (me.value2=="3") && (me.value3=="3")) {
11     vUNLOCK.setProperty("value", "Unlock!");
12 } else {
13     vUNLOCK.setProperty("value", me.value1+"-"+me.value2+"-"+me.value3);
14 };
15
16 return false;
17 };
18

```

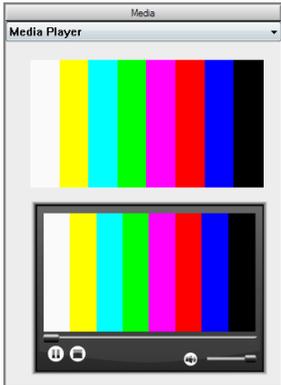
请参阅["控件事件"](#)在本页460,了解 `onDataUpdate` 的说明

多媒体播放器控件

路径: **控件库 > Media > Media Player**

使用这些控件播放播放列表中的视频。视频文件可以储存在 **USB 闪存、闪存卡或 SD 卡** 上。

有两个控件可用:一个包含一个多媒体框架,有播放和停止按钮,另一个是简单的框架,视频在没有用户控制的情况下播放。



参数	说明
多媒体播放器列表	打开 Windows 的文件浏览器以选择收集到播放列表中的视频文件。将在 HMI 设备中使用项目下载所选文件
循环样式	定义视频如何播放。 <ul style="list-style-type: none"> • 无循环:播放播放列表中的所有视频,然后停止。 • 首个循环:重复播放列表中的第一个视频。 • 全部循环:重复整个播放列表。 • 随机:以随机顺序播放视频。

 注:“多媒体播放器”控件只在一些 HMI 设备上有效(请参阅["HMI 设备地址"](#)在本页525)。它不适用于 LRH SW Client。

 注:一个页面中只能有一个多媒体播放器。

支持的视频编码

支持两组编译器:

- 基于 DSP 的视频编译器
- 软件视频编译器

 支持 DSP(视频硬件加速)的 HMI 设备列表在["HMI 设备地址"](#)在本页525上可用。

DSP 视频编解码器

这些包括：

- H264 使用 AVI/MP4 格式，无 CABAC，三级(建议)
- MPEG4 使用 MP4 格式



需要 *BSP v1.0.269* 或更高版本。

软件视频编解码器

只是：

- Microsoft MPEG4 v3 使用 AVI 格式。

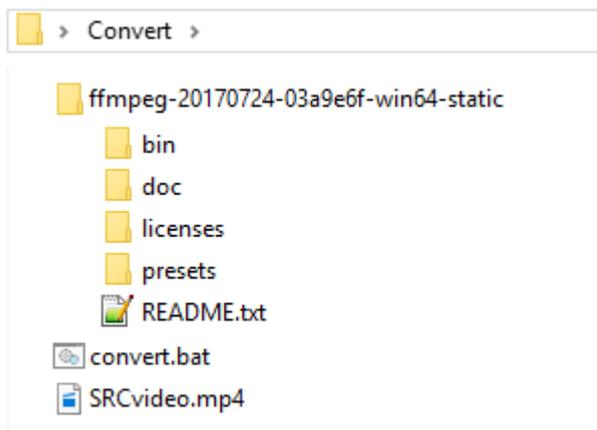


请注意，视频性能取决于所选的分辨率、比特率和设备功能。如果视频渲染不顺畅，请尝试降低视频的分辨率或比特率。

Microsoft MPEG4 v3 编码的视频没有使用硬件加速，有更多限制。要阻止视频运行不稳定，建议最大分辨率是 640x512 像素，比特率是 1300 kb/s。此外，该页面上使用的多媒体播放器控件大小应与播放列表中的视频相同，以避免放大和缩小。不支持音频。

转换视频

FFMPEG (www.ffmpeg.org) 可用于将视频转换为 HIM 设备支持的正确编解码器。通过使用下图文件夹架构，以下批处理文件可用于转换任何视频文件。



```

convert.bat x
1 @echo off
2 set FFMPEG=ffmpeg-20170724-03a9e6f-win64-static\bin\ffmpeg.exe
3
4 %FFMPEG% -i SRCvideo.mp4 ^
5         -y ^
6         -an ^
7         -s 240x160 ^
8         -b:v 4200k ^
9         -maxrate 4200k ^
10        -c:v libx264 ^
11        -profile:v baseline ^
12        -level:v 3 ^
13        -bufsize 3000k ^
14        -minrate 0 ^
15        -f avi ^
16        -preset slow ^
17        HMIvideo.avi
18
19 pause
20

```

现在您可以用标准的视频播放器打开转换的视频，如 Windows Media Player，验证质量。您可以将结果视频添加到多媒体播放器控件的播放列表。



注：LRH SW 不包含 FFMPEG 工具。

在 JavaScript 上使用多媒体播放器

多媒体播放器控件也可以用如下句法在 JavaScript 控件中引用：

```

//get the mediaplayer widget.
var mediaWgt = page.getWidget('MediaPlayerWgt2');
//load the play list
mediaWgt.setProperty('medialist', '/Storage Card/demo_3.avi,/Storage Card/video1_3.avi');
// set the loopstyle 0 - noloop, 1 - loop one, 2- loop all, 3 - random
mediaWgt.setProperty('loopstyle', 2);
//start playing the first file.
mediaWgt.mediapath = '/Storage Card/demo_3.avi';

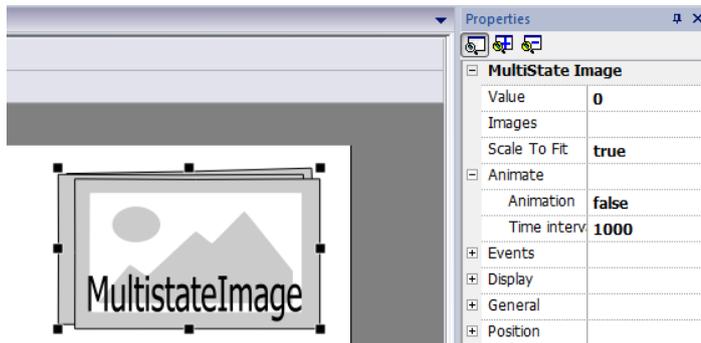
```

请参阅"JavaScript"在本页455了解如何使用 JavaScript 的详细信息。

多状态图像控件

路径: 控件库 > 基本 > 图片

使用该控件根据用于索引的标签的值从收藏中显示一个图像。您也可以将该控件用于简单的动画。

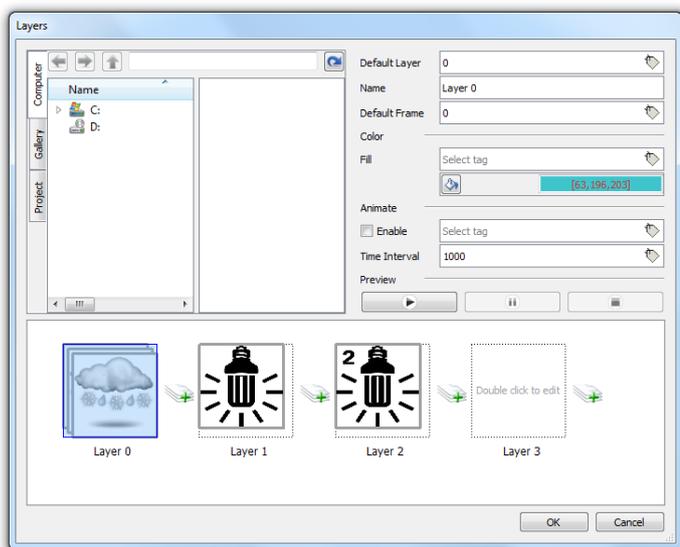


参数	说明
值	要显示的图像的索引。 例如，设置值=0，在图像收藏中显示索引为0的图像。
图像	有相关索引的图像收藏。
动画	设置为 true 来启用幻灯片。
时间间隔	幻灯片中图片之间的时间间隔。

多状态图像多层控件

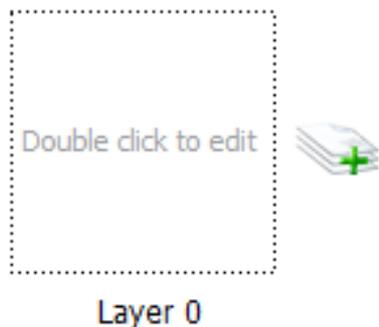
路径: 控件库 > 基本 > 图片

使用该控件创建不同的动画，在运行时选择最适合的。

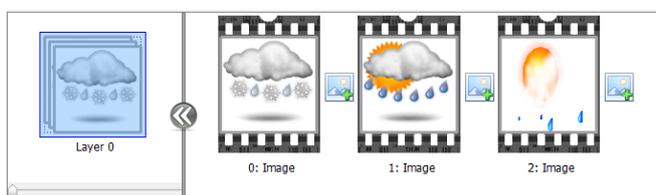


设置控件层

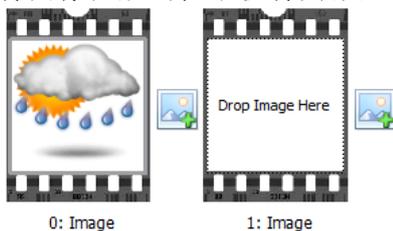
1. 从属性窗格中打开层对话框。
2. 单击 + 添加您需要的层。



3. 双击每一层添加尽可能多您想要该层包括的图像。



4. 将图像拖放到框中以将其添加到当前层。



5. 定义控件属性。

参数	说明
默认层	运行时显示的层。
名称	所选层的名称。
默认框	当前层显示时显示的框架。
颜色/填充	对当前层中的图像填充颜色。
动画	对活跃的层启用幻灯片放映。为其附属标签，运行时可以启动/停止动画。
时间间隔	如果启用的话，幻灯片的时间间隔。
预览	幻灯片模拟。



注：默认层、默认框、颜色和填充将其附加到标签可以在运行时更改。

网络适配器控件

路径: 控件库 > 基本 > 控件

使用 IP 控件设置网络适配器参数。

系统变量“网络 -> 状态”包含 IP 控件执行的最后操作的结果(请参阅“网络变量”在本页 131 了解详细信息)

RSS Feed 控件

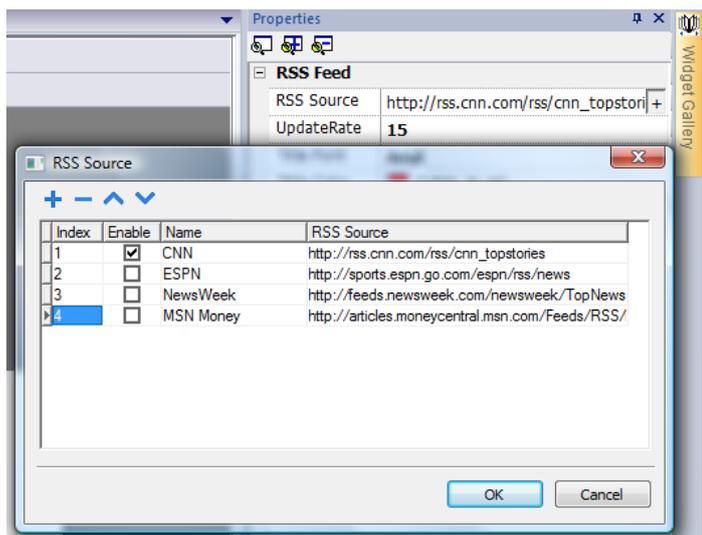
路径: 控件库 > 多媒体 > RSSFeed 源

使用该控件将您最喜欢的 RSS 源直接从 Internet 显示在 HMI 设备上。

RSSFeed



参数	说明
RSS 源	馈送 URL  注: 馈送源在运行时无法修改。
更新速率	刷新时间



RSS Feed控件经过特别设计以在袖珍网络浏览器上使用。

滚动 RSS Feed 控件

路径: 控件库 > 多媒体 > RSSFeed 滚动

使用该版本的主 RSS Feed 控件使用平滑滚动文本在文本行中显示高亮。

RSSFeed Scroll



RSS Scroll Widget : RSSScrollWgt	
RSS Source	http://rss.cnn.com/rss/cnn_topstories +
UpdateRate	15
Title Separator	
Title Font	Tahoma
Title Color	■ [23, 30, 40]
Title Size	12
Scrolling	Normal

该控件有额外的属性。

参数	说明
滚动	滚动速度
标题分隔符	在高亮之间的分离器字符

表格控件

路径: 控件库 > 基本 > 表格

使用此控件利用数据源中提供的数据创建表格。

配置表格:

1. 将表格控件放在屏幕上，配置表格模板。
2. 将控件添加到单元格中以配置一行或多行，当使用数据源提供的数据填充表格时，它们将用作行模板。
3. 选择将用于填充表格行的数据源
4. 定义控件和数据源中的链接。

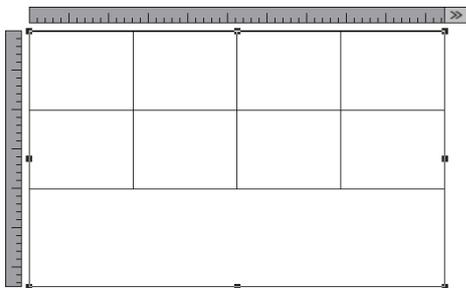
配置表格控件

表格控件有两种状态：

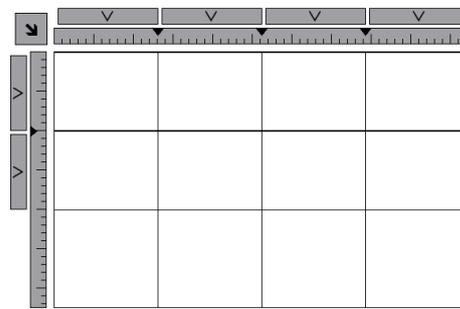
- 查看模式
- 编辑模式。

单击表格可在查看模式下管理控件，双击可进入编辑模式。要退出并返回到查看模式，请单击表格外的区域。

View Mode



Edit Mode



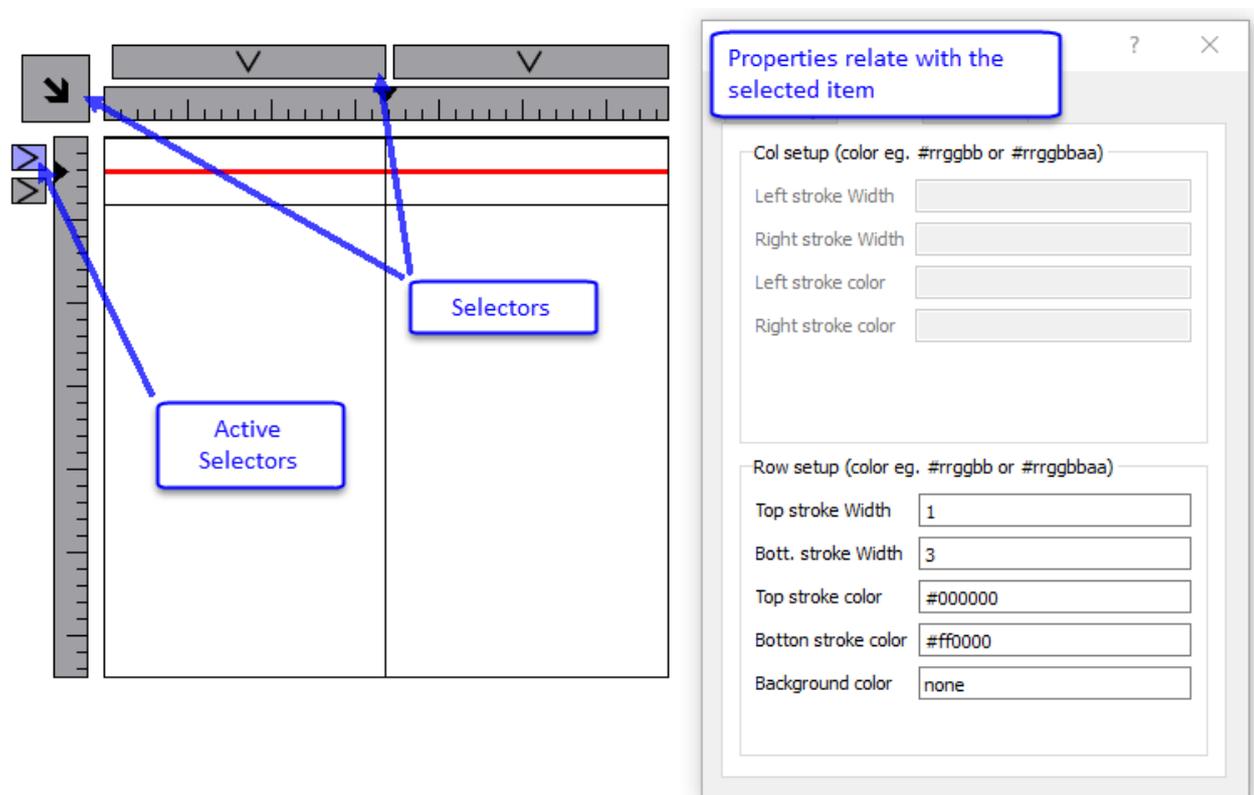
查看模式

在查看模式下，可以配置表格布局。将表格拖放到页面上，重新调整表格尺寸，定义模板行数、列数及主要表格属性。

Properties	
TableGroupWgt : TableWgt	
Current selected row	-1
Table model	
Grid Layout Group	
Num rows	2
Num columns	2
Horizontal Overflow	Scroll
Horizontal underflow mode	Center
Scrollbar color	■ [255, 0, 0]
Scrollbar image	
Scrollbar offset	2
Scrollbar size	20
Scrollbar autohide	Auto
Margin Collapsed	true
External margin width	0
External margin color	■ [0, 0, 0]
Events	
General	
Position	

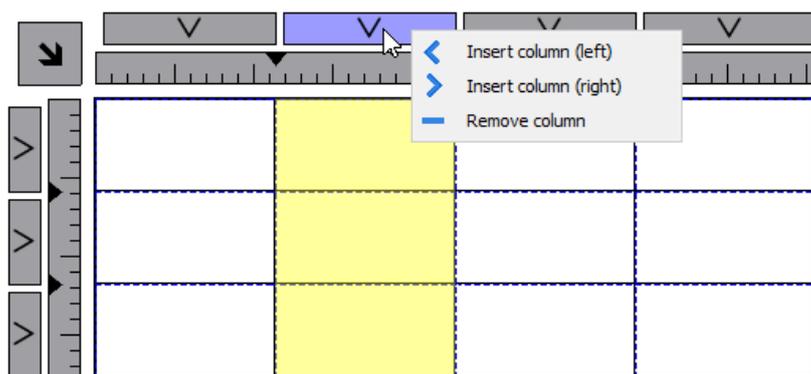
编辑模式

在编辑模式下，可以配置表格中各个单元格的格式和内容。表格中的每一行都将充当行模板。要配置表格外观，请单击表格选择器选择要配置的项目。



添加或删除行或列

要添加或删除行或列，请双击网格，进入编辑模式，并右击列或行选择器以打开上下文菜单。



合并或拆分行或列

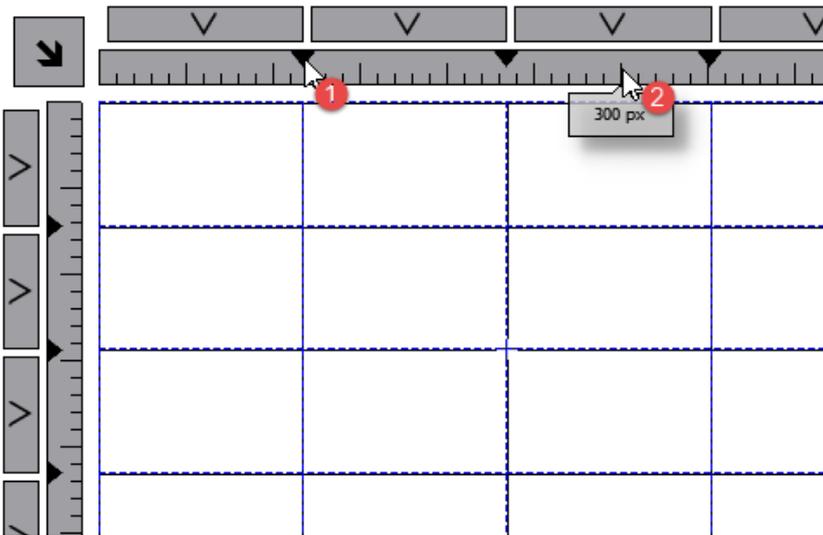
要合并或拆分行或列，请双击网格，进入编辑模式，然后让光标在功能区上移动：

- 双击黑色三角符号合并两个相邻行或列 (1)

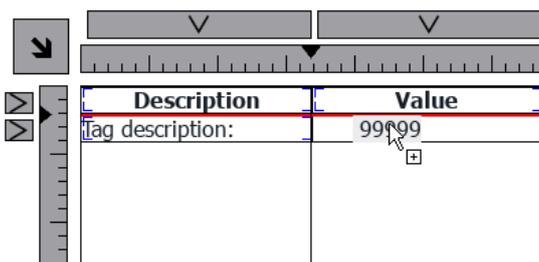


请注意，只能与空白行或空白列合并。

- 双击功能区拆分所进行或列 (2)



要配置单元格的内容，请将控件拖放到单元格内。



如果一个单元格内需要多个控件，请创建控件组并将其从页面复制到单元格中。

配置数据源

提供用于填充表格的数据的数据源，可以是表格数据源控件或 JavaScript 对象。

表格数据源控件

路径: 控件库 > 基本 > 表格

1. 将表格数据源控件拖放到页面上
2. 将表格模式参数设为链接至数据源。

1  ← Table Data Source Widget

Description	Value
tag description:	99999

Properties

TableGroupWgt : TableWgt

- Current selected row: -1
- Data Source
- Table model
- DataLink: model:TableDataSrcWgt
- Access Type: R
- Rows background
- Table filter
- Table Sorting 1 Column
- Grid Layout Group
 - Num rows: 2
 - Num columns: 2
 - Horizontal overflow: Scroll
 - Horizontal underflow mode: Center
 - Scrollbar Handle Color: [255, 0, 0]
 - Scrollbar Background Color: none
 - Scrollbar image
 - Scrollbar offset: 2
 - Scrollbar size: 20
 - Scrollbar autohide: Auto
 - External border mode: Auto

选择数据源，并在 TableDataSrcWgt 编辑器内添加需要的行和列。在下例中，我们定义了两个行模板：

- 行 0
表格标题。只包含静态文本。
- 行 1
行模板，含数据。在第一列中，我们添加了含有描述的标签，在第二列中，添加了含有值的字段。

Row 0
Row 1

Description	Value
tag description:	99999

TableDataSrcWgt Editor

Table rows + - ^ v Fixed header Table columns + - < > >] <]

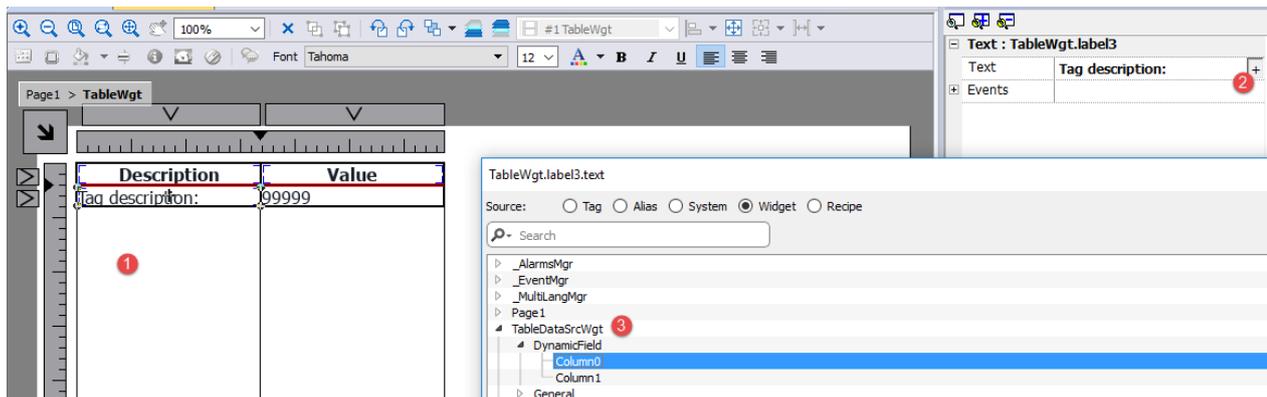
Row type	Column0	Column1
1 0	N/A	N/A
2 1	Temperature	Tag1 R/W
3 1	Humidity:	Tag2 R/W
4 1	Noise:	Tag3 R/W
5 1	Brightness:	Tag4 R/W

TableDataSrcWgt Editor Script

每一行都必须分配有行类型。行将采用相应行模板的格式。被置于行模板各个单元格中的控件将显示在这一类型的行中。

定义至数据源的链接

1. 双击表格控件，进入编辑模式并选择一个控件
2. 选择要从数据源中读取的属性
3. 选择将提供数据的数据源列



下图显示如何在运行时渲染示例

Description	Value
Temperature	111
Humidity:	222
Noise:	333
Brightness:	444

固定标题

如果不需要滚动第一行，请选中数据源工具箱上的“固定标题”复选框，或在数据源属性面板中设置“显示标题”属性为真(请注意，该参数仅在高级视图中可用)。

列覆盖

您可以在运行时使用整数数组来定义或修改列顺序。使用此属性时，请确保附加一个整数数组并将索引设置为-1(以选择整个数组)。

Column override (array of int):

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

Description:	Col 1	Col 2	Col 3	Col 4
00	1	2	3	4
Row 1	Data 1	Data 11	Data 111	Data 1111

Column override (array of int):

0	1	3	3	7	5	6	7	8
---	---	---	---	---	---	---	---	---

Description:	Col 1	Col 3	Col 3	Col 7
00	1	3	3	7
Row 1	Data 1	Data 111	Data 111	fdqfd

多语言

要启用多语言，请右键单击列的多语言图标。图标的颜色将变化以指出已启用支持。



避免在不必要时启用多语言支持以获得更佳的性能。

Table rows		Table columns	
Row type	Column0	Column1	
1 0	N/A	N/A	
2 1	Temperature	Tag1 R/W	
3 1	Humidity:	Tag2 R/W	
4 1	Noise:	Tag3 R/W	
5 1	Brightness:	Tag4 R/W	

导入/导出数据源

可以使用 xml 文件导入/导出数据源的配置

Table rows		Table columns	
Row type	Column0	Column1	
1 0	N/A	N/A	
2 1	Temperature	Tag1 R/W	

JavaScript 对象

除了数据源控件外，还可以从 JavaScript 对象提供用于填充表格的数据。在这种情况下，我们必须用要使用的数据填充元素数组并将数组分配给表格控件。

```
var myTable = page.getWidget("TableWgt1");
```

```
myTable.model = model;
```

模式是元素数组，含表格定义和数据。数组的第一个元素将包含行模板，而其他元素将包含要填充到表格行中的数据

```
model[0] = row_templates; // row templates
model[1] = row_data1; // data of the row1
model[2] = row_data2; // data of the row2
model[3] = row_data3; // data of the row3
model[4] = row_data4; // data of the row4
model[5] = row_data4; // data of the row5
```

行模板是多维数组，每个数组定义一个模板行的数据链接。

在下例中，我们使用一个两行模板。

```
var row_templates = {
  _h : [
    [ [] , [] ], //rowType = 0
    [ ["text"] , ["value"] ] //rowType = 1
  ]
}
```

第一行中有两列不包含数据链接。将此模板用作表格第一行的标题。

第二行定义第一行的模板，控件的“文本”属性填充到第一列中，控件的“值”属性填充到第二列中。系统将使用模式变量中提供的数据进行动态填充。

在下例中，我们定义**数据行**

```
var row_data = {
  _t : 1,
  _v : ["Temperature:", { _c : "dl" , s : "_TagMgr", a : "Tag1", i : 0, m : 2 }]
}
```

第一个元素是要使用的行模板，而第二个元素是含有要使用的数据的数组。在本例中，“Temperature:”是要在第一列上的控件中使用的文本，而另一元素是数据链接，将提供值填充在第二列的控件值属性中。

数据链接元素：

参数	说明
_c : "dl"	将元素识别为数据链接
s : "_TagMgr"	指定数据源是标签管理器
a : "Tag1", i : 0, m : 2	指定标签名称和索引(标签是数组时必需)以及读/写模式 <ul style="list-style-type: none"> • m=0 为只读 • m=1 为只写 • m=2 为读/写

以下 JavaScript 代码生成的表格与上述示例中用表格数据源控件生成的表格相同

```
var model = [];  
  
var row_templates = {  
  _h : [  
    [ [] , [] ], //rowType = 0  
    [ ["text"] , ["value"] ] //rowType = 1  
  ]  
}  
  
var row_data1 = {  
  _t : 0,  
  _v : [],  
  _h : true  
}  
  
var row_data2 = {  
  _t : 1,  
  _v : ["Temperature:", { _c : "dl" , s : "_TagMgr", a : "Tag1", i : 0, m : 2 } ]  
}  
  
var row_data3 = {  
  _t : 1,  
  _v : ["Humidity:", { _c : "dl" , s : "_TagMgr", a : "Tag2", i : 0, m : 2 } ]  
}  
  
var row_data4 = {  
  _t : 1,  
  _v : ["Noise:", { _c : "dl" , s : "_TagMgr", a : "Tag3", i : 0, m : 2 } ]  
}  
  
var row_data5 = {  
  _t : 1,  
  _v : ["Brightness:", { _c : "dl" , s : "_TagMgr", a : "Tag4", i : 0, m : 2 } ]  
}  
  
model[0] = row_templates;  
model[1] = row_data1;  
model[2] = row_data2;  
model[3] = row_data3;  
model[4] = row_data4;  
model[5] = row_data5;  
  
var myTable = page.getWidget("TableWgt1");  
myTable.model = model;
```

请注意第一行(`row_data1`)包含指令 `_h:true` 以避免第一行滚动。

```
var row_data1 = {  
  _t : 0,  
  _v : [],
```

```

    _h : true
  }

```

多语言

可以使用以下元素输入多语言文本：

```
{ _c : "ml" , mltext : {"en-US" : "Temperature:", "it-IT" : "Temperatura:"}}
```

参数	说明
<code>_c : "ml"</code>	将元素识别为多语言文本
<code>mltext : { ...}</code>	对列表:"ID 语言":"文本" 示例： <ul style="list-style-type: none"> • "en-US" : "Temperature:" • "it-IT" : "Temperatura:"

示例：

```

var row_data2 = {
  _t : 1,
  _v : [ { _c : "ml" , mltext : { "en-US" : "Temperature:",
                                "it-IT" : "Temperatura:" } },
        { _c : "dl" , s : "_TagMgr", a : "Tag1", i: 0, m : 2 }
      ]
}

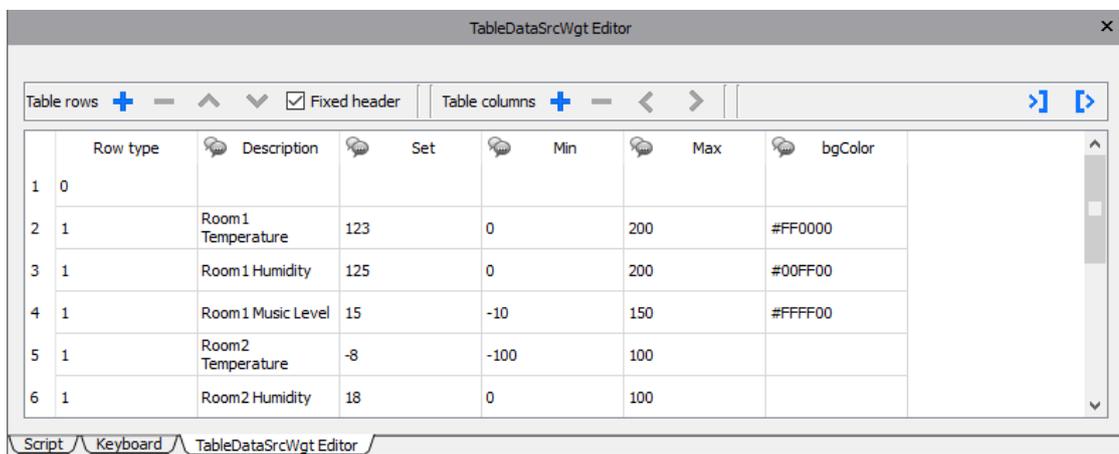
```

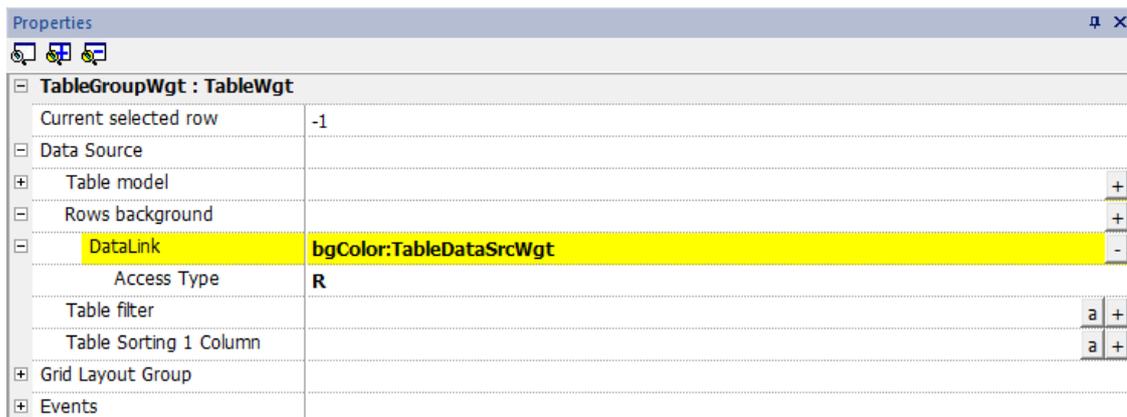
行背景颜色

使用行背景参数可以定义将包含关联行的背景颜色的数据源控件列。

要配置表格行的背景颜色，请执行以下操作：

1. 在数据源控件中添加新列以包含每行的背景颜色
2. 配置表格的“行背景”颜色参数以指向数据源控件的颜色列





Filter:

Description	Set	Min	Max
Room1 Temperature	123	0	200
Room1 Humidity	125	0	200
Room1 Music Level	15	-10	150
Room2 Temperature	-8	-100	100
Room2 Humidity	18	0	100
Room2 Music Level	12	0	150
Room3 Temperature	15	0	150
Room3 Humidity	134	0	500

表格筛选器

可使用“表格筛选器”属性筛选表格中可见的内容。在数据链接上，您可以使用公式(请参阅“公式”在本页44章节了解更多详细信息)定义要用于筛选数据的条件。

仅当表格筛选器的数据链接返回 **true** 值时，表格的每行才可见。

示例 1

如果要选择仅看到“说明”列中包含“内容”的行，则可以使用以下公式：

```
= $Contains($('Description:TableDataSrcWgt'), $('value:SearchOnTable'))
```

其中

- 'Description:TableDataSrcWgt' 是表格中数据源控件的动态字段，用于识别要检查的列
- 'value:SearchOnTable' 是将包含要搜索的字符串的文本字段值

Filter: X

Description	Set	Min	Max
Room1 Temperature	123	0	200
Room1 Humidity	125	0	200
Room1 Music Level	15	-10	150
Room2 Temperature	-8	-100	100
Room2 Humidity	18	0	100
Room2 Music Level	12	0	150
Room3 Temperature	15	0	150
Room3 Humidity	134	0	500

Filter: X

Room2 Temperature	-8	-100	100
Room2 Humidity	18	0	100
Room2 Music Level	12	0	150

Filter: X

Room1 Temperature	123	0	200
Room2 Temperature	-8	-100	100
Room3 Temperature	15	0	150
Room4 Temperature	2	0	10

Properties

TableGroupWgt : TableWgt

- Current selected row: -1
- Data Source
 - Table model: +
 - Rows background: a +
- Table filter: +
- DataLink: = \$Contains(\$"Description:TableDataSrcWgt"),\$("value:SearchOnTable'")) -
- Access Type: R
- Table Sorting 1 Column: a +
- Grid Layout Group
- Events

示例 2

要使用标志定义要在表格中显示的参数，请执行以下操作：

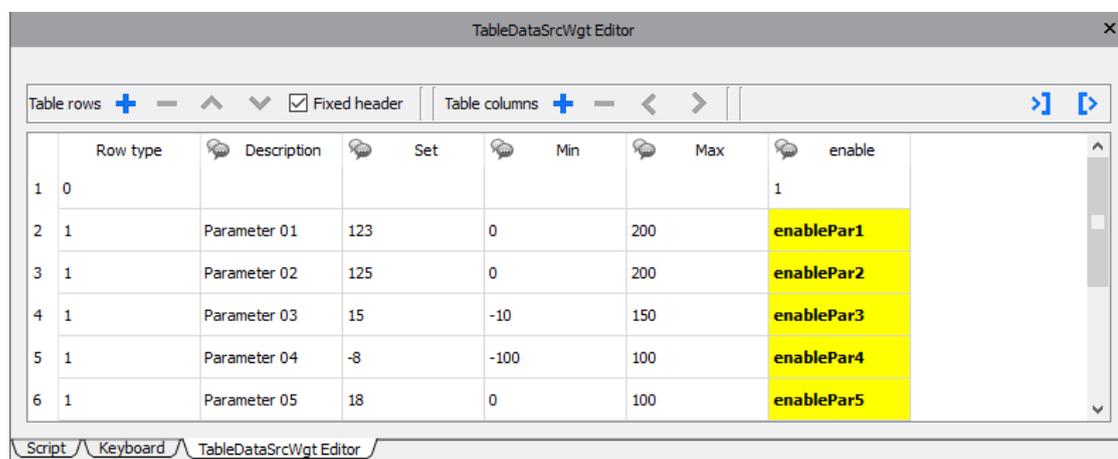
首先，在数据源中添加一个新列，该列将包含启用关联行的标志。然后，将表格筛选器的数据链接链接到包含标志的新列

enablePar1 = 1
 enablePar2 = 1
 enablePar3 = 1
 enablePar4 = 1
 enablePar5 = 1

Description	Set	Min	Max
Parameter 01	123	0	200
Parameter 02	125	0	200
Parameter 03	15	-10	150
Parameter 04	-8	-100	100
Parameter 05	18	0	100
Parameter 06	12	0	150
Parameter 07	15	0	150
Parameter 08	134	0	500

enablePar1 = 1
 enablePar2 = 1
 enablePar3 = 0
 enablePar4 = 0
 enablePar5 = 1

Description	Set	Min	Max
Parameter 01	123	0	200
Parameter 02	125	0	200
Parameter 05	18	0	100
Parameter 06	12	0	150
Parameter 07	15	0	150
Parameter 08	134	0	500
Parameter 09	44	0	50
Parameter 10	2	0	10



TableGroupWgt : TableWgt	
Current selected row	-1
Data Source	
Table model	+
Rows background	a +
Table filter	+
DataLink	enable:TableDataSrcWgt1
Access Type	R
Table Sorting 1 Column	a +
Grid Layout Group	
Events	

表格排序

要对表格的行进行排序，请选择适用于表格排序的数据源列

- 排序模式可为升序或降序
- 排序规则可按字母或数字

TableGroupWgt : TableWgt	
Current selected row	-1
Precached Pages	0
Data Source	
Table model	+
Rows background	a +
Table filter	a +
Table Sorting 1 Column	+
DataLink	Column0:TableDataSrcWgt
Access Type	R
Table Sorting 1 Mode	Ascendent
Table Sorting 1 Rule	Alphabet
Table Sorting 2 Column	a +
Grid Layout Group	
Events	
General	

当一列具有重复项时，多重排序(稳定排序)能派上用场。您最多可使用三个排序列。

Example of sorting:

Alphabetic

Value
1
10
15
7

Numeric

Value
1
7
10
15

 甚至可以使用 `SetTableSortingColumn` 宏对表格进行排序(请参阅 "[SetTableSortingColumn](#)" 在本页 213 了解更多信息)。

水平滚动位置

“水平滚动位置”可以同步两个表格的水平滚动移动。

External margin color	 [0, 0, 0]
Horizontal scroll position	0 +
DataLink	relHorScrollPos:GroupWgt2.TableWgt -
Access Type	R



水平滚动位置参数仅在“高级属性视图”模式下可用

预缓存页面

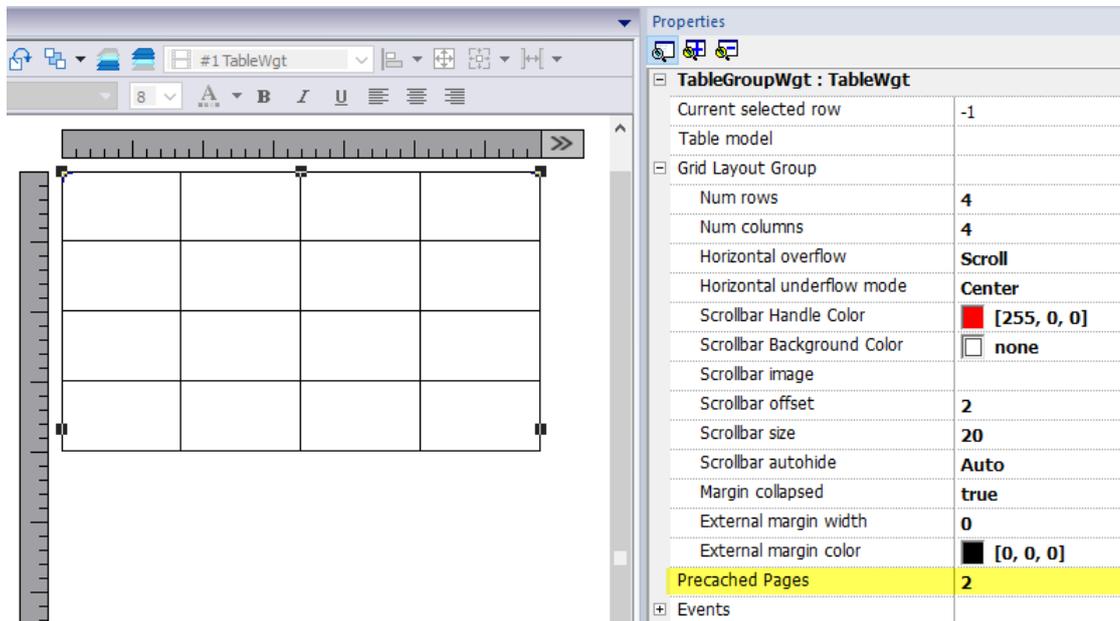
通常, LRH SW HMI Runtime 仅检索将在显示中可见的数据。为了方便表格滚动, 预加载显示页面下一行和前一行的数据可能有用。使用预缓存页面参数可以定义将预加载的页数

- 0 = 未预加载页面
- N = 要预加载的页数

示例:

使用包含 4 行和预缓存页面 = 2 的表格

- 要预加载的行数为 8(2 页 x 4 行)
- 前 4 行(已准备管理表格向上滚动)
- 上面 4 行(已准备管理表格向下滚动)



The screenshot shows a software interface with a table and a properties panel. The table has 4 columns and 4 rows. The properties panel on the right is titled 'TableGroupWgt : TableWgt' and contains the following settings:

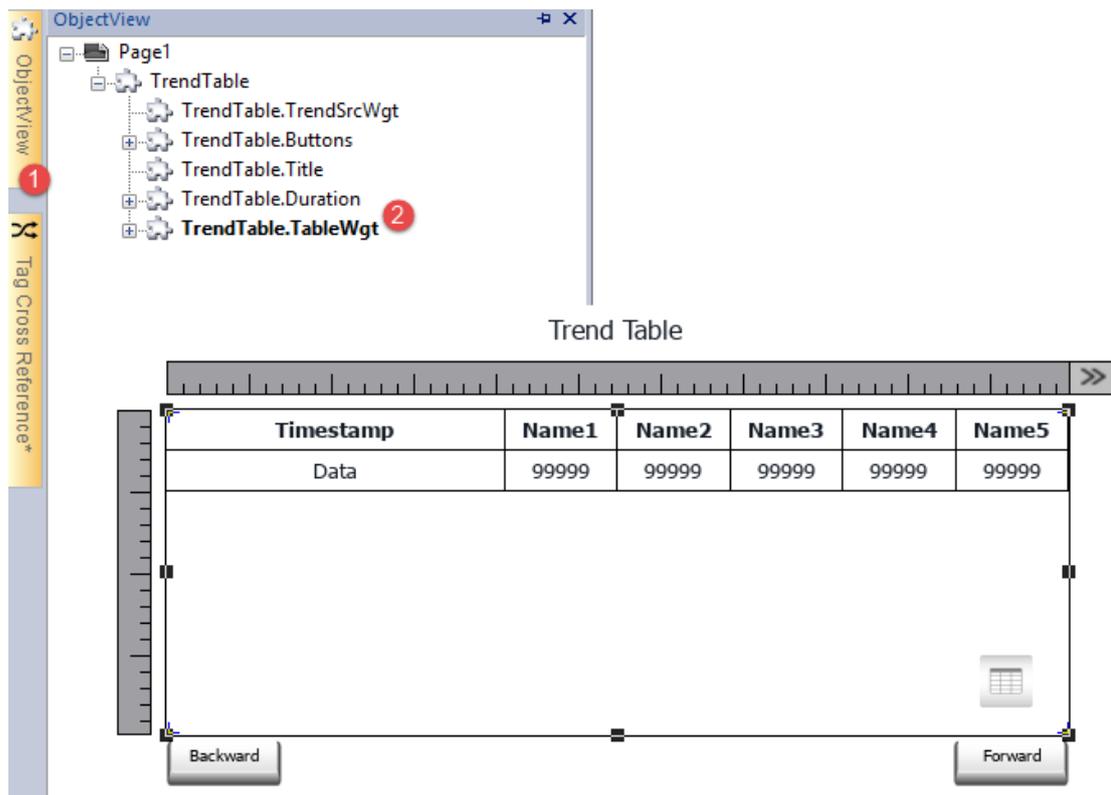
Current selected row	-1
Table model	
Grid Layout Group	
Num rows	4
Num columns	4
Horizontal overflow	Scroll
Horizontal underflow mode	Center
Scrollbar Handle Color	[255, 0, 0]
Scrollbar Background Color	none
Scrollbar image	
Scrollbar offset	2
Scrollbar size	20
Scrollbar autohide	Auto
Margin collapsed	true
External margin width	0
External margin color	[0, 0, 0]
Precached Pages	2
Events	



预缓存页面参数仅在“高级属性视图”模式下可用

包含表格的控件

在库里，有一些控件包含表格，例如趋势表，审核表等。要打开表的属性或数据源的属性，可以使用“对象视图”选项卡并选择需要配置的组件。



打印表格

可从打印报表库查找并使用表格控件。

TextEditor 控件

路径: 控件库 > 高级 > 编辑器

使用此控件编辑文本文件。该控件可以使用 FTP 连接从本地 HMI 设备加载文本文件或者从远程设备下载文件。

 注: TextEditor 控件仅在需要时作为插件(请参阅"插件"在本页 79 了解详细信息)下载到 LRH SW HMI Runtime。



控件按钮

按钮	说明
打开	加载 TextEditor 内的文本文件
保存	保存文本文件
取消	移除上一 OPEN 或 SAVE 命令中的所有更改
编辑	进入编辑模式
插入	插入新行
删除	删除当前的行
上/下	上移/下移光标

控件属性

选项	说明												
键盘	<p>TextEditor 控件具有嵌入式键盘。使用无嵌入式键盘的控件时，系统将在您进入编辑模式时显示字母键盘。</p> <ul style="list-style-type: none"> • 隐藏 • 可见 												
FTP 配置	<p>用于从远程 FTP 服务器下载文本文件的 FTP 参数。将此字段留空以从本地 HMI 设备加载文本文件。</p> <table border="1"> <thead> <tr> <th>参数</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>FTP 地址</td> <td>FTP 服务器 IP 地址</td> </tr> <tr> <td>服务器端口</td> <td>FTP 连接端口(默认 = 21)。</td> </tr> <tr> <td>验证</td> <td> 选择要使用的 FTP 身份验证： <ul style="list-style-type: none"> • 标准(需要用户名和密码) • 匿名 </td> </tr> <tr> <td>用户名</td> <td>远程 FTP 帐户的用户名</td> </tr> <tr> <td>密码</td> <td>远程 FTP 帐户的密码</td> </tr> </tbody> </table>	参数	说明	FTP 地址	FTP 服务器 IP 地址	服务器端口	FTP 连接端口(默认 = 21)。	验证	选择要使用的 FTP 身份验证： <ul style="list-style-type: none"> • 标准(需要用户名和密码) • 匿名 	用户名	远程 FTP 帐户的用户名	密码	远程 FTP 帐户的密码
参数	说明												
FTP 地址	FTP 服务器 IP 地址												
服务器端口	FTP 连接端口(默认 = 21)。												
验证	选择要使用的 FTP 身份验证： <ul style="list-style-type: none"> • 标准(需要用户名和密码) • 匿名 												
用户名	远程 FTP 帐户的用户名												
密码	远程 FTP 帐户的密码												
文件名	有待编辑的文件名。留空时，文件浏览器将打开，供您加载本地文件												
语法突出显示类型	<p>根据所选语言将文本显示为不同的颜色</p> <ul style="list-style-type: none"> • 无 • GCode 												

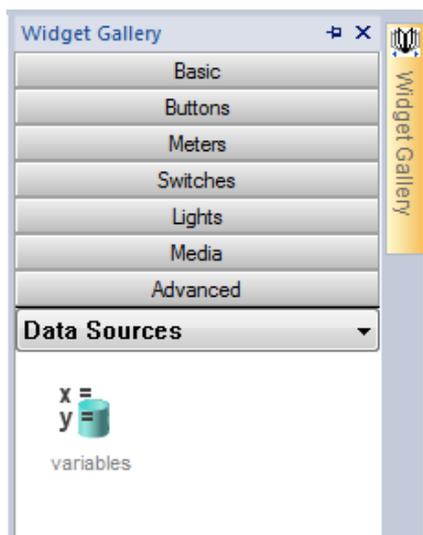
变量控件

路径: 控件库 > 高级 > 数据源

使用该控件为数据传输或 JavaScript 程序添加内部变量。



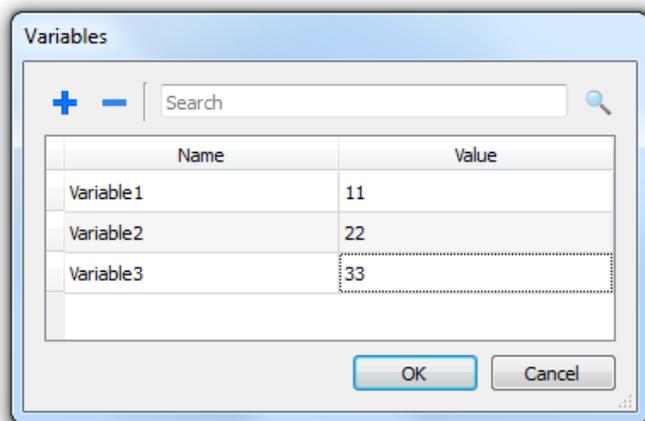
注: 这些变量位于控件插入的页面。



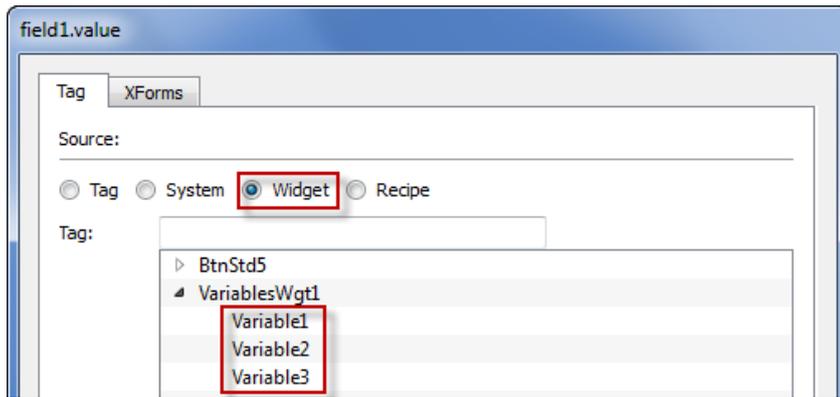
当您将该控件拖拽到您的页面，会显示一个占位符表明控件位置，但是在运行时不可见。

设置该控件

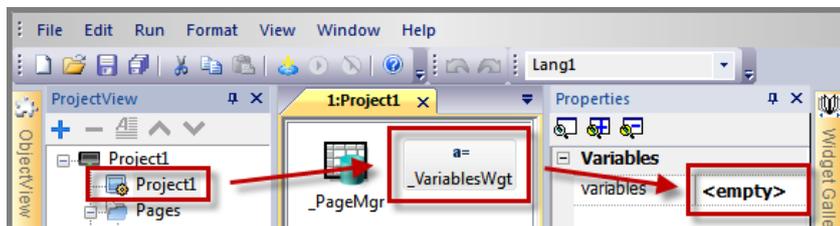
要创建变量并为其指定值，请从属性窗格的变量属性中打开变量对话框。



您随后可以从页面编辑器的附加标签对话框中引用这些变量。



如果您需要全局变量，在项目级别对其配置，为全局变量控件添加想要的变量。



JavaScript 使用变量

变量也可以用如下句法在JavaScript程序中引用：

对于局部变量：

```
var varWgt = page.getWidget("_VariablesWgt");
var compVar = varWgt.getProperty("VariableName");
```

对于全局变量：

```
var varWgt = project.getWidget("_VariablesWgt");
var compVar = varWgt.getProperty("VariableName");
```


37 自定义控件

LRH SW 有一个巨大的控件库，其中包括预定义动态控件(按钮、灯、仪表、开关、趋势、配方以及对话框项目)以及静态图像(形状、管道、容器、电机)。

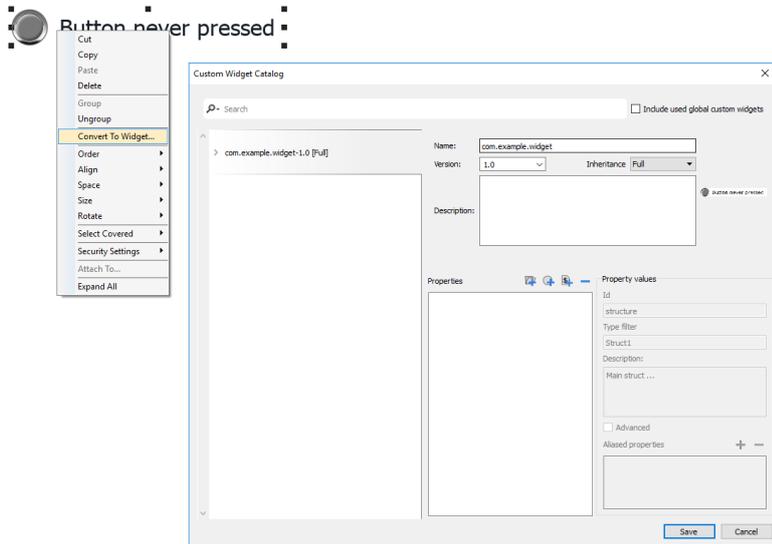
您可以从图库中拖拽一个对象到页面上，然后对其改变大小、移动位置、改变方位或改变形状。图库中的所有控件都是基于矢量的，所以改变大小时不会失真。

但是，您可以修改任何预定义的控件来创建您的自定义控件。自定义控件可以由几个元素组成，只包括所需的属性。

创建一个自定义控件	440
将属性添加到自定义控件中	442
使用结构标签	444
自定义控件的 JavaScript	447
用户库	450

创建一个自定义控件

1. 将您要用于组成自定义控件的所有控件拖放到页面上。
2. 选择并对其进行编组。
3. 右键单击分组对象并选择**转换为控件: 自定义控件目录**对话框将显示。



参数	说明
包括已用自定义控件	选中时，列出项目内使用的所有控件。甚至列出系统控件。
名称	您可以定义您喜欢的任何名称，但通常应保留名称结构。文件夹 com.hmi 为系统控件保留
说明	控件描述。
版本	控件版本。 共享同一版本的所有控件共享“继承”参数中定义的属性。
继承	在同一版本的控件间共享的属性 <ul style="list-style-type: none"> • 全部(图形和逻辑) • 仅图形 • 仅逻辑 • 禁用

修改自定义控件

要修改自定义控件，只需双击自定义控件即可进入编辑模式。

如果启用继承标志，将出现锁定图标，提示您正在添加将传播至共享同一版本的所有其他自定义控件的修改。单击挂锁图标以确认进入编辑模式，挂锁将打开。完成修改后再次单击。

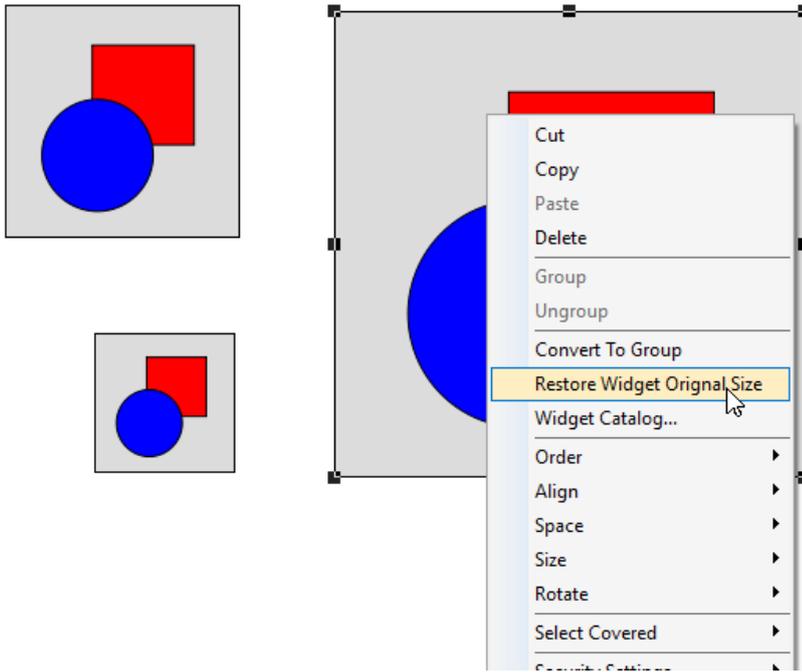


仅当启用继承时才显示挂锁。

调整自定义控件大小

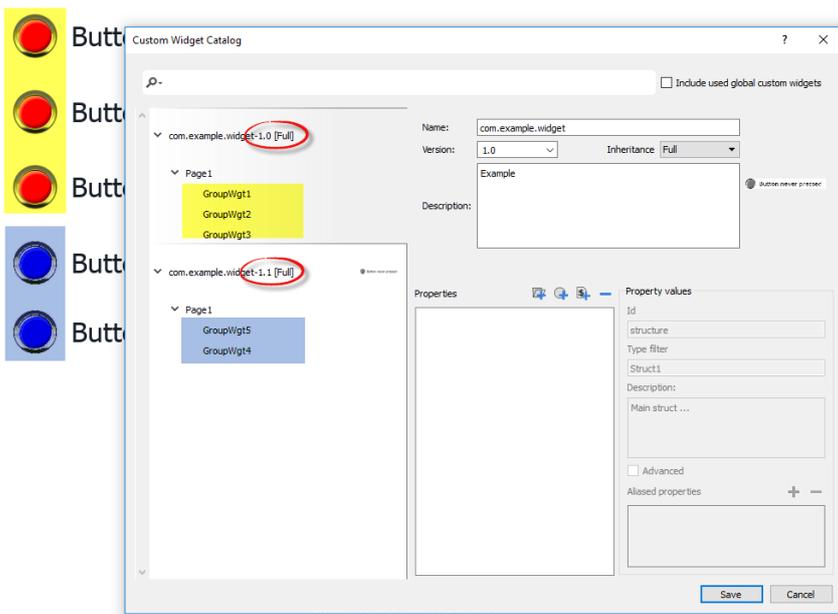
更改自定义控件大小时，新大小不会传播至其他控件实例。

可使用“恢复控件原始大小”命令。



共享属性

修改自定义控件时，所有修改均将传播至共享同一版本且配置为继承控件属性的所有其他自定义控件。

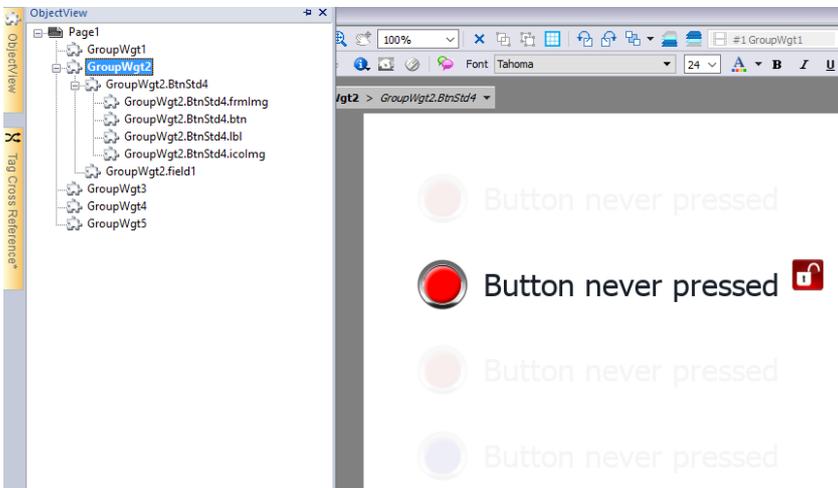


使用控件组件

控件通常有许多部分组成，例如，一个按钮是一个复杂的控件，包括两个图像控件，一个按钮控件和标签。

要显示控件所有元素的列表，请选择该控件，打开挂锁并打开**对象视图**窗格：系统按层级顺序显示构成复杂控件的所有元素。

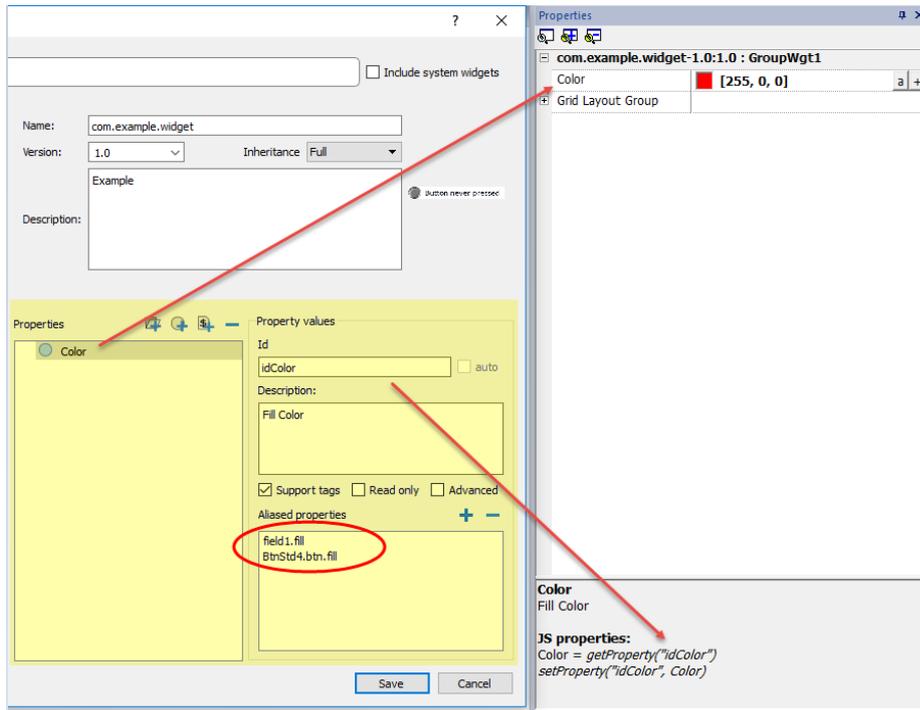
要选择单个控件，请直接从**对象视图**窗格中选择。



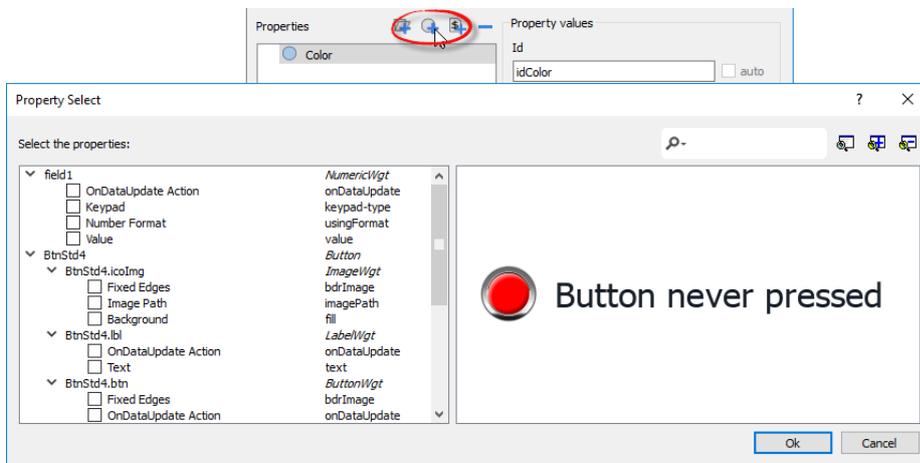
将属性添加到自定义控件中

创建自定义控件时，您需要在**属性**窗格中定义为其显示的属性。

1. 右键单击分组对象并选择**控件目录：“属性”**对话框将显示。



2. 单击 **+** 以打开**属性选择**对话框：这将列出所有分组控件的所有属性。



3. 选择为自定义控件定义的属性。

4. 定义每个属性的细节。



请注意，您可以创建文件夹，并使用拖放移动或重新组织**属性**列表

参数	说明
属性	属性窗格中显示的名称。
说明	属性窗格中显示的属性的注释。

参数	说明
Id	LRH SW 为 JavaScript 函数和“附加标签”对话框显示的名称。
支持标签	规定该属性是否支持“附加”属性。
只读	仅显示为读取模式的属性
高级	规定各种属性是否应该出现在高级或属性窗格的简单视图模式中。
别名属性	与显示属性相链接的内部属性

组合属性

要组合两个或多个属性：

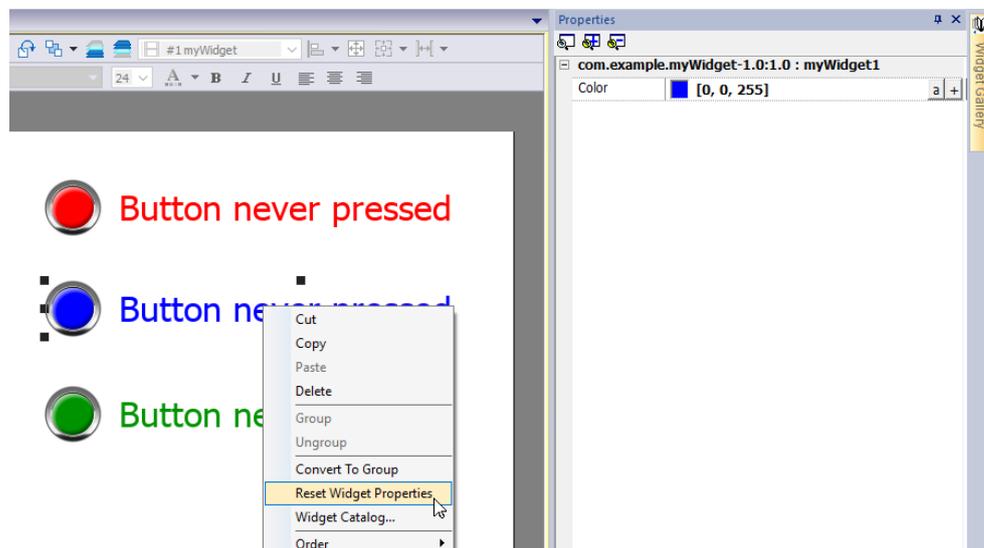
1. 在“属性”列表框中选择主属性。
2. 单击别名属性工具栏中的 **+: 属性选择** 对话框将显示。
3. 选择您需要的组合属性。
4. 单击 **确定**：组合属性将显示在别名属性列表框中。

例如

如果您将“颜色”属性插入所有控件(例如 `filed1.fill` 和 `BtnStd4.btn.fill`) 的填充颜色中，那么在您设置自定义控件的显示“颜色”属性时，所包含控件的所有颜色均将更改。

重置控件属性

“重置控件属性”可以将修改后的属性值重置为原始值。



使用结构标签

通过创建控件实例使用包含许多标签的控件，我们只需要提供包含所有标签的结构标签名称，而无需配置各个单独标签。

我们以下的控件为例。它使用四个标签，房间名称、温度、湿度和气压。如果我们需要将此标签的两个实例用于两个不同的房间，我们需要配置八个标签，每个房间四个标签。

Bathroom

Temperature: 23.0
Humidity: 52
Pressure: 105

Living room

Temperature: 21.0
Humidity: 22
Pressure: 101

通过使用参数属性，只需提供结构名称，即可设置控件的所有数据链接。

Bathroom

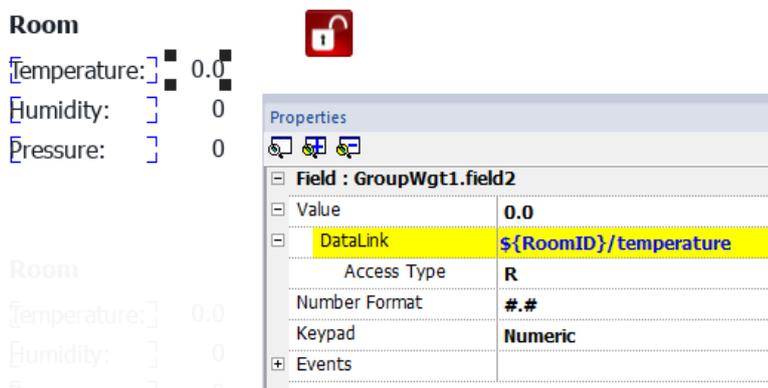
Temperature: 23.0
Humidity: 52
Pressure: 105

Living room

Temperature: 21.0
Humidity: 22
Pressure: 101

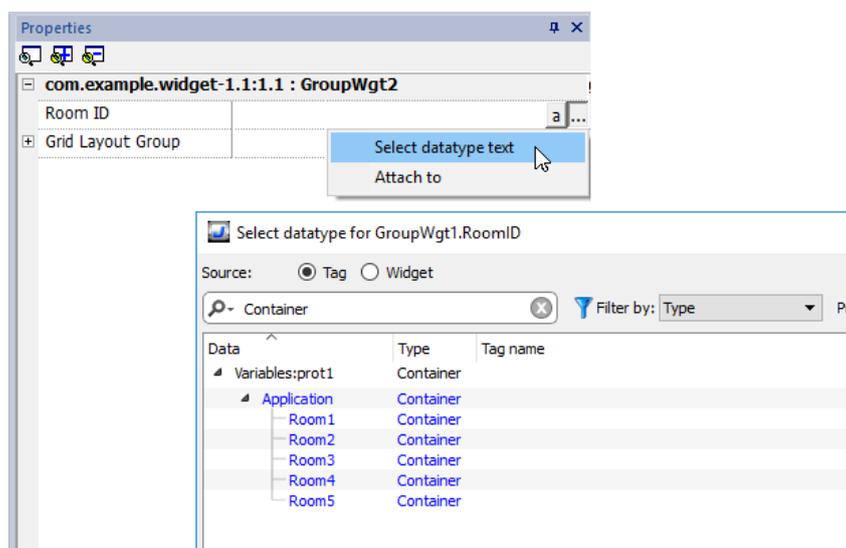
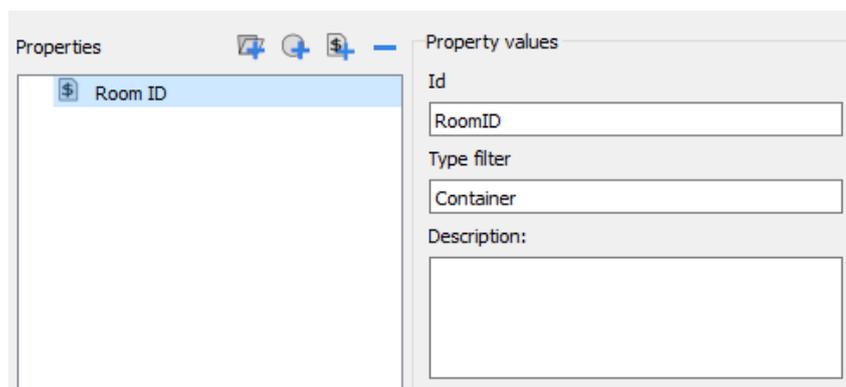
可以使用“添加参数”图标在自定义控件内添加“参数”字段。

要配置自定义控件的数据链接，可以使用关键字 `#{RoomID}` 在结构实例处引用



类型筛选器

通常，参数值将为结构标签元素。通过使用“类型筛选器”参数，“选择数据类型文本”将列出筛选标签。



当“附加至”将数据链接返回至包含要使用的字符串的标签时，“选择数据类型文本”将返回一个字符串。

getParameter

可使用 `getParameter()` 从 JavaScript 读取参数值

对象 `getParameter(paramID)`

示例：

```
var myWidget = page.getWidget("myWidget");
function BtnStd3_btn_onMouseClicked(me, eventInfo)
{
    alert("Room is: " + myWidget.getParameter("RoomID"));
}
```

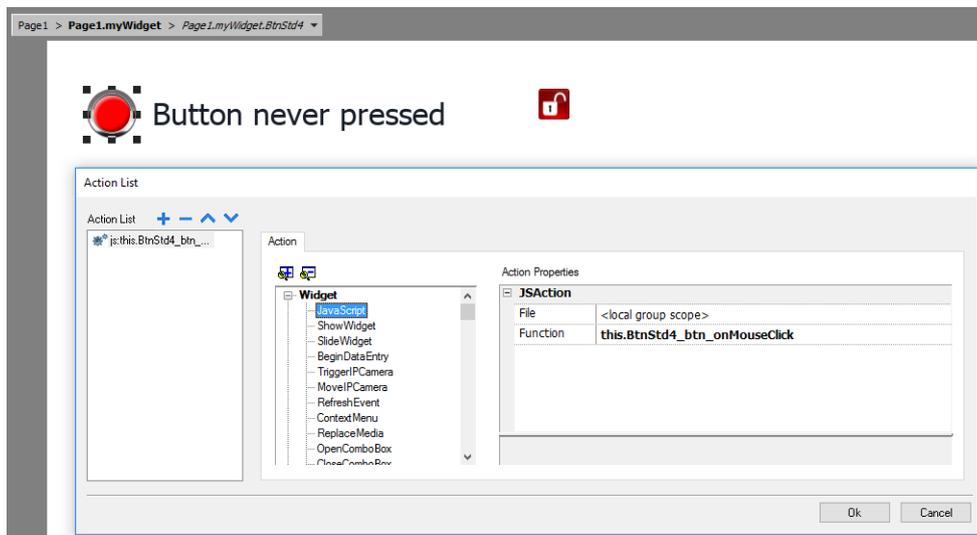


您也可以使用 `getProperty()`，但使用 `getParameter()` 读取自定义控件参数更加有效

自定义控件的 JavaScript

JavaScript 功能可嵌入自定义控件中。

双击自定义控件然后单击挂锁后，编辑模式激活，可以将 JavaScript 代码关联至可用事件。





请注意，必须使用运算符 **this**，才能关联自定义控件的多个实例。

如果需要引用控件元素，则可以使用关键字 **wgt**。例如，使用 **wgt.id** 在活动控件实例的 ID 处引用。



如果您剪切并粘贴上述示例中自定义控件的部分实例并执行，例如在模拟器内部，则将获得以下结果。

 Tue Jan 31 2017 14:51:18

 Button never pressed

 Tue Jan 31 2017 14:51:12

 Tue Jan 31 2017 14:51:14

 Button never pressed

onActivate 属性

要初始化自定义控件，可以使用初始化函数定义 `onActive` 属性，如以下示例所示。

在加载页面时将执行 `onActivate()` 函数

```

Script
1  /*!
2  javascript module: widget-1.0.js
3  javascript source file path: lib\com\example\widget-1.0\widget-1.0.js
4  */
5
6  this.wMSG = wgt.getWidget(wgt.id+".field1")
7
8  this.BtnStd1_btn_onMouseClicked = function (me, eventInfo)
9  {
10     var now = new Date();
11     this.wMSG.setProperty("value", now.toString().slice(0, 24));
12 }
13
14
15 this.onActivate = function()
16 {
17     this.wMSG.setProperty("value", "Button never pressed");
18 };
19 this.onActivate();

```



请注意，自定义控件也可粘贴在用户库内，以便以后使用。

用于本章示例中的 JavaScript 代码

```

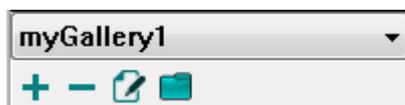
/*! javascript module: widget-1.0.js javascript source file path:
lib\com\example\widget-1.0\widget-1.0.js */ this.wMSG = wgt.getWidget
(wgt.id+".field1") this.BtnStd1_btn_onMouseClicked = function (me, eventInfo) { var now
= new Date(); this.wMSG.setProperty("value", now.toString().slice(0, 24)); }
this.onActivate = function() { this.wMSG.setProperty("value", "Button never pressed");
}; this.onActivate();

```

用户库

开发人员创建的控件可以保存在用户库中，以供开发新项目时使用。

用户库工具栏

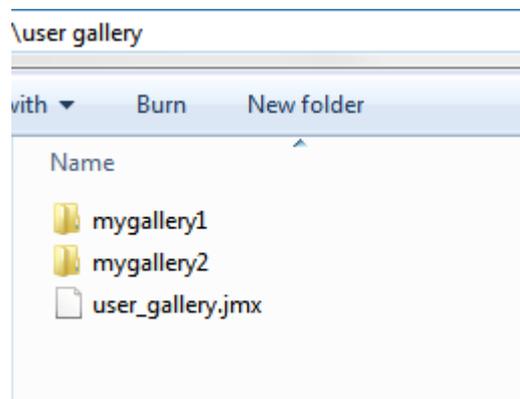


命令	说明
	在 LRH SW 编辑器中打开所选控件文件夹
	添加新控件文件夹
	删除当前所选文件夹
	选择用户控件文件夹

要将新控件添加到用户库，打开控件文件夹，然后编辑库页面，创建或添加新控件。



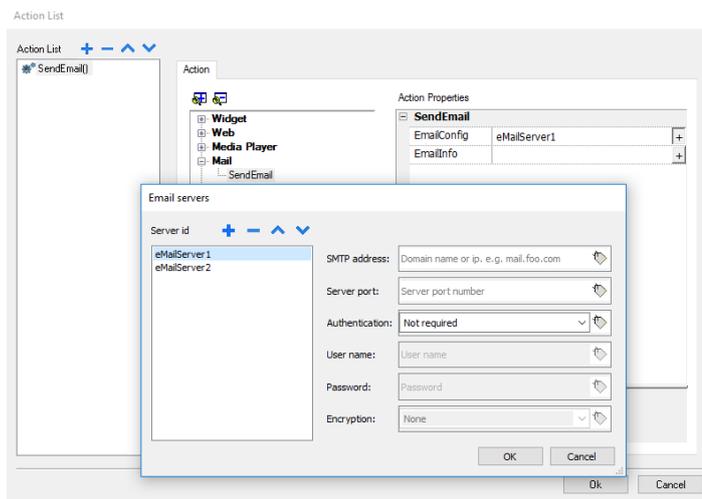
提示：要导入用户库子文件夹，只需将要导入的文件夹复制到主用户库文件夹中即可。



38 发送邮件信息

使用发送邮件动作发送邮件，包括邮件正文标签和附件。

创建发送邮件动作是为了处理报警和计划任务，也可以由其他的事件启动和执行。



配置电子邮件服务器	452
配置电子邮件	452

配置电子邮件服务器

要配置电子邮件服务器，请在**电子邮件配置**设置中输入以下信息：

参数	说明
SMTP 地址	SMTP 服务器地址。
服务器端口	SMTP 服务器连接端口 (默认 = 25)。
要求验证	选择 SMTP 服务器是否需要验证。
用户名	使用 SMTP 服务器发送邮件的用户名。
密码	使用 SMTP 服务器发送邮件的密码。
加密	加密类型 (无或 SSL)。

单击 **+** 添加更多的电子邮件服务器。



提示：可以动态更改 LRH SW HMI Runtime 中的服务器参数，但需要使用标签。

配置电子邮件

在**电子邮件信息**设置中输入以下信息：

参数	说明
名称	可选，此信息仅用于日志。
说明	可选，此信息仅用于日志。
发件人	发件人电子邮件地址 (例如, John@domain.com)。
收件人	收件人电子邮件地址。要输入多个地址，用分号隔开。
主题	电子邮件主题。
附件	要作为附件发送的文件的完整路径。一次只能发送一个附件。  注：附件大小的最大值通常由 SMTP 服务器设置。
正文	电子邮件主要内容。如果您将活动标签放在方括号里，您可以在这里插入活动标签。 例如，一个像“标签 1 的值是 [标签 1]”的消息将被发送为“标签 1 的值是 45”，如果当前标签 1 的值是 45。



提示：将一个字符串标签附加到**发件人**、**收件人**和**主题**字段中，以便能够在 LRH SW HMI Runtime 中更改。



警告:消息正文最大是 4096 字节,超出的文本将被截去。

添加电子邮件模版

单击 + 添加更多模板。

The screenshot shows a software dialog box titled "Emails". On the left side, there is a "Drafts" section with a list containing one item, "eMail1". Above the list are navigation icons: a plus sign (+), a minus sign (-), an up arrow (^), and a down arrow (v). To the right of the list are several input fields: "Name" (text box), "Description" (text box), "From" (text box with a dropdown arrow), "To" (text box with a dropdown arrow), "Subject" (text box with a dropdown arrow), and "Attachment" (dropdown menu). Below these fields is a large "Message" text area with a small icon in its top right corner. At the bottom of the dialog are two buttons: "OK" and "Cancel".

39 JavaScript

本部分旨在描述 JavaScript 如何在 LRH SW 应用程序中使用，而不是解释 JavaScript 语言。

LRH SW JavaScript 基于 ECMAScript 编程语言 <http://www.ecmascript.org>，如标准 ECMA-262 中所定义。

如果您熟悉 JavaScript，则可以在 LRH SW 中使用和 Web 浏览器中相同类型的命令。如果您对 ECMAScript 语言不熟悉，参考：

<https://developer.mozilla.org/en/JavaScript>

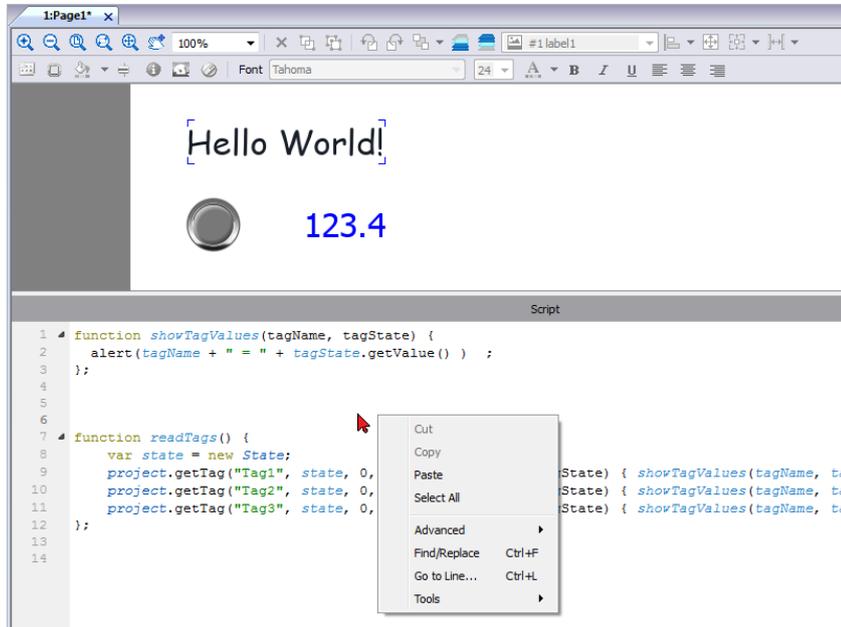
JavaScript 编辑器	457
JavaScript 功能的执行	457
事件	459
控件事件	460
页面事件	462
系统事件	463
对象	465
控件级对象	465
控件属性	466
控件函数	469
页面对象	470
页面对象属性	470
页面对象函数	471
项目对象	473
项目对象属性	473
项目对象函数	474
项目对象控件	482
打印报表对象	483
组对象	486
组对象函数	487
状态对象	488
状态对象函数	488
关键字	489
全局函数	489
处理读写文件	490

从 JavaScript 登录	493
在 JavaScript 控件操作的限制	495
JavaScript 调试	495

JavaScript 编辑器

LRH SW 有一个强大的 JavaScript 编辑器。

在编辑器中右击，显示可用的命令。



JavaScript 功能的执行

JavaScript 功能在事件发生时执行。例如，用户可以对鼠标点击事件定义脚本，JavaScript 脚本将在按钮在 HMI 设备上按动时执行。

JavaScript 功能只有在编辑的事件发生时执行，并且不是周期性的。该方法使 HMI 设备执行逻辑需要的系统开销最小化。

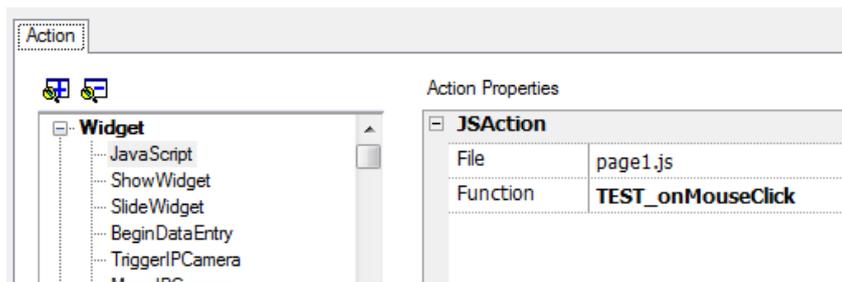
LRH SW 提供一个在客户端运行的 JavaScript 引擎。每一个项目页面可以包含仅限于当前页面的脚本；可以创建全局脚本由计划任务事件或报警事件执行。

这两种情况下，脚本在客户端运行。这意味着，如果不止一个客户端连接至 HMI 设备(针对运行 LRH SW Client 的外部计算机)，则每一个客户端都将运行相同的脚本，根据输入提供不同的输出结果，因为提供给不同客户端的输入可能不同。

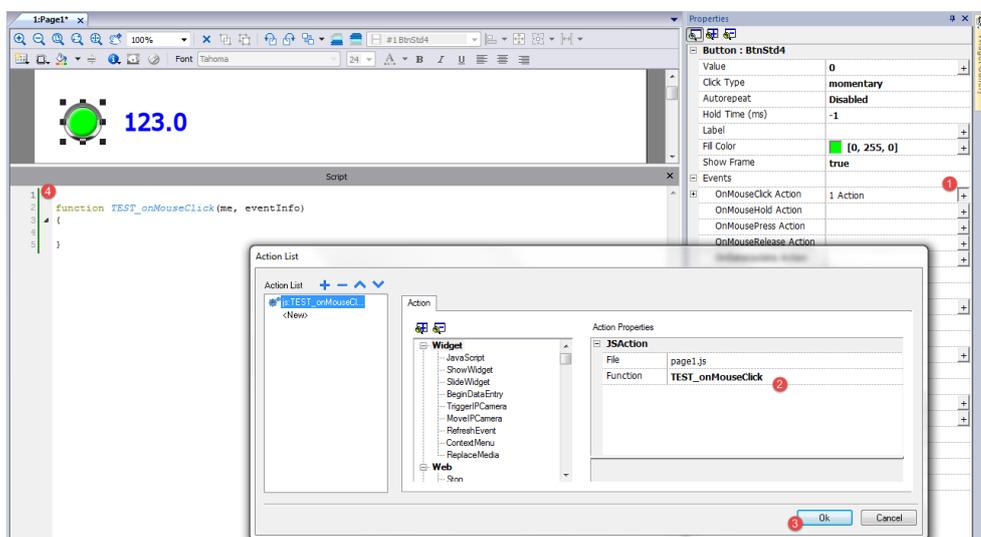
例如，如果一个脚本根据滑块的位置动作，该位置在不同的客户端不同，脚本的结果在不同的客户端也不同。

页面事件的 JavaScript 功能

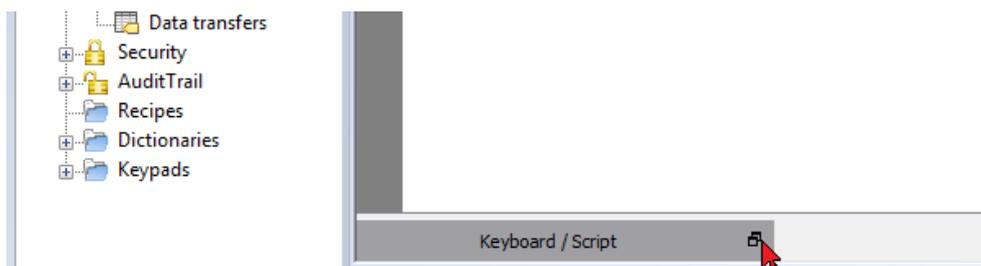
JavaScript 编辑器将在通过动作列表添加 JavaScript 动作时打开。



1. 选择将执行该动作的事件。
2. 从**控件**目录中添加 **JavaScript** 动作。
3. 留下默认功能名称或者键入一个新的名称。
4. 单击**确定**确认:JavaScript 编辑器显示您的函数结构。



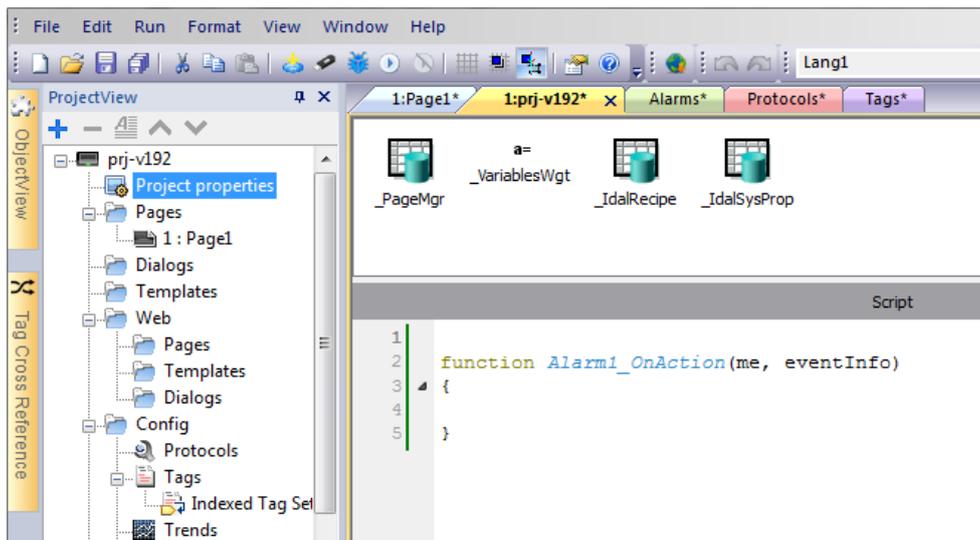
您也可以从位于工作区底部的**脚本**选项卡中打开 JavaScript 编辑器。



针对报警和计划事件的 JavaScript 功能

与报警和计划事件相关、与特定页面不相关的 JavaScript 代码，可以从主**项目属性**页面编辑。

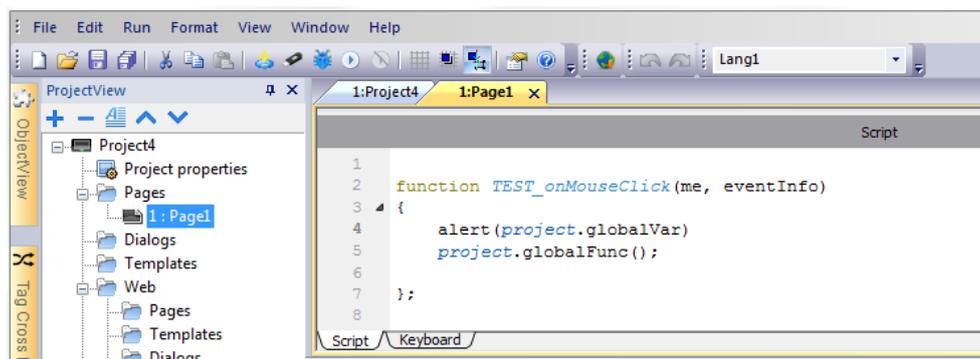
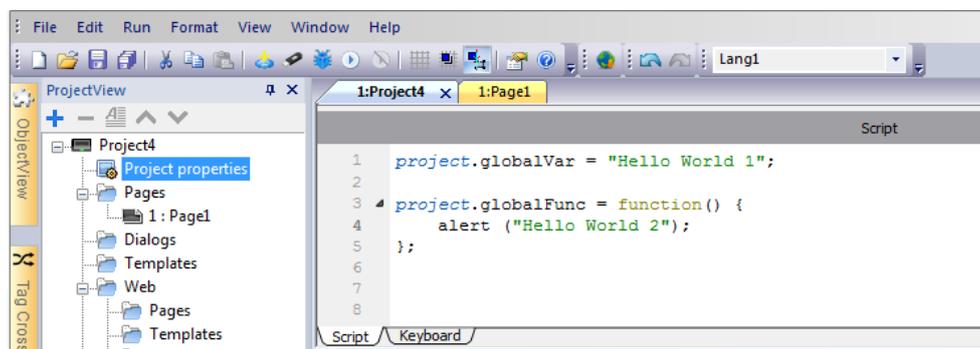
路径: **项目视图** > 双击**项目属性**



注：JavaScript 动作是客户端动作，因此只有登录客户端时才能执行。

共享的 JavaScript 码

project 全局变量可用于在页面间共享 JavaScript 代码。变量从主项目属性页面的主 JavaScript 代码创建/初始化，然后从项目页面使用。



事件

您可以添加 JavaScript 到以下种类的事件中：

- 控件事件
- 页面事件
- 系统事件

对于以下类型的事件：

- 按动鼠标
- 鼠标释放
- 鼠标点击
- 滚轮

JavaScript **eventInfo** 参数包含以下额外属性：

参数	说明
eventInfo.posX	本地鼠标/触摸控件坐标的X坐标
eventInfo.posY	本地鼠标/触摸控件坐标的Y坐标
eventInfo.pagePosX	页面X鼠标/触摸轴
eventInfo.pagePosY	页面Y鼠标/触摸轴
eventInfo.wheelDelta	鼠标滚轮 delta 。整数值，标记代表旋转方向。 实际值是每度八分之一的旋转量。最小的值取决于鼠标分辨率。通常这是 120，对应 15 度。

控件事件

onMouseClicked

`void onMouseClick (me, eventInfo)`

该事件仅对按钮有效，当按钮被快速按下和放开时事件发生。

参数	说明
me	对象启动事件
eventInfo	启动的事件细节

```
function buttonStd1_onMouseClicked(me, eventInfo) {
    //do something...
}
```

onMouseHold

`void onMouseHold (me, eventInfo)`

该事件仅对按钮有效，当按下按钮且在持续控件属性中设置的**保持时间**秒数后释放时出现此事件。

参数	说明
me	对象启动事件
eventInfo	启动的事件细节

```
function buttonStd1_onMouseHold(me, eventInfo) {
    //do something...
}
```

onMousePress

void onMousePress (me, eventInfo)

该事件仅对按钮有效，当按钮被按下时事件发生。

参数	说明
me	对象启动事件
eventInfo	启动的事件细节

```
function buttonStd1_onMousePress(me, eventInfo) {
    //do something...
}
```

onMouseRelease

void onMouseRelease (me, eventInfo)

该事件仅对按钮有效，当按钮被放开时事件发生。

参数	说明
me	对象启动事件
eventInfo	启动的事件细节

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    //do something...
}
```

onDataUpdate

boolean onDataUpdate (me, eventInfo)

该事件在控件上附属的数据更改时发生。

参数	说明
me	对象启动事件
eventInfo	字段对象(您可以用“.”点符号引用字段) oldValue = 更改前的控件值 newValue = 将对控件更新的值 attrName = 生成事件的属性 index = 整数特性索引(如果有的话), 默认=0 mode = W(当用户向控件写入时)。所有其他的状态为 R。

该事件在该值向其他控件传递时触发。**JavaScript** 代码可以拦截该事件并决定避免通过返回真值来更新控件。



注:如果事件上有额外的宏关联,所有宏将执行,无论 **JavaScript** 代码中使用的返回值如何。

```
function buttonStd1_onDataUpdate(me, eventInfo) {
if ( eventInfo.newValue > 100) {
//do something...
return true; // To avoid to continue and update
// the widget (e.g. not update the linked tag)
}
return false; // To continue and update the widget
// (e.g. update the linked tag)
}
```

页面事件

onActivate

void onActivate(me, eventInfo)

该事件在该页显示时发生。

参数	说明
me	对象启动事件
eventInfo	保留以供将来使用

JavaScript 将在页面活跃时执行,也就是页面被加载时。

```
function Page1_onActivate(me, eventInfo) {
//do something...
}
```

onDeactivate

```
void onDeactivate( me, eventInfo )
```

该事件在离开该页面时发生。

参数	说明
me	对象启动事件
eventInfo	保留以供将来使用

```
function Page1_onDeactivate(me, eventInfo) {
    //do something...
}
```

onWheel

```
void onMouseWheelClock( me, eventInfo )
```

该事件在滚轮设备移动时发生(例如,鼠标滚轮)。

参数	说明
me	对象启动事件
eventInfo	启动的事件细节

```
function Page1_onMouseWheelClock(me, eventInfo) {
    //do something...
}
```

系统事件

系统事件可能与如下相关:

- 计划任务
- 报警
- 滚轮设备

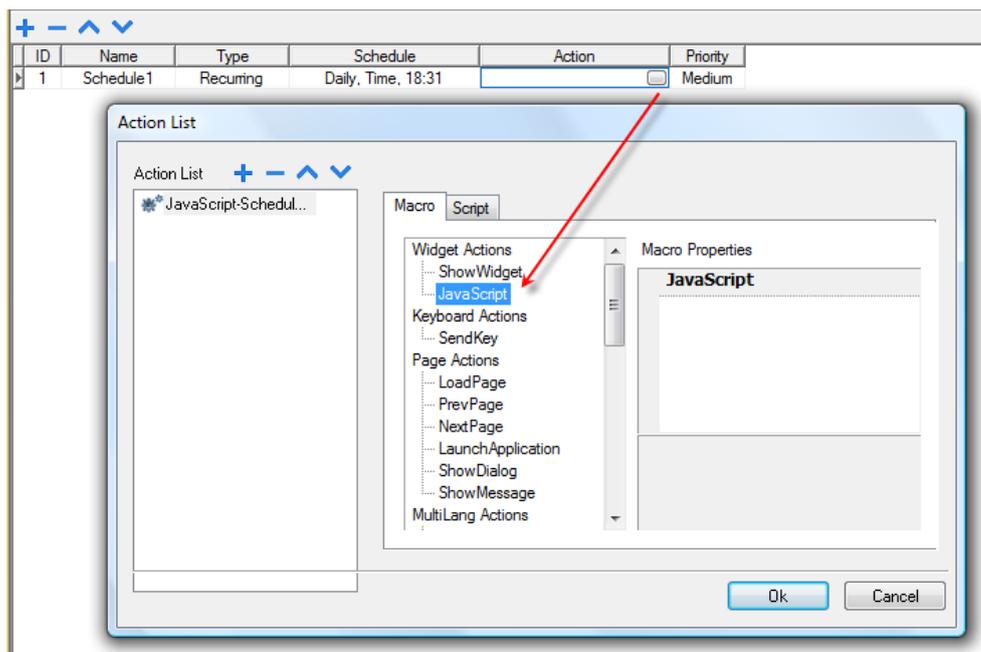


重要: 确保未在页面和项目级别 **JavaScript** 函数名称不重复。当出现冲突时,即当前页面和项目级别的两个功能有着相同的名称,系统在页面级别执行 **JavaScript** 回调。

当在当前页面未找到 **JavaScript** 回调时,系统自动在项目级别搜索。

计划任务事件

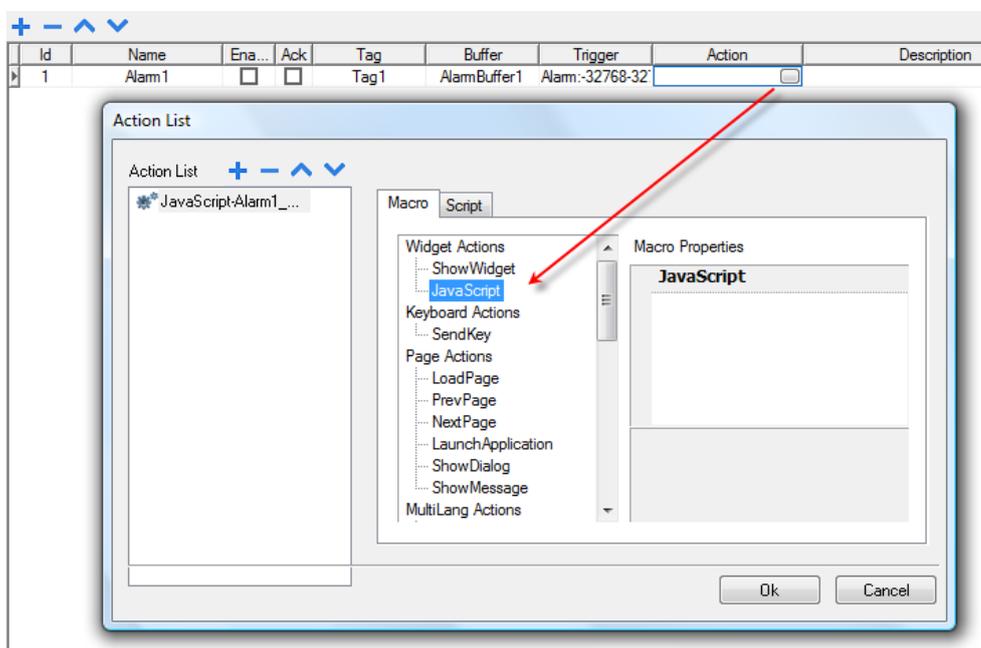
当被计划任务中相关的动作触发时,这些事件发生。



您可以从 **项目属性** 选项卡中编辑 JavaScript。

报警事件

当被相关的报警条件触发时，这些事件发生。



您可以从 **项目属性** 选项卡中编辑 JavaScript。

onWheel

```
void onMouseWheelClock( me, eventInfo )
```

该事件在滚轮设备移动时发生(例如，鼠标滚轮)。

参数	说明
me	对象启动事件
eventInfo	启动的事件细节

```
function Project1_onMouseWheelClock(me, eventInfo) {
    //do something...
}
```

对象

LRH SW 使用 JavaScript 对象访问页面元素。每一个对象由属性和函数组成，属性和函数用于定义页面元素的操作和外观。以下对象用于与 HMI 设备页面的元素交互：

对象	说明
控件	这是页面所有元素包括页面元素的基本级
页面	该对象参照当前 HMI 设备页面。页面是屏幕的顶层对象。
编组	该对象与一组标签相关，使得可以在一组逻辑相连的标签上进行统一的操作
项目	该对象定义了项目控件。项目控件用于检索关于项目的数据，例如标签、报警、配方、计划等等。项目只有一个控件，可以通过项目变量引用。
状态	该对象保持从控制的环境中获取的变量的状态。除了值本身，它还包含时间戳，指示收集值的时间并标记该值的质量。

控件级对象

该控件级是页面所有元素包括页面元素的基本级。

在这种情况下，控件不用于表明特定的屏幕对象而是一个 JavaScript 级。

用 JavaScript 更改控件属性

如果您想用 JavaScript 更改控件属性，请将控件属性 **静态优化** 设为 **动态**。



重要：如果控件属性静态优化未设为动态，对属性的更改将被忽略。

在调用 `getWidget` 失败时，远程调试器将报告以下错误：

“尝试访问静态优化的控件标签 1。停用控件静态优化以从脚本访问控件”。

使用如下编码，该错误将出现：

```
var wgt;
try {
wgt = page.getWidget('label1');
} catch(err) {
alert("" + err);
}
```

控件属性

这些属性是所有控件通用的。

objectName

string objectName

获取控件名称，一个唯一的 id。

```
function btnStd04_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    var name = wgt.objectName;
}
```

(可在网页上使用)

x

number x

获取或设置控件的 x 像素位置。

```
function btnStd1_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    wgt.x = 10;
}
```

(可在网页上使用)

y

number y

获取或设置控件的 y 像素位置。

```
function btnStd1_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    wgt.y = 10;
}
```

```
}
```

(可在网页上使用)

width

number width

获取或设置控件的控件宽度像素。

```
function btnStd1_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.width = 10;  
}
```

(可在网页上使用)

height

number height

获取或设置控件的控件高度像素。

```
function btnStd1_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.height = 10;  
}
```

(可在网页上使用)

visible

boolean visible

获取或设置控件可见状态。

```
function btnStd4_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.visible = false;  
}  
  
function btnStd5_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.visible = true;  
}
```

值

number value

获取或设置控件值。

```
function btnStd6_onMouseRelease(me) {  
    var wgt = page.getWidget("field1");  
    wgt.value = 100;  
}
```

opacity

number opacity (range from 0 to 1)

获取或设置控件不透明性。值是从 0-1 的十进制的，其中 1 是 100% 不透明的。

```
function btnStd8_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.opacity = 0.5;  
}
```

(可在网页上使用)

rotation

number rotation (in degrees)

为控件获取或设置旋转角度。向右旋转完成要转的度数，从东边的位置开始。

```
function btnStd9_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.rotation = 45;  
}
```

(可在网页上使用)

userValue

string userValue

获取或设置用户定义的控件值。该域可以被 JavaScript 功能用来存储额外的控件数据。

```
function btnStd9_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.userValue = "Here I can store custom data";  
}
```

每一个控件有一些特定的属性，您可以使用点符号访问。对于最新和详细的属性列表，您可以使用 JavaScript 调试器控件检查控件函数和属性。

控件函数

一些函数是所有控件通用的。

getProperty

```
object getProperty( propertyName, [index] )
```

返回一个属性。

参数	说明
propertyName	包含要获取的属性的名称的字符串
索引	要从数组获取的元素索引(默认 = 0)

在 LRH SW 的**属性**窗格中显示的几乎所有属性都可使用 `getProperty` 函数检索。索引值是可选的，仅用于支持数组的控件。

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    var shape = page.getWidget("rect2");
    var y_position = shape.getProperty("y");
}
```

```
function buttonStd2_onMouseRelease(me, eventInfo) {
    var image = page.getWidget("multistate1");
    var image3 = image.getProperty("imageList", 2);
    //...
}
```

(可在网页上使用)

setProperty

```
boolean setProperty( propertyName, value, [index] )
```

为控件设置属性。

参数

参数	说明
propertyName	包含要设置的属性的名称的字符串
值	包含要设置属性的值的字符串。
索引	要在数组中获取的元素索引(默认 = 0)

在 LRH SW 的**属性**窗格中显示的几乎所有属性都可使用此函数设置。索引值是可选的，仅用于支持数组的控件(例如，多状态图像控件)。`setProperty` 函数返回一个 **boolean** 值(`true` 或 `false`)，指示此属性是否经过设置。

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    var setting_result = shape.setProperty("y", 128);
    if (setting_result)
        alert("Shape returned to start position");
}
```

```
function buttonStd2_onMouseRelease(me, eventInfo) {
    var image = page.getWidget("multistate1");
    var result = image.setProperty("imageList", "Fract004.png", 2);
    //...
}
```

(可在网页上使用)

页面对象

该对象参照当前 HMI 设备页面。页面是屏幕的顶层对象。

页面对象属性

页面级别可用的属性。

backgroundColor

string backgroundColor (in format rgb(xxx, xxx, xxx) where xxx range from 0 to 255)

页面背景色。

```
function btnStd11_onMouseRelease(me) {
    page.backgroundColor = "rgb(128,0,0)";
}
```

(可在网页上使用)

width

number width

页面像素宽度。

```
function btnStd05_onMouseRelease(me) {
    var middle_x = page.width / 2;
}
```

(仅可在网页上获取)

height

number height

页面像素高度。

```
function btnStd05_onMouseRelease(me) {
    var middle_y = page.height / 2;
}
```

(仅可在网页上获取)

userValue

string userValue

获取或设置用户定义的控件值。该域可以被 JavaScript 功能用来存储额外的页面数据。

```
function btnStd9_onMouseRelease(me) {
    page.userValue = "Here I can store custom data";
}
```

(可在网页上使用)

页面对象函数

可以在页面级别使用的函数。

getWidget

object getWidget(wgtName)

返回指定名称的控件。

参数	说明
wgtName	包含控件名称的字符串

返回值

代表该控件的对象。如果该控件不存在，返回空。

```
function btnStd1_onMouseRelease(me) {
    var my_button = page.getWidget("btnStd1");
}
```

(可在网页上使用)

setTimeout

number setTimeout(functionName, delay)

启动计时器在指定延迟后调用指定功能。

参数	说明
functionName	包含要调用的功能的名称的字符串
延迟	延迟毫秒数

返回值

与计时器 ID 对应的数。

```
var duration = 3000;
var myTimer = page.setTimeout("innerChangeWidth()", duration);
```

(可在网页上使用)

clearTimeout

```
void clearTimeout( timerID )
```

使用指定的计时器停止和清除超时计时器。

参数	说明
timerID	要被清除和停止的计时器

```
var duration = 3000;
var myTimer = page.setTimeout("innerChangeWidth()", duration);
// do something
page.clearTimeout(myTimer);
```

(可在网页上使用)

setInterval

```
number setInterval( functionName, interval )
```

启动在给定间隔执行指定功能的计时器。

参数	说明
functionName	包含要调用的功能的名称的字符串
interval	间隔毫秒数

返回值

与计时器 ID 对应的数。

```
var interval = 3000;
var myTimer = page.setInterval("innerChangeWidth()", interval);
```

(可在网页上使用)

clearInterval

```
void clearInterval( timerID )
```

停止和清除使用指定计时器的间隔计时器。

参数	说明
timerID	要被清除和停止的计时器

```
var interval = 3000;
var myTimer = page.setInterval("innerChangeWidth()", interval);
// do something
page.clearInterval(myTimer);
```

(可在网页上使用)

clearAllTimeouts

```
void clearAllTimeouts()
```

清除所有启动的计时器。

```
page.clearAllTimeouts();
```

(可在网页上使用)

项目对象

该对象定义了项目控件。项目控件用于检索关于项目的数据，例如标签、报警、配方、计划等等。项目只有一个控件，可以通过项目变量引用。

项目对象属性

在项目级别设置的属性。

startPage

```
string startPage
```

当项目启动时显示的页面。

```
var startPage = project.startPage;
project.startPage = "Page2.jmx";
```

项目对象函数

在项目级别要使用的函数。

nextPage

```
void nextPage()
```

执行下一页动作。

```
project.nextPage();
```

(可在网页上使用)

prevPage

```
void prevPage()
```

执行前一页动作。

```
project.prevPage();
```

(可在网页上使用)

lastVisitedPage

```
void lastVisitedPage()
```

执行上一次访问页面动作。

```
project.lastVisitedPage();
```

(可在网页上使用)

homepage

```
void homePage()
```

执行主页动作。

```
project.homePage();
```

(可在网页上使用)

loadPage

```
void loadPage(pageName)
```

加载脚本中定义的设置页面。

```
project.loadPage("Page5.jmx");
```

(可在网页上使用)



警告:如页面发生变化,系统将强制删除所有活动时间事件,并且 **JavaScript** 程序将一直运行,直到切换到新页面。

显示对话框

```
void showDialog(pageName)
```

显示对话框页面。

```
project.showDialog("Dialog.jmx");
```

(可在网页上使用)

closeDialog

```
void closeDialog()
```

关闭当前打开的对话框页面。

```
project.closeDialog();
```

(可在网页上使用)

showMessage

```
void showMessage( message )
```

显示消息。

```
project.showMessage("Hi This is test message");
```

(可在网页上使用)

getGroup

```
number getGroup( groupName, groupInstance, [callback] )
```

快速读取函数;将所有标签的值放在一组。

参数	说明
groupName	<p>包含组名称的字符串。</p> <p>AND/OR 表达式支持从多个组检索标签列表。</p> <p> OR 操作</p> <p>& AND 操作</p> <p>(...) 括号可用于定义如何计算表达式</p> <p>示例：</p> <ul style="list-style-type: none"> • project.getGroup("one", group); • project.getGroup("(one two)", group); • project.getGroup("((one&two)*three)", group);
groupInstance	要填入的组元素
callback	包含当组就绪时要调用的函数的名称的字符串

返回值

可表示状态的数字值：1 表示成功，0 表示失败。

```
var group = new Group();
var status = project.getGroup ("enginesettings", group);
if (status == 1) {
    var value = group.getTag("Tag1");
    if (value!=undefined) {
        // do something with the value
    }
}
```

```
var g = new Group();
var status = project.getGroup ("enginesettings", g,
    function (groupName, group) { fnGroupReady(groupName, group);} );

function fnGroupReady(groupName, group) {
    var val = group.getTag("Tag1");
    if (val!=undefined) {
        // do something with the value
    }
}
```

(可在网页上使用)

getTag

object getTag(tagName, state, index, forceRefresh)

```
void getTag( tagName, state, index, callback, forceRefresh)
```

如果指定标签名称的索引值是 -1, 会返回标签值或完整的数组。

参数	说明
标签名称	标签名称
state	要填入的状态元素
索引	如果标签是数组类型, 则建立索引。-1 返回完整数组。默认 = 0。
callback	如果需要异步读取, 则是函数名称。默认 = ""。
forceRefresh	(可选参数) true= Runtime 将直接从设备读取标签的一个更新值。默认为 false。

返回值

标签值返回。如果标签是数组类型, 索引 = -1, 则完整的数组被返回。对于非数组标签, 提供索引为 0。

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
//
//for non array type
//tags index is not considered, so can be left as 0
//
if (value!=undefined) {
//...do something with s
}
```

```
var state = new State();
project.getTag("Tag1", state, -1,
    function(tagName, tagState) { fnTagReady(tagName, tagState); });
function fnTagReady(tagName, tagState) {
    if (tagName=="Tag1") {
        var myValue = tagState.getValue();
    }
}
```

(可在网页上使用)

setTag

number setTag(tagName, tagValue, [index], [forceWrite])

在项目中设置给定标签。名称和值在字符串中。

参数	说明
标签名称	标签名称
tagValue	要写入的值
索引	如果标签是数组类型，则建立索引。-1 传递完整数组。默认 = 0。
forceWrite	启动标签强制写入布尔值，该函数在返回前等待该值写入。默认 = false。

返回值

整数值表示成功，当强制写入为 **true** 时，动作失败。0 意思是成功，-1 意思是失败。如果强制写入是 **false**，返回值将会是未定义的。

```
var val = [1,2,3,4,5];
var status = project.setTag("Tag1", val, -1, true);
if (status == 0) {
    // Success
} else {
    // Failure
}
```

```
var val = "value";
project.setTag("Tag1", val);
```

(可在网页上使用)

updateSystemVariables

```
void project.updateSystemVariables()
```

强制系统变量刷新。

```
project.updateSystemVariables()
```

selectAllAlarms

```
void project.selectAllAlarms(bool selected)
```

选择/未选择所有报警

```
project.selectAllAlarms(true)
```

(可在网页上使用)

ackAlarms

```
void project.ackAlarms()
```

确认所有选择的报警

```
project.selectAllAlarms(true);
```

```
project.ackAlarms();
project.selectAllAlarms(true);
```

(可在网页上使用)

resetAlarms

```
void project.resetAlarms()
```

复位所有选择的报警

```
project.selectAllAlarms(true);
project.resetAlarms();
project.selectAllAlarms(true);
```

(可在网页上使用)

启用报警

```
void project.enableAlarms()
```

启用所有选择的报警

```
project.selectAllAlarms(true);
project.enableAlarms();
project.selectAllAlarms(true);
```

(可在网页上使用)

getRecipeItem

```
object getRecipeItem (recipeName, recipeSet, recipeElement)
```

设置指定配方集元素的值。

参数	说明
recipeName	代表配方名称的字符串
recipeSet	代表配方集的字符串，可以是配方集的名称或是基于0的集索引。
recipeElement	代表配方元素的字符串，可以是元素的名称或是基于0的元素索引。

返回值

带有配方值的对象。如果无效，未定义被返回。如果是类型数组，队列对象类型被返回。

```
var value = project.getRecipeItem("recipeName", "Set", "Element");
```

setRecipeItem

```
number setRecipeItem (recipeName, recipeSet, recipeElement, value )
```

设置指定配方集元素的值。

参数	说明
recipeName	代表配方名称的字符串
recipeSet	代表配方集的字符串，可以是配方集的名称或是基于0的集索引。
recipeElement	代表配方元素的字符串，可以是元素的名称或是基于0的元素索引。
值	要在配方中存储的值。可以是数组类型。

返回值

整数值表示成功或失败。“0”意思是成功，-1意思是失败。

```
var val = [2,3,4];
project.setRecipeItem("recipeName", "Set", "Element", val);
if (status == 0) {
    // Success
} else {
    // Failure
}
```

downloadRecipe

void downloadRecipe (recipeName, recipeSet)

下载配方集到相应的标签。

参数	说明
recipeName	代表配方名称的字符串
recipeSet	代表配方集的字符串，可以是配方集的名称或是基于0的集索引。

```
project.downloadRecipe("recipeName", "Set");
```

uploadRecipe

void uploadRecipe (recipeName, recipeSet)

将标签值上传到提供的配方集中。

参数	说明
recipeName	代表配方名称的字符串
recipeSet	代表配方集的字符串，可以是配方集的名称或是基于0的集索引。

```
project.uploadRecipe("recipeName", "Set");
```

launchApp

```
void launchApp( appName, appPath, arguments, singleInstance)
```

执行外部应用程序。

参数	说明
appName	包含应用程序名称的字符串
appPath	包含应用程序绝对路径的字符串
变量	包含要发送到应用程序的参数的字符串
singleInstance	true=仅允许单一示例, false=允许多个示例

请注意, 路径名的语法取决于本机 OS 格式(请参阅 "HMI 设备地址" 在本页 525)。

在 Linux 设备上, 路径名的语法需要斜杠字符(或允许使用双斜杠字符)。

```
project.launchApp
("pdfViewer", "/mnt/data/hmi/qthmi/deploy", "/mnt/usbmemory/test.pdf", "true");
```

getClientType

```
字符串 getClientType()
```

返回客户端类型

客户端类型	说明
本地	在 HMI 设备上运行
远程	在 LRH SW Client 客户端上运行
Web	在 Web 客户端上运行

```
var clientType = project.getClientType();
if (clientType=="web") {
    // Currently running on web client
} else if (clientType=="remote") {
    // Currently running on LRH SW Client
} else if (clientType=="local") {
    // Currently running on HMI Device
}
```

(可在网页上使用)

登录

```
int project.login("username", "password")
```

使用指定凭证访问系统

```
var ReplyCode;
ReplyCode = project.login("admin", "admin");
if (ReplyCode != 0) {
    alert("Access denied");
}
```

返回值

0	无错误
1	错误:您未经授权。
2	错误:Runtime连接中断。
3	错误:输入的用户名或密码不正确
4	错误:输入的密码不正确
5	错误:无法执行行动
6	错误:密码不匹配
7	错误:密码长度太短
8	错误:密码必须包含数字
9	错误:密码必须包含特殊字符
10	错误:密码必须不同于之前密码
11	错误:用户已存在
12	错误:密码不能为空
13	错误:您的密码已过期
14	警告:您的密码将过期

注销

```
project.logout(AllowDefaultUser)
```

退出系统

```
project.logout();           // Logout even from default user
project.logout(true);       // Logout even from default user
project.logout(false);      // Logout only if not logged as default user
```

项目对象控件

getCurrentPageName

```
string getCurrentPageName()
```

返回当前活动页面的名称

```
// Get PageMgr widget
var pageMgr = project.getWidget( "_PageMgr" );

// Show Current Page
var currentPageName = pageMgr.getCurrentPageName();
project.showMessage( "Current active page is: " + currentPageName );
```

(可在网页上使用)

hasPage

boolean hasPage(string pageName)

如果页面存在, 返回 **true**, 否则返回 **false**

```
// Get PageMgr widget
var pageMgr = project.getWidget( "_PageMgr" );

//Page exists
var pageExists = pageMgr.hasPage( "Page10" );
if (pageExists) {
    project.showMessage( "Page10 exists" );
} else {
    project.showMessage( "Hei Page10 not exists!");
}
```

(可在网页上使用)

curLangCode

字符串 curLangCode

MultiLangMgr 控件属性。包括有效的语言代码。

```
// Get MultiLangMgr widget
var MultiLangMgr = project.getWidget( "_MultiLangMgr" );

// Show curLangCode
var curLangCode = MultiLangMgr.curLangCode;
project.showMessage( "Current active language is: " + curLangCode );
```

打印报表对象

printGfxReport

void printGfxReport(reportName, silentMode)

打印报表名称指定的图形报表。

参数	说明
reportName	包含报表名称的字符串
silentMode	True = 无声模式已启用。无打印机设置对话框显示。

```
project.printGfxReport("Report Graphics 1", true);
```

emptyPrintQueue

```
void emptyPrintQueue();
```

清空打印队列。当前任务不会取消。

```
project.emptyPrintQueue();
```

pausePrinting

```
void pausePrinting();
```

中止打印操作。将不会中止已经发送到打印机的页面打印。

```
project.pausePrinting();
```

resumePrinting

```
void resumePrinting();
```

继续先前暂停的打印。

```
project.resumePrinting();
```

abortPrinting

```
void abortPrinting();
```

取消当前打印操作，继续队列中的下一个。该命令将不会取消已经发送到打印机的页面打印。

```
project.abortPrinting();
```

printStatus

```
project.printStatus;
```

返回代表当前打印状态的字符串。

状态字符串	说明
error	打印时出错
printing	正在进行的打印
idle	系统准备就绪来接受新任务
paused	打印已中止

```
var status = project.printStatus;
project.setTag("PrintStatus",status);
```

printGfxJobQueueSize

```
project.printGfxJobQueueSize;
```

返回队列要打印的图形报表的数量。

```
var gfxqueuesize = project.printGfxJobQueueSize;
project.setTag("printGfxJobQueueSize",gfxqueuesize);
```

printTextJobQueueSize

```
project.printTextJobQueueSize;
```

返回队列要打印的文本报表的数量。

```
var textjobqueuesize = project.printTextJobQueueSize;
project.setTag("printTextJobQueueSize",textjobqueuesize);
```

printCurrentJob

```
project.printCurrentJob;
```

返回代表当前打印任务的字符串

```
var currentjob = project.printCurrentJob;
project.setTag("printCurrentJob",currentjob);
```

printActualRAMUsage

```
project.printActualRAMUsage;
```

返回打印队列 RAM 使用量的估算值

```
var myVar = project.printActualRAMUsage;
alert(" actual ram usage is "+ myVar);
```

printRAMQuota

```
project.printRAMQuota;
```

返回打印队列最大允许的 RAM 使用量

```
var ramquota = project.printRAMQuota;  
project.setTag("printRAMQuota", ramquota);
```

printActualDiskUsage

```
project.printActualDiskUsage;
```

返回卷盘文件夹磁盘使用量(对PDF打印输出)

```
var myVar1 = project.printActualDiskUsage;  
alert(" actual disk usage is "+ myVar1);
```

printDiskQuota

```
project.printDiskQuota;
```

返回卷盘文件夹最大允许大小(对PDF打印输出)。

```
var ramquota = project.printRAMQuota;  
var diskquota = project.printDiskQuota;
```

printSpoolFolder

```
project.printSpoolFolder;
```

返回当前卷盘文件夹路径(对PDF打印输出)。

```
var spoolfolder = project.printSpoolFolder;  
project.setTag("printSpoolFolder", spoolfolder);
```

printPercentage

```
project.printPercentage;
```

返回当前任务完成百分比(只有在多页图形报表中有意义)

```
var percentage = project.printPercentage;  
project.setTag("printPercentage", percentage);
```

组对象

一组合乎逻辑的标签联合起来的基本逻辑元素。

组对象函数

组对象使用的函数。

getTag

```
object getTag( TagName )
```

从组对象获取标签名称指定的标签。

参数	说明
标签名称	代表标签名称的字符串

返回值

一个是标签值，如果标签是一个数组，则是整个数组的值。如果需要检索数组元素，请检查 `getTag` 在项目对象中是否可用。如果标签无效，未定义被返回。

```
var group = new Group();
project.getGroup("GroupName", group);
var value = group.getTag("Tag1");
```

(可在网页上使用)

getCount

```
number getCount()
```

返回组中标签的全部数量。

```
var group = new Group();
project.getGroup("GroupName", group);
var value = group.getCount();
```

(可在网页上使用)

getTags

```
object getTags()
```

返回组中所有标签的列表。

```
功能 {
var group = new Group();
project.getGroup("enginesettings", group);
var tagList = group.getTags();
for(var i = 0; i < tagList.length; i++){
    var tagName = tagList[i];
    //do something...
```

```
};
```

(可在网页上使用)

状态对象

这从控制的环境中获取标签的状态。

状态对象函数

要与状态对象使用的函数。

getQualityBits

```
number getQualityBits()
```

返回一个整数—表明标签值质量的位结合。

```
var state = new State();  
var value = project.getTag("Tag1", state, 0);  
var qbits = state.getQualityBits();
```

(可在网页上使用)

getTimestamp

```
number getTimestamp()
```

返回标签值取样时间。

返回值

包含时间戳的数(例如1315570524492)。



注:日期是本地 JavaScript 数据类型。

```
var state = new State();  
var value = project.getTag("Tag1", state, 0);  
var ts = state.getTimestamp();
```

isQualityGood

```
boolean isQualityGood()
```

返回该状态对象中包含的值是否可靠。

返回值

如果质量好,布尔量为 **true**, 否则为 **false**。

```

var state = new State();
var value = project.getTag("Tag1", state, 0);
if (state.isQualityGood()) {
    // do something...
}

```

(可在网页上使用)

关键字

全局对象被预定义，可以通过如下名称引用。

page

对象页面

对当前页面引用页面对象。

```

function btnStd04_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    var name = wgt.objectName;
}

```

project

object project

引用项目控件

```

var group = new Group();
project.getGroup("GroupName", group);
var value = group.getCount("Tag1");

```

全局函数

print

void print(message)

向 HMI 登录窗口打印一条消息。

参数	说明
消息	消息字符串

```
print("Test message");
```

alert

```
void alert( message )
```

显示带给定消息的弹出对话框。用户必须按对话框中的**确定**按钮才能继续执行脚本。

参数	说明
消息	消息字符串



注：报警功能可用于调试 JavaScript 函数。

```
alert("Test message");
```

(可在网页上使用)

处理读写文件

Create folder

```
boolean fs.mkdir(strPath);
```

如果不存在，在指定的路径中创建文件夹。成功则返回 **true**，失败则返回 **false**。

参数	说明
strPath	路径

Remove folder

```
boolean fs.rmdir(dirPath);
```

如果存在并为空，移除目录。成功则返回 **true**，失败则返回 **false**。

参数	说明
dirPath	文件夹字符串

Read folder content

```
object fs.readdir(dirPath);
```

读取文件夹内容。返回文件夹中文件名称的数组，排除“.”和“..”。如果失败，返回空白列表。

参数	说明
dirPath	文件夹字符串

Read file

```
object fs.readFile(strfile [, strFlag]);
```

以读取模式打开“strFile”文件，读取内容并返回。

参数	说明
strFile	文件名字符串
strFlag	读取文件模式： “b”意味着读取和返回二进制文件(否则返回文本文件)

Write file

```
fs.writeFile(strFile, fileData, [strFlag]);
```

如果文件不存在，创建“strFile”文件。以写入模式打开字符串文件，向文件写入数据。

参数	说明
strFile	文件名字符串
fileData	字节数组，文件要写入的数据
strFlag	写入文件模式： <ul style="list-style-type: none"> • “a”: 将文件数据附加到文本文件末尾 • “r”: 用文件数据代替文件内容 • “ab”: 附加文件数据到二进制文件的结尾 • “rb”: 用文件数据代替二进制文件的内容

默认标签用于附加和写入模式写入文本文件。文件路径如果没有的话将被创建。

如果写入错误，返回 -1。

Append file

```
int fs.appendFile(strFile, fileData);
```

如果文件不存在，请创建它，否则附加到现有文件。返回写入的字符数，出错时返回 -1。

参数	说明
strFile	文件名字符串
fileData	字节数组，文件要写入的数据

File exists

```
boolean fs.exists(strPath)
```

如果文件或文件夹在指定路径存在，返回 true。

参数	说明
strPath	路径

Remove file

```
boolean fs.unlink(strPath)
```

如果存在，从文件系统移除 **strPath** 的给定文件。成功则返回 **true**，失败则返回 **false**。

参数	说明
strPath	路径

File status

```
object fs.stat(strPath)
```

获取在指定路径上存在的文件/文件夹的信息。

参数	说明
strPath	文件/文件夹路径字符串

```
var fileStats = var fs.stat(strPath)
```

fileStats.isFile	如果路径是一个文件则为 true
fileStats.isDir	如果路径是一个文件夹则为 true
fileStats.size	该文件的字节大小
fileStats.atime	代表上一次读取访问时间的日期对象
fileStats.mtime	代表上一次写入访问时间的日期对象
fileStats.ctime	代表创建时间的日期对象
fileStats.perm	文件许可

如果路径无效，则 **isFile** 和 **isDir** 字段返回 **false**。

文件许可表格

0x4000	文件对文件所有者可读
0x2000	文件对文件所有者可写入
0x1000	文件对文件所有者可执行
0x0400	文件对用户可读
0x0200	文件对用户可写入
0x0100	文件对用户可执行
0x0040	文件对小组可读
0x0020	文件对小组可写入

0x0010	文件对小组可执行
0x0004	文件对任何人可读
0x0002	文件对任何人可写

文件处理方面的重要注解

文件和文件夹的路径预期是 **UNIX** 风格。这意味着反斜线符号字符 (\) 不能被识别。使用斜线符号 (/) 代替。

文件系统对象是客户端对象。因此操作都是在本地文件系统上执行，而不是在服务器文件系统。

当前要获得设备文件系统访问 **JavaScriptAPI** 是控制小文件。当文件被读取，整个文件内容被暂时存储在 **JavaScript** 环境 (16MB) 可用的 **RAM** 中，当没有足够的存储空间时例外。好的编程操作是在 **try/catch** 块中包括 `fs.readFile()` 调用。

从 JavaScript 登录

使用 `project.login()` 和 `project.logout()` 函数，用户可以自动从远程设备登录。使用徽章读取器设备读取用户徽章来执行登录这可能有用。

本章显示如何配置应用程序的示例以管理通过远程设备登录。

该应用程序必须具有默认用户

由于该项目的函数仅在应用程序处于活动状态时才有效，因此应用程序必须由可能具有只读权限的默认用户启动。通过读取徽章，可将该应用程序切换到具有额外权限的用户。之后，注销命令将重新激活不具有任何特定权限的默认用户

在以下示例中，我们使用三个标签与远程设备进行通信：

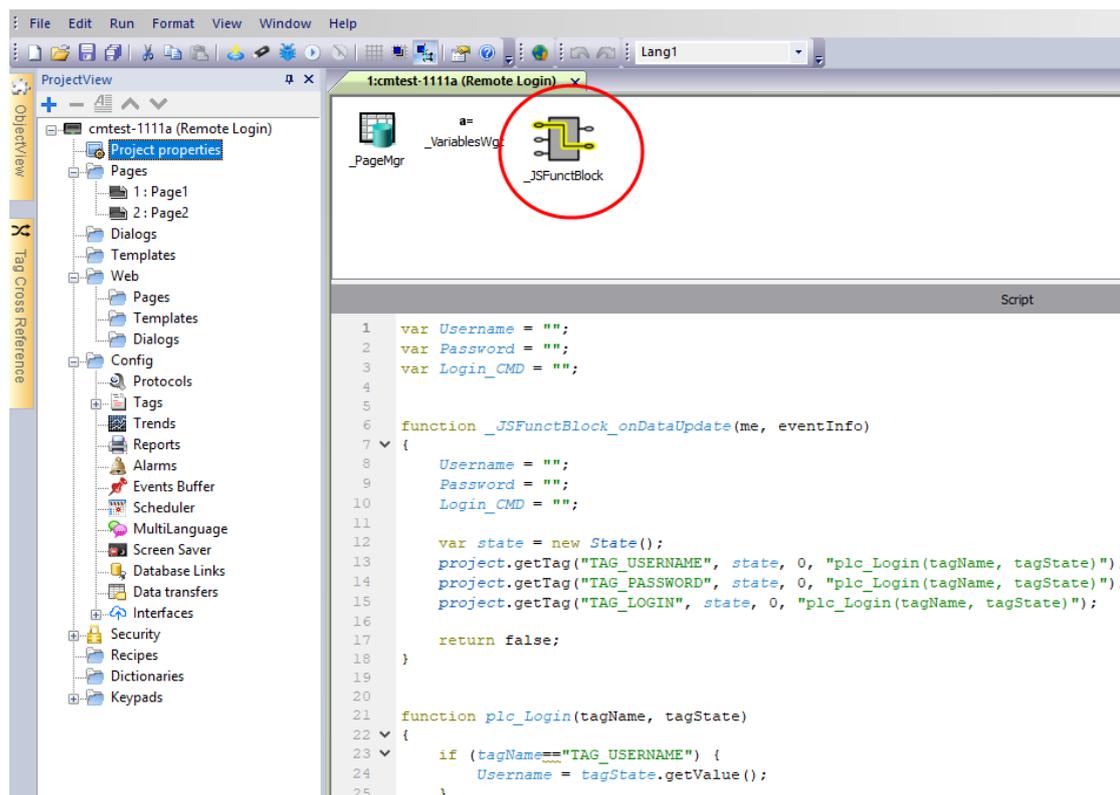
- TAG_USERNAME
- TAG_PASSWORD
- TAG_LOGIN

TAG_LOGIN 将是要执行的命令代码。

远程设备必须填充 TAG_USERNAME 和 TAG_PASSWORD 参数，然后使用所需的登录或注销命令填充 TAG_LOGIN 参数。HMI-RUNTIME 上的引擎将检测到 TAG_LOGIN 更改并执行所需命令，然后将 TAG_LOGIN 重置为其闲置状态。

TAG_LOGIN 命令	
0	闲置
1	登录请求
2	注销请求

在项目级别上，我们必须添加 **JavaScript** 功能块以检测 TAG_LOGIN 何时发生更改。附加在 **JavaScript function** 功能块的 `OnDataUpdate` 动作上的 **JavaScript** 代码将执行所需的登录/注销命令。



附加在 OnDataUpdate 动作上的 JavaScript 代码

```

var Username;
var Password;
var Login_CMD;

function _JSFuncBlock_onDataUpdate(me, eventInfo)
{
    Username = "";
    Password = "";
    Login_CMD = "";

    var state = new State();
    project.getTag("TAG_USERNAME", state, 0, "plc_Login(tagName, tagState)");
    project.getTag("TAG_PASSWORD", state, 0, "plc_Login(tagName, tagState)");
    project.getTag("TAG_LOGIN", state, 0, "plc_Login(tagName, tagState)");
    return false;
}

function plc_Login(tagName, tagState)
{
    if (tagName=="TAG_USERNAME") {
        Username = tagState.getValue();
    }
    if (tagName=="TAG_PASSWORD") {
        Password = tagState.getValue();
    }
    if (tagName=="TAG_LOGIN") {

```

```

    Login_CMD = tagState.getValue();
}

if (Username!="" && Password!="" && Login_CMD!="") {
    if (Login_CMD==1) {
        Reply = project.login(Username, Password);
    };
    if (Login_CMD==2) {
        Reply = project.logout(false); // Logout only if not logged as default
user
    };
    project.setTag("TAG_LOGIN", 0);
    project.setTag("TAG_REPLY", parseInt(Reply));
}
}

```

另请参阅：

- "登录" 在本页 481

在 JavaScript 控件操作的限制

控件不能由 JavaScript 实例化，您只能对其进行访问和更改。如果页面上需要更多控件，您可以在页面上添加隐藏控件，然后使用 JavaScript 对其进行显示和定位。

JavaScript 调试

LRH SW 和 LRH SW HMI Runtime 中包含 JavaScript 调试器。

有两种类型的调试装置：

- **runtime 调试装置**：直接在 HMI 设备上运行的调试装置
- **远程调试器**：在通过以太网与 HMI 设备连接的远程计算机上运行的调试器（通常是运行 LRH SW 的计算机）

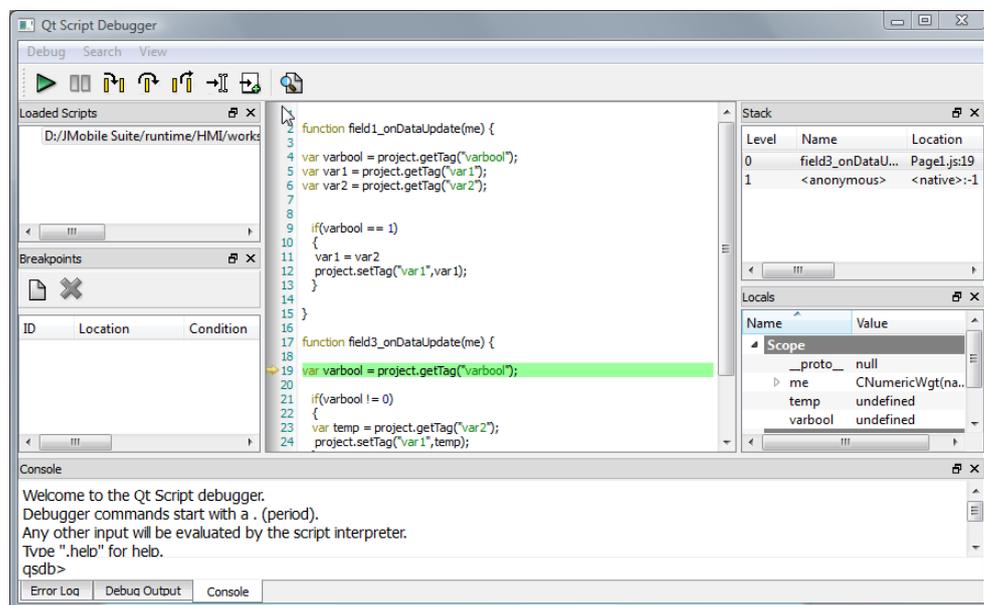
启用调试

在页面的属性窗格中，将 **JavaScript 调试** 设为 **true**。

Project Widget		Page	
Id	Project	Id	Page1
Full Path		Width	1024
Version		Height	768
Context Menu	on delay	Background	<input type="checkbox"/> [255, 255, ;
Developer Tools	false	Template	none
Keyboard	true	Static File Type	png
JavaScript Debug	true	JavaScript Debug	true
Allow JavaScript Remote	true		

对于计划任务和报警调试，启用项目属性中的 JavaScript 调试。

在 LRH SW HMI Runtime 中，调用事件时，调试器将显示调试信息。在本地窗格中，您可以查看所有变量和元素。



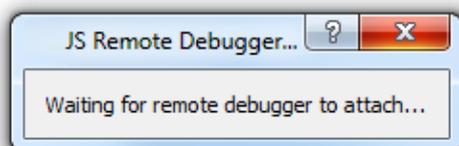
要了解 JavaScript 调试器的完整参考指南，查看：

<http://qt-project.org/doc/qt-4.8/qtscriptdebugger-manual.html>

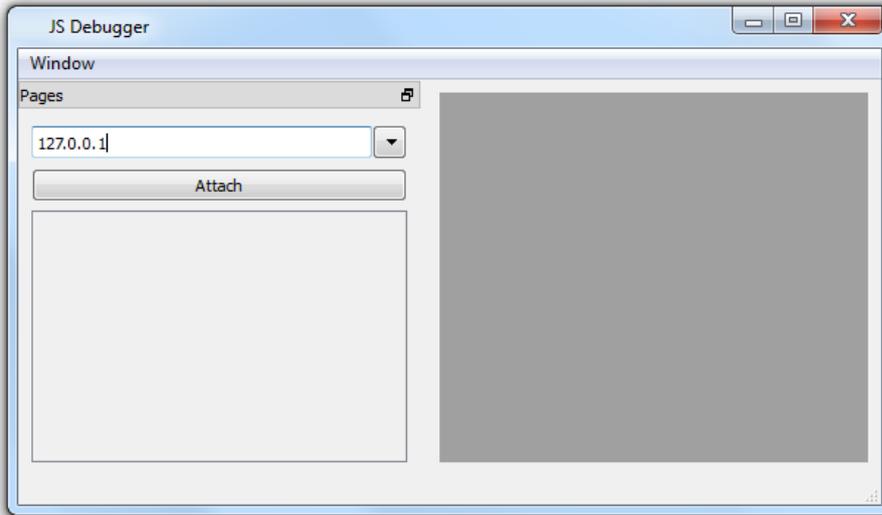
远程 JavaScript 调试装置

路径：运行 > 启动 JS 远程调试器

1. 在需要调试的所有页面中，将项目属性中的允许 JavaScript 远程和 JavaScript 调试参数设为 true。
2. 下载项目：以下信息显示在 runtime 上。



3. 在 **JS 调试器** 窗口中，选择 HMI 设备的 IP，然后单击**附加**将调试器连接至 HMI 设备。



远程 JavaScript 调试器使用端口 5100/TCP 连接至 LRH SW HMI Runtime。



注：远程 JavaScript 调试工具在 LRH SW Client 中不受支持。

JavaScript 内存使用

内存超过最大值时，将报出内存不足异常和自定义消息。请注意，我们无法精确控制实际内存使用，因此这基本是一个软限制。此外，我们无法禁止分配（这将中断引擎），因此仅当内存已经超过限制时报出异常。引发此异常前，强制回收站回收以查看是否可以释放一些内存。

JavaScript 内存限制可从全局对象 **\$EngineMemory** 中访问。默认为 16MB，应足以满足典型 JavaScript 使用（主要是控制，无多个分配）。

- **\$EngineMemory.setLimit()**
设置 JavaScript 允许的最大内存（默认限制为 0x00FFFFFF）
- **\$EngineMemory.getLimit()**
获得 JavaScript 允许的最大内存
- **\$EngineMemory.getSize()**
获得 JS (fastMallocStat) 当前已用内存

测试内存异常

要生成和测试内存异常，可以使用下面的代码。请注意，我们需要将内存限制复位为 0xffffffff 才能运行提醒，否则弹出提醒的所需的内存分配将失败。

```
try
{
    // Generate out-of-memory error
    var a = [];
    while(1)
    {
        a.push("a");
    };
} catch(e)
{
    // Ensure there is enough memory to pop up error message
    $EngineMemory.setLimit(0xffffffff);
    alert("Exception: " + e);
};
```

40 处理手势

一些控件可检测并管理平移和缩放手势。

- 趋势(请参阅 ["趋势控件手势"](#) 在本页 270 了解详细信息)
- 报警控件
- 组合框控件
- 表格控件
- PDF查看器
- 手势控件。特殊控件旨在自定义手势事件处理(请参阅 ["手势控件"](#) 在本页 402 了解详细信息)

对于基于表格显示的控件，当 **滚动条类型** 参数设为“手势”时，平移手势用于流畅滚动表格。

- 报警
- 控制列表



警告: 缩放和旋转手势需要两根手指操作。它们仅适用于支持多点触摸操作的 HMI 设备。(请参阅 ["HMI 设备地址"](#) 在本页 525)



提示: 凭借使用多点触摸 HMI 设备，可通过编程只有同时按下两个按钮才能执行的命令来实现安全命令。



PAN



PINCH



ROTATE

41 网页连接

LRH SW 使用户可以从计算机或移动设备(如平板电脑或手机)上运行的远程 Web 浏览器中访问 HMI 项目。使用 LRH SW, 用户可以创建一个显示在远程位置的 Web 项目, 画面与 HMI 设备上显示的图形相同。LRH SW 项目基于 HTML5 技术, 这意味着无需任何插件或外部软件即可显示信息。

假定您基本了解如何在移动设备上操作网络浏览器, 以及如何与服务器运行的 HMI 设备建立连接。例如, 如果您使用平板电脑或手机设备访问 HMI 设备上的 LRH SW 页面, 则必须知道如何建立 Wi-Fi 访问。

支持的平台和浏览器	502
Web 页面	502
测试 Web 项目	504
下载网页项目	504
网络连接问题	506
网络支持的功能	507
检修和 FAQ	510
隐私	511

支持的平台和浏览器

LRH SW 支持 3 个平台：

- 台式电脑网络浏览器
- 手机，智能手机设备
- 平板电脑，平板电脑设备

因此您可以为不同的平台创建不同内容和尺寸的页面。例如，您可能想在您的项目中为手机创建一系列更小的页面，而为台式电脑和平板电脑创建完整尺寸的页面。

与计算机兼容

LRH SW 与所有现代 Web 浏览器兼容。以下浏览器已经过测试：LRH SW

- Mozilla Firefox 52+
- Microsoft Edge 42+
- Apple Safari 11+
- Google Chrome 57+



与平板电脑或手机兼容

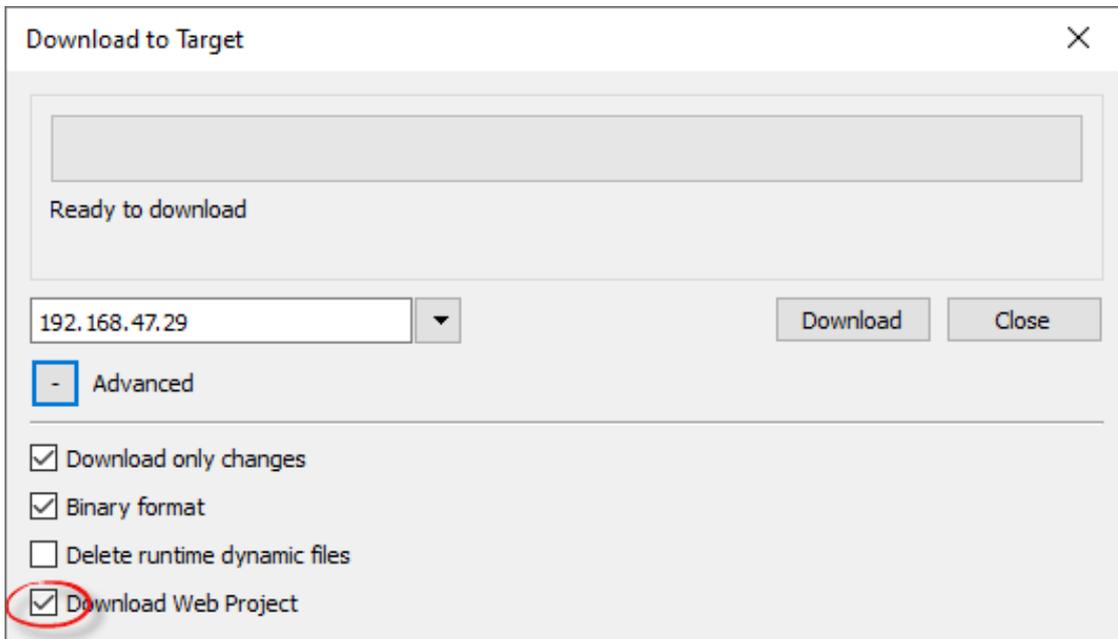
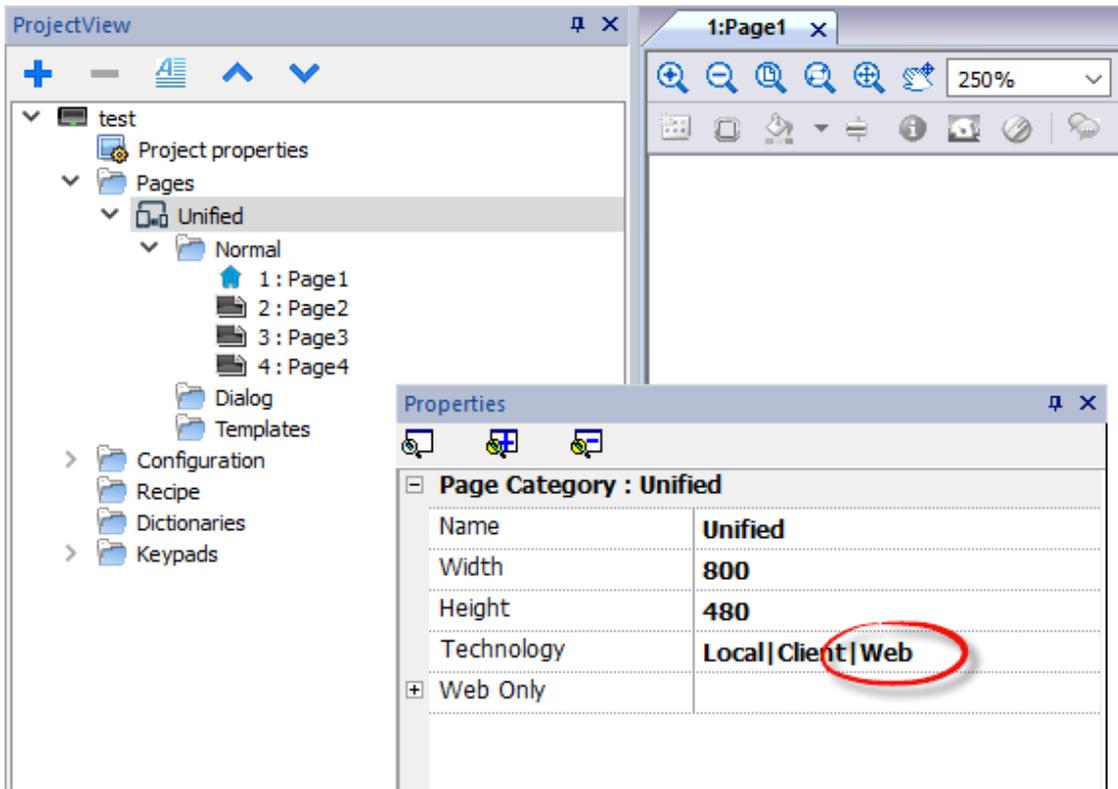
LRH SW 与大多数平板电脑和手机设备兼容。以下平板电脑已经过测试：LRH SW

- iOS 10+ - Mobile Safari
- Android 7+ - Android版 Chrome 55+



Web 页面

为了使 Web 客户端能够访问页面，必须在页面类别的技术参数中包含“Web”引用，并确保在将项目下载至 HMI 设备时选择了**下载 Web 项目**选项(通常默认已选)。



如果应用程序需要将不同的页面发送到不同的 Web 客户端(例如, 智能手机而非 PC 浏览器等), 请参阅 "区分页面" 在本页 66 篇章。

网页属性

任何控件和功能都可以在 LRH SW 中使用; 但是目前并不是所有功能在 LRH SW 中都可用。如果项目包含不可用的功能, LRH SW 仍然会正确运行, 但该功能在远程客户端设备上将不可用。请参阅 "网络支持的功能" 在

本页 507, 了解 LRH SW 中支持的功能和存在的限制。

您可以使用项目验证程序工具来检查您的项目是否包含使用 Web 技术当前不支持的属性进行配置的控制 (请参阅 "项目验证程序" 在本页 64)

除了标准页面属性外, 还有一个其他属性可配置页面, 让其适应浏览器的视区。

属性	说明
缩放至屏幕大小	<p>页面如何适应浏览器的视图端口</p> <ul style="list-style-type: none"> • 无 • 缩放至屏幕 只需修改缩放水平, 以使页面适应浏览器的视图端口 • 响应式设计 考虑网格布局内定义的限制, 智能修改缩放水平, 以使页面适应浏览器的视图端口

使用 url 请求重定向到特定页面

您可以通过使用以下语法输入 URL 来访问特定网页:

```
http://address/index.html?loadPage=pageName
```

测试 Web 项目

您可以直接从浏览器打开独立计算机网页的在线模拟器测试您的 LRH SW 项目。

使用在线模拟器测试

LRH SW 在线模拟器中包含一个 Web 服务器。您可以启动模拟器并从 Web 浏览器访问您的 LRH SW 项目。页面将由模拟器提供。

1. 创建您的项目(请参阅"Web 页面" 在本页 502)。
2. 在运行文件上, 选择**启动模拟器**: 项目将在独立的窗口中运行。
3. 打开 Web 浏览器(请参阅"支持的平台和浏览器" 在本页 502, 获取与 LRH SW 兼容的浏览器列表)。
4. 输入以下网址: `http://localhost:81`: 这样使 Web 浏览器从本地计算机读取网页, 并使用 LRH SW 中在线模拟器默认使用的端口 81。
5. 在浏览器测试您的项目。



重要: 如果您对 LRH SW 中的项目页面做出任何更改, 必须停止并重启模拟器。



注: 如果您使用的设备不是运行模拟器的主机(例如智能手机), 则需要输入用户名和密码。

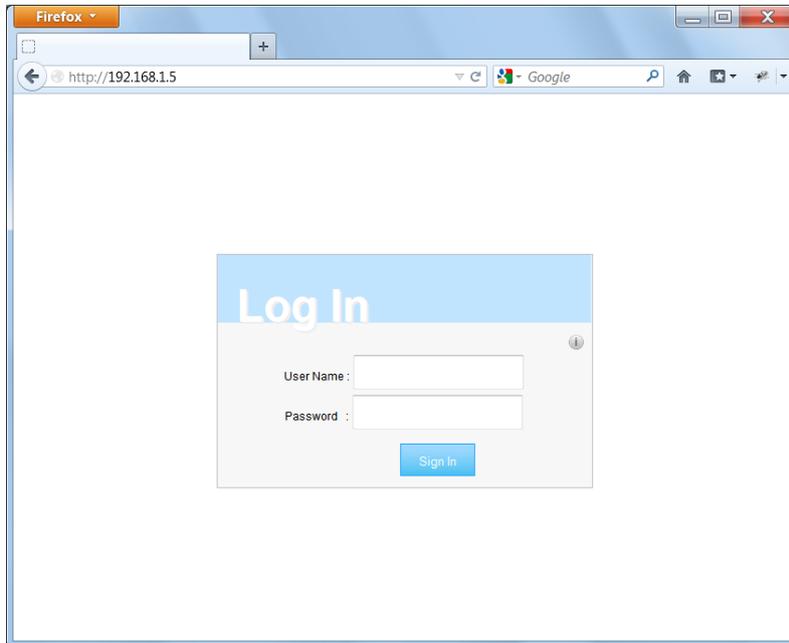
下载网页项目

测试 LRH SW 页面后, 您可以将项目下载到所需的 HMI 设备。

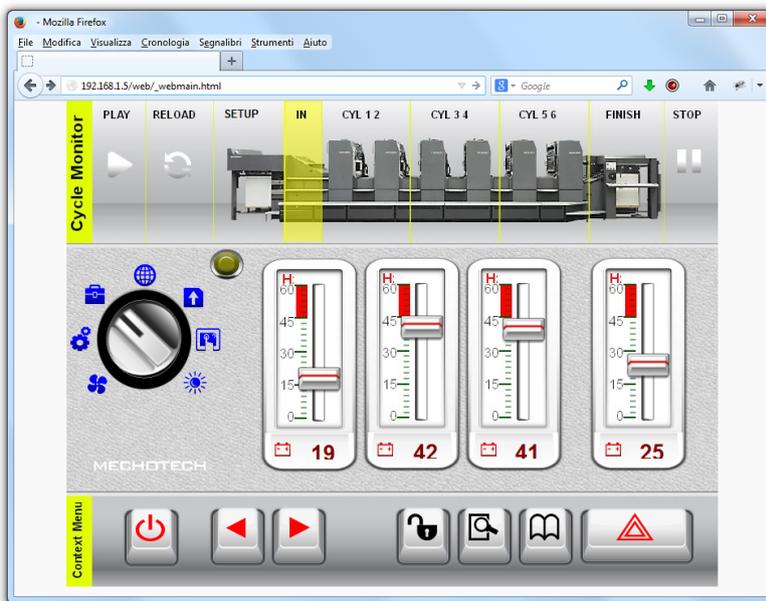
LRH SW 项目和 LRH SW 项目一起下载，请参阅“下载到 HMI 设备”在本页 92 了解详细信息。
下载过程完成后，HMI 项目自动在 HMI 设备上启动，LRH SW 项目准备就绪可供使用。

在浏览器上运行 LRH SW

1. 在一个网络浏览器上，输入您的 HMI 设备的 IP 地址：登录页面显示。



2. 输入用户名和密码，然后单击登录：显示主页。



请参阅“用户管理和密码”在本页 317，了解如何创建权限管理的详细信息。

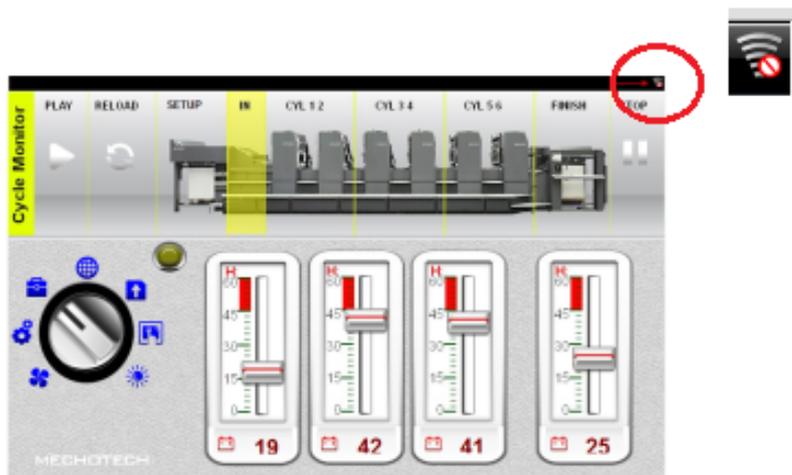
您使用浏览器与设备交互的方式和使用触摸屏与设备交互的方式相同：点击按钮更改页面，查看指示器和计量器，拖拽滑块更改值等等。LRH SW 项目将在您远程与 HMI 设备交互时用 Web 服务器管理所有的通信。

网络连接问题

这里描述的是远程连接 HMI 设备时，您可能最常遇到的问题。

服务器切断

由于 LRH SW 是以远程方式访问 HMI 设备，服务器可能与浏览器断开(例如服务器停止或者网线被拔掉)。如果这种事情发生，“断开”图标将出现在 LRH SW 顶端的工具栏中，如本例所示。



一旦服务器重新连接，红色的圆圈条图标将消失，说明与设备正常通信。

 此“连接状态”系统变量可用于了解连接的状态。查看“[远程客户变量](#)”在[本页 133](#)了解更多详细信息。

 注:如果服务器断开连接时在 LRH SW 页面上做更改，这些变更在客户端可见，但是在通信恢复前不能传输到服务器。

不活动超时

如果浏览器未活动时间达几分钟，LRH SW 将要求您重新登录。例如，如果检测到 10 分钟没有活动，登录屏幕将重新显示，您需要输入您的登录凭据，以继续操作。超时功能保证无法进行未经授权的访问。Web 不活动超时时间可以在[项目属性表](#)中修改。

用户会话终止

用户会话可以被服务器或被用户终止。

在特定的条件下，服务器可能向客户端(浏览器)发出请求，执行登录过程。在这种情况下，用户被重新引导到登录页面，然后返回刚才工作的页面。比如，如果用户清除了浏览器快速缓存或浏览器 cookie，这种情况会发生。

 注:如果用户正在对话框中操作时被要求重新登录，他会被重定向至打开该对话框的页面。

非活动 LRH SW 项目

在您的浏览器中显示的 LRH SW 页面可能来自设备中不再活动的项目。在这种情况下，会显示一个确认窗口，您可以返回到活跃的项目中。



注：假定当前活动项目中有 LRH SW 页面。

如果您选择留在非活动项目，您在浏览器中执行的所有动作可能无法合理执行，因为 LRH SW 无法执行任何与服务器绑定的通信。

网络支持的功能

LRH SW 不支持某些功能或控件的属性。如果使用了不受支持的控件，则只能在 HMI 面板上的页面上获得控件，而在网页上不受支持的控件将不可见。请注意，您可以运行项目验证程序来检查页面是否包含不受支持的控件(请参阅 ["项目验证程序" 在本页 64](#))

不受支持的控件列表

- 模拟时钟
- 模拟视频
- BACnet
- 按钮 `gstArea`
- 耗量表
- 控制列表
- 日期时间组合(可以使用其他控件设置日期或时间)
- IP 控件(可以使用浏览器通过系统设置来配置IP)
- IPCamera(受 Chrome 和 Firefox 支持。请参阅 ["Web 浏览器" 在本页 410](#) 了解更多详细信息)
- 多媒体播放器
- 多状态图像多层(可以使用多状态图像控件)
- 旋转菜单小工具
- RSS 接口
- RSS 滚动
- 计划任务
- 文本编辑器
- Web 浏览器
- 超链接

不受支持的动作列表

- 控件
SlideWidget、BeginDataEntry、TriggerIPCamera、MoveIPCamera、RefreshEvent、ContextMenu、ReplaceMedia、OpenComboBox、CloseComboBox、SelectAlarmsOnSrc、ShiftTableDataSrcColumns、ResetTableDataSrcColumns、SetTableSortingColumn
- Web 浏览器
不支持所有动作
- 文本编辑器
不支持所有动作
- 媒体播放器
不支持所有动作
- 邮件
不支持所有动作
- FTP
不支持所有动作
- 键盘
不支持所有动作
- 页面
LaunchApplication、LaunchBrowser、LaunchVNC、LaunchPDFViewer、LaunchUpdater、LaunchHMICloudEnabler、LockScreen、LoadProject、LastVisitedProject
- 打印
不支持所有动作
- 标签
DataTransfer、ActivateGroup、DeactivateGroup、EnableNode、BACnetClearPriority、BACnetClearAllPriorities、BACnetSetPriority、ClearRetentiveMemory、ForceReadTag
- 趋势/图
滚动耗量表页面
- 系统
Restart、ResetProtoErrCount、SafelyRemoveMedia、ControlUserLED、SaveEventArchive、LogMessage、CopyCodesysProject
- 数据库操作
不支持所有动作
- 用户管理
SwitchUser、ResetPassword、AddUser、DeleteUser、EditUsers、DeleteDynamicFiles、ExportUsers、ImportUsers
- 远程客户端
不支持所有动作

不受支持的功能列表

- 上下文菜单
- 触摸时蜂鸣器
- Javascript 调试
- 滚轮操作(浏览器使用滚轮事件管理滚动条)
- 组合框全屏模式(支持标准“上下文”模式)
- 键盘
- 屏幕保护程序
- 外部数据库(SQL4Automation)
- 显示旋转
- 电子签名

系统变量

使用“附加到”，仅支持下面列出的系统变量，而使用协议“系统变量”支持所有系统变量

- 系统时间
- X Screen resolution
- Y Screen resolution
- 用户群组名称
- 客户用户名
- 连接状态
- 本客户端 ID
- 可用的系统存储
- 当前语言 ID
- 当前语言名称
- 当前语言代码

字体文件

除非已经在设备运行浏览器的操作系统中安装了字体，否则不会从 LRH SW 中加载不具有“字体可嵌入性=可安装”属性(待在字体文件属性>详细信息中验证)的字体文件。

报警

- Web 中不支持基于触发器条件的报警颜色
- 无法在 runtime 中编辑报警控件
- 在基于硬件的智能手机/平板电脑(常规嵌入式设备)中，预计 > 500 报警时出现性能问题。
- 报警触发条件中不支持页面动作

其他

- 对话框页面仅支持模式对话框。
- 尚不支持某些特定的控件属性，在这种情况下，将使用默认值。您可以使用项目验证程序来检查所使用的控件是否包含不受支持的属性(请参阅“项目验证程序”在本页 64)。

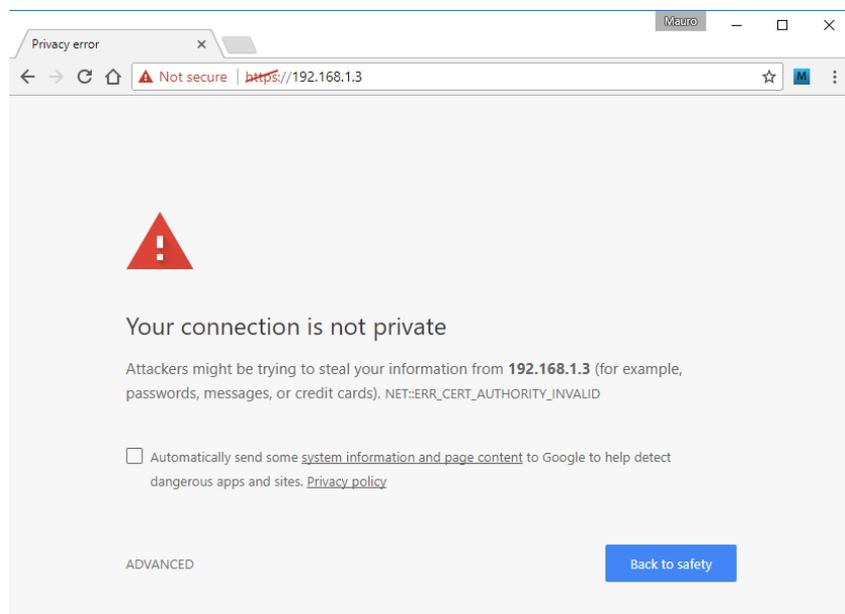
安全套接层 (HTTPS)

Linux 设备支持安全套接层传输协议 (HTTPS)。要使用网页上的此协议访问，请使用以下语法操作：

`https://<device_ip_address>`

请注意，由于已知机构未固定 HMI 设备提供的自签名证书，您将收到一条警告消息。

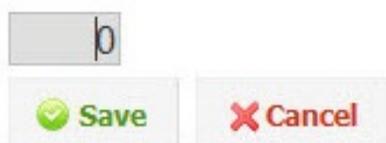
只需单击“高级”按钮以继续。



使用 LRH SW 中的键盘

用户可以点击数字控件，一个文本对话框将显示，可以在其中插入新的值。

插入值后，用户可以按下 **Enter** 或触屏设备上相同作用的键，或单击 **保存** 永久保留新插入的值。在保存过程中只有有意义的数字被接受。其他的将被忽略，不会引起值的更改。



检修和 FAQ

启用 JavaScript

LRH SW 要求 JavaScript 使服务器和用户间能够交互。如果 JavaScript 在您的浏览器被禁用，LRH SW 将无法运行。

按照默认，大部分浏览器在 JavaScript 被启用时开始运行。但是如果已经禁用了 JavaScript，请在访问 LRH SW 页面前重新启用 JavaScript。

浏览器高速缓存

LRH SW 包含一些不会经常变化的资源，例如 CSS 文件、图像文件和 JavaScript 文件。这些资源需要花时间从网络下载，增加 LRH SW 页面在浏览器加载的时间。浏览器高速缓存使这些资源被浏览器保存并使用，不需要每次向浏览器请求。这样可以更快地加载 LRH SW 页面。

高速缓存默认是启用的，要获得最佳的 LRH SW 性能，请确保其未被禁用。



注:LRH SW 页面在浏览器高速缓存被禁用时仍然会正常运行，但是资源加载与正常的高速缓存操作相比会更慢。

使用代理

一些用户可以通过代理访问 LRH SW 项目。代理可以控制浏览器并行连接的数量。

确保允许的最大并行连接(最大连接)不超过 16 且不小于 12。

为什么看不见网页上的更改？

每次新的页面被添加到项目中时，您需要将项目下载到设备。但是，当您连接设备 IP 地址时，Web 浏览器可能显示高速缓存的页面，而不是最近下载的页面。要避免这种状态，您可以：

- 停用网络浏览器的高速缓存
- 强制网页更新
- 旁路浏览器缓存

隐私

我们不使用 cookie 来收集任何用户的私人信息。

Cookie 是存储在用户硬盘上的一条数据，其中包含与用户有关的信息。Cookie 的使用绝不会与我们设备上的任何个人身份信息产生关联。一旦用户关闭浏览器，cookie 就会终止。

42 保护 HMI 设备使用

HMI 设备上的以下操作使用密码保护：

- LRH SW HMI Runtime 管理：安装 LRH SW HMI Runtime，更新 LRH SW HMI Runtime
- 主板管理：更换主要的 BSP 成分，如主要 OS、配置 OS、启动加载器等
- 下载和上传项目文件
- Linux 设备上的可选服务（例如 SSH 协议、VNC 服务器）



警告：为安全起见

- 更改默认密码（请参阅：“[密码保护](#)”在本页 546 适用于 Linux 平台上的 HMI 设备）
- 启用安全管理（请参阅：“[启用/停用安全管理](#)”在本页 318）
- 强制远程登录（请参阅：“[强制远程登录](#)”在本页 327）



警告：未经授权访问设备会造成损害或故障。将设备连接至网络时，保护网络免受未经授权的访问。

用于保护网络的措施包括：

- 防火墙
- 入侵防护系统 (IPS)
- 网络分段
- 虚拟 LAN (VLAN)
- 虚拟专用网 (VPN)
- 物理访问级别方面的安全性（端口安全性）。

与信息技术安全性相关的进一步信息、指示和标准：IEC 62443、ISO/IEC 27001。

更改 HMI 设备密码	514
端口和防火墙	514

更改 HMI 设备密码

如需更改 HMI 设备上的密码，可使用以下方法之一：

- 从 LRH SW HMI Runtime 上下文菜单中：**设置 > 密码** 选项卡。



- 使用更新包中的**设置目标密码**功能：在更新过程完成后 LRH SW HMI Runtime 立即更新密码。
- 使用 HMI 设备的**"系统设置"**在**本页 527**工具



如果未设置目标密码，“旧密码”默认留空。

端口和防火墙

此处是 LRH SW 组件使用的所有端口的列表。

端口	用途	远程访问	板管理	Runtime / 项目管理
80/tcp	HTTP 端口	是	-	是
21/tcp	FTP 命令端口	-	-	是
2100/tcp	板端口	-	是	-
16384-17407/tcp	FTP 数据端口(被动模式)	-	是	是
990/udp	UDP传播 (设备发现)	-	可选的	可选的
991/udp	UDP传播 (设备发现)	-	可选的	可选的
998/udp	UDP传播 (设备发现)	-	可选的	可选的
999/udp	UDP传播 (设备发现)	-	可选的	可选的
5900/tcp	VNC 服务器	仅 VNC	-	-

端口	用途	远程访问	板管理	Runtime / 项目管理
5100/tcp	JS 远程调试器	-	-	可选的
48010/tcp	OPC UA 服务器	-	-	可选的
25/tcp	SMTP 服务器	-	-	可选的
查看您的 MQTT Broker	MQTT	-	-	可选的

远程访问

远程访问需要使用以下工具才能连接到 LRH SW HMI Runtime:

- LRH SW Client
- 网络浏览器

Runtime 和项目管理端口

使用这些端口连接到 LRH SW HMI Runtime, 以便执行更新、安装及项目下载等操作。

板管理端口

使用这些端口连接到 HMI 设备, 以便执行 BSP 更新、封面图像下载等板操作。



注: 当传播服务不可用时, 例如在 VPN 网络中, 键入确切的 IP 地址从 LRH SW 连接到 HMI 设备。

43 改善性能的提示和技巧

LRH SW 为项目设计者提供很大的灵活性。

按照以下建议创建项目，项目在启动、页面更换和动画等方面性能更高。

静态优化	518
静态优化方面的常见问题	521
页面高速缓存	521
图像数据库	521
预缓存	522
预缓存常见问题	522

静态优化

静态优化是使用 LRH SW 的一种方法，用于改善运行性能。

项目中使用的许多图像和图片可能降低性能，静态优化将一些图像融合成一个单一的背景图像，因此减少渲染和加载次数。使用该方法，仅需加载和渲染一张点阵图影像，而不是许多单一的点阵图影像和 / 或矢量图像。

当您在 LRH SW 中创建一个项目时，该页面可能包含文本、图像、背景图像、背景色等控件，它们可以分为：

- **静态的**：运行时值或属性不发生改变。
- **动态的**：运行时值或属性发生改变。

 **注**：基于安全设置，控件的静态部分不能融合到背景中。当控件在安全设置中被配置为“隐藏”时情况会这样。

 **重要**：当您要**在 JavaScript 中更改控件属性时，将控件的“静态优化”设为“动态”，否则属性更改将被忽略。**

下载或验证一个项目时，LRH SW 识别静态组件并将其呈现为 .png 文件的背景图像。这些背景图像作为项目的一部分保存在 /opt 文件夹下。

背景图像可以按如下创建：

- **整页背景图像**，包含融合到页面背景的所有控件
- **编组背景图像**，包含一组静态控件，融合在一起形成一组背景。例如，计量器组通常由一个背景、一个刻度、一个标签和一个指针组成，其中背景刻度和标签都可以融合为一个单一的背景图像。

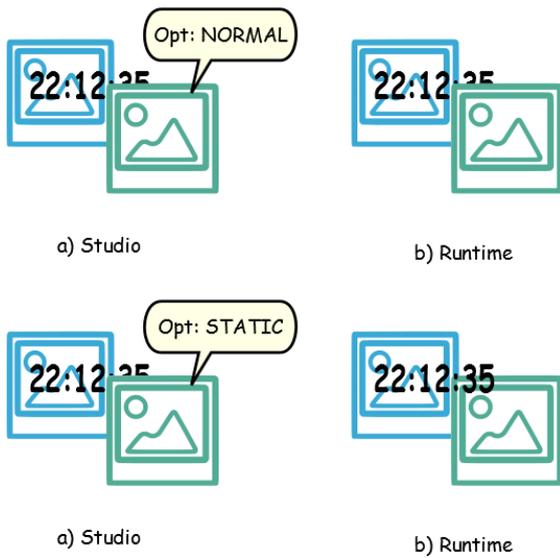
静态优化页面属性支持启用或禁用整个页面的静态优化。如果它设为 **false**，则完全禁用优化。

按照以下方法设置每个控件的**静态优化**属性可以实现更好的控制：

- **正常**：LRH SW 自动检测控件是否与背景融合。如果控件不是动态的，并且不会重叠，即该控件没有重叠在一个动态控件之上，那么这一点可用。
- **静态**：强制图像与背景融合。当静态控件重叠在一个动态透明控件之上时，这一点可用。

 **注**：在这种情况下，自动优化会失败，因为这不会对运行时渲染的不可见区域做假设。

- **动态**：控件完全未被优化。当该静态控件需要由 JavaScript 更改时，使用该选项。



获得最佳性能的建议

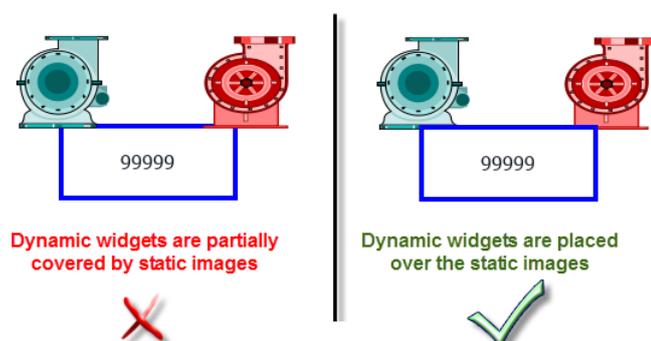
1. 首先:避免将静态控件放在动态控件上。重叠的区域根据控件的边界矩形计算,即通过编辑处理限定的矩形。
2. 如果您的页面几乎只包含动态对象,请勿使用静态优化。静态优化会为每一页保存几乎相同的完整大小的图像,会用去很多存储空间,如果使用其他方法(如页面高速缓存),这些存储空间会得到更有效的利用,改善项目性能。
3. 边界矩形可包括透明区域,让透明区域最小化(例如将图像分成多个图像),因为这些区域即使被优化也会浪费资源。
4. 优化图像大小。图像将以包含图像的图像控件的大小进行渲染。要获得最佳性能,控件需要与图像大小相同。
5. 避免对图像控件使用**改变大小以适应**,因为这样会在运行时强制改变动态图像大小,并在编辑过程中“隐藏”实际图像大小。
6. 使用**缩放至合适大小**使控件与内容实际大小一致。
7. 如果不能避免重叠,要确保将静态图像放在后面,即在动态控件之后。
8. 根据您所连接的HMI设备选择图像文件格式。

9. 避免在一个单一的页面使用太多控件。通常控件被置于可视范围之外，或者它们的透明度由一个标签控制，即使不可见，控件会被加载。一个页面有太多的控件会严重减慢页面的更新时间。
10. 将有许多控件的页面分成多个含有少量控件的页面。
11. 为了在页面中弹出新的图像元素，推荐将对话页面与受控定位的透明部件关联。
12. 选中 **opt** 文件夹查看静态优化是否按预期进行，控件 **z** 顺序可能需要调整。
13. 数字域通常在数据更新事件上运行 **JavaScript** 代码，即使控件不需要在页面上可见。在这种情况下，对于页面可视区域之外的控件，不要使其不可见，而是更改其字体颜色或可见度属性。在后一种情况下，您可能最终会有很多剩余的程序。
14. 如果需要触摸区域对用户输入有反应，请使用热键。
15. 如果您重复使用控件库中的一个控件，或者创建自己的控件，要设置正确的优化属性。例如，按钮控件是动态控件，如果您仅因为控件的外形使用一个按钮控件，该控件不会被优化，因为按钮控件是动态的。如果仅需要外形，您需要使用“向上”的图像。
16. 许多页面有动态控件且使用通用模板时：
 1. 将模板静态优化设为 **true**,
 2. 将页面静态优化设为 **false**, 因为背景已由模板提供。

在这种场景下，背景图像可以被许多不同的页面重复使用，因此可以节省存储空间。

17. 不要仅因图像目的使用动态控件，例如按钮，当不需要按钮功能时，使用图像控件获得同样的图表效果。

这里是一个正确使用和错误使用静态优化的例子。



支持的图像格式

LRH SW 支持多种位图格式，如 BMP、PNG、JPEG、TIFF 和矢量格式 SVG。以下是优点和缺点列表：

图像格式	优点	缺点
位图	<ul style="list-style-type: none"> • 快速渲染 • 标准化 	<ul style="list-style-type: none"> • 文件比较大 • 固定分辨率
矢量 (SVG)	<ul style="list-style-type: none"> • 文件比较小 • 改变大小不会影响质量 • 能够处理动态属性 	<ul style="list-style-type: none"> • 含许多图形项目和图层的复杂 SVG 图像可能渲染很慢。 • 创建一个优化的 SVG 比较复杂。 • 仅 Tiny 1.2 (http://www.w3.org/TR/SVGTiny12/) 受支持。



注：擦除软件是免费工具，可用来清除文件中的外部代码 (<http://www.codedread.com/scour/>)。

模版的静态优化

模版页面可以有大量的静态内容。但是，静态优化不能应用于模版页面，因为模版使用的位置是基于页面设计的。

如果一个大的背景图像要在使用相同模版的每页中重复，这将增加设备的存储负担，模版页面为每一页创建静态图像。

静态优化方面的常见问题

问题：在有一些相同控件的页面的 **opt** 文件夹中，我看到每一个控件都有 **PNG**。如果它们的确是相同的，为什么软件要将其复制，而不是仅有几个 **PNG**？

回答：软件不知道图像是否确实相同，因为每个控件有不同的设置/属性，在运行时根据实际更改。

问题：为什么静态图像存储在 **opt** 独立文件夹中而不是直接将它们存储在项目文件夹中？

回答：这样避免名称冲突，可以跳过优化图像上传

问题：为什么静态图像存储为 **PNG** 文件而不是普通的 **JPEG** 文件？

回答：**PNG** 格式对图像使用无损压缩且支持透明度。与 **PNG** 文件相比，**JPEG** 文件渲染会更模糊，在 **LRH SW** 中(不使用优化)和 **LRH SW HMI Runtime** 中会产生不同的结果。

问题：当软件中未完成优化时，会发生什么？

回答：每一个控件在运行时都会被渲染。尤其是 **SVG** 图像，在嵌入的平台可能需要很多时间来渲染。

页面高速缓存

被访问过的所有页面被保留在一个 **RAM** 高速缓存中，最多可到允许的最大高速缓存大小。这使得访问更加快速，因为高速缓存的页面重新加载需要重绘内容，不需要重新加载所有页面资源。

图像数据库

图像数据库是用于跟踪图像文件用途和通过快速存储最常用图像(例如，按钮图像，计数器指针，滑块等等)来减少图像加载消耗的一种手段。这样在不同的地方使用的同一图像只加载一次。

图像数据库功能将在启动时预先加载最常使用的图像，直到达到存储限制。这样将进一步增加单独图像加载的次数。

文件 **imagecachelist.xml** 在 **project/opt** 文件夹中创建，包含以下相关信息：

- 填充颜色(在 **SVG** 图像的情况下)
- **SVG** 图像的大小
- 图像在项目中被使用的次数
- 同一图像的不同尺寸

使用图像数据库功能的注意事项

1. 使用按钮、计量器和其他控件时，尽可能采用统一尺寸。
2. 在同一种类的控件中使用相同的颜色主题。

预缓存

页面的预缓存属性可用来通知 LRH SW HMI Runtime 在启动时预加载 RAM 中的一些页面以快速访问。预缓存对含有许多动态控件的复杂页面比较有用。

当该功能在一个页面被启用，对该页面的访问更快，但是，这将延长启动时间，因为系统在所有要预缓存的页面没有保存到 RAM 中之后才能准备就绪。

预缓存提示

1. 仅对一些具有动态控件或用户频繁使用的页面启用预缓存功能。
2. 不要对项目中的所有页面启用预缓存功能，因为这样会内存不足，无任何意义。
3. 对启用预缓存功能的页面停用静态优化以减少内存的使用。

预缓存常见问题

预缓存的页面限制

根据页面的大小和复杂性，预缓存要求的空间可以是 1,5Mb 到 3Mb。

加载项目时，LRH SW HMI Runtime 按如下方式运行：

1. 页面图像被预加载，直到 76MB 的存储空间仍然可用 (imageDBLowMem)
2. 预加载预缓存设为 **true** 的页面，直到仍有 64 MB 的存储空间可用 (pageCacheLowMemMax)。这些页面的图像被加载到 RAM 上(图像数据库)。

当项目准备就绪：

1. 被访问的新页面和所有相关图像被保存在缓存 (RAM) 中，直到 40MB 的存储空间仍然可用 (pageCacheLowMemMin)
2. 页面切换且 RAM 中的空间处于临界状态 (<40MB)，LRH SW HMI Runtime 开始删除页面和相关图像以清空缓存 (RAM)，直到有 64MB 的存储空间可用。LRH SW HMI Runtime 按照以下顺序删除存储在缓存中的数据：
 1. 最后访问的页面及较大且未使用的图像 (>320x240)，
 2. 如果需要更多的存储空间，预缓存中的页面和所有在图片数据库中加载的图片可能被移除。

44 功能规格说明与兼容性

这里是支持功能和相关限制概览。这里说明的局限代表安全限制，不保证非正确的操作和超系统性能的使用。

功能限制表	524
HMI 设备地址	525
兼容性	525
在不同的 HMI 设备间转换项目	526

功能限制表

功能	最大限制
页数	1.000
基础控件的数量	2.000 x 页
标签数量	10.000
对话框页的数量	50
可以同时打开的对话框页面数	5
配方数量	32
配方的参数集数量	1.000
每个配方的元素数量	1.000
用户小组的数量	50
用户的数量	500
并行的远程客户的数量	4
计划任务的数量	30
报警数量	请参阅 "HMI 设备地址" 对页
数据传输数	1000
模版页的数量	50
每个按钮状态程序可控的动作数量	32
趋势缓存的数量	30
每个趋势缓存的标签数量	200
趋势缓存保留的内存	请参阅 "HMI 设备地址" 对页
每个趋势控件曲线的数量	5
每个散点图控件的曲线数量	10
趋势表最大可打印的行数	LRH SW HMI Runtime上, 最大行数 10.000
一个信息域信息的数量	1024
语言的数量	24
每个缓存的事件数量	2.048
事件缓存的数量	4
每页的 JavaScript 文件大小	16 KB

功能	最大限制
磁盘上项目的大小	请参阅 "HMI 设备地址" 向下
索引事例的数量	100
索引别名 (Alias) 的数量	100
索引标签集的数量	30
物理协议数量	4
报表数量	32
报表页的数量	32
变量控件中的变量最大数量	255
用户文件夹大小 (UpdatePackage.zip)	5 MB
并行的FTP会话数	4
FTP 附加文件夹	5

HMI 设备地址

请参阅"功能限制表"上一页了解标准功能。

Panel	Device OS	Touch	Media Player	Media Player Portrait Mode	PDF	Max Project Size	Dialogs	Alarms	Protocols	JavaScript	Reports	Trend Buffers	Max Tags inside a Trend	Curves per Trend Widget	Events Buffers Size	User Folder Size
LRHA04	Linux		na	Yes	Yes	60 MB	50	500	4	64 KB	32	25 Mb	200	5	2 K	100 MB
LRHA07	Linux		MPEG4	Yes	Yes	60 MB	50	2.000	4	64 KB	32	25 Mb	200	5	10 K	100 MB
LRHA10	Linux		MPEG4	Yes	Yes	60 MB	50	2.000	4	64 KB	32	25 Mb	200	5	10 K	100 MB

兼容性

系统已采用以下兼容性策略：

- LRH SW 版本必须始终与设备上的 LRH SW HMI Runtime 一致，
- 用户负责在 LRH SW 更新时更新 HMI 设备上的 LRH SW HMI Runtime 组件，
- LRH SW HMI Runtime 更新可以使用“运行”\“管理目标”对话框中的“更新目标”命令直接从 LRH SW 完成，
- 在不低于 V1.00 (00) 版本的 LRH SW 上创建的项目可以通过任何更新的版本打开和处理。
- 使用旧版本的 LRH SW 创建的项目，用后来的版本打开并在兼容的 LRH SW HMI Runtime 中配置，可以保持其性能和功能。
- LRH SW HMI Runtime 的新版本与使用旧版本的 LRH SW 创建并配置的项目之间的兼容性不能保证。



重要：请勿使用比创建项目更旧版本的 LRH SW 来编辑项目。这可能导致项目损坏和 LRH SW HMI Runtime 不稳定。

在不同的 HMI 设备间转换项目

支持不同 HMI 设备型号间的项目转换，但是，如果项目使用目标设备中不支持的功能，则可能需要一些手动操作。

指示

转换项目之前，了解是否有不受支持的功能(请参阅["HMI 设备地址" 上一页](#))，并调整您的项目，在转换项目前，移除不受支持的功能。

尤其是：

- 验证限制和功能是否不受新 HMI 设备支持(请参阅["功能限制表" 在本页 524](#)了解详细信息)。
- 除去不支持的控件、动作、系统变量、协议、项目属性。
- 如果项目使用外部存储，请验证同一存储路径是否仍可用。
- 调整 OS 特定的外部应用或路径。
- 如有必要，根据新 HMI 设备的类型限制减少项目大小(请参阅["限制"](#)了解详细信息)。
- HMI 设备基于不同的硬件平台，有不同的 CPU 速度、RAM 存储大小、缓存大小，因此请确保检查项目启动时间和每页的加载时间。
- 验证 OS 特定操作的 JavaScript 码。

OS 特征

Linux 区分大小写，而 Windows CE 不区分。因此，Linux HMI 设备上的项目可以用大小写为不同的文件命名，例如“dump1.csv”和“Dump1.csv”，这在 Windows CE HMI 设备上是不可能的。

45 系统设置

“系统设置”是 HMI 设备的内部工具，可用于基本设备设置或用于系统组件更新。



注：您可以从 LRH SW 更新系统组件(请参阅["更新 HMI 设备中的系统组件"](#) 在本页 551 了解详细信息)



小心：使用“系统设置”工具是很危险的操作，如果操作不当，可能对产品造成损坏，需要维修产品。
联系技术支持以求帮助。

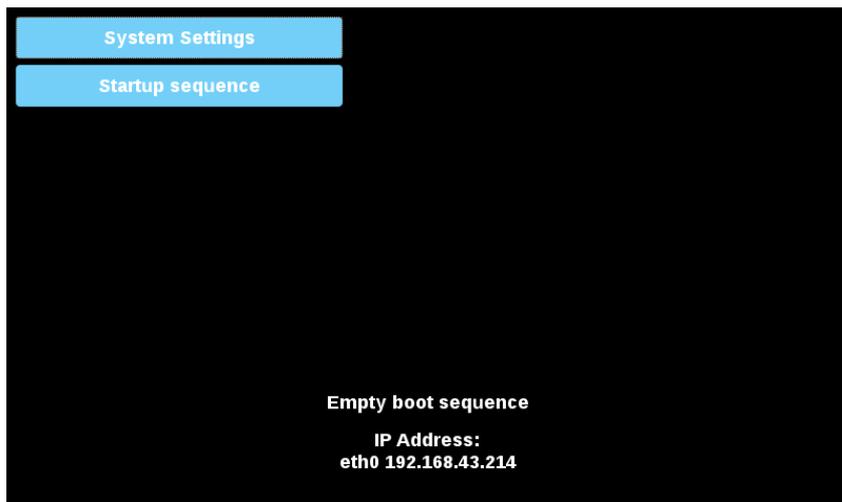
Linux 设备	528
----------------	-----

Linux 设备

Linux 产品提供一个功能强大的集成工具，名为“系统设置”，它能够管理和升级系统组件。可以直接在 HMI 执行操作或使用 Web 浏览器以远程方式执行。

Runtime 安装

HMI 设备出厂时没有 Runtime，首次启动 HMI 时会显示“Runtime 加载器”屏幕。



可以通过以下方式安装 Runtime：

- 自动，通过以太网使用 LRH SW 下载首个项目
- 手动，通过 USB 闪存，通过“更新包”安装。（请参阅“更新包”在本页 94 以创建 Runtime 包）

通过以太网安装 Runtime

要通过以太网安装 Runtime，请按照“下载到 HMI 设备”在本页 92 过程操作。



警告：通过以太网下载安装 Runtime 要求 HMI 有有效的 IP 地址。

IP 地址的分配方式有三种：

- 自动通过 DHCP 服务器分配。默认启用此选项。如果 DHCP 服务器在网络上可用，该服务器将自动分配 IP 地址。
- 自动通过“自动分配 IP”功能分配。如果已启用 DHCP 分配，但网络上没有 DHCP 服务器可用，则 HMI 会给自己分配一个 IP 地址，范围为 169.254.x.x，子网掩码为 255.255.0.0
- 手动通过“系统设置”分配。在“系统设置”菜单中的“网络”部分下，可以手动分配 IP 地址，需禁用 DHCP 服务器分配功能。

通过 USB 闪存安装 Runtime

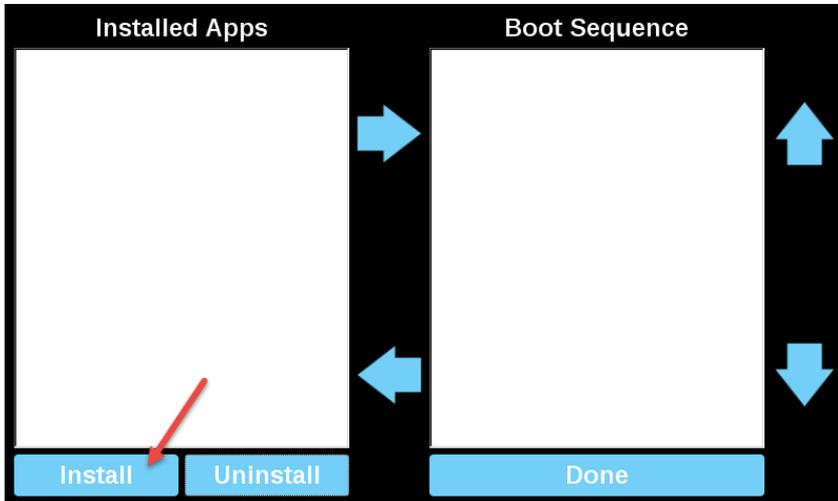
要通过 USB 设备安装 Runtime 更新包或备份包，请按照以下过程操作：

1. 在 LRH SW 中创建更新包并将其复制到空的 USB 闪存条中

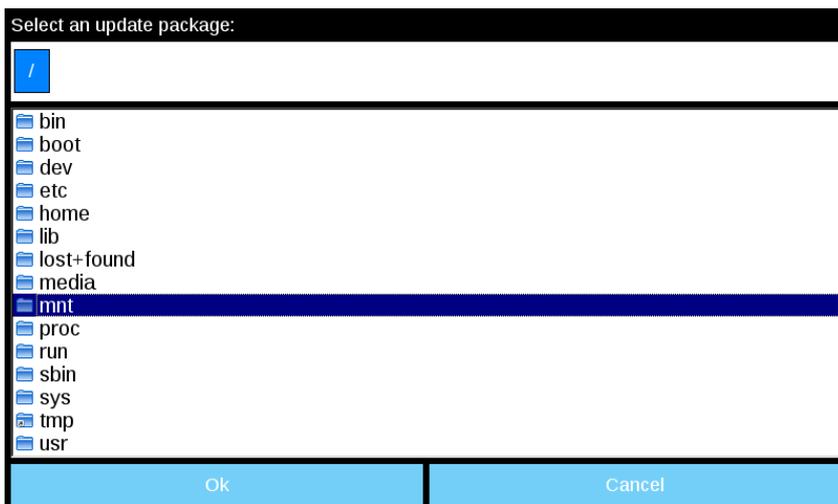


注：支持的文件系统是 FAT16/32 和 Linux Ext2、Ext3 和 Ext4。

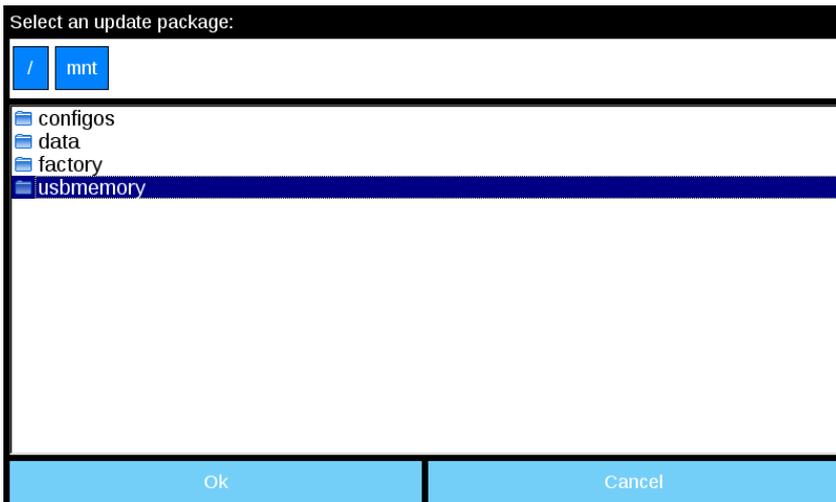
2. 在 HMI 上，选择 [启动序列]，然后选择 [安装]



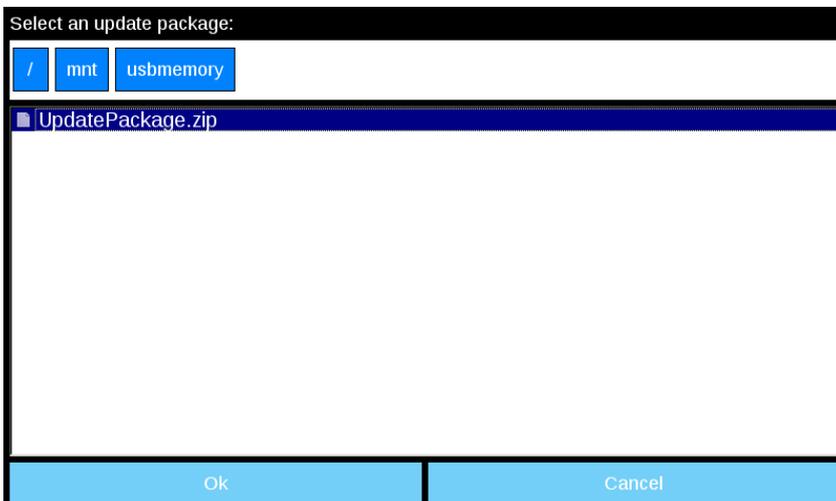
3. 双击“mnt”以访问该文件夹



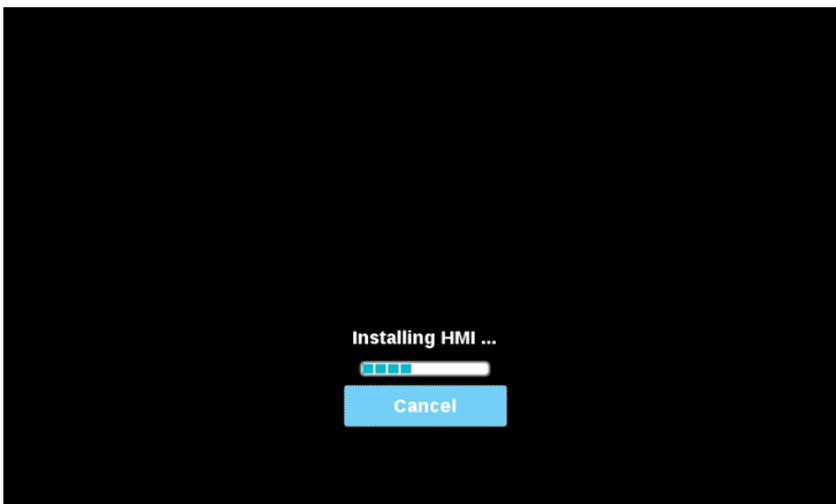
4. 然后双击“usbmemory”



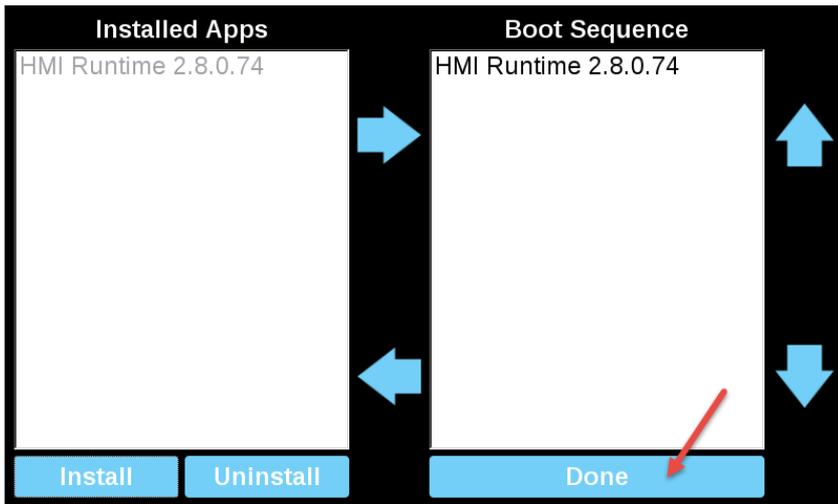
5.选择“UpdatePackage.zip”并按 [确定] 确认



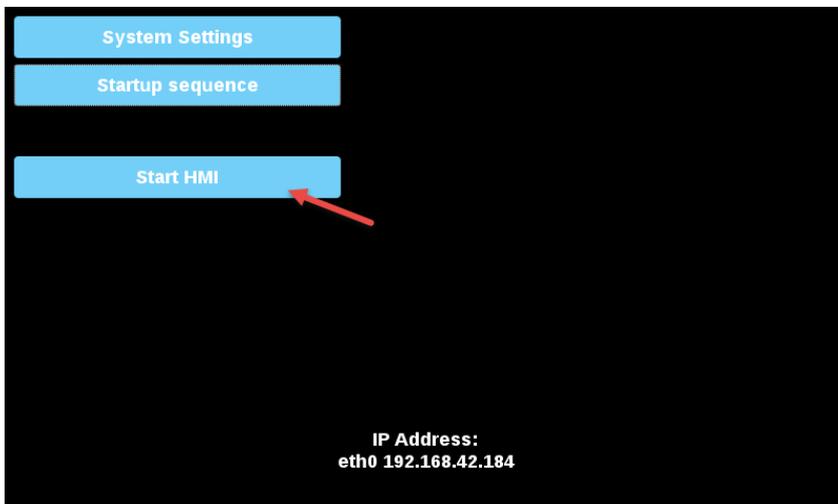
6.Runtime 安装开始



7.结束时，按“完成”按钮



8.然后，按“启动 HMI”按钮



系统设置

系统设置的用户界面基于 HTML 页面，可以在 HMI 设备屏幕上本地访问，也可以使用网络浏览器远程访问。具有完全访问权的管理员用户名是“admin”，默认密码“admin”。普通用户名为“user”，默认密码“user”



警告：出于安全原因，请更改两个用户名的默认密码(可以通过“系统设置 -> 身份验证”命令修改密码)



在未更改默认“admin”密码之前，从 HMI 设备访问系统设置不需要输入密码。

从 Web 浏览器访问系统设置

要使用网络浏览器访问系统设置，用以下格式输入设备的 IP 地址：

`https://IP/machine_config`

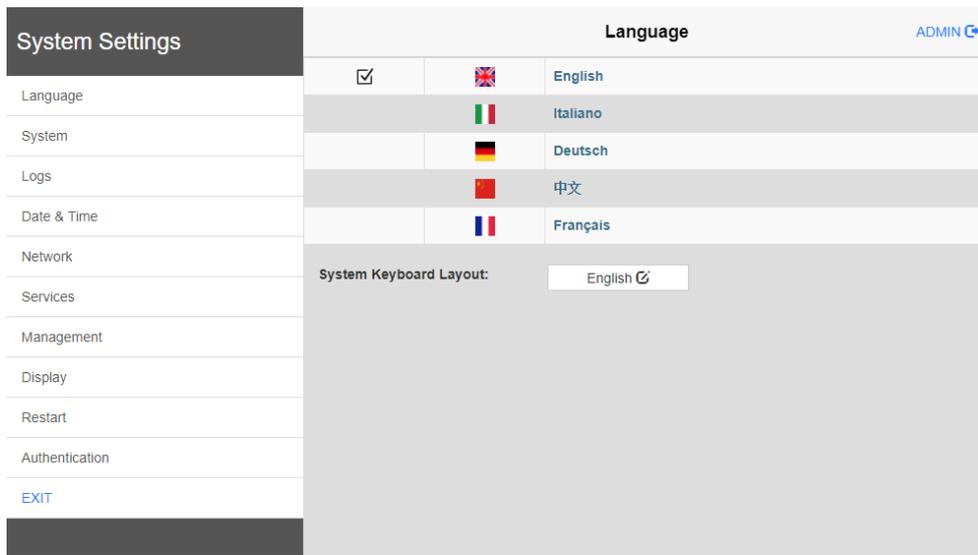


请注意，远程访问使用端口 443 上的已加密 https 协议。建立连接时，HMI 设备会发送用于加密的证书。由于该证书未由证书颁发机构签名，您将收到一条警告消息。请单击高级选项和选择以继续。





通过左边的菜单选择：活跃的项被高亮，相关信息显示在右边。



Linux HMI 设备中 HTTPS 服务器默认安全协议为：

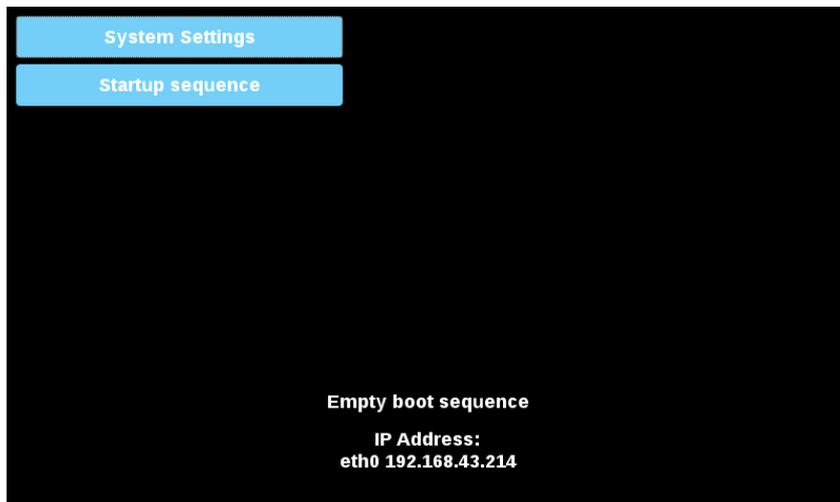
- SSLv3 256 位 ECDHE-RSA-AES256-SHA
- TLSv1 256 位 ECDHE-RSA-AES256-SHA



警告：我们不鼓励在 **SSL3** 或 **TLSv1.0** 连接环境中使用 **CBC** 网络套件，因为可能受一些漏洞影响。

从 HMI 设备访问系统设置

未安装运行时，可以从“Runtime 加载器”屏幕访问系统设置，

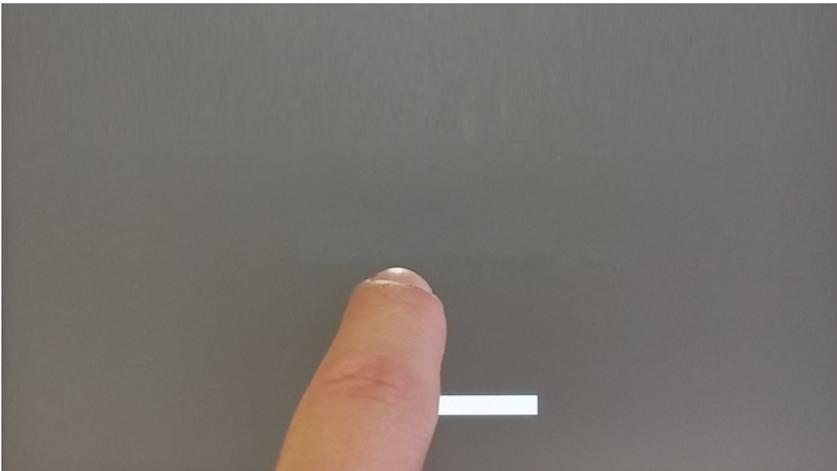


安装 Runtime 后，可以选中“上下文菜单”中的“显示系统设置”选项访问系统设置，



通过 Tap-Tap(连续轻敲)过程进入系统设置

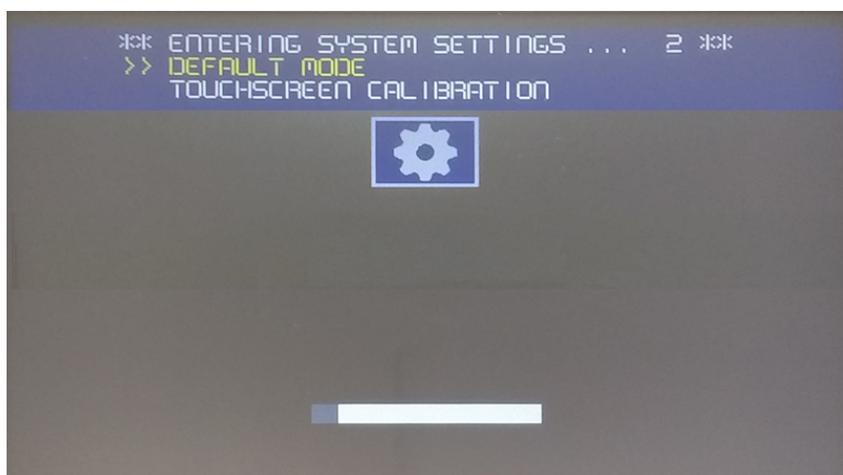
Tap-Tap(连续轻敲)是由多个触摸激活动作组成,在设备通电阶段通过手指轻敲触摸屏即可执行,并在设备通电后立即启动。



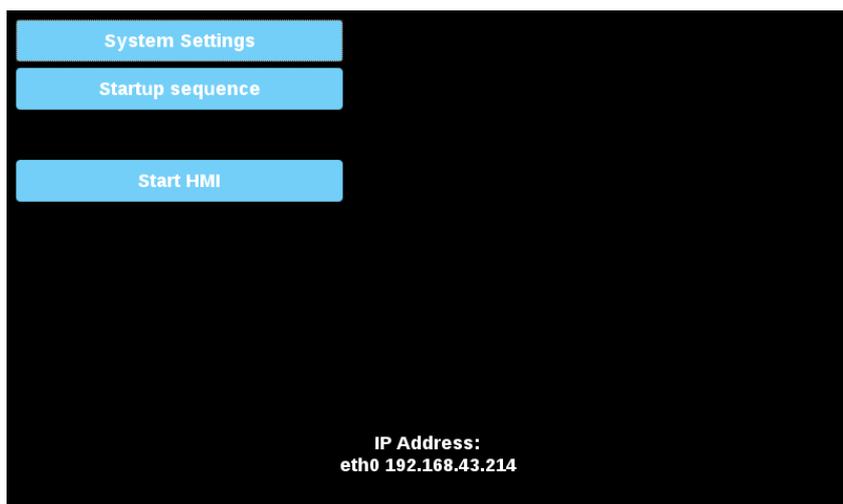
在屏幕顶部显示“tap-tap detected”(检测到连续轻敲)消息时。等待 5 秒钟(不要触摸屏幕)以进入“系统设置”子菜单



等待 5 秒钟(不要触摸屏幕)以进入“默认模式”



从 HMI 的“默认模式”屏幕中选择“系统设置”



系统设置部分

要更改系统设置值，请单击右上角的“编辑”按钮进入编辑模式。



“编辑”按钮仅在包含可修改参数的对话框内可用。

语言

选择系统设置界面上使用的语言

- 系统键盘布局: 选择虚拟键盘的布局

系统

参数	说明
信息	设备信息
状态	设备状态(剩余 RAM、正常运行时间、CPU 负载)
计时器	设备计时器(系统运行时间、背光亮起时间)
插件	硬件插件信息

日志

如果要维护在电源重置后保存的日志文件，请设置持久日志选项。

使用保存按钮导出日志文件的副本。



日志文件管理器周期性地填充 3 个 4Mb 文件

日期和时间

设备日期和时间。

参数	说明
当前时区	时区区域
当前日期 本地时间	仅当禁用“自动更新”时，才能手动设置日期和时间。
自动更新 (NTP)	<p>启用此参数，使日期和时间与远程服务器同步</p> <ul style="list-style-type: none"> • NTP 服务器 指定 Internet NTP 服务器地址 <p> HMI 设备的 NTP 客户端是网络时间协议(NTP)版本 4 的完整实现，但仍保留与 RFC-1305 定义的版本 3 及与分别由 RFC-1059 和 RFC-1119 定义的版本 1 和版本 2 的兼容性。</p> <p>轮询过程按时钟规则算法确定的时间间隔发送 NTP 数据包。该过程旨在提供足够的更新速率，以最大程度地提高准确性，同时最大程度地减少网络开销。该过程旨在 8 秒至 36 个小时期间以可变模式操作。</p>

网络

网络参数。处于编辑模式的可用参数如下：

参数	说明
通用设置	设备主机名 Avahi 主机名(请参阅 "Avahi 守护程序" 向下)
网络接口	可用接口的网络参数 <ul style="list-style-type: none"> • DHCP • IP 地址 • 网络掩码 • 网关
DNS	DNS 服务器 通常由 DHCP 服务器提供，但可在编辑模式下进行修改 搜索域 将与提供的 URL 串联使用的可选域

服务



仅当以 **admin** 身份登录时，服务才可用。

用鼠标单击“启用”按钮可启用/禁用服务。单击服务名称可列出关联参数。

来自外部存储的自动运行脚本

当将 USB 密钥插入设备时，启用/禁用运行“**autoexec.sh**”脚本文件的可能性。如果要防止通过 USB 接口未经授权的访问，请禁用此服务。

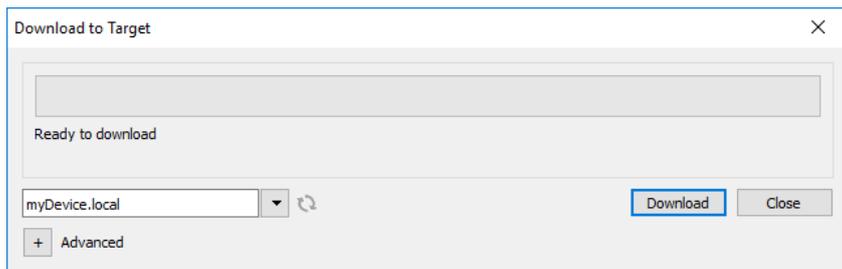


需要 BSP v1.0.212 或更高版本

Avahi 守护程序

Avahi 是一个系统，使程序能够发现和公布在本地网络上运行的服务和主机。当它启用时，即使使用设备的主机名(除 IP 地址之外)也可访问 HMI 设备。

General Settings	
Hostname	myDevice
Avahi Hostname	myDevice.local



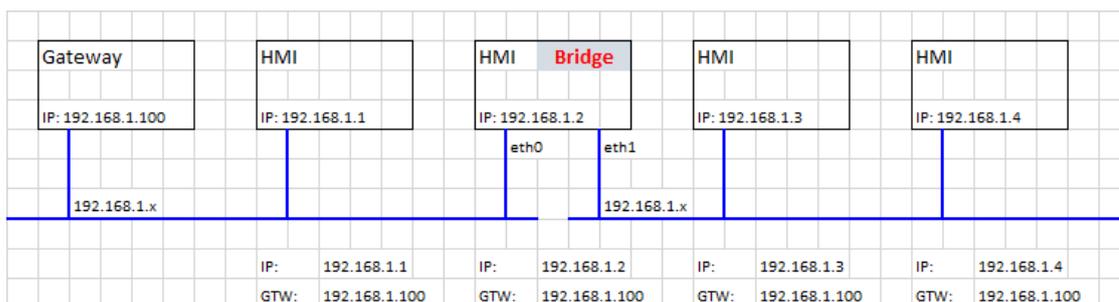
Avahi 守护程序在 UDP 端口 5353 上运行

 在 Linux 和 Apple PC 上，Avahi 服务与操作系统一起免费提供。相反在 Windows PC 上，需要安装 Avahi 服务以便能够通过其 Avahi 主机名访问面板(例如，需要安装 Apple Bonjour 应用程序 - Bonjour 是 Apple inc. 的商标)。

桥接/切换服务

使用桥接服务可以将 WAN (eth0) 网络适配器和其他网络接口连接在一起。使用时，两个以太网接口进行桥接并共享相同的 IP 地址。

桥接服务在两个或多个网络接口之间创建基于 Linux 的层 2 网络桥。如果 WAN 和端点设备均连接到此桥接上，则两个网络将通过物理方式连接且端点将如同直接连接到 WAN 一样可用



DHCP 服务器

在选定界面提供 DHCP 服务器。

参数	说明
已启用	在选定界面启用 DHCP 服务器
起始 IP 停止 IP	从 DHCP 服务器分配的 IP 地址
网关	网关地址
网络掩码	提供的网络掩码
DNS 服务器	DNS 服务器地址
租约时间(秒)	租约时间,默认为 86400s(1 天) 可接受的值是 60s 至 864000s(10 天)

快速启动

在上电时启用快速启动时，HMI 设备将尽快启动 HMI 应用程序。在此模式下，不会显示诊断信息(例如加载条)，但是在加载用户界面(例如，系统设置、VNC、SSH 等数据将在加载 HMI 应用程序之后加载)之前只加载了最少的必要功能。

为获得最佳性能，除启用快速启动模式之外，还建议：

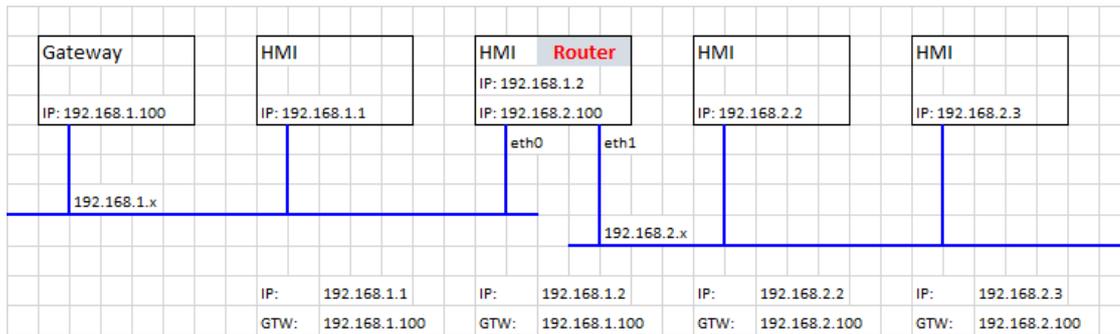
- 禁用任何不必要的服务
- 避免保留已启用的持久日志
- 使用静态 IP 地址，而非 DHCP 服务



需要 BSP v1.0.242 或更高版本

路由器服务

该服务使用 IP 转发和网络地址转换共享 WAN (eth0) 至 LAN(eth1 或 eth2) 的连接：所连接的端点可访问通过网关可访问的相同网络(如可用，包括 Internet)



在启动过程中显示加载栏

在引导阶段启用/禁用显示加载栏。

SNMP 服务器

SNMP 是一个允许管理网络基础设施的网络协议。通常用于监控连接到 LAN 网络的交换机、路由器等网络设备。

当 SNMP 服务启用时，SNMP 管理器可以使用 SNMP 协议从 HMI 设备检索信息。当前，没有可用的专用 MIB。仅标准公共社区 MIB 在只读模式下可用。

The screenshot shows the iReasoning MIB Browser interface. The address bar is set to 192.168.57.98 and the OID is .1.3.6.1.2.1.1.5.0. The MIB Tree on the left shows the hierarchy: iso.org.dod.internet.mgmt.mib-2.system.sysName. The Result Table on the right displays the following data:

Name/OID	Value	Type	IP:Port
sysName.0	myDevice	OctetString	192.168.57.98:161
sysDescr.0	Linux myDevice 3.14.28-rt25-1.0.0_ga-g4f85bca #...	OctetString	192.168.57.98:161
sysUpTime.0	65 hours 42 minutes 25 seconds (23654530)	TimeTicks	192.168.57.98:161
memAvailReal.0	570808	Integer	192.168.57.98:161
memTotalFree.0	570744	Integer	192.168.57.98:161
ssCpuIdle.0	97	Integer	192.168.57.98:161

Below the table, the details for sysName are shown:

Name	sysName
OID	.1.3.6.1.2.1.1.5
MIB	RFC1213-MIB
Syntax	DisplayString (OCTET STRING) (SIZE (0..255))
Access	read-write
Status	mandatory
DefVal	

示例：

系统名称： .1.3.6.1.2.1.1.5.0
 系统说明： .1.3.6.1.2.1.1.1.0
 系统开启时间： .1.3.6.1.2.1.1.3.0
 使用的总 RAM： .1.3.6.1.4.1.2021.4.6.0
 可用的总 RAM： .1.3.6.1.4.1.2021.4.11.0
 空闲 CPU 时间 (%)： .1.3.6.1.4.1.2021.11.11.0

SNMP 服务器在 UDP 端口 161 上运行



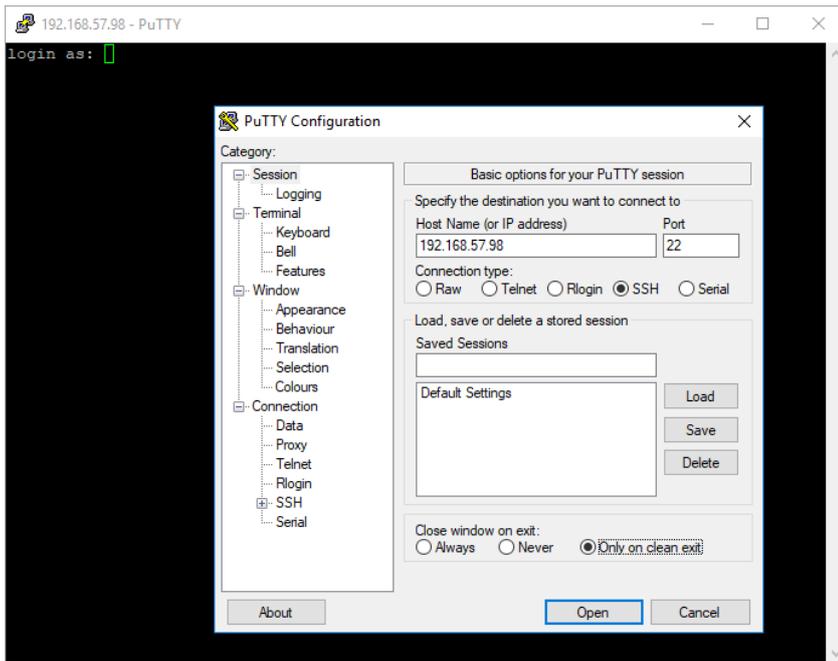
该服务设计用于开发阶段。出于安全原因，请在切换到生产之前停用该服务。

SSH 服务器

SSH 服务仅针对高级用户而设计。它使用安全 shell 协议提供对 HMI 设备的远程登录。您可以在 PC 上运行 SSH 客户端，如 PuTTY，它是一个在 MIT 许可证下分发的开源软件。



admin 用户名的默认密码为“admin”。有关更多信息，请参阅“验证”在本页 542 章节。



SSH 服务器在 TCP 端口 22 上运行



该服务设计用于开发阶段。出于安全原因，请在切换到生产之前停用该服务。

VNC 服务

VNC 服务允许对 HMI 设备的显示屏进行远程访问。VNC 客户端可用于获取 HMI 设备的远程控制。



使用后应禁用 VNC，不建议自动启动。

- 端口: VNC 服务器侦听 TCP 端口 5900(默认)上的连接
- 多个客户端: 允许在同一端口上执行多个会话(如果已停用, 之前记录的客户端将在新传入连接上断开)
- 仅查看: 不允许有效的用户交互(客户端只能观看)
- 加密: 激活连接的 SSL 加密(未广泛支持 - 检查客户端兼容性)
- 身份验证: 在创建会话时是否对用户进行身份验证。可设置自定义 VNC 特定密码或使用系统密码(此选项仅在也启用加密时可用)

插件

本页显示可用于配置连接到 HMI 设备上的可选插件模块的参数。有关更多信息, 请参阅每个插件模块的说明。

管理



仅当以 admin 身份登录时, 管理才可用。

在管理区域内, 可以对 HMI 设备执行"更新系统组件"在本页 543 操作。



小心:在管理区域内操作很危险,如果操作不当,可能对产品造成损坏,需要维修产品。联系技术支持以求帮助。

使用“数据”部分中的“清除”命令删除设备上的 HMI Runtime(恢复出厂设置)

显示

参数	说明
亮度	显示屏的亮度级别
背光超时	背光不活动超时时间
方向	显示方向

重启

HMI 设备重启命令

验证

进入编辑模式更改身份验证密码。

具有完全访问权的管理员用户名是“admin”，默认密码“admin”。普通用户名为“user”，默认密码“user”



警告:出于安全原因,请更改两个用户名的默认密码

X.509 证书

HMI 设备使用自签名证书通过 HTTPS 协议对 Internet 通信加密。您可以使用公司数据个性化证书并要求证书颁发机构进行固定。

用于个性化和固定证书的程序如下：

1. 进入编辑模式并填写必要的参数,然后按“生成”按钮可使用数据生成自签名证书。
2. 导出“证书已签名请求”
3. “证书已签名请求”已发送至证书颁发机构进行固定(通常这是付费服务)
4. 将已签名证书导入到 HMI 设备上

证书参数

参数	说明
设备名称	设备名称
组织	组织的法定名称
单位	处理证书的组织部门
状态	组织所在的州/地区
位置	组织所在的城市

参数	说明
国家	组织所在国家的两个字母 ISO 代码
有效(天数)	证书有效性
密钥长度	加密算法中使用的密钥位数

托管证书采用 base64 编码形式



需要 BSP v1.0.239 或更高版本

退出

退出“系统设置”工具。

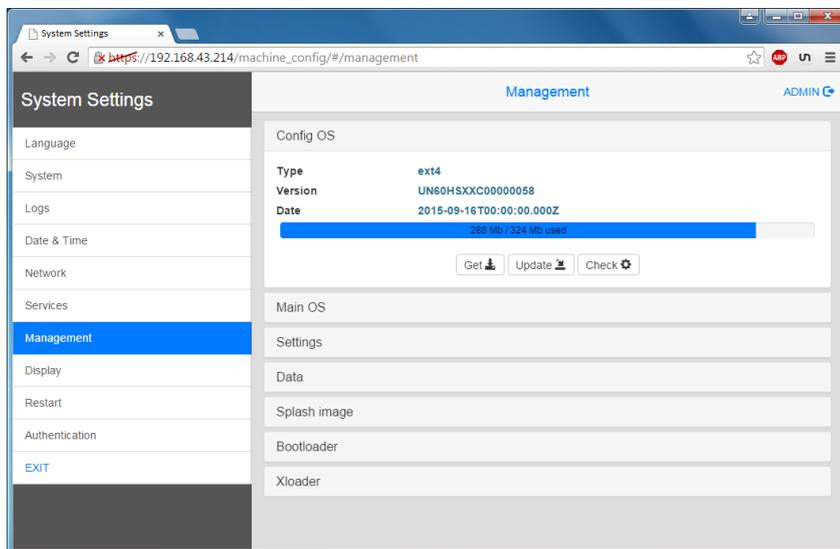
更新系统组件



小心：在管理区域内操作很危险，如果操作不当，可能对产品造成损坏，需要维修产品。联系技术支持以求帮助(最新BSP文件将由技术支持人员提供)。

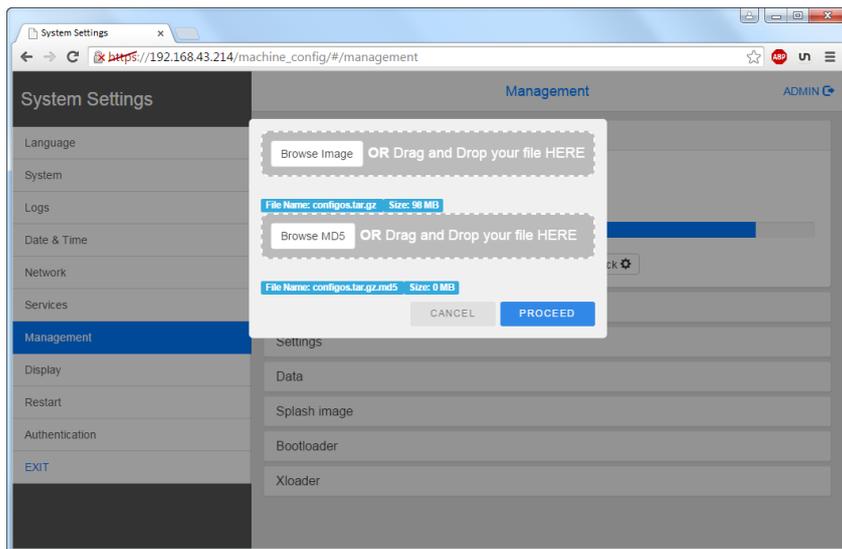
可以使用 U 盘以本地方式或通过 Web 浏览器以远程方式更新 Linux 设备的系统组件。

要更新系统组件，请在 HMI 上执行连续轻敲过程，在“配置操作系统”模式下进入“系统设置”，或打开 Web 浏览器，输入网址 https://<HMI IP 地址>/机器_配置，并选择“管理”部分。



展开要更新的组件并选择 [更新]

在打开的对话框中，单击 [浏览图像]，然后选择“xxx-mainos-xxx.tar.gz”文件。然后单击 [浏览 MD5] 并选择“xxx-mainos-xxx.tar.gz.md5”文件。



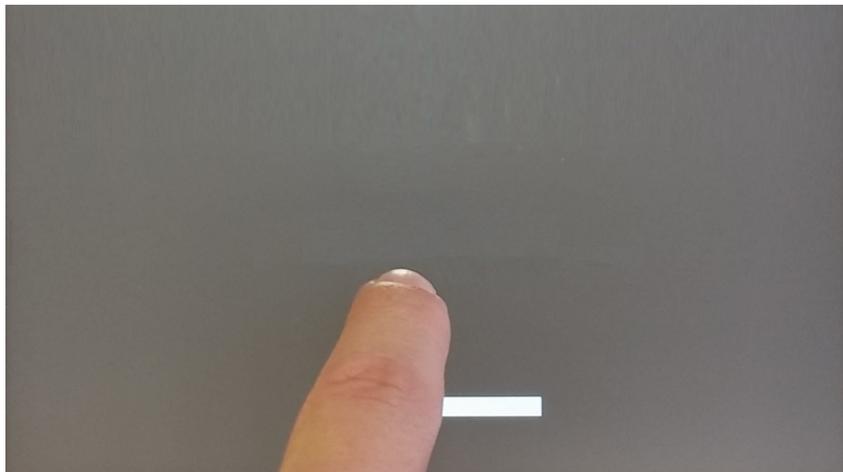
重要:在升级系统组件时切勿关闭设备。

组件更新结束后，重启 HMI 并让其正常启动。

通过 Tap-Tap(连续轻敲)过程在“配置操作系统”模式下进入系统设置

可以通过 Tap-Tap(连续轻敲)在“配置操作系统”模式下进入系统设置，即使 HMI 出现软件故障，此模式仍可访问。

Tap-Tap(连续轻敲)是由多个触摸激活动作组成，在设备通电阶段通过手指轻敲触摸屏即可执行，并在设备通电后立即启动。



在屏幕顶部显示“tap-tap detected”(检测到连续轻敲)消息时，手指按住触摸屏，选择“Restart: Config OS”(重启：配置操作系统)



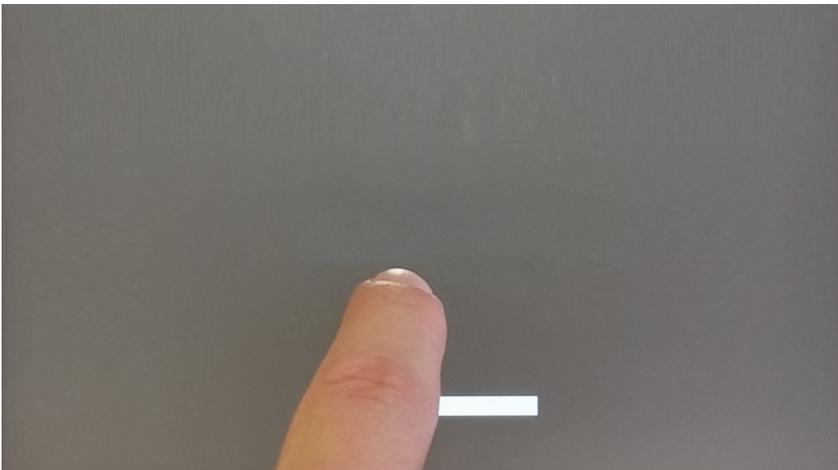
HMI 将在“配置操作系统”模式下重启进入系统设置：



触摸屏校准

系统设置校准支持校准触摸屏设备，可通过连续轻敲过程访问此校准过程。

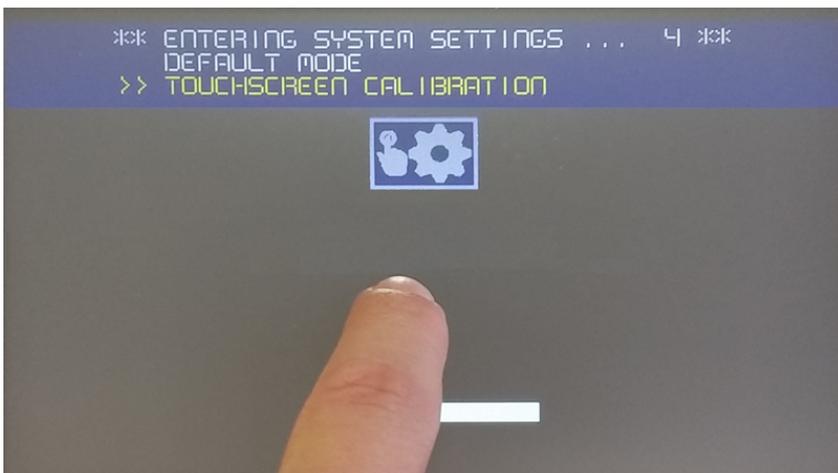
Tap-Tap(连续轻敲)是由多个触摸激活动作组成，在设备通电阶段通过手指轻敲触摸屏即可执行，并在设备通电后立即启动。



在屏幕顶部显示“tap-tap detected”(检测到连续轻敲)消息时,等待 5 秒钟(不要触摸屏幕)以进入“系统设置”子菜单



点按触摸屏,“Touchscreen calibration”(触摸屏校准)语音提示将突出显示为黄色,按住屏幕几秒钟,直到触摸屏校准过程启动

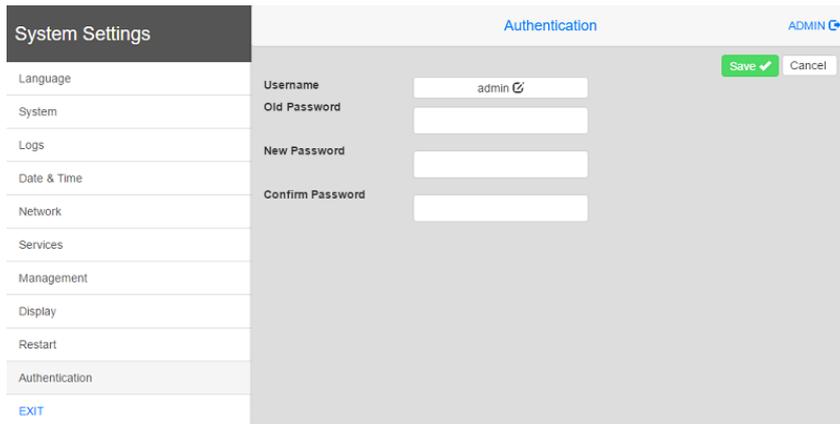


按照屏幕上的说明完成校准过程,系统将提示您触摸特定的点以校准触摸屏设备。

密码保护

HMI 设备的内部密码。

从“系统设置”在本页 531 中的“身份验证”选项卡下,激活编辑模式并选择用户名以更改关联密码。



甚至可以从LRH SW HMI Runtime的上下文菜单中(请参阅“背景菜单选项”在本页8了解详细信息)或升级套餐中(请参阅“更新包”在本页94了解详细信息)修改 admin 用户的密码



在未更改默认“admin”密码之前，从 HMI 设备访问系统设置不需要输入密码。

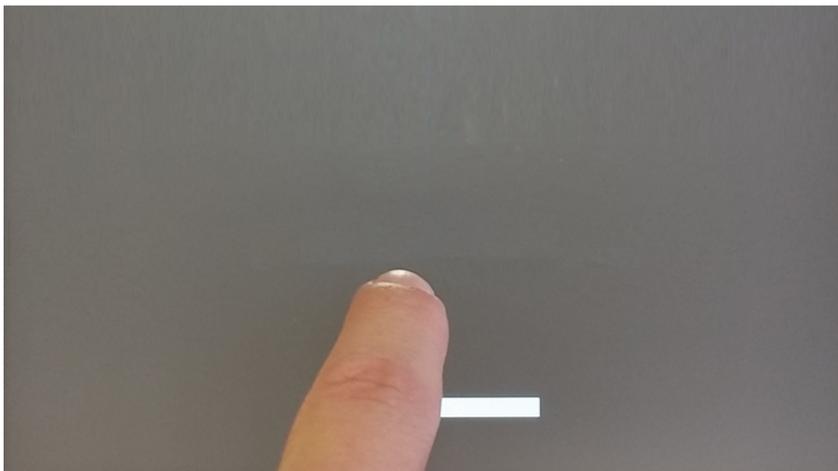
有两种用户名：

- 具有完全访问权的管理员用户名是“admin”，默认密码“admin”。
- 普通用户名为“user”，默认密码“user”

恢复出厂设置

处于“默认”模式下的系统设置支持卸载 HMI Runtime 或更改启动序列，这种模式可通过 Tap-Tap(连续轻敲)获得，即使 HMI 出现软件故障，此模式仍可访问。

Tap-Tap(连续轻敲)是由多个触摸激活动作组成，在设备通电阶段通过手指轻敲触摸屏即可执行，并在设备通电后立即启动。



在屏幕顶部显示“tap-tap detected”(检测到连续轻敲)消息时。等待 5 秒钟(不要触摸屏幕)以进入“系统设置”子菜单



等待 5 秒钟(不要触摸屏幕)以进入“默认模式”

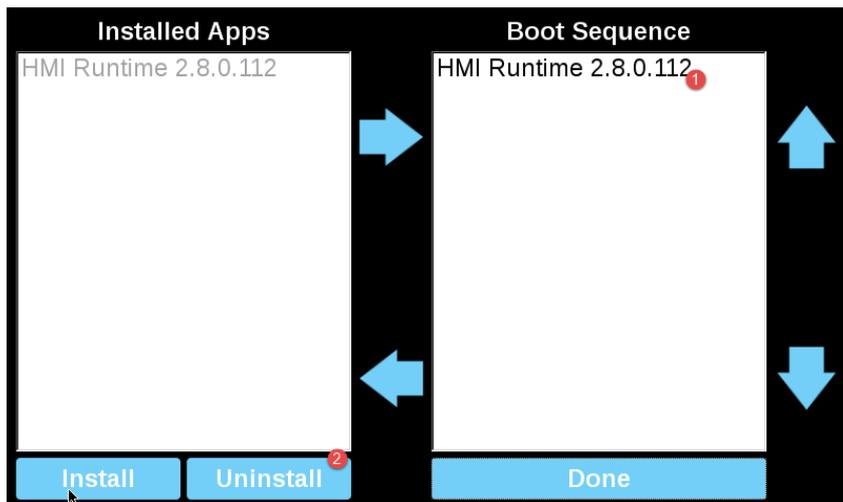


要从 HMI 上卸载 Runtime, 请在“默认模式”屏幕中选择 [启动序列]:

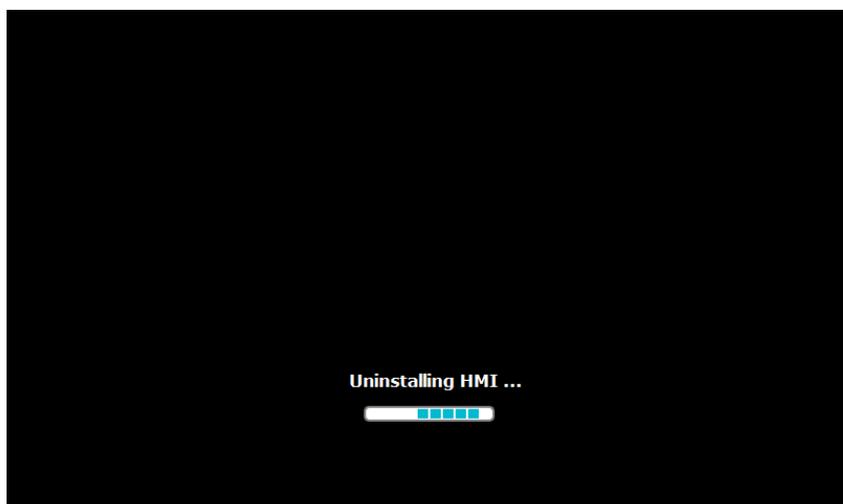


从安装的应用程序视图中:

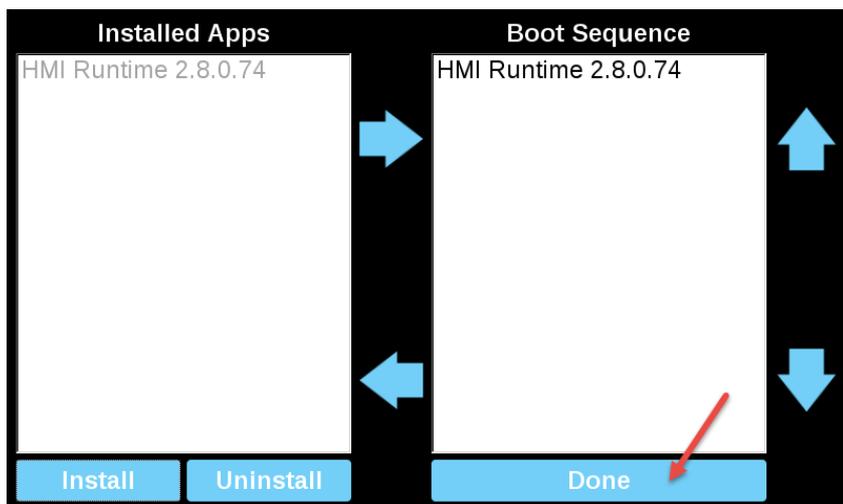
1. 选择您想移除的 Runtime
2. 卸载所选 Runtime



系统将执行 Runtime 卸载过程：



结束时，按“完成”按钮



配置模式

甚至在无法启动设备的情况下，可以通过一个特殊过程即在称为配置模式的特殊模式下启动设备来恢复设备。在此模式下，您可以打开设备管理对话框，从中删除用户数据，恢复系统设置或更新设备固件。

要以配置模式启动设备，请选择以下过程之一

- 打开设备电源，然后在屏幕上显示初始屏幕时立即关闭电源(如果看不到初始屏幕，请在听到哔哔声时关闭设备电源)。重复此过程三遍，再次打开设备电源，等待配置模式出现。
- 创建一个名为“\$0030D8\$.bin”的特殊文件，并将其放入空的 SD 卡中。将 SD 卡插入设备并打开设备电源。设备将以配置模式启动。

46 更新 HMI 设备中的系统组件

大部分系统软件组件可以轻易地升级，确保为已有和正在运行的系统提供更新和修复时具有高度的灵活性。可以更新的新软件模块

- 直接在 HMI 设备上使用 USB 闪存驱动器(请参阅"系统设置"在本页 527 了解详细信息)
- 从 LRH SW 应用程序中(请参阅"从应用程序更新系统组件"下一页了解详细信息)

每一个 HMI 设备上都有一条标签，上面有一个包含所有出厂设置(硬件、软件和固件组件)的产品编码。请查看该标签了解您的 HMI 设备的信息。HMI 设备更新工具也提供在该设备上运行的组件的详情。



小心: 确保使用正确的升级文件，因为使用与设备不符的升级文件将导致严重的系统故障。始终检查您的设备产品编码。



注: 升级文件在接到请求时作为技术支持活动的一部分分配。



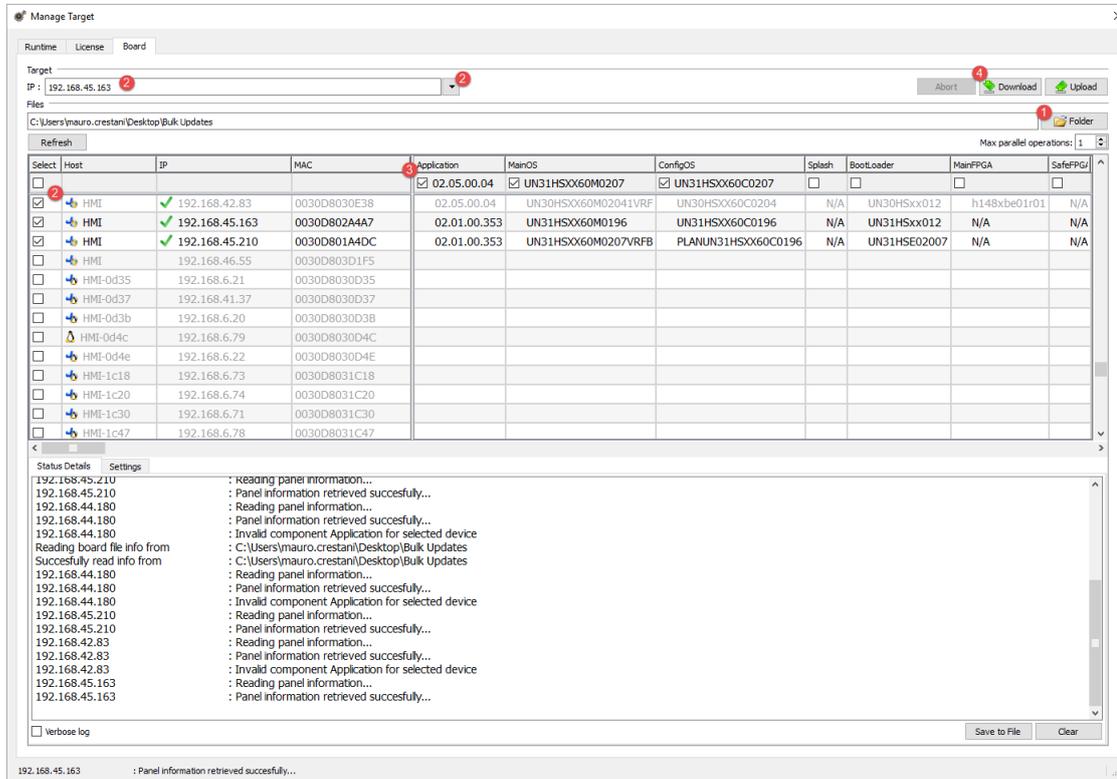
服务呼叫: 降级操作是复杂的，如果未正确执行，可能对您的设备造成严重的损害。这些操作由技术支持来完成。

从应用程序更新系统组件	552
设置	552

从应用程序更新系统组件

您可使用以太网连接接口将系统组件下载到单一HMI设备或相同类型的批量HMI设备。

路径: 运行 > 管理目标 > 主板



1. 选择包含将下载至 HMI 设备的文件或通过 HMI 设备上传的文件的文件夹
2. 选择一个或多个 HMI 设备。
3. 选择您将下载(或上传)到(或从)设备的组件
4. 启动“下载至 HMI”或“通过 HMI 上传”操作

注:

- 此工具旨在更新同一类型的多个 HMI 设备。请避免将不同设备类型文件放在同一文件夹中
- 如未列出所需的目标 IP, 请直接将其输入框中。发现服务是广播服务。当通过 VPN 或从外部网络完成一项远程连接时, 它将无法正常工作, 您需要手动输入地址。
- 仅可将所选组件下载至兼容设备
- 您可根据您的网络和硬件功能增加并行更新的设备数量
- 您需要重新启动 HMI 设备以完成更新。

设置

从设置选项卡中, 可以指定用于与 HMI 设备通信的端口和密码参数。如未在 HMI 设备侧设置密码, 则将密码留空。



警告: 批量模式仅可用于有相同连接参数的 HMI 设备

下载初始屏幕图片

您可以在充电阶段更改设备显示的默认初始屏幕图片。

用于作为初始屏幕的图像必须符合以下要求：

文件名	splash.bmp
格式化	Bitmap 位图, RGB 565 格式
大小	< 500 KB
Bitmap 位图宽度	偶数 (例如 430x239)

下载初始屏幕图像：

1. 重命名新的初始图片 **splash.bmp** 并将其复制到源文件夹中。
2. 选择 HMI 设备
3. 单击 **下载**。



要确保最好的视觉效果，闪动屏幕图片必须有黑色的背景。

