SIEMENS

SIMATIC Visualization Architect

系统手册

在线帮助打印输出

05/2021
在线帮助打印输出

法律资讯

警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。

⚠ 危险

表示如果不采取相应的小心措施，**将会**导致死亡或者严重的人身伤害。

⚠ 警告

表示如果不采取相应的小心措施，**可能**导致死亡或者严重的人身伤害。

⚠ 小心

表示如果不采取相应的小心措施，可能导致轻微的人身伤害。

注意

表示如果不采取相应的小心措施，可能导致财产损失。

当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。 由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用 Siemens 产品

请注意下列说明：

⚠ 警告

Siemens 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 Siemens 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

商标

所有带有标记符号 ® 的都是 Siemens AG 的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
德国

文件订购号: 在线帮助打印输出
Ⓟ 06/2021 本公司保留更改的权利

目录

安全性信息

1

安全性信息

Siemens 为其产品及解决方案提供了工业安全功能，以支持工厂、系统、机器和网络的安全运行。

为了防止工厂、系统、机器和网络受到网络攻击，需要实施并持续维护先进且全面的工业安全保护机制。Siemens 的产品和解决方案构成此类概念的其中一个要素。

客户负责防止其工厂、系统、机器和网络受到未经授权的访问。只有在必要时并采取适当安全措施（例如，使用防火墙和/或网络分段）的情况下，才能将这些系统、机器和组件连接到企业网络或 Internet。

此外，应考虑遵循 Siemens 有关相应安全措施的指南。有关工业安全的更多信息，请访问

http://www.siemens.com/industrialsecurity (http://support.automation.siemens.com)

Siemens 不断对产品和解决方案进行开发和完善以提高安全性。Siemens 强烈建议您及时更新产品并始终使用最新产品版本。如果使用的产品版本不再受支持，或者未能应用最新的更新程序，客户遭受网络攻击的风险会增加。

要及时了解有关产品更新的信息，请订阅 Siemens 工业安全 RSS 源，网址为

http://www.siemens.com/industrialsecurity。 (http://www.industry.siemens.com/topics/global/en/industrial-security/Pages/Default.aspx)

网络驱动器

确保您的网络基础架构和计算机中的网络驱动器已受到保护，未经授权无法访问。

通过以太网通信

在基于以太网的通信中，最终用户自己负责数据网络的安全。不能保证设备在所有情况下都能正常运行；例如，遭受蓄意攻击就会导致设备过载。

数据保护

2

西门子遵守数据保护准则，特别是数据最少化原则（从设计着手保护隐私）。这意味着对于 SIMATIC 产品，产品不处理/保存任何个人信息。

基本知识

3

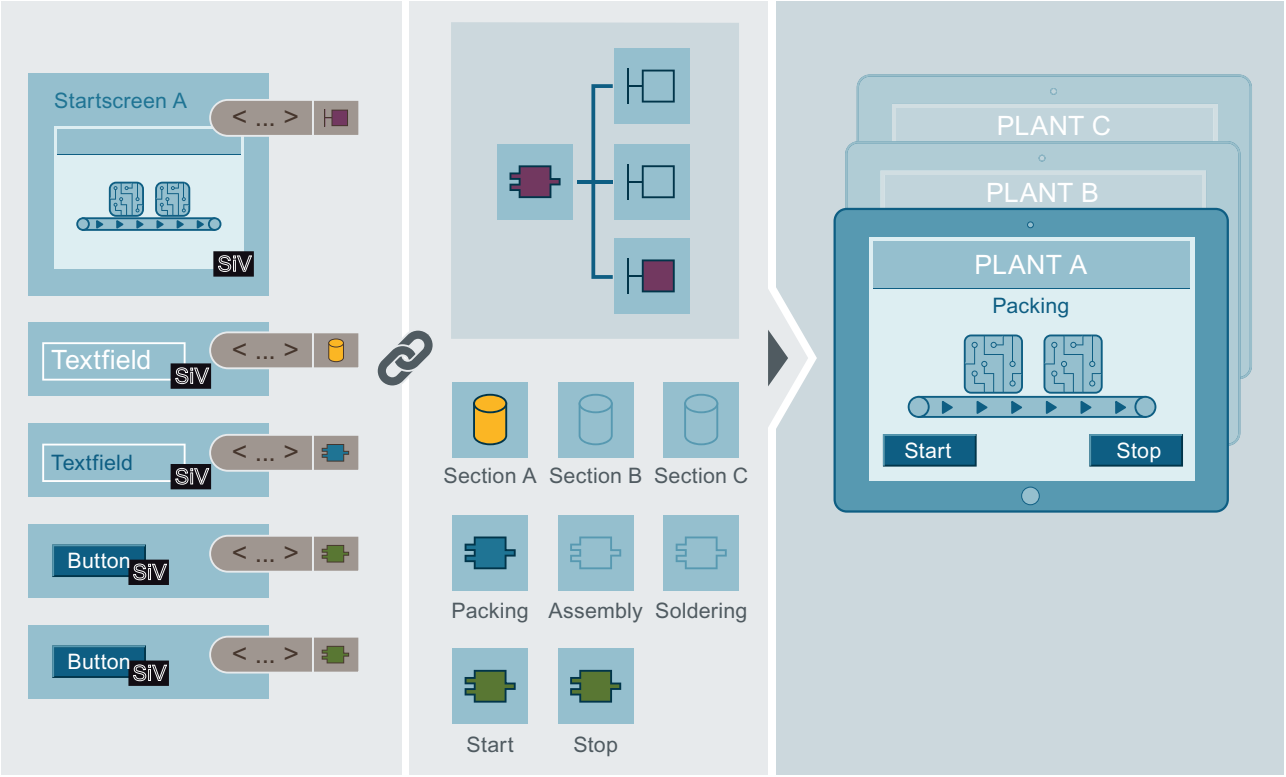
3.1 简介

什么是 SiVArc?

SiVArc (SIMATIC WinCC Visualization Architect) 是 TIA Portal 中的选件包。

借助 SiVArc，可通过程序块和生成模板为多个 HMI 设备和 PLC 生成可视化。

可使用生成规则指定哪些 HMI 对象针对哪个块和设备生成。



功能范围

可以使用 SiVArc 通过控制器数据生成下列 HMI 对象：

- 画面、面板、显示元素和操作元素选项
- 外部变量
- HMI 文本列表

在不引用控制程序的情况下，可使用 SiVArc 通过 WinCC 项目库为 WinCC 项目生成选定的对象，或将其用作实例。

使用项目库中的生成模板，或者使用全局库来执行生成操作。

SiVArc 可同时为多个 HMI 设备、多个 PLC 和设备代理生成可视化。在使用 SiVArc 生成虚拟化时，可在另一个实例中继续使用 TIA Portal。利用 SiVArc 和 TIA Portal 选项"TIA Portal Multiuser"，还允许不同的用户访问一个 SiVArc 项目。

在 SiVArc 编辑器中，可对系统或脚本功能进行复制、粘贴、撤消、重做。使用全局搜索，可搜索系统或脚本功能的名称或参数值。搜索结果中包含有"转至"链接，可跳转到所搜索的条目处。

SiVArc 设置可作为全局设置的组成部分。

有关 SiVArc 自动化任务的更多信息，请参见 Siemens YouTube 频道 (https://www.youtube.com/watch?v=txLWqFOmAlM)。

有关 SiVArc 的更多信息，请参见 Siemens 工业在线支持 (https://support.industry.siemens.com/cs/ww/en/view/109751096)。

参见

用户程序中支持的对象 (页 110)

通过 SiVArc 组态 HMI 解决方案 (页 20)

SiVArc 表达式概述 (页 113)

画面图例 (页 287)

生成模板的创建 (页 95)

3.2 应用

概述

在高度标准化的自动化解决方案中使用 SiVArc。

SiVArc 可在以下任务的工程组态期间支持组态工程师：

- 包括过程连接的可视化自动生成
- 用户界面的统一布局

- 操作元素的一致命名
- 组态数据的结构存储

SiVArc 还可在运行阶段提供支持：

- 调试
  SiVArc 可在调试期间提供帮助，即使没有 SiVArc 的专业知识，调试工程师也可使用生成矩阵在短时间内对项目执行更改。
- 调整
  要对整个项目应用更改，仅需通过 SiVArc 调整中心模板。
- 设备维护
  例如，生成特定的各个设备意味着可轻松交换 HMI 设备。

SiVArc 也适用于促进项目标准化和连续优化项目。

优势

与传统的可视化组态相比，SiVArc 的基本附加值包含以下 SiVArc 原理：

- 生成的可视化保留对 SiVArc 项目的引用。SiVArc 的调整和优化功能可确保高性能和结构清晰的数据库。
- 可视化可直接链接到用户程序。用户程序的更改几乎无须对 WinCC 项目进行调整。
- 通过 STEP 7 和 WinCC 可集中控制显示屏上的布局、设计和一致名称。

对组态工程师的要求

使用 SiVArc 需要下列先验知识：

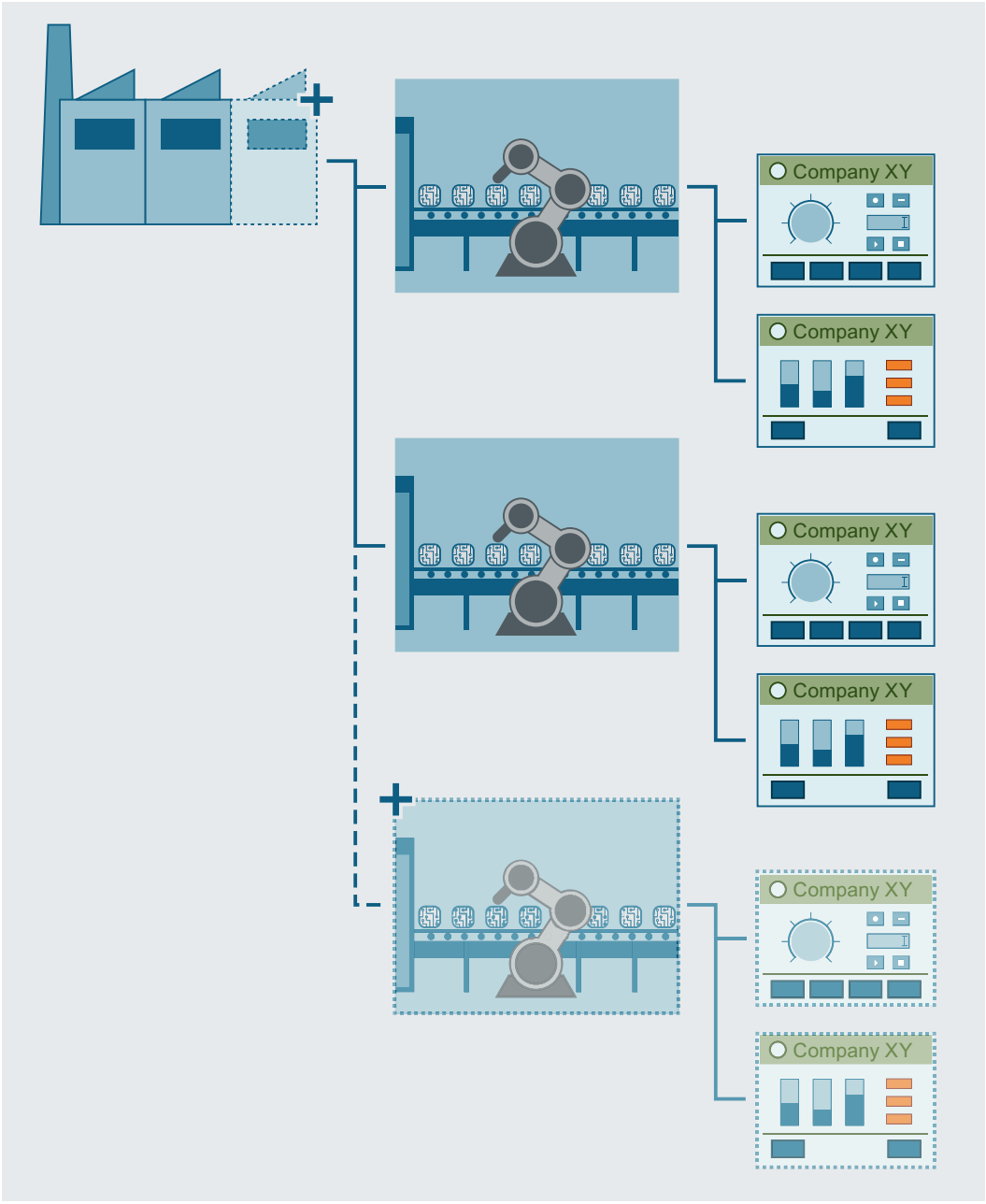
- 有使用 STEP 7 和 WinCC 进行组态的经验。
- 具备 Visual Basic Script (VBS) 的基本知识。

3.3 示例：使用 SiVArc 生成可视化

示例场景

目前，一家印刷电路板工厂将扩大生产规模，引入第三条生产线。该公司委托外部工程部根据现有控制程序和可视化设计，针对此次扩建执行可视化处理。工程公司的任务包括以下要求：

- 这是一家新客户。此前并未实施同类项目。
- 客户将在可视化中融入新的企业设计。
- 客户希望优化公司内部的标准化程序。



设计参数化解决方案

这家工程公司决定改造现有 SiVArc 示例项目，从而实施这项任务。为此，他们为 PLC 程序员和可视化专家分配了重要任务，即分析用户程序和可视化设计。

他们将精诚合作，确定以下信息：

- 布局模板数量，取决于使用的 HMI 设备
- 所需外部变量

- 外部变量的命名约定
- 用户程序中用于可视化的文本源
- 针对生产模板的程序块分配
- 生成模板中的 SiVArc 表达式结构
- SiVArc 项目中的存储结构

此后，可视化专家确定所需生成模板的数量和类型、SiVArc 规则和 SiVArc 变量。

实施参数化解决方案

PLC 程序员调整用户程序，将其应用于设计解决方案：

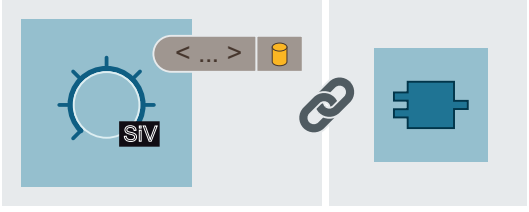
- 将 PLC 变量设置为“可通过 HMI 访问”
- 如有必要，检查 STEP 7 中的文本源并使其保持一致
- 优化项目树中程序块的存储
- 扩展现有库

可视化专家在画面模板中针对多个 HMI 设备实施客户的企业设计。

根据画面模板为每部设备创建专属定位方案。

调整标准对象示例项目的生成模板，使其适应可视化设计。

生成模板和函数块在画面规则中彼此相关联。



SiVArc 生成模板

引用文本源（如：数据块）的 SiVArc 属性


函数块

结果

根据 SiVArc 示例项目创建了一个客户特定的 SiVArc 新项目，特点是灵活敏捷。现在，微调 SiVArc 项目如即可进一步扩展工厂和用户程序。

3.4 示例：使用 SiVArc 生成变量

示例场景

工厂建筑人员在调试过程中频繁遭遇计划外的延期。经分析发现，现有变量命名约定的执行并未保持一致。重新创建变量将为 HMI 设备的存储容量造成沉重负荷。

该公司求助于一家工程公司，希望对变量名称进行标准化并重新链接。

停机时间应缩短至最低并且 HMI 设备具有空闲存储空间。

解决方案理念

这家工程公司分析了用户程序并将所需变量设置为“可通过 HMI 访问”。

根据使用的 PLC 变量、UDT 或数组类型，工程师针对变量名称的同步进行了组态。

SiVArc 开始生成变量。SiVArc 仅生成执行可视化的必要变量。

所需变量名称基于变量命名概念，通过 SiVArc 表达式生成。

说明

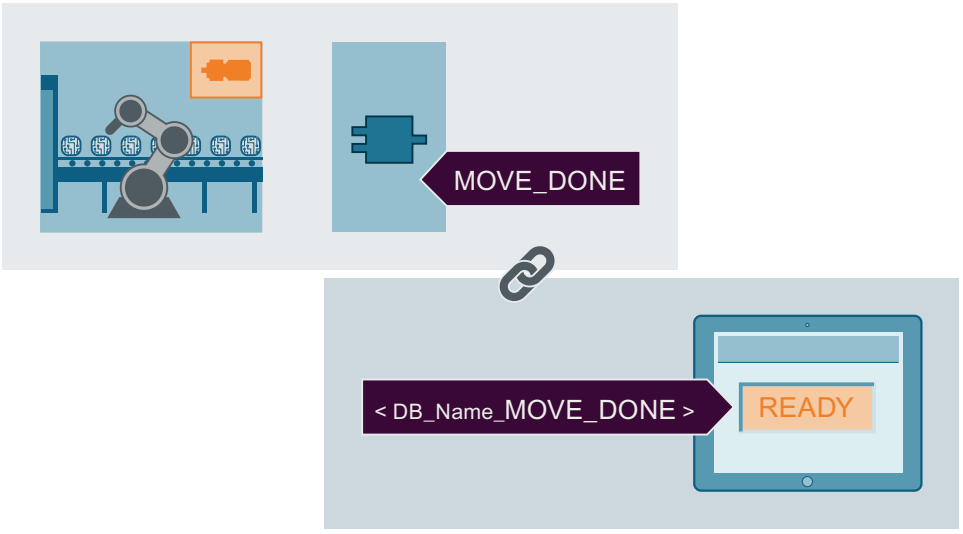
变量名称

WinCC 支持的字符数要比 STEP 7 支持的字符数少。如果 PLC 变量名称中使用的是 WinCC 不支持的字符，则会在生成外部变量名称时删除该字符。这可能会导致有多个变量具有相同名称。这会产生错误，因为 SiVArc 不生成具有相同名称的变量。

为 PLC 变量分配名称时，只能使用 WinCC 支持的字符。

结果

所需变量均经统一命名。针对 PLC 变量的引用可在 WinCC 项目的互连点处读取。



函数块

WinCC 项目中仅包含真正不可或缺的变量。可通过 SiVArc 对变量执行进一步处理和连续调整。

参见

变量生成的原理 (页 185)

3.5 通过 SiVArc 组态 HMI 解决方案

简介

通过 SiVArc 组态 HMI 解决方案需要标准化项目。项目的标准化程度越高，使用 SiVArc 创建可视化就越轻松高效。

要求

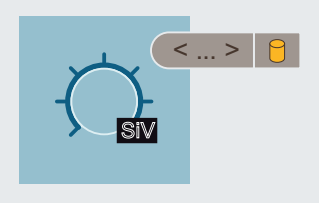
- 设备为标准设施。
- 已创建结构化用户程序。
- 已创建可视化和操作规则。
- 通过库可访问用户程序中的标准块。

- 通过库可访问标准应用程序的面板。
- 项目为标准化、可传送项目。

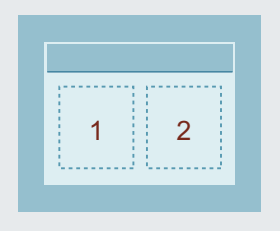
步骤

要通过 SiVArc 生成 HMI 解决方案，请按照下列步骤操作：

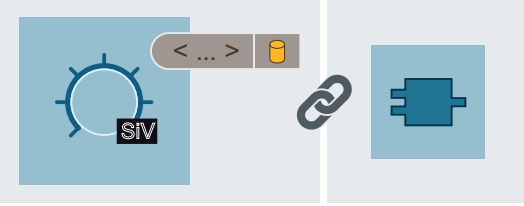
1. 设计布局。
   - 使用哪些 HMI 设备？
   - 如何在画面中实现企业设计？
   - 需要多少个用于画面的生成模板？
   - 需要多少个定位方案？
2. 定义要通过 SiVArc 生成的外部变量。
3. 创建 HMI 对象的生成模板并将其存储在库中。



4. 创建画面的定位方案并将其存储在库中。



5. 创建用于将生成模板和函数块相关联的画面规则。

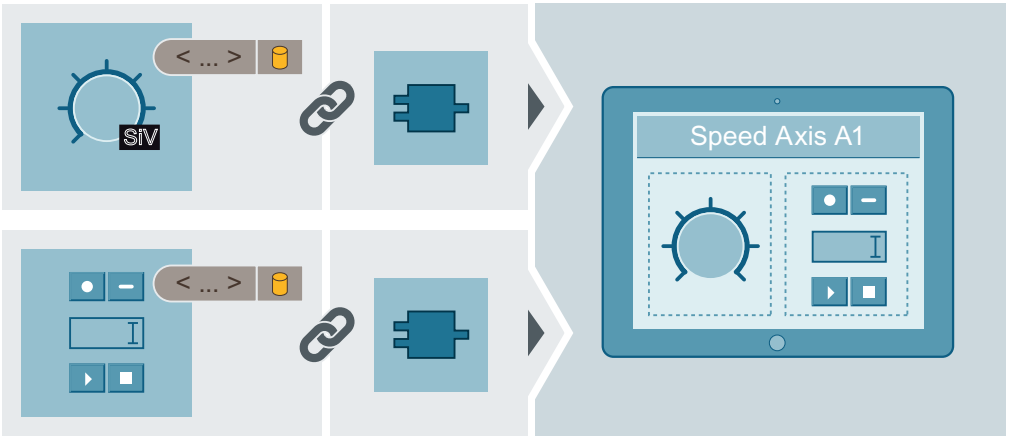


6. 创建控制生成的变量存储的变量规则。
7. 创建将收集的 HMI 对象从库复制到项目的复制规则。
8. 定义文本列表条目。
9. 生成完整可视化或仅为所选设备生成可视化。

结果

生成的 HMI 对象已创建于项目树中，并且已被标记为生成的对象。

已根据生成画面中的定位方案排列生成的画面对象。



SiV	SiVArc 生成模板
< ... >	引用文本源（如：数据块）的 SiVArc 属性
	函数块

进一步处理

说明

所生成对象的后续名称更改

如果更改了生成的 HMI 对象的名称，则在下次 SiVArc 生成中将重新创建该对象并且互连。重命名的对象仍然可用。

仅在用户程序中更改所生成对象的名称。

每次更改了 SiVArc 项目或用户程序后，均将重新启动生成。

SiVArc 将处理更改的信息，替换现有的生成并生成其他 HMI 对象（必要时）。

可使用全局搜索功能搜索 SiVArc 已组态的表达式。

参见

规划布局 (页 61)

生成模板的创建 (页 95)

创建规则 (页 166)

生成可视化 (页 209)

安装 4

4.1 安装 SiVArc

简介

将安装介质插入相应驱动器后，“SiVArc”附加软件包的安装程序便会立即启动。

安装 SiVArc 需要有效的许可证。使用“Automation License Manager”管理许可证密钥。

说明

版本兼容性

SiVArc 版本仅与相应的 STEP 7 和 WinCC Professional、WinCC Advanced 或 WinCC Unified 版本兼容。

若要升级 TIA Portal 的版本，还必须升级 SiVArc 的版本，反之亦然。如果卸载 WinCC 或 STEP 7，同时也会卸载 SiVArc。

选择并排安装，以便使用不同版本的 TIA Portal。

要求

- 已安装 STEP7 Professional V17.0。
- 已安装 SIMATIC WinCC Professional V17.0、SIMATIC WinCC Advanced 或 SIMATIC WinCC Unified V17.0。

步骤

要安装“SiVArc”附加软件包，请按以下步骤操作：

1. 将安装数据介质置于驱动器中。
   要启动手动安装，请双击资源管理器中的“Start.exe”。
2. 选择安装语言并单击“下一步”(Next)。
3. 选择所需产品并单击“下一步”(Next)。
4. 要继续安装，请阅读并接受所有许可协议，并单击“下一步”(Next)。
   如果 TIA Portal 安全和权限设置阻止安装，则会打开安全设置对话框。
5. 要继续安装，请接受对安全和权限设置的更改。
6. 检查概览中所选的安装设置。

7. 根据需要更改设置，然后单击“安装”(Install)。
   安装随即启动。
   系统显示安装完成。
8. 根据要求重启 PC 或退出安装。

结果

PC 上已安装“SiVArc”附加软件包。

如何处理现有 SiVArc 项目

通过基本安装，即使未进行 SiVArc 安装，也可以在 TIA Portal 中打开现有 SiVArc 项目。

如果随后使用 SiVArc 打开项目，则所有 SiVArc 功能会再次激活。

要升级 SiVArc 项目，需要安装 SiVArc。

基本安装包括以下软件包：

- STEP7 Professional
- SIMATIC WinCC Professional
  或
- SIMATIC WinCC Advanced
- SIMATIC WinCC Unified

要删除项目中对 SiVArc 的引用，需从项目中删除所有 SiVArc 组态。当通过基本安装打开项目时，将不再输出有关未安装 SiVArc- 的信息。

元素和基本设置 5

5.1 SiVArc 设置

生成设置

在“设置> SiVArc > 生成 > “警告设置”(Settings > SiVArc > Generation > "Warning settings") 中，可启用以下选项：

- 画面对象存在时隐藏警告 - 生成的画面中存在画面对象时适用
- 多语言语境中变量未定义时隐藏警告 - 特定语言语境中的变量适用
- 画面大小发生变更时隐藏警告 - 跨设备盛彻那个不同画面大小时适用
- 文本条目无法解析时隐藏警告 - “S7 插入式编辑器 > 文本定义”(S7 plug-in editor > Text definitions) 中未定义任何文本时适用

库类型设置

“SiVArc 生成时使用默认的库类型版本”(Use default version of library type for SiVArc generation) - SiVArc 生成时可使用面板中库类型的默认版本。

WinCC Unified 的采集周期和模式设置

将为包含 Unified HMI 设备和 Advanced 设备的项目组态块变量成员设置或项目级设置；如果 Advanced 设备支持获取周期：

- 默认情况下，将为 Unified 设备中的 HMI 变量组态 T1s 的获取周期

将为包含 Unified HMI 设备和 Advanced 设备的项目组态块变量成员设置或项目级设置；如果 Unified 设备支持获取周期：

- 默认情况下，将为 Advanced 设备中的 HMI 变量组态 1s 的获取周期

可在项目级或块级组态获取模式。在项目级，类型包括：

- 循环操作
- 按需

在 **PLC 块级**对于变量成员设置，可选择 “**使用公共组态**”(Use common configuration) 选项将各输入变量的获取周期和模式设为相同设置。

更多关于 WinCC Unified 设备功能的信息，请参见 *TIA portal 用户指南*.。

SiVArc 支持的、为 WinCC Unified 设备组态的对象称为 Unified 对象。例如：Unified 按钮用于画面规则的模板副本中。对 SiVArc 对象的所有更改均同样适用于 Unified 对象。对于 Unified 设备，可组态画面插件属性。可在插件属性中组态 SiVArc 支持的表达式，如 `HmiDevice.Name`、`HmiDevice.Type`、`HmiApplication.Name`、`HmiApplication.Type`、`HmiDevice.Resolution.Width`、`HmiDevice.Resolution.Height`。

5.2 SiVArc 中的用户管理控制 (UMAC)

UMAC 在 SIVArc 中的作用

通过使用用户管理控制 (UMAC)，将只允许授权用户访问 SiVArc 相关功能。要在 SiVArc 中使用“用户管理控制”，用户需要具备 SiVarc 管理权限。此权限可由作为工程组态标准 (ES) 管理员角色的用户进行分配。关于使用“用户管理控制”的更多信息，请参见 TIA Portal 在线帮助。

说明

- 用户必须分配工程组态标准角色以访问工程组态属性和功能。若未提供此访问权限，则用户将无法打开项目，且将显示错误消息。
- 如果用户具有工程组态管理员权限，但没有 SiVArc 管理权限，可尝试关闭项目并重新打开，将显示一条与未经授权的访问相关的错误消息。

注意
专有技术保护 将 TIA portal V15 及更低版本的项目转换为 V15.1 时，具有“专有技术保护”特性的凭证集将被自动删除。请确保通过 UMAC 保护项目。

5.3 SiVArc 编辑器

5.3.1 “画面规则”编辑器

说明

在“画面规则”(Screen Rules) 编辑器中，为多个设备定义生成画面中的 SiVArc HMI 对象所使用的画面规则。规则包括：

- 名称
  画面规则的唯一名称
- 程序块
  在用户程序的任何位置都可调用的 FB 或 FC。
- 画面对象
  生成的 HMI 对象的主副本或类型。主副本或类型必须存储在库中。
- 画面
  用于生成 HMI 对象的画面的生成模板。生成模板必须存储在库中。
- 布局字段
  包含在画面定位方案中的布局字段。使用布局字段可指定要生成的 HMI 对象的定位。
- 条件（可选）
  处理此画面规则时进行求值的 SiVArc 表达式。如未指定条件，则将始终执行画面规则。此条件适用于整个规则组。用户可为规则组中的各项规则细化条件。
  WinCC Unified 画面支持 SiVArc 表达式。
- 注释（可选）
  各画面规则注释

需要时，可单击工具栏中的图标显示以下列：

- PLC
  针对选定的控制器执行画面规则。如果未选择任何控制器，则该规则适用于项目中的所有控制器。
- “HMI 设备”(HMI device)
  针对选定的 HMI 设备执行画面规则。如果未选择任何 HMI 设备，该规则适用于项目中的所有 HMI 设备。
- “HMI 设备类型”(HMI device type)
  如果项目中有多个相同类型的 HMI 设备，则还可选择 HMI 设备的类型。生成期间，将进行相关检查并指示某个规则可应用于 HMI 设备还是控制器。

如果要为程序块生成没有画面对象的画面，请将“画面对象”(Screen object) 字段留空。

在不以手动方式保存表达式的情况下，如果切换编辑器，则所有支持表达式的编辑器默认会将数值保存到相应属性中。

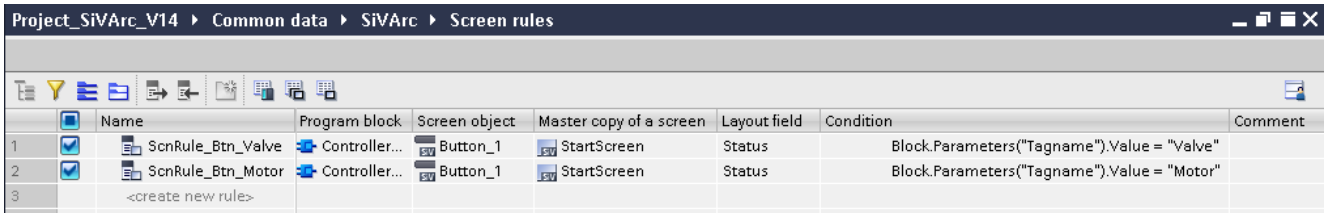
通过右键单击“规则 > 插入新规则/插入新规则组”(rule > insert a new rule/insert a new rule group)，可在所有规则编辑器中的所有所需索引处添加新规则/规则组。

1. 在索引点添加新规则/规则组时，新规则/规则组将立即插入所选索引的下方。
2. 通过选择分组添加新规则/规则组时，新规则/规则组将立即添加到所选规则组的第一个索引中。

示例

可使用程序块控制阀门或电机。根据程序块的使用，画面将生成标注有“打开阀门”(Open valve) 或“启动引擎”(Start engine) 的按钮。

阀门符号和电机符号需要相应的画面规则。



如果 SiVArc 在 HMI 对象生成期间处理程序块，SiVArc 会评估每个画面规则的条件。在此示例中，通过输出定义程序块的使用，例如 `Block.Parameters("Tag name").Value = "Valve"`。这种情况下，第一个画面规则的条件适用，然后会生成标记有“打开阀门”(Open valve) 的按钮。“条件”的最大字符数限制为 1000。

参见

SiVArc 变量 (页 114)

在 SiVArc 编辑器中编辑视图 (页 286)

导出和导入 SiVArc 规则 (页 205)

编辑和管理 SiVArc 规则 (页 203)

5.3.2 “变量规则”编辑器

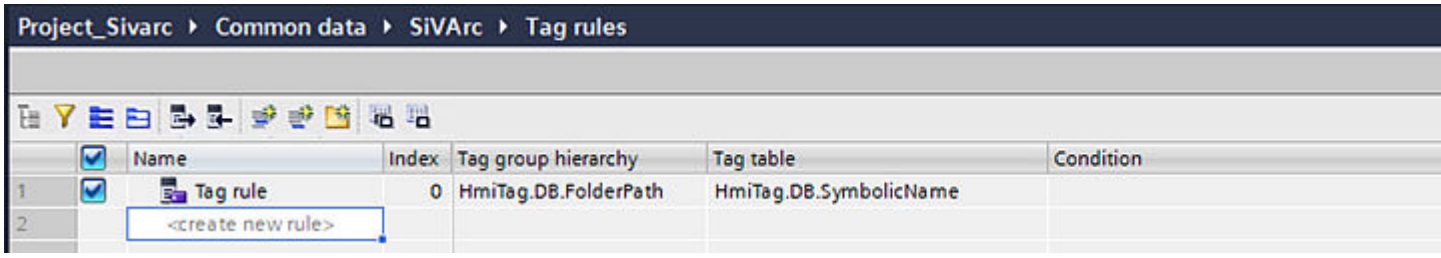
说明

在“变量规则”(Tag rules) 编辑器中，可定义通过 SiVArc 生成的外部变量所采用的变量规则，这些外部变量以结构形式存储。

双击项目树中的“公共数据 > SiVArc > 变量规则”(Common data > SiVArc > Tag rules) 可打开“画面规则”(Tag rules) 编辑器。

变量规则包含下列元素：

- 名称
  变量规则的唯一名称
- 索引
  指定规则执行顺序。可在表格行中通过拖放操作更改索引。
- 变量组
  生成外部变量的变量组的名称
- 变量表
  生成外部变量的变量表的名称
- 条件（可选）
  处理此变量规则时进行求值的 SiVArc 表达式
- 注释（可选）
  各变量规则注释
- 工具栏包含的设备特定列会显示 TIA portal 项目中的可用 HMI 设备和类型。“SiVArc 属性”(SiVArc Properties) 区域下提供相同的选项。



参见

SiVArc 对象属性 (页 264)

在 SiVArc 编辑器中编辑视图 (页 286)

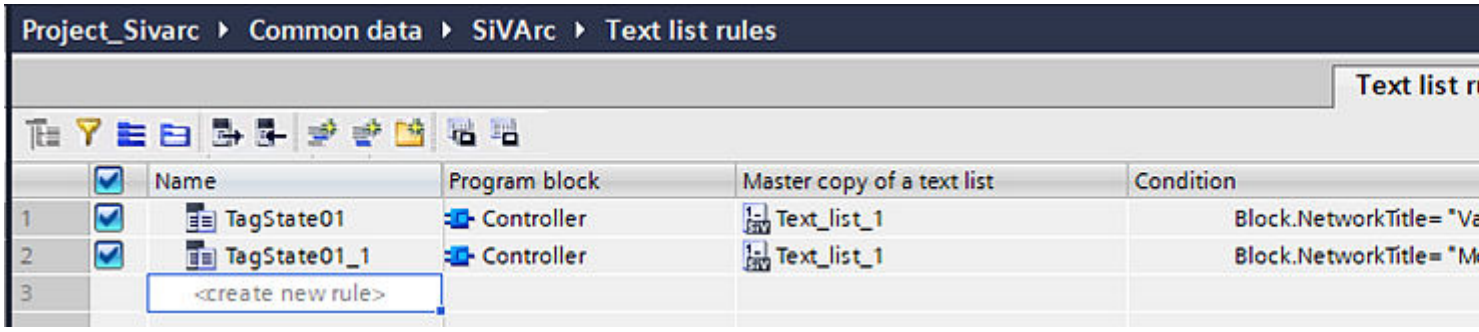
导出和导入 SiVArc 规则 (页 205)
编辑和管理 SiVArc 规则 (页 203)
“复制规则”编辑器 (页 36)
检查结果 (页 222)

5.3.3 “文本列表规则”编辑器

说明

在“文本列表规则”(Text list rules) 编辑器中，定义为不同设备生成文本列表所用的 SiVArc 规则。文本列表规则包括：

- 名称
  文本列表规则的唯一名称
- 程序块
  在用户程序的任何位置都可调用的 FB 或 FC。
- 文本列表
  在生成期间，文本列表的主副本保存在“文本和图形列表”(Text and Graphic Lists) 编辑器中。
- 条件（可选）
  处理此文本列表规则时进行求值的 SiVArc 表达式。如未指定条件，则将始终执行文本列表规则。
- 注释（可选）
  各文本列表规则的注释
- 工具栏包含的设备特定列会显示 TIA portal 项目中的可用 HMI 设备和 PLC。“SiVArc 属性” (SiVArc Properties) 区域下提供相同的选项。



参见

在 SiVArc 编辑器中编辑视图 (页 286)

导出和导入 SiVArc 规则 (页 205)

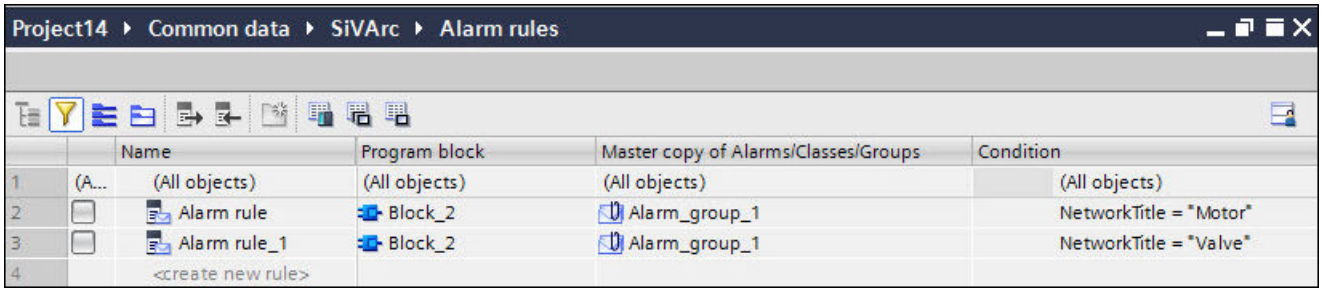
编辑和管理 SiVArc 规则 (页 203)

5.3.4 “报警规则”编辑器

简介

在“报警规则”(Alarm rules) 编辑器中，可定义为不同 HMI 设备生成报警时遵循的 SiVArc 报警规则。“报警规则”编辑器的组成为：

- 名称 - 报警的唯一名称。
- 规则触发器 - 在用户程序的任何位置都可调用的函数块或函数调用。
- 报警/类别/组的主站副本 - 报警、类别或组的主站副本可以浏览和获取。
- 条件（可选） - 在进行报警规则时评估 SiVArc 表达式。如未指定条件，则将始终执行报警规则。
- 注释（可选项） - 报警规则的单个注释。



以下功能在“报警规则”编辑器中可用：

- 只能感知支持
- 自动完成（拖动单元格的角以自动完成值）
- 剪切和复制/粘贴选项
- 仅当对象不同时，创建具有相同条目的单个规则。可以选择“报警/类别/组的主副本”(Master copy of Alarms/Classes/Groups) 下的对象。如果选择多个报警对象，编辑器中将该条目显示为“多选”(Multiple Selection)，对于单个选择，将显示对象名称。

报警生成的要求

- PLC 构建的程序块（FB 和 FC）已组态并连接到 HMI 设备。
- 在 WinCC 的“报警”(Alarm) 编辑器中创建和组态报警。
- 在全局库中创建和组态主副本（报警或块类型）。

您可以使用“报警规则”(Alarm rules) 编辑器在全局库中组态和存储报警规则。在报警规则组态期间，S7 设备的 PLC 块映射到存储在库中的主副本，并定义条件。在 SiVArc 生成过程中，根据用户定义的条件；系统相应地处理规则并生成报警对象。有关报警规则生成的更多详细信息，请参见“生成矩阵” (页 38)部分。

说明

- 只有在规则满足用户定义条件之后才对报警对象进行处理。
- 在选择程序块时，您可以在 PLC 块或 PLC 类型之间进行选择。

常规注意事项

- 不能删除系统生成的报警类别，而仅可以编辑该报警类别。
- SiVArc 属性完全与工程系统属性间接相关。打印处静态值的选项和打印出变量与工程系统属性类似。
- 也可以使用 SiVArc “插入式”编辑器组态常规报警。
- 在 SiVArc 生成期间，如果解雇哦对象名称与默认报警类别名称相同，SiVArc 将显示错误。
- 在“类别组”(Class group) 中显示报警组构成类似类别。SiVArc 属性运行组态确认、分配状态和抑制报警组合子组。
- 任何新添加的报警类别都将自动显示在“类别组”(Class group) 区域中。
- 在 SiVArc 中可以编辑存在的报警对象。
- 当创建了相同的报警规则，系统显示以下错误：
  “<'object name'> 的报警规则与 <'object name'> 相同，所以不建议生成并可以删除。”
- 对于枚举型属性，不支持剪切、复制、粘贴。

参见

创建报警规则 (页 195)

“生成矩阵”编辑器 (页 38)

5.3.5 “SiVArc 表达式”编辑器

说明

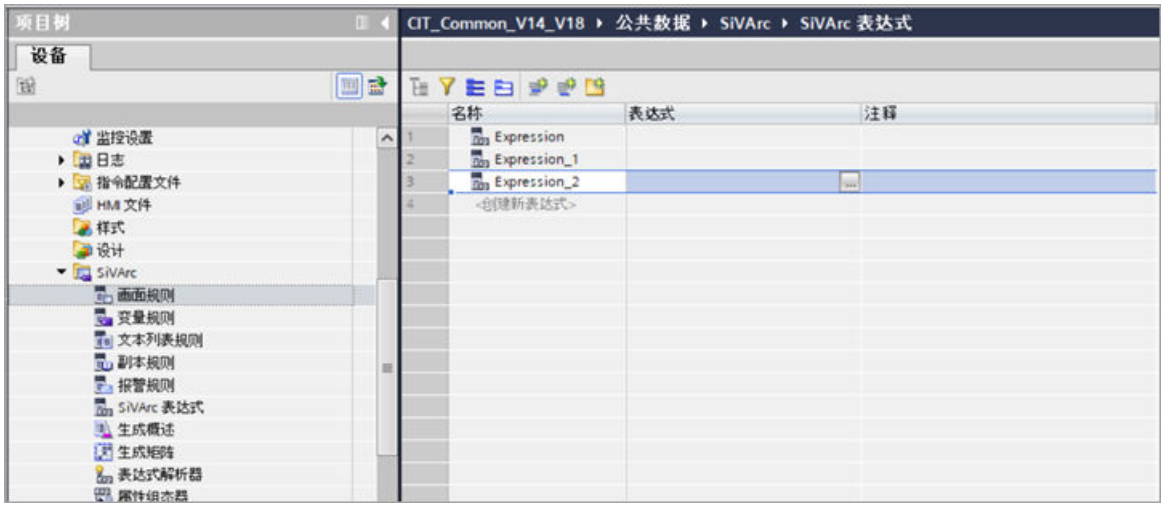
SiVArc 表达式编辑器可用于添加、定义和分组表达式，这些表达式可用于项目库和全局库中。可通过调用表达式名称在全局使用表达式。

“SiVArc 表达式”的组成为：

- 名称：
  表达式的唯一名称
- 表达式：
  输入 SiVArc 支持的表达式。表达式编辑器支持 StringPos(Parameter1,Parameter2)。
  情形 1：使用 StringPos 函数组态表达式时：StringPos (Parameter1, Parameter2)；系统返回 Parameter1（主字符串）中 Parameter2（子字符串）第一个字符的位置。
  示例：StringPos("Sivarc_Project","Project")=8，原因：（“Project”是“Sivarc_Project”的子字符串 – 返回 8，原始字符串中“P”的位置）
  情形 2：使用“StringPos”组态表达式时，如果参数提取大小写不匹配的字符或字符串不可用，系统会返回 0。
  示例：StringPos("Sivarc_Project","Por")=0，原因：（“Por”不是“Sivarc_Project”的子字符串 – 返回 0）
  情形 3：使用“StringPos”组态表达式时，如果“StringPos”函数中的两个字符串或任一字符串为空，则该函数将返回 -1。
  示例：StringPos("","")=-1，原因：（两个参数均为空）
- 注释：
  各 SiVArc 表达式的注释

工具栏包括分组、过滤、收起所有展开的行、展开所有采用层级结构的行、在上方插入、在下方插入。在表达式编辑器中可以执行下列操作：

1. 在表达式编辑器中，单击“创建新表达式”(Create new expression) 可添加新表达式。
2. 可选择通过单击工具栏中的组选项的方式来选择表达式并对其进行分组。
3. 右键单击表达式行 >“添加新表达式组”(Add a new expression group) 即可创建组。
4. 右键单击组 >“插入新表达式组”(Insert a new expression group) 即可在现有组中创建组。
5. 可使用工具栏选项在任何选定行的上方/下方插入新的表达式。



可将已创建的表达式作为对象拖放到项目/全局库中。在这种情况下，对象在表达式编辑器的对象选择器中可用。表达式可在多个实例中重复使用。此类表达式称为全局表达式。可在画面、报警、变量、复制规则的规则网格条件列中使用全局表达式。

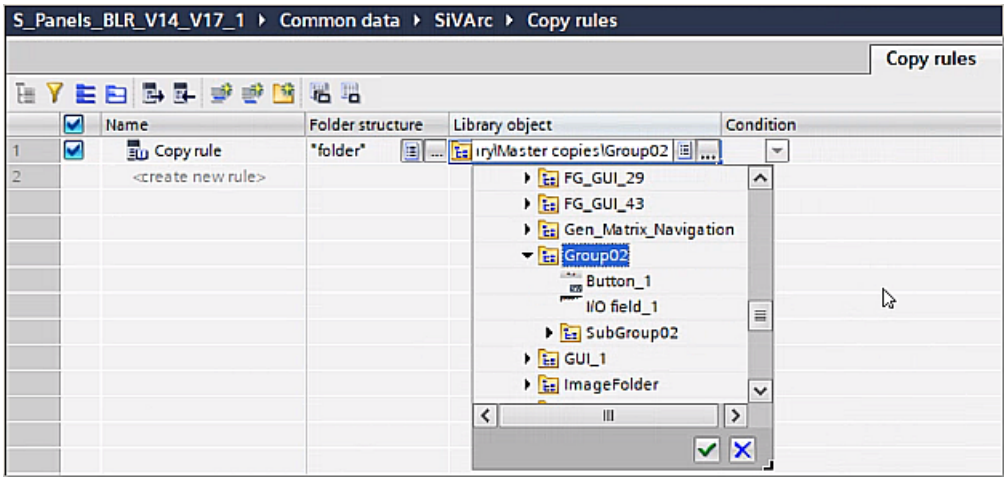
5.3.6 “复制规则”编辑器

说明

规则包括：

- 名称
  复制规则的唯一名称
- 文件夹结构
  可以通过手动或在表达式的支持下完成用户定义的文件夹结构。这适用于支持创建文件夹的库对象。
  如果“文件夹结构”(Folder structure) 列中的值为空，则 SiVArc 在 PNV 默认文件夹中将从根节点开始生成所有组态的对象，直至嵌套层级为“4”的子文件夹。
  如果要按照项目库中的组态复制带有文件夹结构的对象，则可使用相应的表达式对文件夹结构列进行组态。
  在文件夹结构列中，也可选择组态 SiVArc 生成时所需的任何文件夹结构。
- 项目库
  该文件夹结构中包含层级“0”处的根目录以及根目录下目录层级分别为“1”、“2”等各级子文件夹。SiVArc 生成时仅支持“4”层文件夹嵌套。

- 库对象
  生成的模板副本或对象类型，或包含“**库对象 > 布局字段**”(library objects > layout fields)的库文件夹。
  如果布局字段当前为可编辑模式，则使用之前已发布的布局字段。模板副本或库类型必须包含在项目库中。Unified 画面可用作对象选择器中的对象，SiVArc 生成时，仅会在 Unified 设备上生成画面。支持将图形作为库对象。
- 条件（可选）- 在处理复制规则时评估的 SiVArc 表达式。如未指定条件，则将始终执行复制规则。
- 注释（可选）
  复制规则各项注释



必要时，可单击工具栏中的图标显示以下列：

- HMI 设备
  针对选定的 HMI 设备执行复制规则。如果未选择任何 HMI 设备，该规则适用于项目中的所有 HMI 设备。对于 Unified 设备，如果复制规则使用系统文本列表，则 SiVArc 生成期间，会出现错误。
- HMI 设备类型
  如果项目中有多个相同类型的 HMI 设备，则还可选择 HMI 设备的类型。生成期间，将进行相关检查并指示某个规则可应用于 HMI 设备还是控制器。

参见

在 SiVArc 编辑器中编辑视图 (页 286)

导出和导入 SiVArc 规则 (页 205)

编辑和管理 SiVArc 规则 (页 203)

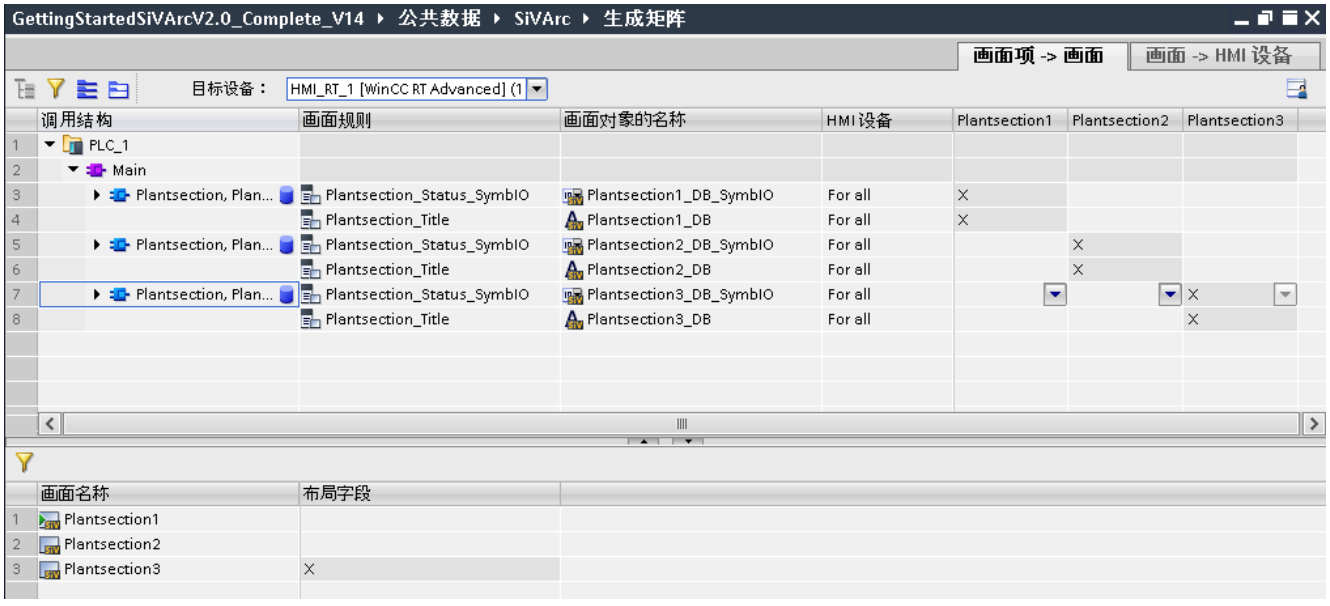
5.3.7 “生成矩阵”编辑器

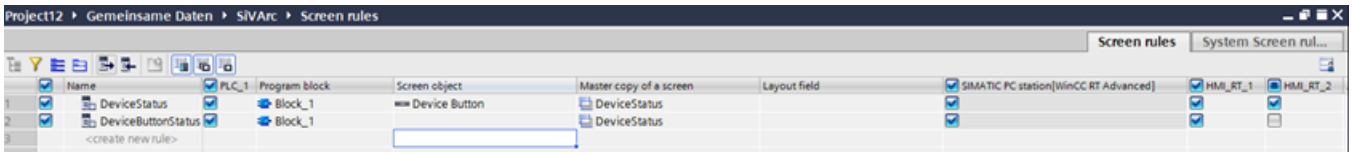
“画面对象 -> 画面”选项卡

在编辑器的工具栏中，可选择在“目标设备”(Target device) 中待显示的 HMI 设备（旧型号/WinCC Unified）的矩阵图。此外，SiVArc 也可显示所有设备的设备类型。

在此选项卡中，可将生成的画面对象分配给其他画面。该选项卡包含以下列：

- 调用结构
  为每行显示供用户程序调用并用于生成画面对象的程序块实例。
- 画面规则
  显示适用于每个块实例执行的画面规则。
- 画面对象的名称
  显示生成的画面对象。
- HMI 设备
  为每个画面对象列出为其生成了画面对象的 HMI 设备。
- 画面列
  每个画面都会在单独一列中显示。各个列按字母顺序进行排序。
  - “X”：画面对象不能位于布局字段中。
  - “<布局字段的名称>”：画面对象包含在特定的布局字段中。





“画面 > HMI 设备”选项卡

在编辑器的工具栏中，可在“设备类型”(Device type) 下选择要为其显示矩阵的 HMI 设备类型。编辑器随即会显示此类型的所有 HMI 设备的画面。

在此选项卡中，可将生成的画面分配给其他 HMI 设备。该选项卡包含以下列：

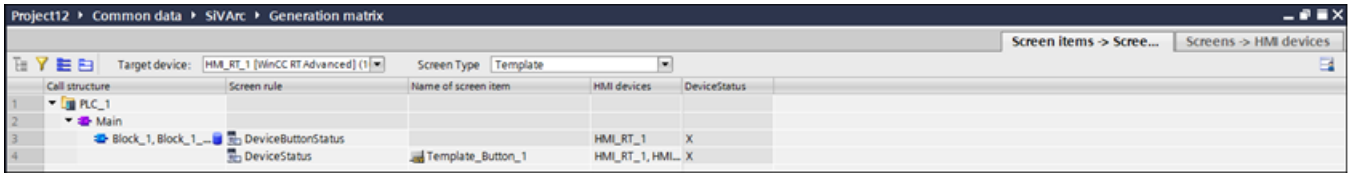
- 画面
  显示生成的画面。
- HMI 设备
  显示 HMI 设备（包括 WinCC Unified）。每个 HMI 设备都会在单独一列中显示。各个列按字母顺序进行排序。

模板和弹出画面

生成矩阵包含两个功能：

允许用户将模板/弹出画面或画面对象分配给不同的目标 HMI 设备。可以将模板/弹出画面对象分配给不同的目标 HMI 设备，方法与画面类似。编辑器工具栏显示“画面类型”(Screen Type) 下拉选项，包括画面、模板和弹出画面，如以下截图所示：

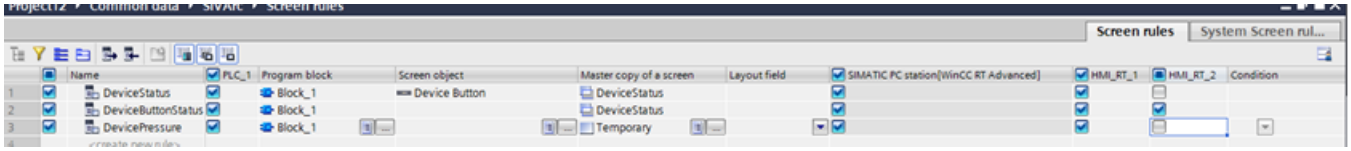
- 矩阵 1 - 画面项 -> 画面 - 允许用户在不同 HMI 设备的画面上分配画面项
- 矩阵 2 - 画面 -> HMI 设备 - 允许用户将画面分配给不同的 HMI 设备



情况：在不访问画面的情况下将对象导航到不同的 HMI 设备

调试工程师无权访问画面规则，因此只能选择使用 SiVArc 可视化将画面对象移动到不同的目标 HMI 设备。请考虑以下画面规则组态：

1. 按如下所示组态画面规则：
   - DeviceButtonStatus 包含仅在 HMI 设备 1 中生成的模板画面
   - DeviceStatus 包含画面对象“Template_button_1”，并在 HMI 设备 1 中生成



2. 生成 SiVArc 时，会在 HMI_RT_1 上生成带有“设备按钮”(Device Button) 画面项的“DeviceButtonStatus”画面，在 HMI_RT_2 上生成“设备状态”(Device Status)。
3. 创建一个包含“设备按钮”(Device Button) 画面项的目标画面。将“单击”(Click) 组态为“激活画面”(Activate Screen)，指向目标画面。

   说明

   组态画面时，可以直接为画面项组态事件。但是，若使用模板和弹出画面，必须使用目标画面将画面项导航至模板画面。由于模板和弹出画面的事件无法直接组态，因此必须使用目标画面。

4. 在“画面规则”(Screen rules) 中将目标画面组态为“临时”(Temporary)，然后将目标 HMI 设备选作 HMI_RT_1。
5. 单击“SiVArc 设置 > 矩阵设置 > 生成导航对象”(SiVArc settings > Matrix settings > Generate navigation objects)。
6. 转至“生成矩阵 > 画面 > HMI 设备”(Generation matrix > Screens-> HMI devices)，选择“画面类型”(screen Type) 作为“画面”(Screen)，然后选择 HMI_RT_2 复选框，如下屏幕截图所示：

7. 在 HMI 设备 2 上执行 SiVArc 生成。请注意以下说明：
   - 步骤 2 中创建的画面项将移动到“画面管理”(Screen management) 文件夹下的画面中。
   - 将生成带有画面项的画面，该画面位于“画面”(Screens) 文件夹下。

   模板画面项“设备按钮”(Device Button) 通过目标画面移动到模板画面“DeviceButtonstatus”，即“画面 > HMI 设备”(Screens -> HMI devices) 中的“临时”(Temporary)。

说明

- 以上步骤也适用于弹出画面。
- 对于复制规则，在修改画面规则后，必须在生成矩阵（包括导航对象）中为相应修改的画面手动设置设备。

5.3.8 表达式解析器

说明

表达式解析器可提供给定表达式源对象的 SiVArc 表达式的解析值。会对输入的表达式进行解析并即时显示结果，而不会在编辑器中生成 SiVArc 。

“表达式源对象”可以特定于块实例、HMI 设备、IO 设备、库对象。

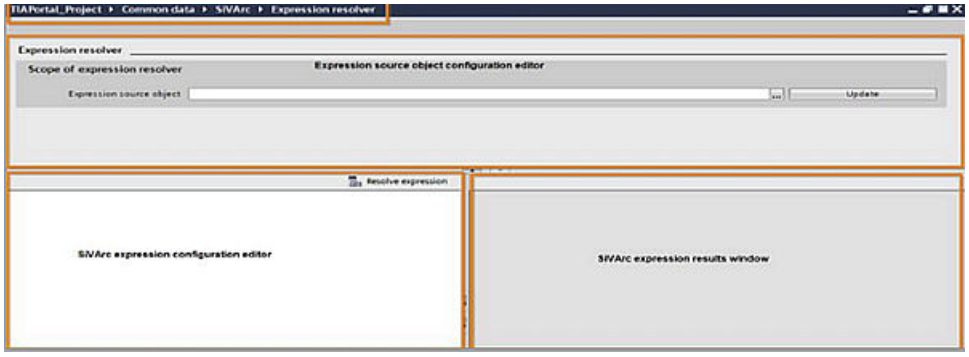
表达式解析器编辑器分为 3 个区域：

1.表达式源对象组态编辑器

2. SiVArc 表达式组态编辑器

3.表达式结果窗口

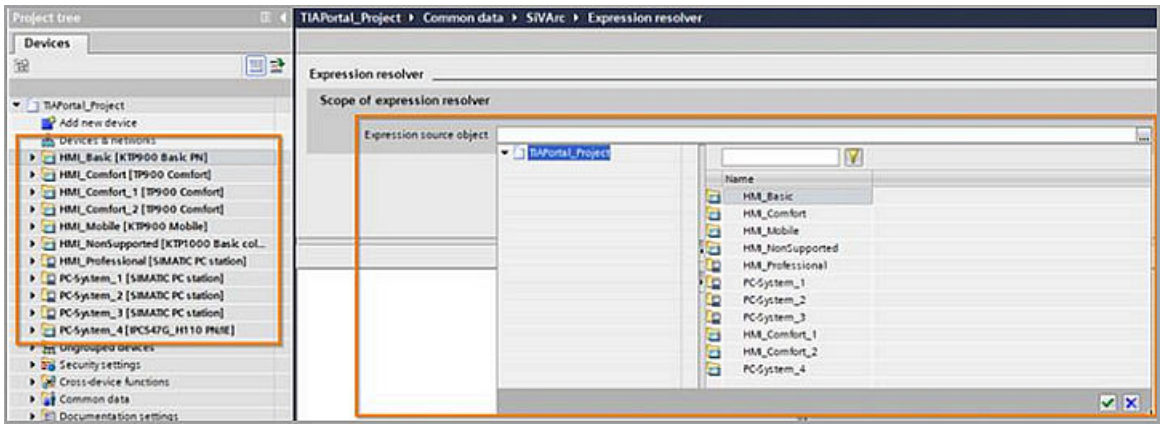
以下屏幕截图显示了表达式解析器编辑器：



源对象组态

表达式解析器仅支持块实例和 HMI 设备作为表达式源对象。

HMI 设备作为源对象 - 组态有 HMI 设备的项目，在左侧/层级视图中选择项目节点后，对象选择器会在右侧显示可用的 HMI 设备，如下面的屏幕截图所示。



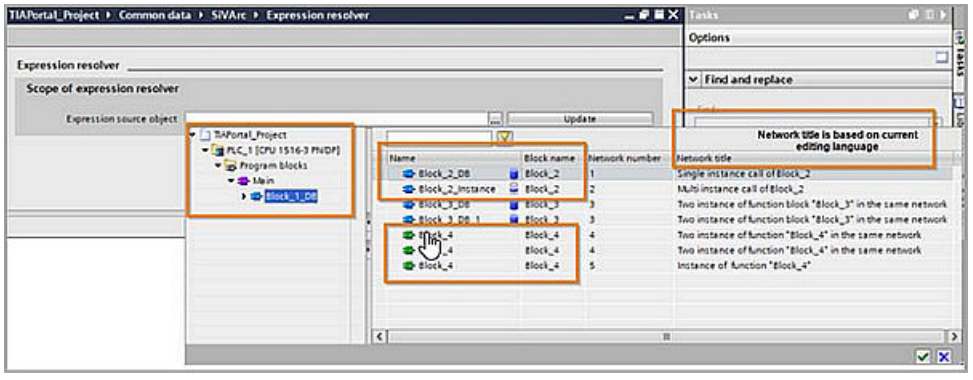
程序块调用作为表达式源对象

对象选择器会显示在 TIA Portal 项目中组态的 PLC 的调用结构。

在树形视图（左侧）中选择组织块或块调用后，对象选择器右侧会在各列中显示所选调用中的可用程序块调用，例如：

- 名称 (Name) - 显示实例名称（函数块调用的情况）和块名称（函数调用的情况）
- 块名称 (Block name) - 显示与调用相关联的块的名称
- 程序段数 (Network number) - 显示当前调用可用的程序段数
- 如果 PLC 程序中存在当前编辑语言对应的组态，则会显示当前调用的程序段标题

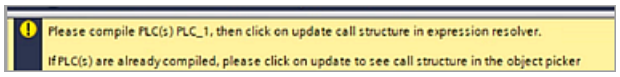
可选择使用各列上的过滤器。以下屏幕截图显示的对象选择器包含上述详细信息：



“表达式源对象”(Expression source object) 选择器仅显示项目中已编译 PLC 的调用结构。

如果某一 PLC 未编译，则对象选择器不会显示在该 PLC 中定义的组织块的调用结构。

如果有 PLC 未编译，表达式解析器会显示包含此类 PLC 信息的横幅。假定一个项目组态为使用“PLC_1”、“PLC_2”和“PLC_3”这 3 个 PLC。如果只有 PLC_1 已编译，则会显示包含相关消息的以下横幅：



首次打开表达式解析器编辑器时，编辑器会自动加载所有已编译 PLC 的调用结构。仅当有 PLC 未编译时，编辑器才会显示横幅，且对象选择器会根据加载的调用结构显示相同横幅。修改 PLC 程序时，表达式解析器不会显示新更改/已修改的更改。

如果希望解析器考虑修改的更改/新更改，则必须编译 PLC，并单击源对象组态编辑器中提供的“更新”(Update) 按钮。“更新”操作会加载已编译 PLC 的可用调用结构，并将使用同样的调用结构解析表达式。

SiVArc 表达式组态编辑器

组态编辑器支持智能感知、语法突出显示、复制、粘贴、删除操作，与现有的本地表达式选择编辑器类似。

组态编辑器的工具栏中包含解析按钮。输入表达式后，单击“解析表达式”(Resolve expression) 按钮可在结果窗口中查看解析值。

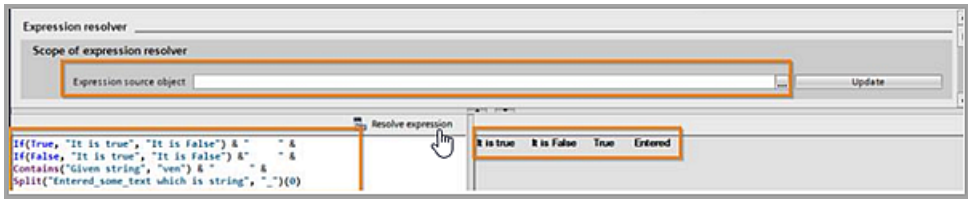
表达式结果窗口

显示在“表达式组态编辑器”(Expression configuration editor) 中输入的表达式的解析值。显示的值取决于在“表达式源对象”(Expression source object) 中选择的内容。

有关解析器的工作原理，请考虑以下场景：

案例 1：基于字符串的操作，没有任何表达式源对象

如果没有在“表达式源对象”(Expression source object) 中选择任何内容，但在 SiVArc 表达式编辑器中输入了表达式，单击“解析表达式”(Resolve expression) 按钮后会解析表达式，如下面的屏幕截图所示：



案例 2：基于 HMI 设备的表达式

如果在“表达式源对象”(Expression source object) 中选择了 HMI 设备，并在 SiVArc 表达式编辑器中输入了表达式，则会显示以下屏幕截图中所示的结果：

案例 3：基于块的表达式

如果在“表达式源对象”(Expression source object) 中选择了块，并在 SiVArc 表达式编辑器中输入了表达式，则会显示以下屏幕截图中所示的结果：

说明

除文本定义之外，解析器支持所有基于 SiVArc 支持的块的表达式、变量定义。

Block.DB.HMITagPrefix - 在 SiVArc 生成期间，会通过移除已解析值中的无效字符来解析表达式“`Block.DB.HMITagPrefix`”。解析器不会验证任何 HMI 设备的解析值。

Block.NetworkTitle 和 Block.NetworkComment - 基于编辑语言解析表达式。解析器获取已组态的程序段标题以及对所选编辑语言的注释，并在结果中显示同样的内容。

案例 4 无穷循环

如果 PLC 调用结构中存在无穷循环，SiVArc 生成会显示一条消息。

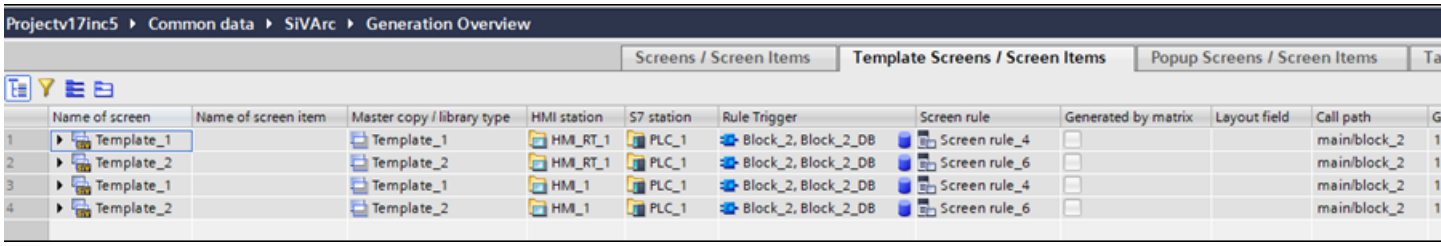
解析器支持 PLC 调用结构中的无穷循环。

5.3.9 生成概览

说明

初次生成可视化后，所有生成的画面对象都会列在生成概览中。生成的对象被分为“画面/画面对象”(Screens/Screen objects)、“变量”(Tags) 和“文本列表”(Text lists) 选项卡。

完成生成过程后，生成概览还会通过各个视图显示画面规则和所生成对象之间的关系。借助生成概览，可以规划和组态附加生成的后续更改。



生成概览的组成部分包括：

“画面/画面项”(Screens/ Screen Items) 选项卡	**“变量”(Tags) 选项卡**	**“文本列表”(Text lists) 选项卡**
画面/画面对象的名称 对象的唯一名称	名称 生成变量表/生成变量的名称	文本列表/文本列表条目 文本列表及其文本列表条目的名称
主副本/类型 对象生成模板的名称	数据类型 生成变量的数据类型针对“UDT”数据类型（PLC 数据类型）显示 UDT 数据类型的名称。	主副本/类型 文本列表生成模板的名称
HMI 设备 为其生成了对象的 HMI 设备的名称	HMI 设备 为其生成了外部变量的 HMI 设备的名称	HMI 设备 为其生成了文本列表的 HMI 设备的名称
PLC 设备 为其生成了对象的 PLC 的名称	PLC 设备 为其生成了变量的控制器的名称	PLC 设备 为其生成了文本列表的控制器的名称
程序块 为其生成了对象的 FB 或 FC	程序块 为其生成了变量的 DB	文本 包含文本列表条目的文本
画面规则 定义了对象的生成的画面规则	PLC 变量 为其生成了外部变量的 PLC 变量的名称。	规则名称 指定文本列表生成的文本列表规则的名称
日期 为其生成了对象的时间戳。	变量表 在其中生成了变量的变量表的名称	程序段 生成过程中评估的程序段的名称
通过矩阵生成 已使用生成矩阵在下游生成过程中创建了对象。	变量文件夹 在其中生成变量表和变量的项目树中的文件夹的名称	程序块 为其生成了文本列表的 FB 或 FC

“画面/画面项”(Screens/ Screen Items) 选项卡	“变量”(Tags) 选项卡	“文本列表”(Text lists) 选项卡
布局字段 如果在布局字段中生成对象，那么该字段的名称将显示在此处。	变量规则 指定生成变量的存储结构的变量规则	调用结构 周期 OB“Main1”中的调用路径，指定文本列表的生成
调用结构 用户程序 (OB1) 中调用层级中的评估块的路径	---	---

弹出窗口画面/画面项、文本列表、报警具有类似的选项。

5.3.10 Energy Suite 生成

简介

SiVArc 将其生成支持扩展到 Energy Suite 等客户端。Energy Suite 根据组态的画面规则生成画面和画面项，这些规则称为系统画面规则。可以通过站对话框“选择用于 SiVArc 生成的站和控制器”(Select station and controllers for SiVArc generation)，选择基于 SiVArc 画面规则和系统画面规则进行生成。在“选择用于 SiVArc 生成的站和控制器”(Select station and controllers for SiVArc generation) 站对话框中，可在“规则集”(Rule Set) 列选择要生成的规则类型，可以是以下规则类型之一：

- 用户创建的规则 - SiVArc 中用户创建的规则
- Energy Suite 规则 - 客户端 Energy suite 创建的规则
- 所有规则 - SiVArc 和 Energy Suite 规则的组合。

说明

选择“所有规则”(All Rules) 后，生成时 SiVArc 创建的规则优先于 Energy Suite 规则。

- 如果未安装 Energy Suite，则在 SiVArc 生成期间，规则集下拉列表将仅以只读模式列出“用户创建规则”(User created rules)。
- 如果 Energy Suite 规则在无有效 Energy Suite 许可证的情况下可用，并选择在规则集设置为“所有规则”(All rules) 时执行 SiVArc 生成，则将会显示缺少许可证错误。

SiVArc 画面规则和系统画面规则的生成方案

用户创建的规则	系统画面规则	结果
在 SiVArc 中为画面 1 组态的规则	由 Energy Suite 为画面 1 组态的规则	系统生成画面 1
在 SiVArc 中使用按钮 1 为画面 1 组态的规则	由 Energy Suite 使用按钮 2 为画面 1 组态的规则	系统在 Screen1 上生成按钮 1 和按钮 2
在 SiVArc 中使用按钮 1 为画面 1 组态的规则	由 Energy Suite 使用按钮 1 为画面 1 组态的规则	系统在 Screen1 上生成按钮 1
用户使用按钮 1 创建的画面 1	由 Energy Suite 使用按钮 1 为画面 1 组态的规则	系统将画面 1 重命名为画面 1_重命名 系统生成画面 1 和按钮 1
用户使用按钮 1 创建的画面 1	由 Energy Suite 使用按钮 2 为画面 1 组态的规则	系统使用按钮 1 将画面 1 重命名为画面 1_重命名 系统生成由 Energy Suite 规则创建的画面 1，并将按钮 2 放置在画面 1 上
用户使用按钮 1 创建的画面 1 在 SiVArc 中使用按钮 2 为画面 1 组态的规则	由 Energy Suite 使用按钮 3 为画面 1 组态的规则	系统使用按钮 1 将画面 1 重命名为画面 1_重命名 系统基于规则选择在画面 1 上生成按钮 2 和/或按钮 3
复制在 SiVArc 中为画面 1 组态的规则	由 Energy Suite 为画面 1 组态的规则	系统生成画面 1，并将画面 1 重命名为 Screen1_重命名 系统通过 Energy Suite 规则生成画面 1
复制在 SiVArc 中为画面 1 组态的规则（使用此规则第二次生成）	由 Energy Suite 为画面 1 组态的规则（使用此规则第一次生成）	第一次生成时，系统生成画面 1 生成时不考虑复制规则，并且由于错误而被忽略

参见

生成概览 (页 44)

“变量规则”编辑器 (页 31)

5.4 HMI 编辑器中的 SiVArc

5.4.1 属性组态器

说明

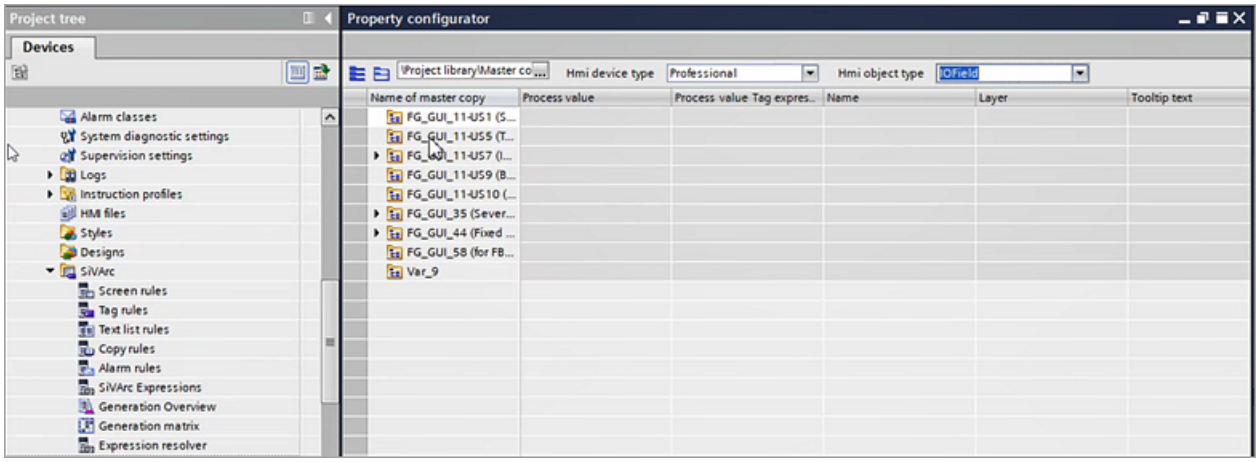
通过属性组态器，可以组态“**模版副本**”(Master copies) 文件夹下库对象的 SiVArc 属性。属性组态器提供了另一种选择库对象的方式，通过这种方式可以查看和组态特定 HMI 设备类型的属性。可以选择在“模版副本”(Master copies) 文件夹中浏览不同的文件夹层级，组态器可在其中显示所选对象和 HMI 设备类型的可组态属性。通过“属性组态器”(Property configurator) 编辑器，可以执行其它编辑器支持的必要任务。

SiVArc 插件编辑器用来反映“属性组态器”(Property configurator) 中的对象。“属性组态器”(Property configurator) 编辑器提供下述选项：

- “选择库文件夹”(Select library folder) - 可用于浏览模版副本的文件夹层级，其中将显示包含所选 HMI 设备类型和 HMI 对象类型的模版副本对象的文件夹
- “HMI 设备类型”(HMI device type) - 下拉列表列出了受支持的 HMI 设备类型
- “HMI 对象类型”(HMI object type) - 下拉列表列出了受支持的 HMI 对象类型

下面介绍组态库对象（例如按钮）的 SiVArc 属性的示例。

1. 在属性组态器编辑器中，浏览“项目库\模版副本”(Project library\Master copies) 中的文件夹，可从中查看/组态包含库对象的文件夹中的 SiVArc 属性。如果想查看/组态库对象的 SiVArc 属性，也可以选择根节点。以下屏幕截图显示了“项目库\模版副本”(Project library\Master copies) 中的文件夹选择：

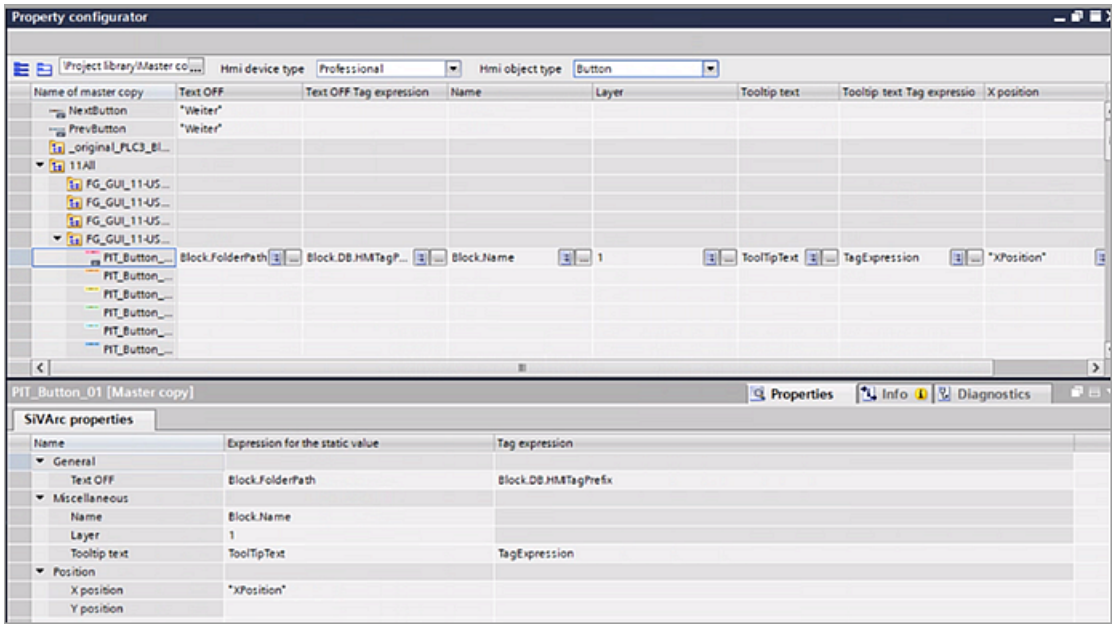


2. 从“HMI 设备类型”(HMI device type) 下拉框中将 HMI 设备选为“专业”(Professional)。WinCC Unified 设备类型将显示在 Unified 环境中。
3. 如以下屏幕截图所示，从“HMI 对象类型”(HMI object type) 下拉框中选择“按钮”(Button) 对象。属性组态器编辑器将显示包含“按钮”(Button) 对象的“模版副本”(Master copies) 文件夹。

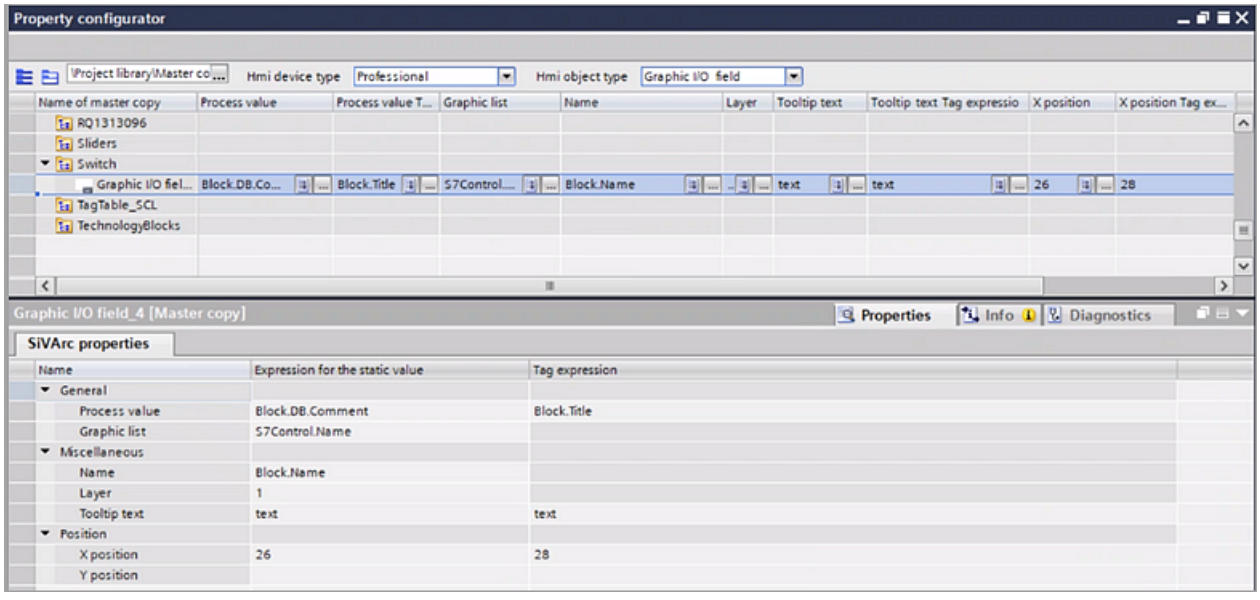
说明

“HMI 对象类型”(HMI object type) 列取决于在“HMI 设备类型”(HMI device type) 列中选择的 HMI 设备类型。

4. “属性组态器”(Property configurator) 编辑器中的列显示可以使用 SiVArc 表达式为“按钮” (Button) 组态的 SiVArc 属性，如以下屏幕截图所示：



5. 如果选择了“HMI 对象类型”(HMI object type) 作为图形 IO 域，则“属性组态器”(Property configurator) 编辑器将显示 SiVArc 支持的列。但是，SiVArc 插件编辑器将显示在工程组态系统上启用的属性。注意“属性组态器”(Property configurator) 编辑器中显示的属性，以及 SiVArc 插件编辑器中的属性，如以下屏幕截图所示。“属性组态器”(Property configurator) 显示 SiVArc 支持的属性，而 SiVArc 插件编辑器显示在工程组态系统上启用的属性。



重要事项：

- 对“模板副本”(Master copies) 文件夹中的库对象属性执行的任何编辑/删除任务都将反映在“属性组态器”(Property configurator) 编辑器中，反之亦然。
- 对“模板副本”(Master copies) 文件夹中的库对象进行重命名或重新分配将反映在“属性组态器”(Property configurator) 编辑器中。

参见

“SiVArc 表达式”编辑器 (页 35)

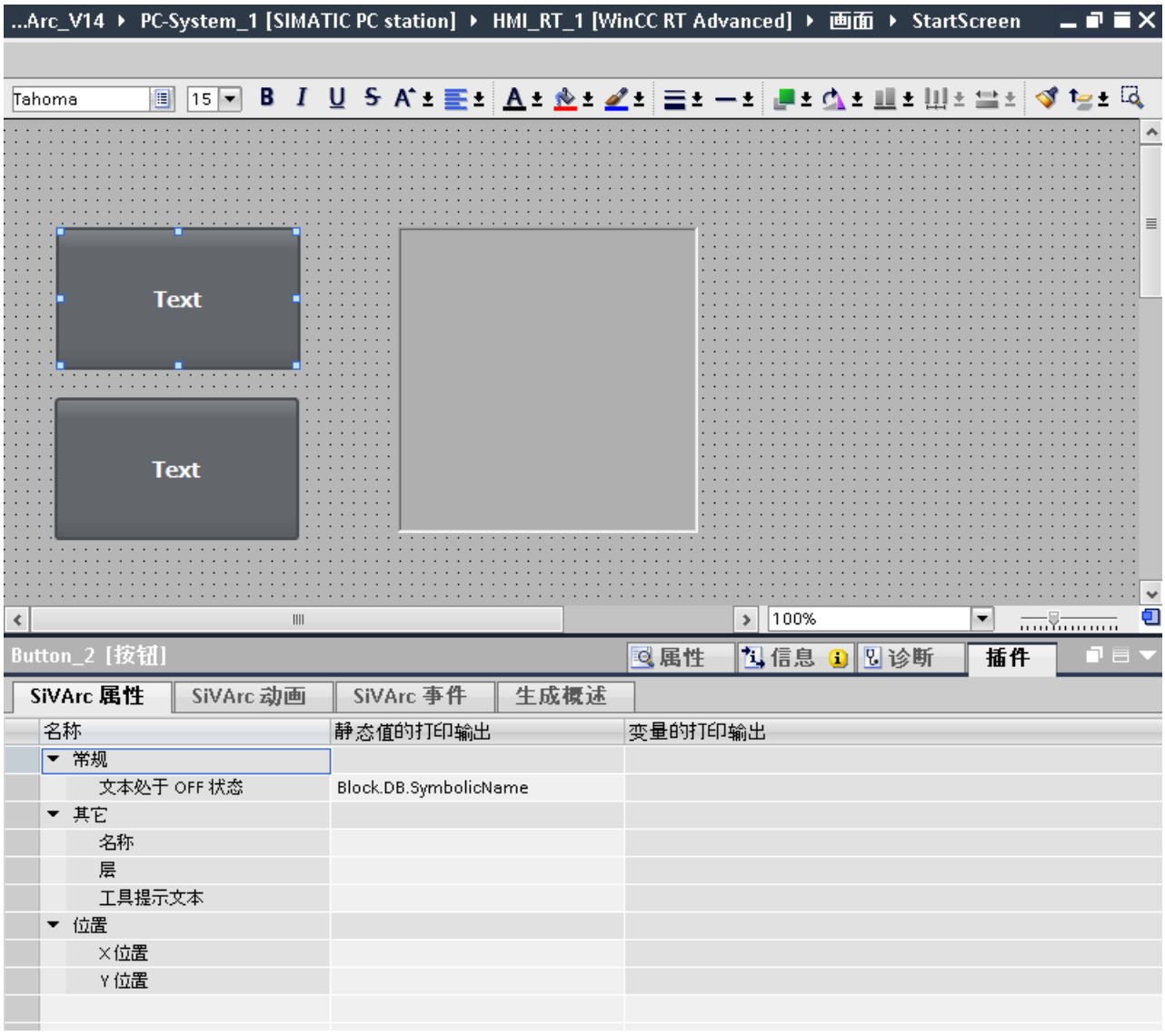
5.4.2 “SiVArc 属性”选项卡

说明

SiVArc 属性是一种可通过 SiVArc 表达式以静态或动态方式组态的对象属性。

在“SiVArc 属性”(SiVArc properties) 选项卡中，可通过 SiVArc 表达式组态文本列表、画面或画面对象的属性。随后可将组态的对象存储到项目库中。SiVArc 表达式在可视化生成期间求值。可以为 Unified 设备中的画面和画面对象组态颜色属性。将颜色属性的值组态为 RGB 格式的值“0-255, 0-255, 0-255”，例如“120, 120, 120”

“SiVArc 属性”(SiVArc properties) 选项卡只适用于 SiVArc 支持的对象。


...Arc_V14 ▸ PC-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC RT Advanced] ▸ 画面 ▸ StartScreen
Tahoma
15
Text
Text
100%
Button_2 [按钮]
属性
信息
诊断
插件
SiVArc 属性
SiVArc 动画
SiVArc 事件
生成概述
名称
静态值的打印输出
变量的打印输出
常规
文本处于 OFF 状态
Block.DB.SymbolicName
其它
名称
层
工具提示文本
位置
X位置
Y位置

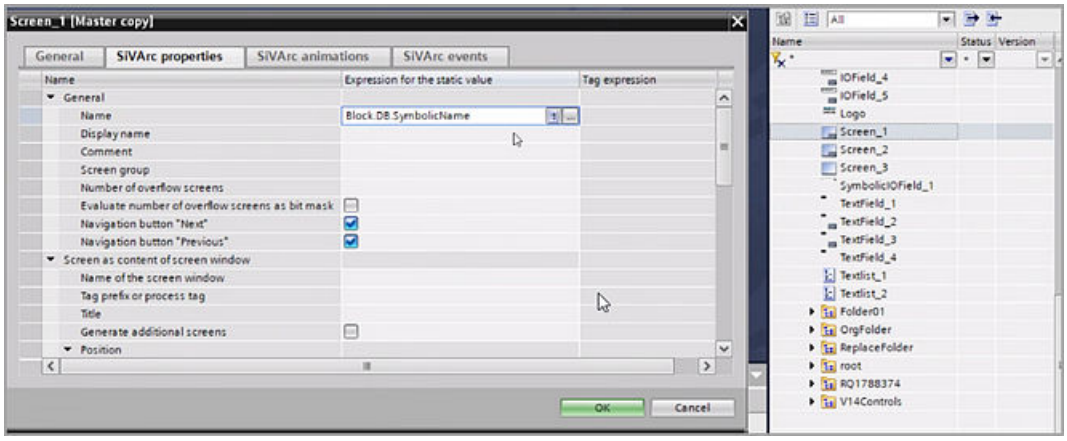
组态溢出画面时，可以在 SiVArc 插件编辑器中单独组态“上一个”(Previous) 和“下一个”(Next) 导航按钮。项目升级期间：

- 如果在旧版本中选择了“生成导航按钮”(Generate navigation button) 选项，则画面主副本将更新为最新版本，如下所示：
  - 如果在旧版本中选择了“生成导航按钮”(Generate navigation button) 选项，则将在最新版本中自动选择“下一个”和“上一个”按钮。
  - 如果在旧版本中取消选择了“生成导航按钮”(Generate navigation button) 选项，则将在最新版本中取消选择“下一个”和“上一个”按钮。
- 旧版本的 HMI 设备中存在的画面将包含所选的导航按钮“上一个”(Previous) 和导航按钮“下一个”(Next)。
- 旧版本的库类型中存在的画面将包含所选的导航按钮“上一个”(Previous) 和导航按钮“下一个”(Next)。

通过主副本访问 SiVArc 属性

作为通过项目文件夹进行导航的替代方式，可通过“项目库 > 主副本”(Project library >Master copies) 文件夹查看/编辑对象的属性。请执行以下操作：

1. 右键单击要选择编辑的任何对象，然后选择“属性”(Properties) 选项。
2. 将出现 SiVArc 属性弹出窗口。利用通过主副本访问 SiVArc 属性的方式，可通过本地表达式和全局表达式组态属性。
3. 更新对象的 SiVArc 属性后，必须保存更改。



说明

屏幕对象的多语言支持并不适用于所有设备（WinCC Unified 设备除外）。

布局

选项卡包括三列：

- 名称
  此列会列出可用属性。
- 静态值的表达式
  在此列中，分配具有固定值的属性或返回字符串或数字的 SiVArc 表达式。
  生成可视化时，该主副本的每个实例中会输入固定值。注意，例如当对象名称在一个画面中多次使用时，利用“名称”(Name) 属性可确保对象名称的唯一性。
- 变量的表达式
  在此列中，可分配带有变量名称或能返回变量名称的 SiVArc 表达式的属性。

5.4.3 “SiVArc 事件”选项卡

说明

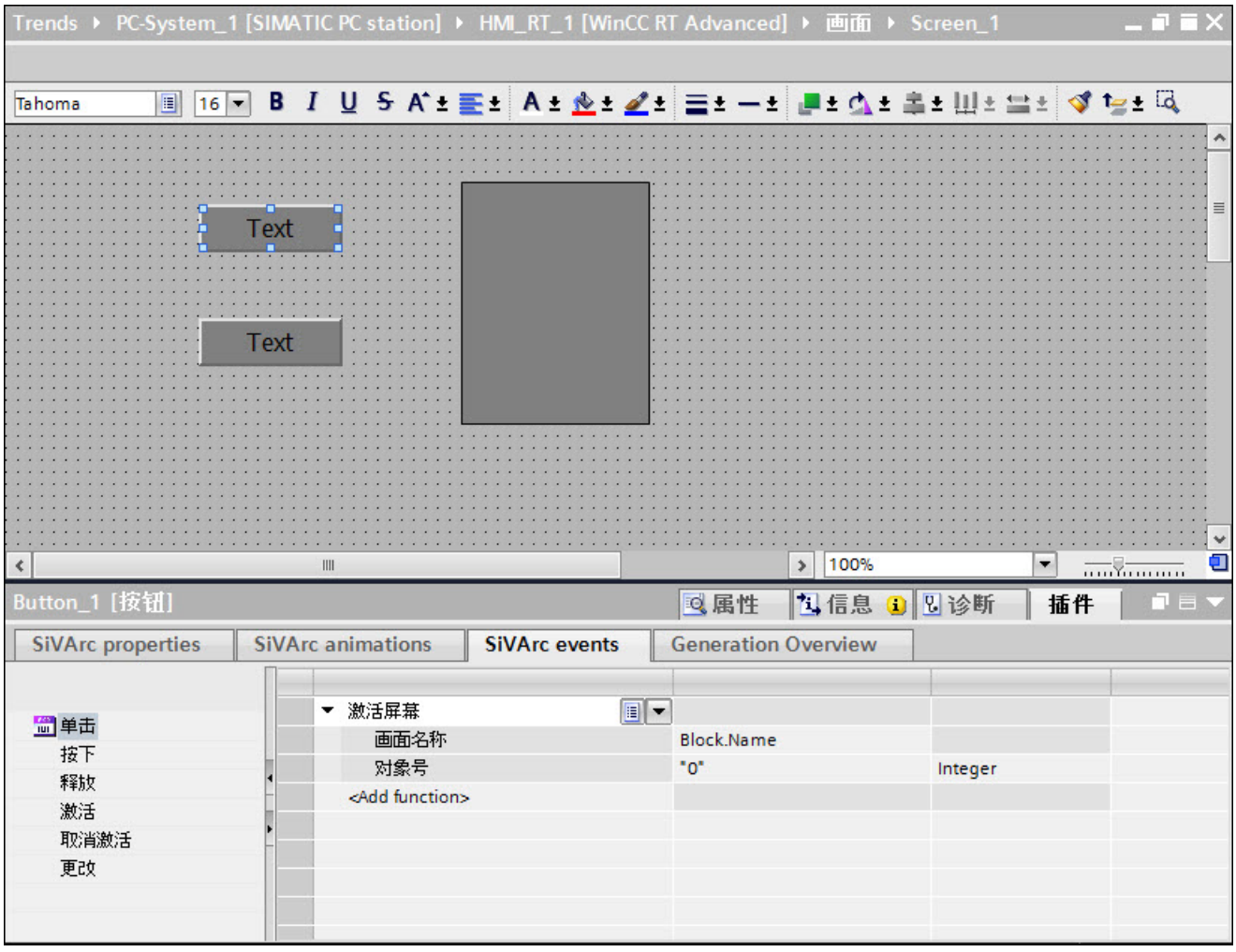
情形 1：在“SiVArc 事件”(SiVArc events) 选项卡中，可为画面或画面对象的生成模板事件组态一个函数列表。可以为画面和画面对象组态事件。可在函数列表中添加系统函数或脚本函数。

通过 SiVArc 表达式，可以组态系统函数或脚本的参数。组态适用于 SiVArc 生成的画面规则。生成后，将解析在 SiVArc 事件下设置的表达式，并将其显示在画面事件窗口中的 SiVArc 插件下。

情形 2：组态了事件的画面可放置于库类型下，并在画面规则中用作画面对象。生成 SiVArc 后，将在 PNV 节点的“**画面**”(Screens) 下生成画面。

对于 WinC Comfort Unified/Unified SCADA RT 设备：

- 支持基于 Java 脚本的全局模块。
- 通过更改画面系统函数组态激活画面事件。



布局

列 1：在列 1 中选择函数或脚本。

列 2：在列 2 中输入 SiVArc 表达式。

列 3：选择脚本后，在列 3 中选择数据类型。

库对象

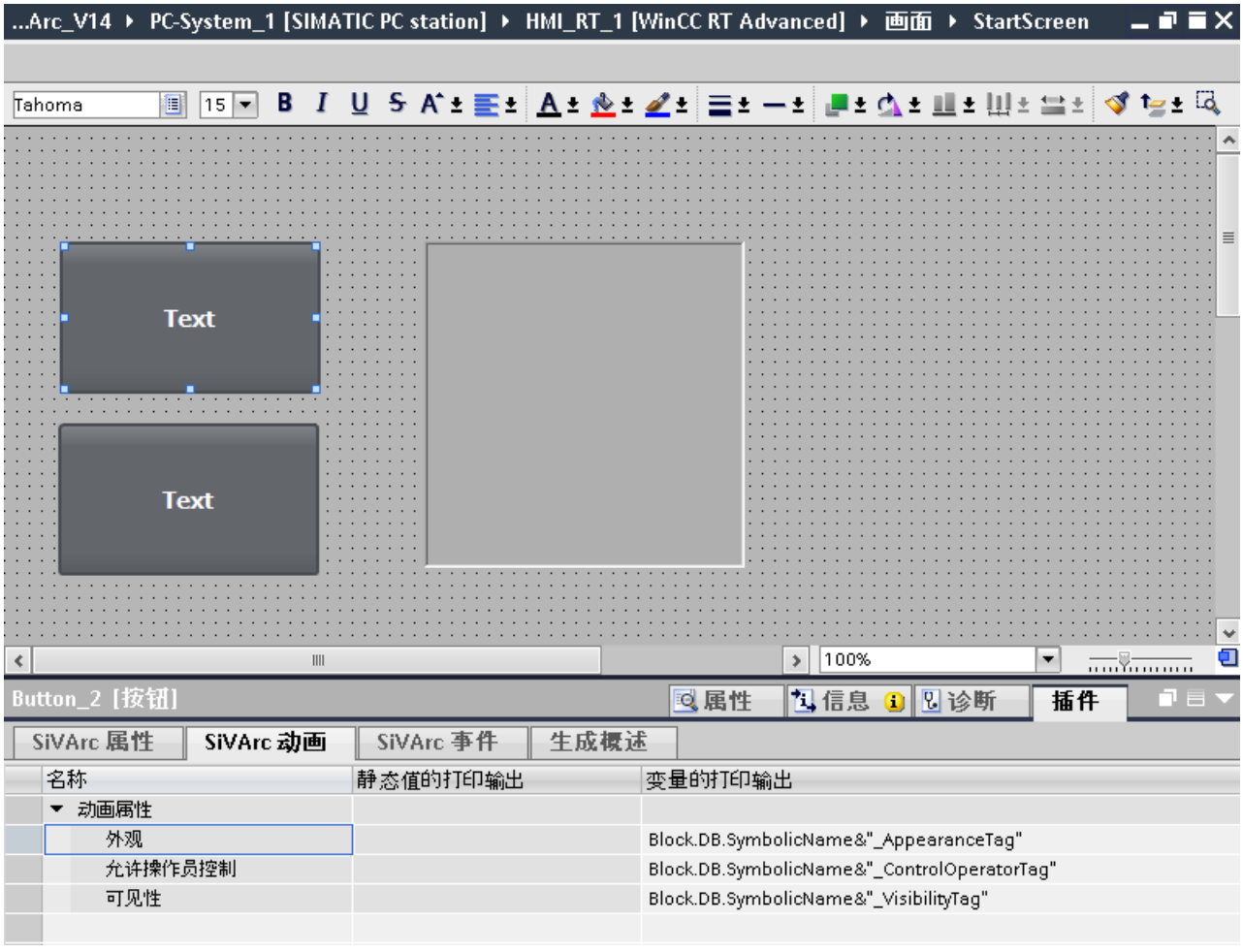
可以通过打开属性对话框为库元素组态事件，SiVArc 事件选项卡可用于查看/编辑选项。

5.4.4 “SiVArc 动画”选项卡

说明

“SiVArc 动画”(SiVArc animations) 选项卡中列出了在画面对象上组态的动画。

“SiVArc 动画”(SiVArc animations) 选项卡只适用于 SiVArc 支持的 HMI 对象。



可以通过右键单击某个对象，为支持的库对象组态动画，SiVArc 插件编辑器可用于查看/编辑选项。

布局

"SiVArc 动画"(SiVArc animations) 选项卡包含以下列：

- 名称
  此列中列出了"属性 > 动画"(Properties > Animations) 下组态的动画。
- 静态值的表达式
  无法针对动画编辑此列。
- 变量的表达式
  在此列中，可通过 SiVArc 表达式组态用于动画的过程变量。SiVArc 表达式必须返回一个变量名称。

库对象

可以通过打开属性对话框为库元素组态动画，SiVArc 动画选项卡可用于查看/编辑选项。

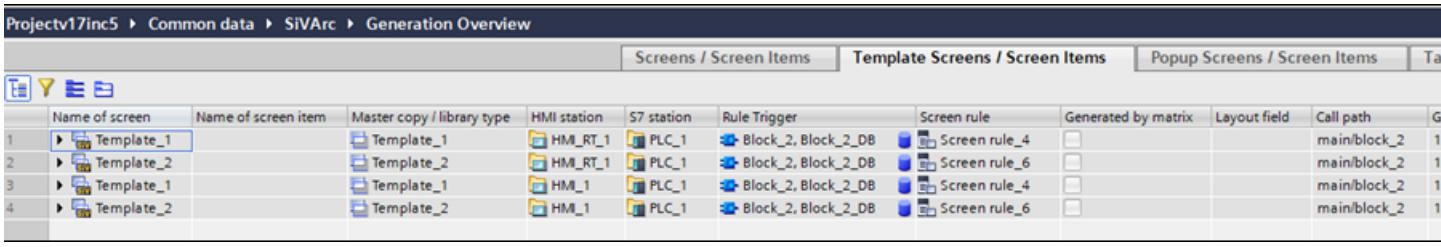
5.4.5 "生成概览"选项卡

说明

首次生成后，"生成概览"(Generation overview) 选项卡会显示在所生成画面的巡视窗口中。所显示对象数受限于显示屏以及所选画面中生成的操作对象。

对于以下例外情况，"生成概览"(Generation overview) 选项卡所包含的编辑选项与"生成概览"(Generation overview) SiVArc 编辑器的相关选项相同：

- 过滤功能
- 排序功能
- "全部打开"(Open all) 和"全部展开"(Expand all) 按钮



"模板画面/画面项目和弹出画面/画面项目"(Template Screens / Screen Items and Popup Screens / Screen Items) 选项卡分别显示模板/画面项目和弹出屏幕/屏幕项目的已生成对象的详细信息和相关规则。

对于生成的画面对象，"插件"(Plug-ins) 选项卡将显示生成概述。

5.5 PLC 编辑器中的 SiVArc

5.5.1 软件单元支持

SiVArc 中的软件单元

软件单元充当分隔各种用户定义程序的容器。可利用 SiVArc 组态 HMI 对象各种属性的表达式，其表现为 SiVArc 规则创建中所包含的 PLC 块的软件单元名称。有关软件单元的更多信息，请参见 *TIA portal 帮助*。

要求

- 存在 TIA Portal 项目
- PLC 固件版本为 v2.6 及以上版本
- PLC 已连接到 HMI 设备

步骤

要组态软件单元的 SiVArc 表达式，请按照以下步骤操作：

1. 在软件单元文件夹下，单击“添加新软件单元”(Add new software unit) 以添加新的软件单元。
2. 根据需要组态程序块。
3. 添加画面，并使用解析为软件单元名称的表达式组态画面名称。例如：名称 = “SoftwareUnit.Name”。有关组态画面的详细信息，请参见“SiVArc 编辑器 (页 29)”。
4. 组态包含从软件单元中选择的程序块的 SiVArc 画面规则。
5. 开始 SiVArc 生成。生成的画面名称解析为包含在 SiVArc 画面规则创建中使用的程序块的软件单元名称。

说明

可以使用 SiVArc 插件编辑器为所有 HMI 对象组态软件单元支持。

5.5.2 SiVArc 支持使用结构化控制语言的程序块

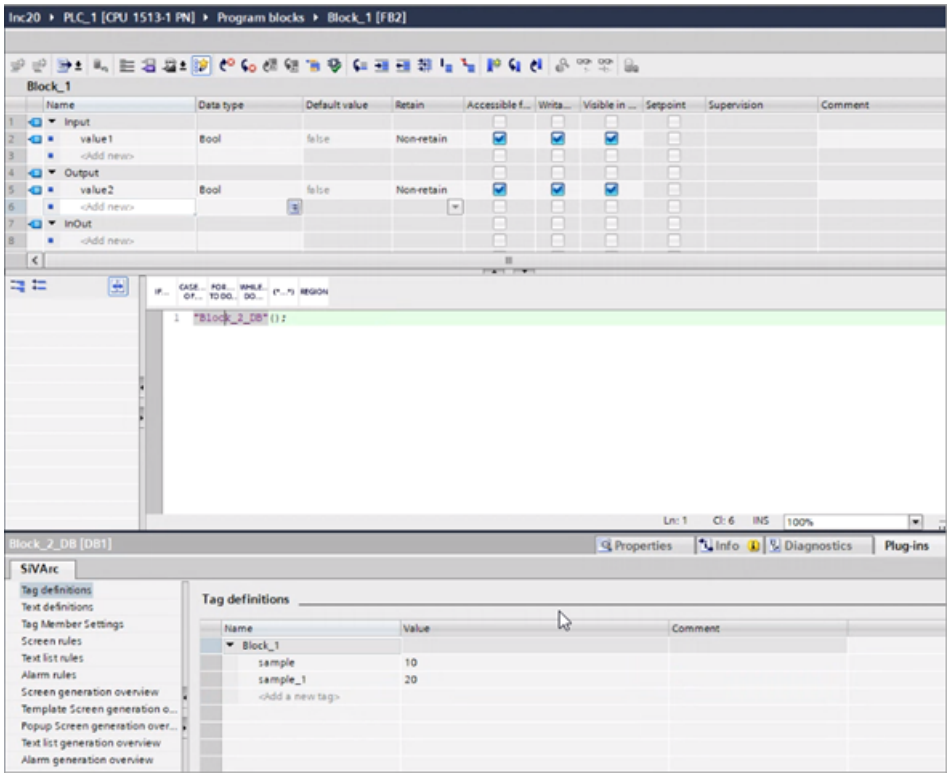
SiVArc 支持使用 SCL 的程序块

对于通过单实例或多实例调用的程序块，SiVArc 可利用 STL、FBD 和 LAD 等编程语言支持 SCL。可以在“插件”(Plug-ins) 编辑器中组态 SCL 块的 SiVArc 属性，如文本/变量定义/变量成员设置/规则编辑器。SiVArc 仅支持 SCL 的块层级组态。可以按块层级组态 SiVArc 数据，这适用于 SCL 块中的所有实例调用

使用 SCL 进行变量定义的示例情况

以组态 SCL 块的变量定义为例。可以组态调用程序块 (SCL) 的变量定义。变量定义适用于 SCL 块中的所有实例调用。定义的变量在 SiVArc 生成期间使用。

1. 例如，使用实例 DB“Block_2_DB”在 Block_1 中调用“Block_2”。
2. 主轴选择 SCL 编程编辑器中的任意区域。单击 SiVArc 插件。
3. 定义并组态“Block_1”的变量，如下图所示：



4. 已定义的变量可用于任何 SiVArc 表达式编辑器，如画面对象的 SiVArc 属性，规则编辑器中的条件。在 SiVArc 插件中组态画面对象时，输入变量定义为“样本”(Sample) 的画面名称。有关组态画面对象的更多详细信息，请参见“画面规则编辑器”部分。
5. 在 SiVArc 生成期间，将生成名为“10”的画面。有关生成的更多详细信息，请参见“生成可视化 (页 209)”部分。

说明

- 对 SCL 块的变量/文本定义/变量成员设置的组态与 LAD 块类似。有关块层级组态的更多详细信息，请参见 TIA portal 用户指南中的 LAD 文件
- 变量成员设置只能在 WinCC Advanced 设备中组态。
- 画面/文本列表/报警规则支持快速访问以便查看和编辑已组态的规则。
- 当选择多实例调用的块时，将显示调用程序块的画面、文本列表和报警规则。
- 画面/文本列表/报警生成概览支持快速访问以查看 SiVArc 生成。

对于 Unified HMI 设备，创建具有用户自定义数据类型/数组数据类型的 PLC 块变量和具有用户自定义数据类型的 PLC 变量时；“SiVArc **生成 > AllHmiTags**”会得出 PLC 块变量的 HMI 变量以及数据类型为用户自定义/数组的 PLC 变量。

参见

生成可视化 (页 209)

使用 SiVArc

6

6.1 规划布局

6.1.1 定位方案

6.1.1.1 所生成对象的定位概述

简介

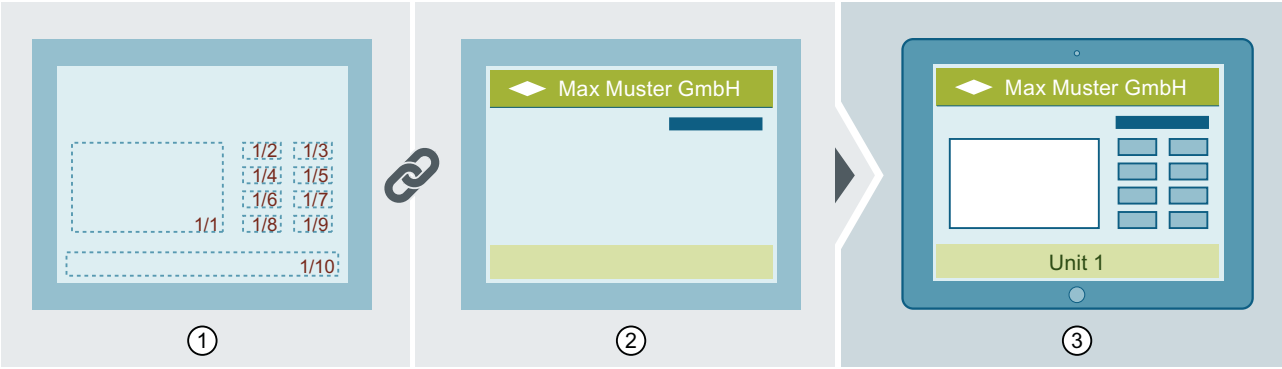
通过 SiVArc，过程画面的布局被分为两个任务：

- 过程画面以及显示和操作对象的图形设计（与 WinCC 中的相关过程一样）。
- 所生成显示和操作对象的定位

SiVArc 支持通过下述方法来控制所生成对象的定位。首次生成后所做的手动位置更改对于所有后续生成都将保留，所有方法均如此。

控制定位

可使用定位方案来控制显示和操作对象在过程画面中的生成位置。为此，将画面的生成模板与定义方案的定位字段相结合。



① 包含所定义画面区域的用户自定义定位方案

② 画面的生成模板

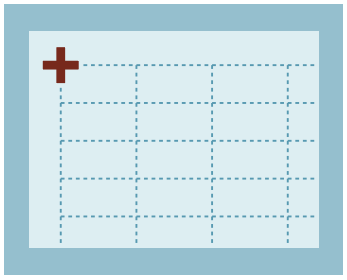
③ 生成的过程画面

自由定位

除了控制定位外，还可以手动定位生成的对象。

为此，针对画面的各个生成模板存储 SiVArc 定位方案。SiVArc 定位方案是可组态的网格。

下图显示了网格的起始位置：



可以设置网格的起始位置以及模板中的行间距和列间距。

完成生成过程后，可根据需要移动对象。在下一次生成时，这些位置也会保留。

另外，可针对单独的对象定义固定位置坐标：

定位方法的优先级

如果在画面规则中为显示和操作对象存储了单独的定位方案，则生成期间将忽略位置的所有其它规范。

如果未保存单独的定位方案，那么生成的显示和操作对象将按固定定位或 SiVArc 定位方案排列。

具有固定位置或定位方案的生成对象将覆盖所组态位置上已存在的显示和操作对象。

SiVArc 将按以下优先级处理各个定位方法：

1. 定位方案
2. 固定位置（SiVArc 属性）
3. 固定位置（WinCC 属性）
4. 自由定位

使用定位方案生成显示和操作对象并将其定位在画面中时，将忽略基于对象组态的固定位置。SiVArc 定位方案也会被忽略。

参见

根据定义的方案进行定位 (页 64)

生成对象的固定位置 (页 73)

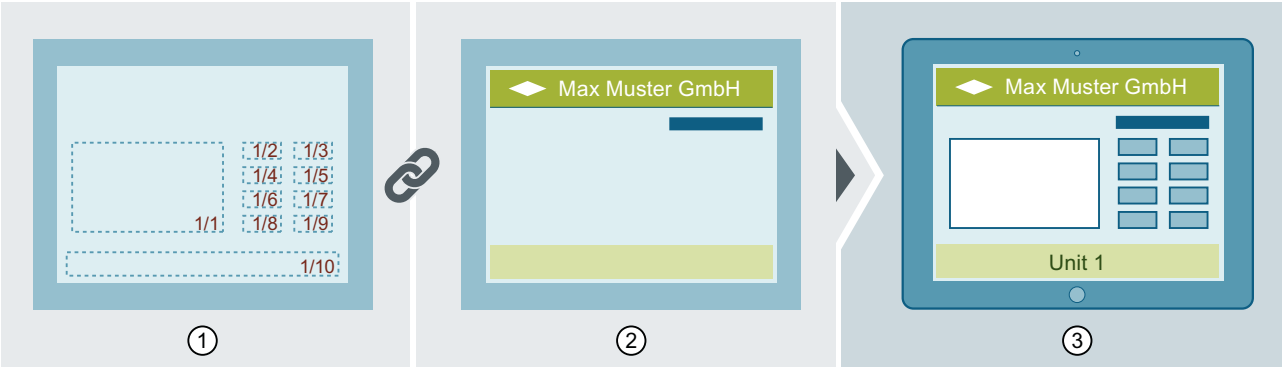
自由定位 (页 74)

设计布局 (页 81)

6.1.1.2 根据定义的方案进行定位

定义

用户自定义定位方案是一种可组态的高像素网格。可将此网格组织到不同的区域。将相应方案分配给画面。随即 SiVArc 会直接在所需区域中生成显示和操作对象。



① 包含所定义画面区域的用户自定义定位方案
② 画面的生成模板
③ 生成的过程画面

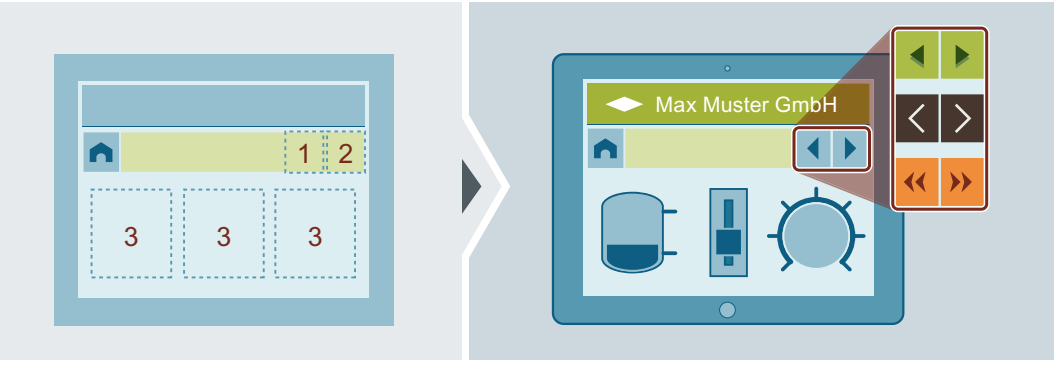
用途

如果项目需要为显示和操作对象提供高像素标准定位，则可以使用库中的用户自定义定位方案。

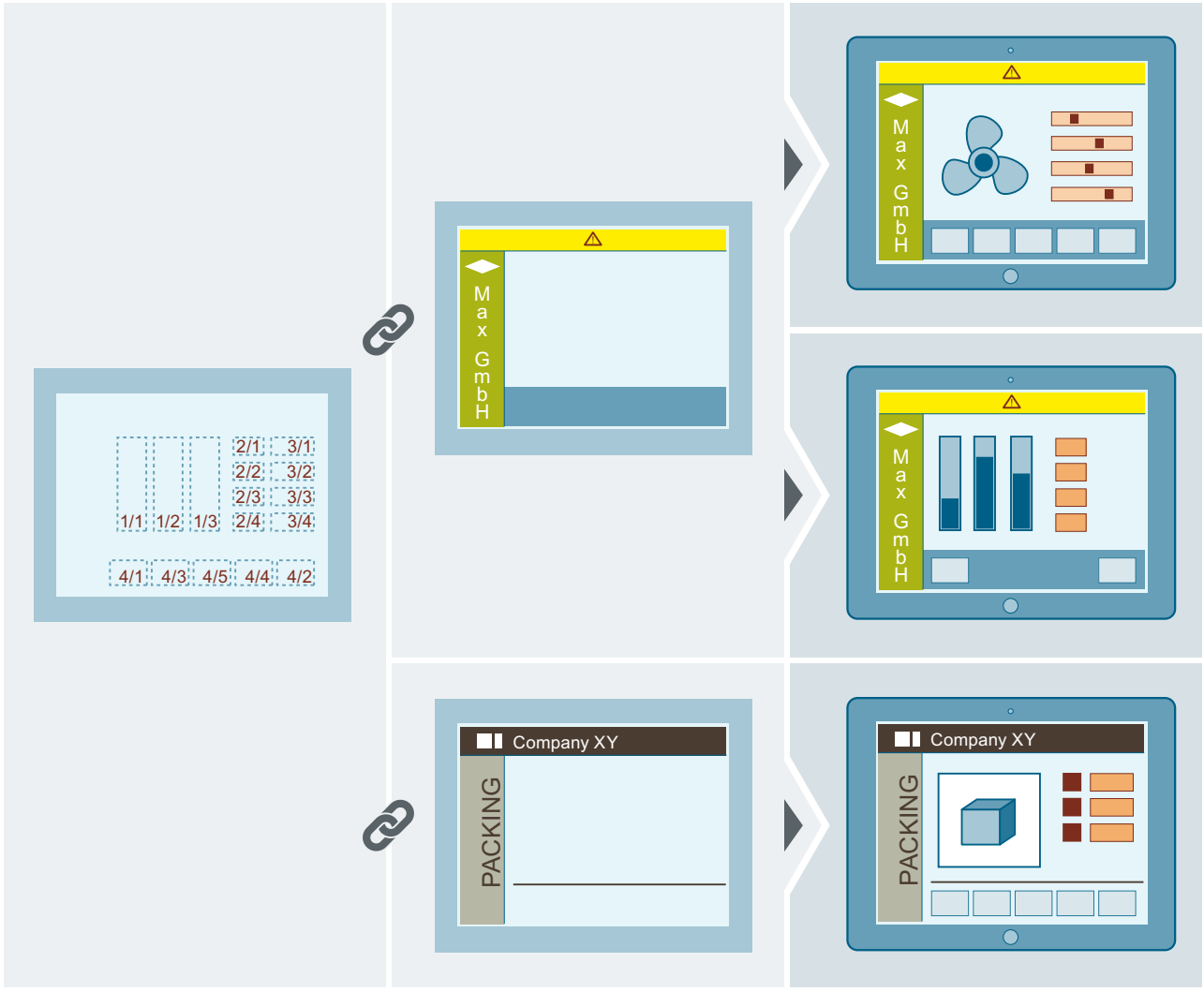
优势

用户自定义定位方案具有多种优势：

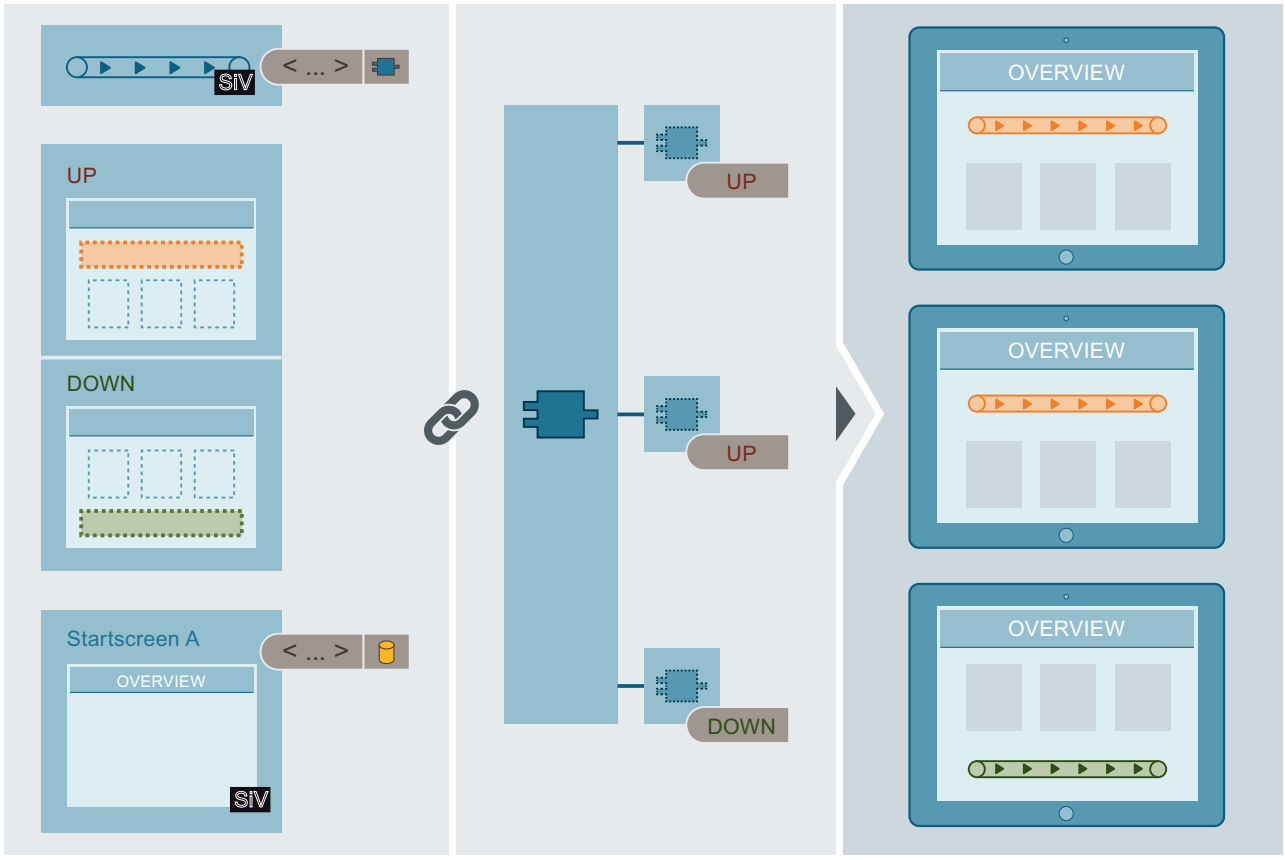
- 项目标准化程度更高
  可使用用户自定义定位方案来控制和管理不同 HMI 设备上所生成对象的排布方式。还可以单独将导航按钮置于方案中的所需位置。



- 布局和定位分离
  由于布局和定位彼此分离，因此还可以跨项目使用现有定位方案。


2/1
3/1
2/2
3/2
2/3
3/3
1/1
1/2
1/3
2/4
3/4
4/1
4/3
4/5
4/4
4/2
Max GmbH
Max GmbH
Max GmbH
Company XY
PACKING
Company XY
PACKING

- 动态定位方案
  例如，可使用用户程序中的条件将定位方案动态分配给画面。这样可以减少 SiVArc 项目中的画面规则数。
  下图显示了传送带图形的画面规则，传送带是位于所生成画面的顶部还是位于底部取决于程序段中的 SiVArc 变量：



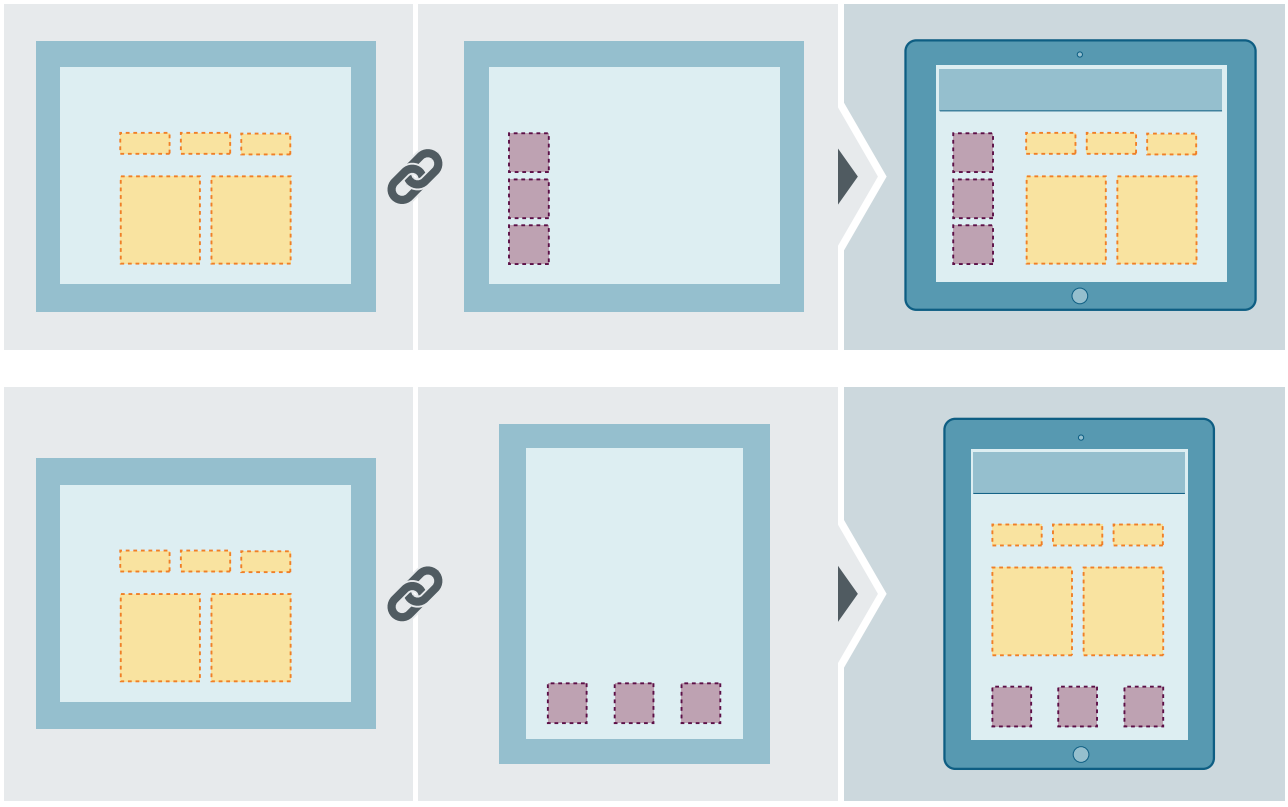
SiV SiVArc 生成模板

< ... > 引用文本源的 SiVArc 属性

函数块

实例化函数块

- 结合的定位方案
  现有定位方案与其它方案相结合。这样，可将一个显示区域划分为多个模块，例如，可针对不同 HMI 设备随机组合成相应版本。
  下图显示了可向其分配其它不同方案的定位方案：



  如果模块所处的较高级别定位方案的名称与较低级别方案的名称相同，则会忽略较低级别的区域。

- 定位预览
- 在首次生成之前即可规划好定位
- 错误敏感度低
- 可集中使用各个布局版本

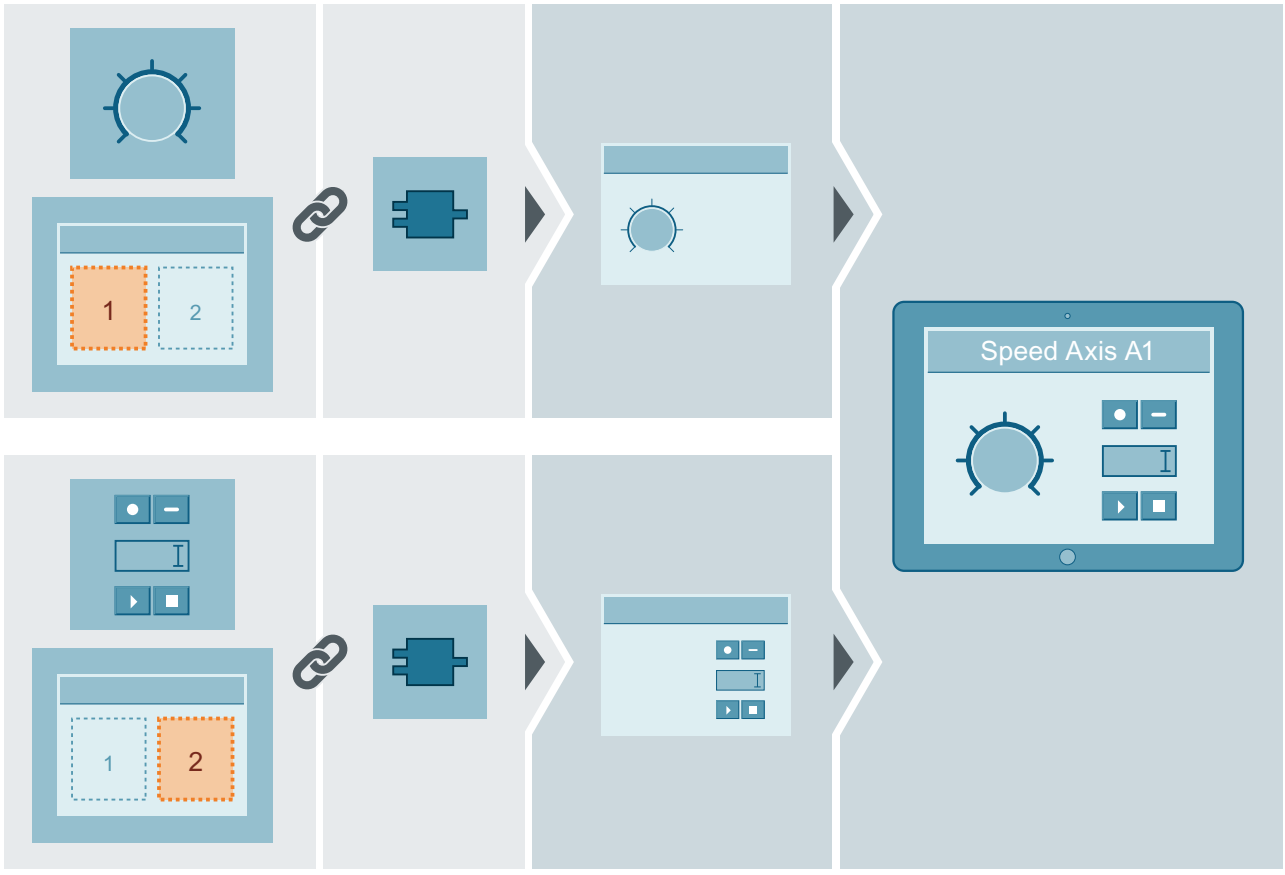
说明

弹出画面的定位方案

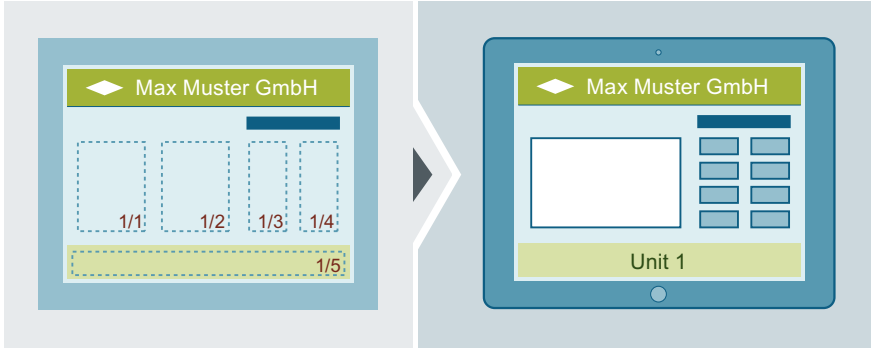
弹出画面的定位方案不能用于任何其它显示和操作对象。

工作原理

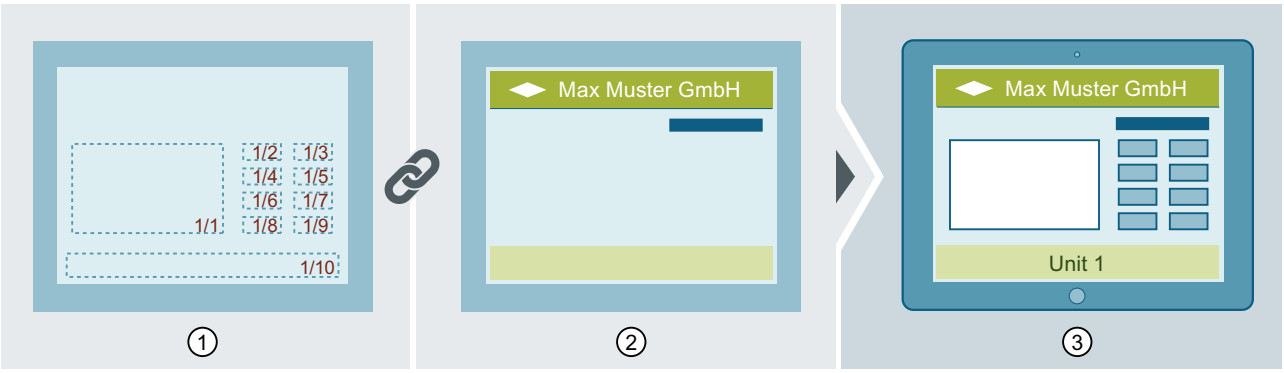
可以将画面规则与显示和操作对象相连，其定义了用于放置对象的定位方案的区域。如果使用组合方案，则可将其中包含的所有布局字段与生成模板进行互连，而无需考虑方案的嵌套深度。



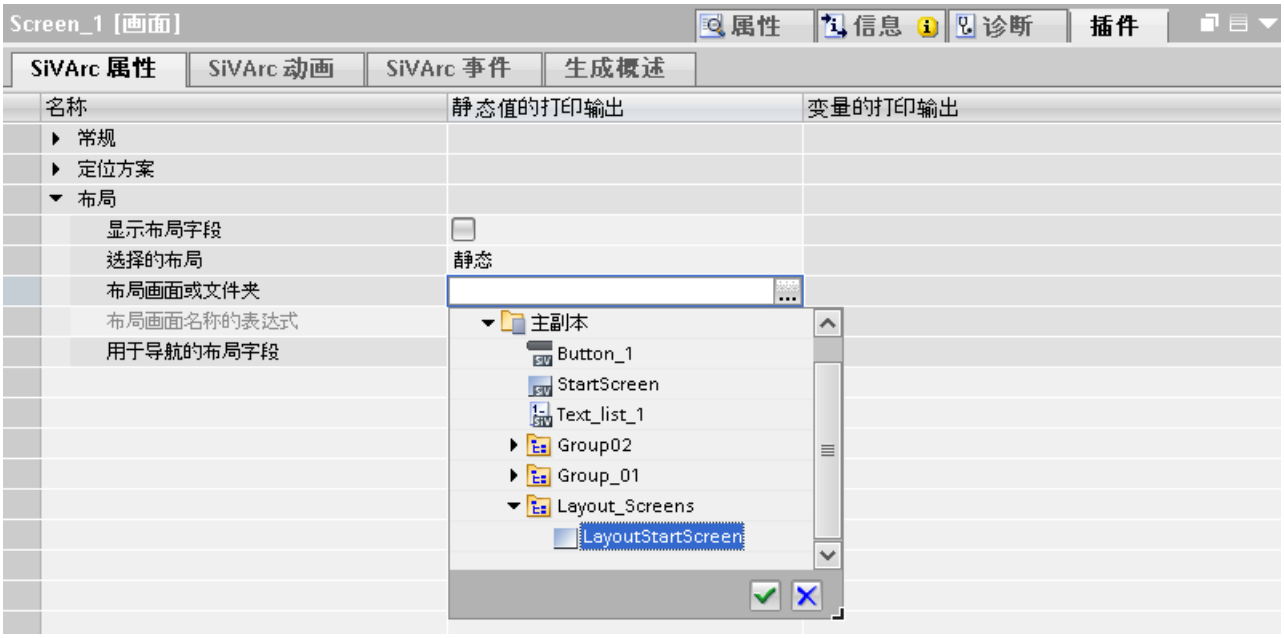
可以在画面规则中使用定位方案，就像使用画面的生成模板一样。之后布局和定位将包含在同一模板中：



为了分离布局和定位，需永久或动态地为画面分配生成模板的定位方案：



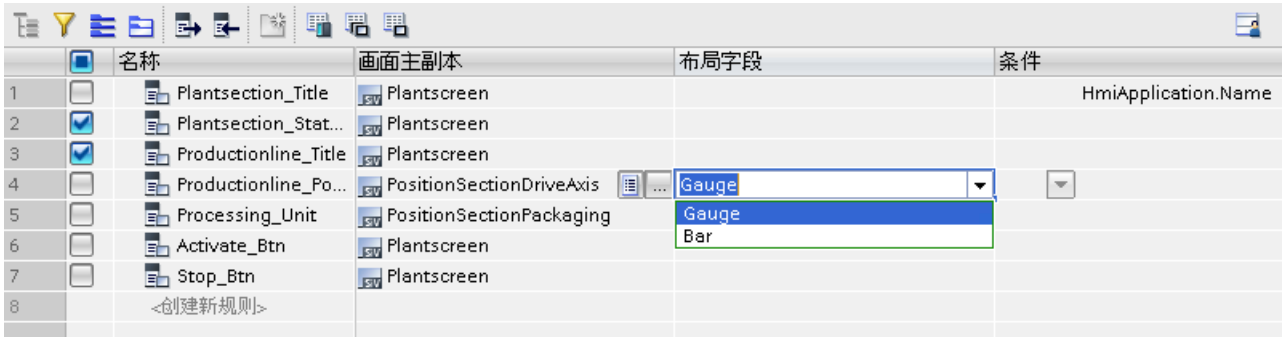
下图显示了画面生成模板 SiVArc 属性中的相关设置：



布局字段

在画面规则中，选择生成对象的定位方案区域。 SiVArc 将该区域中的画面对象生成到索引为 1 的布局字段中。下一个生成的对象会生成到索引为 2 的字段中，依此类推。

下图显示了画面规则中布局字段的组态：



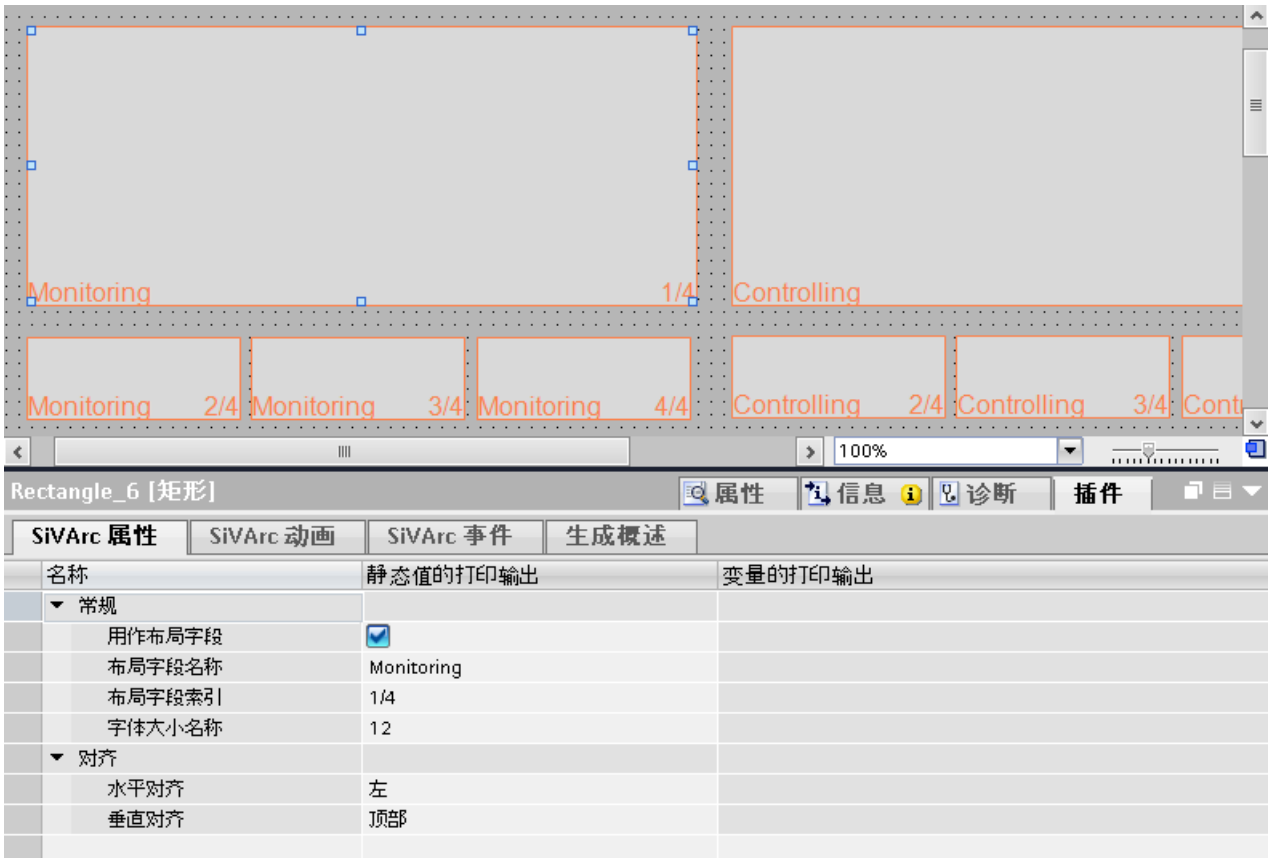
层分配

如果在生成过程中为主副本分配一个固定层并使用自定义定位方案，则 HMI 对象会在定位方案中指定的层中生成。

结构

用户自定义的定位方案包含一个画面，其中含有用于所生成显示和操作对象的布局字段。为生成模板分配定位方案，进而创建一个过程画面。

通过为布局字段分配相同的名称，可将这些布局字段分组到一个逻辑单元。布局字段按逻辑单元内的索引顺序填充。



后续更改

如果手动更改所生成显示和操作对象的位置，则在下次生成期间保留该更改。即使该位置采用自身的定位方案进行定义也是如此。即使更改了定位方案，手动组态的位置在下次生成后也仍会保留。

参见

所生成对象的定位概述 (页 61)

溢出机制 (页 76)

创建用户自定义定位方案 (页 83)

示例：使用动态布局 (页 90)

示例：使用组合布局 (页 93)

示例：使用生成的画面导航 (页 94)

画面图例 (页 287)

6.1.1.3 生成对象的固定位置

如果要始终将特定对象（例如，标准对象）固定在画面的同一位置，请选择一个固定位置。

固定位置取决于画面分辨率。对于显示和操作对象，在高分辨率的 HMI 设备中的显示位置相比于相同大小但分辨率较低的 HMI 设备中的显示位置更偏向左上。

单独定义对象坐标，而与与显示和操作对象生成模板的 SiVArc 属性中的定位方案无关。

说明

不可更改的画面对象固定位置

对于具有固定定位的画面对象，位置的手动更改会在下一个生成过程中被忽略。

组态 X 和 Y 位置属性的表达式编辑器

SiVArc 支持通过可解析为整数值的表达式或直接使用整数值，在 SiVArc “插件” **(Plug-ins)** 编辑器 >“位置”(Position) 中组态静态和动态画面对象的位置。可以在 “位置”(Position) 属性中通过 SiVArc 表达式组态画面对象的 X 和 Y 位置，并将画面对象添加至主副本。可以定义画面规则，该规则包含位于主副本下的对象和要生成的画面。在 SiVArc 生成期间，为 X 和 Y 位置定义的表达式必须解析为整数值，并在该位置生成相应的画面对象。

还可以通过 “变量打印输出”(Printout of tags) 为 X 和 Y 位置组态动态变量值。在 SiVArc 生成期间， “变量打印输出”(Printout of tags) 列下定义的表达式将解析为变量名称，并生成相应的画面对象。有关生成的更多信息，请参见 “所生成对象的定位概述 (页 61)”部分。

参见

所生成对象的定位概述 (页 61)

关于生成的基础知识 (页 209)

6.1.1.4 自由定位

概述

生成过程中，在生成的画面上存储网格以用于排列画面对象。可以组态网格。

初始生成期间，对象在画面上的网格内生成。随后可单独排列生成的对象。新布局将被保留，供后续的生成操作使用。

此方法具有以下优势：

- 不必事先详细规划布局。
- 每次生成后，可进一步调整布局并添加更多的定义。
- 布局随 SiVArc 项目不断改进。

此步骤非常适合单独的小型开发项目。当项目规模变大时，后续编辑要求也会有所增加。

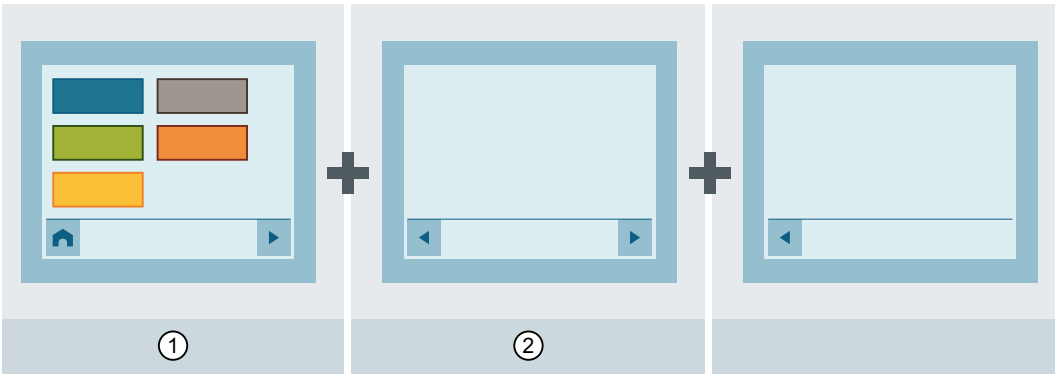
定位方案的结构和填充

可在画面的 SiVArc 属性中组态对象的定位方案。

初次生成后，HMI 对象将根据定位方案进行定位。定位方案基于第一个对象的起始位置和位置 x 和位置 y 指示的距离。

如果未将画面对象分配给溢出画面，则初次生成可视化后，画面对象将按默认方式在基本画面中排列。

下图显示了画面对象在基本对象中的默认排列方式。



① 生成的画面对象将在各画面中从上到下，从左到右逐列定位。画面对象的间距始终相等。

② 如果在画面中生成溢出画面，SiVArc 将自动插入可实现组态的画面切换功能的导航按钮。

参见

生成可视化 (页 219)

所生成对象的定位概述 (页 61)

溢出机制 (页 76)

示例：使用支持自由定位的布局 (页 87)

6.1.1.5 嵌套深度

画面层内的嵌套深度

可通过层级结构在 SiVArc 主副本中设置要生成对象的嵌套深度。该设置在生成过程中保留。

说明

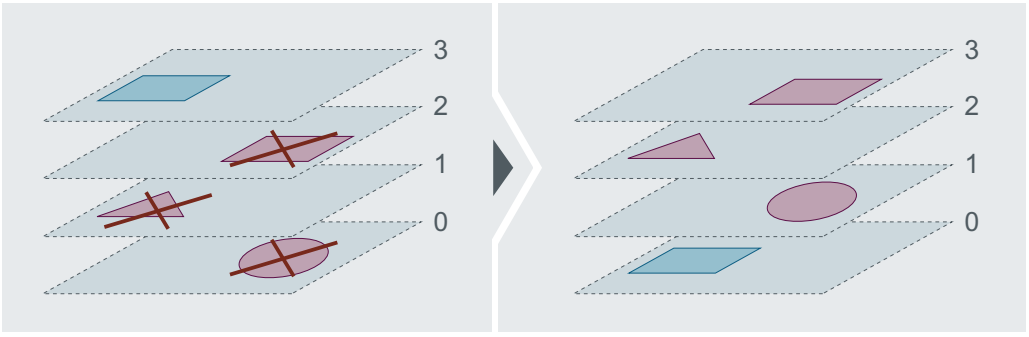
更改层

在生成画面中更改对象的层分配时，此分配将在后续生成过程中保留。

画面层内的嵌套深度

以下内容适用于生成画面中的同一层：

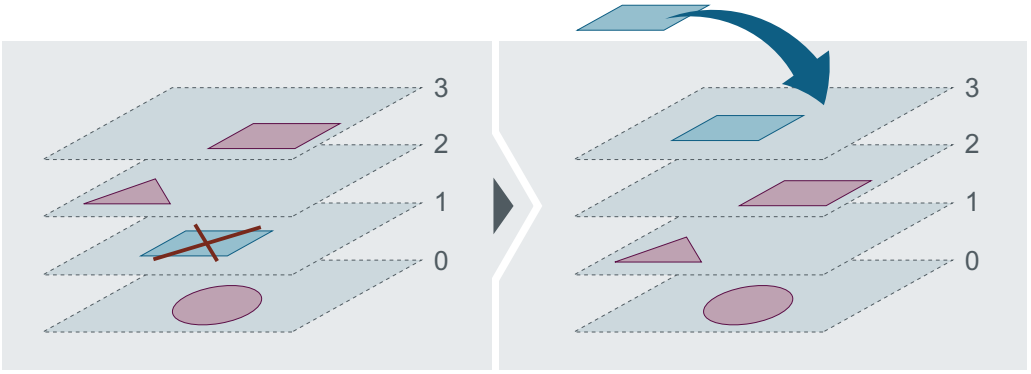
- 删除生成的对象并手动插入对象时，对象在下一次生成过程中的生成嵌套深度也将高于手动插入的对象。



生成的显示和操作对象

手动插入到生成画面中的显示和操作对象

- 如果在生成画面中以指定深度排列手动插入的对象，然后将其删除，那么这种预先的排列与 SiVArc 不相关。下次生成过程中，画面对象将排列在层的最底部位置。如果再次手动插入删除的对象，那么它将位于最高位置。



生成的显示和操作对象

手动插入到生成画面中的显示和操作对象

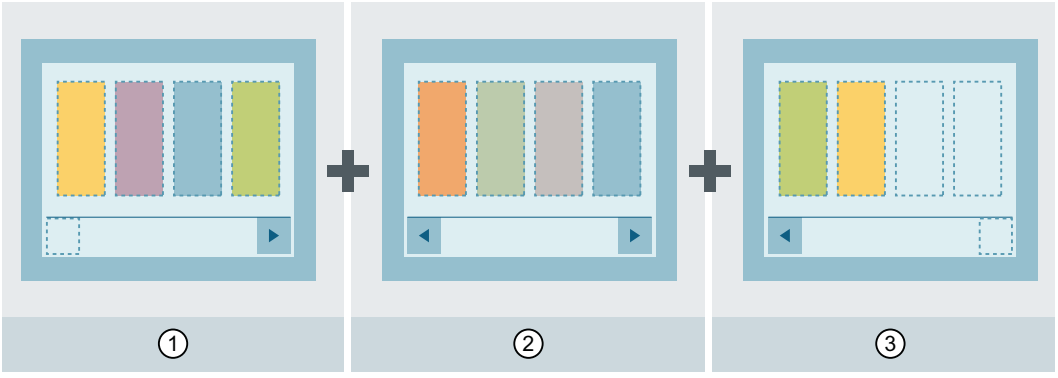
6.1.2 溢出机制

定义

溢出画面即为画面空间不足以用于显示所生成的画面对象数目时生成的画面。根据所使用的定位方案，溢出画面的生成方式有所不同。为生成模板的各个实例生成溢出画面。

基于所定义定位方案的溢出画面

如果组态的布局字段不足以供所有生成的显示和操作元素使用，则会根据定位方案生成溢出画面。



① 基本画面
② 包含基本画面的定位方案的第一个溢出画面
③ 包含基本画面的定位方案的第二个溢出画面

手动填充溢出画面

如果不使用自身的定位方案，并在画面生成模板的 SiVArc 属性中将溢出画面数指定为十进制数，则画面对象仅在基本画面中排列。要限制溢出画面的生成，在“溢出画面数”(Number of overflow screens) 下设定一个条件。禁用选项“按位掩码计算溢出画面数”(Evaluate number of overflow screens as bit mask)。

首次生成后，可在溢出画面中将画面对象移动至所需位置。

之后每次生成生成时都将保留修改后的画面对象位置。

说明

将生成的显示和操作元素复制到溢出画面

将溢出画面数定义为十进制数字时请注意：

如果手动将 SiVArc 生成的对象从基本画面复制到溢出画面，则在重新生成时将保留此更改。副本随后将与基本画面上的 HMI 对象一并处理，处理方式与 SiVArc 生成的对象一样，并且引用 SiVArc。

要求：副本名称必须与原名称匹配。

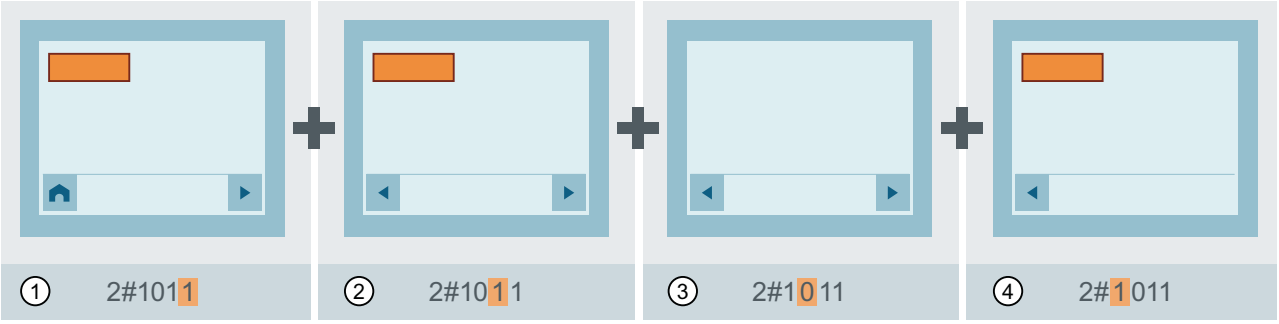
通过位掩码控制溢出画面

可在程序块上或画面生成模板中定义位掩码。为此，请使用静态值或变量。

如果将溢出画面分配指定为位掩码，则画面对象将排列在基本画面和溢出画面中。

需针对位掩码定义以下内容：

- 溢出画面数
  位掩码中的位位数用于定义溢出画面数。位掩码的第一位对应主副本画面。第二位对应于第一个溢出画面，第三位对应于第二个溢出画面，以此类推。位掩码被限制为 31 个溢出画面。使用位掩码 2#0 时不生成溢出画面。
- 带有画面对象的溢出画面
  如果要将所使用画面规则中的画面对象生成到溢出画面，请将位掩码中的对应位设置为 1。
  示例：您正在使用位掩码 2#1011。在生成期间创建了三个溢出画面。生成的所使用画面规则的画面对象如下：



① 带有生成的画面对象的基本画面
② 第一个溢出画面
③ 第二个溢出画面
④ 第三个溢出画面

在画面中组态布局字段后，在生成期间将不再评估画面中所含溢出画面的组态。

说明

将对象复制或移动到溢出画面

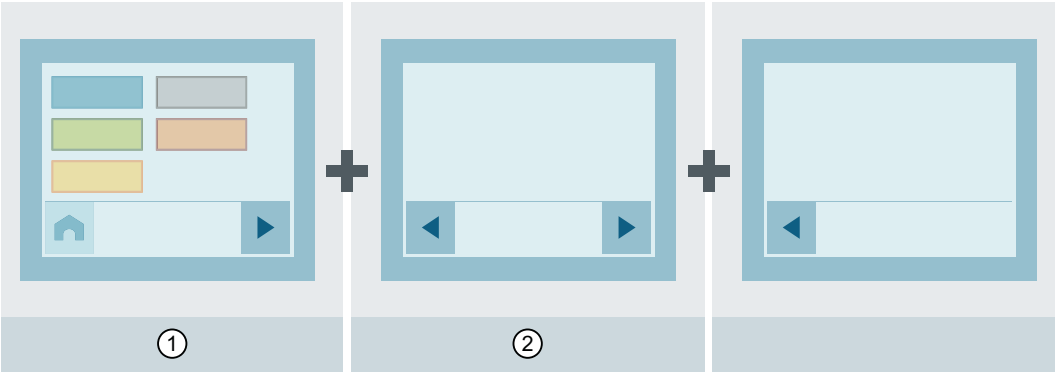
使用位掩码组态溢出画面时请注意：

将生成的显示和操作元素复制或移动到溢出画面或基本画面时，这些对象在新生成过程中将被视为手动创建的对象。下一次生成过程中会再次创建显示和操作元素。

浏览按钮

如果 SiVArc 生成溢出画面，则会生成用于移动到上一个画面和下一个画面的导航按钮。

如果自由定位生成的显示和操作元素，则生成模板的原始画面将作为基本画面生成。基本画面连接到具有导航按钮的第一个溢出画面。



① 生成的主副本画面（基本画面）

② 带有自动生成且可实现组态的画面切换功能的导航按钮的第一个溢出画面

如无需导航按钮，可禁用画面生成模板中的“导航按钮”(Navigation buttons) 选项。

说明

可以在库中存储导航按钮的主副本。

更多相关信息，请参见“SiVArc 中的生成模板 (页 95)”部分。

弹出画面

不会为弹出画面生成溢出画面。当生成的显示和操作元素的数量超出可定位的数量后，系统将输出一则错误消息。不会生成弹出画面上不存在的显示和操作元素。

参见

自由定位 (页 74)

示例：使用溢出机制 (页 88)

组态没有画面对象的溢出画面 (页 86)

6.1.3 支持的设备

概述

SiVArc 可用于以下设备：

- PLC
  - SIMATIC S7-1200
  - SIMATIC S7-1500
  - SIMATIC S7-1500 软件控制器
  - ET 200SP CPU
- 设备代理
  设备代理仅用于生成外部变量。
- HMI 设备
  - 装有 WinCC RT Professional 的 HMI 设备
  - 装有 WinCC RT Advanced 的 HMI 设备
  - 精智面板
  - 第二代移动面板
  - 精简系列面板
  - WinCC Comfort Unified/Unified Scada RT

6.1.4 设计布局

简介

选择的定位方法取决于自动化项目的规模和公司内部的要求。

通过以下项目示例对各种定位方法的工作原理和用途进行说明。

标准化大型项目

对于这种情况，我们建议您使用自身的定位方案。这有助于针对不同 HMI 设备精确确定显示区域的位置以及实现最佳使用。可以为每个 HMI 设备分配单独的定位方案。例如，如果要实现标准形式的更改，则可将 HMI 设备的垂直和水平对齐设置为单独的定位方案。

即使生成了多个对象，也仍可对位置进行管理，因为每个对象均通过画面规则接收其位置信息。如果显示区域不足，则会自动创建具有相同布局的溢出画面。可以通过组合定位方案将放置区域模块化，进而实现极高的模板可复用性。

较小的单个项目

对于用户而言，自由定位几乎无须对布局进行规划。在小型项目中，手动更改布局仅需很短的时间。

自由定位时，还可以选择使用几个画面的生成模板，并使用溢出画面扩展可视化。使用自由定位时，还可以单独向溢出画面装配对象。

如果要使用特殊概念将对象分配到溢出画面，则可以使用位掩码来控制溢出画面。可将位掩码存储在控制器中，因此能直接从控制器连至画面布局。例如，此连接可在排除故障时为用户提供支持。

开发和测试阶段

对于这种情况，自由定位具有诸多优势。可长时间保持灵活定位。可单独移动生成的对象，因此可在项目中使用许多不同的 HMI 设备。溢出画面可确保清晰显示过程画面。会自动生成溢出画面的画面导航。

HMI 设备的切换需要利用手动定位重新生成来实现。

标准过程画面

固定位置取决于 HMI 设备。这意味着，固定位置仅适用于具有多个相同 HMI 设备的项目或仅适用于 HMI 设备的左上方区域。固定位置会被定位方案忽略，因此只能将固定位置与自由定位结合使用。

如果不计划进行更改，并且不在项目中对 HMI 设备进行切换，则固定位置是最佳选择。

规划画面层

可以控制生成的显示和操作对象在画面层中的位置，如同在 WinCC 中一样（使用生成模板的 WinCC 属性进行）。SiVArc 定义的嵌套深度仅适用于单个层之内。

参见

所生成对象的定位概述 (页 61)

6.1.5 创建用户自定义定位方案

要求

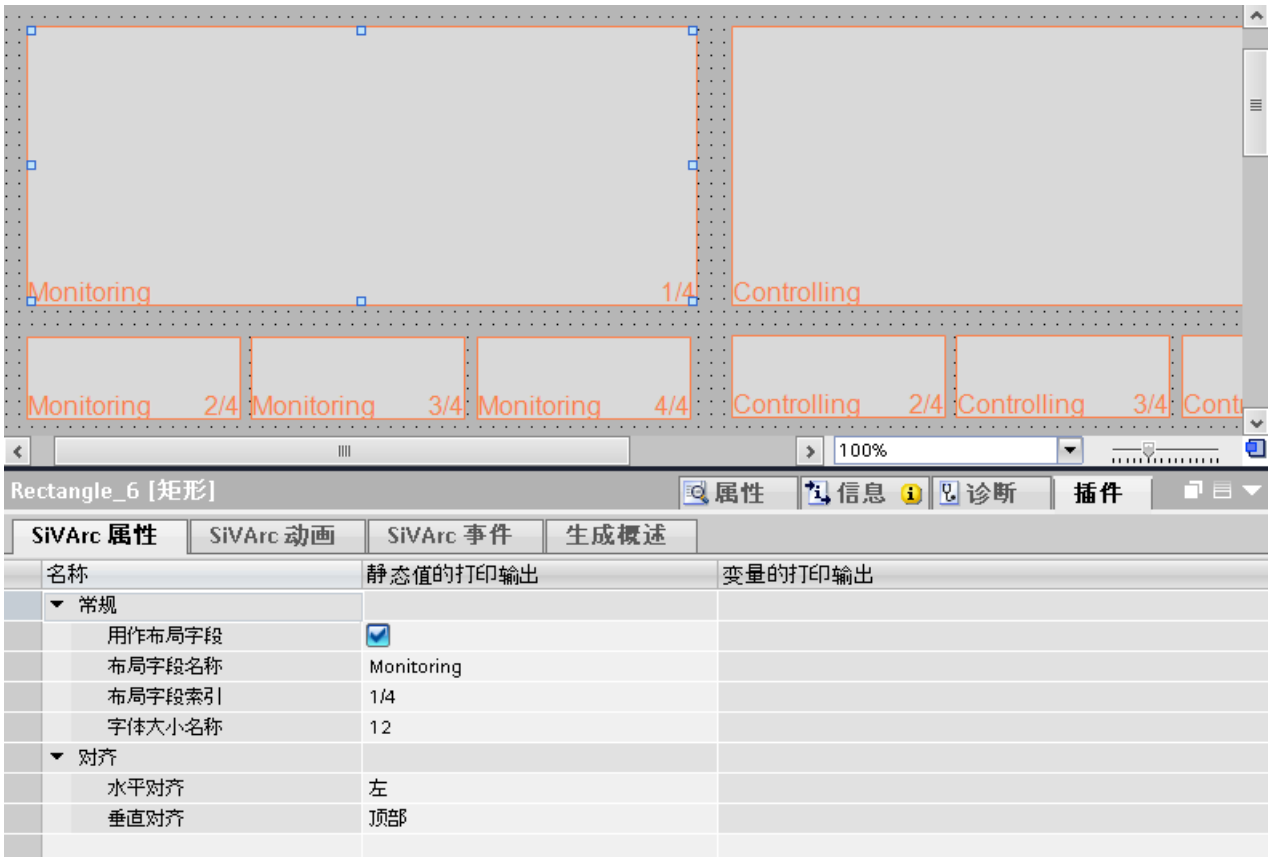
- “画面”编辑器已打开。
- “概览”(Overview) 画面已创建。

步骤

要创建定位方案，请按以下步骤操作：

1. 从工具箱窗口的“基本对象”(Basic objects) 组中，向画面添加多个矩形。
   确保用于生成的显示和操作对象的矩形足够大；否则显示对象和操作对象可能会在生成的画面中发生重叠。
2. 在矩形的 SiVArc 属性中，选择“SiVArc 属性 > 常规 > 用作布局字段”(SiVArc properties > General > Use as layout field)。

3. 定义画面中的区域。
   – 为属于同一逻辑单元的布局字段分配相同名称，例如“监视”和“控制”。
   – 为此，请在“常规 > 布局字段名称”(General > Layout field name) 下更改布局字段的名称。
   – 在“常规 > 字体大小名称”(General > Font size name) 下设置字体大小。
   – 在 WinCC 属性的“属性 > 属性 > 外观”(Properties > Properties > Appearance) 下指定布局字段的边框和字体颜色。



4. 如有必要，请在“常规 > 布局字段索引”(General > Layout field index) 下更改字段的填充顺序。布局字段将显示有名称和索引。
5. 将“概览”(Overview) 画面作为主副本保存到库中。
6. 在项目树中删除“概览”(Overview) 画面。

索引顺序

索引分配遵循编辑索引的时间顺序。如果后续为其它逻辑单元分配了一个布局字段，则无论画面中的分配规则如何，该字段都将获得该单元的最后一个索引编号。

每次变更后，索引顺序都会自动重新调整。

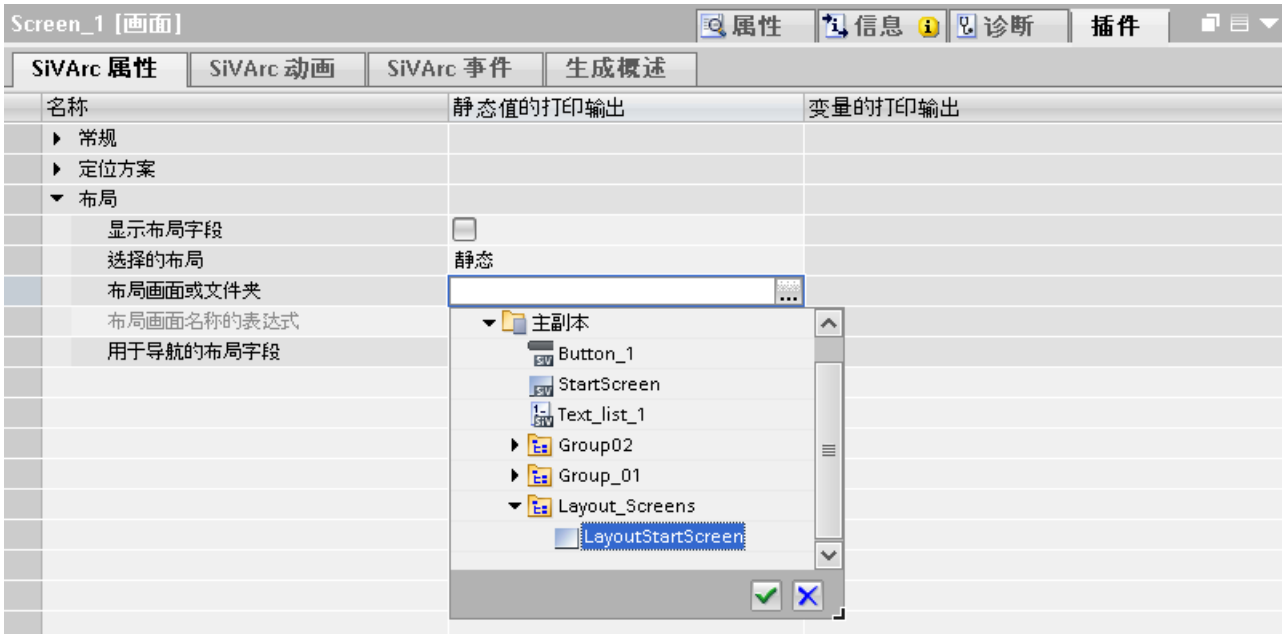
结果

定位方案已创建。可以存储画面生成模板的方案或在画面规则中将相应方案用作画面。

为生成模板永久分配定位方案

要在生成模板中使用定位方案，请按以下步骤操作：

1. 根据要用于存储新定位方案的生成模板来生成一个新画面。
2. 选择“布局选择”(Layout selection) 下的“静态”(Static) 选项。
3. 在“布局画面或文件夹”(Layout screen or folder) 下，选择所需的定位方案。



4. 删除库中的生成模板。
5. 将编辑的画面保存为库中的生成模板。
6. 在项目树中删除画面。

如果按某项画面规则使用生成模板，还需在画面规则中指定布局字段。SiVArc 在此布局字段中生成画面对象，字段索引为 1。下一个生成的对象会生成到索引为 2 的字段中，依此类推。

说明

层分配

如果在生成过程中为主副本分配一个固定层并使用自定义定位方案，则 HMI 对象会在定位方案中指定的层中生成。

在生成的画面中显示布局字段

要在生成的画面中显示布局字段，请在画面的 SiVArc 属性中选择“SiVArc 属性 > 布局 > 显示布局字段”(SiVArc properties > Layout > Show layout fields)。

参见

根据定义的方案进行定位 (页 64)

6.1.6 组态没有画面对象的溢出画面

简介

可以在 SiVArc 项目中设置带有和不带画面对象的溢出画面。组态没有画面对象的溢出画面时，可在首次生成后将生成的显示和操作对象移到溢出画面。之后可以为所有后续生成设置此位置。

步骤

要组态没有画面对象的溢出画面，请按以下步骤操作：

1. 在“插件 > SiVArc 属性 > 常规”(Plug-Ins > SiVArc properties > General) 巡视窗口中，针对“溢出画面数”输入所需的画面数。

   说明

   该主副本的每个实例均会生成溢出画面。

   要限制溢出画面的生成，在“溢出画面数”(Number of overflow screens) 下设定一个条件。

2. 禁用选项“按位掩码计算溢出画面数”(Evaluate number of overflow screens as bit mask)。
3. 必要时，启用导航按钮生成功能。
4. 定义一个或多个画面规则。
5. 启动生成过程。

SiVArc 会将所有画面对象生成到已生成的基本画面中。首次生成后，可在溢出画面中将已生成的画面对象移动至所需的位置。之后每次生成生成时都将保留修改后的画面对象位置。

说明

将生成的显示和操作对象复制到溢出画面

将溢出画面数定义为十进制数字时请注意：

如果手动将 SiVArc 生成的对象从基本画面复制到溢出画面，则在重新生成时将保留此更改。副本随后将与基本画面上的 HMI 对象一并处理，处理方式与 SiVArc 生成的对象一样，并且引用 SiVArc。

要求：副本名称必须与原名称匹配。

溢出画面和布局字段

如果在画面中组态了布局字段，则“溢出画面数”(Number of overflow screens) 和“以位掩码的形式计算溢出画面数”(Evaluate number of overflow screens as bit mask) 属性将不再生效。

参见

示例：使用溢出机制 (页 88)

溢出机制 (页 76)

6.1.7 示例：使用支持自由定位的布局

示例场景

印刷电路板制造商升级制造设备，并决定使用显示区域更大的新一代 HMI 设备。

要求

对于将对可视化进行调整的工程公司，会针对新 HMI 设备中由 SiVArc 生成的显示和操作元件开发匹配的定位方案。

执行规则

为了开发适合 SiVArc 的定位方案，可视化工程师首先在没有定位方案的情况下生成了现有项目。

然后组态工程师在 HMI 设备的较大画面上重新排列生成的显示和操作设备。后续的每一次生成过程都会保留此排列，之后组态工程师将根据需要做进一步的调整。

只有在生成的显示和操作元素的排列经过优化之后，组态工程师才会为新的 HMI 设备创建定位方案。

参见

自由定位 (页 74)

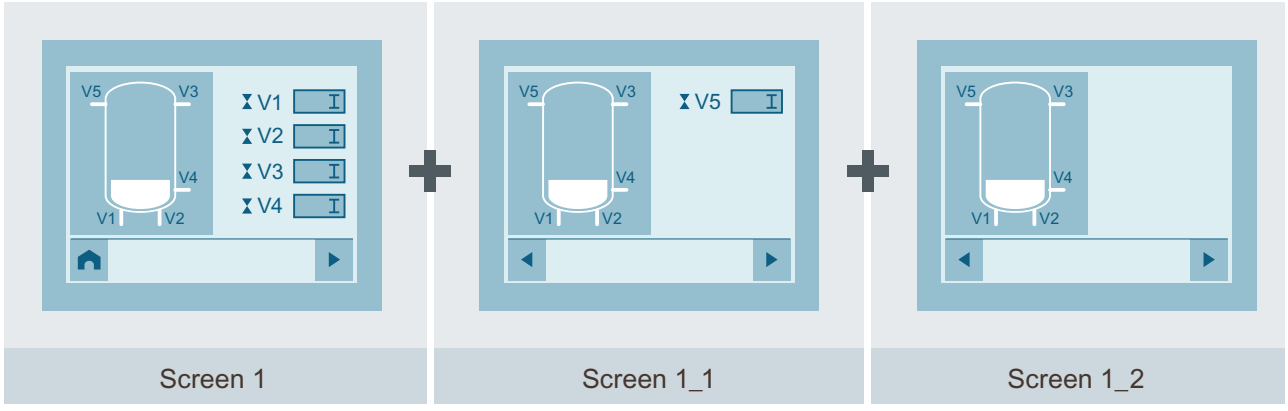
6.1.8 示例：使用溢出机制

溢出画面示例场景

将阀门添加到现有设备中，并通过状态输出进行可视化。相关 HMI 设备的显示对于附加的显示和操作元素而言过小。

执行规则

在没有任何其它组态的情况下，在第一个溢出画面中生成附加显示对象。



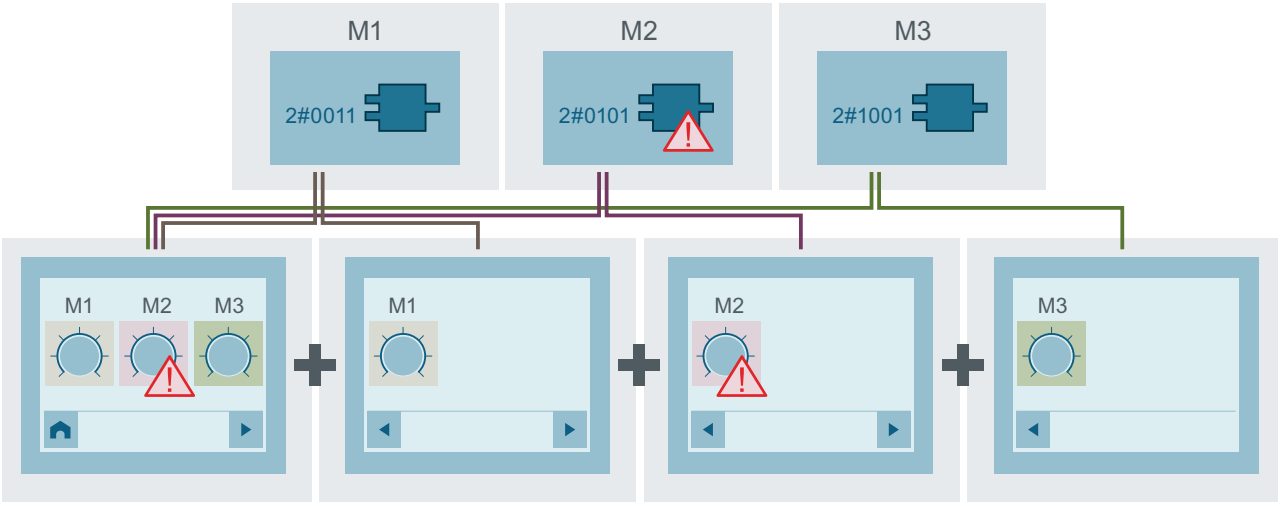
具有位掩码的溢出画面示例场景

如果设备发生错误，则当输出报警时，应确保导航到错误源。

执行规则

用于生成溢出画面的位掩码存储在控制器的输入端。由位掩码来指定到所生成溢出画面的生成对象分配。为每个位生成画面规则对象（值为 1）。

如果某个位发生错误，它将被包含在过程画面中同一位位置的溢出画面内。



以位掩码的形式计算溢出画面数

要使用位掩码组态溢出画面，请按以下步骤操作：

1. 在巡视窗口中的“插件 > SiVArc 属性 > 常规”(Plug-Ins > SiVArc properties > General) 下，输入“溢出画面数”(Number of overflow screens) 所需的位掩码，例如 11 (2#1011)。
   或
   在巡视窗口中的“插件 > SiVArc 属性 > 常规”(Plug-Ins > SiVArc properties > General) 下，针对“溢出画面数”(Number of overflow screens) 选择用于设置溢出画面的位掩码的块输入，例如 Block.Parameters("OVERFLOW_PIC").Value。
2. 启用选项“按位掩码计算溢出画面数”(Evaluate number of overflow screens as bit mask)。
3. 必要时，启用导航按钮生成功能。
4. 定义一个或多个画面规则。
5. 启动生成过程。

如果已按数字输入位掩码，则本示例中的生成期间将生成三个溢出画面。即在第一个和第三个溢出画面及基础画面中生成所使用画面规则的画面对象。

如果选择了块输入，则将在参数中处理值。如果未设置任何有效值，则将仅在基本画面中生成所使用画面规则的画面对象并输出一条错误消息。

将溢出画面的位掩码组态为变量

要使用已保存在变量中的位掩码组态溢出画面，请按以下步骤操作：

1. 在巡视窗口的“插件 > SiVArc 属性 > 常规”(Plug-ins > SiVArc properties > General) 下，针对“溢出画面数”(Number of overflow screens) 输入已为溢出画面位掩码定义的 SiVArc 变量名，例如“SiVArcVariable”。
2. 启用选项“按位掩码计算溢出画面数”(Evaluate number of overflow screens as bit mask)。
3. 必要时，启用导航按钮生成功能。
4. 定义一个或多个画面规则。
5. 启动生成过程。

在生成期间处理所选变量的当前值。如果未创建任何变量，SiVArc 将在基本画面中生成所使用画面规则的画面对象。

参见

溢出机制 (页 76)

组态没有画面对象的溢出画面 (页 86)

6.1.9 示例：使用动态布局

示例场景

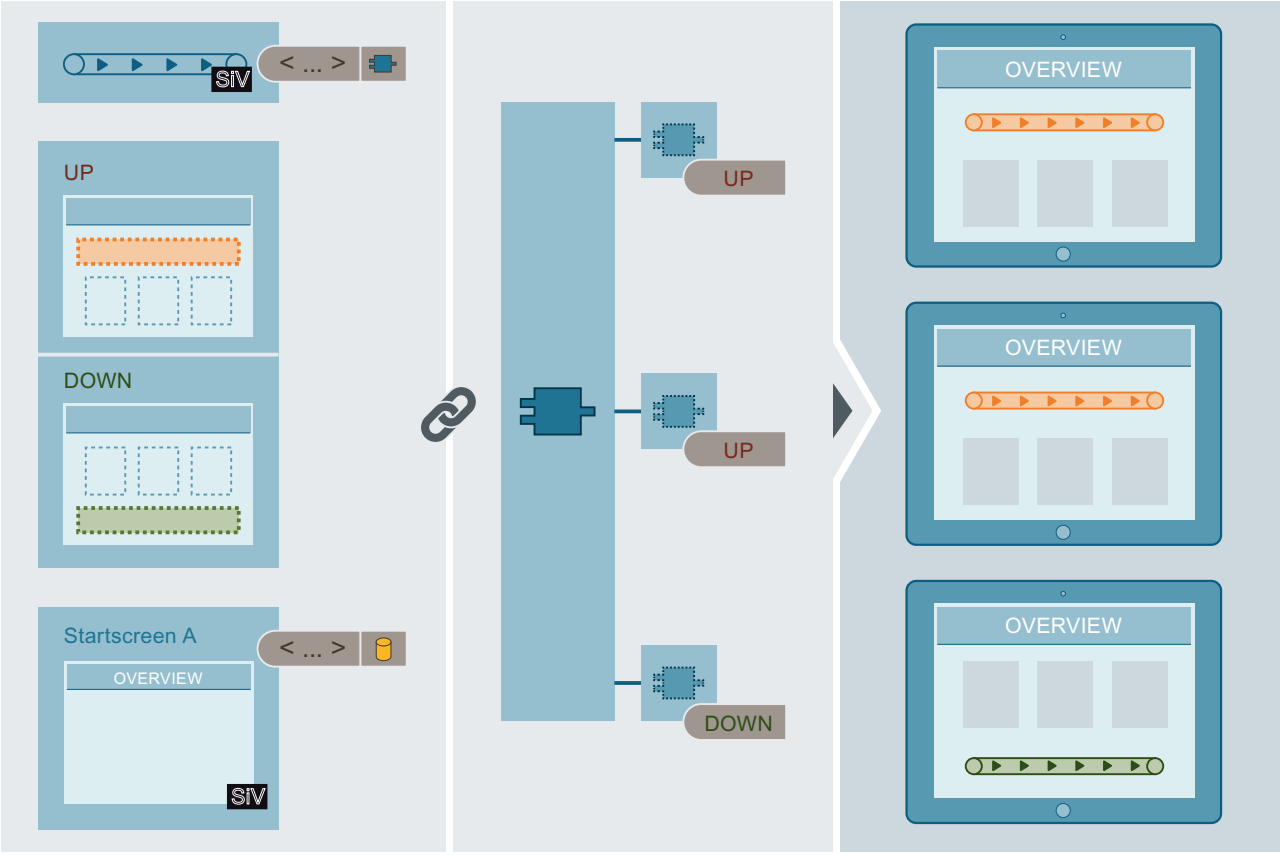
两个传送带需要不同的画面排列，但具有相同的操作对象。

要求

画面规则的数量应减少。

执行规则

在库文件夹中创建和存储两个定位方案。在程序段中定义一个 SiVArc 变量。SiVArc 变量控制传送带在画面中的排列位置。在评估画面规则期间选择匹配的定位方案。其中一个传送带排列在画面区域的顶部；另一个则显示在底部。

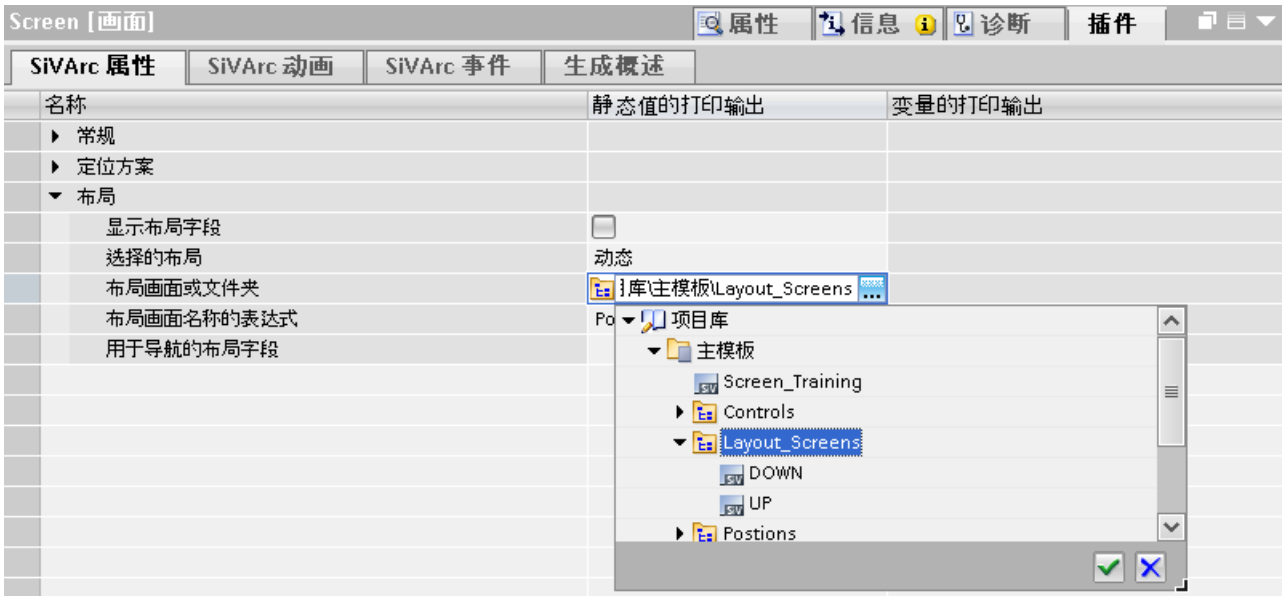


建立动态定位方案

如果要根据具体条件为画面分配定位方案，请将含定位方案的文件夹分配给生成模板。然后分配一个 SiVArc 表达式，该表达式可返回所选文件夹中包含的定位方案的名称。

1. 在库文件夹中创建多个定位方案。
2. 例如，将文件夹命名为“Layout_Screens”。
3. 打开要为其分配动态定位方案的画面的生成模板。
4. 在 SiVArc 属性中的“布局选择”(Layout selection) 下，选择“动态”(Dynamic) 模式。

5. 在“布局画面或文件夹”(Layout screen or folder) 下，选择“Layout_Screens”文件夹。



6. 在“布局画面名称表达式”(Expression for layout screen name) 下组态一个 SiVArc 表达式，该表达式可返回所选文件夹中包含的布局画面的名称。
   例如，可在用户程序中定义一个 SiVArc 变量，并将其用作条件。随后将此程序块所需的定位方案的名称分配给该变量。
7. 将编辑的画面保存为库中的生成模板。
8. 在项目树中删除画面。

说明

如果为“布局选择”(Layout selection) 选择“动态”(Dynamic) 模式以生成一个画面的多个画面元素，则不会显示所有动态分配的布局字段。

即使在画面的 SiVArc 属性中启用“SiVArc 属性 > 布局 > 显示布局字段”(SiVArc properties > Layout > Show layout fields)，也将仅显示首个生成的画面元素的布局字段。

参见

根据定义的方案进行定位 (页 64)

画面图例 (页 287)

6.1.10 示例：使用组合布局

示例场景

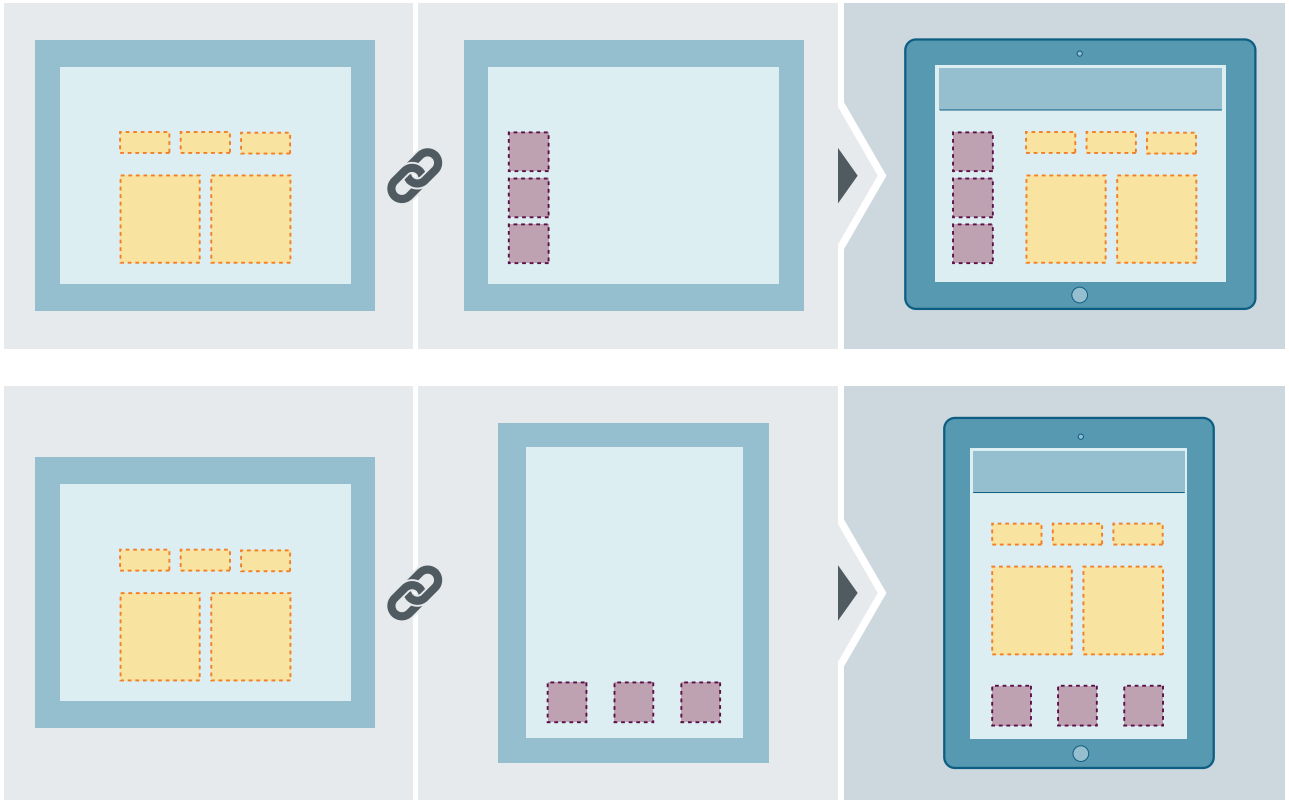
用于放置、焊接和封装的生产线的概览画面具有相似的布局，但在两种不同 HMI 设备（横向格式和纵向格式）的使用上各不相同。这会导致画面的定位有所不同。

要求

基于现有定位方案的高度可复用性实施过程画面的布局。

执行规则

生产线中心画面对象的定位方案与其它定位方案结合使用。借此，显示和操作对象排列在纵向格式 HMI 设备的底部。对于横向格式的设备，对象则被排列在一侧。



参见

根据定义的方案进行定位 (页 64)

6.1.11 示例：使用生成的画面导航

示例场景

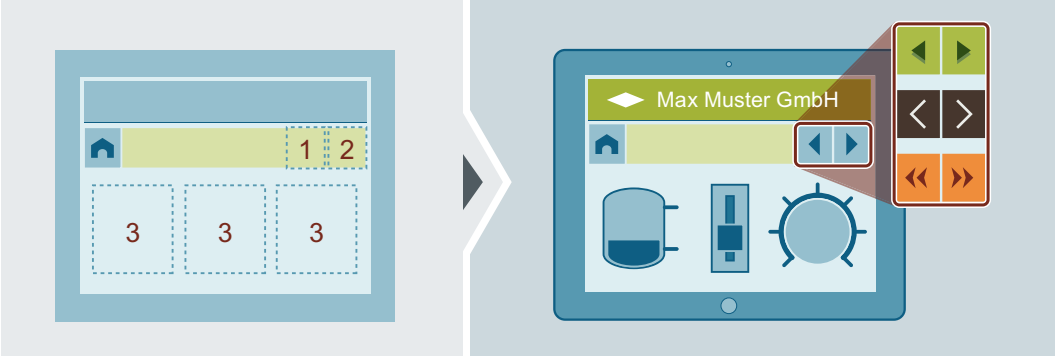
现有可视化对象的画面导航与公司内有关企业设计和标准化的新准则有偏差。

要求

使画面导航的组态变得更加生动高效。

执行规则

负责这项工作的工程公司决定使用用户自定义定位方案实施新的画面导航。这样便可根据公司特定的说明使用企业设计的操作对象。

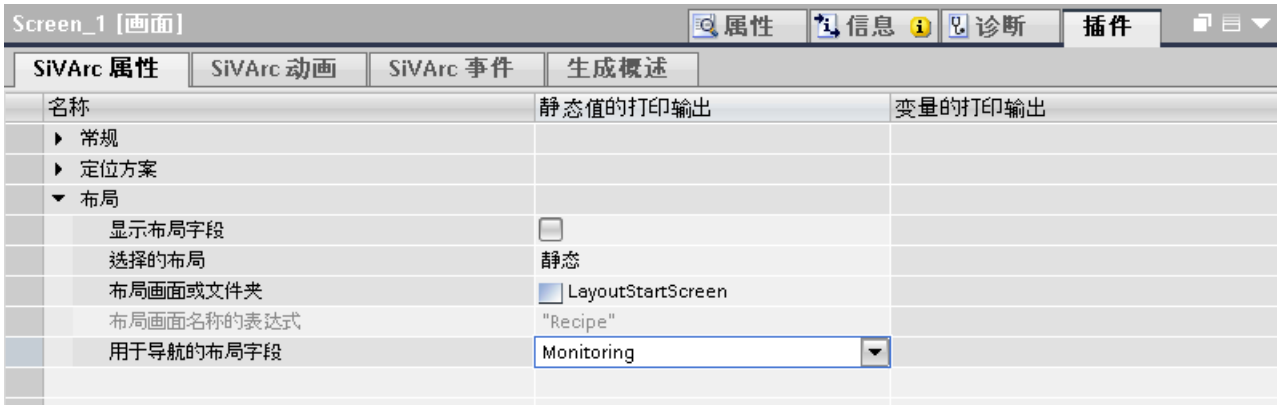


在溢出画面中启用自动画面导航。因此，定义的设备部分会显示在具有多个溢出画面的单个画面中。溢出画面减少了画面的生成模板数，并对显示进行了标准化。此外，还确保了所有对象均在可见区域中生成。

同时，也会对公司特定的画面对象进行存储，以用于自动画面导航。

将布局字段用于导航按钮

要使用用户自定义定位方案实现画面导航，请在画面 SiVArc 属性的“SiVArc 属性 > 布局 > 导航布局字段”(SiVArc properties > Layout >Layout field for navigation) 下选择用于导航按钮的布局字段，例如“监视”(Monitoring)。



如果在生成过程中使用此生成模板时出现溢出画面，则导航按钮将位于“监视”(Monitoring) 布局字段中。

在溢出画面中启用自动画面导航

要在溢出画面中启用自动画面导航，请在画面的生成模板中选择“导航按钮”(Navigation buttons) 选项。

参见

根据定义的方案进行定位 (页 64)

6.2 生成模板的创建

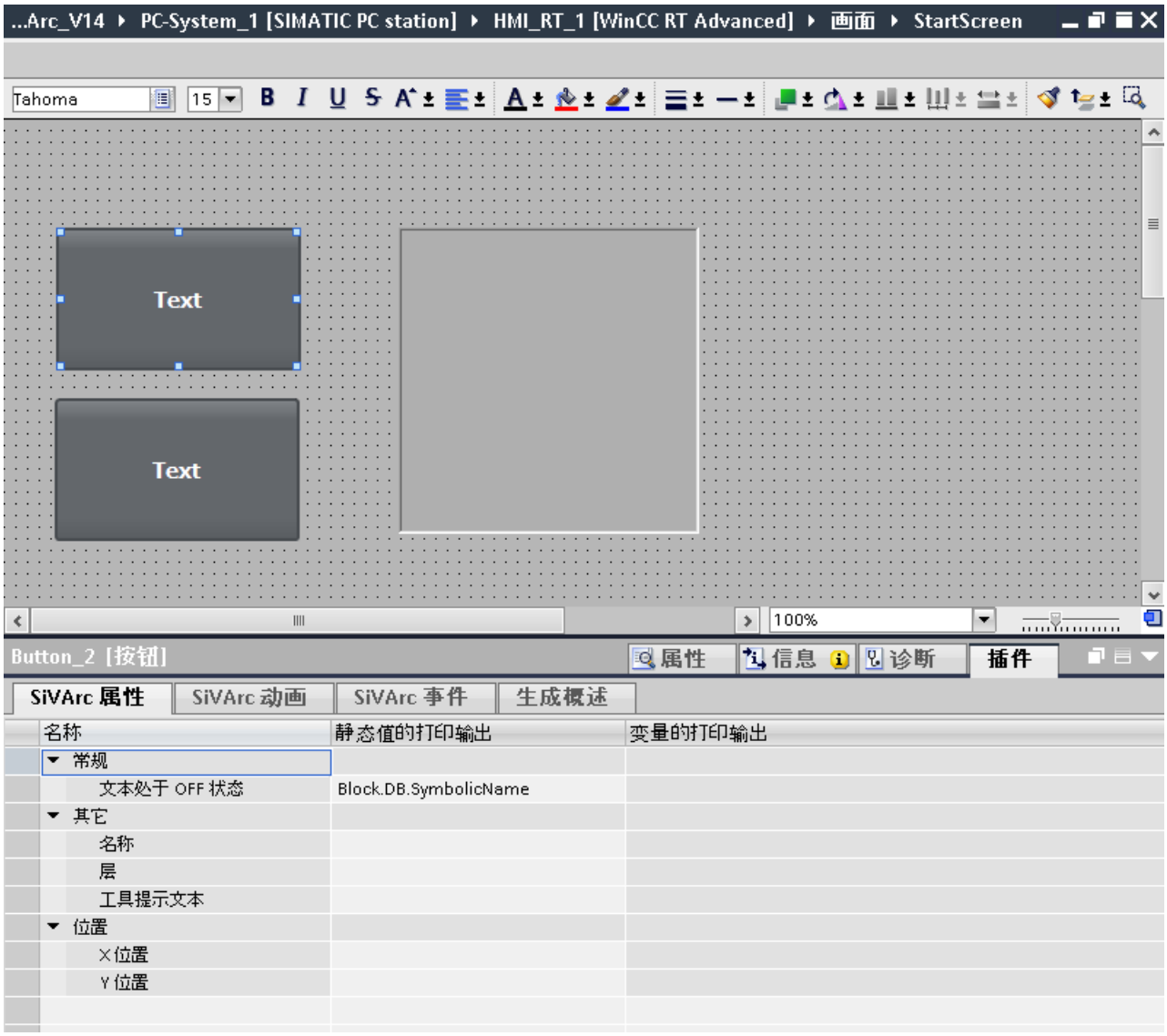
6.2.1 SiVArc 中的生成模板

定义

生成模板 是来自库的 HMI 对象，其不仅组态了固定定义的 WinCC 属性，还组态了 SiVArc 属性。SiVArc 属性为对象属性，首先被分配为变量/表达式。根据 SiVArc 机制，SiVArc 属性仅包含文本，如生成期间的对象名称、标签或变量名称。

工作原理

SiVArc 属性可以是静态或动态的。在“SiVArc 属性”(SiVArc properties) 选项卡中，可组态生成模板的属性。



选项卡包括三列：

- 名称
  此列会列出可用属性。
- 静态值表达式
  在此列中，分配具有固定值的属性或返回字符串或数字的 SiVArc 表达式。
  生成可视化时，该主副本的每个实例中会输入固定值。注意，例如当对象名称在一个画面中多次使用时，利用“名称”(Name) 属性可确保对象名称的唯一性。
- 变量表达式
  在此列中，可分配带有变量名称或能返回变量名称的 SiVArc 表达式的属性。

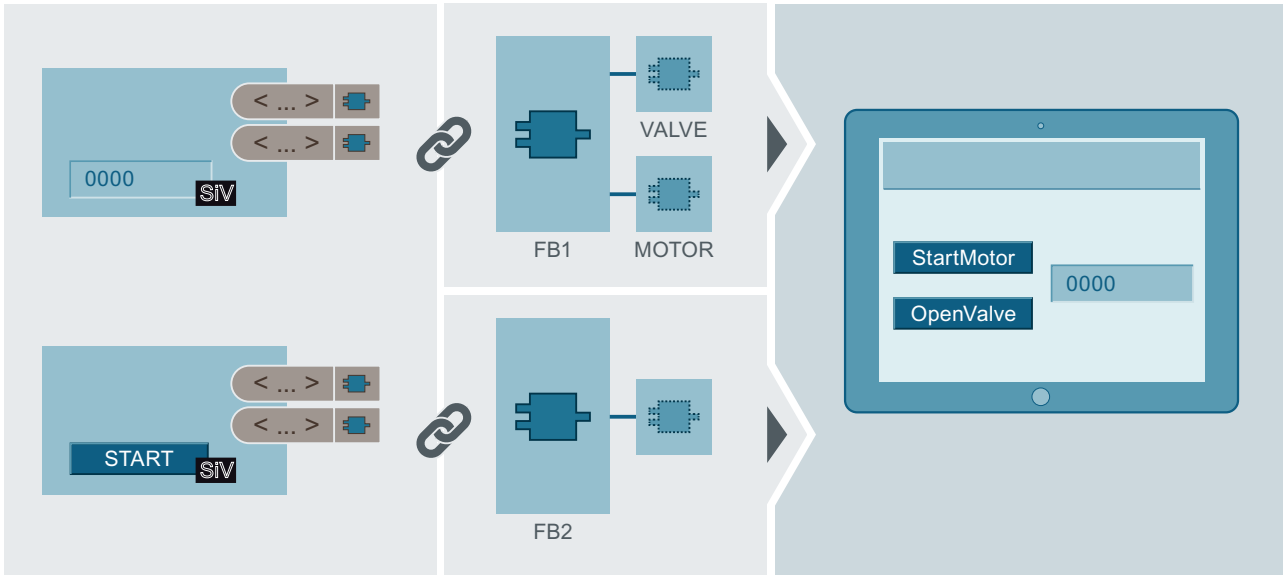
然后将组态的生成模板存储在项目库中，之后将其互联到函数块。为各个生成模板创建单独的互联。可视化生成期间，将评估 SiVArc 属性。

画面和画面对象的生成模板

每个 HMI 对象生成一个生成模板。

“SiVArc 属性”(SiVArc properties) 选项卡只适用于 SiVArc 支持的对象。

下图给出了通过参考来自 STEP 7 的指令的生成模板生成画面对象的图解。生成 HMI 对象时，将评估 SiVArc 属性。会生成“Label”或“Name”等对象属性。



SiV SiVArc 生成模板

< ... > 引用文本源的 SiVArc 属性

函数块

实例化函数块

在模板画面中生成画面对象

SiVArc 支持在模板画面中生成画面项以及创建模板。可以使用**插件**编辑器组态模板。在模板画面中可以执行下列操作：

- 完整生成和合并场景的行为与正常画面相似
- 通过将模板选作主副本来组态规则。
- 根据已组态的模板属性来更新画面模板。

说明

除专业设备外，所有其它设备均支持模板画面。

将类型用作生成模板

生成画面类型实例和面板实例时，请注意以下事项：

- 如果使用 SiVArc 生成所需画面类型，会更新项目里的所有实例，甚至包括非 SiVArc 创建的实例。
- 即使删除生成的实例与画面类型之间的连接，然后更改画面，所做更改仍然会在下一个生成过程中被该画面类型的新实例所覆盖。
- 如果使用 SiVArc 生成所需的全局库画面类型，该画面类型将添加到项目库中。
- SiVArc 在生成过程中会更新项目和库中最新类型版本所使用的画面类型。

说明

将画面类型用作生成模板

“画面规则”(Screen rules) 编辑器中的画面类型只能作为画面对象使用。因此，画面类型将始终显示在画面窗口中。

文本列表的生成模板

借助 SiVArc，可以直接在用户程序中创建多语言文本列表条目（例如用于函数块的状态文本或用于块参数的接口说明）。生成过程中，可将这些文本与相应的显示和操作对象互连。这样，便可为项目生成描述性文本。

用户可以在块中以多语言格式存储文本列表条目，也可以从块参数的符号表取出这些条目：

- 块中的文本列表条目
  在程序段或程序块中创建文本列表条目。要选择正确的程序段，请在巡视窗口中单击程序段的任意区域中的程序块。还可将 SiVArc 表达式用于文本列表条目。
  文本定义和文本列表条目在文本列表主副本内以相同的名称链接在一起。
- 块参数中的文本列表条目
  对于符号表中的各个参数，可为文本列表条目创建一个由 SiVArc 处理的注释。

  说明

  使用 STEP 7 中的文本源

  在一个文本列表中只能处理一个文本源。因此，将相应块中的多个文本**或**一个符号表中的多个文本用于一个文本列表。

自动生成对象的生成模板

通过 SiVArc 自动生成以下对象：

- 用于显示显示屏画面的画面窗口
- 溢出画面的导航按钮

可使用生成模板自定义自动生成的对象。

为此，将自定义对象保存在项目库的“主副本”(Master copies) 下。

存储自定义对象时，请遵守以下准则：

- 必须使用名称“DefaultScreenWindowControl”对画面窗口的生成模板进行存储。
- 必须使用名称“NextButton”和“PrevButton”将导航按钮的生成模板存储在库中。可单独组态这些按钮。

如果未自定义生成模板，会将工具箱的默认模板用于生成。

定位方案的生成模板

要指定生成画面中画面对象的位置，请在画面规则中指定分配给画面对象的布局字段组。

使用复制规则的画面生成模板

在 SiVArc 中，您可以通过使用复制规则生成模板来生成 HMI 画面。使用复制规则的模板画面生成利用将单个或多个模板画面添加到在项目中组态的多个 HMI 设备的任务。模板画面通过复制规则自动从项目库或全局库复制到 HMI 设备。作为 SiVArc 生成的一部分，HMI 设备的模板画面在“画面管理”(Screen management) 下的“模板”(Template) 文件夹中可用。

使用模板画面属性生成 HMI 画面

在 SIVArc 中，您可以使用“插件”(Plug-ins) 下的“SiVArc 属性”(SiVArc properties) 通过模板画面组态 HMI 画面。必须确保以下内容:

- 在 HMI 画面的巡视窗口中输入的“模板画面”名称必须与模板画面的名称相似。
- “模板画面”的解析表达式必须与项目中现有模板画面的名称相匹配。

在使用模板画面组态 HMI 画面后，HMI 画面将放置在“库”下的“主副本”文件夹中。创建了一个画面规则，它使用现有的模板画面，并在 SiVArc 生成期间应用于 HMI 画面。SiVArc 生成与组态的模板画面相同名称的 HMI 画面。

说明

面板和 RT Advaned 设备支持使用复制规则和 SiVArc 属性的模板画面。

参见

生成模板要求 (页 116)

示例：使用复制规则生成模板画面 (页 139)

示例：使用模板属性生成 HMI 画面 (页 140)

6.2.2 支持的 HMI 对象

可通过控制数据生成的 HMI 对象

SiVArc 可生成以下 HMI 对象，具体取决于为哪个 HMI 设备而生成：

HMI 对象	精简系列面板	精智面板/第二代移动面板 RT Advanced	RT Professional	WinC Comfort Unified/ Unified SCADA RT
外部变量 [1]	√	√	√	---
基于库中的模板副本：				
棒图	√	√	√	√
画面 [1]	√	√	√	√
画面窗口	---	---	√	---
模板画面	---	√	---	---
I/O 域	√	√	√	√
图形 I/O 域	√	√	√	---
GRAPH 概览	---	√	√	---
趋势图	√	√	---	---
f(x) 趋势图	---	√	√	---
f(t) 趋势图	---	---	√	---

HMI 对象	精简系列面板	精智面板/第二代移动面板 RT Advanced	RT Professional	WinC Comfort Unified/ Unified SCADA RT
PLC 代码视图	---	√	√	---
弹出画面 [1]	---	√	---	---
ProDiag 概览	---	√	√	---
圆形按钮	---	---	√	---
开关	---	√	---	√
按钮	√	√	√	√
滚动条	---	√	√	√
符号 I/O 域	√	√	√	√
文本域	√	√	√	√
文本列表	√	√	√	√
仪表	---	√	√	√
依照库中的类型：				
画中画窗口 [1]	---	---	√	---
报警	√	√	√	---
面板	---	√	√	√
事件	√	√	√	√
动画	√	√	√	-

[1]：可以采用结构化存储

说明

根据所使用的 HMI 设备，SiVArc 支持的 HMI 对象的属性可能会改变。

无需控制数据即可生成的 HMI 对象

SiVArc 可通过库的类型或模板副本生成或实例化以下对象：

HMI 对象	精简系列面板	精智面板/ 第二代移动面板 RT Advanced	RT Professional	WinCC Unified SCADA RT/ Unified 精智面板
画面	√	√	√	
变量				√
内部变量	√	√	√	√
变量表	√	√	√	√
脚本				
C 脚本	---	---	√	
VB 脚本	√	√	√	√
文本列表	√	√	√	√

带有设备相关最大值的属性

为以下 HMI 设备生成可视化时，各个属性的最大值会受限：

属性	精简系列面板	精智 面板	移动面板 第二代	WinCC Unified 精智面板	WinCC Unified SCADA RT
OFF 时文本（长度）	320	500	500	1000	1000
工具提示（长度）	320	1000	1000	320	320
文本域（文本）	320	32767	32767	30000	30000
文本列表条目（文本）	320	320	320	320	255

6.2.3 文本源

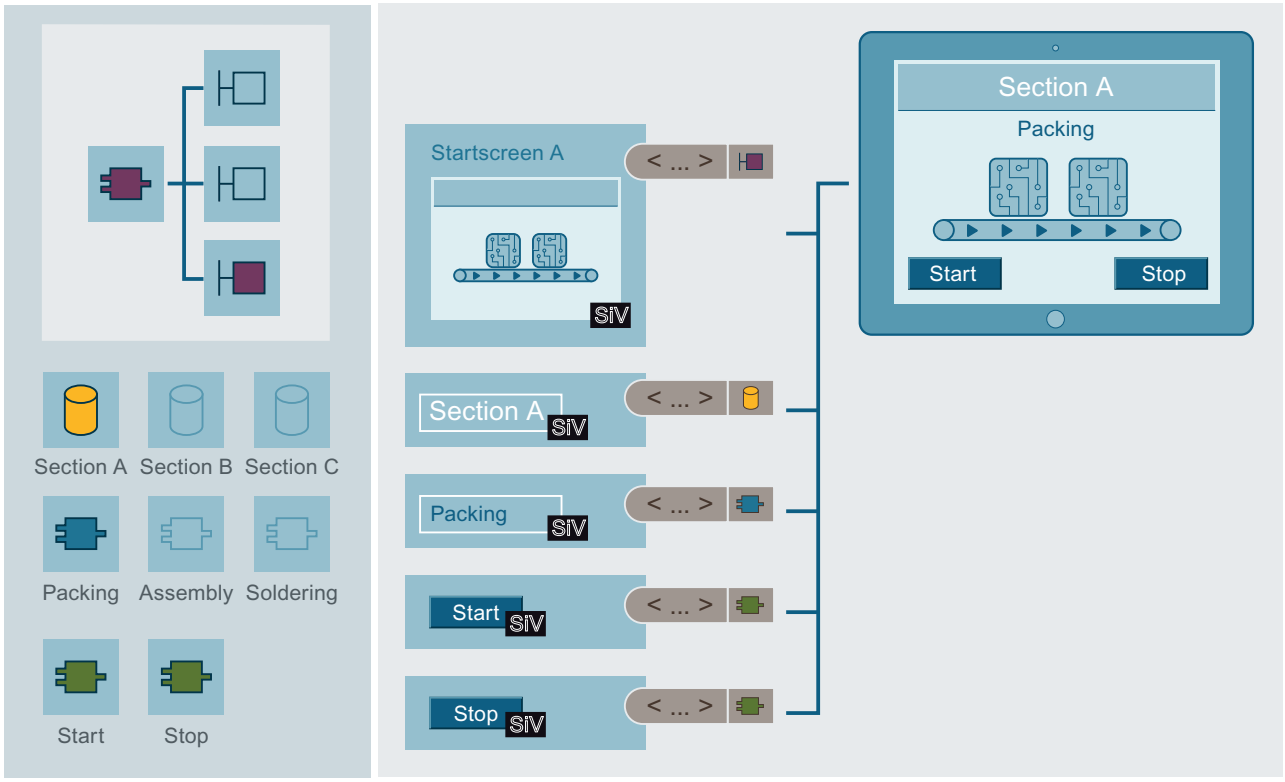
6.2.3.1 SiVArc 项目中的文本源概述

定义

通过 SiVArc 可访问来自 STEP 7 和其它 TIA Portal 编辑器的文本来进行可视化。与传统的 WinCC 组态不同，这些文本由控制器程序员创建。在使用 SiVArc 的可视化中多次使用这些文本。

下图给出了如何使用 SiVArc 构建画面：

- STEP 7 中有多种文本源可用，例如，程序段、数据块或函数块。
- 一个画面由多个生成模板组成。生成模板的 SiVArc 属性可访问文本源。
- 生成期间，SiVArc 将处理引用的文本源并填充 HMI 对象的 SiVArc 属性。



SiV SiVArc 生成模板

< ... > 引用文本源的 SiVArc 属性

Main [OB1]

用户程序中的程序段

数据块

过程块

标准块

优势

SiVArc 可生成多种文本，具体取决于链接到生成模板的函数块。因此，一个生成模板可用于多种位置。WinCC 项目可轻松调整。文本的一致性通过可视化中的用户程序进行传送。

String 函数

为了最大程度地提高生成模板的可复用性或优化显示文本，SiVArc 提供多种 string 函数，如“Split”、“Contains”或“Trim”。

来自 STEP 7 的文本源

SiVArc 属性可参考来自 STEP 7 的以下文本：

- 程序段
  - SiVArc 文本和 SiVArc 变量
  - 程序段标题
  - 程序段注释
- 数据块
  - 符号名称
  - 项目树中的存储路径
  - 注释
  - 块编号
  - TagPrefix
  - 类型（IDB、MDB）
- 函数块
  - 注释
  - 参数值
  - 项目树中的存储路径
  - 块编号
  - 标题
  - 库类型的类型版本
- 主块 (OB)
  - 符号名称
  - 符号地址
  - 版本
  - 文件夹路径
  - 类型
  - 编号

- I/O 设备
  - 产品编号
  - 不变量类型
  - 名称
- PLC 变量
  - HMI 前缀
- HMI 设备
  - 设备名称/分辨率/类型
- HMI 应用程序
  - 名称
  - 类型

来自硬件数据的文本源

SiVArc 属性可访问 HMI 设备的以下属性：

- 运行系统软件
  - 名称
  - 类型
- HMI 设备
  - 名称
  - 类型

来自库的文本源

SiVArc 属性可访问库对象的以下属性：

- 名称
- 库中的存储路径

参见

用户程序对生成模板的影响 (页 125)

SiVArc 表达式 (页 113)

6.2.3.2 SiVArc 文本

简介

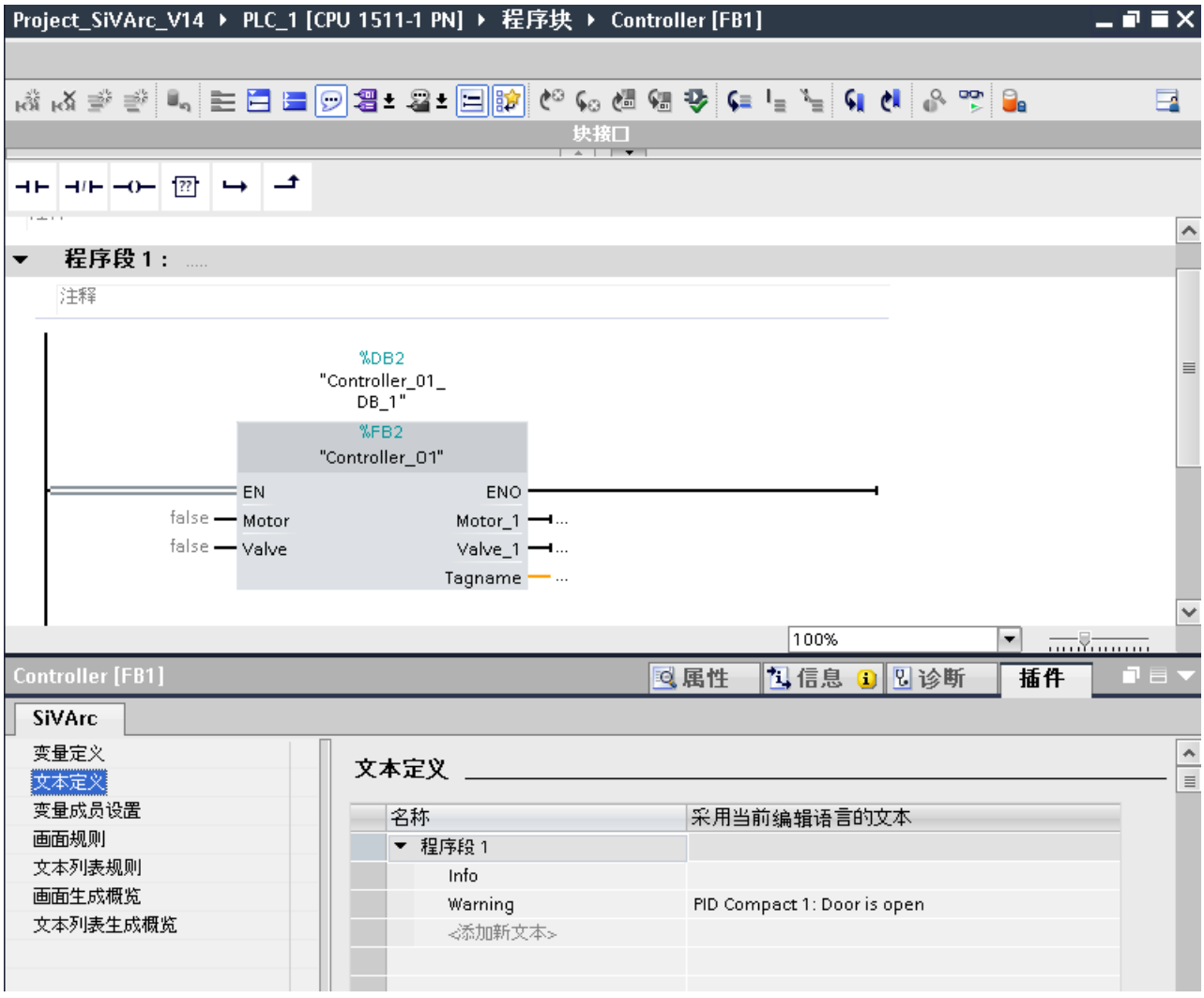
凭借 SiVArc 可将文本定义为文本列表条目，用于生成可视化。此功能集成在 STEP 7 的用户程序中，并可在任何程序段标题和块中使用。

要定义 SiVArc 文本，请在所需程序段标题或块标题的巡视窗口中选择“插件”(Plug-ins) 选项卡。

定义

SiVArc 文本是 STEP 7 中创建的文本列表条目。当生成可视化时，将其用于在 HMI 中生成文本列表条目。文本定义和文本列表条目通过名称相链接。

在文本列表规则中使用程序块时，SiVArc 文本以文本列表中的文本列表条目的形式生成。



可以静态或动态方式指定 SiVArc 文本：

- 静态：分配文本作为文本定义。也可以多种语言组态此文档。
- 动态：指定 SiVArc 表达式作为文本定义。

指定文本和 SiVArc 表达式时，将使用 SiVArc 表达式。

参见

SiVArc 变量 (页 114)

多语言对生成模板的影响 (页 128)

6.2.4 用户程序中支持的对象

程序块

SiVArc 支持下列程序块：

- 函数块 (FB)
- 函数 (FC)
- 数据块 (DB)
  - 全局 DB
  - 背景数据块

FB 和 FC 在用户程序中调用。画面规则中只能使用 FB 和 FC。也可以将 FB 和 FC 作为库中的主副本和类型。

程序块的语言

SiVArc 允许为程序块使用下列编程语言：

- STL
- FBD
- LAD
- SCL

工艺对象

SiVArc 支持以下工艺对象：

- PID 控制：
  - Compact PID
  - PID 基本功能

参见

PLC 变量支持的数据类型 (页 283)

6.2.5 SiVArc 脚本语言

定义

SiVArc 脚本语言是一种衍生自 VBS 的脚本语言，专用于具有 SiVArc 选项的 TIA Portal。

SiVArc 脚本语言可对 TIA Portal 中的文本源进行寻址。这样一来，SiVArc 脚本语言可链接用户程序和 TIA Portal 中的可视化。

使用 SiVArc 脚本语言可建立生成模板中的 SiVArc 表达式。生成期间，SiVArc 将评估所有 SiVArc 表达式。利用这种方式，通过一个模板可创建大量一致的 HMI 对象。

“SiVArc 表达式”编辑器

在 SiVArc 编辑器中单击表格行以编程 SiVArc 表达式时，会打开 多行编辑器。“SiVArc 表达式”编辑器支持使用多种功能：

- 自动完成
  如果输入字母或字符，“SiVArc 表达式”编辑器会建议使用兼容的运算符、SiVArc 对象属性、属性以及以此字母或字符开头的函数。输入的表达式也会自动保存在编辑器中。
- 语法突出显示
  在“SiVArc 表达式”编辑器中，关键字使用不同的颜色突出显示。未知词语也如此标记。下表列出了最重要条目的预设颜色：可在“选项 > 通用 > 脚本/文本编辑器”(Options > General > Script / text editors) 下更改默认设置。

颜色	含义	示例
蓝色	运算符 And、Or、Xor、Not Boolean If 函数	And True If
黑色	其它运算符 字符 其它函数	+ , TrailNum
深青色	字符串	“SG_NR”
红色	未知元素	$

- 错误显示
  “SiVArc 表达式”编辑器突出显示脚本中的错误并以工具提示的方式显示错误原因。

通过选择表达式或使用快捷菜单中的命令，可更改已创建的 SiVArc 表达式。

可以复制或剪切一个或多个表达式并将它们粘贴到其它 HMI 对象的“SiVArc 属性”(SiVArc properties)选项卡。

可在 SiVArc 插件编辑器中拖放脚本。

公式规则

注意下列构成 SiVArc 表达式的规则：

- 空 SiVArc 表达式返回空字符串。
- 使用引号标记字符串常量
- 通常字符串常量允许所有的字符。
- 如果在字符串中使用引号或反斜线，在前面放置一个反斜线作为转义字符：
  `\"`
  `\\`。
- 字符串常量中的换行，使用 `\n` 进行声明。
- 程序块的绝对调用只允许使用下列关键字（SiVArc 对象）。
  - `Block`
  - `StructureBlock`
  - `ModuleBlock`
  - `SubModuleBlock`

  要对程序块的属性寻址，可使用点将 SiVArc 对象连接到 SiVArc 对象属性，例如，`ModuleBlock.SymbolicName` 用于对符号名称寻址。

以二进制代码形式输入数据

要以二进制代码形式输入数据，需使用前缀“2#”，例如 2#00000101，表示变量的位 0 和位 2 置位。

如果要使用二进制代码，则请遵照以下信息：

- 必要时可使用所有运算符和二进制代码，例如 `2#1010 + 2#1111` = 25
- 必要时可在表达式中使用二进制代码和 SiVArc 变量，例如 `VAR_1 Or 2#11100` = 29
- 必要时可使用二进制代码和其他常数值，例如 `25 * 2#11100` == 700
- 一个二进制代码最多可包含 32 个位。
- 也可以使用“格式”(Format) 功能指定二进制格式。为此，可以使用“`b`”作为作为第二操作数。

更多信息

可以在参考中找到有关 SiVArc 表达式结构的详细信息。

参见

参考 (页 239)

6.2.6 SiVArc 表达式

6.2.6.1 SiVArc 表达式概述

定义

SiVArc 表达式是一种返回文本的函数。这些文本将填充生成的 HMI 对象的所选属性。

SiVArc 表达式通过 SiVArc 对象访问文本源。SiVArc 对象可对 STEP 7、HMI 设备或库数据中的程序调用中的块进行寻址。

与 ES 或运行系统脚本不同，SiVArc 表达式可将来自其他 TIA Portal 编辑器的数据和结构永久链接到 WinCC 组态。用户程序、库或 HMI 设备的更改和调整将直接影响可视化。

SiVArc 表达式的语法元素

SiVArc 表达式基于 SiVArc 脚本语言形成。

在 SiVArc 表达式中可以使用下列语法元素：

- SiVArc 对象
- SiVArc 对象属性
- SiVArc 变量
- 布尔值 `True`/`False`
- 字符串
- 数字
- 运算符
- 预定义函数
- If 条件

使用 SiVArc 表达式组态

SiVArc 表达式用于以下组态：

- 制定生成 HMI 对象的条件

  说明

  请注意公式和寻址条件中名称的正确拼写。
  仅在生成期间输出错误消息。

- 实现用于可视化的属性、事件和动画的动态生成
- 组态外部变量的存储位置和存储结构
- 组态生成的 HMI 对象的存储结构

参见

SiVArc 变量 (页 114)

SiVArc 对象属性 (页 264)

“Format”函数 (页 267)

SiVArc 表达式的结构 (页 123)

6.2.6.2 SiVArc 变量

定义

SiVArc 变量是用户定义的变量。可以为组织块“主程序 (OB1)”和每个程序段创建多个变量。

可以定义变量名称和所需值。

可在用户程序中使用 SiVArc 变量存储关于程序块的实例特定信息。可在 SiVArc 表达式和条件中使用 SiVArc 变量。

说明

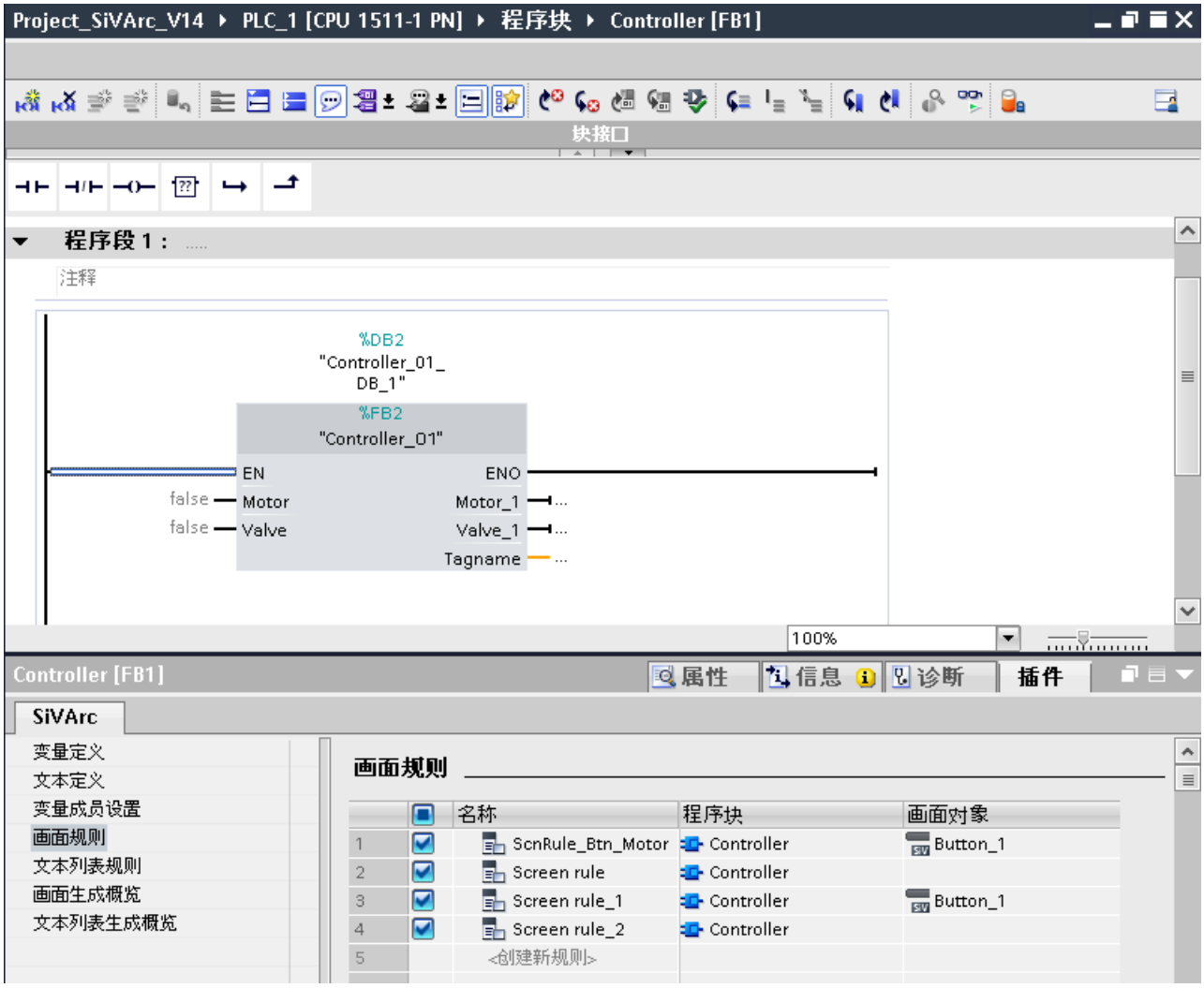
在画面规则中使用 SiVArc 变量

如果在画面规则中使用 SiVArc 变量评估实例特定信息，则要使用 `IsDefined("Variablenname")` 函数。通过这种方式可以查询是否存在 SiVArc 变量。这样可避免因不存在变量而导致生成错误。

创建和使用 SiVArc 变量

在程序段或程序块中创建 SiVArc 变量。

要选择正确的程序段，请在巡视窗口中单击程序段的任意区域中的程序块。



基于巡视窗口中的显示画面，决定是否应在该程序段中创建相应变量。

按如下所述使用 SiVArc 变量：

- 在程序段中
  变量定义在此程序段中有效。
- 在程序块中
  变量定义在此程序块的所有程序段中都有效。通过变量，可对从相应程序块中调用的所有程序块进行寻址。
  如果在程序块中使用 SiVArc 变量，则 SiVArc 变量必须位于调用块中。
  示例：
  在 FB1 中定义 SiVArc 变量。FB1 调用 FB2。要访问 SiVArc 变量，需要为 FB2 定义画面规则。

说明

优先化 SiVArc 变量

如果使用多个具有相同名称的 SiVArc 变量，则将使用 SiVArc 最近找到的条目。例如，如果 SiVArc 变量在程序段和程序块注释中的名称相同，则 SiVArc 会使用来自程序段注释的变量值。

参见

SiVArc 表达式概述 (页 113)

“画面规则”编辑器 (页 29)

SiVArc 文本 (页 108)

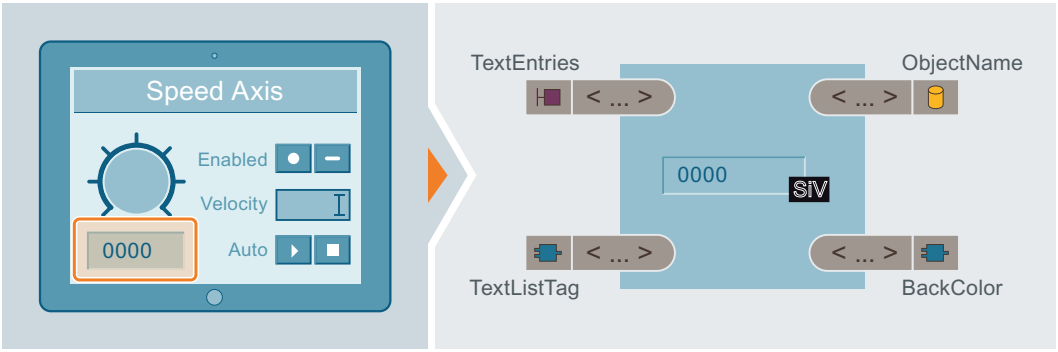
6.2.7 生成模板要求

优化的可复用性

对于生成模板而言，最为重要的要求是可复用性。您可以通过使用库中的类型实例概念实现最佳可复用性。

例如，如果用户程序中的块可供使用并通过库在整个公司内使用，则将一组生成模板分配给库中的块类型是一种可行操作。

然后，您可以通过 SiVArc 属性对这些生成模板进行组态，从而使用户程序中的每个实例以及项目树中库类型的每个实例均可通过该生成模板组进行可视化处理。应具备尽可能多的变型。



SiV SiVArc 生成模板

< ... > 引用文本源的 SiVArc 属性

生成模板作为类型

独立的生成模板甚至可以创建为类型，从而保留其与生成的 HMI 对象的直接关系。

由于通过类型生成的 HMI 对象是类型实例，因此适用库的类型/实例概念规则。

说明

类型版本

SiVArc 始终仅使用最新版本的类型。如果项目中的 FC 或 FB 类型的实例不是最新的，SiVArc 将中止生成过程。

在每次 SiVArc 生成之前，更新项目中的所有类型。

如果将类型用作生成模板，您将对特定任务及 HMI 对象应用程序类型进行评估。例如，主副本更适合导航按钮。

在画面规则中使用类型时的规则

使用类型时，以下规则适用：

- 如果使用来自全局库的类型，SiVArc 在生成时将在项目库中生成该类型的副本。
- 只要在类型中编辑 SiVArc 表达式，就需要新的 SiVArc 生成过程。
- 即使在由 SiVArc 生成的类型的实例中，对类型所做的其它更改也会在所用的实例中自动更新.

说明

同时使用类型和实例

如果为项目中某种类型的实例以及类型本身自定义画面规则，SiVArc 会将类型处理两次。确保 SiVArc 会处理实例或类型。

设备相关性

画面对象的可用性和显示尺寸取决于 HMI 设备。创建生成模板时，请注意生成模板适用的设备。创建不同的定位方案并链接您的生成模板，从而针对不同 HMI 设备控制生成 HMI 对象的排布方式。

也可以在生成模板中设置溢出画面以确保对于不同尺寸 HMI 设备进行正确定位。

生成模板的参数化

为了尽量经常使用生成模板，其命名和标记应保证一致。应最大程度地将属性链接至用户程序中适当的使用位置。

此外，在理想情况下，生成模板还要考虑项目中的存储结构以及项目的多语言。为此，您可以使用结构化 SiVArc 对象属性（例如“<Object>.FolderPath”）和可以组态为多语言的表达式（例如“DB.Comment”）。

动态调整大小

始终手动更改画面窗口、面板和文本域的大小。

虽然 SiVArc 支持动态调整大小，但它可能产生不理想的效果，例如，画面对象的重叠。

更改生成模板

要在下次生成期间应用生成模板的更改和优化，应在库中再次存储更改的生成模板。画面规则中将引用生成模板的名称。因此，更新的生成模板必须使用原始生成模板的名称存储在库中。否则，相关画面规则将无效。

优势

针对生成模板的维护和优化将有助于您更为高效地使用 SiVArc。您的 SiVArc 项目将保持灵活性，并且能够轻松适应使用标准化结构及命名约定的其他 STEP 7 用户程序。

例如，此举有助于您在创建用户程序的同时建立 SiVArc 项目。即使是多语项目，您也可通过 SiVArc 在公司内建立并维护标准化程序。

另一方面，生成模板仍具备独立调整能力，足以在 SiVArc 中实施标准化程度较低的项目。

参见

SiVArc 中的生成模板 (页 95)

6.2.8 参数化规则

6.2.8.1 示例：生成统一面板

Unified 画面中的面板

SiVArc 支持 Unified 设备的以下面板属性：

- 变量接口 - 允许组态变量接口名称、数据类型（受变量支持）、用户数据类型（变量支持的 UDT）
- 属性接口 - 允许组态接口属性和数据类型（颜色、资源列表）

要求

- 面板类型的生成模板已存储在库中
- 已在画面规则中组态 Unified 画面

步骤

1. 为 Unified 设备创建面板。有关面板的更多信息，请参见“生成面板 (页 154)”。
2. 向统一面板添加一个按钮。按钮的插件属性会自动更新。
3. 可为面板组态变量接口和属性接口，相同的内容会反映在插件属性区域中。
4. 将包含画面对象的画面规则组态为统一面板。
5. SiVArc 生成时，会以面板容器形式生成统一面板。容器链接到面板类型，并采用库中最新可用的面板类型。在面板组态期间设置的接口属性在生成的统一面板属性中可用。

参见

示例：为面板生成动画 (页 154)

6.2.8.2 示例：创建参数化规则

简介

要通过 SiVArc 自动生成尽量多的 HMI 对象，可使用不同的方法和选项。

示例场景

某家工程公司负责基于完成的用户程序导出生成模板。项目非常广泛，且具有高度标准化的特点。

分析项目后，工程师决定从可视化的模块化用户程序中导出尽量多的文本，并通过 SiVArc 最大程度地减少自定义和扩展工作。

由于用户程序具有模块化结构化和标准化的特点，因此可最大程度地减少 SiVArc 组态数：

- 数量最少的生成模板
- 数量最少的 SiVArc 规则

如果项目的下一扩展处于等待状态，工程公司可通过 SiVArc 项目中的少量调整生成扩展的可视化。

优势

可通过结构化的表达式公式以及一致的指令和 HMI 对象分配创建清晰并且透明的 HMI 项目。可轻松、可靠地实现设备或用户程序中的改动。SiVArc 以此简化经常重复发生的任务。该方法可避免出错。

此外，更便于用户执行企业标准。

6.2.8.3 块和生成模板的分配

简介

用户程序的标准化可在 SiVArc 组态中进行映射。用户程序在结构、模块化和标准化方面形成地越好，SiVArc 组态的应用质量越高。

HMI 对象和功能模块之间的最佳基本关系衍生自用户程序的模块。

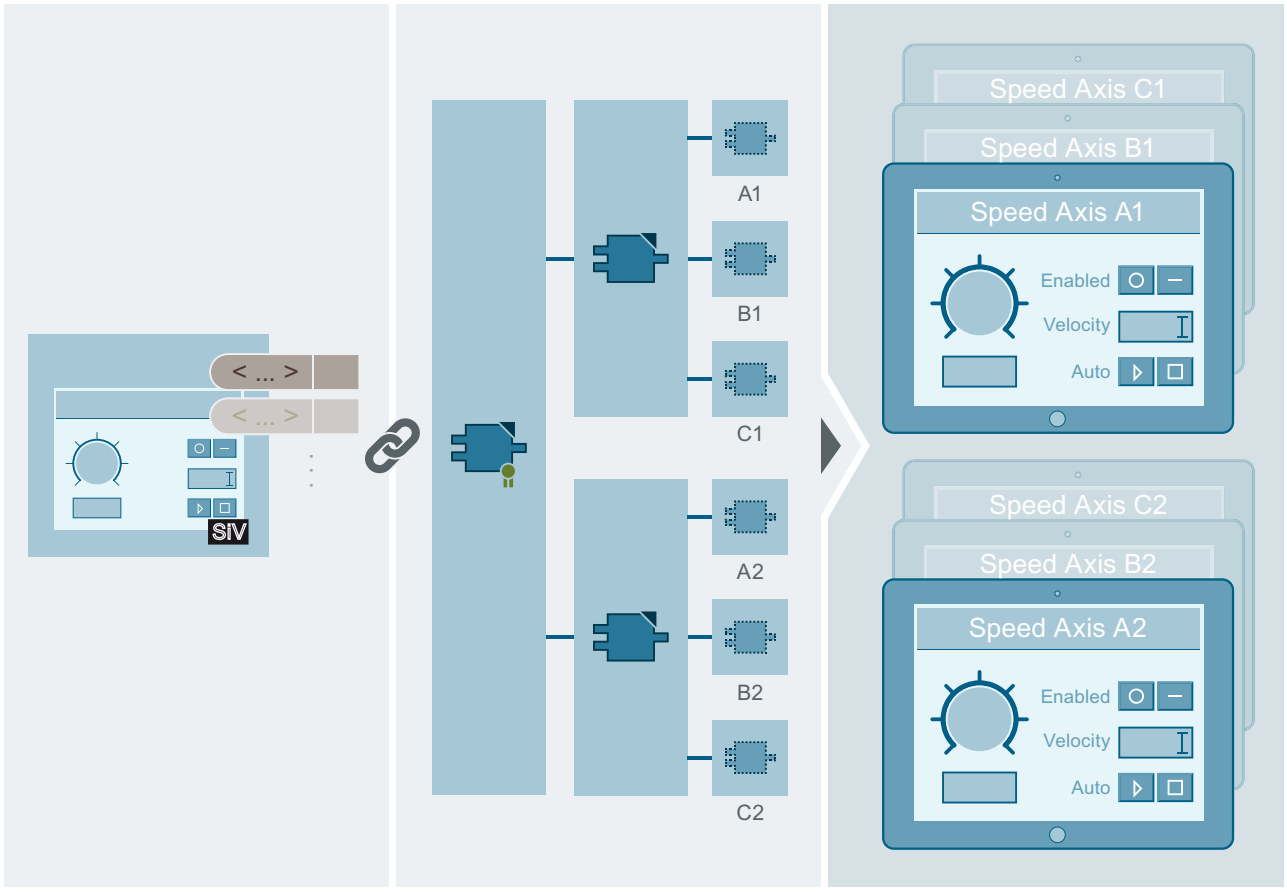
示例场景

用户程序使用相同的指令控制所有虚拟速度轴。该指令作为类型存储在库中。有两种类型实例用于 STEP 7 中，并在用户程序中实现实例化。

SiVArc 规则

以下规则衍生自示例：

- 速度控制轴的控制在面板中进行映射。
- 该面板组态为 SiVArc 的生成模板。
  借助 SiVArc 属性组态名称和互联。
- SiVArc 属性可访问仅在用户程序中相应使用位置处定义的文本，例如，程序段标题中的速度控制轴名称。
- 画面规则可将指令的库类型链接到面板生成模板。
- 生成期间，SiVArc 将在用户程序中运行块类型的所有实例及其所有实例化调用。



SiV SiVArc 生成模板

指令的库类型

STEP 7 中库类型的实例

主 OB 中程序段中的实例化类型实例

结果

通过来自用户程序的文本为库类型实例的每次调用生成面板。

仅通过一个画面规则和一个生成模板即可在项目中实现所有速度轴的可视化。最佳地实现了 HMI 对象和用户程序结构之间的关系。

6.2.8.4 SiVArc 表达式的结构

WinCC 中的 SiVArc 表达式规则

为了确保 WinCC 项目中的唯一性和清晰度，可使用 SiVArc 表达式访问指令实例或程序段标题中的数据块等。这意味着为数据块和程序段标题命名时应牢记唯一性和一致性。

SiVArc 支持组织块的符号表达式。可使用组织块相对寻址定义表达式。表达式 "....SymbolicName"中的点表示在调用层级中，点会进入主程序段，因此会在进入主程序段前解析表达式。如果表达式中的点没有得出或解析出任何值，会显示错误。

SiVArc 支持使用组织块绝对寻址。

例如，下表给出了如何在通过 SiVArc 表达式生成的 HMI 对象中使用函数块的符号名称 "SG01_FB"。

SiVArc 表达式	结果
`"MyBlock"`	MyBlock
`"My\"Block"`	My"Block
`Block.SymbolicName`	SG01_FB
`"MyBlock_"&Block.SymbolicName`	MyBlock_SG01_FB
`"MyBlock_"&Block.SymbolicName&"_An"`	MyBlock_SG01_FB_An

示例：唯一的 HMI 对象名称

要唯一地将已生成且可对在项目中多次使用的过程进行可视化的 HMI 对象分配到过程使用中，可通过路径调用为生成的对象命名。

标记传送带的文本域示例：

- 对象名称：Productionline_Instance_1_Dispatchunit_Instance_1_Conveyor
- 文本：Dispatching

为此，需在对调用层次中前三个调用层次的指令进行寻址的 SiVArc 表达式中使用 SiVArc 对象（关键字）。

示例：文本域对象名称的 SiVArc 表达式：

- ```
  ModuleBlock.DB.SymbolicName&"_"&SubModuleBlock.DB.SymbolicName
  &"_Conveyor
  ```
  ```

调用层级的各个层级的命名规则在用户程序中进行定义。为各个过程显示提供不同的 HMI 对象。

SiVArc 对象	函数类型	指令名称	数据块的 符号名称	生成模板 的名称
StructureBlock	主函数	“Plantsection”	“Plantsection_1_Instanz_1_DB”	Label 01
ModuleBlock	支持函数	“ProductionLine”	“ProductionLine_Instanz_1_DB”	Label 02
SubModuleBlock	标准函数	“DispatchUnit”	“DispatchUnit_Instance_1_DB”	Label 03
块	引用的指令	“Initialize”	“Initialize_Instance_1_DB”	按钮

- 生成的文本域的对象名称的 SiVArc 表达式规则

  ```
  ModuleBlock.DB.SymbolicName&"_"&SubModuleBlock.DB.SymbolicName
  &"_<Name_Generiervorlage>
  ```

- 生成的对象名称
  ProductionLine_Instance_1_DispatchUnit_Instance_1_Label_03
- 生成的标签：
  DispatchUnit
- 标签的 SiVArc 表达式
  `Split("SubModuleBlock.DB.SymbolicName","_" (1)`

示例：唯一的触发变量

为了唯一地互联 HMI 对象中的触发变量，需确保 PLC 变量的名称和同步变量的运行系统设置是一致的。

来自 DB 的符号名称中的变量名称与 PLC 变量的名称均在 WinCC 中形成：

- 第二调用级中 DB 的 PLC 变量名称
  Activate
- 来自 DB 的符号名称
  Plantsection01
- 生成的 HMI 变量名称
  Plantsection01_Activate

相关指令位于第二调用级。

- 变量名称的 SiVArc 表达式
  `StructureBlock.DB.SymbolicName&_Activate`

标签示例

- 如果标签需要具备唯一性，请保持指令名称简短，以便于在按钮上显示，例如：
  - 停止
  - 激活

  用户可在 SiVArc 表达式的生成模板中直接通过指令名称分配标签：
  - SiVArc 表达式：`Block.Title`
- 如果无法采用简短的指令名称，请使用 string 函数：
  - 相关数据块的名称：Plantsection1_DB
  - SiVArc 表达式：`Split(StructureBlock.DB.SymbolicName,"_",0)`
  - 生成的标签：“Plantsection1”

参见

SiVArc 表达式概述 (页 113)

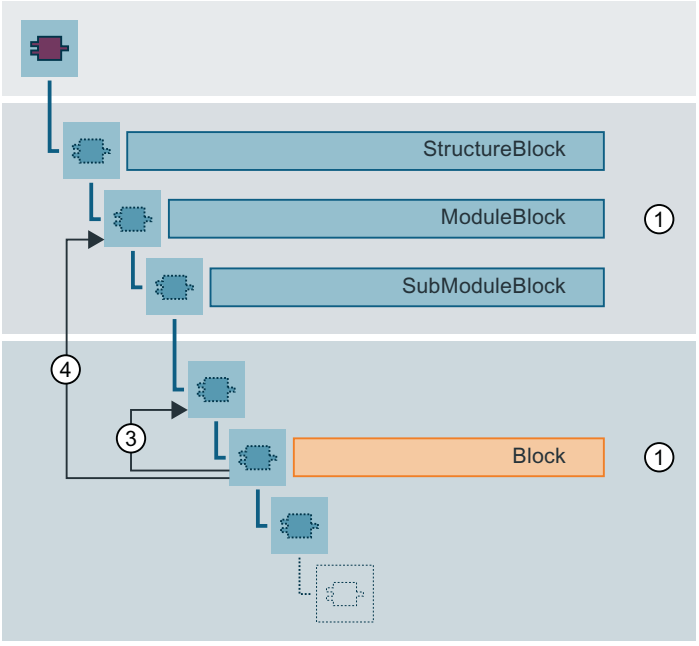
6.2.9 用户程序对生成模板的影响

简介

通过 SiVArc 生成 HMI 对象时，SiVArc 将评估用户程序中程序块的所有调用。用户程序从上到下依次执行。如果程序块中调用了其它程序块，SiVArc 将首先执行较低层级的程序块。

对程序块属性进行寻址

下图显示了程序块的调用层级和程序块属性的访问之间的关系：



前三个优先级的块通过 SiVArc 对象映射。可使用 SiVArc 对象对这些块进行绝对寻址。

SiVArc 对象 `Block` 始终显示 SiVArc 当前正在执行的程序块，无论其位置在调用层级的何处。

在较低的层级中，可对每个更高层级的程序块进行寻址。寻址方法取决于调用层级的当前位置。在该图中，SiVArc 当前正在执行层级的第五级中的程序块。

① 无需 SiVArc 对象，仅通过相对寻址就可到达更高级的块。

在 SiVArc 当前正在执行的块开头，在每个层级前均输入一个点“.”：

在该示例中，按照下面的方法对更高级的块的名称进行寻址：

```
.Name
```

② 可使用 SiVArc 对象访问更高级的块，相对寻址或绝对寻址皆可：

在该示例中，按照下面的方法对第二层级的块进行寻址：

- 相对寻址：`...名称`
- 绝对寻址：`ModuleBlock.Name`

说明

“SiVArc 表达式”编辑器中的使用情况

“SiVArc 表达式”编辑器不支持相对寻址。为实现程序块相对寻址，请在 SiVArc 属性的输入字段中直接输入地址。

示例

在 8 级调用层级中，可通过 FB 寻址层级 8，如下所示：

- 使用 SiVArc 对象对优先级为 1 - 3 的块进行寻址，或进行相对寻址，例如 `StructureBlock.Version` 或 `.......Version`
- 相对寻址调用级别为 4 - 7 的块，例如 `...Version` （层级 5）

使用 SiVArc 对象属性寻址程序块的属性。

访问程序块示例

以下示例显示了在相应调用层级中程序块属性的寻址方法：

示例	标准调用	对调用块进行相对访问	对 1 级调用中更高级别的块进行绝对访问
访问块名称	`Block.Name`	`.Name`	`StructureBlock.Name`
访问 DB 的符号名称	`Block.DB.SymbolicName`	`.DB.SymbolicName`	`StructureBlock.DB.SymbolicName`
访问块参数的值	`Block.Parameters("<Name Parameter>").Value`	`.Parameters("<Name Parameter>").Value`	`StructureBlock.Parameters ("<Name Parameter>").Value`
访问已分配给块参数的变量的注释	`Block.Parameters(<Name Parameter>).AssignedTag.Comment`	`.Parameters(<Name Parameter>).AssignedTag.Comment`	`StructureBlock.Parameters(<Name Parameter>).AssignedTag.Comment`

示例	标准调用	对调用块进行相对访问	对 1 级调用中更高级别的块进行绝对访问
访问寻址块的路径	`Block.FolderPath` `ModuleBlock.FolderPath`	`.FolderPath` 映射调用层级	`StructureBlock.FolderPath`
访问寻址块的背景 DB 的路径 背景 DB 可以是单背景或者多背景的。	`Block.DB.FolderPath` 注意：通过 `DB.FolderPath` 只能引用含数据块的块。	`.DB.FolderPath`	`StructureBlock.DB.FolderPathTagNaming` `.SeparatorChar`

如果使用带有 SiVArc 对象 `Block` 的标准调用，将对当前 SiVArc 表达式正在执行的程序块进行寻址。

参见

SiVArc 对象属性 (页 264)

SiVArc 项目中的文本源概述 (页 104)

6.2.10 多语言对生成模板的影响

项目语言和运行系统语言

用户可优化并高效执行 SiVArc 功能的多语言组态，即使对于生成模板也同样使用。

通过生成模板中的多语言 SiVArc 属性组态多语言属性时，例如，针对每种运行系统语言都生成相关字符串。

TIA Portal 中的语言设置以及含多语言 SiVArc 对象的生成模板均可定义生成对象中要生成哪些多语言文本。

TIA Portal 的语言设置

可以所有项目语言生成一个 SiVArc 项目。为此，请激活所需项目语言作为运行系统语言。

当多语言 SiVArc 对象属性设置为单语言属性时，SiVArc 使用默认的生成语言。默认生成语言取决于 HMI 设备：

- 安装有 RT Advanced 的 HMI 设备
  使用“运行系统设置 > 语言和字体 > 运行系统语言和字体选择”(Runtime settings > Language & font > Runtime language and font selection) 下，列表顶部的运行系统语言
- 安装有 RT Professional 的 HMI 设备
  使用在“运行系统设置 > 语言和字体 > 运行系统语言和字体选择”(Runtime settings > Language & font > Runtime language and font selection) 下被设置为“单语言对象的运行系统语言”(Runtime language for single-language objects) 的运行系统语言

如果某种项目语言未被设置为运行系统语言，则项目中的多语言属性将使用主副本中用于此项目语言的值来生成。在此项目语言中不会评估该属性的 SiVArc 表达式。

如果未在多语言变量中设置值，将生成空字符串作为该语言的属性值。

多语言的 SiVArc 对象

组态多语言的 SiVArc 项目时，需使用下列 SiVArc 对象：

- 多语言属性
- 多语言 SiVArc 对象属性
- 文本列表条目的 SiVArc 文本

多语言 WinCC 属性

SiVArc 支持下列多语言属性。

SiVArc 将为每种运行系统语言分别评估这些属性的表达式。如果表达式中包含多语言 SiVArc 对象属性，那么针对相应运行系统语言的评估将产生不同的结果。

HMI 对象	属性
棒图	标题 工具提示 单位
画面	显示名称
画面窗口	标题
文本域	文本
I/O 域	信息文本
图形 I/O 域	工具提示

HMI 对象	属性
开关	标题 TextOFF TextON 工具提示
圆形按钮	文本 工具提示
按钮	TextOFF
量表	标题 单位

对于其它可使用 SiVArc 表达式的属性，将以默认生成语言评估表达式。

多语言 SiVArc 对象属性

以下 SiVArc 对象属性可组态为多种语言：

- `Title`
- `SymbolComment`
- `DB.Comment`
- `NetworkTitle`
- `NetworkComment`

在多语言环境中使用 SiVArc 表达式

可在 SiVArc 表达式中使用多语言和单语言 SiVArc 对象属性。该参考信息描述了哪些 SiVArc 对象属性为多语言。可在多语言和单语言 SiVArc 属性中使用 SiVArc 对象属性。此时 SiVArc 表达式的评估方式如下：

	单语言 SiVArc 对象属性	多语言 SiVArc 对象属性
单语属性	针对每种运行系统语言都生成相同的字符串。	采用默认的生成语言评估变量。 在 HMI 设备的运行系统设置中指定默认的生成语言。
多语属性	针对每种运行系统语言都生成相同的字符串。	针对所有组态的运行系统语言评估变量。 针对每种运行系统语言都生成组态的字符串。

参见

标题 (页 261)

SymbolComment (页 260)

注释 (页 254)

DB (页 243)

NetworkTitle (页 258)

NetworkComment (页 258)

SiVArc 文本 (页 108)

6.2.11 生成对象的存储策略

概述

SiVArc 可按画面规则或生成模板中的 SiVArc 表达式为画面和变量提供选项以控制生成对象的存储结构。

对此，有各种不同的存储策略：

- 在 STEP 7 的项目树中映射存储结构
- 在项目库中映射存储结构
- 单个存储结构

SiVArc 存储策略是基于画面和变量区域中 HMI 设备下项目树中生成的 HMI 对象。

SiVArc 规则的结构化存储策略为用户提供了 SiVArc 编辑器功能。

应用实例

块存储在项目树中（例如，按功能存储）。该存储形式可针对相关画面自动创建。用户可在画面的生成模板中组态引用项目树中块路径的 SiVArc 表达式。

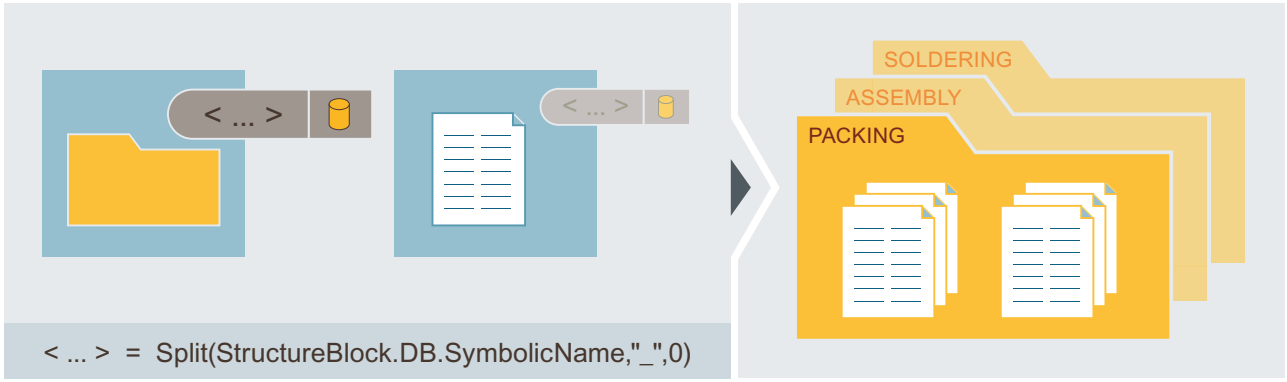
优点

SiVArc 存储策略可提高可视化项目的一致性和标准化程度。如需使用其他存储策略，可毫不费力地重新构建项目。

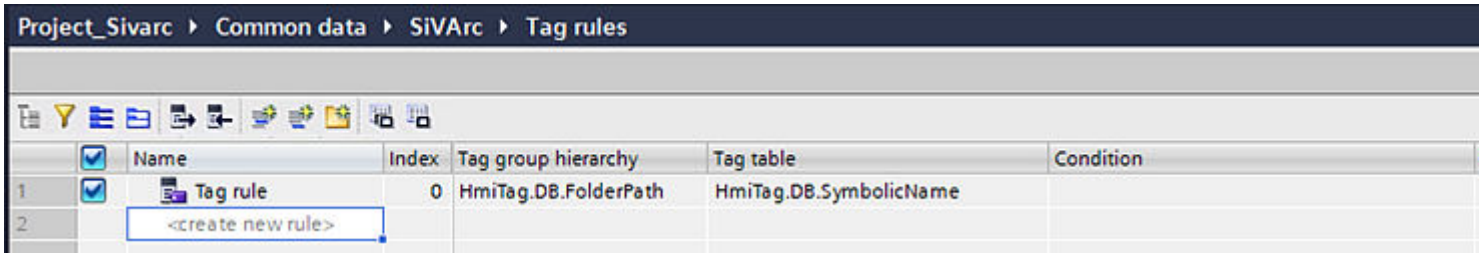
控制生成变量的存储

以下策略可用于变量存储：

- 在 STEP 7 的项目树中映射存储结构
  在“变量规则”(Tag rules) 编辑器中，可以使用 SiVArc 表达式 `HmiTag.DB.SymbolicName` 和 `HmiTag.DB.FolderPath` 基于仅使用一种变量规则的控制程序对变量表进行结构化。
  项目只能在控制器端结构化一次。
- 单个存储结构
  可通过“变量组”(Tag group) 和“变量表”(Tag table) 列分别定义目标文件夹和变量表。



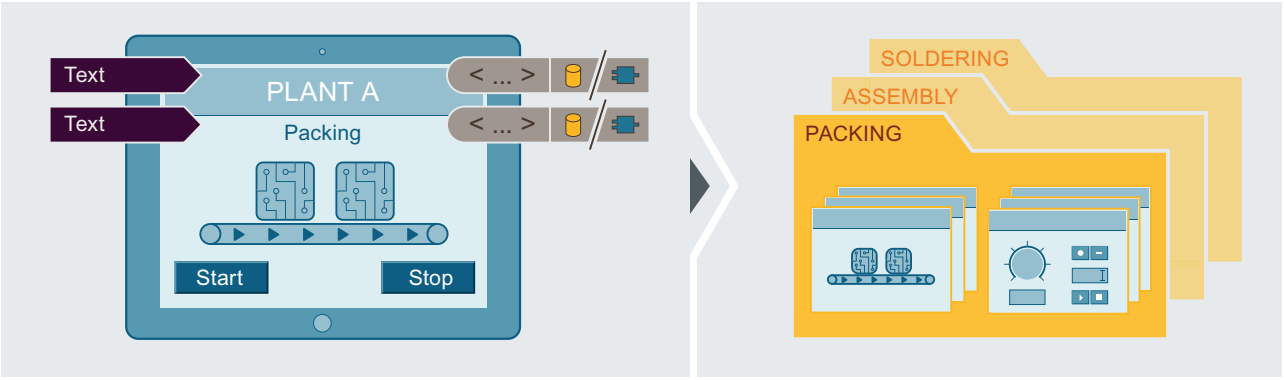
双击项目树中的“公共数据 > SiVArc > 变量规则”(Common data > SiVArc > Tag rules) 可打开“画面规则”(Tag rules) 编辑器。



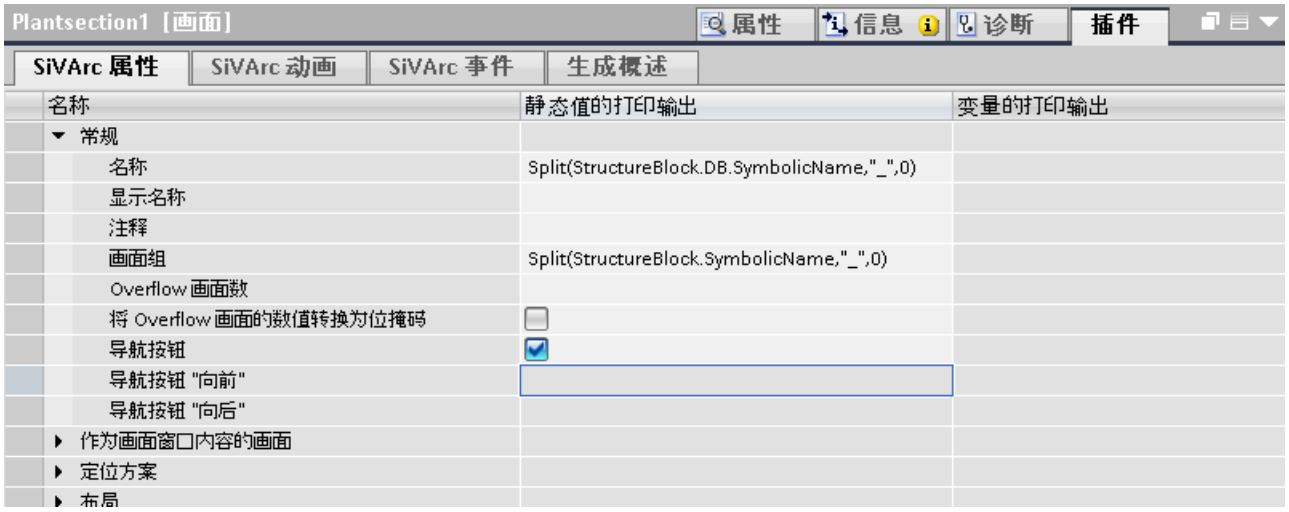
控制生成画面的存储

以下策略可用于画面存储：

- 在 STEP 7 的项目树中映射存储结构
- 在项目库中映射存储结构
- 单个



可通过画面生成模板的“名称”(Name) 和“画面组”(Screen group) 属性控制画面在项目树中的存储。如果在“画面组”(Screen group) 下指定了文本字符串，则系统将按该名称在项目树中创建组。基于生成模板创建的画面也将存储于该位置。



可通过库中某一类型画面的生成模板同步已生成对象的存储结构和命名。

为此，需使用 SiVArc 表达式“LibraryObject.FolderPath”和“LibraryObject.Name”

SiVArc 对象属性	引用的对象	SiVArc 属性
LibraryObject.FolderPath	库中画面类型的存储路径	画面组： 在项目树中生成库的存储路径。 名称*： 在存储画面图形的文件夹后面即为生成画面的名称。
LibraryObject.Name	库的画面类型名称	名称： 画面类型后面即为画面名称。 画面组： 画面存储在项目树中带有画面类型名称的文件夹中。

*) 只有在库中的画面类型仅存储在一个层级中时，才能将 LibraryObject.FolderPath 用于 SiVArc 属性“名称”(Name)。如果想要使用多层存储层级，可以用表达式和 LibraryObject.FolderPath 代替反斜线。

或者，可分别指定存储文件夹和画面名称。

参见

创建变量规则 (页 191)

为画面创建生成模板 (页 165)

名称 (页 257)

FolderPath (页 255)

LibraryObject (页 246)

HMITag (页 245)

6.2.12 示例：实现高灵活性

示例场景

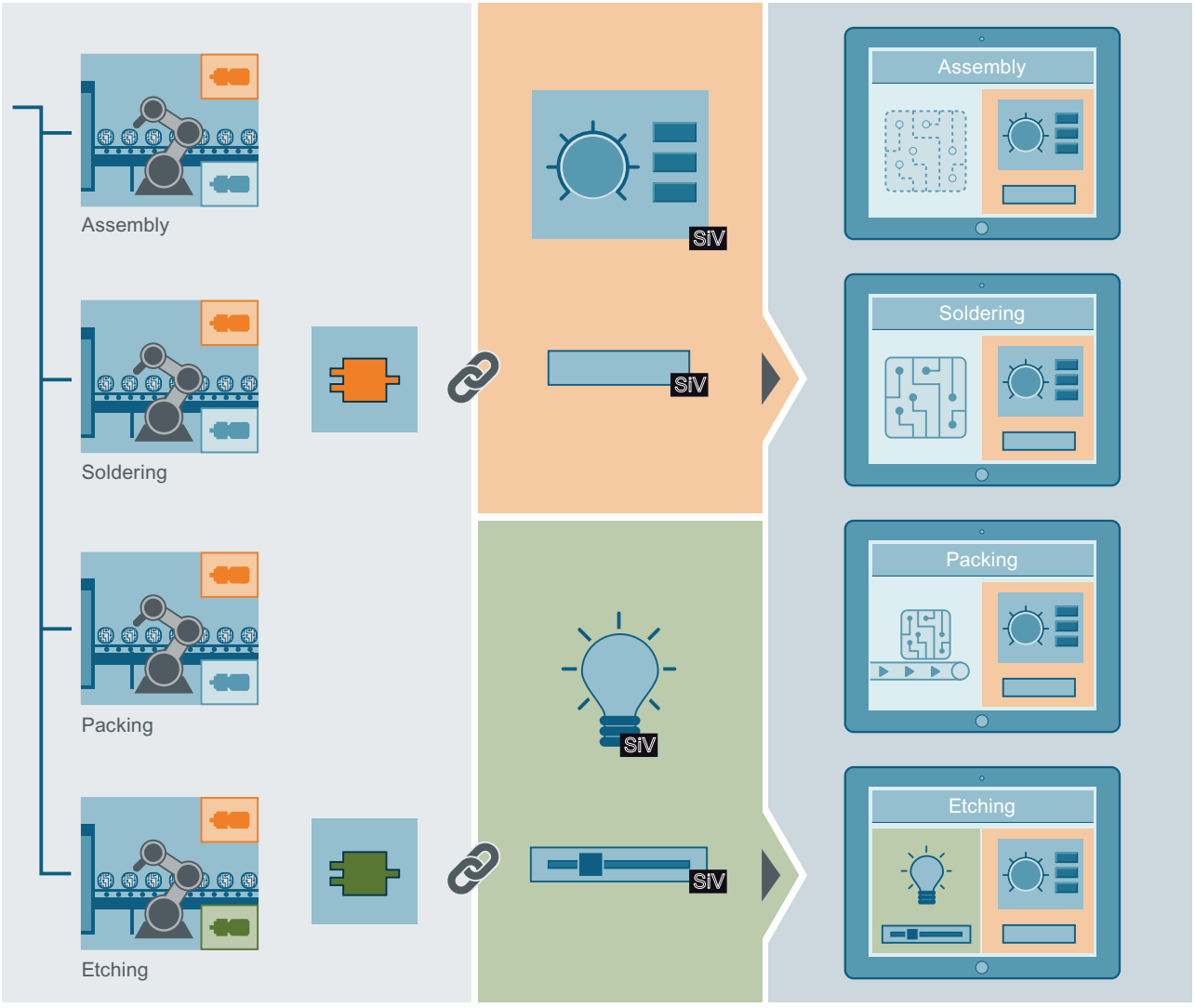
一家印刷电路板工厂分为“装配”、“焊接”、“封装”等部门。生产某种新型电路板需要额外规划并实施一个处理阶段，即“蚀刻”。该工厂主要由多个标准单元构成。

一旦建立了新的工厂部门，将需要对相应的模块运行进行测试和优化。

执行规则

由于“焊接”工厂部门的大部分组态可以重复用于“蚀刻”部门，因此，无需生成额外的 SiVArc 组态。

“焊接”工厂部门所需的额外功能。这些功能是已分配在画面规则中的生成模板的标准功能。工程师们将“蚀刻”生产阶段所需的额外规则归为一组，并启用相关的 HMI 设备。



函数块“转速”(Rotational speed)

函数块“蚀刻”(Etching)

为了对工厂进行测试，项目工程师共同启用了一个画面规则组并禁用了测试不需要使用的模块。可视化工程师针对生成的用户界面进行测试。根据测试结果，通过使用 SiVArc 属性条件或对属性进行修，优化了生成模板和规则。

6.2.13 示例：实现高可复用性

示例场景

某工程公司接到一笔新客户订单，要组态家标准工厂用于制造印刷电路板。

该公司以拥有针对 PCB 生产优化过的 SiVArc 项目，并希望将其重新用于新客户。

方案旨在最大限度确保相同功能的操作和可视化一致性，所包括功能如下：

- ON/OFF
- 行进至基本位置
- 显示状态

执行方案

标准化用户程序包含很多函数块和标准函数。系统块创建为库类型。因此该工程公司可为各标准函数设置现有完整生成模板集。

现有标准函数生成模板可通过 SiVArc 表达式直接访问标准块上的文本源。不考虑调用层级。触发变量通过系统块中数据块的名称进行唯一引用。每次对类型进行复用都会根据相同的生成模板集在可视化中生成相关操作元素。因此无需调整。

生成模板的颜色和形式可根据操作规则调整并获取自项目特定库。

操作画面的新企业设计与生成模板通过定位方案连接。

6.2.14 示例：为画面窗口创建生成模板

示例场景

出于培训目的，将在采用冗余设计的工作位置复制多个现有 HMI 设备。

目的

在操作员站生成多个带有对应起始画面的画面窗口。画面窗口的名称可指示自身可视化的程序块。

SiVArc 中的画面窗口

画面窗口未直接作为画面对象而生成。而是在将画面指定为画面对象时隐式创建的。

如果项目中有"DefaultScreenWindowControl"生成模板，SiVArc 将根据该模板生成画面窗口。如果不存在该模板，则 SiVArc 将通过工具箱创建画面窗口副本。

要求

- 将冗余工作位置的空白操作员画面存储为名称是"Screen_Training"的画面窗口生成模板。生成模板的 SiVArc 属性“名称”(Name) 由 SiVArc 表达式 “`Block.DB.SymbolicName&"_SWC`”组态。
- "Plantsection_Soldering"程序块包含在用户程序中，并在 OB1 中反复调用。

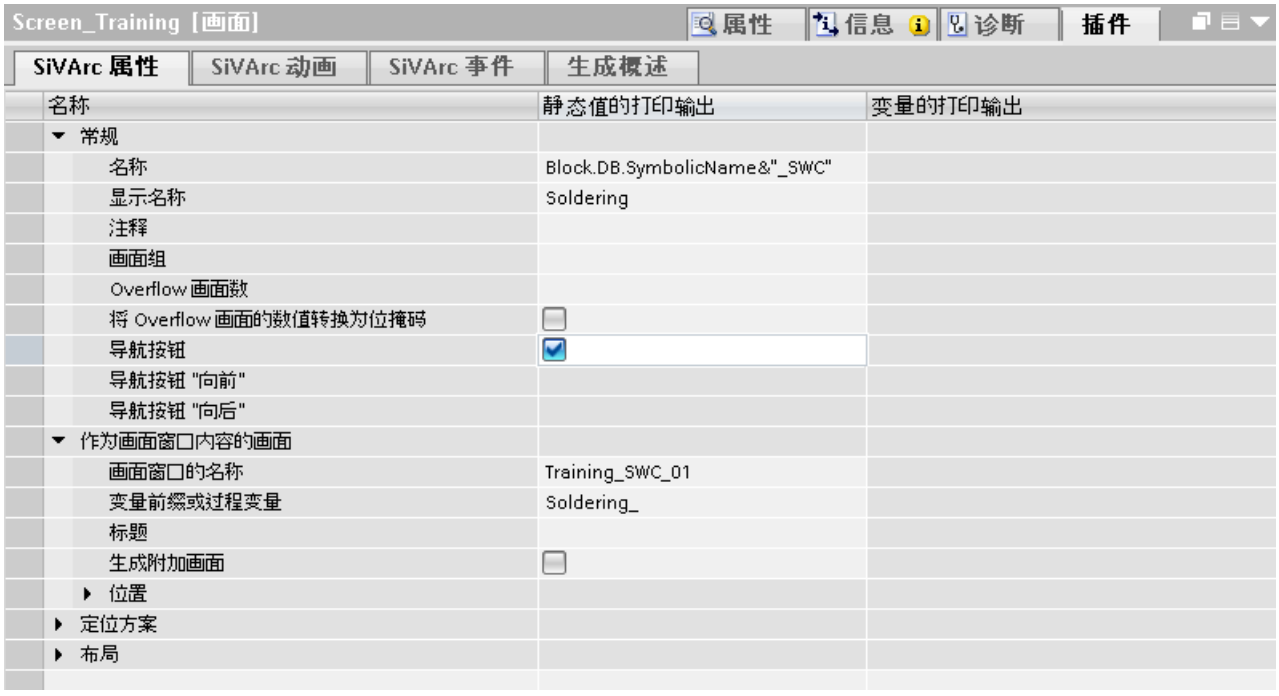
步骤

要为画面窗口创建生成模板副本，请按以下步骤操作：

1. 打开库中的生成模板"Screen_Training"。
2. 在“插件 > SiVArc 属性”(Plug-ins > SiVArc properties) 下通过 SiVArc 表达式 “`Block.DB.SymbolicName&"_SWC`”组态画面窗口的名称。
   `Block.DB.SymbolicName` 部分根据块调用的类型引用以下名称中的一个：
   - 全局：背景数据块的符号名称
   - 本地：块实例的名称

   `&"-SWC"` 部分用于为名称添加后缀，表示“画面窗口控制”(SWC)。

3. 在巡视窗口的“插件 > SiVArc 属性 > 作为画面窗口内容的画面”(Plug-ins > SiVArc properties > Screen as content of the screen window) 下组态所需的属性：
   – 在“画面窗口的名称”(Name of the screen window) 下，输入一个唯一的名称或者 SiVArc 表达式作为面向目标画面生成的画面窗口。
   – 在“变量前缀”(Tag prefix) 下，输入使用用户数据类型的变量的名称。如有必要，使用 SiVArc 表达式。



4. 将“Station_Training”作为新画面规则的名称输入。
5. 在注释中输入“仅限培训使用”。
6. 选择中央程序块“Plantsection_Soldering”。
7. 在“画面对象”(Screen object) 下，选择“Startscreen”生成模板。
8. 在“画面”(Screen) 下，选择“Screen_Training”生成模板。

结果

每次调用“Plantsection_Soldering”程序块时都将生成带有画面窗口的“Screen_Training”画面。“Soldering”设备的起始画面含在画面窗口内。

画面窗口的名称包含指向自身所可视化程序块的引用并且后缀为 -SWC，例如“Plantsection_Soldering_Instance01_SWC”。

多个画面的画面窗口

要显示生成的画面窗口中的其他画面（例如，诊断画面），请将所需生成模板放入同一库文件夹，例如“Training_Screens”。还要组态以下 SiVArc 属性：

- 在已选作“画面对象”(Screen object) 的画面中组态 SiVArc 属性“生成其它画面”(Generate additional screens)。
- 在已选作“画面对象”(Screen object) 的画面中组态 SiVArc 属性“画面窗口中的画面”(Screen in screen window)。

参见

示例：使用复制规则生成模板画面 (页 139)

6.2.15 示例：使用复制规则生成模板画面

示例：使用复制规则生成模板画面

出于培训目的，请考虑在汽车制造厂中组装零件的程序，其中 PLC 将“Assemble_body”作为主程序段，“Production_time”和“Production_order”作为函数块。

目的

在汽车制造厂，工程部门收到合同，在 SiVArc 生成过程中将单个模板画面应用于所有 HMI 设备。复制规则功能复制、创建模板画面的多个副本，并使用用户定义的 SiVArc 表达式应用于 PLC 中组态的所有块。因此，复制规则优化并通过模板画面提高制造厂的生产力。

要使用复制规则生成名称为“Production_order”的模板画面，请按照下述步骤进行操作。

要求

- 使用 PLC 块组态的 HMI 设备。
- 模板画面已组态。

步骤

要使用复制规则创建模板画面，请按以下步骤操作：

1. 在“模板 > 添加新模板”(Templates > Add new template) 下创建新模板。
2. 在模板画面的“属性”(Properties) 窗口中，组态名为“Production_order”的模板。
3. 在“SiVArc > 复制规则”(SiVArc > Copy rules) 下，将模板画面放在“主副本”文件夹中。
4. 单击“创建新规则”(create new rule)。
5. 输入复制规则名称。
6. 在“库对象”下，从“主副本”文件夹中浏览模板。库对象支持智能提示。
7. 在 HMI 设备上生成 SiVArc 可视化。

结果

在“模板”文件夹下，生成名称为“Production_order" 的模板画面。现有的模板画面名称后缀为 "_Renamed”。

参见

SiVArc 中的生成模板 (页 95)

6.2.16 示例：使用模板属性生成 HMI 画面

示例：使用模板属性生成 HMI 画面

出于培训目的，请考虑在汽车制造厂中组装零件的程序，其中 PLC 将“Assemble_body”作为主程序段，“Production_time”和“Production_order”作为函数块，“Production_order". ”作为模板画面。

目的

在汽车制造厂中，当必须使用模板生成多个 HMI 画面时，将使用模板属性。模板是使用 SiVArc 属性添加的，并使用用户定义的 SiVArc 表达式应用于 PLC 中组态的所有块。因此，可以提高多个 HMI 设备的生产力和可扩展性。

要通过模板属性“Production_order”生成 HMI 画面，请按照以下步骤进行操作。

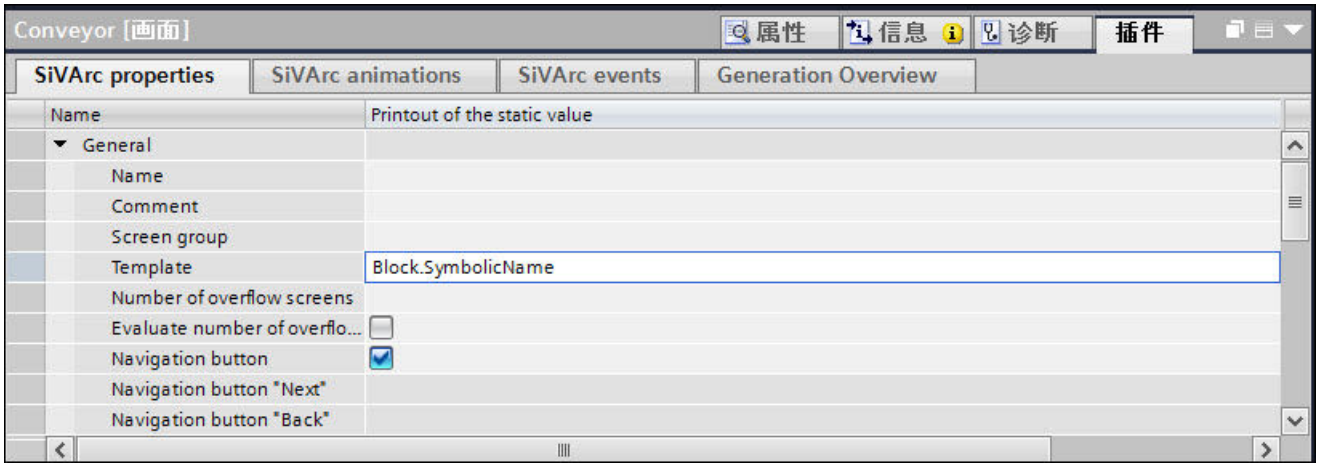
要求

- 使用 PLC 块组态的 HMI 设备。
- 模板画面已组态。

步骤

要使用模板属性生成 HMI 画面，请按照下列步骤操作：

1. 在“画面 > 添加新画面 > Conveyor”(Screens > Add new screen > Conveyor) 下。
2. 在“插件 > SiVArc 属性”(Plug-ins > SiVArc properties) 下通过 SiVArc 表达式“`Block.SymbolicName`”组态画面的名称。
3. `Block.SymbolicName` 部分根据块调用的类型引用以下名称中的一个：
   - 全局：块的符号名
   - 本地：块实例的名称
4. 在巡视窗口的“插件 > SiVArc 属性 > 模板”(Plug-ins > SiVArc properties > Template) 下，您可以输入模板名称或组态 SiVArc 表达式，使表达式解析为项目库中现有的模板名称。解析表达式名称将显示在目标画面中。

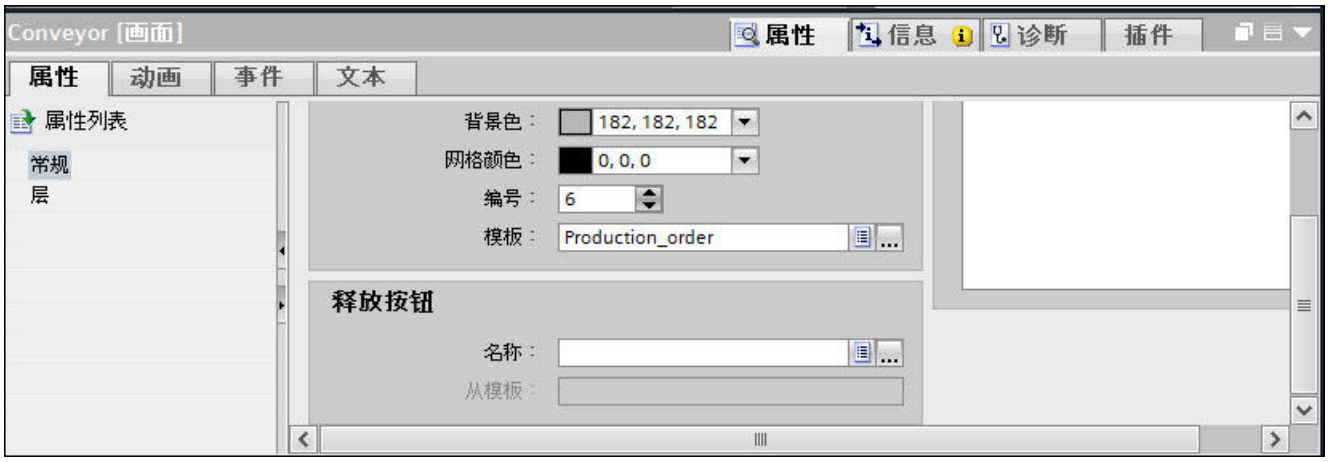


5. 将组态的 HMI 画面放在“主副本”文件夹中。
6. 在“SiVArc > 画面规则”(SiVArc > Screen rules) 下，单击“创建新规则”(Create new rule)。输入画面规则的名称，如“传送规则”。
7. 在“库对象”下，从“主副本”文件夹中浏览 PLC 块和画面模板。库对象和程序块支持智能提示。
8. 在 HMI 设备上生成 SiVArc 可视化。

结果

生成模板名称为“Production order”的 HMI 画面“Conveyor”。旧的模板画面名称后缀为 "_Renamed”。

以下截图显示了在 SiVArc 生成后自动应用的模板名称。



参见

SiVArc 中的生成模板 (页 95)

6.2.17 示例：创建带有动画的生成模板

示例场景

机器人移动至基本位置时，机器人画面应始终闪烁并改变颜色。

目的

使组态了设计动画的图形 I/O 域成为机器人的生成模板。状态规范遵循 SiVArc 表达式。

SiVArc 动画

SiVArc 支持以下几种类型的动画：

- 具有变量连接的动画（仅在 WinCC Runtime Professional 中提供，用于 S7-GRAPH 总览）
- “显示”类别的动画

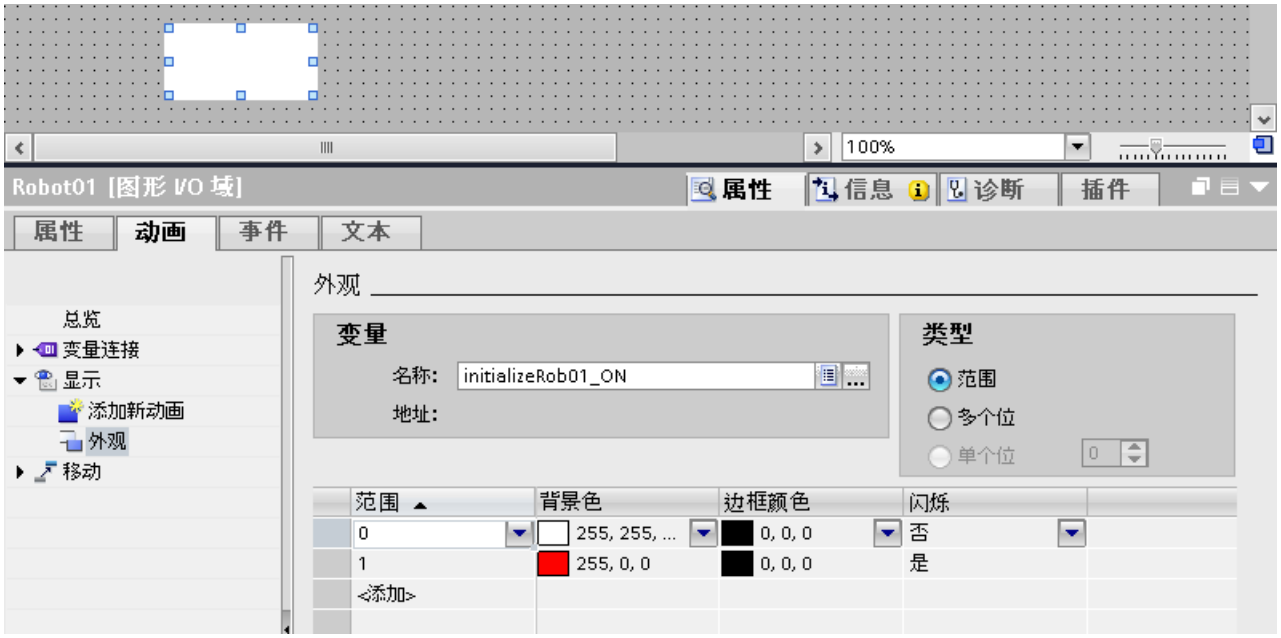
对于这些动画，可使用 SiVArc 表达式定义在运行系统中触发动画的过程变量。

要求

- 为“Robot01”图形 I/O 域组态为生成模板，以显示机器人。
- PLC 变量与 HMI 变量同步。
- “initializeRob01_ON”变量包含行进至基本位置的状态信息，并连接至外部变量“`Soldering_Instance_01_initializePosRob01`”。
- “Rob01”程序块包含在用户程序中。
- 系统将创建用于将图形 I/O 域“Robot01”连接至程序块“Rob01”的画面规则。

步骤

1. 打开图形 I/O 域的生成模板。
2. 组态设计动画。



   - 选择“区域”(Area) 类型。
   - 针对“1”区域，选择红色作为背景色并启用“闪烁”(Flashing) 选项。
3. 打开“属性 > 插件”(Properties > Plug-ins) 下的“SiVArc 动画”选项卡。
4. 针对“变量表达式”(Tag expression) 下的“外观”(Appearance) 动画，组态 SiVArc 表达式“`StructureBlock.DB.SymbolicName&"_initialPosRob01"`”
5. 覆盖库中的现有生成模板。

结果

生成模板针对每个“Rob01”程序块实例创建“Robot01”图形 I/O 域。每个图形 I/O 域均组态有动画。当机器人在运行系统中移动至基础位置时，机器人画面闪烁并显示红色。

6.2.18 示例：创建带有事件组态的生成模板

示例场景

“Activate”按钮应触发铣削/焊接或定位机器人的行进至基本位置。

目的

在“激活”(Activate) 按钮的生成模板中，将“单击”事件组态为“SetBit”系统函数。

生成期间，系统函数的唯一“变量”(Tag) 参数由来自 STEP 7 的文本源组成。

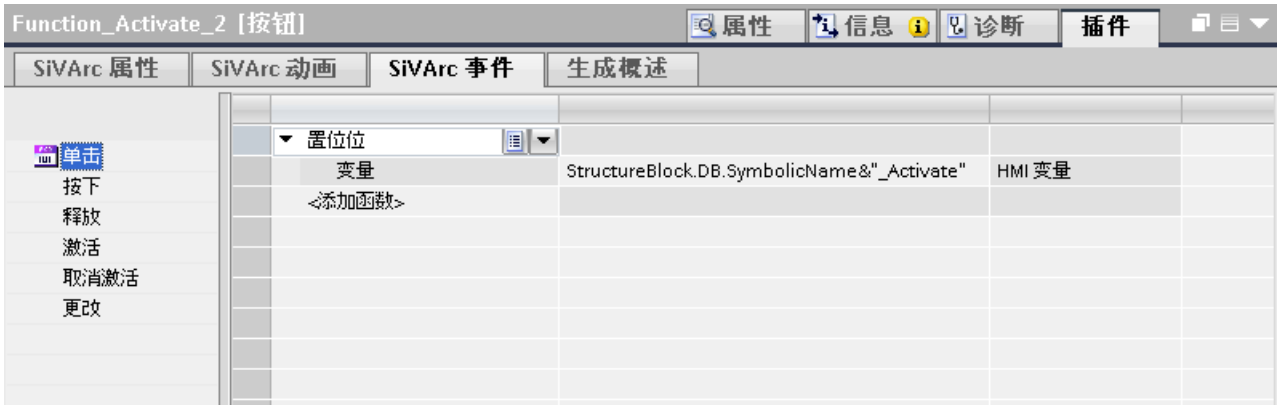
要求

- 为“Activate”按钮的生成模板组态“SetBit”系统函数。
- 系统将画面规则，以将“Activate”生成模板链接至相关函数块。
  本示例中的函数块位于调用层级的第二级并通过 SiVArc 对象“`StructureBlock`”寻址。

步骤

要创建带有事件组态的生成模板，请按以下步骤操作：

1. 在 WinCC 中打开生成模板“Activate”按钮。
2. 在巡视窗口的“插件 > SiVArc 事件 > 单击”(Plug-ins > SiVArc events > Click) 下，将 SiVArc 表达式“`StructureBlock.DB.SymbolicName&"_Activate`”组态为变量。



3. 覆盖库中的现有生成模板。

结果

生成可在每次调用相关函数块时触发并退出函数的按钮。实现了变量与所有实例的互连。

WinCC Unified 设备的事件

- 这些事件支持画面或画面项的脚本功能。脚本功能参数组态非强制要求。如果该功能参数未组态，则系统默认此参数类型为“无”。
- 可通过“LCID 参数”中语言 ID 或语言名称，对“`SetLanguage`”参数进行组态。例如，语言 ID 为“1033”；语言名称为“en-US”。
- “SelectionType”的参数值需进行手动组态，否则 SiVArc 将显示一条错误消息。
- 如果系统由于值无效而显示警告消息，则工程组态系统将不执行任何更改。

6.2.19 示例：创建带有脚本组态的生成模板

示例场景

在 SiVArc 示例项目中，除单位是华氏度的值以外，温度读数应始终以摄氏度为单位输出。

执行规则

要切换其他显示对象，需在各项目中生成带有相应脚本的按钮。

如果是无需转换的项目，SiVArc 组态工程师将在下次生成时禁用相应画面规则或将画面规则限制为选定的 HMI 设备。

系统函数和脚本处理的可用性

将脚本连接到事件时，这些脚本必须存在于每个目标设备上。如果组态的脚本不存在于目标设备的“脚本”(Scripts) 编辑器中，那么显示画面和操作对象会在未连接此脚本的条件下生成。

SiVArc 支持在发生所有画面和画面对象事件时通过 SiVArc 表达式组态系统函数和脚本。
SiVArc 支持以下类别的系统函数：

- 计算
- 位处理
- 画面

SiVArc 为面板提供有限的 SiVArc 事件和系统函数支持。您可在“参考”部分找到支持的系统函数的概述。

说明

设备相关性

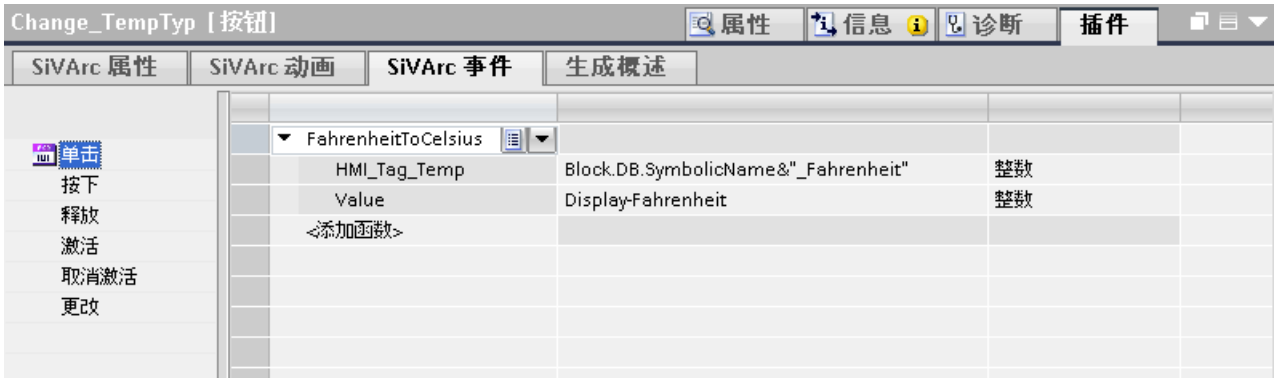
显示和操作对象中事件的数量和类型取决于组态的 HMI 设备。
有关事件的设备相关性的更多信息，请参见 TIA Portal 在线帮助中的“使用系统函数和运行系统脚本”部分。

要求

- 完成“FahrenheitToCelsius”脚本的编程，该脚本可将华氏度转换为摄氏度并在 I/O 域中结果输出。
- 该脚本拥有参数“HMI_Tag_Temp”和“Value”。
- 所有目标设备上都将创建该脚本。
- 该按钮以“Change_TempTyp”生成模板的形式创建并画面规则链接至相关的温度测定函数。

步骤

1. 在 WinCC 中打开“Change_TempTyp”按钮的生成模板。
2. 在巡视窗口的“插件 > SiVArc 事件”(Plug-ins > SiVArc events) 下为“单击”事件组态“FahrenheitToCelsius”脚本。
3. 通过 SiVArc 表达式“`Block.DB.SymbolicName&"_Fahrenheit`”组态“HMI_Tag_Temp”参数。
4. 按输出字段“Display_Fahrenheit”的名称组态“值”(Value) 参数。



5. 覆盖库中现有的“Change_TempTyp”生成模板。
6. 为“Display_Fahrenheit”I/O 域创建画面规则。

结果

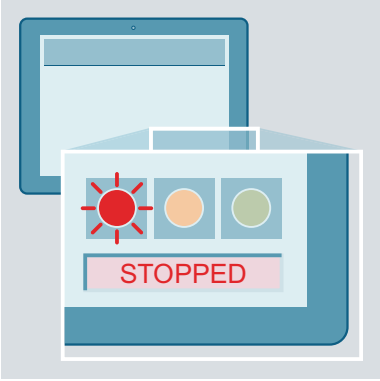
可视化生成后，每次调用相关函数块时系统都将创建“Change_TempTyp”按钮和“Display_Fahrenheit”I/O 域。

“FahrenheitToCelsius”脚本链接至“Change_TempTyp”按钮。在运行系统中，各函数实例“Fahrenheit”变量相应的转换值显示于“Display_Fahrenheit”I/O 域。

6.2.20 示例：为文本列表创建生成模板

示例场景

交通灯指示工厂状态。每种颜色分配一个状态文本并通过交通灯旁的符号 I/O 域进行显示。



目的

通过库提供文本列表的生成模板。所需文本定义将在网络的用户程序中进行维护。

文本列表的生成模板分配了动态触发变量。相关函数的数据块称为“Plantsection1_DB”。文本列表名称应指向块符号名称的第一部分：“Plantsection1_Textlist”。通过“Split”函数，在 SiVArc 表达式中将文本列表名称简化为“_DB”

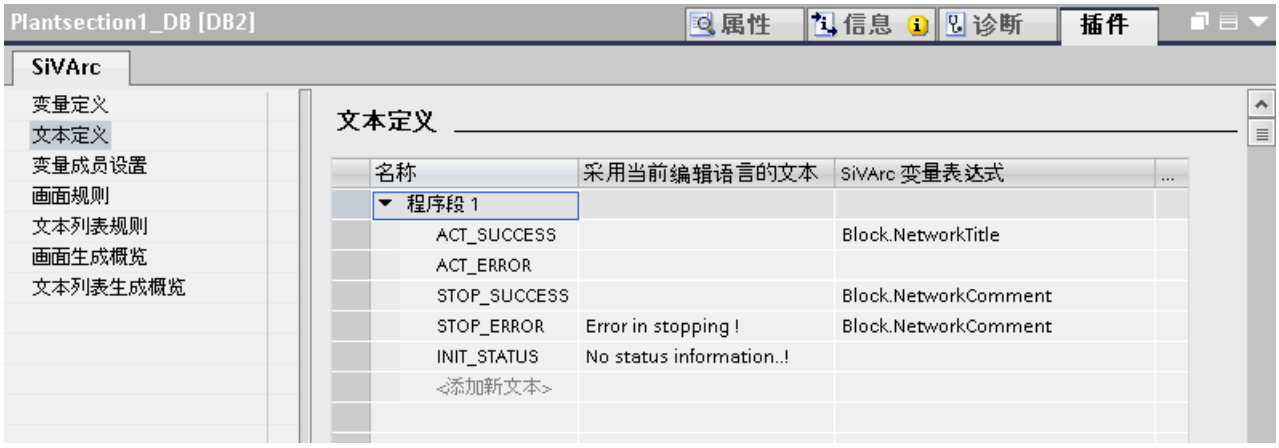
要求

- “Textlist_State”生成模板已在库中存储。
- 创建一项列表规则，将“Textlist_State”生成模板链接至“Plantsection”函数快。

定义文本列表条目

为了创建文本定义，请按以下步骤操作：

1. 根据需要在用户程序中选择文本列表条目需要定义的网路。
2. 在“插件 > SiVArc”(Plug-ins > SiVArc) 下选择“文本定义”(Text definitions) 类别。
   以下文本定义将在首个“Plantsection”实例网络的用户程序中进行定义。



3. 在“名称 > 网络”(Name > Network) 下输入文本列表条目名称。
4. 在“当前编辑语言中的文本”(Text in current editing language) 下输入静态文本列表条目。
   如果未指定动态文本，SiVArc 生成静态文本。
5. 在“SiVArc 变量表达式”(Expression of the SiVArc tag) 下输入 SiVArc 表达式，从而动态分配文本列表条目。在示例的生成过程中，
   SiVArc 使用在文本列表规则下链接的函数块网络标题 (`Block.NetworkTitle`) 和网络注释 (`Block.NetworkComment`)。
   可通过 SiVArc 表达式组态文本定义，表达式会解析出文本列表名称（若文本列表放入主副本中）。不能生成名称与默认文本列表名称相似的文本列表。

步骤

要为文本列表创建生成模板，请按以下步骤操作：

1. 打开库中的“Textlist_State”生成模板。
2. 选择“区域”(Area) 文本列表类型。
3. 打开文本列表的文本列表条目。
4. 将用户程序中的文本定义名称复制到“名称”(Name) 列的顺序值中。
5. 输入默认文本列表条目。

6. 在巡视窗口的“插件 > SiVArc 属性”(Plug-ins > SiVArc properties) 下，通过 SiVArc 表达式 “Split(StructureBlock.DB.SymbolicName,"_",0)&"_Textlist”组态文本列表名称。

7. 覆盖库中的现有生成模板。

结果

在生成过程中，将为“Plantsection”功能块的第一个实例创建文本列表。使用“Split”函数和数据块的名称生成“Plantsection1_Textlist”文本列表。

为了另外生成文本列表以作为块的其它用途，在用户程序中块的所有使用点中输入文本定义。

如果无法评估条目，则根据 SiVArc 主副本生成文本列表。

如果在生成过程中检测到多个相同 SiVArc 文本名称，SiVArc 使用最近创建的 SiVArc 文本。

对于 Unified 设备，可为 PLC 组态包含多语言注释的变量表，并可同时组态作为主副本的文本列表，系统会比较包含多语言注释的变量表和函数块中的变量表。匹配条目在新创建的 HMI 文本列表中可用

注意事项

- 对于 WinCC Unified 设备，可为 PLC 组态包含多语言注释的变量表，并可同时组态作为主副本的文本列表，系统会比较包含多语言注释的变量表和函数块中的变量表。匹配条目在新创建的 HMI 文本列表中可用。
- PLC 连接到 WinCC Unified 和 Runtume Advanced 设备，可在 WinCC Unified 设备上创建文本列表，并将文本列表添加到主副本。在文本列表规则创建过程中，如果为 WinCC Unified 和 Runtime Advanced 设备选择设备特定的列，则 SiVArc 生成时，仅会在 Unified 设备上生成文本列表，并会在 Runtime Advanced 设备上显示错误。这一点同样适用于复制规则。

6.2.21 示例：为块参数的文本列表创建生成模板

示例场景

系统将传送带的设备状态持续输出到操作员画面中。

目的

使 SiVArc 生成与文本列表互连的 I/O 域，并且条目来自“Conveyor”函数块的“State_A”块输出。

要求

- “Textlist_Parameter”生成模板已存储在库中。
- 已创建可将“Textlist_Parameter”生成模板与“Conveyor”函数块互连的文本列表规则。
- “State_A”块输出时，传送带状态已包含在变量注释中：
  - OFF
  - ERROR
  - STOP
  - RUN

步骤

要为块参数的文本列表创建生成模板，请按以下步骤操作：

1. 打开库中的“Textlist_Parameter”生成模板。
2. 在 SiVArc 属性中激活“使用块参数和相关 PLC 变量”(Use block parameters and relevant PLC tags)。
3. 输入参数名称“State_A”和 I/O 类型“输出”(Output)。
   要选择多个参数，请使用包含星号的正则表达式。系统随后会按照指定的方式评估名称中包含此字符串的所有参数。
4. 选择要用于文本列表生成的数据类型（“BOOL”）和变量数（“4”）。
   例如，如果选择数字“17”，则处理前 17 个变量。如果只有 15 个变量，则只处理这 15 个变量。
5. 覆盖库中的现有生成模板。

结果

生成过程中会记录和评估已组态的数据类型的变量。在以下每种情况下，都会为这些变量中的四个创建一个文本列表条目：：

- 文本列表条目对应于变量的相应注释。
- 文本列表条目的名称包含参数名称、参数的数据类型以及一个序列号，例如，State_A_Bit _1、State_A_Bit _2 等。

如果符号表中不含变量名称，则会创建已组态数量的文本列表条目并分配值和名称。随后，可从参数中获取文本条目的名称。此时，可在变量注释中补充所需文本列表条目，然后再次生成。如果是手动输入文本条目，则文本将仅保留至下一生成。

6.2.22 示例：生成弹出画面及弹出画面的使用

示例场景

在操作员画面中，空间只能容纳过程控制必要的所有显示和操作元素。因此，用于进行语言切换的对话框改为在弹出画面中显示。

执行规则

弹出画面的调用在包含语言切换设置的按钮上组态。

在 SiVArc 中使用弹出画面

可按其他 WinCC 画面的使用方式将弹出画面与 SiVArc 配合使用。要为弹出画面应用单独的定位方案，请使用基于弹出画面创建的定位方案。

要在弹出画面上生成显示和操作对象，请按画面规则将弹出画面用作“画面的主副本”。

要求

- 以下生成模板已存储在库中：
  - 弹出画面“PopUp_ChangeLang”
  - 按钮“Button_PopUp_ChangeLang”
  - 起始画面“StartScreen”
- 库中已创建一个弹出画面作为“PopUp_Pos_ChangeLang”定位方案。
- 将为以下 HMI 对象创建画面规则：
  - 按钮“Button_PopUp_ChangeLang”
  - 弹出画面“Button_PopUp_ChangeLang”

为弹出画面创建生成模板

要创建调用弹出画面所需的生成模板，请按以下步骤操作：

1. 打开库中弹出画面“PopUp_ChangeLang”的生成模板。
2. 在巡视窗口的“插件 > 属性 > 常规”(Plug-ins > SiVArc properties > General) 下组态以下 SiVArc 属性：
   - 要生成唯一的画面名称，在“名称”(Name) 下输入一个 SiVArc 表达式或字符串。将调用的程序块的名称与“`Block.DB.SymbolicName&"_PopUp`”整合，作为弹出画面的名称。
   - 如果生成的画面将存储在组或者工厂结构中，则在“画面组”(Screen group) 下输入一个 SiVArc 表达式。
   - 选择“固定”(Fixed) 作为定位方案的模式和“PopUp_Pos_ChangeLang”定位方案。
3. 覆盖库中的现有生成模板。

为调用按钮创建生成模板

要创建调用弹出画面所需的生成模板，请按以下步骤操作：

1. 打开库中调用按钮“Button_PopUp_ChangeLang”的生成模板。
2. 在巡视窗口的“插件 > SiVArc 属性”(Plug-ins > SiVArc properties) 下组态所需的 SiVArc 属性。将调用的程序块的名称与“`Block.SymbolicName&"_ButtonPopUp`”整合，作为按钮的名称。

3. 在巡视窗口的“插件 > SiVArc 事件”(Plug-ins > SiVArc events) 下，为“单击”事件组态系统函数“ShowPopupScreen”等。
   - 对于“画面的名称”(Name of the screen) 参数，在“插件 > SiVArc 属性 > 常规 > 名称”(Plug-ins > SiVArc Properties > General > Name) 下分配弹出画面生成模板中组态的 SiVArc 表达式：“`Block.DB.SymbolicName&"_PopUp"`”
   - 为弹出画面的显示位置坐标组态一个整数值。
   - 选择显示值。

4. 覆盖库中的现有生成模板。

结果

设备的起始画面、语言切换按钮、弹出画面和中央函数模块在画面规则中链接在一起。生成后，设备区起始画面中将生成其他按钮，用于调用语言切换弹出窗口。

6.2.23 示例：为面板生成动画

示例场景

设计一条用于重型负载的制造工厂装配线，并且仅用于特殊包装形式。因此两轴的速度控制界面应仅在生产线运行时显示于设备画面。

执行规则

速度控制的可视化在面板中进行准备。该面板在项目中用作控制所有速度控制轴的生成模板。可视化工程师根据面板新建带有可视动画的生成模板。

在画面规则中，利用各种条件控制何时生成带有动画的面板。

SiVArc 中以动画显示的面板

SiVArc 支持面板的以下动画：

- 可视性
- 允许操作员控制
- 外观

若要使用 SiVArc 为面板生成动画，请在用作生成模板的面板类型中组态动画的动态属性。

要求

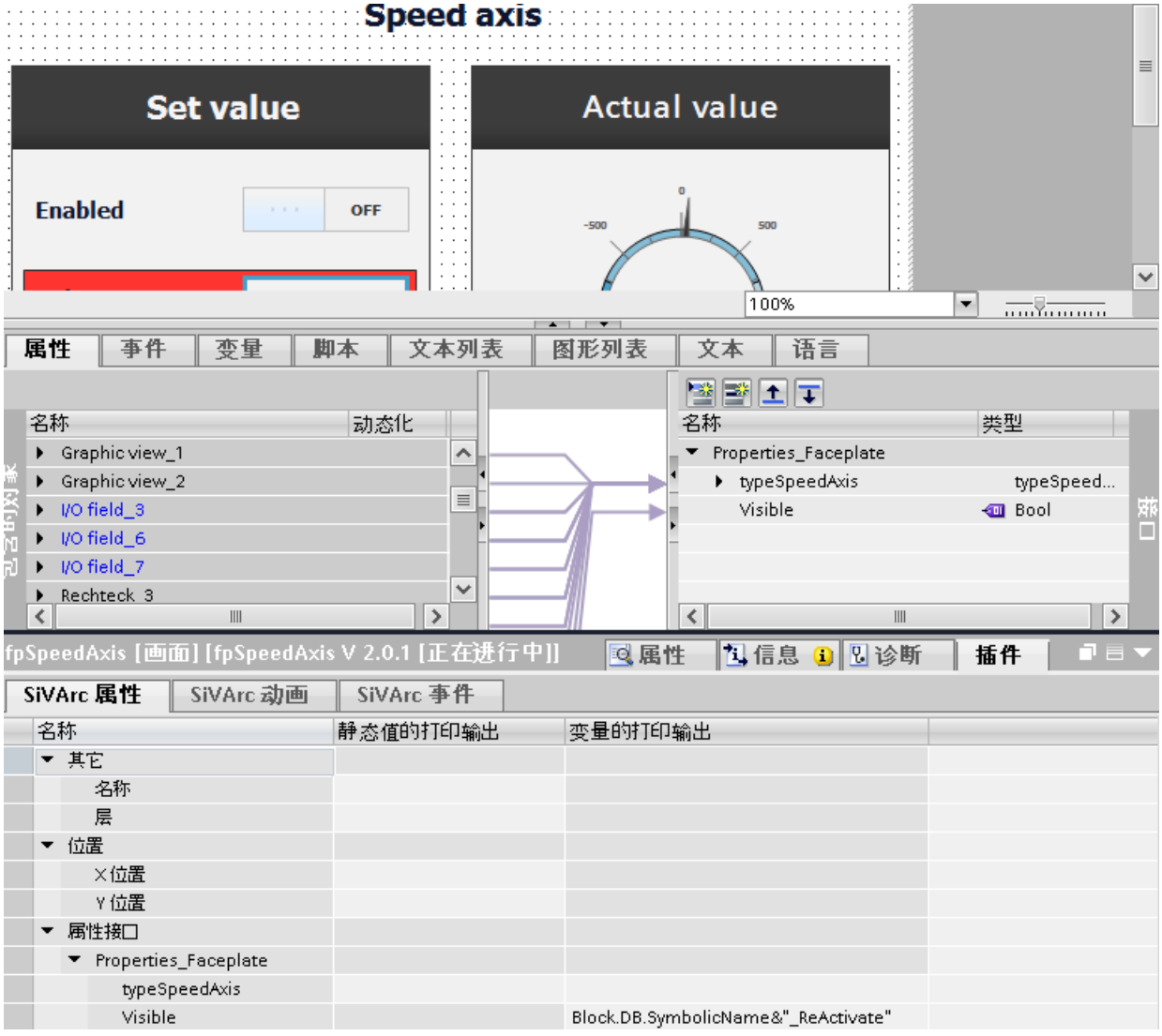
- 面板类型的"fpSpeedAxis"生成模板已存储在库中。
- "Conveyor_HeavyLoad_Instance01_ReActivate"变量已包含在相关函数块的块接口中。
- 以"_ReActivate"结尾的变量仅用于重型操作。

步骤

按照下列步骤生成带有动画的面板：

1. 打开面板类型"fpSpeedAxis"的生成模板。
2. 在面板类型的“接口”(Interface) 列表中，创建 BOOL 数据类型的名称为“可视”的属性。
3. 在面板类型中包含的全部对象的 WinCC 动画中组态此“可视”动画。每次执行该操作时均使用“可视”接口属性作为过程变量。

4. 在面板类型的 SiVArc 属性中，通过 SiVArc 表达式“Block.DB.SymbolicName&"_ReActivate”组态“可视”接口属性。



5. 创建新的面板类型作为生成模板。
6. 按画面规则使用面板类型和相关程序块。

结果

使用此生成模板创建画面规则后，会在生成过程中评估 SiVArc 表达式。SiVArc 生成的外部变量分配给面板类型生成的每个实例的属性。

本示例中，面板上仅针对重型传送带进行互连，原因是只有该部分带有“_ReActivate”结尾的变量。

合并面板的界面属性

组态具有画面项的界面属性的面板时，可以保留对静态值和动态值所做的更改。通过面板**插件**编辑器中的“**保留手动更改**”(Retain manual changes) 复选框，可保留对当前面板实例的界面属性值所做的更改。

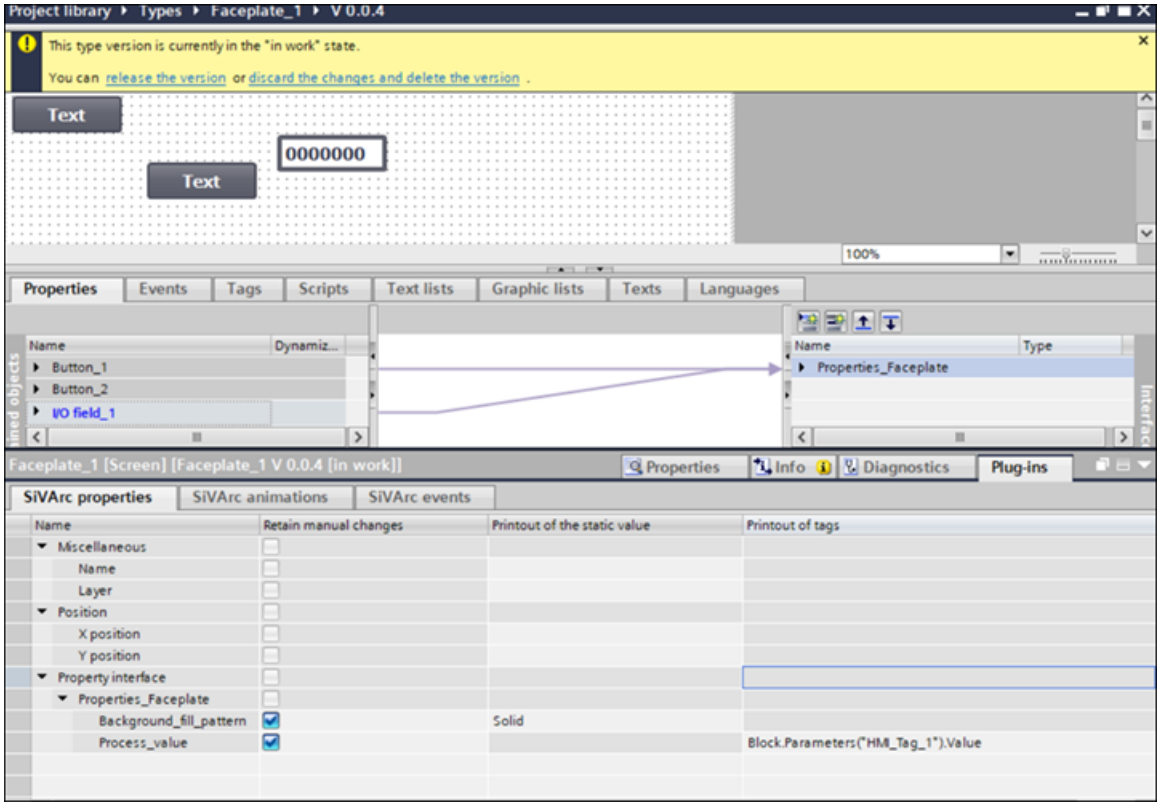
SiVArc 生成后，创建面板实例并选中“**保留手动更改**”(Retain manual changes) 复选框时：

- 如果用户手动更改生成的面板实例的属性值，则生成 SiVArc 后，这些属性值会保留。

步骤

要使用画面中的面板界面属性生成 SiVArc，请按以下步骤操作：

1. 使用画面项组态面板的界面属性。
2. 选中“保留手动更改”(Retain manual changes) 复选框，然后组态静态列的值和动态列的“变量表达式”(Tag expression) 列。

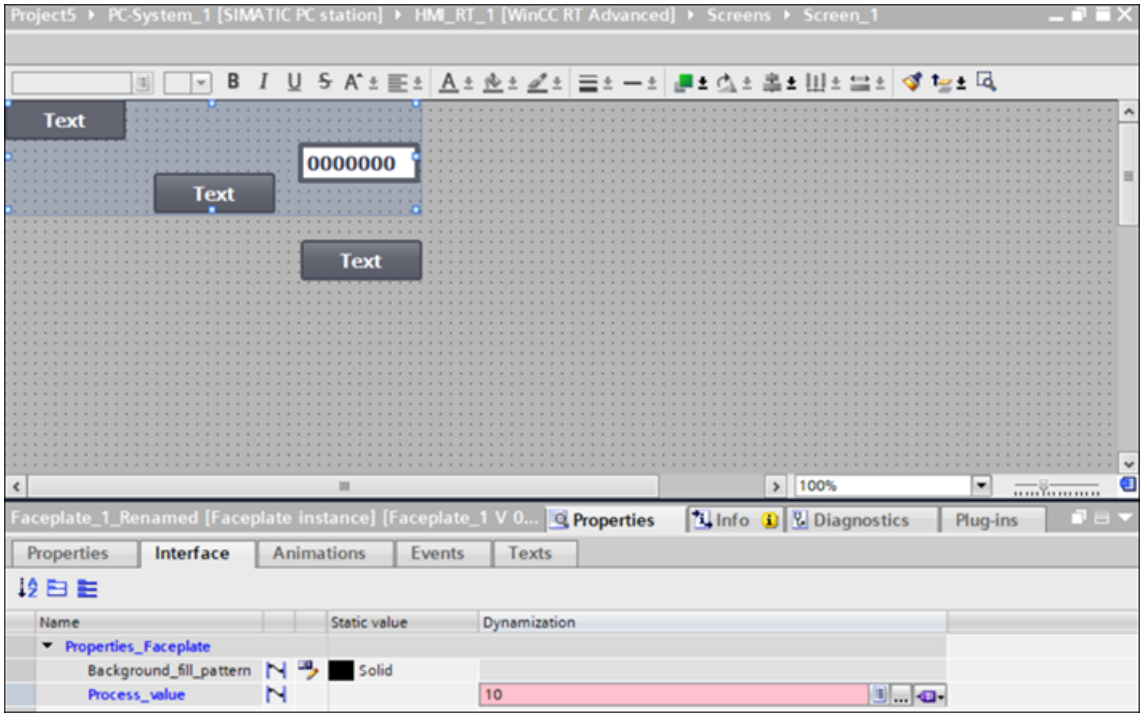


3. 创建一个画面规则，其中画面对象作为面板，画面显示为“画面的主副本”。有关组态画面规则的更多信息，请参见“创建规则 (页 44)”。
4. 生成 SiVArc 可视化。
5. 生成的画面对象包含面板的界面属性。有关生成的更多信息，请参见生成 (页 44)。

编辑面板属性的实例

SiVArc 生成后，可编辑在画面上生成的面板实例。选中“保留手动更改”(Retain manual changes) 复选框后，对生成的面板实例的界面属性执行的任何更改都将保持不变，供下一版本使用。

请考虑面板界面属性包含第一次生成的“Process_value”10 时的情形。生成后，面板的界面属性可通过面板库窗口中选中的“保留手动更改”(Retain manual changes) 复选框进行修改，并触发 SiVArc 生成。对于所有后续生成，更改界面属性后，“Process_value”将根据手动更改而相应更改。以上同样适用于静态属性。例如：Background_fill-pattern。



例外：

- 只有面板库和生成的面板中具有类似的版本时，手动更改才适用。
- 如果删除了面板，则“保留手动更改”(Retain manual changes) 将不适用，并且将在最新的可用版本生成面板。
- 支持动态属性的高级面板不支持静态界面属性。
- 对于高级面板，某些无法由 SiVArc 保留的属性将具有由工程组态系统设置的值。
- 对于动态值，支持的输入类型为 HMI 变量。

参见

“画面规则”编辑器 (页 29)

“生成矩阵”编辑器 (页 38)

生成概览 (页 44)

6.2.24 示例：为面板生成“位置”动画

示例场景

在印刷电路板厂中，制造电路板在“Packaging”机组中进行盒装后传送至推车。该过程将以动画形式显示于 HMI 设备上。

执行规则

包装后的盒子以面板生成模板的形式存储在库中。要显示包装完成的盒子在推车上的水平运动，请为面板组态“位置”动画。水平运动的位置值由控制器向面板提供。

SiVArc 中的模板“位置”动画

面板支持 RT Professional 的“位置”(Position) 动画。

若要使用 SiVArc 为面板生成动画，请在用作生成模板的面板类型中组态动画的动态属性。

要求

- “Plate_Box_Ready”面板类型的生成模板已存储在库中。
- “包装”函数块包含 INT 数据类型的“XPosition”输入参数。
- “XPosition”参数的值存储在相关数据块中。
- 用于生成包装设备可视化的目标 HMI 设备具有相同的屏幕分辨率。

步骤

按照以下步骤为面板生成“位置”动画：

1. 打开库中的面板类型“Plate_Box_Ready”。
2. 在面板类型的“接口”(Interface) 列表中，为水平动画创建 INT 数据类型的“IFace_XPosition”属性。

3. 在面板类型中包含的全部对象的 WinCC 动画中组态新的变量连接。将该变量与“X 位置”属性连接起来。
4. 通过接口属性“IFace_XPosition”组态连接“X 位置”属性的变量。
5. 在面板类型的 SiVArc 属性中，通过 SiVArc 表达式“Block.DB.SymbolicName&"_XPosition”组态“IFace_XPosition”接口属性。
6. 创建新的面板类型。
7. 按画面规则使用面板类型和相关程序块。

结果

生成后，“Plate_Box_Ready”面板的所有已生成实例均组态有动画。在运行系统中，面板的位置遵循互连变量的位置值，例如“Block_1_DB_XPosition”。

6.2.25 示例：为趋势视图创建生成模板

示例场景

在制造工厂使用水力学仿真，当与时间参数对照测量温度参数时，显示的值有使用 HMI 设备的趋势。

执行规则

使用趋势视图图形化展示基于特定参数的变量值，并展示为趋势。SiVArc 自动生成变量值，并将变量值趋势化展示在 HMI 设备中。水力仿真测量温度在特定时间段内上升或下降将趋势化显示在 HMI 设备中。HMI 设备中显示的趋势数据将用于监视水力学组件的状态。

可以在 TIA 和 SiVArc 中组态趋势视图属性。参考 TIA Portal 在线帮助，获取趋势视图属性组态的详细描述。

在 SiVArc 的“SiVArc 属性 > 插件”(SiVArc Properties > Plug-ins) 下组态趋势视图属性。

根据设备可以组态趋势视图的不同属性。关于趋势视图属性的更多信息，请参见 TIA Portal 在线帮助。

要求

- 为 HMI 设备创建包含名为“TrendView1”的趋势视图。

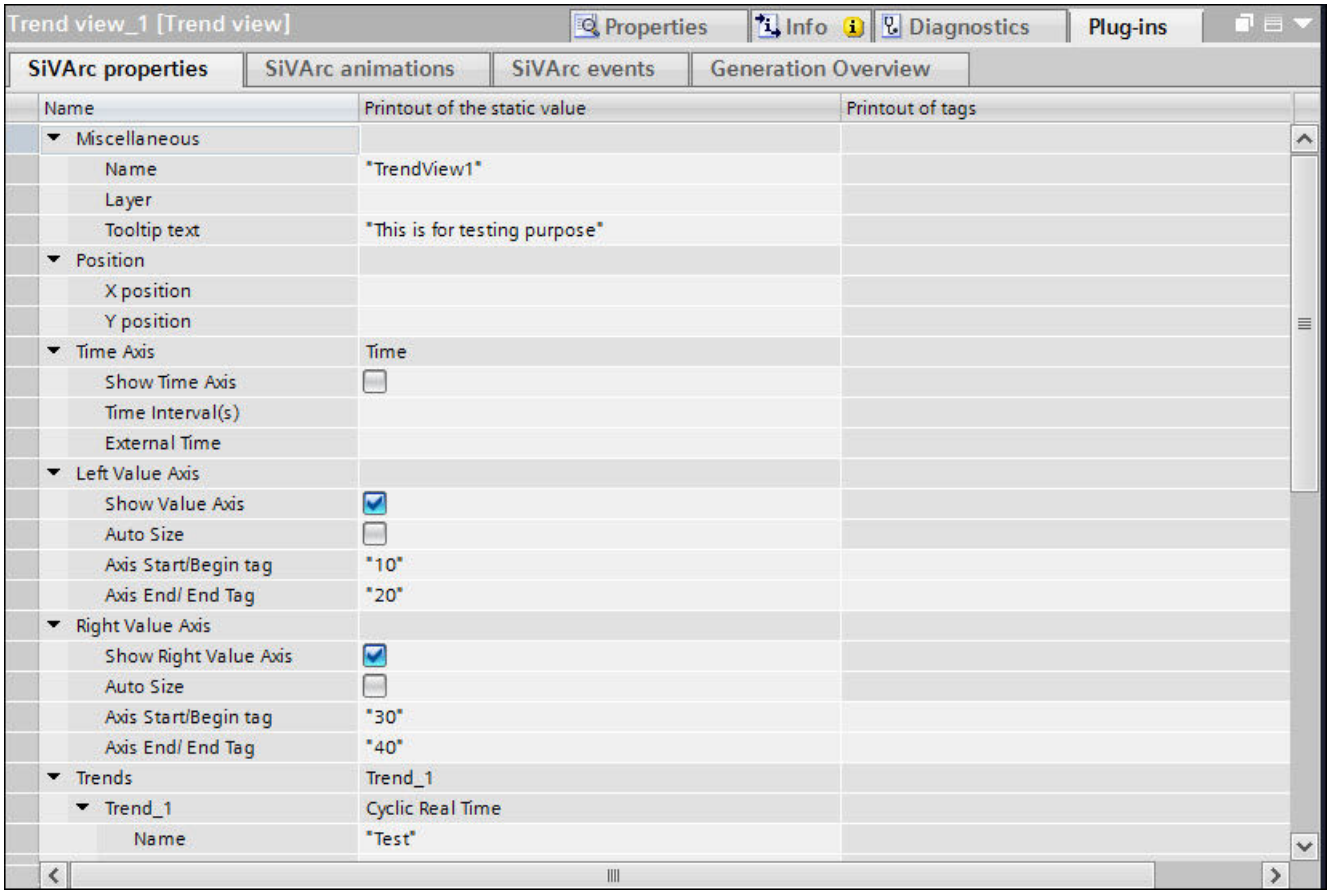
示例：为趋势视图创建生成模板

以下示例显示了如何在 RT Advanced 的 SiVArc 组态趋势视图。

1. 在“SiVArc 属性 > 插件”(SiVArc Properties > Plug-ins)下组态 TrendView1 的属性，例如“层”(Layer) 和“位置”(Position)。

   说明

   在 RT Advanced 中也可以组态“工具提示文本”(Tooltip text) 属性。



2. 为趋势视图组态时间轴，在“插件”(Plug-ins) 下扩展“时间轴”(Time axis) 属性并激活“显示时间轴”(Display time axis) 复选框以在趋势视图控制中显示时间轴。

3. 要组态时间轴，扩展“左侧值轴”(Left value axis) 和“右侧值轴”(Right value axis) 属性。
   – 激活复选框“显示值轴”(Show value axis) 以便在趋势视图中显示轴。
   – 要在对象生成期间设置轴的值为自动值，激活复选框“自动大小”(Auto size)。
   – 为“轴开始/开始变量”(Axis start/begin tag) 和“轴结束/结束变量”(Axis end/end tag) 输入变量值。

   说明

   在 RT Advanced 支持以下“轴模式”：
   - 点：数值数目显示
   - 变量/常数：数值数目显示
   - 时间：显示外部时间。

4. 为组态趋势值，扩展“趋势”(Trend) 属性：
   – 在“名称”下为相关趋势输入有效的名称。
   – 在“数据源过程值”(Data source process values) 下为过程值输入有效值。
5. 为包含趋势的指定画面组态画面规则。关于组态画面规则的更多信息，请参见“定义用于生成画面对象的画面规则”(Defining a screen rule for generating screen object) 部分。
6. 生成可视化。有关生成的更多信息，请参见“生成可视化”(Generating visualization) 部分。

结果

在生成后，温度对时间的变量值作为趋势以图形显示在 HMI 中。

6.2.26 示例：画面的硬件配置

简介

可以组态连接到 PLC 的 IO 设备，还可以生成结果为 IO 设备名称、Invariant 类型和订货号的画面（通过表达式）。连接到 PLC 的 IO 设备可以是相似或不同的 Invariant 类型。将 IO 设备分配给 PLC 时，若尝试组态 SiVArc 规则，“规则触发器”(Rule Trigger) 列将显示可用于已连接 PLC 的 Invariant 类型。SiVArc 生成操作会为 IO 设备名称、invariant 类型和订货号的单个或多个实例创建画面。

示例场景

考虑在汽车制造厂中组装零件的程序，其中 PLC 连接到 IO 设备，具有相似或不同的 Invariant 类型。

目的

在汽车制造厂中，使用多个 HMI 画面来显示零件的组装过程。通过表达式组态 SiVArc 属性，这些表达式将生成具有分配给 PLC 的 IO 设备名称的画面。因此，可以提高多个 HMI 设备的生产率和可扩展性。

要求

- PLC 分配给单个或多个具有相似或不同 invariant 类型的 IO 设备（仅限 ET200 Profinet）
- HMI 设备连接到 PLC
- SiVArc 表达式支持 IO 设备名称、Invariant 类型、订货号

步骤

要使用 IO 设备生成 HMI 画面，请按照下列步骤操作：

1. 为 PLC 分配单个或多个具有所选 Inavarint 类型的 IO 设备。
2. 添加画面。在巡视窗口的“插件 > SiVArc 属性 > 名称”(Plug-ins > SiVArc properties > Name) 下，使用表达式组态画面属性，以生成 IO 设备名称、Invariant 类型和订货号。例如：`"Device.Name"`. 解析表达式名称将显示在目标画面中。
3. 将画面添加到具有画面对象的“主副本”(Master copies) 文件夹中。
4. 在“SiVArc > 画面规则”(SiVArc > Screen rules) 下，单击“创建新规则”(Create new rule)。
5. 在“规则触发器”(Rule Trigger) 列中，浏览至选择的 Invariant 文件夹，以及“主副本”(Master copies) 文件夹中的画面对象。
6. 在 HMI 设备上生成 SiVArc 可视化效果。

说明

- “主副本 > 类型”(Master Copies > Types) 下的 IO 设备不生成画面。
- 如果画面规则包含块和 invariant 类型，则首先执行带有块的画面规则，然后执行 Invariant 类型。

结果

将生成 HMI 画面“<<设备_名称>>”。例如：IO device_1。

6.2.27 示例：报警的硬件配置

简介

可以组态连接到 PLC 的 IO 设备，还可以生成结果为 IO 设备名称、Invariant 类型和订货号的报警、报警组和报警类（通过表达式）。连接到 PLC 的 IO 设备可以是相似或不同的 invariant 类型。将 IO 设备分配给 PLC 时，若尝试组态 SiVArc 规则，“规则触发器”(Rule Trigger) 列将显示可用于已连接 PLC 的 Invariant 类型。SiVArc 生成操作会为 IO 设备名称、Invariant 类型和订货号的单个或多个实例创建报警。

示例场景

考虑监视制造厂的状态。具有特定设备名称的报警表示设备需要重要注意。

目的

在汽车制造厂中，使用多个 HMI 报警来监视组装零件的状态。通过表达式组态 SiVArc 属性，这些表达式将生成具有分配给 PLC 的 IO 设备名称的报警。

要求

- PLC 分配给单个或多个具有相似或不同 Invariant 类型的 IO 设备（仅限 ET200 Profinet）
- HMI 设备连接到 PLC
- SiVArc 表达式支持 IO 设备名称、Invariant 类型、订货号

步骤

要使用 IO 设备生成 HMI 报警，请按照下列步骤操作：

1. 为 PLC 分配单个或多个具有所选 Invariant 类型的 IO 设备。
2. 选择“HMI 报警”(HMI alarms) 并单击“添加新报警”(Add new) 以添加新报警。
3. 在巡视窗口的“插件 > SiVArc 属性 > 名称”(Plug-ins > SiVArc properties > Name) 下，使用表达式组态报警名称属性，以生成 IO 设备名称、Invariant 类型和订货号。例如：`"Device.Name"`. 解析表达式名称将显示在目标报警中。
4. 将报警添加到“主副本”(Master copies) 文件夹中。
5. 在“SiVArc > 报警规则”(SiVArc > Alarm rules) 下，单击“创建新规则”(Create new rule)。
6. 在“规则触发器”(Rule Trigger) 列中，浏览至选择的 Invariant 文件夹，以及“主副本”(Master copies) 文件夹中的报警对象。
7. 在 HMI 设备上生成 SiVArc 可视化效果。

说明

- “主副本 > 类型”(Master Copies > Types) 下的 IO 设备不生成报警。
- 如果报警规则包含块和 Invariant 类型，则首先执行带有块的报警规则，然后执行 Invariant 类型。

结果

将生成 HMI 报警“<<设备_名称>>”。例如：IO device_1。

6.2.28 为画面创建生成模板

要求

- WinCC 项目已经打开。

步骤

要为画面创建生成模板副本，请按以下步骤操作：

1. 创建新画面。

   说明

   分配一个有意义的名称。由于画面名称将用作生成模板的名称，因此名称的唯一性有助于后续工作。

2. 组态画面的属性并根据需要添加所需的画面对象。
3. 在巡视窗口的“插件 > SiVArc 属性 > 常规”(Plug-ins > SiVArc properties > General) 下组态所需的属性：
   - 要生成唯一的画面名称，在“名称”(Name) 下输入一个 SiVArc 表达式或字符串。
   - 如果生成的画面将存储在组或者工厂结构中，则在“画面组”(Screen group) 下输入一个 SiVArc 表达式。
   - 根据需要组态溢出画面。
4. 要创建主副本，在“主副本”(Master copies) 下存储库中的画面。
5. 要创建画面类型，在“类型”(Types) 下存储库中的画面，并且分配类型名称。

   说明

   画面类型的 SiVArc 属性

   画面类型中可使用的 SiVArc 属性比画面的主副本中的少。

结果

已经为画面创建了一个生成模板。

参见

创建画面规则 (页 192)

生成可视化 (页 219)

SiVArc 对象属性 (页 264)

生成对象的存储策略 (页 131)

6.3 创建规则

6.3.1 SiVArc 规则

定义

SiVArc 规则用于定义如何在生成期间处理 HMI 对象。

各 SiVArc 规则定义不同的生成任务：

- 画面和文本列表规则用于链接生成模板与控制指令。
- 变量规则用于控制 SiVArc 所生成 HMI 变量的存储结构。
- 复制规则可触发生成以下 HMI 对象（基于主副本或类型）：
  - 画面
  - C 和 VB 脚本
  - 文本列表
  - 报警
  - 变量表
  - 图形列表
  - 图形表
  - 图形
  - 计划任务

SiVArc 规则是 SiVArc 的关键功能并且与用户程序有直接关系。因此，可像指令一样通过专有技术保护分配 SiVArc 规则.在组态画面规则时，可以在“主副本/画面类型”(Master copy/Type of a screen) 中浏览“画面类型”(Screen type)。画面类型可通过属性和事件进行组态。

通过右键单击“索引 > 插入新规则/插入新规则组”(index > insert a new rule/insert a new rule group)，可在所有规则编辑器中的所有所需索引处添加新规则/规则组。

- 在索引点添加新规则时，新规则将立即插入所选索引的下方，命名方式为“*所选索引名称*_<<*递增数字*>>”。
- 添加新规则组时，新规则组将立即添加到所选规则组的第一个索引点中，命名方式为“*索引组名称*<<_<<*递增数字*>>”。

与不使用 SiVArc 规则组态的区别

与传统 WinCC 组态不同，SiVArc 项目中包留了 SiVArc 规则与已生成 HMI 对象之间的关系。

更改 SiVArc 规则后，基于该规则生成的对象将在下次生成期间被覆盖。删除规则后，与该规则相关联的已生成对象将在下一次生成时自动删除。

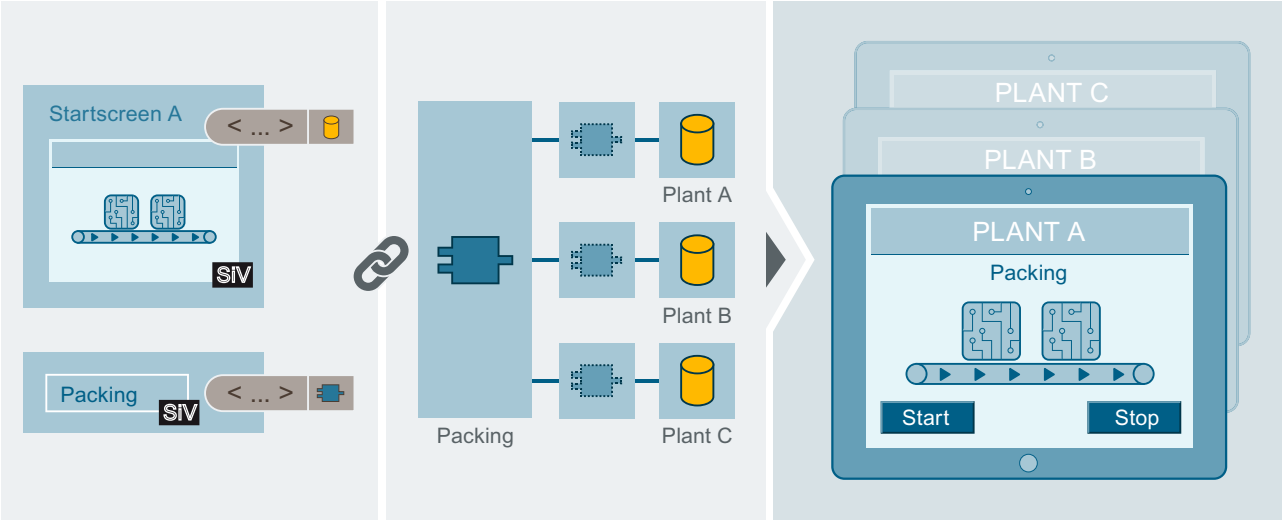
还可以通过 SiVArc 规则为各设备单独创建可视化。

SiVArc 规则的用途与优势

可使用 SiVArc 规则分别针对各 HMI 设备集中控制与控制程序有直接关系的 HMI 对象。因此可在整个项目范围内集中应用所做改动。从可控性和 WinCC 项目效率角度，SiVArc 规则的设计和开发的附加值高。

示例：画面规则

以下示例抽象展示了如何通过生成模板集成 HMI 画面的数据块中的文本：



SiV SiVArc 生成模板

< ... > 引用文本源的 SiVArc 属性

处理指令

主 OB 中指令的实例

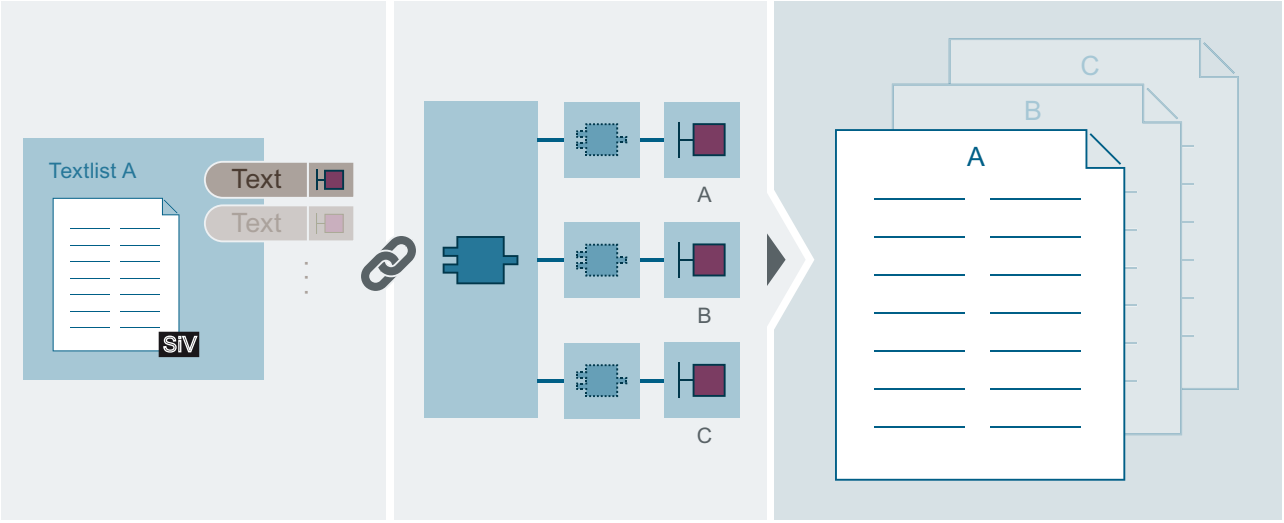
数据块

结果

针对每个引用块实例创建了所引用 SiVArc 主副本的显示和操作对象。根据 SiVArc 规则和 SiVArc 属性创建了显示和操作对象的属性。SiVArc 将根据组态存储已生成的画面。

示例：文本列表规则

以下示例抽象展示了如何通过网络文本生成文本列表：



SiV SiVArc 生成模板

< ... > 引用文本源的 SiVArc 属性

指令

主 OB 中指令的实例

程序段

结果

针对每个引用块实例创建了所引用 SiVArc 主副本的文本列表。根据 SiVArc 规则和 SiVArc 属性创建了文本列表的属性。

SiVArc 随后为每个被调用的程序块在用户程序中组态的文本列表条目生成相应的值。

参见

为 SiVArc 项目设置专有技术保护 (页 208)

更改 SiVArc 规则 (页 216)

创建 SiVArc 规则 (页 170)

创建画面规则 (页 192)

创建文本列表规则 (页 194)

6.3.2 创建 SiVArc 规则

定义

在 SiVArc 编辑器中创建并编辑 SiVArc 规则。

使用 SiVArc 规则编辑器集中控制 SiVArc 生成功能。可使用此方法共同编辑和组织所有规则。可以共同导出或导入规则。

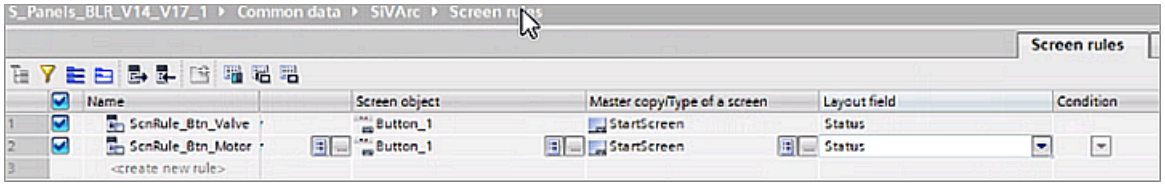
SiVArc 编辑器是表格编辑器。每个 SiVArc 规则类型都有各自的编辑器。编辑器具有不同的对象。

访问 SiVArc 规则编辑器

要打开 SiVArc 编辑器，请双击项目树中的“公共数据 > SiVArc”(Common data > SiVArc) 中的相关条目。

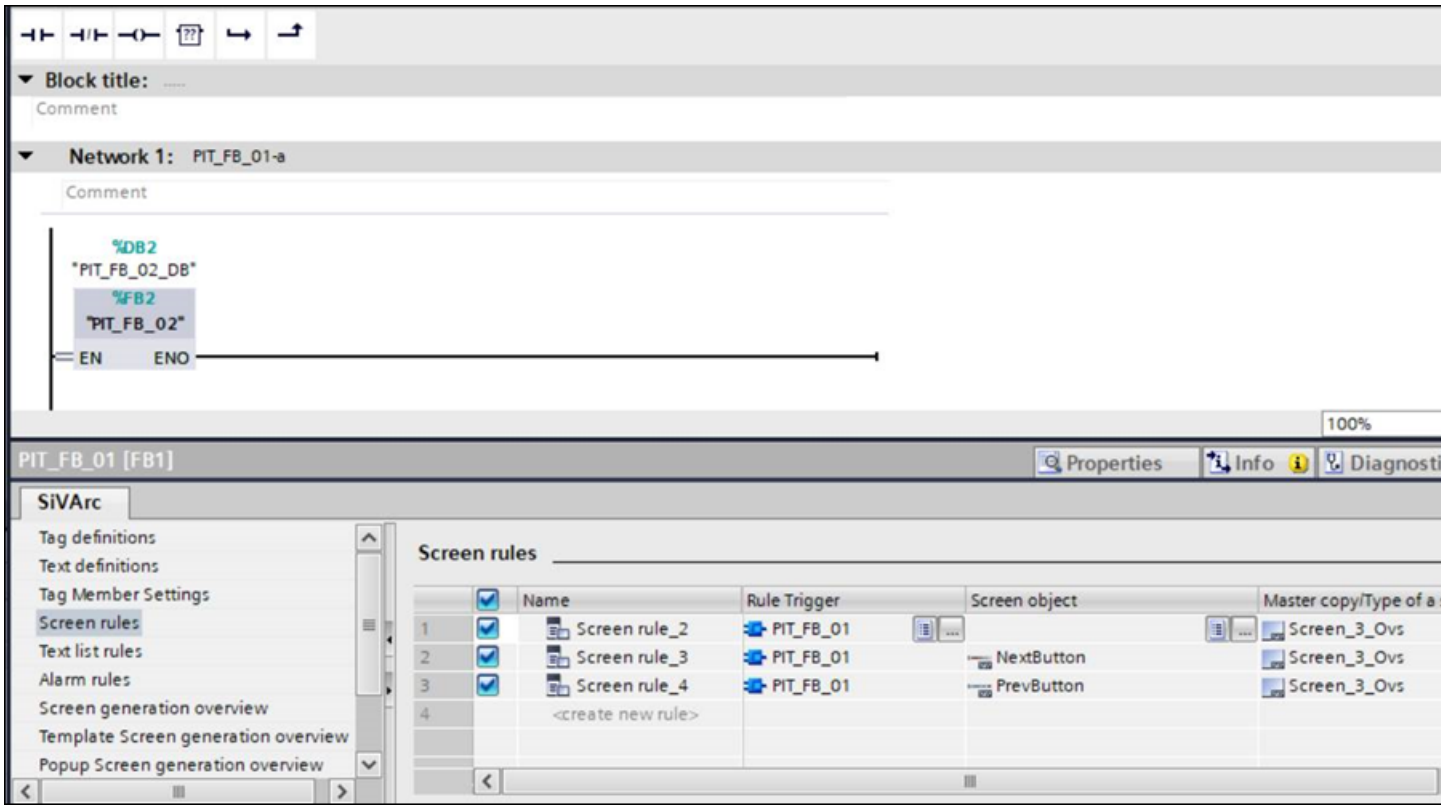
使用工具栏中的图标隐藏或显示单个列，例如“PLC”、“HMI 设备”或“HMI 设备类型”列。

通过在解析为目标设备的宽度和高度的“条件”(Condition) 列中组态表达式，可以生成画面。



在 STEP 7 中也可访问画面和文本列表：

为所选程序块创建的所有画面和文本列表规则都可在程序块上直接访问。所显示规则的范围取决于控制器。



除了导入/导出外，在 STEP 7 中创建和编辑 SiVArc 规则的方式与在实际 SiVArc 编辑器中相同。巡视窗口中没有工具栏。

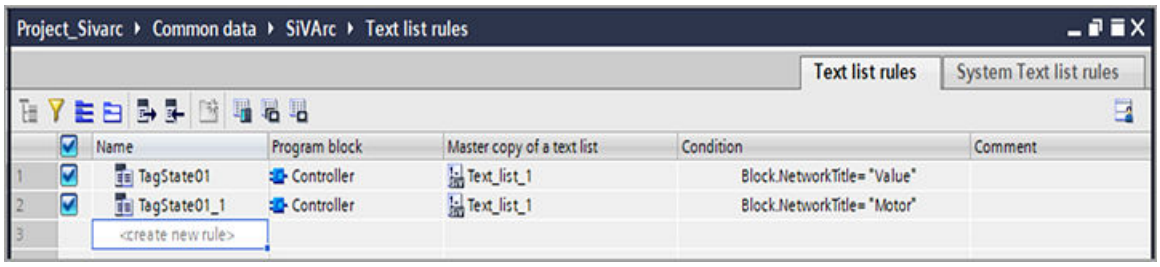
在 STEP 7 中，只能使用项目树中的“公共数据 > SiVArc”(Common data > SiVArc) 下的快捷键菜单命令来移除 SiVArc 规则的专有技术保护。

首次生成后，还可在巡视窗口的“插件 > SiVArc”(Plug-ins > SiVArc) 下获取“模板画面生成概览”(Template screen generation overview) 和“弹出画面生成概览”(Popup screen generation overview) 选项卡。

HMI 对象与程序块的互连

在“画面规则”(Screen Rules) 编辑器中，为多个设备定义生成画面中的 SiVArc HMI 对象所使用的画面规则。

在“文本列表规则”(Text list rules) 编辑器中，定义为不同设备生成文本列表所用的 SiVArc 规则。



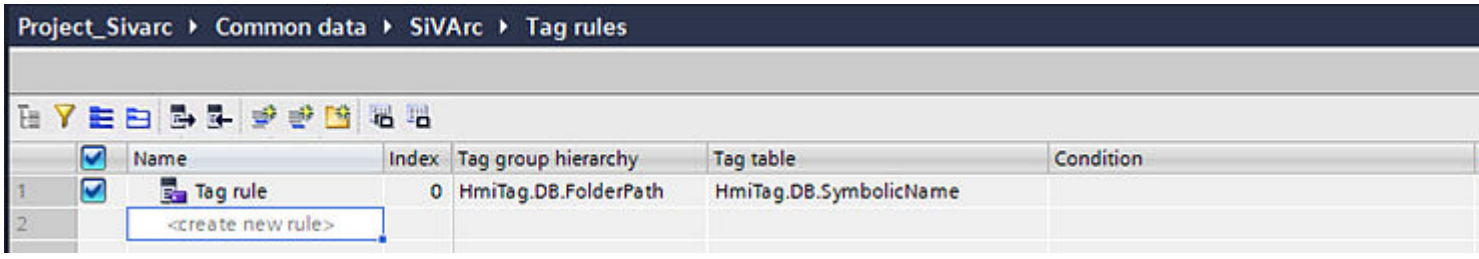
此时，可以创建规则并为每条规则指定以下内容：

- 已连接组件
  - 画面规则：程序块、画面对象、画面、布局字段
  - 文本列表规则：程序块、文本列表
- 执行规则的条件
- 规则注释

在“画面规则”(Screen rules) 编辑器中，可以显示“PLC”、“HMI 设备”或“HMI 设备类型”列。在这些列中，可启用或禁用在生成期间 SiVArc 为其应用这些规则的设备。通过这种方式，可为所选设备生成 HMI 对象。

控制变量的储存结构

在“变量规则”(Tag rules) 编辑器中，可定义通过 SiVArc 生成的外部变量所采用的变量规则，这些外部变量以结构形式存储。

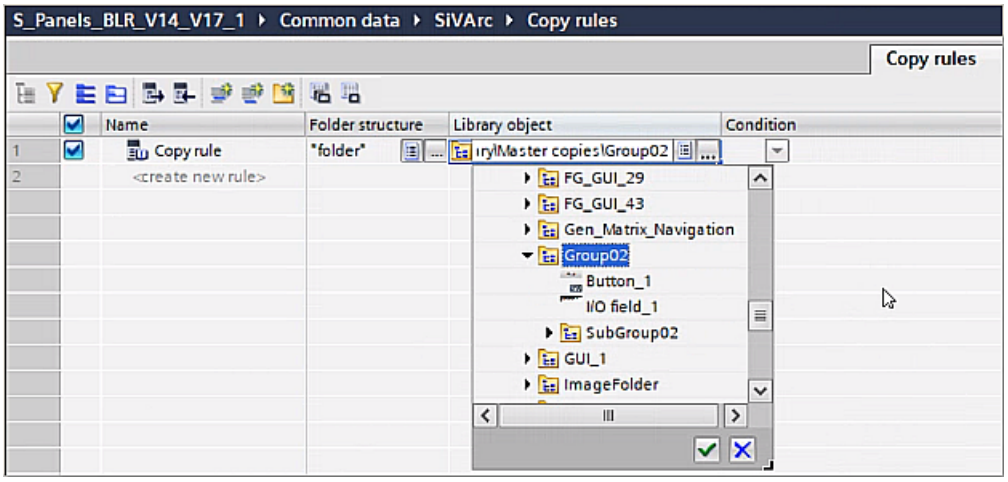


此时，可以创建规则并为生成的变量定义以下内容：

- 变量组的名称
- 变量表的名称
- 规则执行顺序
- 执行规则的条件
- 规则注释

将 HMI 对象系统地插入项目中。

在“库规则”(Library rules) 编辑器中，可定义为不同 HMI 设备从库中生成所选对象时遵循的规则。



此时，可创建规则并指定以下内容：

- 待创建的库对象
  或
  待创建的库对象组的 HMI 对象
- 规则注释

也可以显示“HMI 设备”(HMI device) 列和“HMI 设备类型”(HMI device type) 列。在这些列中，可启用或禁用在生成期间 SiVArc 为其应用这些规则的设备。通过这种方式，可为所选设备生成 HMI 对象。

参见

导出和导入 SiVArc 规则 (页 205)

为 SiVArc 项目设置专有技术保护 (页 208)

编辑和管理 SiVArc 规则 (页 203)

创建画面规则 (页 192)

创建文本列表规则 (页 194)

在 SiVArc 编辑器中编辑视图 (页 286)

6.3.3 在 SiVArc 规则中使用 SiVArc 脚本

定义

在执行规则的条件下，可在 SiVArc 规则中使用 SiVArc 脚本。在变量规则中，可同样使用 SiVArc 表达式对外部变量进行结构化。

基本上，当“SiVArc 表达式”(SiVArc expressions) 编辑器储存在输入字段时，可始终使用 SiVArc 表达式。

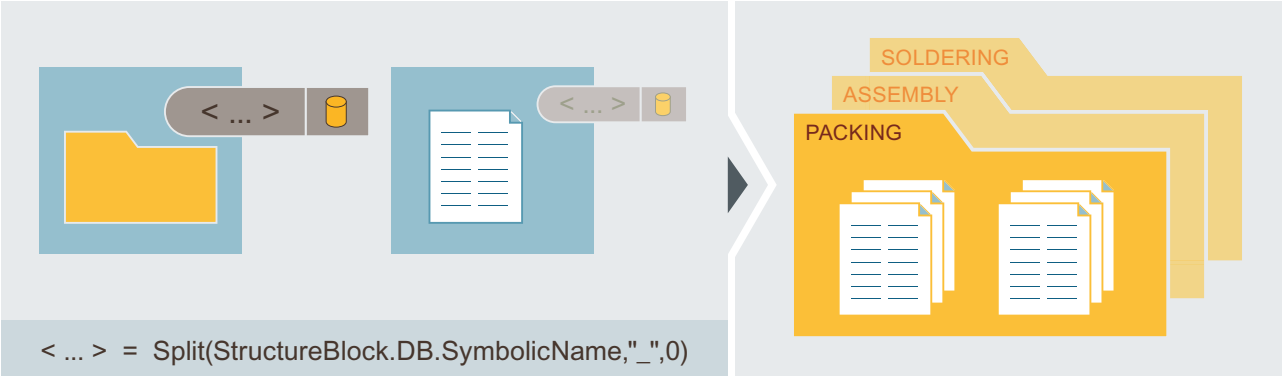
SiVArc 规则的条件

在使用返回布尔值 TRUE 或 FALSE 的 SiVArc 脚本函数的条件下。

如未指定条件，则将始终执行画面规则。可将条件分配给整个规则组。随后，该条件可应用于组内包含的所有规则。用户可为规则组中的各项规则细化条件。

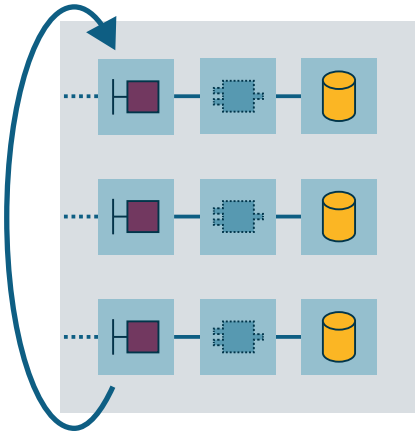
变量的存储结构

可在项目树中使用 SiVArc 表达式指定变量组和变量表的名称。通过这种方法，可将存储结构链接到用户程序。例如，使用这种方法根据块实例对变量进行排序。



在“变量规则”(Tag rules) 编辑器中，可以使用 SiVArc 表达式 `HmiTag.DB.SymbolicName` 和 `HmiTag.DB.FolderPath` 基于仅使用一种变量规则的控制程序对变量表进行结构化。仅控制器程序员可对项目进行结构化。对于可视化，SiVArc 应用来自 STEP 7 的存储结构。

仅在“变量规则”(Tag rules) 编辑器中，可以使用用于寻址 SiVArc 对象 `HMITag` 的 SiVArc 表达式。


名称
索引
变量组层级
变量表
条件
1
Plantsection_1
0
HmiTag.DB.FolderPath
HmiTag.DB.SymbolicName
2
<创建新规则>


6.3.4 规则处理：

工作原理

SiVArc 在生成期间执行用户程序。如果规则适用于函数块，则执行该规则。同时，SiVArc 通过数据库运行。

当必须生成外部变量时，SiVArc 从上至下运行变量规则并根据规范存储变量。因此，每个变量仅应用一个规则。

执行已启用且符合组态条件的项目中的所有画面和文本列表规则。

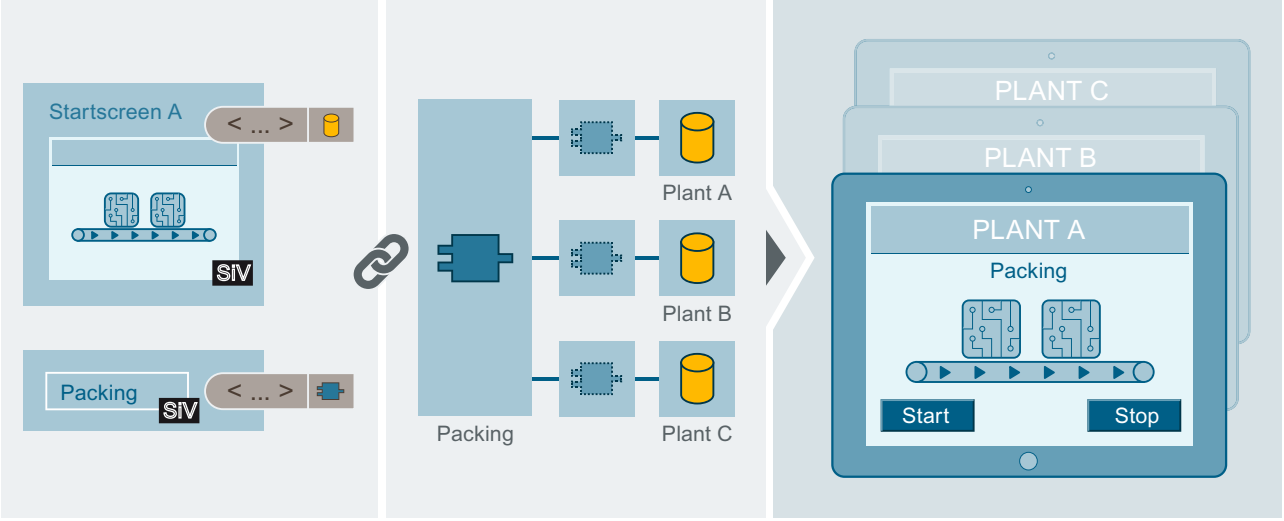
评估画面规则

下列原理适用于画面规则：

- 必须为每个要生成的画面对象定义画面规则。
- 如果要通过程序块生成不同的画面对象，必须使用条件为每个画面对象定义一个画面规则。在条件中指定要生成的画面对象。
- 如果要生成的画面对象的画面不存在，则会在生成过程中创建该画面。

- 如果“画面规则”(Screen Rules) 编辑器的多个画面规则中包含某个块，将按画面规则的顺序创建对象。
- 如果要为程序块生成没有画面对象的画面，请将“画面对象”(Screen object) 字段留空。

SiVArc 按照所选 PLC 的所有 OB 的调用层级来执行用户程序。为每个 PLC 执行“主”模块。SiVArc 评估每个调用的程序块的画面规则。



对于各个适用的画面规则，根据生成模板在指定的画面中生成对应的显示和操作对象。在生成过程中，将评估生成模板的 SiVArc 属性、事件和动画中的 SiVArc 表达式。

评估变量规则

变量规则的处理顺序与外部 HMI 变量的存储相关。如有必要，可通过拖放操作更改顺序。

例如，将工厂特定的规则安排为第一项，将以结构化形式存储工厂范围函数变量的规则安排为最后一项。这保证了所有工厂特定的变量储存在一起。

SiVArc 执行站选择对话框内已启用的全部 PLC 的所有数据块。如果已选中数据块中的“可通过 HMI 访问”(Accessible from HMI) 选项，SiVArc 为每个数据块变量生成一个外部变量。

对于要生成的每个外部变量，SiVArc 都会自上而下运行变量规则并评估相关条件。只要条件为真，该规则就适用并且规则的外部变量会相应地存储在项目树中。将不再处理后续的变量规则。但 SiVArc 会继续处理下一个要生成的外部变量。

如果没有适用于要生成的外部变量的变量规则，则该外部变量将存储在默认变量表中。

根据“选项 > 设置 > SiVArc”(Options > Settings > SiVArc) 中的设置，SiVArc 仅会生成外部变量，这类变量还会在生成的 SiVArc 项目中进行互连。

在生成期间，SiVArc 会处理 HMI 设备的运行系统设置中的变量设置。生成的外部变量名称表示与 WinCC 变量同步对应的数据块中的变量的符号地址。

评估复制规则

SiVArc 处理复制规则。对于每个复制规则，每个指定 HMI 设备对应的 HMI 对象均在项目树中创建。

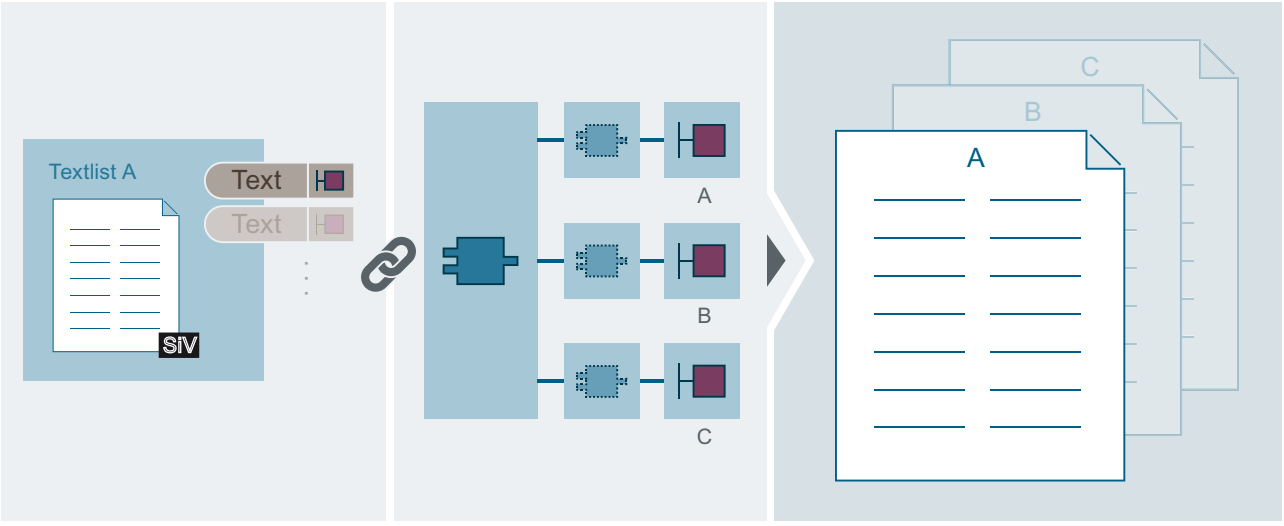
评估文本列表规则

由于文本列表规则的使用通过用户程序中程序块的调用层级来定义，所以文本列表规则的顺序是不相关的。

SiVArc 始终会处理程序块中包含的正在由 SiVArc 进行评估的所有文本列表规则。

因此，系统会评估文本列表的 SiVArc 属性。生成文本列表后，文本列表中将扩充新条目并覆盖现有的相同条目。

该文本列表将存储在 HMI 设备中（已对其触发生成）。



SiVArc 随后为每个被调用的程序块在用户程序中组态的文本列表条目生成相应的值。在此过程中， SiVArc 按照所选 PLC 的所有 OB 的调用层级来执行用户程序。

存在命名冲突时生成对象的优先级

存在命名冲突时，SiVArc 会在生成过程中设定以下优先级：

1. 根据画面、变量和文本列表规则生成的对象
2. 根据复制规则生成的对象
   根据复制规则生成的对象被视为与手动创建的对象相同。首先，它们在生成过程中创建。如果与之后生成的对象存在命名冲突，根据复制规则生成的对象将以扩展名"_renamed"重命名。
3. 手动创建的对象
   如果手动创建的对象的名称与生成的对象相同，手动创建的对象将被重命名。

参见

创建画面规则 (页 192)

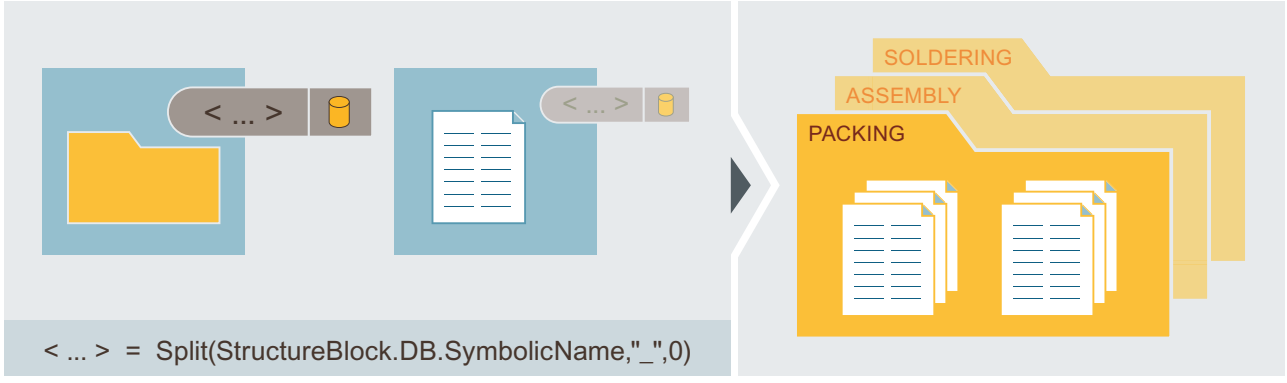
创建文本列表规则 (页 194)

画面图例 (页 287)

6.3.5 生成变量

定义

SiVArc 为每个带有“可通过 HMI 访问”(Accessible from HMI) 选项的 PLC 变量生成一个外部 HMI 变量。根据“选项 > 设置 > SiVArc”(Options > Settings > SiVArc) 下模式指定的项目范围，SiVArc 生成所有变量或仅生成项目中使用的变量。从生成开始，就根据结构规范将外部变量存储在项目树的变量表和组中。



内部 HMI 变量和 SiVArc 变量

如果要通过 SiVArc 创建内部 HMI 变量，则需要在库中存储相应主副本。然后在库规则中使用此主副本。

外部变量通常在内部 HMI 变量之前生成。发生命名冲突时，重命名生成的内部 HMI 变量和手动创建的变量。

不同于手动创建的 HMI 变量和 SiVArc 变量，生成的外部变量可直接链接到 SiVArc 并在每个生成期间被自动收集。

变量生成的优势

变量的生成机制使显示和操作对象内的 PLC 变量能够自动互连。通过已选的变量生成模式可不断从项目中移除不需要的变量。从而进行高效无错误的组态，且无需占用不必要的存储空间。

外部变量的互连

通过在生成模板中的将过程变量组态为 SiVArc 表达式来创建自动互连。如果在生成期间将 SiVArc 表达式评估为已存在的变量名称，则已生成的显示对象将与此变量互连。

设置更新周期和采集类型

如有必要，可按多个步骤设置生成的外部 HMI 变量的更新周期和采集类型：

- 针对各个程序块
  可通过数据块巡视窗口的“插件 > SiVArc > HMI 变量设置”(Plug-Ins > SiVArc > HMI tag settings) 下的“使用通用组态”(Use Common Configuration) 选项定义程序块变量的更新周期和采集类型。此设置会禁用各个变量的设置。
- 针对各个变量
  当“使用通用组态”(Use Common Configuration) 选项禁用时，可在数据块中分别组态各个变量。
- 项目范围
  在“公共数据 > SiVArc > SiVArc 设置 > 变量生成设置”(Common data > SiVArc > SiVArc settings > Tag generation settings) 下的 SiVArc 设置中，可组态项目生成的所有外部变量。此设置只能在未定义变量生成的其它设置时评估。

用户数据类型仅支持循环采集类型。如果为整个项目或某个程序块设置“按需”(On demand) 采集类型，则会针对各个用户数据类型将更新周期设置为 1 s（“循环操作”(Cyclic in operation) 采集类型）。

对于不支持设置采集类型和更新周期的 HMI 设备，为其自动设置 500 ms 的更新周期。

对于 Unified 设备，提供 250ms 的采集周期。

说明

复制程序块和变量组态

再次为每个程序块设置更新周期和采集类型。即使是复制完整组态的程序块，也要再次为变量生成组态其设置。

变量名的默认设置

对 TIA Portal 中生成的变量名称进行了以下默认设置：

- 分隔符始终为“_”
- 用“｛”和“｝”替换方括号“[”和“]”

如有必需，使用 SiVArc 对象 `TagNaming`，其在 SiVArc 表达式中处理这些设置。有关更多信息，请参见“变量生成的原理 (页 185)”部分。

说明

结构化变量中的分隔符

在结构化变量中，层级始终由“.”隔开。

变量名称中的空格

生成变量时，SiVArc 不考虑空格。例如，即使通过 SiVArc 为其提供变量的函数块的名称中包含空格也是如此。

SiVArc 将忽略该空格。在此情况下，互联期间可能发生错误。

示例

函数块的 DB 实例名称包含空格：“TT5684 Temperature”，因为其用作消息文本。如果不删除函数块名称中的空格，则将创建块并通过不存在的变量“TT5684Temperature”（无空格）以红色突出显示接口属性。

请删除函数块变量名称中的空格。这样一来，通过变量规则创建变量名称时，可根据其名称调整 SiVArc 表达式。

变量生成场景

PLC 变量通过 PLC 变量窗口中的“可通过 HMI 访问”(Accessible from HMI) 复选框在 HMI 设备上使用。创建画面时，画面项的输入和输出字段使用 PLC 变量进行组态，HMI 生成时也是如此。有关变量的详细信息，请参见“变量生成原则”。

情形 1：选择所有 HMI 变量时

在“PLC 变量”(PLC tags) 窗口中，组态 PLC 变量后，如果选中“可通过 HMI 访问”(Accessible from HMI) 复选框，并且在变量生成模式设置为“所有 HMI 变量”(All HMI tags) 时触发生成，则系统会在“HMI 变量”(HMI tags) 文件夹下生成 HMI 变量。生成的 HMI 变量在默认变量表下生成，变量名称为“块名称”_“输入/输出变量名称”。

情形 2：选择使用的 HMI 变量时

添加输入/输出设备，并在 SiVArc 画面编辑器中组态具有动态变量名称的输入/输出字段。PLC 变量组态为 PLC，动态变量组态为画面或画面项。动态变量名称将引用任何解析为以 PLC 变量值为后缀的 PLC 名称的表达式。如果选中 PLC 变量窗口中的“可通过 HMI 访问”(Accessible from HMI) 复选框，并且在“变量生成”(Tag generation) 模式设置为“使用的 HMI 变量”(Used HMI tags) 时触发 SiVArc 生成，则只会在“HMI 变量 > 默认变量表 ”(HMI tags > Default tag table) 文件夹下生成在 HMI 设备中使用的 PLC 变量。

情形 3：使用多个 PLC 和变量规则

连接多个 PLC 时，组态表达式为“S7.Control.Name”和“HmiTag.SymbolicName”的变量规则，并选中“可通过 HMI 访问”(Accessible from HMI) 复选框；组态 SiVArc 画面，并在“变量生成模式”(Tag generation mode) 设置为“所有 HMI 变量”(All HMI Tags) 或“使用的 HMI 变量”(Used HMI Tags) 时触发 SiVArc 生成；生成的 HMI 变量命名为“PLC 名称”_“PLC 变量名称”。生成的 PLC 变量位于“生成概述”(Generation Overview) 下。HMI 变量根据变量规则中组态的表达式在相应的“变量文件夹”(Tag Folder) 和“变量表”(Tag Table) 下生成。“生成概览”(Generation overview) 可显示 PLC 变量和块变量。有关生成的更多详细信息，请参见“生成可视化 (页 209)”。

参见

示例：调整变量名称 (页 201)

创建变量规则 (页 191)

6.3.6 使用复制规则

定义

当生成可视化时，复制规则会根据主副本或类型复制 HMI 对象。

仅为以下 HMI 对象创建复制规则：

- 内部变量
- 文本列表
- 报警
- 模板
- 画面
- 脚本
- 图形列表
- 图形
- 图形表
- 计划任务

复制规则支持 SiVArc 表达式或条件。虽然从复制规则创建的 HMI 对象与用户程序无关，但与其它生成的对象一样，它们仍然具有到 SiVArc 的链接。因此，如果需要，复制的对象在下一次生成期间再次收集并进行更新。如果您使用的主副本从库中删除，则其之前生成的对象将从项目树中删除。

将计划任务用作库对象，可以通过复制规则创建规则。

针对库对象使用复制规则：

- 使用计划任务可以执行 SiVArc 生成操作，这将导致生成对象作为计划任务，并扩展到 WinCC Unified 设备。具有相同库对象的多项复制规则将被视作单一规则。
- 如果由 SiVArc 生成和用户自行创建的计划任务名称相同，则 SiVArc 会自动将用户创建的计划任务重命名为“任务 <<递增数字>>_Renamed”。
  – 专业设备仅支持类型为“函数列表”和“打印作业”的计划任务。
- 在 WinCC Runtime 设备中创建类型为“打印作业”的计划任务，同时在生成期间，如果选择在 WinCC Advanced 设备中生成对象，SiVArc 将显示错误。

- 借助存储计划任务、画面和变量表的文件夹，用户可以执行 SiVArc 生成操作，这将导致在 TIA portal 的特定 HMI 设备中生成作为计划任务、画面和变量表的对象。
- 从“选择用于 SiVArc 生成的站点和控制器”(Select stations and controllers for SiVArc generation) 中可以为上述要点选择 Energy Suite 规则。

复制规则的用途

在标准化操作员控制和监视解决方案中，HMI 对象通常集中创建，然后作为全局库分配给组态工程师。使用复制规则，可为项目中的每个 HMI 设备自动生成这些 HMI 对象。

根据特定项目标准，可以使用库规则系统地复制大量已预先分类的对象。这样还可确保项目中的脚本、文本、内部变量和图形实现标准化。

生成与脚本互连的显示画面和操作对象时，可以使用复制规则确保这些脚本也存在于 HMI 设备上。

此外，还可以使用复制规则来创建内部变量和变量表，例如，最初只用于一个 HMI 设备。对于所有其它 HMI 设备，将变量存储在项目库中，以便在下次生成时 SiVArc 可以将其自动复制到所有设备上。

可通过复制规则/系统复制规则将项目库和全局库中可用的图形复制到文件夹路径“语言和资源 > 项目图形”(Languages & resources > Project graphics)。组态复制规则/系统复制规则时，可选择以库对象形式浏览图形。可使用复制规则/系统复制规则中的图形执行以下操作：

- 在 TIA Portal 项目中：
  - 如果为复制规则/系统复制规则组态图形，SiVArc 生成时，系统会自动在“语言和资源” (Languages & resources) 中创建文件夹“项目图形”(Project graphics)。
  - 即使未在站选择对话框中选择 HMI/PLC 设备，组态了图形的复制规则/系统复制规则也可生成图形对象。生成的图形对象将位于文件夹路径“语言和资源 > 项目图形” (Languages & resources > Project graphics) 下。
  - 第二次生成期间的复制规则/系统复制规则会重命名用户自定义图形对象。
  - 如果为复制规则/系统复制规则组态了包含 SiVArc 对象的文件夹路径，SiVArc 生成时，系统会忽略不受支持的对象，并会在“项目图形”(Project graphics) 文件夹中生成图形。
  - 组态复制规则/系统复制规则时，可选择浏览已生成的图形对象。如果生成的图形对象已重命名，且执行第二次生成，则系统会生成新的图形对象。

工作原理

可以在库中创建要复制的 HMI 对象作为组的一部分或作为单个对象。可以根据需要在库规则或单个库元素中使用此组。可以将规则分组，必要时，可按组启用或禁用。在复制规则中，还可以控制正在执行规则的 HMI 设备。

对于每个复制规则，每个指定 HMI 设备对应的 HMI 对象均在项目树中创建。

命名冲突

根据复制规则生成的对象在命名冲突时被视为与手动创建的对象相同。首先，它们在生成过程中创建。如果与之后生成的对象存在命名冲突，根据复制规则生成的对象将以扩展名“_renamed”重命名。

生成内部变量

按照下列步骤生成内部变量：

1. 创建一个变量表。
2. 在此变量表中组态内部变量。
3. 将变量表作为主副本保存到项目库中。
4. 创建一个复制规则，将变量表的主副本复制到指定 HMI 设备。

参见

SiVArc 规则 (页 166)

创建复制规则 (页 197)

6.3.7 SiVArc 表达式和条件之间的相关性

应用示例

例如，可以在变量规则中使用 SiVArc 表达式和条件。可以使用 SiVArc 表达式来指定组名称和变量表的名称。可以使用条件来指定变量规则是否应用于变量。

以下条件指定，例如，变量规则仅适用于名称包含文本字符串“Plantsection1”的变量：

说明

SiVArc 表达式返回访问文本源的文本。还可以在 SiVArc 表达式中制定条件。返回值始终为文本。

条件返回布尔值。使用 SiVArc 规则中的条件。

主要在生成模板中使用 SiVArc 表达式。SiVArc 表达式在生成过程中评估。生成的 HMI 对象的 SiVArc 属性使用生成的文本进行组态。这意味着生成的显示画面和操作对象在文件夹中进行命名、标记和保存，例如使用 SiVArc 表达式并与正确的变量互连。

可以使用画面、变量和文本列表规则的条件。在生成过程中评估条件，并执行或忽略规则。这样可以限制规则并允许异常。

参见

SiVArc 脚本语言 (页 111)

If 条件 (页 282)

SiVArc 表达式 (页 113)

6.3.8 变量生成的原理

简介

使用 SiVArc 的变量生成会利用 WinCC 中的变量同步，并在生成过程中处理相关的运行系统设置。

此外，SiVArc 具有控制变量生成范围和生成变量存储的功能。在使用 SiVArc 生成之前，可以定义变量的更新周期和采集类型。

任务	实现	TIA Portal
生成的范围	在 STEP 7 中生成的变量的标识	数据块中的“可从 HMI 访问”(Accessible from HMI)
	使用 SiVArc 的变量生成模式	“选项 > 设置> SiVArc”(Options > Settings > SiVArc)
变量名称	WinCC 中的变量运行系统设置	HMI 设备的运行系统设置
存储结构	SiVArc 的变量规则	“变量规则”(Tag rules) 编辑器
更新周期 和 采集类型	STEP 7 中的数据块上	“插件 > SiVArc > HMI 变量”(Plug-ins > SiVArc > HMI tags)
	项目范围	“公共数据 > SiVArc > SiVArc 设置 > 变量生成设置”(Common data > SiVArc > SiVArc settings > Tag generation settings)

变量的运行系统设置

生成期间，SiVArc 会考虑 HMI 设备的运行系统设置中的变量设置。SiVArc 会根据所设置的命名约定命名生成的外部变量。

在第一次 SiVArc 生成后，如果更改了变量设置，则 SiVArc 将根据新的设置生成所有变量。现有 SiVArc 变量将重新命名。

可在第一次 SiVArc 生成前组态一次变量设置。如果项目的 HMI 设备变量需要不同的设置，则可以使用 SiVArc 表达式访问带 SiVArc 对象 `TagNaming` 的运行系统设置。

使用 SiVArc 指定生成变量的范围

可以为项目在所需选项“选项 > 设置 > SiVArc”(Options > Settings > SiVArc) 下选择变量生成模式。

如果已启动生成操作，请在用于生成可视化的对话框中选中这些选项的设置。

如果在首次生成之后仅选择此设置，则会根据生成对象的规则处理现有外部变量：

- 将删除 SiVArc 项目中未使用的外部变量。这释放了内存空间。
- 手动编辑的变量将保留，必要时，可对其进行重命名。

使用 SiVArc 控制存储结构

如果在变量规则中使用带 SiVArc 对象 `HMITag` 的 SiVArc 表达式，则在下一次生成期间的存储结构中实现用户程序的更改。如果工厂的功能区域是新分布的，例如，互连的变量也会根据项目树中的新分布进行排序。

SiVArc 仅在适用于变量的首个变量规则之后存储变量。如有必要，可通过拖放操作更改顺序。如果另一个适用的变量规则位于顶部，则在下一次生成期间变量将根据此规则进行存储。

指定使用 SiVArc 的更新周期和采集类型

可以分三个阶段设置生成的更新周期和采集类型：

- 项目范围
- 针对各个程序块
- 针对各个变量

这些设置是在每一次生成期间重新分配的。尽管中央控制，您仍然可以灵活组态，从而在运行系统中优化项目的性能。

用途与优势

变量生成可以实现变量的高效和无差错组态。此生成可以实现变量的统一命名和生成显示与操作元素的自动互连。

使用 SiVArc 的变量生成可以更轻松地处理现有项目的更改和优化。

参见

示例：调整变量名称 (页 201)

生成变量 (页 178)

创建变量规则 (页 191)

示例：使用 SiVArc 生成变量 (页 19)

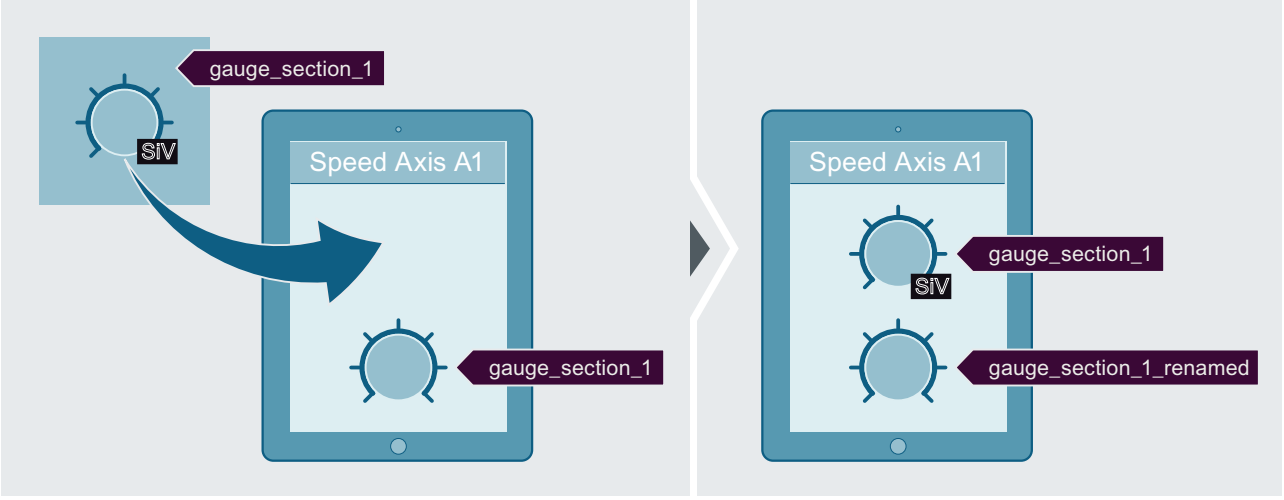
6.3.9 生成期间避免冲突

现有的和生成的 HMI 对象之间的命名冲突

如果生成的 HMI 对象和手动创建的 HMI 对象在 SiVArc 项目中并排使用，则可能导致命名冲突。评估的 SiVArc 表达式可以产生现有对象的名称。

除了画面和文本列表，在命名冲突期间 SiVArc 行为如下：

如果采用 SiVArc 所生成名称的手动创建 HMI 对象已存在，则将对现有对象追加后缀 "_renamed"。如果该名称也已占用，则名称会自动递增。



示例：符号 I/O 域手动互连到项目中的文本列表。然后使用文本列表规则，通过 SiVArc 创建同名的文本列表。现有的手动互连文本列表被重命名，互连保持不变。

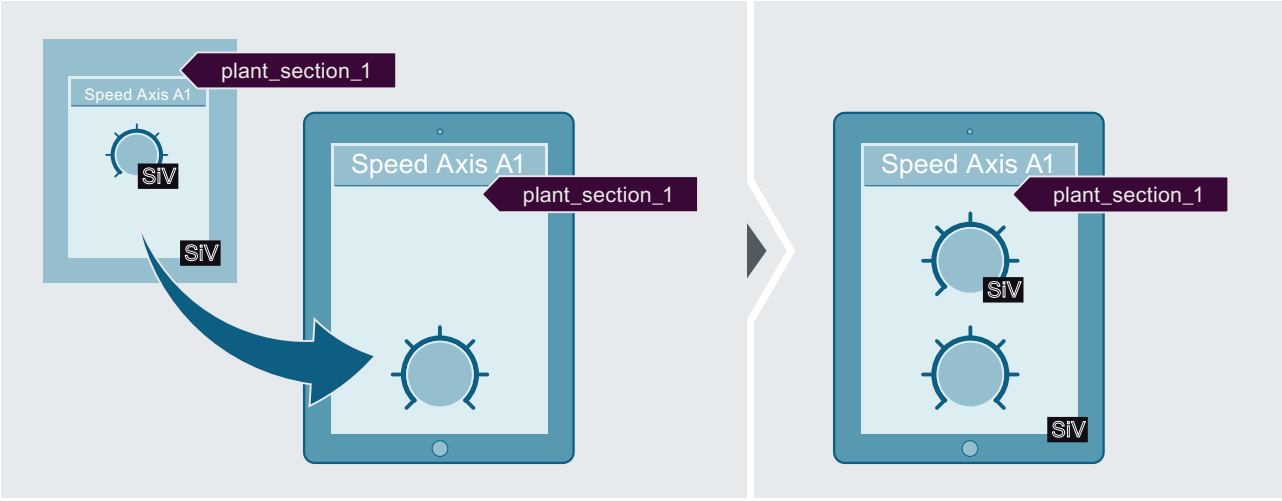
确保从一开始，生成的 HMI 对象和手动创建的 HMI 对象的命名概念不一样。这样可以防止以后进行更正。

画面和文本列表条目的相同名称

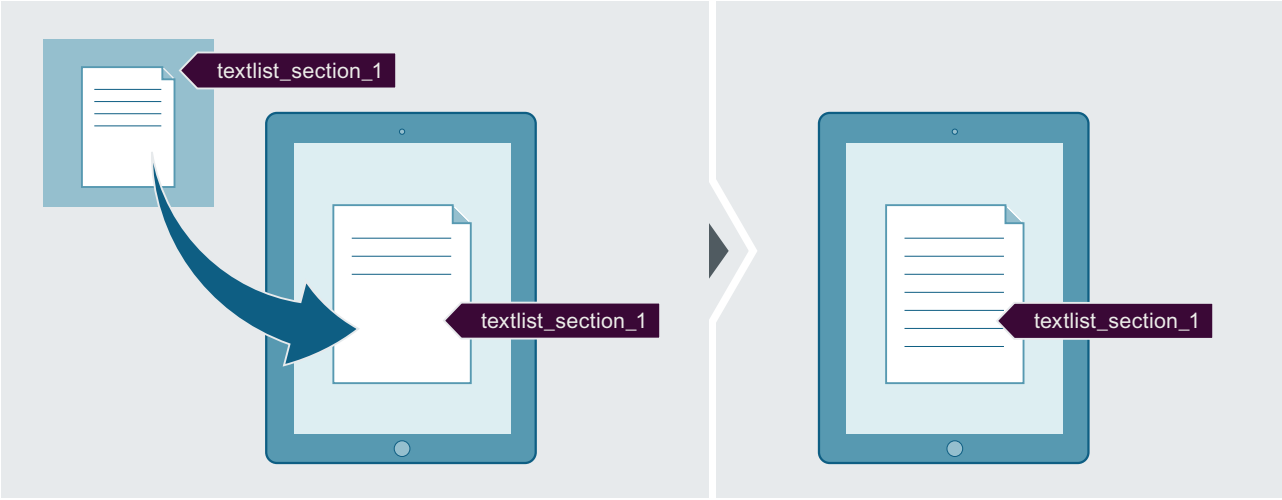
对于画面和文本列表，SiVArc 以不同的方式处理相同名称：

如果已存在名称相同的已生成画面或文本列表，则尽管命名冲突，生成期间 SiVArc 都将捕获这些画面或文本列表。记住：文本列表会输出错误消息，但画面不会输出。

下图显示了为手动创建的画面生成具有相同名称的画面。现有的画面由生成进行处理：



下图显示了为手动创建的文本列表生成具有相同名称的文本列表。两个文本列表的文本列表条目合并到现有文本列表中：



存在命名冲突时生成对象的优先级

生成的对象的优先级高于 SiVArc 对象中手动创建的 HMI 对象。生成的对象具有到 SiVArc 的固定链接，并且在每次生成期间更新。对所生成对象的手动更改将重置。

存在命名冲突时，SiVArc 会在生成过程中设定以下优先级：

1. 根据画面、变量和文本列表规则生成的对象
2. 根据复制规则生成的对象
   根据复制规则生成的对象被视为与手动创建的对象相同。首先，它们在生成过程中创建。如果与之后生成的对象存在命名冲突，根据复制规则生成的对象将以扩展名"_renamed"重命名。
3. 手动创建的对象
   如果手动创建的对象的名称与生成的对象相同，手动创建的对象将被重命名。

项目中多个 PLC 引起的命名冲突

如果由于使用多个连接 PLC 在生成期间出现命名冲突，则 SiVArc 将只生成生成期间已捕获的第一个 HMI 对象并输出错误消息。

对于具有多个 PLC 的项目中的变量的唯一分配，请使用变量运行系统设置中的“使用 PLC 名作为 HMI 变量名前缀”(Use PLC name as prefix in the HMI tag names) 选项。

为多个 PLC 生成文本列表条目时，可能会出现命名冲突，这是因为一个程序块可能位于多个 PLC 中。根据您所使用的 STEP 7 文本源，文本列表条目的生成在以下冲突条件下会做出不同的响应：

- 块参数
  其它文本列表条目以后缀的形式生成。
- 网络
  仅为评估的第一个 PLC 创建文本列表条目。要为所有后续 PLC 生成的文本列表条目将被忽略。报警和日志中会显示该错误。

导入规则时命名冲突

可通过以下选项导入 SiVArc 规则。

- 通过导入覆盖现有规则
  更新同名规则和规则组。保留所有其他规则。
- 如果规则名已存在，则重命名要导入的规则
  如果存在命名冲突，导入的规则和规则组名称会被分配一个连续编号。
- 在导入前删除所有现有规则
  导入后，规则编辑器中只包含导入文件中的规则。

用途与优势

如果在开始之前使用 SiVArc 避免组态期间命名冲突，则结果是一致的项目且无错误。根据命名概念独特差异的成功概念，可以轻松获取额外的自动化项目。

参见

导出和导入 SiVArc 规则 (页 205)

6.3.10 创建变量规则

简介

根据具体设置，SiVArc 会生成所有外部变量，或仅生成与 SiVArc 项目有关的变量。SiVArc 为背景数据块和全局数据块生成外部变量。

在变量规则中，可指定以下信息：

- 存储生成变量的文件夹名称
- 创建生成变量的变量表名称

要求

- 已经创建一个带数据块的函数块。
- 可以在相关的 PLC 变量的块界面设置“可从 HMI 访问”(Accessible from HMI) 选项。
- “变量规则”(Tag rules) 编辑器已打开。

步骤

要创建变量规则，请按以下步骤操作：

1. 创建一个变量规则。
2. 给该规则分配唯一名称。
3. 在“变量组”(Tag group) 下打开“SiVArc 表达式”(SiVArc expressions) 编辑器。
   - 要将变量组的名称定义为文本，请在引号内输入字符串。
   - 要从用户程序中取出名称，请输入一个 SiVArc 表达式。
   - 要在项目树中映射用户程序的结构，请输入 SiVArc 表达式 `HmiTag.DB.FolderPath` 。
4. 在“变量表”(Tag table) 下打开“SiVArc 表达式”(SiVArc expressions) 编辑器。
   - 指定文本或 SiVArc 表达式作为变量表的名称。
   - 要将数据块的所有变量写入变量表，请输入 SiVArc 表达式 `HmiTag.DB.SymbolicName`。
5. 要输入条件，必要时使用 SiVArc 脚本。

结果

变量规则已创建。

附加设置

优化以下设置以控制变量生成：

- “选项 > 设置> SiVArc”(Options > Settings > SiVArc) 下的变量生成模式
- 变量的运行系统设置
- 更新周期和采集类型

更改变量规则的布局

可使用拖放操作或通过快捷菜单命令布局变量规则。此功能仅在“变量规则”(Tag rules) 编辑器的列未排序、也未过滤的前提下可用。使用快捷菜单还可在过滤的编辑器中对“变量规则”(Tag rules) 重新排序。

若要使用拖放操作更改变量规则的布局，请按以下步骤操作：

1. 选择规则的第一个单元格。
2. 将规则拖放到编辑器中的所需位置。

参见

生成变量 (页 178)

变量生成的原理 (页 185)

示例：调整变量名称 (页 201)

更改 SiVArc 规则 (页 216)

生成对象的存储策略 (页 131)

在 SiVArc 编辑器中编辑视图 (页 286)

6.3.11 创建画面规则

要求

- 已创建用户程序。
- 已创建显示和操作对象的生成模板。
- 已创建画面的生成模板。
- 已打开 “画面规则”(Screen rules) 编辑器。
- 列“PLC”和列“HMI 设备”(HMI device) 可见。

步骤

要定义用于生成显示和操作对象的画面规则，请按以下步骤操作：

1. 创建画面规则。
2. 给该规则分配唯一名称。
3. 在"PLC"下，选择要应用画面规则的控制器。
   如果未选择任何控制器，则画面规则适用于项目中的所有控制器。
4. 选择为其生成 HMI 对象的程序块。
5. 在“画面对象"(Screen object) 下，选择显示和操作对象的生成模板。
6. 在“画面"(Screen) 下，选择要在其中生成对象的画面的生成模板。
   如果为生成模板存储定位方案，请在“布局字段"(Layout field) 下选择定位区域。如果未指定定位区域，则生成的 HMI 对象将根据 SiVArc 定位方案在画面中定位。
7. 在"HMI 设备"(HMI device) 下，选择要应用画面规则的 HMI 设备。使用工具栏图标显示 HMI 设备的设备类型。
   如果未选择任何 HMI 设备，则画面规则适用于与所选控制器相连的所有 HMI 设备。
   如果此规则只对符合特定条件的对象或程序块执行，则使用 SiVArc 脚本编写“条件"(Condition) 下的相应表达式。

还可以通过拖放操作从库中添加程序块和模板。

结果

生成可视化时，在指定的画面中生成对象。

如果已在画面规则中选择了一个定位区域，则 HMI 对象将定位在此区域内，而不是定位在布局字段内。所使用的布局字段取决于画面规则的生成顺序以及布局字段的索引。

参见

生成可视化 (页 219)

用户程序中支持的对象 (页 110)

示例：创建带条件的画面规则 (页 198)

示例：组织画面和文本列表规则 (页 200)

SiVArc 规则 (页 166)

创建 SiVArc 规则 (页 170)

规则处理： (页 175)

更改 SiVArc 规则 (页 216)

在 SiVArc 编辑器中编辑视图 (页 286)

6.3.12 创建文本列表规则

简介

文本列表规则指定为程序块生成哪个文本列表。

要求

- 已创建用户程序。
- 文本列表的生成模板存储在相应文件夹中的库中。
- “文本列表规则”(Text List Rules) 编辑器已打开。

步骤

要定义文本列表规则，请按以下步骤操作：

1. 创建文本列表规则。
2. 给该规则分配唯一名称。
3. 选择所需的程序块。
4. 选择所需的文本列表生成模板。
5. 要输入条件，必要时使用 SiVArc 脚本。

可通过拖放操作添加程序块或主副本。

结果

生成可视化时，在“文本和图形列表”(Text and Graphic Lists) 编辑器中创建文本列表。

参见

示例：组织画面和文本列表规则 (页 200)

示例：为文本列表创建生成模板 (页 148)

示例：为块参数的文本列表创建生成模板 (页 151)

SiVArc 规则 (页 166)

创建 SiVArc 规则 (页 170)

规则处理： (页 175)

更改 SiVArc 规则 (页 216)

在 SiVArc 编辑器中编辑视图 (页 286)

创建报警规则 (页 195)

6.3.13 创建报警规则

简介

报警规则是 SiVArc 生成的组成部分。当创建报警规则时，使用 STEP 7 编程 PLC 块以在 HMI 设备中生成对象。因此，报警规则作为将 STEP 7 与 HMI 设备连接起来的抽象类型。

条件

- 含主副本和类型的全局库和项目库。
- 生成的 HMI 对象存储在库中。

步骤

要在 SiVArc 中创建报警规则，请按以下步骤操作：

1. 在 SiVArc 文件夹下，单击“报警规则”(Alarm rules)。
2. 要在“报警规则”(Alarm rule) 编辑器中添加新规则，单击“创建新规则”(Create new rule)。默认情况下，第一个规则名称显示为报警规则。连续的规则名称使用递增数字后缀显示，如 <Alarm rule_1>
3. 在“规则触发器”(Rule Trigger) 列中，浏览至相应的程序块或程序类型。
4. 在“报警/类别/组的主副本”(Master copy of Alarms/Classes/Groups) 列中，浏览至主副本的单个或多个对象。
5. 在“条件”列中，为要执行的规则设置条件。这是 SiVArc 种的选项。

说明

- 不会处理组态的相似规则。
- 在程序块中，启动快捷菜单命令“转到引用对象”(Go to referenced object)。所选程序块在“规则触发器”(Rule Trigger) 区域中突出显示。在 SiVArc“插件”(Plug-in) 编辑器中，选择报警规则，这将显示块的过滤的报警规则列表。
- 如果在“报警/类别/组的主副本”(Master copy of Alarms/Classes/Groups) 列下进行了单选，则可以使用“转到引用对象”。

结果

在 SiVArc 生成期间，只有满足条件才能创建报警对象，并且成功处理规则。您可以在“生成概览”(Generation Overview) 中查看生成的对象。有关生成概览的更多详细信息，请参见主题 生成可视化 (页 209)。

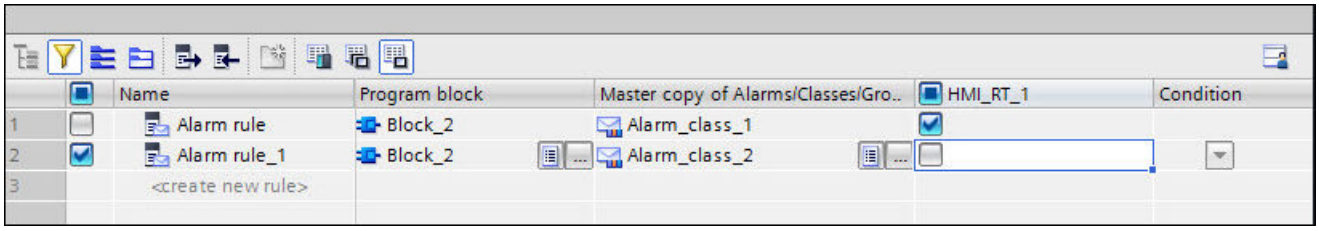
常规注意事项

- 在关闭编辑器并重新打开后，在“报警规则”(alarm rule) 编辑器中创建的规则将仍然保存。
- 您可以在“报警规则”(alarm rule) 编辑器中执行复制、粘贴、拖放和撤消操作
- 在编辑“报警规则”(alarm rule) 编辑器选项时，如果修改或删除了库对象，“报警规则”(alarm rule) 编辑器中的引用对象将突出显示为粉红色来显示无效的引用。
- 您可以通过选择“报警规则”(alarm rule) 编辑器中的规则来分组多个规则，然后右键单击报警，选择“添加新的规则组”(Add a new rule group)。
- 您可以在“规则组”(Rule group) 中添加新的报警规则。
- 在分组报警规则时，将自动显示“与”条件。您可以从运算符下拉列表中选择不同的条件运算符。或者，您可以手动定义组的条件。
- 在选择程序块和主副本时，SiVArc 支持智能提示。

设备特定规则

设备特定报警规则符合 HMI 设备特定的报警规则。“报警规则”(Alarm rule) 编辑器在工具栏中为您提供“显示或隐藏 HMI 设备”(show or hide HMI device) 选项。要组态设备特定规则，请按以下步骤操作：

1. 单击“显示或隐藏 HMI 设备”(show or hide HMI device) 选项。可用的 HMI 设备显示时带有复选框选项。
2. 您可以为特定报警选择或取消选择 HMI 设备。通过这样做，“报警规则”(Alarm rules) 在 SiVArc 生成过程中仅适用于选定的 HMI 设备。



说明

- 默认情况下，选择第一个可用的 HMI 设备。
- 您也可以选择或取消选择 PLC 设备。

使用属性选项选择设备类型

为便于在 PLC/HMI 设备列表中选择 PLC/HMI 设备，使用“报警规则”(alarm rule) 编辑器中的“属性”(Properties) 选项卡。在“报警规则”(Alarm rule) 编辑器中，“属性”(Properties) 选项卡提供了选择或取消选择 PLC/HMI 设备的选项。通过这样做，报警规则仅适用于选定的 PLC/HMI 设备。

参见

生成可视化 (页 209)

6.3.14 创建复制规则

简介

在标准化操作员控制和监视解决方案中，HMI 对象通常集中创建，然后作为全局库分配给组态工程师。

要求

- 含类型和主副本的全局库。
- 要生成的 HMI 对象存储在库中。

步骤

要使用复制规则通过 SiVArc 创建 HMI 对象，请按以下步骤操作：

1. 打开含主副本和类型的全局库。
2. 将已打开的全局库的内容同步到项目库。
3. 为待生成的每个 HMI 对象创建复制规则。
   或
   为库文件夹创建复制规则。
4. 给规则分配唯一名称。

结果

生成过程中，HMI 对象创建在项目树的相应文件夹中。为规则中指定的每个 HMI 设备创建 HMI 对象。

常规注意事项

- 在 SiVArc 生成过程中，如果 HMI 设备中的默认报警对象的名称和复制规则生成的对象相同，应用程序将显示以下错误消息：
  “对象 <<Alarm_name>> 被其他 SiVArc 编辑器修改。Copy_rule 生成器不能修改此对象。”
  (Object <<Alarm_name>> was modified by other SiVArc editor. Copy_rule generator can’t modify this object.)
- “复制规则”(copy rule) 编辑器允许用户选择和取消选择需要为 HMI 设备生成的规则。
- 可以将主副本中的对象拖放到“库对象”列中。
- “复制规则”(copy rule) 编辑器支持智能提示。

说明

复制报警规则

使用“复制规则”(copy rule) 编辑器组态的规则将在 SiVArc 生成过程中进行处理并生成 HMI 对象。

参见

使用复制规则 (页 182)

更改 SiVArc 规则 (页 216)

在 SiVArc 编辑器中编辑视图 (页 286)

6.3.15 示例：创建带条件的画面规则

示例场景

在标准化用户程序中使用相同的程序块来控制阀门或电机。

要求

根据程序块的使用，在用户界面中标记为 "Open Valve" 的按钮或将要生成的 "Start Motor"。阀门按钮总是放置在画面的顶部，电机按钮总是放置在画面的底部。

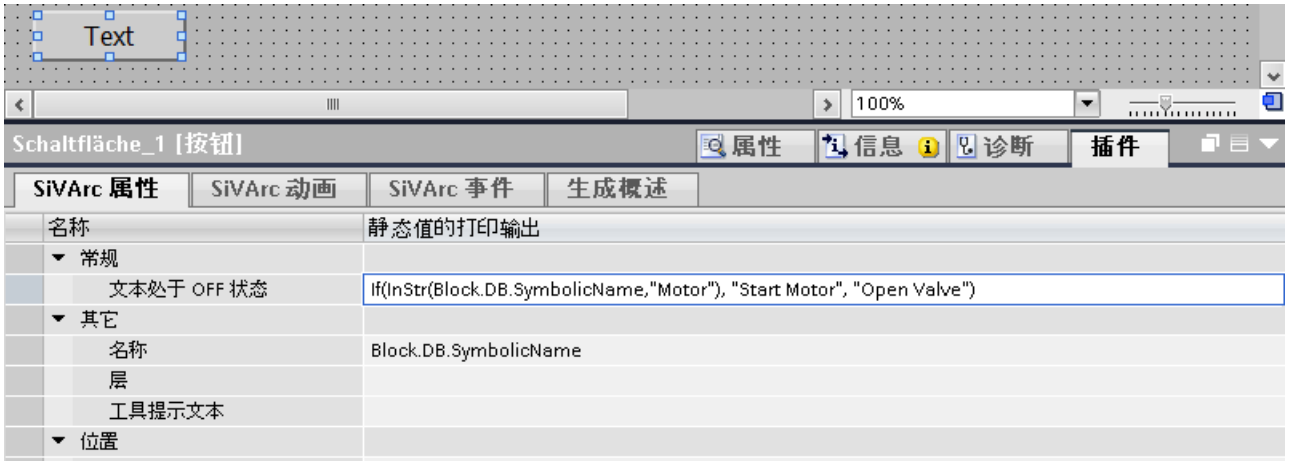
控制画面规则，以便为程序块的每个用例生成不同的按钮。

执行规则

组态工程师使用条件来控制按钮的标记和画面规则。SiVArc 为程序块的每个用例生成不同的按钮。为此，组态工程师使用条件来访问块实例的数据块符号名称：

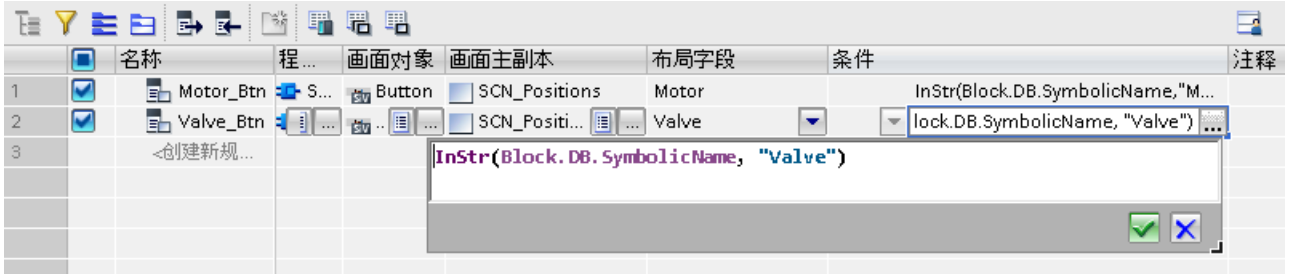
- 如果程序块用于阀门控制，则数据块的名称为“Valve_DB_Instance_<n>”。
- 如果程序块用于电机控制，则数据块的名称为“Motor_DB_Instance_<n>”。

要控制按钮的标记，组态工程师将在 SiVArc 属性“文本关闭”(Text OFF) 中对条件进行编程：



为了控制按钮的定位，组态工程师将使用布局字段 "Valve 1/1" 和 "Motor 1/1" 创建定位方案。组态工程师将此方案存储在过程画面的生成模板中，并创建以下两个画面规则。

要控制画面规则的执行，组态工程师需要对条件进行编程。条件指定仅当背景数据块的符号名称包含字符串“Valve”或“Motor”时才执行规则。



参见

创建画面规则 (页 192)

6.3.16 示例：组织画面和文本列表规则

示例场景

多个组态工程师在一家工程公司的大型项目上工作。每个工程师都有独立的负责区域和专业领域。

要求

画面规则将按功能单独处理。为一名组态工程师分配了配方的操作和表示。另一名同事设置了所有诊断功能。

根据需要，这些功能将在生成期间共同启用或禁用。

一名同事负责 HMI 设备的布局。她应当能够快速了解所使用的 HMI 设备。

执行规则

在规则编辑器中创建一个功能特定的文件夹结构，例如：

- 起始画面
  - ComfortPanel 19"
  - ComfortPanel 19" 立式
  - MobilePanel 277 8"
  - PC 站
- 诊断画面
  - ...
  - ...
- 配方画面
  - ...
  - ...

要检查画面和文本列表规则，控制器程序员使用工具栏来仅使规则编辑器中的相关列可见。在文件夹中，程序员根据“程序块”(Program block) 列对规则进行排序，并按相关函数块进行过滤。

共同启用或禁用主要组可为每个工厂生成项目。

启用或禁用 HMI 设备类型的相应组，以便为每个设备生成项目。

使用注释列为组态工程师分配组，这些组包含其专业领域的规则。每个工程师将各自的组复制到测试项目中。工程师编辑现有规则并创建新规则。管理层发布后，规则将再次导入并重新用于其它项目。

负责布局的组态工程师为每个画面类型的相应 HMI 设备分配模板。

参见

创建画面规则 (页 192)

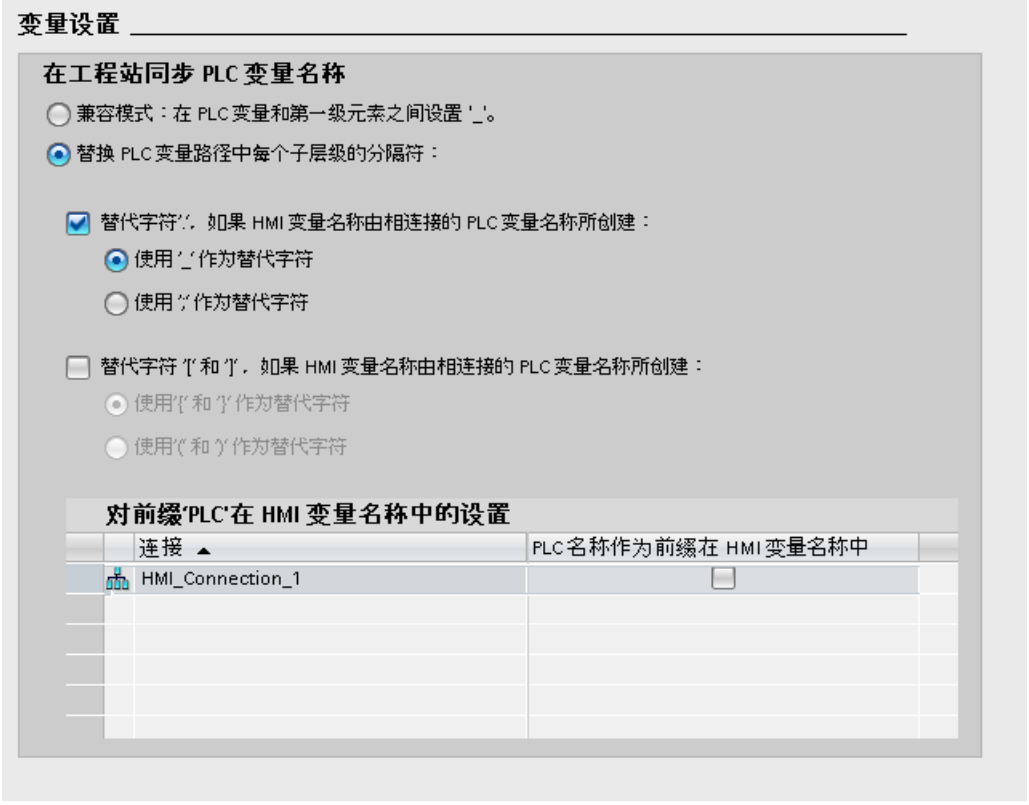
创建文本列表规则 (页 194)

6.3.17 示例：调整变量名称

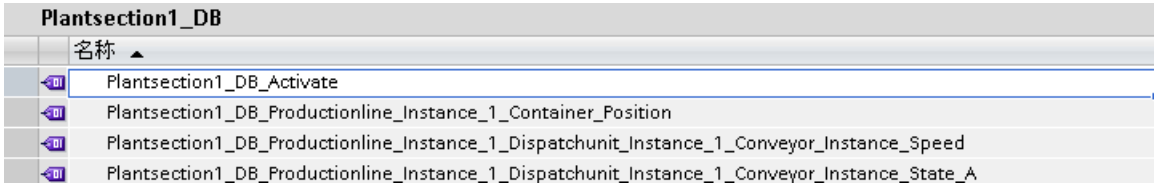
示例场景

在控制程序中使用新的、结构化的 PLC 数据类型。为确保 PLC 变量与外部 HMI 变量同步，组态工程师可以更改一个 HMI 设备的变量设置。因此，变量的运行系统设置不同于项目中的 HMI 设备：

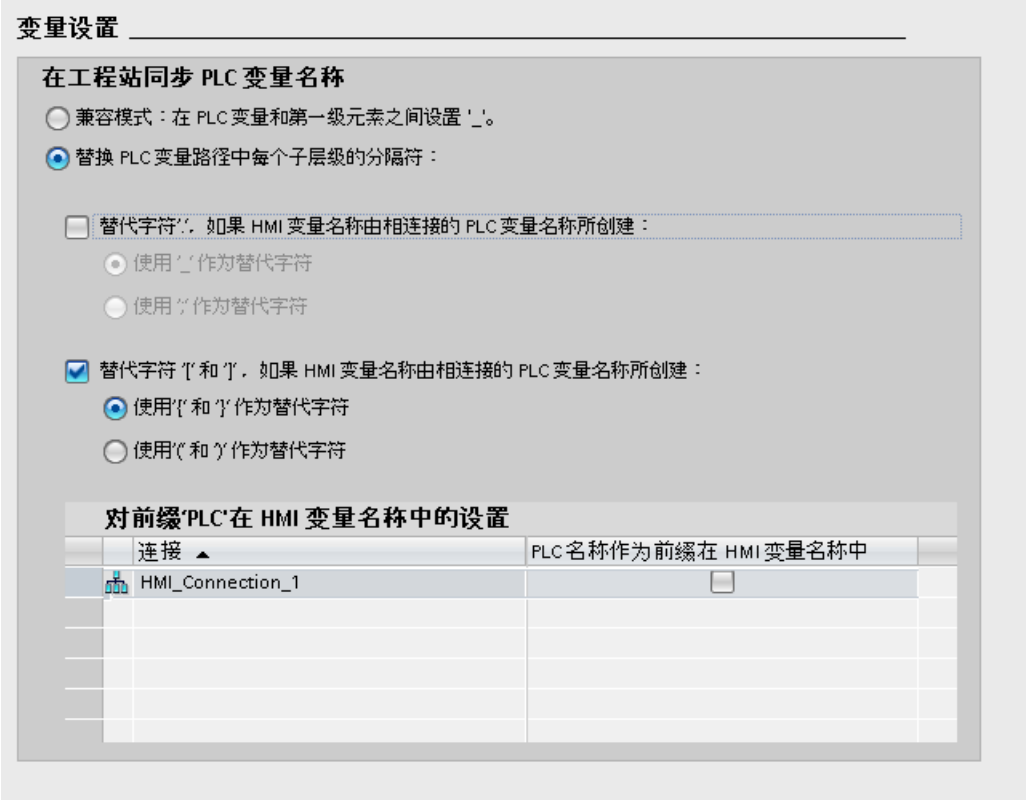
HMI 设备 1 的运行系统设置：



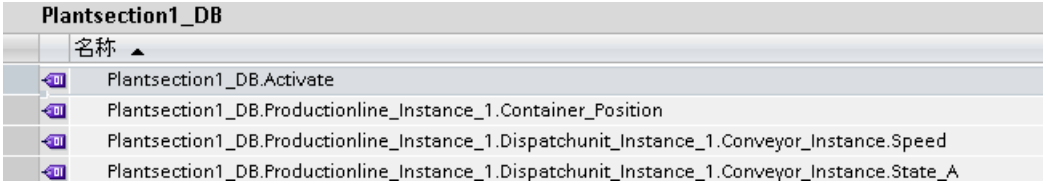
同步 HMI 变量：



HMI 设备 2 的运行系统设置：



同步 HMI 变量：



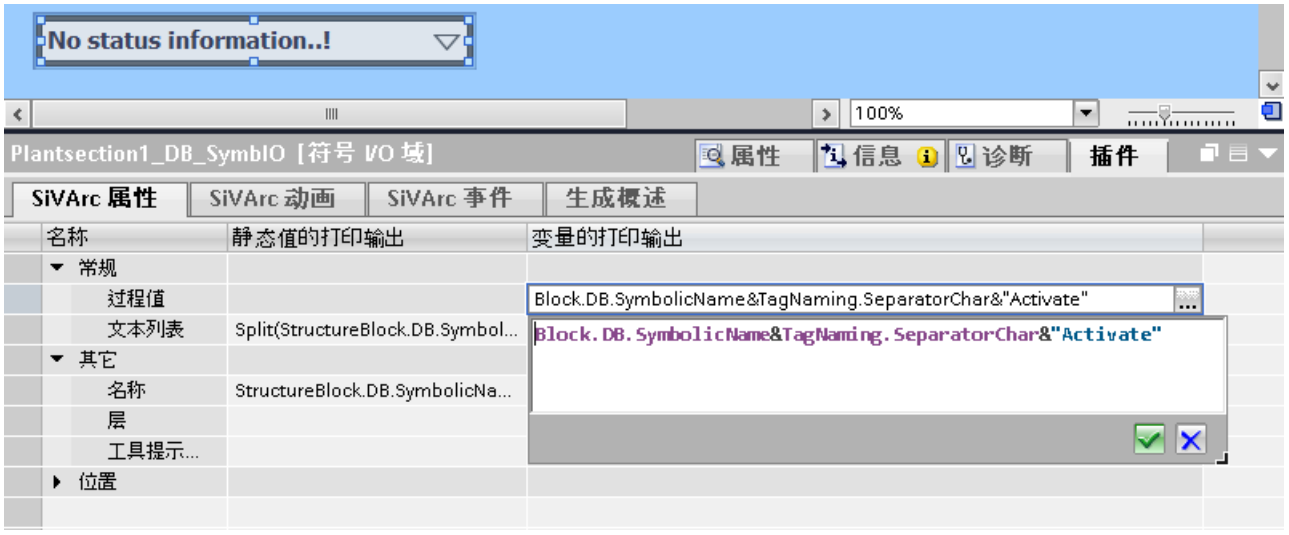
要求

即使在项目中使用不同的同步模式，生成的显示和操作对象的互连也将起作用。

执行规则

在定义触发器变量的 SiVArc 表达式中，所选变量同步使用 SiVArc 对象 `TagNaming` 进行寻址。

以下 SiVArc 表达式在生成可视化后产生变量名称：HMI 设备 1 的“Plantsection1_DB.Activate”和 HMI 设备 2 的“Plantsection1_DB_Activate”。



参见

TagNaming (页 253)

变量生成的原理 (页 185)

SiVArc 表达式 (页 113)

6.3.18 编辑和管理 SiVArc 规则

简介

在复杂的 SiVArc 项目中，存在大量的 SiVArc 规则。因此，用户应以一种清晰的方式对 SiVArc 规则进行排序和组织，并使其可在库中使用。

可通过多种功能来显示结构清晰的规则：

- 过滤功能
- 分组和排序功能

- 快捷菜单
- 拖放功能

要分析规则，需通过快捷键菜单的“跳转至...”(Go to...) 命令在 SiVArc 编辑器、用户程序和生成模板之间导航。

创建一个 SiVArc 规则

1. 单击“添加规则”(Add rule)。
   在表格编辑器中创建一个新行。
2. 给该规则分配唯一名称。
3. 通过拖放操作从库中插入程序块和生成模板。

或者，输入要引用的对象的前几个字母。SiVArc 给出了可引用且引用路径中包含此字母序列的对象的列表。

通过拖放操作在“名称”(Name) 下插入程序块时，将使用所选程序块创建新的规则。

SiVArc 规则分组

如果根据自定义标准对 SiVArc 规则分组，则能够更好地掌握 SiVArc 项目的概况：

- 可同时启用和禁用规则组中包含的规则。
- 一个规则组的条件适用于该组内的所有规则。可通过操作数设置特殊案例。
- 用户可按需求移动和排列组内或组外的各个规则。
- 跨组移动规则时，可应用为当前组设置的选项。

按以下步骤创建一个规则组：

1. 选择需要组的各个规则。
2. 在快捷菜单中选择“添加新规则组”(Add new rule group)。
   选定的规则移到一个新组。
3. 为规则组命名。

要创建子组，请完全遵照组内所用的方式编辑所需规则。

若要一次性打开或关闭所有规则组，请单击“全部展开”(Expand all) 或“全部折叠”(Collapse all) 按钮。

说明

过滤规则组

只有当组的所有规则都满足过滤条件时，才会在过滤期间显示组。

SiVArc 规则的嵌套分组

可以为一个规则组设置条件。可使用规则组按需对 SiVArc 规则进行排序，例如可按工厂结构、画面结构或 WinCC 主题排序。

规则组的应用示例

SiVArc 项目中的所有画面规则均可按以下画面类型分组：

- 起始画面
- 诊断画面
- 配方画面

这样一来，用户便可按特定主题将 SiVArc 组态分配给某部门的组态工程师。

可使用分组条件指定哪些变量必须包含在程序块中才能使相应的规则组包含在生成过程中。

可通过使用组内某个规则的条件操作数生成多种不同的画面，具体取决于所包含的参数。这样，就可以凭借一个 SiVArc 项目和几项规则对多个工作区域进行可视化。

使用库中的 SiVArc 规则

若要集中、一致地更新多个项目的 SiVArc 规则，请将 SiVArc 规则或规则组以主副本形式存储在库中。如果项目中已存在同名的 SiVArc 规则，可覆盖现有规则或者创建新规则。

如果使用库中的规则覆盖现有规则，则以下条件适用：

- SiVArc 会检查上一个生成过程中生成的 HMI 对象，并将这些对象纳入生成。
- 对所生成对象的手动更改将被覆盖。

参见

在 SiVArc 编辑器中编辑视图 (页 286)

6.3.19 导出和导入 SiVArc 规则

简介

SiVArc 规则和规则组可导出到 MS Excel 以及从 MS Excel 导入。

导出和导入功能适用于每个 SiVArc 编辑器或整个项目。

还可以将组外的个别规则直接从 MS Excel 工作表复制到 SiVArc 编辑器中，反之亦然。

说明

导出和复制规则

复制和粘贴规则时，只能插入可见列。

导出 SiVArc 编辑器的 SiVArc 规则

1. 打开所需的 SiVArc 编辑器。
2. 单击编辑器工具栏中的“导出”(Export) 按钮。
   将打开一个对话框。
3. 选择所需的存储位置以及导出文件的名称。
4. 单击“确定”(OK)。

创建导出文件。

导出项目的 SiVArc 规则

1. 在项目树中选择“公共数据 > SiVArc”(Common data > SiVArc)。
2. 在快捷菜单中选择“导出所有规则”(Export all rules)。
   将打开一个对话框。
3. 选择所需的存储位置以及导出文件的名称。
4. 单击“确定”(OK)。

创建导出文件。

导出文件结构

在每个 SiVArc 编辑器的工作簿中创建带有导出的 SiVArc 规则的电子表格。电子表格具有下列标题：

- ScreenRules
- AlarmRules
- TagRules
- TextlistRules
- CopyRules

导入规则

将 SiVArc 规则导入单个 SiVArc 编辑器时请注意：

- 导入文件的格式必须为“*.xlsx”。
- 如果导入文件中只有一个电子表格，则导入此电子表格时无需考虑其名称。
- 只有在导入文件的电子表格重命名或删除后，才能使用对话框选择所需的电子表格。
  - 要导入重命名的电子表格，请在对话框中单独确认导入。
  - 要从导入中排除电子表格，请在对话框中跳过该电子表格。如果在导入前将其删除，仍需要在对话框中跳过一个空视图。

说明

导入期间，请确保项目中已设置的组态语言与导入文件中使用的语言相同。

导入选项

可通过以下选项导入 SiVArc 规则。

- 通过导入覆盖现有规则
  现有规则编辑器中的规则/规则组将与导入文件的规则/规则组层级对齐。
- 如果规则名已存在，则重命名要导入的规则
  如果存在命名冲突，则导入的规则和规则组将附加一条下划线。例如，将“Rule”重命名为“Rule_1”。
- 在导入前删除所有现有规则
  导入后，规则编辑器将删除现有规则，并显示导入文件中的规则/规则组。

导入规则组

如果某个规则组不能具体分配，那么它将在以下条件下加入到编辑器的第一个层级中：导入文件包含循环引用或导入文件中缺少高级组。

如果现有规则未在导入过程中重命名，那么导入文件中包含的规则组将被多次覆盖，即每次都将被导入文件底部列出的规则组覆盖。

将 SiVArc 规则导入 SiVArc 编辑器

1. 打开所需的 SiVArc 编辑器。
2. 单击编辑器工具栏中的“导入”(Import) 按钮。
   将打开一个对话框。

3. 选择所需的导入文件和导入选项。
   如果导入文件包含多个电子表格，将打开一个对话框。
4. 选择所需电子表格。
5. 单击“确定”(OK)。

将 SiVArc 规则导入项目

1. 在项目树中选择“公共数据 > SiVArc”(Common data > SiVArc)。
2. 在快捷菜单中选择“导入所有规则”(Import all rules)。
   将打开一个对话框。
3. 选择所需的导入文件和导入选项。
4. 单击“确定”(OK)。

结果

将在 SiVArc 编辑器中创建 SiVArc 规则。完成消息中包含日志文件的链接。另外，导入日志位于“公共数据 > 日志”(Common data > Logs) 下。

6.3.20 为 SiVArc 项目设置专有技术保护

简介

用户的 SiVArc 项目包含通过 SiVArc 脚本功能单独创建的 SiVArc 生成规范。为保护整个项目中的 SiVArc 表达式，请为项目启用专有技术保护。

专有技术保护仅限 SiVArc 编辑器，不包括 SiVArc 的设置。库和巡视窗口中的 SiVArc 选项卡以及生成的对象都不会受影响。

密码

为专有技术保护分配一个密码。密码长度必须至少为 8 个字符，其中包含以下字符类型：

- 大写字母和小写字母
- 特殊字符
- 数字

设置专有技术保护

1. 在项目树中选择“公共数据 > SiVArc”(Common data > SiVArc)。
2. 在快捷菜单中选择“专有技术保护 > 启用”(Know-how protection > Activate)。
   将打开一个对话框。
3. 指定密码。
4. 保存项目。

还可以使用快捷菜单来编辑密码和解除专有技术保护。

结果

为所有 SiVArc 编辑器启用了专有技术保护。如果要在项目树中打开一个 SiVArc 编辑器（在 STEP 7 中或从其他编辑器跳转），系统将提示用户输入密码。SiVArc 规则导入和导出时，也会启用专有技术保护。

6.4 生成可视化

6.4.1 关于生成的基础知识

定义

生成可视化时，根据用户程序的文本和结构生成 HMI 对象。此外，还可以在生成期间不参考用户程序而复制 HMI 对象。

与基于库元素复制 HMI 对象不同，通过 SiVArc 生成的产品将再次被后续生成捕获和调整。

生成阶段

可以使用 SiVArc 在多个阶段生成可视化。这样就可以将项目从一次生成改进到另一次生成。在此过程中，将不断清理和更新现有生成。

SiVArc 可区分首次生成和后续生成。后续生成基于首次生成。手动更改，例如，生成的显示和操作对象的重新定位将保留用于后续生成。

可以触发一次完全新生成，或者更改用于后续生成的设备选择。

跨设备生成可视化

如果包含多个 PLC 和多个操作员面板的项目发生个别更改，最好为各个设备生成可视化，例如，更换设备或排除故障时。生成和下载时间会相应缩短。可用一个画面规则或一个画面规则组为多个 HMI 设备生成 HMI 对象。

这样就可以在工厂的大型 SiVArc 项目中分别为所有设备或设备类型更新和优化过程画面。以下功能可在组态过程中提供帮助：

- 在 SiVArc 编辑器中使用工具栏隐藏和显示设备特定的各个列
- 将各项规则分配给连接的设备和控制器
- 显示画面规则中的设备类型，以方便分配匹配的定位方案
- 在画面规则的巡视窗口中显示设备类型，具体取决于 PLC

如果使用 Unified 设备模板副本的画面规则在进行 SiVArc 生成时包含 Unified 和 HMI 设备，则仅会在 Unified 设备上生成画面。仅会在 Unified 设备上生成 Unified 已组态画面。

恢复 HMI 设备的缺失分配

对项目中的设备进行复制、粘贴和重命名时，设备会保留运行系统名称。重命名后，不能移除对该运行系统名称的文本交叉引用。

1. 在“设备和网络 > 网络视图 > 网络概览 > 设备”(Devices and networks > Network view > Network overview > Device) 下，可展开相关的 HMI 设备条目。
2. 相应地调整运行系统设备名称。

在后续生成中，分配将重新生效。

生成的范围

SiVArc 提供了几个选项来控制生成范围：

- “选项 > 设置> SiVArc”(Options > Settings > SiVArc) 下的变量生成模式
  如有必要，可以将变量的生成限制为正在使用的变量
- 在规则编辑器中免除生成的规则
- 在生成对话框中免除生成的设备

如果不创建任何 SiVArc 规则，则 SiVArc 仅生成外部变量。

首个设备相关的生成

如果项目包含多个 HMI 设备或连接的 PLC，则 SiVArc 会为所选 HMI 设备和 PLC 生成可视化

在项目中初次生成开始时，将显示站选择对话框。在站选择对话框中，选择 SiVArc 将生成可视化视图的设备。

SiVArc 会针对设备逐一生成可视化。

- 如果某个设备无法生成，则 SiVArc 会继续处理下一个设备。
- 如果取消生成，则设备仍完全生成可视化。

首次生成后，站的可用选择将被冻结。后续的每次生成均基于此选择。

相同的名称

如果手动创建的 HMI 对象的名称相同，则手动创建的 HMI 对象将被重命名。新生成的对象总是由 SiVArc 在要生成的名称下创建。

如果采用 SiVArc 所生成名称的手动创建 HMI 对象已存在，则将对现有对象追加后缀 "_renamed"。如果该名称也已占用，则名称会自动递增。

如果在使用多个已连接 PLC 生成期间出现命名冲突，则 SiVArc 将只生成生成期间已捕获的 HMI 对象并输出错误消息。

说明

HMI 设备运行系统设置

生成多个 PLC 的变量时，系统将判断 HMI 设备运行系统设置中的“PLC 前缀”(PLC prefix) 选项。

确保每台 PLC 的运行系统设置中的“PLC 前缀”(PLC prefix) 选项均已启用。否则将取消 SiVArc 生成。

说明

例外

画面和文本列表存在名称相同情况时的行为如下：

如果已存在名称相同的已生成画面或文本列表，则无论命名是否冲突 SiVArc 都将再次生成这些画面或文本列表。记住：文本列表会输出错误消息，但画面不会输出。

首次生成的结果

SiVArc 会根据 SiVArc 规则生成 HMI (Unified) 对象，并根据组态对其进行保存。所生成对象的插件属性将使用已组态数据进行更新。

更新的设备相关生成

如果未在站选择对话框中为下一次生成而再次启用设备，则项目中会保留生成的对象和手动更改。

说明

要删除在下次生成期间不再启用的 PLC 的已生成对象，需删除 PLC 与 HMI 设备之间的连接。

如果删除了 HMI 设备与控制器之间的现有连接，则会在站选择的对话框中发出警告。当删除 PLC 和 HMI 设备之间的连接时，下一次生成期间将删除所有相关的生成对象。

使用新站选择进行生成。

项目中初次生成时，将始终显示站选择对话框。

下次开始生成时，将不再显示站选择对话框。SiVArc 随后会生成与之前生成期间相同的 HMI 和 PLC。要更改设置，请按以下步骤操作：

1. 在项目导航中选择项目或设备。
2. 单击“可视化 (SiVArc)”(Visualization (SiVArc))。从下列选项中选择：
   - 生成可视化 (SiVArc)
   - 使用站选择进行生成
   - 清除可视化 (SiVArc)

   或者，按下快捷键 <ALT+Shift+G>。
   组态包含多实例 DB 的面板时，如果启用“公共数据 > SiVArc 设置 > 警告设置 > 画面对象已存在时隐藏警告”(Common data > SiVArc Settings > Warning Settings > Hide warnings if screen object already exists)，则生成期间不会显示警告消息。默认情况下，会禁用“画面对象已存在时隐藏警告”(Hide warnings if screen object already exists) 复选框。默认情况下，警告消息将在“公共数据 > 日志”(Common data > Logs) 文件夹下提供。

对控制器的更改

如果删除了已用于生成可视化的 PLC，则在下次生成期间，所有使用该 PLC 生成的对象将被删除。

如果删除用户程序中的块调用并重新生成，则已为该块调用生成的对象将被删除。

限制 PLC 编译的检查

可以通过“SivarcDisableCompileClean”文件限制 PLC 编译的检查。在这种情况下，即使 PLC 编译出错，也会运行生成过程。

为此，请在包含“Siemens.Simatic.Sivarc.dll”文件的 SiVArc 安装目录中使用名称“SivarcDisableCompileClean”创建一个空文件。

说明

如果名称为“SivarcDisableCompileClean”的文件未包含在 SiVArc 安装目录中，并且 PLC 编译有错误，则会取消生成 SiVArc。

如果“SivarcDisableCompileClean”文件包含在 SiVArc 安装目录中，并且 PLC 编译没有任何错误，则会运行 SiVArc 生成过程。

参见

生成可视化 (页 219)

生成期间避免冲突 (页 188)

后续更改 (页 213)

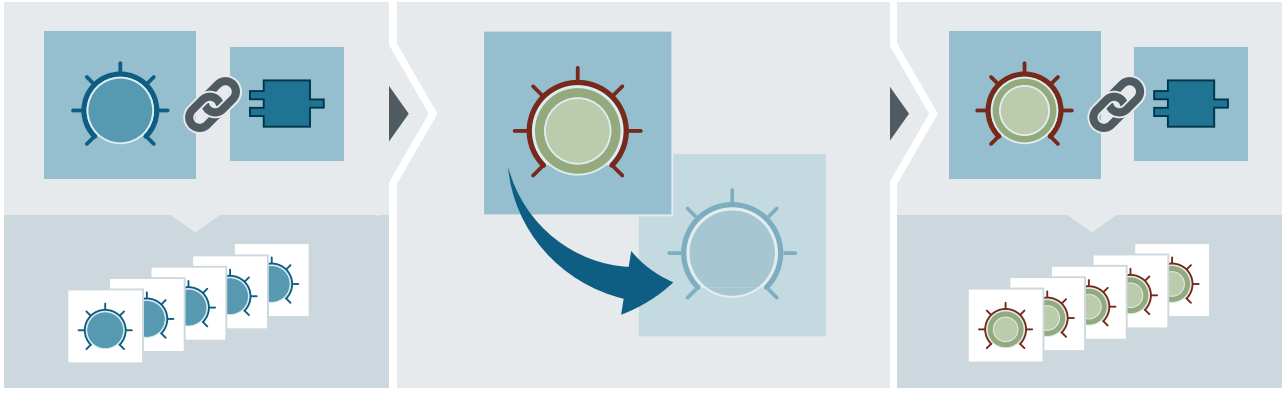
6.4.2 后续更改

6.4.2.1 更改生成的对象

应用参考

可以使用生成模板或在用户程序中，通过 SiVArc 集中更改生成的显示和操作元素。下一次生成期间，生成对象的手动更改将丢失。

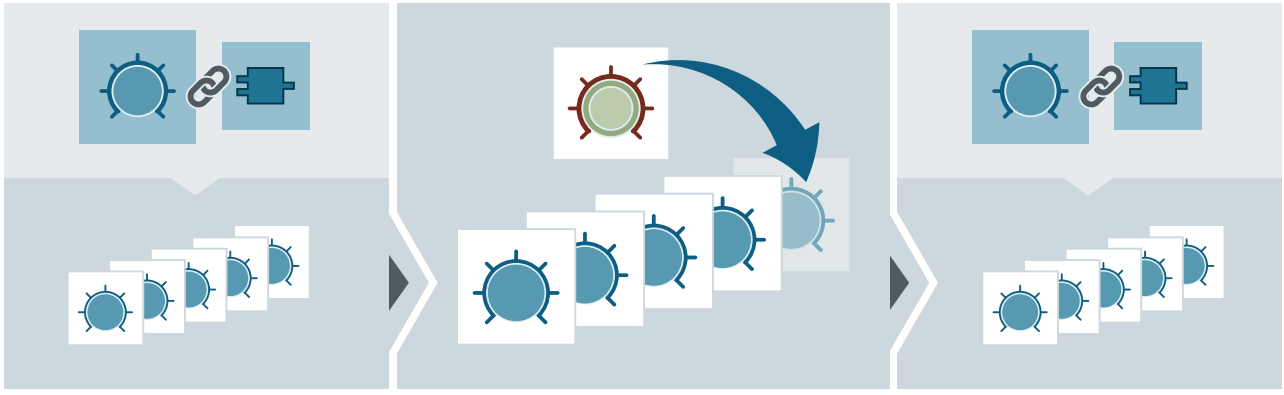
如果显示和操作元素将在之后以图形方式进行调整，则可视化工程师只改变相关的生成模板。组态工程师将模板存储在与前一个名称相同的库中。下一次生成期间，显示和操作元素以图形方式进行调整，无需额外的组态。



生成对象的 SiVArc 引用

生成的 HMI 对象对来源于其中的 SiVArc 规则具有永久引用。该引用在每次生成新对象时都会得到以下结果：

- 不再引用 SiVArc 组态的对象将被删除。引用丢失，例如，删除规则时。
- 生成规范发生更改的对象会进行更新。
- 除符合 SiVArc 的手动更改外，对生成对象的所有手动更改都将被撤销。



符合 SiVArc 的手动更改

生成的显示和操作元素的以下更改对于所有后续生成仍然有效：

- 生成对象的首次全新定位
  即使该位置采用自有的定位方案定义，新的定位仍然有效。即使更改了定位方案，手动组态的位置在下次生成后也仍会保留。
  仅对于具有固定位置的画面对象，手动更改后的位置会在下一个生成过程时重置为保存的固定位置。
- 除了面板和画面窗口，尺寸和旋转角度的更改仍然有效。
- 可以在生成的画面窗口中更改显示的画面。
- 手动更改的文本列表条目在后续生成操作后仍保留。

生成矩阵

使用生成矩阵进行的更改将随后续生成而保留：

- 用户可使用生成矩阵来为其它画面生成对象。
- 用户可使用生成矩阵来为其它设备生成画面。

手动创建的 HMI 对象

手动创建的 HMI 对象不包含在 SiVArc 生成中，除非存在命名冲突。

如果在项目中通过复制和粘贴重复使用生成的显示和操作元素，则这些会被认为是手动创建的对象并且会丢失其 SiVArc 引用。如同所有其它手动创建的画面对象一样，复制的对象永远不会被 SiVArc 删除。

说明

从画面的生成模板生成显示和操作元素

如果要通过复制和粘贴的方式来重复使用画面对象，则只能使用画面主副本生成以外的画面对象。

生成的显示和操作元素的名称更改

如果更改了生成的 HMI 对象的名称，则在下次 SiVArc 生成中将重新创建该对象并且互连。已更改名称的对象也包含在项目中。

只能使用生成模板中的 SiVArc 表达式或用户程序中的文本源来更改生成的显示和操作元素的名称。生成对象的名称在下一次生成期间相应更新。

手动覆盖的文本列表条目

当覆盖生成的文本列表条目时，更改后的文本列表条目会在下一次生成期间（仅限主副本的默认文本）保留。

如果文本列表的文本是通过 STEP 7 中的网络文本定义或符号表生成的，更改该文本时，则所做更改会由下次生成所覆盖。

以下示例说明了 SiVArc 如何处理更改的文本列表条目：

文本列表包含两个条目：“Entry_1”和“Entry_2”。“Eintry_1”包含由 SiVArc 生成的文本。“Entry_2”包含从文本列表的主副本中复制的文本。

- 更改“Entry_2”并启动 SiVArc 生成操作。生成之后，所做更改将体现在“Entry_2”中。
- 更改“Entry_1”并启动 SiVArc 生成操作。生成之后，将由 SiVArc 生成的文本覆盖对“Entry_1”所做的更改。
- 更改“Entry_1”和“Entry_2”并启动 SiVArc 生成操作。生成之后，将由 SiVArc 生成的文本覆盖对“Entry_1”所做的更改。对“Entry_2”的更改由来自文本列表的主副本中的文本覆盖。

SiVArc 更改机制的优势

通过用于更改生成的显示和操作元素的 SiVArc 功能，组态工程师可以轻松启用有效和一致的调整。

SiVArc 使公司可以在整个公司内分配标准化显示和操作元素，甚至将其应用到正在进行的项目。

参见

示例：使用生成矩阵 (页 227)

SiVArc 项目中的标识 (页 218)

画面图例 (页 287)

6.4.2.2 更改 SiVArc 规则

影响

当更改 SiVArc 规则时，会集中干扰现有项目。

必须编辑 SiVArc 规则，例如，当功能始终在另一个画面上可视化或者由于工厂已更改而要删除操作对象时。

随后编辑 SiVArc 规则

可通过选择规则然后使用快捷菜单命令的方式更改已创建的规则。如果更改项目中的对象名称和存储路径，将相应地更新相关规则。

仅更改项目或项目库中对象的名称和存储路径。SiVArc 不支持更改全局库或引用对象的路径信息。

如果 SiVArc 画面规则和复制规则包含相似的库对象，SiVArc 生成时，复制规则优先级较高，并会生成库对象以及复制规则中定义的文件夹结构。

如果 SiVArc 变量规则和复制规则包含相似的库对象（变量表），SiVArc 生成时，复制规则优先级较高，并会生成库对象以及复制规则中定义的文件夹结构。会生成复制规则中定义的包含 PLC 变量的变量表。如果生成时使用变量规则，会显示警告消息。

如果复制规则不可用，并选择先生成 SiVArc 画面规则，则会生成画面对象。为第二次生成添加复制规则时，将显示一条警告消息。

使用库中的 SiVArc 规则

若要集中、一致地更新多个项目的 SiVArc 规则，请将 SiVArc 规则或规则组以主副本形式存储在库中。如果项目中已存在同名的 SiVArc 规则，可覆盖现有规则或者创建新规则。

使用库中的规则覆盖现有规则时，SiVArc 会将该操作视为手动更改规则进行响应：

- SiVArc 会检查上一个生成过程中相关的 HMI 对象，并且将这些 HMI 对象纳入生成。
- 对相关 HMI 对象所做的手动改动将被覆盖。

更改 SiVArc 规则主副本的名称

按以下步骤，在库中重命名的画面规则与项目中基于此规则的画面规则之间创建链接：

1. 根据主副本在库中的新名称手动更改项目中的画面规则。
2. 现在，将重命名的主副本复制到您的项目中。覆盖项目中现有的新命名的画面规则。

编辑 SiVArc 规则的引用

如果在项目或项目库中编辑引用的 HMI 对象或程序块，则会自动调整 SiVArc 规则。

如果更改全局库中的引用对象，相应的 SiVArc 规则会变为无效。

编辑 SiVArc 规则的优势

由于 SiVArc 规则在项目中包含动态链接，因此可以轻松进行调整，而不会导致项目不一致。可以使用固定规则集，例如，可以单独调整或继续在整个公司内进一步开发。

参见

SiVArc 规则 (页 166)

创建变量规则 (页 191)

创建画面规则 (页 192)

创建文本列表规则 (页 194)

创建复制规则 (页 197)

6.4.2.3 SiVArc 项目中的标识

项目中生成的对象和 SiVArc 组态

SiVArc 项目中可识别以下对象：

- 下一次生成检测到的生成对象
- 包含 SiVArc 组态的对象

下表列出了在 SiVArc 中识别的对象形式：

位置	图标/识别	对象
项目树		相关的对象（HMI 画面）
项目库或全局库		带有组态的 SiVArc 属性、事件或动画的主副本。
		带有组态的 SiVArc 属性、事件或动画的类型。
		带有组态的 SiVArc 属性、事件或动画的类型版本。

在“画面”(Screens) 编辑器的“选项 > 设置 > SiVArc”(Options > Settings > SiVArc) 下，指定生成的画面对象的标识。

生成的 HMI 对象

生成的 HMI 对象将在下一次生成期间再次生成并覆盖。在下次生成之前，将对上一次生成中带有匹配名称的对象进行记录。

如果更改生成对象的名称，则该生成将不再被捕获。

说明

将生成对象复制到其它项目

如果将生成的对象复制到具有或不具有 SiVArc 的其它项目中，则会保留该标识。

“画面”(Screens) 编辑器中的标识

“画面”(Screens) 编辑器中的标识为可选项。可以在 TIA Portal 设置中的“选项 > 设置 > SiVArc”(Options > Settings > SiVArc) 下启用标识并为边框和背景指定所需颜色。

6.4.3 生成可视化

要求

- 用户程序和硬件已编译并且没有错误。
- 已定义画面规则。
- 画面规则中使用的主副本和面板类型存储在项目库或全局库中。
- 已定义变量。
- 文本列表规则已定义。
- 所有使用的类型实例均已更新到最新版本。

说明

对控制器的更改需要编译

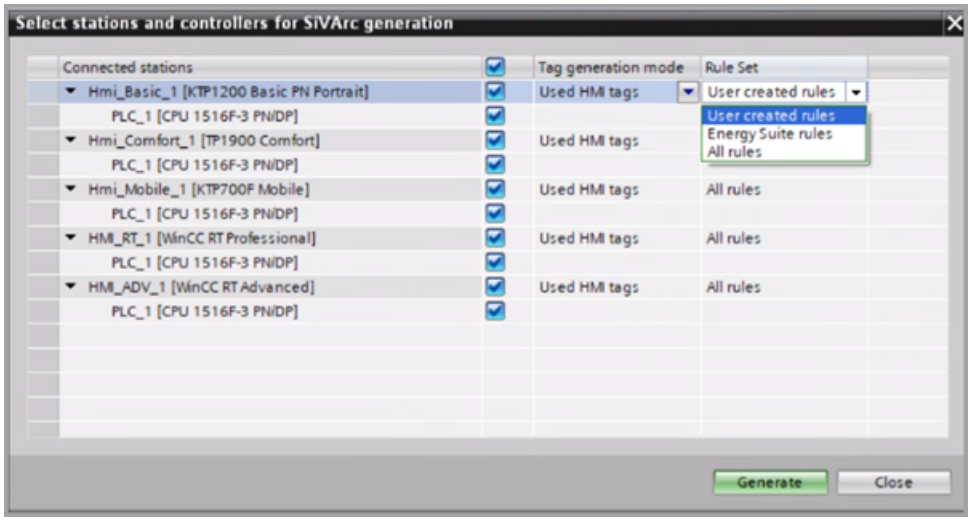
必须先编译用户程序或硬件组态更改，然后才能生成可视化。

不使用站选择进行生成

1. 在运行系统的快捷菜单或项目树的 HMI 设备中单击“可视化 (SiVArc) >”(Visualization (SiVArc) >)。从下列选项中选择：
   - 生成可视化 (SiVArc)
   - 使用站选择进行生成
   - 清除可视化 (SiVArc)

使用站选择进行生成。

1. 确保所有 PLC 的 HMI 设备运行系统设置中的"PLC 前缀"(PLC prefix) 选项均已启用。
2. 在项目树中项目的快捷菜单下单击"生成可视化 (SiVArc) > 使用站选择生成"(Generation of visualization (SiVArc) > Generate with station selection)。
   对话框"选择并生成设备"(Select and generate devices) 已打开。



3. 激活为其生成可视化的 HMI 设备和 PLC。
   要为所有设备均生成可视化，请激活标题中的选项。
4. 单击"生成"(Generate)。

清除生成数据

可以选择通过执行以下操作来清除设备的生成数据：

1. 右键单击"**HMI 设备 > 可视化 (SiVArc) > 清除可视化 (SiVArc)**"(**HMI device > Visualization (SiVArc) > Clear the Visualization (SiVArc)**)。
2. 从站选择弹出窗口中为任何已选 HMI 设备选择规则集：
   - 使用创建的规则 (User created rules) - 系统将仅删除出现的所有使用用户创建的规则生成的 SiVArc 对象
   - Energy Suite 规则 (Energy Suite rules) - 系统将仅删除使用 Energy Suite 规则生成的 SiVArc 对象
   - 所有规则 (All rules) - 系统将删除出现的所有使用上述规则生成的 SiVArc 对象

   说明

   - 将根据用户创建的规则或 Energy Suite 规则生成所选 HMI 设备的 HMI 变量。执行"**清除可视化 (SiVArc)**"(Clear the Visualization (SiVArc)) 任务时，无论生成模式如何，都将删除所有 HMI 变量。
   - 在清除生成数据期间，将验证 SiVArc 许可证和 Energy Suite 许可证。

3. SiVArc 清除对话框会向用户提示一条警告消息，说明"生成概览"和"生成矩阵"中的数据将被删除。

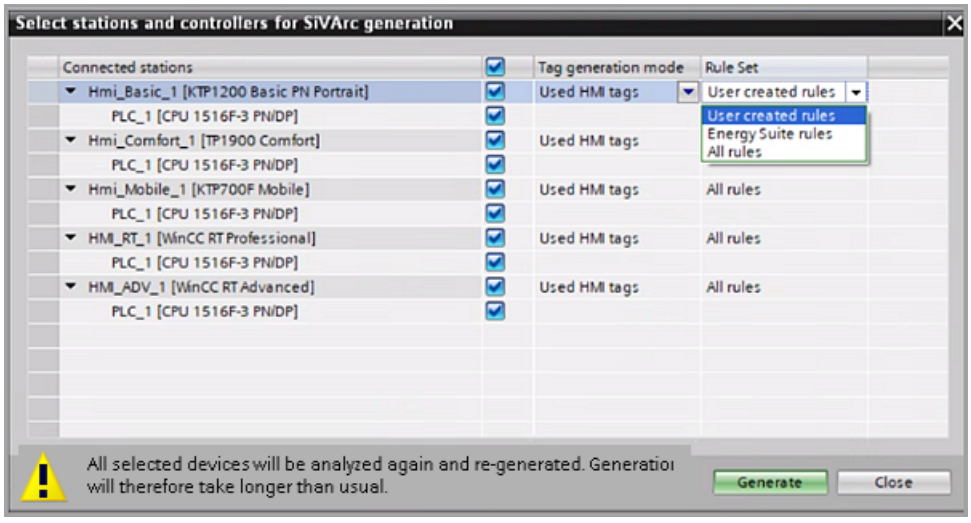
4. 在清除生成数据期间，随时可以选择通过单击清除进度对话框中的“取消”(Cancel) 按钮来取消该过程。
5. 单击“取消”(Cancel) 按钮后，将不会删除任何数据，并且事务将回滚/中止。

重新启动整体生成过程

如果对用户程序进行了更改，可能需要重新启动整体生成。即使在用户程序中没有任何变化，重新启动的整体代码也将在整个用户程序中运行。

选定的连接站保持与第一次生成时相同并且无法修改。

通过选择项目或设备并使用快捷键 <Alt>+<Shift>+<F>，可启动整体生成过程。



设备显示

项目中存在但未连接到控制器的设备不会显示在 SiVArc 编辑器中。

说明

IPI 设备

通过 IPI 与项目相连的控制器和设备不会显示在选择窗口中。

6.5 检查结果

6.5.1 请检查结果。

简介

全面的 SiVArc 项目需要在首次生成后进一步分析和优化。

本文档概述了用于分析和后处理 SiVArc 项目的选项。

检查机制

SiVArc 提供不同的功能和编辑器来检查生成。其中包括以下要点：

- 实际生成哪些对象？
  可以在“公共数据 > SiVArc”(Common data > SiVArc) 下的“生成概览”(Generation overview) 编辑器中找到概览。生成概览是在首次生成期间创建，并在后续生成中更新。
- 哪些对象未生成或生成错误？
  在生成期间，可以在巡视窗口中导航到错误位置的“信息”(Info) 下找到错误。
  此外，生成的日志会显示在项目树中的“公共数据 > 日志”(Common data > Logs) 下。
  编辑器中规则的组织结构在故障排除期间是有帮助的。这样，可以分段生成项目，以便更容易找到任何潜在的错误。

要获得一个将要运行的项目，请编译并下载以测试该项目。测试将确保项目完成并在运行系统中正常运行。

检查生成的完整性

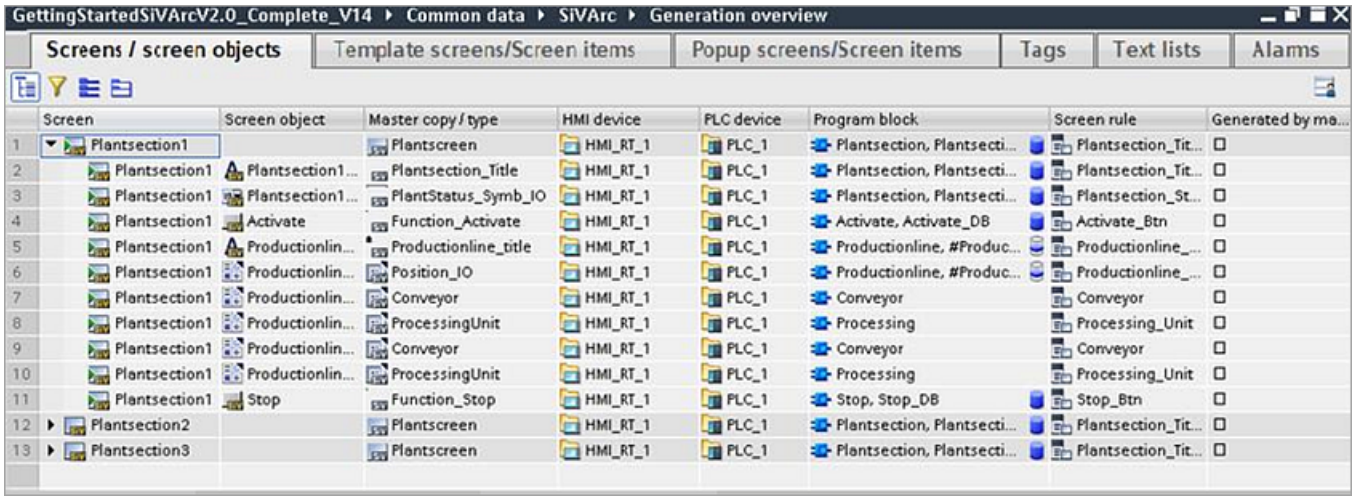
要打开生成概览，请双击项目树中的“公共数据 > SiVArc > 生成概览”(Common data > SiVArc > Generation overview)。还可以通过完成消息打开生成概览，以在巡视窗口中生成可视化。

要在生成概览所列出的项目中识别块、画面规则或生成显示和操作对象，请选择快捷菜单命令“转到引用对象”(Go to referenced object)。

使用编辑器的过滤和排序功能在生成概述中显示不同的视图。画面规则、生成的显示和操作对象以及设备之间的关系可以通过这种方式读取。

借助生成概览，可以规划和组态附加生成的后续更改。

也可以从 SiVArc 项目的多个位置处获取生成概览：



	Screen	Screen object	Master copy / type	HMI device	PLC device	Program block	Screen rule	Generated by ma...
1	Plantsection1		Plantscreen	HMI_RT_1	PLC_1	Plantsection, Plantsecti...	Plantsection_Tit...	
2	Plantsection1	Plantsection1...	Plantsection_Title	HMI_RT_1	PLC_1	Plantsection, Plantsecti...	Plantsection_Tit...	
3	Plantsection1	Plantsection1...	PlantStatus_Symb_IO	HMI_RT_1	PLC_1	Plantsection, Plantsecti...	Plantsection_St...	
4	Plantsection1	Activate	Function_Activate	HMI_RT_1	PLC_1	Activate, Activate_DB	Activate_Btn	
5	Plantsection1	Productionlin...	Productionline_title	HMI_RT_1	PLC_1	Productionline, #Produc...	Productionline_...	
6	Plantsection1	Productionlin...	Position_IO	HMI_RT_1	PLC_1	Productionline, #Produc...	Productionline_...	
7	Plantsection1	Productionlin...	Conveyor	HMI_RT_1	PLC_1	Conveyor	Conveyor	
8	Plantsection1	Productionlin...	ProcessingUnit	HMI_RT_1	PLC_1	Processing	Processing_Unit	
9	Plantsection1	Productionlin...	Conveyor	HMI_RT_1	PLC_1	Conveyor	Conveyor	
10	Plantsection1	Productionlin...	ProcessingUnit	HMI_RT_1	PLC_1	Processing	Processing_Unit	
11	Plantsection1	Stop	Function_Stop	HMI_RT_1	PLC_1	Stop, Stop_DB	Stop_Btn	
12	Plantsection2		Plantscreen	HMI_RT_1	PLC_1	Plantsection, Plantsecti...	Plantsection_Tit...	
13	Plantsection3		Plantscreen	HMI_RT_1	PLC_1	Plantsection, Plantsecti...	Plantsection_Tit...	

- WinCC
  所生成画面的巡视窗口
  所选画面的全部已生成显示和画面对象均显示在“生成概览”(Generation overview) 选项卡中。
- STEP 7
  块的巡视窗口
  “画面生成概览”(Screen generation overview) 和“文本列表生成概览”(Text list generation overview) 显示画面显示了通过所选程序块生成的所有画面、相关画面对象和文本列表。

凭借用于对“生成概览”(Generation overview) 编辑器进行过滤和排序的多个功能，可轻松地从不同方面了解概况。

使用有针对性的单个生成进行故障诊断

也可以逐段检查项目。可以使用规则编辑器生成项目的各个部分。

- 规则也可成组开、关。
- 如果在生成后禁用某个规则，所有相关的生成对象将从生成过程中移除。
- 启用或禁用规则会覆盖规则的条件。例如，当规则的条件为“TRUE”时，只有启用规则时该条件才适用。当规则的条件为“FALSE”时，即使规则已启用也不包含在生成中。
- 为下一次生成再次启用规则时，相关的对象会再次生成。

最终更改

可使用生成矩阵实现最终更改，而无需分析和更改 SiVArc 规则。

参见

使用生成矩阵 (页 224)

示例：使用生成矩阵 (页 227)

编辑和管理 SiVArc 规则 (页 203)

在 SiVArc 编辑器中编辑视图 (页 286)

6.5.2 使用生成矩阵

生成矩阵的应用

生成矩阵可用于对生成对象分配的后续更改；它主要适用于必须对项目进行最终调整的调试工程师。

组态期间只使用画面规则生成画面和画面对象时，可以充分利用生成矩阵。

说明

每次生成操作后，为 HMI 设备或 HMI 设备类型生成的画面和画面对象将显示在“生成矩阵”(Generation matrix) 编辑器中。

此外，可以按如下方式调整分配：

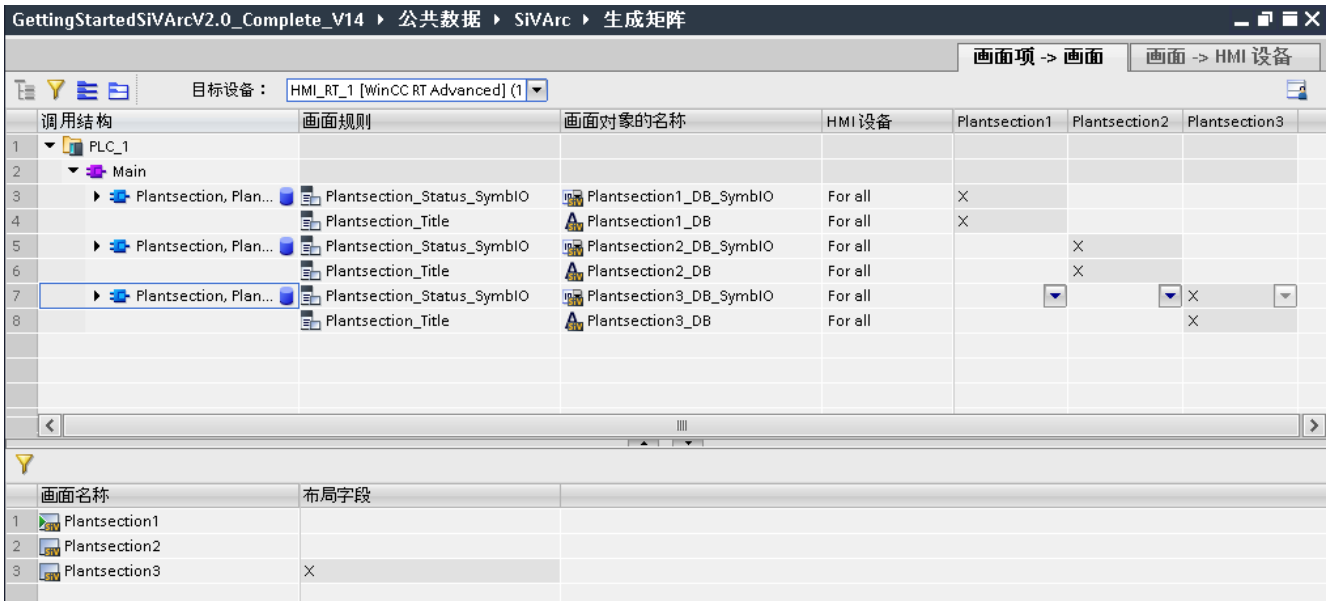
- 在其他画面中生成画面对象
- 在其他 HMI 设备中生成画面

更改的分配在下一次生成操作时生效。根据具体设置，同时还可调整画面导航。

“画面对象 -> 画面”选项卡

在编辑器的工具栏中，可在“目标设备”(Target device) 下选择要为其显示矩阵的 HMI 设备。SiVArc 还会显示所有设备的设备类型。

在此选项卡中，可将生成的画面对象分配给其他画面。



“画面 > HMI 设备”选项卡

在编辑器的工具栏中，可在“设备类型”(Device type) 下选择要为其显示矩阵的 HMI 设备类型。编辑器随即会显示此类型的所有 HMI 设备的画面。

在此选项卡中，可将生成的画面分配给其他 HMI 设备。

调整生成的画面对象和画面的分配

1. 要更改画面对象的分配，请选择布局字段或选择“画面对象 -> 画面”(Screen objects -> Screens) 选项卡的相应单元格中的“X”。
2. 要更改画面的分配，请在相应单元格中选择“画面 -> HMI 设备”(Screen -> HMI devices) 选项卡中的复选框。
3. 生成可视化。

调整画面的导航按钮

使用矩阵最新生成的画面的导航按钮会根据画面层级再次生成。

1. 激活“SiVArc > SiVArc 设置> 矩阵设置 > 生成导航对象”(SiVArc > SiVArc settings > Matrix settings > Generate navigation objects) 选项。
2. 重新分配画面。
3. 生成可视化。

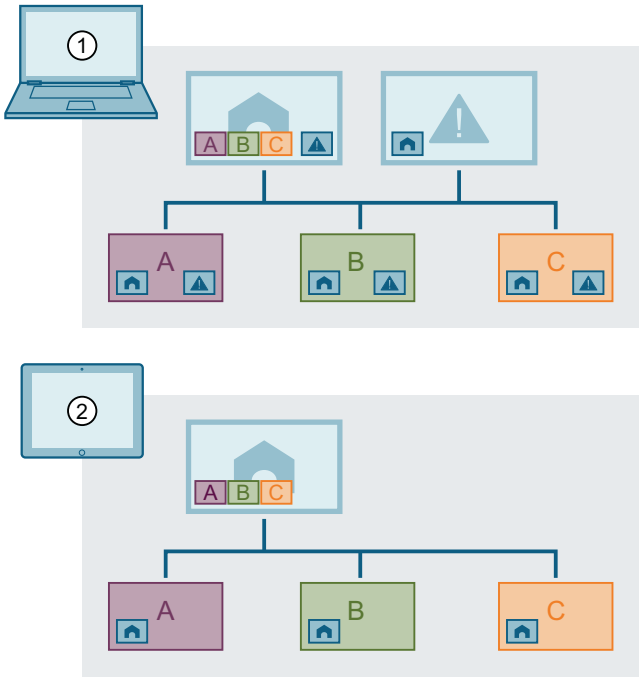
画面以及该画面的导航按钮再次生成。

组态画面的“溢出画面数”属性，然后将其移至库类型。将画面类型用作“主副本/画面类型”，然后将相应地生成导航对象。

示例：使用生成矩阵通过导航方式将画面移动到其它设备中

已在 HMI 设备 1 上生成一个起始画面、一个诊断画面和下层画面。每个下层画面都可以通过导航按钮显示起始画面和诊断画面。

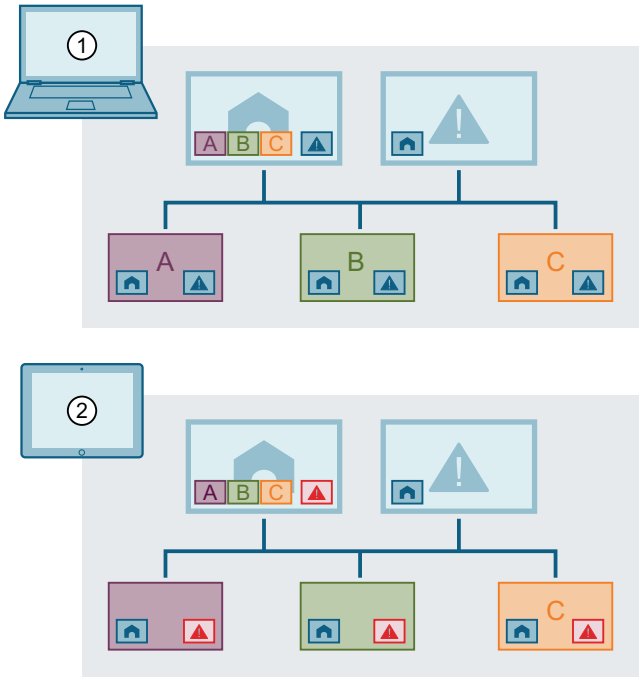
HMI 设备 2 上未生成诊断画面。



① HMI 设备 1

② HMI 设备 2

使用生成矩阵将诊断画面移动到 HMI 设备 2 时，导航按钮会相应调整。



① HMI 设备 1
② HMI 设备 2

参见

示例：使用生成矩阵 (页 227)

在 SiVArc 编辑器中编辑视图 (页 286)

6.5.3 示例：使用生成矩阵

示例场景

当调试工厂时，HMI 设备仍然缺失。供应商没有交付，项目可能会延期。因此，HMI 设备的所有内容都集成到其它 HMI 设备中，直到供应商交付缺失的 HMI 设备。

要求

为了确保调试不延期，调试工程师应该在最后一刻更改 WinCC 项目中的工厂结构。

执行规则

调试工程师使用生成矩阵来移动生成的显示和操作对象以及画面以满足新的要求。

在下一次生成期间对更改进行评估。导航由层级结构自动调整。

原始项目得到保留，并可在缺失的 HMI 设备交付后再次生成。

参见

使用生成矩阵 (页 224)

SiVArc Openness 7

7.1 简介

简介

TIA portal openness 应用程序允许将 SiVArc 实例化。必须使用客户端应用程序访问 TIA portal，并通过 openness 功能启动 SiVArc 服务。有关设置和访问 Openness 的更多详细信息，请参见 TIA portal 用户指南。

设置应用程序

要设置客户端应用程序，请按照以下步骤操作：

1. 创建控制台应用程序。从 _deployed\TIAPV15SP1_11010001\PublicAPI\V15.1\ 936 Siemens.Engineerin.dll 或安装的二进制位置 PublicAPI\V15.1\ 937 Siemens.Engineerin.dll 添加公共 API (Siemens.Engineering.dll) 的参考。
2. 将组态详细信息添加到组态文件中。有关组态详细信息和访问公共 API 的步骤的详细信息，请参见 TIA openness 的维基百科。
3. 要访问 Sivarc 服务，请使用下述 API：

```
using (TiaPortal  tia = new
TiaPortal(TiaPortaMode.WithUserInterface))
{
 Project myProject = tia.Projects.Open(new FileInfo(@"C:\Users
\z003exve\Documents\Automation\Project_Demo\Project_Demo.ap15));
 //if SiVArc is not installed, user will not be able to access
SiVArc service (compiler error)
 Sivarc sivarc =myproject?.GetService<Sivarc>():
 if (sivarc !=null)
  {
  }
}
```

7.2 SiVArc 服务属性

SiVArc 服务属性

下表列出了 SiVArc 所支持的属性和方法：

属性名称	说明	数据类型
AlarmRules	所有报警规则对象的锚对象	AlarmRulesBrowsable
ScreenRules	所有画面规则对象的锚对象	ScreenRulesBrowsable
TextlistRules	所有文本列表对象的锚对象	TextlistRulesBrowsable
TagRules	所有变量规则对象的锚对象	TagRulesBrowsable
CopyRules	所有复制规则对象的锚对象	CopyRulesBrowsable
报警规则	枚举所有即时的一级报警规则	AlarmRuleComposition
组	枚举所有即时的一级报警规则组	AlarmRuleGroupCompositio n
ScreenRules	枚举所有即时的一级画面规则	ScreenRuleComposition
ScreenRulesGr oups	枚举所有即时的一级画面规则组	ScreenRuleGroupCompositi on
TextlistRules	枚举所有即时的一级文本列表规则	TextlistRuleComposition
TextlistGroups	枚举所有即时的一级文本列表规则组	TextlistRuleGroupCompositi on
TagRules	枚举所有即时的一级变量规则	TagRuleComposition
TagRulesGroup s	枚举所有即时的一级变量规则组	TagRuleGroupComposition
CopyRules	枚举所有即时的一级复制规则	CopyRuleComposition
CopyRulesGro ups	枚举所有即时的一级复制规则组	CopyRuleGroupComposition

下表列出了 AlarmRule 和 AlarmRuleGroup 的构成。这也同样适用于其它 SiVArc 对象。

方法名称	参数	说明	数据类型
Find	String – 报警规则/规则组名称	从报警规则/报警组集中查找报警规则/报警规则组	AlarmRule
CreateFrom	MasterCopy – 报警规则/报警规则组主副本	使用默认替换选项将报警规则/报警规则组主副本从库中复制到项目	AlarmRule
CreateFrom	MasterCopy – 报警规则/报警规则组主副本，CreateOptions – 重命名/替换	使用创建选项将报警规则/报警规则组主副本从库中复制到项目	AlarmRule

7.3 从库中复制规则或组

要求

- 启动 TIA portal openness 应用程序。有关连接的更多信息，请参见 TIA portal 用户指南。
- 已存在包含画面规则编辑器、画面规则组和主副本的 TIA portal 项目。

案例 1：将规则/规则组从主副本复制到画面规则编辑器时

“`CreateFrom`”全局库允许将规则和规则组从全局库复制到 SiVarc 规则编辑器。如果操作成功，API 函数将返回 ScreenRule/ScreenRuleGroup。以下代码说明了如何将规则或规则组从主副本复制到 SiVArc 编辑器：

```
// Finds screen rule master copy "Screen rule_1"
MasterCopy screenRuleMasterCopy =
           myProject.ProjectLibrary.MasterCopyFolder.MasterCopies.Find("Screen rule_1");

    if (screenRuleMasterCopy != null)
    {
        var rule = sivarc.ScreenRules.Rules.CreateFrom(screenRuleMasterCopy);
        if (rule != null)
        {
            Console.WriteLine("Copied Screen Rule Name: " + rule.Name);
            Console.WriteLine("Copied Screen Rule Comment: " + rule.Comment);
            Console.WriteLine("Copied Screen Rule Enabled: " + rule.Enabled);
            Console.WriteLine("Copied Screen Rule Condition: " + rule.Condition);
        }
    }
```

默认情况下，该行为将被替换。

案例 2：从主副本复制规则/规则组时，将基于第二个参数输入重命名或替换现有的规则/规则组

如果主副本中的规则/规则组已存在于 SiVArc 编辑器中，则尝试复制时将会重命名这些规则/规则组。如果 SiVArc 编辑器中不存在这些规则/规则组，“CreateOptions”API 将会创建它们，否则，它们将替代现有的规则/规则组。如果操作成功，API 函数将重命名 ScreenRule/ScreenRuleGroup。替代规则/规则组的过程如下列代码片段所示：

```
// Finds screen rule master copy "Screen rule_1"
MasterCopy screenRuleMasterCopy =
           myProject.ProjectLibrary.MasterCopyFolder.MasterCopies.Find("Screen rule_1");

    if (screenRuleMasterCopy != null)
    {
        var rule = sivarc.ScreenRules.Rules.CreateFrom(screenRuleMasterCopy, CreateOptions.Rename);
        if (rule != null)
        {
            Console.WriteLine("Copied Screen Rule Name: " + rule.Name);
            Console.WriteLine("Copied Screen Rule Comment: " + rule.Comment);
            Console.WriteLine("Copied Screen Rule Enabled: " + rule.Enabled);
            Console.WriteLine("Copied Screen Rule Condition: " + rule.Condition);
        }
    }
```

7.4 查找画面规则和画面规则组

要求

- TIA portal openness 应用程序已连接到 TIA portal。有关连接的更多信息，请参见 TIA portal 用户指南。
- TIA portal 项目包含画面规则和画面规则组。

在画面组内查找画面规则

案例 1：在画面规则编辑器内查找画面规则。

可通过 API `sivarc.ScreenRules.Rules` 在 SiVArc 画面规则编辑器内查找可用的规则，如下所示：

```
// Collection of all immediate first level screen rules
ScreenRuleComposition screenRules = sivarc.ScreenRules.Rules;
if(screenRules != null && screenRules.Count > 0)
{
    // Finds screen rule
    ScreenRule rule = screenRules.Find("Screen rule_7");
    if(rule != null)
    {
        Console.WriteLine("Screen Rule Name: " + rule.Name);
        Console.WriteLine("Screen Rule Comment: " + rule.Comment);
        Console.WriteLine("Screen Rule Enabled: " + rule.Enabled);
        Console.WriteLine("Screen Rule Condition: " + rule.Condition);
```

案例 2：在画面规则编辑器内查找画面规则组。

可通过 API `sivarc.ScreenRules.Rules` 在 SiVArc 画面规则编辑器内查找可用的规则，如下所示：

```
var groups = sivarc.ScreenRules.Groups;
if (groups != null && groups.Count > 0)
{
    var rule = groups.Find("Screen rule group").Rules.Find("Screen rule_2");
    if (rule != null)
    {
        Console.WriteLine("Found Screen Rule Name: " + rule.Name);
        Console.WriteLine("Found Screen Rule Comment: " + rule.Comment);
        Console.WriteLine("Found Screen Rule Enabled: " + rule.Enabled);
        Console.WriteLine("Found Screen Rule Condition: " + rule.Condition);
    }

    var group = groups.Find("Screen rule group").Groups.Find("Screen rule group_2");
    if (group != null)
    {
        Console.WriteLine("Found Screen Rule Group Name: " + group.Name);
        Console.WriteLine("Found Screen Rule Group Comment: " + group.Comment);
        Console.WriteLine("Found Screen Rule Group Enabled: " + group.Enabled);
        Console.WriteLine("Found Screen Rule Group Condition: " + group.Condition);
    }
}
```

7.5 删除规则和规则组

要求

- TIA portal openness 应用程序已连接到 TIA portal。有关连接的更多信息，请参见 TIA portal 用户指南。
- TIA 项目包含画面规则和画面规则组。

删除规则和规则组

要删除规则或规则组，请使用以下 API：

```
sivarc.ScreenRules.Rules.Find("Screen rule_1").Delete();
```

`ScreenRules` 是所有画面规则对象的锚对象。要执行删除，必须在规则编辑器中找到该规则。有关查找画面规则的更多信息，请参见“查找画面规则和画面规则组”部分。

要从画面组中删除画面，请使用以下 API：

```
sivarc.ScreenRules.Rules.Find("Screen rule group_1").Delete();
```

7.6 Openness 的 UMAC 设置

关于 UMAC

要通过 openness 访问 UMAC，请确保具有 UMAC 凭证和访问特权。如果不是有效的 UMAC 用户，应用程序会对所有 SiVArc 锚规则对象返回 NULL 值。有关 UMAC 的更多信息，请参见 SiVArc 用户指南中的 UMAC 主题。

7.7 SiVArc 生成

要求

- 启动 TIA portal openness 应用程序。有关连接的更多信息，请参见 TIA portal 用户指南。
- 已存在连接到 HMI 设备的 TIA portal 项目，且已组态 PLC。

重要注意事项

- 确保 PC 上已安装 SiVArc 许可证，否则生成期间会出现例外情形 -*"SiVArc 许可证缺失，修改数据必须使用 SiVArc 许可证"*。
- 确保使用有效的设备名称，否则会出现例外情形 -*"未找到 HMI 设备的deviceName'"*。
- 确保调用有效的 PLC 名称，否则会出现例外情形 -*"未找到PLC 设备的plcDeviceName'"*。
- 确保调用受支持的设备名称，否则会出现例外情形 -*"不支持 HMI 设备的deviceName'"*
- 确保调用受支持的 PLC 名称，否则会出现例外情形 -*"不支持 PLC 设备的 'plcDeviceName'"*
- 确保传递有效的 GenerationOption 参数。如果未传递参数，将生成 SiVArc 并为 SiVArc 生成使用默认的 TIAP 项目设置
- 确保使用未用于先前生成的有效 PLC 名称，否则系统将冻结。

GenerationOptions- Enum (Flag)

SiVArc 支持 Enum 选项，可以在 `Generate` API 中将两个值结合起来传递。Enum 选项如下表所示：

SN	值	说明
1	None	未选中任何操作，将采用默认生成设置
2	AllTags	生成所有变量
3	UsedHmiTags	仅生成相关（已用）变量
4	FullGeneration	如果未选中 FullGeneration 选项，SiVArc 将基于组态自行决定执行全部生成或增量生成。 传递的参数为 `FullGeneration` 时，将使用强制全部生成方式生成 SiVArc。

要生成 SiVArc，请使用以下 API：

```
sivarc.Generate("HMI_1", new List<string>  {PLC_1},
GenerateOptions.AllTags | GenerateOptions.FullGeneration);
```

SiVArcGenerationResult and SivarcFeedbackMessage

生成成功后，SiVArc 生成将使用以下属性：

- IsGenerationSuccessful - 通知是否成功生成 SiVArc。
- WarningCount - 生成 SiVArc 后的警告总数
- ErrorCount - 生成 SiVArc 后的错误总数
- 消息 - 形成反馈消息

要生成 SiVArc 结果，请使用以下 API：

- Print Sivarc generated feedback messages

```
private void WriteSivarcGenerationResults(SivarcGenerationResult result)
     {
          sb.Append("Is SiVArc generation successful:" +
          result.IsGenerationSuccessful);
        sb.Append(Environment.NewLine);
        sb.Append("Total Warning Count:" + result.WarningCount);
        sb.Append(Environment.NewLine);
        sb.Append("Total Error Count:" + result.ErrorCount);
        sb.Append(Environment.NewLine);

        RecursivelyWriteMessages(result.Messages);
     }
```

生成成功后，SiVArc 生成将使用以下反馈消息：

- 路径：反馈消息的标头文本（标头消息始终具有空白描述字段）
- DateTime：反馈消息的日期时间
- MessageType：反馈消息类型
- 说明：反馈消息描述/内容（仅限于路径为空的情况，确保不是标头消息）
- WarningCount：标头消息的警告数
- ErrorCount：标头消息的错误数
- 消息：形成反馈消息 (SivarcFeedbackMessage

可以使用以下代码片段查看递归的反馈消息：

```
private void RecursivelyWriteMessages(SivarcFeedbackMessageComposition messages)
  {
     foreach (SivarcFeedbackMessage message in messages)
      {
          sb.Append("Path: " + message.Path);
          sb.Append(Environment.NewLine);
          sb.Append("DateTime: " + message.DateTime);
          sb.Append(Environment.NewLine);
          sb.Append("State: " + message.MessageType);
          sb.Append(Environment.NewLine);
          sb.Append("Description: " + message.Description);
          sb.Append(Environment.NewLine);
          sb.Append("Warning Count: " + message.WarningCount);
          sb.Append(Environment.NewLine);
          sb.Append("Error Count: " + message.ErrorCount);
          sb.Append(Environment.NewLine);
          sb.Append(Environment.NewLine);

          RecursivelyWriteMessages(message.Messages, indent);
      }
  }
```

参考

8

8.1 SiVArc 对象

8.1.1 PLC 变量

说明

PLC 变量表示 PLC 支持的变量。

使用

可以使用 PLC Tag 对象将生成的外部 PLC 支持的变量以结构化形式存储到项目树中。

按如下步骤使用“PLC Tag”对象：

- “SymbolicName”对象属性
  `PLCTag.DB.SymbolicNam`
  访问数据块的用户自定义名称。

8.1.2 I/O 设备

说明

智能设备作为 IO 设备连接到“较高级别”的 IO 控制器。

使用

可以使用 IODevice 对象访问项目中的 IO 设备。

按如下步骤使用“Device.name”对象：

- “Name”对象属性
  IODevice.Name
  访问 IO 设备的用户自定义名称，例如 HMI_1。

8.1.3 软件单元

说明

可组态 HMI 对象各种属性的表达式，其表现为 SiVArc 规则创建中所包含的 PLC 块的软件单元名称。

使用

可以使用 softwareunit 对象访问项目中的软件单元设备。

按如下步骤使用“Device.name”对象：

- “Name”对象属性
  SoftwareUnit.Name
  访问软件单元的用户自定义名称。

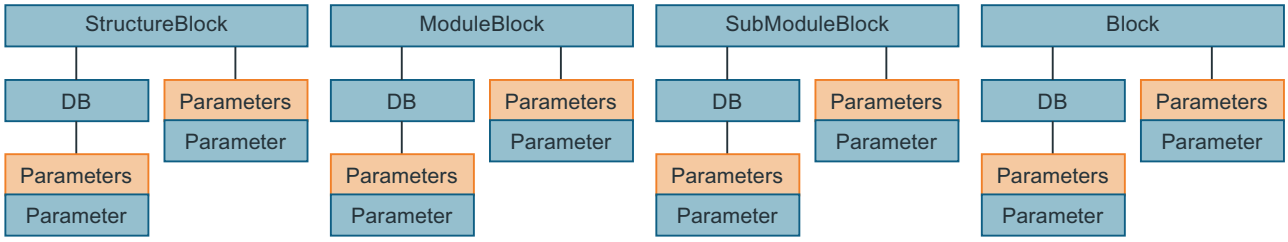
8.1.4 对象层级

简介

可以使用 SiVArc 表达式直接从 TIA Portal 的不同区域中寻址数据。

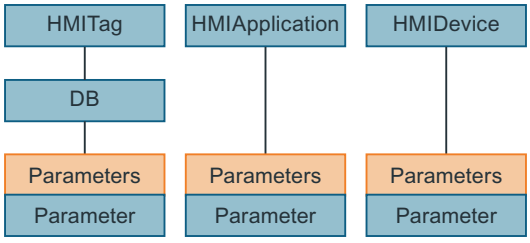
STEP 7 中的程序调用

可以使用关键字访问用户程序中的块、相关数据块及其参数。



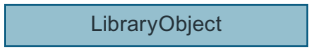
WinCC 数据

可以使用以下关键字访问 HMI 可视化的外部变量、设备和应用程序。



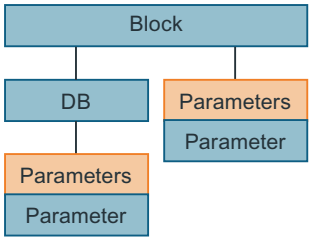
库数据

可使用关键字 `LibraryObject` 访问库中生成模板的存储位置。



8.1.5 Block (Panels, Comfort Panels, RT Advanced, RT Professional)

描述



表示 SiVArc 当前正在执行的程序块，无论其位置在调用层级的何处。

使用

按如下所述使用“Block”对象：

- “FolderPath”对象属性
  `Block.FolderPath`
  访问块在项目树的“程序块”文件夹中的路径，例如，“Plant\Plantsection \Productionline”
- “Name”对象属性
  `Block.Name`
  访问块的内部名称，例如，“FB1”。
- “SymbolicName”对象属性
  `Block.SymbolicName`
  访问用户自定义的块名称。
- “NetworkComment”对象属性
  `Block.NetworkComment`
  访问在块的程序段中输入的注释。
- “NetworkTitle”对象属性
  `Block.NetworkTitle`
  访问用于对块进行实例化的程序段的标题。
- “Number”对象属性
  `ModuleBlock.DB.Number`
  访问块属性中的块编号。
- “Parameters”列表
  `ModuleBlock.Parameters("Activate").Value`
  访问块参数。
- “SymbolComment”对象属性
  `Block.SymbolComment`
  访问块属性中用户自定义的注释。
- “Title”对象属性
  `Block.Title`
  访问块属性中的块标题。

- “Version”对象属性
  `Block.Version`
  如果块是一个块类型的实例，则通过该表达式可访问库中块类型的类型版本。
- “Parameters”列表
  `Block.Parameters(<Name Parameter>).AssignedTag.Comment`
  访问已分配给块参数的变量的注释。

8.1.6 DB (Panels, Comfort Panels, RT Advanced, RT Professional)

描述

表示块的数据块。DB 对象为第二层级的 SiVArc 对象。调用层级的块或 `HMITag` 对象始终在 DB 对象之前。

使用

按如下所述使用 "DB" 对象：

- “Comment”对象属性
  `ModuleBlock.DB.Comment`
  访问块属性中的注释。
- “FolderPath”对象属性
  `HMITag.DB.FolderPath`
  访问块在项目树的“程序块”文件夹中的路径，例如，“DBs\Plant”
- “Number”对象属性
  `SubModuleBlock.DB.Number`
  访问块属性中的块编号。
- “SymbolicAddress”对象属性
  `StructureBlock.DB.SymbolicAddress`
  访问数据块的用户自定义名称。
  如果数据块为多背景数据块，则会返回块的符号地址。

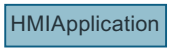
- “TagPrefix”对象属性
  `StructureBlock.DB.TagPrefix`
  访问数据块的用户自定义名称。
  如果数据块为多背景数据块，则会返回 HMI 格式的符号地址。使用“_”（而不是“.”）作为数据块名称和变量名称间的定界符。
- “SymbolicName”对象属性
  `HMITag.DB.SymbolicName`
  访问数据块的用户自定义名称。
- “Type”对象属性
  `ModuleBlock.DB.Type`
  访问数据块的类型：单背景 (IDB) 或多背景 (MDB)。

参见

多语言对生成模板的影响 (页 128)

8.1.7 HMIApplication (Panels, Comfort Panels, RT Advanced, RT Professional)

描述



表示 HMI 设备上的 Runtime 软件。

使用

可以使用 HMIApplication 对象访问 HMI 设备的 Runtime 应用程序。

按如下所述使用“HMIApplication”对象：

- “Name”对象属性
  `HMIApplication.Name`
  访问 HMI 设备的运行系统软件的用户自定义名称，例如，RT_HMI_1。
- “Type”对象属性
  `HMIApplication.Type`
  访问运行系统软件的类型，例如，WinCC RT Advanced。

说明

如果 HMI 设备是一个面板，则 HMIDevice 和 HMIApplication 对象相同。

8.1.8 HMIDevice (Panels, Comfort Panels, RT Advanced, RT Professional)

描述

表示项目中的 HMI 设备。

使用

可以使用 HMIDevice 对象访问项目中的 HMI 设备。

按如下所述使用“HMIDevice”对象：

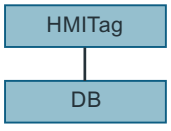
- “Name”对象属性
  `HMIDevice.Name`
  访问 HMI 设备的用户自定义名称，例如，HMI_1。
- “Type”对象属性
  `HMIDevice.Type`
  访问 HMI 设备的类型，例如 KTP400。

说明

如果 HMI 设备是一个面板，则 HMIDevice 和 HMIApplication 对象相同。

8.1.9 HMITag (Panels, Comfort Panels, RT Advanced, RT Professional)

描述



表示外部变量。

使用

可以使用 HMITag 对象将生成的外部变量以结构化形式存储到项目树中。

说明

可能的应用

仅在“变量规则”(Tag rules) 编辑器中使用 HMITag 对象。

按如下所述使用“HMITag”对象：

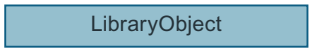
- “FolderPath”对象属性
  `HMITag.DB.FolderPath`
  访问块在项目树的“程序块”文件夹中的路径，例如，“Plant\Plantsection \Productionline”
- “SymbolicName”对象属性
  `HMITag.DB.SymbolicName`
  访问数据块的用户自定义名称。

参见

生成对象的存储策略 (页 131)

8.1.10 LibraryObject (Panels, Comfort Panels, RT Advanced, RT Professional)

描述



表示项目库中的画面类型。

使用

仅在画面生成模板的 SiVArc 属性“名称”(Name) 和“画面组”(Screen group) 中使用 LibraryObject 对象。

- “FolderPath”对象属性
  `LibraryObject.FolderPath`
  引用库中的画面类型路径。如果在 SiVArc 属性“画面组”(Screen group) 中使用 SiVArc 表达式，则通过项目树中的库创建存储路径。如果在“Name”属性中使用 SiVArc 表达式，则在存储画面类型的文件夹后面即为生成画面的名称。

  说明

  对于库中的一个层级，只能在“Name”下使用该表达式。如果想要使用多层存储层级，可以用表达式 `LibraryObject.FolderPath` 代替反斜线。

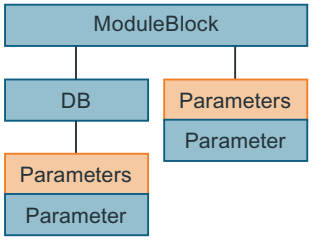
- “Name”对象属性
  `LibraryObject.Name`
  引用库的画面类型名称。如果在 SiVArc 表达式中使用 SiVArc 属性“画面组”(Screen group)，则画面存储在项目树中带有画面类型名称的文件夹中。如果在 SiVArc 属性“Name”中使用 SiVArc 表达式，则在画面类型后面即为画面的名称。

参见

生成对象的存储策略 (页 131)

8.1.11 ModuleBlock (Panels, Comfort Panels, RT Advanced, RT Professional)

描述



表示调用层级的第二级程序块。可使用 ModuleBlock 对象对第二级块进行绝对寻址。

使用

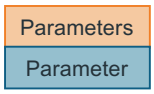
可使用 ModuleBlock 对象访问块和相关数据块的各种属性。

按如下所述使用“ModuleBlock”对象：

- “FolderPath”对象属性
  `ModuleBlock.FolderPath`
  访问块在项目树的“程序块”文件夹中的路径，例如，“Plant\Plantsection \Productionline”
- “Name”对象属性
  `ModuleBlock.Name`
  访问块的内部名称，例如，“FB1”。
- “NetworkComment”对象属性
  `ModuleBlock.NetworkComment`
  访问在块的程序段中输入的注释。
- “NetworkTitle”对象属性
  `ModuleBlock.NetworkTitle`
  访问用于对块进行实例化的程序段的标题。
- “Number”对象属性
  `ModuleBlock.DB.Number`
  访问块属性中的块编号。
- “Parameters”列表
  `ModuleBlock.Parameters("Activate").Value`
  访问块参数。
- “SymbolComment”对象属性
  `ModuleBlock.SymbolComment`
  访问块属性中用户自定义的注释。
- “SymbolicName”对象属性
  `ModuleBlock.SymbolicName`
  访问用户自定义的块名称。
- “Title”对象属性
  `ModuleBlock.Title`
  访问块属性中的块标题。
- “Version”对象属性
  `ModuleBlock.Version`
  如果块是一个块类型的实例，则通过该表达式可访问库中块类型的类型版本。

8.1.12 Parameters (Panels, Comfort Panels, RT Advanced, RT Professional)

描述



Parameters 对象是块中所有参数的列表。Parameter-Objekt 表示指定数据块或块中的参数。

使用

可以使用 Parameters 对象访问块中的特定参数值。

按如下所述使用"Parameters"对象：

- "Assigned"对象属性
  `StructureBlock.Parameters("<Name Parameter>").Value`
  如果已分配参数，则返回 TRUE。
- "Comment"对象属性
  `Parameters("<Name Parameter>").Comment`
  访问参数的注释。
- "InitialValue"对象属性
  `Parameters("<Name Parameter>").InitialValue`
  访问参数的默认值。
- "Value"对象属性
  `Parameters("<Name Parameter>").Value`
  访问参数值。

8.1.13 S7Control (Panels, Comfort Panels, RT Advanced, RT Professional)

描述

表示项目中的 PLC。

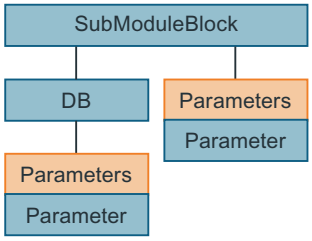
使用

使用 S7Control 对象访问 PLC 的名称：

- “Name”对象属性
  `S7Control.Name`

8.1.14 SubModuleBlock (Panels, Comfort Panels, RT Advanced, RT Professional)

描述



表示调用层级的第三级程序块。可使用 SubModuleBlock 对象对第三级块进行绝对寻址。

使用

可使用 SubModuleBlock 对象访问块及其数据块的各种属性。

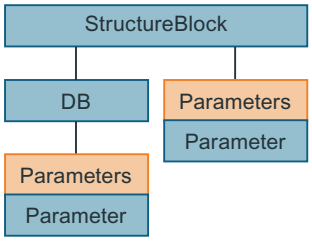
按如下所述使用“SubModuleBlock”对象：

- “FolderPath”对象属性
  `SubModuleBlock.FolderPath`
  访问块在项目树的“程序块”文件夹中的路径，例如，“Plant\Plantsection \Productionline”
- “Name”对象属性
  `SubModuleBlock.Name`
  访问块的内部名称，例如，“FB1”。
- “NetworkComment”对象属性
  `SubModuleBlock.NetworkComment`
  访问在块的程序段中输入的注释。
- “NetworkTitle”对象属性
  `SubModuleBlock.NetworkTitle`
  访问用于对块进行实例化的程序段的标题。

- “Number”对象属性
  `SubModuleBlock.DB.Number`
  访问块属性中的块编号。
- “Parameters”列表
  `SubModuleBlock.Parameters("Activate").Value`
  访问块参数。
- “SymbolComment”对象属性
  `SubModuleBlock.SymbolComment`
  访问块属性中用户自定义的注释。
- “SymbolicName”对象属性
  `SubModuleBlock.SymbolicName`
  访问用户自定义的块名称。
- “Title”对象属性
  `SubModuleBlock.Title`
  访问块属性中的块标题。
- “Version”对象属性
  `SubModuleBlock.Version`
  如果块是一个块类型的实例，则通过该表达式可访问库中块类型的类型版本。

8.1.15 StructureBlock (Panels, Comfort Panels, RT Advanced, RT Professional)

描述



表示调用层级的第一级程序块。可使用 StructureBlock 对象对第一级块进行绝对寻址。

使用

可使用 StructureBlock 对象访问块及其数据块的各种属性。

按如下所述使用"StructureBlock"对象：

- "FolderPath"对象属性
  `SubModuleBlock.FolderPath`
  访问块在项目树的"程序块"文件夹中的路径，例如，"Plant\Plantsection \Productionline"
- "Name"对象属性
  `SubModuleBlock.Name`
  访问块的内部名称，例如，"FB1"。
- "NetworkComment"对象属性
  `SubModuleBlock.NetworkComment`
  访问在块的程序段中输入的注释。
- "NetworkTitle"对象属性
  `SubModuleBlock.NetworkTitle`
  访问用于对块进行实例化的程序段的标题。
- "Number"对象属性
  `SubModuleBlock.DB.Number`
  访问块属性中的块编号。
- "Parameters"列表
  `SubModuleBlock.Parameters("Activate").Value`
  访问块参数。
- "SymbolComment"对象属性
  `SubModuleBlock.SymbolComment`
  访问块属性中用户自定义的注释。
- "SymbolicName"对象属性
  `SubModuleBlock.SymbolicName`
  访问用户自定义的块名称。
- "Title"对象属性
  `SubModuleBlock.Title`
  访问块属性中的块标题。
- "Version"对象属性
  `SubModuleBlock.Version`
  如果块是一个块类型的实例，则通过该表达式可访问库中块类型的类型版本。

8.1.16 TagNaming (Panels, Comfort Panels, RT Advanced, RT Professional)

描述

表示变量的运行系统设置。

使用

可使用 TagNaming 对象访问在运行系统设置中为 PLC 变量路径的较低层级选择的替换定界符。

按如下所述使用“TagNaming”对象：

- “SeparatorChar”对象属性
  `TagNaming.SeparatorChar`
- “IndexStartChar”对象属性
  `TagNaming.IndexStartChar`
- “IndexEndChar”对象属性
  `TagNaming.IndexEndChar`

返回值

“PLC1”控制器包含结构化数据块“DB1”。“Db1.a[1].b.c[3]”数据块元素已在画面中使用。根据设置的不同，TagNaming 对象有以下几种返回值：

返回值	WinCC 变量名称	所选运行系统设置
`TagNaming.SeparatorChar` = "." `TagNaming.IndexStartChar` = "[" `TagNaming.IndexEndChar` = "]"	Db1_a[1].b.c[3]	兼容模式
	Plc1.Db1.a[1].b.c[3]	PLC 前缀
	Db1.a[1].b.c[3]	不使用字符选择替换定界符 将在画面中的使用位置用引号将变量名称括起来："Db1.a[1].b.c[3]"
`TagNaming.SeparatorChar` = ; `TagNaming.IndexStartChar` = "(" `TagNaming.IndexEndChar` = ")"	Db1;a(1);b;c(3)	用 ; ( ) 替换句号和括号
`TagNaming.SeparatorChar` = "_" `TagNaming.IndexStartChar` = "{" `TagNaming.IndexEndChar` = "}"	Plc1_Db1_a{10}_b_c{3}	用 _ { } 替换句号和括号 PLC 前缀

8.2 SiVArc 对象属性

8.2.1 已分配

描述

如果指定的块参数存在分配，则返回 TRUE。

语法

<Object**>**.Assigned

Object

- Parameter

8.2.2 注释

描述

返回输入的注释。

语法

<Object**>**.Comment

Object

- Parameter
- DB

注释

如果查询数据块的注释，则会返回块属性的注释。

如果查询参数的注释，则会返回符号表的注释。

多种语言

SiVArc 表达式"DB.Comment"可组态为多种语言。

参见

多语言对生成模板的影响 (页 128)

8.2.3 FolderPath

描述

返回路径。

语法

<Object**>**.FolderPath

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block
- DB
- LibraryObject

注释

如果查询参数块的存储路径，则会返回“程序块”文件夹中的存储路径。

如果查询库对象的存储路径，则会返回“主副本”或“类型”文件夹中的存储路径。

将返回"\"作为文件夹层级之间的分隔符。

参见

生成对象的存储策略 (页 131)

8.2.4 HMITagPrefix

描述

返回画面窗口的“TagPrefix”属性的值。

例如，“TagPrefix”属性为 SiVArc 当前正在评估的程序块的相关数据块名称。

语法

<Object>.HMITagPrefix

Object

- DB

8.2.5 IndexEndChar

描述

返回在对外部变量进行结构化时在运行系统设置中设置的闭括号。

语法

<Object>.IndexEndChar

Object

- TagNaming

8.2.6 IndexStartChar

描述

返回在对外部变量进行结构化时在运行系统设置中设置的开括号。

语法

<Object>.IndexStartChar

Object

- TagNaming

8.2.7 InitialValue

描述

返回参数的默认值。

语法

<Object**>**.InitialValue

Object

- Parameter

8.2.8 名称

描述

返回内部名称，例如，“FB1”

语法

<Object**>**.Name

Object

- S7Control
- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block
- HMIApplication
- HMIDevice

参见

生成对象的存储策略 (页 131)

8.2.9 NetworkComment

描述

返回程序段注释。

语法

<Object**>**.NetworkComment

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block

多种语言

“NetworkComment”对象属性可组态为多种语言。

参见

多语言对生成模板的影响 (页 128)

8.2.10 NetworkTitle

描述

返回程序段标题。

语法

<Object**>**.NetworkTitle

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block

多种语言

"NetworkTitle"对象属性可组态为多种语言。

参见

多语言对生成模板的影响 (页 128)

8.2.11 编号

描述

返回块编号。

语法

<Object**>.**Number

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block
- DB

8.2.12 SeparatorChar

描述

返回运行系统设置中指定的分隔符。

该分隔符置于 PLC 变量路径的各较低层级（包含在同步的外部变量名称中）之间。

语法

<Object**>**.SeparatorChar

Object

- TagNaming

8.2.13 SymbolComment

描述

返回块属性中用户自定义的注释。

语法

<Object**>**.SymbolComment

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block
- DB

多种语言

“SymbolComment”对象属性可组态为多种语言。

参见

多语言对生成模板的影响 (页 128)

8.2.14 SymbolicName

描述

返回用户自定义的块或变量名称。

语法

```
<Object>.SymbolicName
```

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block
- DB
- HMITag

注释

如果查询多背景 (MDB) 数据块的用户自定义名称，则会调用存储在块接口中的块名称。MDB 的块名称存储在本地静态数据下。

8.2.15 标题

描述

返回块标题。

语法

```
<Object>.Title
```

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block

多种语言

"Title"对象属性可组态为多种语言。

参见

多语言对生成模板的影响 (页 128)

8.2.16 类型

描述

返回类型。

语法

```
<Object>.Type
```

Object

- DB
- HMIApplication
- HMIDevice

注释

如果查询数据块的类型，则会以字符串的形式返回类型"MDB"（多背景块）或"IDB"（实例块）。

如果查询 HMI 设备的类型，则会以字符串的形式返回设备类型，例如"KTP400"。

如果查询运行系统软件的类型，则会以字符串的形式返回软件类型，例如"WinCC RT Advanced"。

8.2.17 值

描述

返回值。

语法

<Object**>**.Value

Object

- Parameter

8.2.18 版本

描述

返回块类型的版本。

语法

<Object**>**.Version

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block

注释

只有在 SiVArc 当前评估的块是库中某块类型的实例时，才评估属性。

8.3 SiVArc 对象属性

访问 HMI 设备

使用以下变量，可以访问项目树中的 HMI 设备。

SiVArc 对象属性	寻址的 属性
HmiDevice.Name	项目树中 HMI 设备的名称 例如“HMI_1”、“PC_system_1”
HmiDevice.Type	项目中 HMI 设备的类型 例如“KTP700 Mobile”、“SIMATIC PC 站”
HmiApplication.Name	应用程序的名称 例如“HMI_1”、“HMI_RT_40
HmiApplication.Type	应用程序的类型 例如“WinCC RT Advanced”、“WinCC RT Professional”
LayoutFieldIndex	在生成期间所使用的布局字段的索引值 在 SiVArc 表达式中待使用

如果 HMI 设备为面板，那么 HmiDevice 和 HmiApplication 相同。

用于控制器名称和外部变量的 SiVArc 对象属性

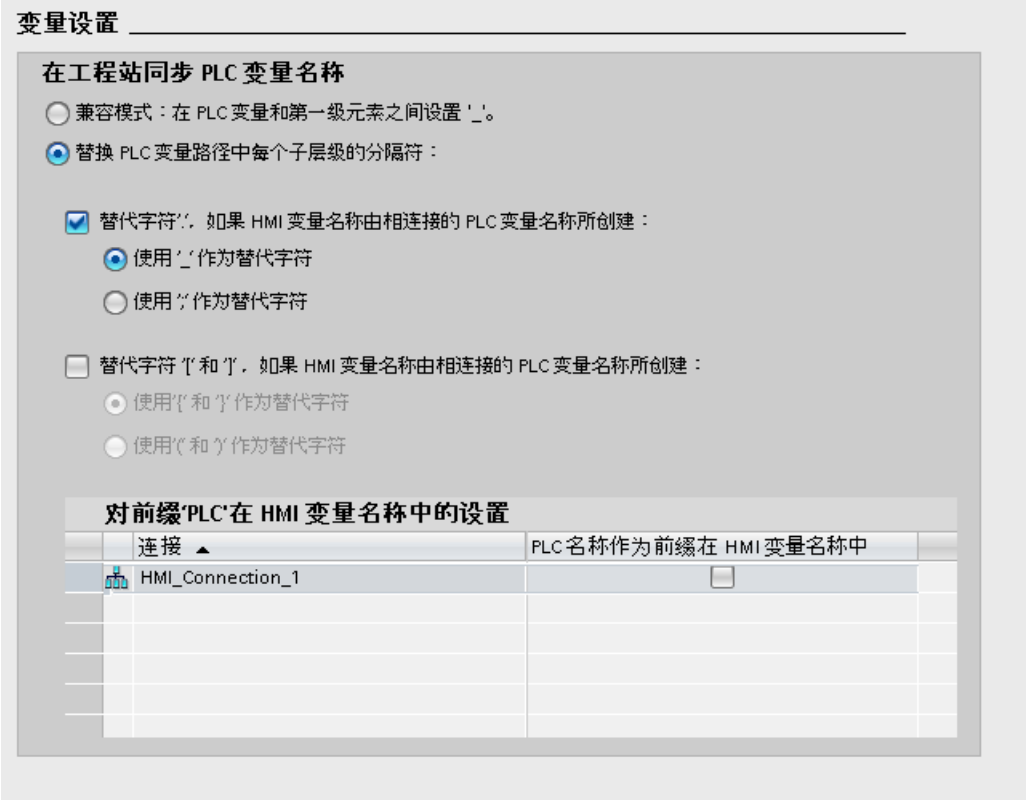
使用 SiVArc 对象属性 Name 和 SymbolicName 来引用 S7 控制器的名称或生成外部变量：

在“变量规则”(Tag rules) 编辑器中仅可以使用 SiVArc 表达式 HmiTag.SymbolicName 和 HmiTag.DB.SymbolicName 。

SiVArc 对象属性	引用的对象	SiVArc 表达式中的公式
Name	S7 PLC 的名称	S7Control.Name
SymbolicName	外部变量的名称（变量名称）	HmiTag.SymbolicName
DB.SymbolicName	DB 的名称	HmiTag.DB.SymbolicName
DB.FolderPath	DB 的路径	HmiTag.DB.FolderPath

用于外部变量名称同步的 SiVArc 对象属性

在 HMI 设备的运行系统设置中定义 PLC 变量和外部变量的名称如何同步：



要使用 SiVArc 根据 TIA Portal 中的变量设置同步外部变量的名称，需要使用 `TagNaming` 变量。

SiVArc 对象属性	引用的对象	SiVArc 表达式中的公式
`SeparatorChar`	PLC 变量路径的每个子级上使用以下分隔符：“.”、“_”、“;”	`TagNaming.SeparatorChar`
`IndexStartChar`	PLC 变量路径的每个子级上使用以下分隔符：“[”、“(”、“{”	`TagNaming.IndexStartChar`
`IndexEndChar`	PLC 变量路径的每个子级上使用以下分隔符：“]”、“)”、“}”	`TagNaming.IndexEndChar`

参见

“变量规则”编辑器 (页 31)

8.4 函数

8.4.1 SiVArc 中的函数

在 SiVArc 中，对以下部分中所列出的函数进行了定义。

可在 SiVArc 表达式中使用函数。无法更改函数名称。

8.4.2 “Contains”函数

Contains 函数

Contains 函数用于确定一个字符串是否包含在另一个字符串中。该函数区分大小写和空格。

函数	结果
Contains("ButtonText", "Text")	True
Contains("ButtonText", "ttonT")	True
Contains("ButtonText", "butt")	False
Contains("ButtonText", "txeT")	False
Contains("ButtonText", "Text")	False
Contains("ButtonText", "Text ")	False
Contains("ButtonText", "Te xt")	False
Contains("ButtonText", "on")	False
Contains("ButtonText 1", "ButtonText 2")	False

8.4.3 “EndsWith”函数

EndsWith 函数

EndsWith 函数用于确定一个字符串的结尾是否与指定字符串匹配。该函数区分大小写和空格。

函数	结果
EndsWith("ButtonText", "Text")	True
EndsWith("ButtonText", "ButtonText")	True
EndsWith("ButtonText", "butt")	False
EndsWith("ButtonText", "Butt")	False
EndsWith("ButtonText", "Text")	False
EndsWith("ButtonText", "Text ")	False
EndsWith("ButtonText", "Te xt")	False
EndsWith("ButtonText", "t")	True
EndsWith("ButtonText", "T")	False
EndsWith("ButtonText ", "Text")	False
EndsWith("ButtonText 1", "ButtonText 2")	False

8.4.4 “Format”函数

Format 函数

Format 函数可返回格式化的字符串。格式字符串中的语句可指定返回字符串的格式。

该函数有两个函数参数：

- 返回的格式化字符串。
- 用于指定字符串格式的格式字符串。
  使用格式字符串“b”可以二进制代码显示结果。如果表达式的结果为符点数，结果将会在四舍五入后以二进制格式显示。

函数	结果
Format(5,"0.00")	5.00
Format((VAR_1 Or 2#11100), "b")	2#11101

有关格式字符串的更多详细信息，可在 Microsoft Developer Network 上搜索“Strings.Format 方法”。

8.4.5 “FormatNumber”函数

FormatNumber 函数

FormatNumber 函数以数字的形式返回格式化的字符串。

该函数有五个函数参数：

位置	参数	描述	说明
1	表达式	返回的格式化为数字形式的字符串。	如果无法将字符串格式化为数字（例如“hello world”），则显示错误。
2	NumberofDigitsAfterDecimalPoint	用于指定十进制分隔符右侧所显示的小数位数的数字。 FormatNumber("12,4",3,-2,-2,-2) ("12 comma 4")= 12,400	默认值 -1 表示使用计算机的国家设置。
3	ApplyLeadingNumber	用于指定分数是否显示前导零的数字。 FormatNumber("0,4",3,-1,-2,-1) = 0,400	“常量列表”(List of constants) 下列出了可行的设置。

位置	参数	描述	说明
4	UseHigherLevelAsNegativeNumbers	用于指定是否在括号中显示负值的数字。 `FormatNumber("-12",1,-2,-1,0)` = (12,0)	如果在括号中显示数字，则不显示负号。 “常量列表”(List of constants) 下列出了可行的设置。
5	GroupNumbers	用于指定是否使用千位分隔符对大数字分组的数字。 `FormatNumber("1288,4",3,-2,-2,0)`= 1288,400	千位分隔符的形式在计算机的国家/地区设置中定义（例如，点、逗号或空格）。 “常量列表”(List of constants) 下列出了可行的设置。

常量列表

表格 8-1 ApplyLeadingNumber

ApplyLeadingNumber	值	
TRUE	-1	显示前导零。
FALSE	0	不显示前导零。
UseDefault	-2	使用计算机的国家设置。

表格 8-2 UseHigherLevelAsNegativeNumbers

ApplyLeadingNumber	值	
TRUE	-1	在括号中显示负值。不显示负号。
FALSE	0	输出负值，不使用括号。显示负号。
UseDefault	-2	使用计算机的国家设置。

表格 8-3 UseHigherLevelAsNegativeNumbers

ApplyLeadingNumber	值	
TRUE	-1	使用千位分隔符对数字分组。
FALSE	0	不使用千位分隔符对数字分组。
UseDefault	-2	使用计算机的国家设置。

示例

下表适用于德国设置。德国的千位分隔符是点，十位分隔符是逗号。

函数	结果
`FormatNumber("12,4",3,-2,-2,-2) ("12 comma 4")`	12.400
`FormatNumber("12.4",3,-2,-2,-2) ("12 point 4")`	124.000
`FormatNumber("1288,4",3,-2,-2,-1)`	1.288,400
`FormatNumber("1288,4",3,-2,-2,0)`	1288.400
`FormatNumber("-12",1,-2,-2,0)`	-12.0
`FormatNumber("-12",1,-2,-1.0)`	(12.0)

8.4.6 函数“InStr”

函数 `InStr`

`InStr` 函数用于检查一个字符串是否完全包含在另一个字符串中。区分大小写。该函数返回布尔值（“True”或“False”）。

该函数有两个函数参数：

- 在其中执行检查的字符串。
- 包含比较文本的字符串。

下列示例显示了 InStr 函数产生的值：

函数	结果
InStr("Hello","Hello")	True
InStr("Hello","hello")	False
InStr("Hello","el")	True
InStr("12345",3)	True
InStr("12345","6")	False

8.4.7 函数“IsDefined”

IsDefined 函数

使用字符串作为参数，IsDefined 函数可用于检查是否存在名称与指定字符串匹配的变量。

可将此函数用于以下语法元素：

- SiVArc 变量
- SiVArc 对象属性
- “String”数据类型的数组

示例：已创建以下用户自定义变量：

```
ButtonText "Cycle_time"
```

函数	结果
IsDefined("ButtonText")	True
IsDefined("ButtonText[0]")	True
IsDefined("ButtonText[1]")	True
IsDefined("ButtonText[2]")	False

8.4.8 函数“LBound”

LBound 函数

LBound 函数使用数组作为参数并返回可能的最小索引。

函数	结果
LBound(Split("SG19_FG97_ST090", "_"))	0
LBound(Split("SG19_FG97", "_"))	0

8.4.9 函数“Left”

Left 函数

Left 函数可返回包含指定数量字符的字符串（始于字符串最左边的字符）。

该函数有两个函数参数：

- 从中返回子串的字符串。
- 指示子串字符长度的数字
  如果该数字为零，则返回空字符串。
  如果该数字大于指定字符串的字符数，则显示错误。

函数	结果
Left("ButtonText", 6)	“Button”
Left("ButtonText", 0)	"" （空字符串）
Left("ButtonText", "10")	“ButtonText”
Left("ButtonText", 11)	错误 （数字大于字符串的字符数）

8.4.10 函数“Len”

Len 函数

Len 函数可返回字符串的字符数。该函数将字符串作为函数参数。

函数	结果
`Len("ButtonText")`	10
`Len("")`	0
`Left("ButtonText", Len("ButtonText"))`	“ButtonText”

8.4.11 函数“LTrim””

LTrim 函数

LTrim 函数可删除字符串的前导空格。该函数将字符串作为函数参数。

函数	结果
`LTrim ("       ButtonText")`	“ButtonText”
`LTrim ("ButtonText")`	“ButtonText”

8.4.12 函数“Max””

Max 函数

Max 函数以两个数字作为参数并返回其中较大的数字。

函数	结果
`Max(12, 3)`	12
`Max(3, 123)`	123

8.4.13 函数“Mid”

Mid 函数

Mid 函数可从字符串的指定位置返回子串。

该函数有三个函数参数：

- 复制子串的来源字符串。
- 用于在字符串中指示起始位置的数字。
  如果起始位置数字大于字符串的字符数，则显示错误。
- 用于指示子串字符长度的数字（自起始位置起）。
  如果指定长度大于可行的最长子串长度（自字符串中的起始位置起），则显示错误。

函数	结果
`Mid("ButtonText", 5,3)`	"nTe"
`Mid("ButtonText", 0, 10)`	“ButtonText”
`Mid("ButtonText", 10, 3)`	错误 （起始位置大于字符串的字符数）
`Mid("ButtonText", 7, 10)`	错误 （指定的长度大于自位置 7 起可行的最大子串长度）

8.4.14 函数“Min”

Min 函数

Min 函数将两个数字作为参数并返回其中较小的数字。

函数	结果
`Min(12, 3)`	3
`Min(3, 123)`	3

8.4.15 函数“Replace”

Replace 函数

Replace 函数可在字符串中从左至右搜索子串并将子串替换为另一个子串。搜索功能区分大小写。将返回更改后的字符串。

该函数有三个函数参数：

- 从中查找并替换子串的字符串。
- 要查找的子串字符串。
  如果要查找的子串为空字符串，则之前传送的字符串将原样返回。
- 插入所查找到子串位置的字符串。

查找和替换功能在发现新的子串后继续执行。

函数	结果
`Replace("ButtonText", "Text", "Button")`	“ButtonButton”
`Replace("ButtonText", "ButtonText", "Hello World")`	“Hello World”
`Replace("aaa", "aa", "bb")`	"bba"
`Replace("a", "a", "a")`	"a"
`Replace("a", "", "b")`	"a"
`Replace("aA", "a", "b")`	"bA"

8.4.16 “Right”函数

Right 函数

Right 函数可输出子串（自字符串最右边的字符起）。字符数在调用函数时指定。

该函数有两个函数参数：

- 从中产生并返回子串的字符串。
- 用于指示返回的字符数（自最右边的字符起）的数字。
  如果该数字为零，则返回空字符串。
  如果该数字大于字符串的字符数，则显示错误。

函数	结果
`Right("ButtonText", 4)`	“Text”
`Right("ButtonText", 0)`	"" （空字符串）
`Right("ButtonText", 10)`	“ButtonText”
`Right("ButtonText", 11)`	错误 （数字大于字符串的字符数）

8.4.17 函数“RTrim”

RTrim 函数

RTrim 函数可删除字符串末尾的空格。将返回生成的字符串。

如果字符串末尾没有空格，则字符串将原样返回。

函数	结果
`RTrim("ButtonText  ")`	“ButtonText”
`RTrim("ButtonText  ")`	“ButtonText”

8.4.18 “Split”函数

Split 函数

Split 函数可拆分字符串。此函数的定界符可自定义。

该函数有两个函数参数：

- 字符串
- 定界符

根据使用的语法，将返回子串或所含的子串数：

- 返回值为子串
  `Split("<String>","<Separator>")(<Index>)`
  通过以 0 起始的索引引用子串。
- 返回值为所含子串数
  `Split("<String>","<Separator>").Length`

下列示例显示了 Split 函数产生的数值：

函数	结果
`Split("SG19_FG97_ST090","_")(0)`	SG19
`Split("SG19.FG97.ST090",".")(1)`	FG97
`Split("42",".")(0)`	42
`Split(".",".")(0)`	"" （空字符串）

下列示例显示了 Split 函数产生的子串数：

函数	结果
`Split("SG19_FG97_ST090", "_").Length`	3
`Split("SG19.FG97.ST090", ".").Length`	3

8.4.19 “StartsWith”函数

StartsWith 函数

StartsWith 函数用于确定一个字符串的开头是否与指定字符串匹配。该函数区分大小写和空格。

函数	结果
`StartsWith("ButtonText", "Butt")`	True
`StartsWith("ButtonText", "butt")`	False
`StartsWith("ButtonText", "Text")`	False
`StartsWith("ButtonText", "ButtonText")`	True
`StartsWith("ButtonText", " Butt")`	False

函数	结果
StartsWith("ButtonText", "Butt ")	False
StartsWith("ButtonText", "Bu tt")	False
StartsWith("ButtonText", "B")	True
StartsWith("ButtonText", "b")	False
StartsWith(" ButtonText", "Butt")	False
StartsWith("B uttonText", "Butt")	False
StartsWith("ButtonText 1", "ButtonText 2")	False

8.4.20 “StrComp”函数

StrComp 函数

StrComp 函数可比较两个字符串。该函数从第一个字符开始对字符串进行字母数字排序，并且区分大小写。函数将根据字符串排序返回数字。

可产生如下结果：

- 第一个字符串的排序在第二个字符串前。返回值为 -1。
  StrComp("ABCD", "Abcd") = -1
  StrComp("A", "a") = -1（“A”在字母表中排在“a”前)
- 第二个字符串的排序在第一个字符串前。返回值为 1。
  StrComp("ABCD", "AAcd") = 1
- 两个字符串相同。返回值为 0。
  StrComp("Abcd", "Abcd") = 0

8.4.21 “TrailNum”函数

TrailNum 函数

TrailNum 函数返回字符串中最后的正数值，例如程序块名称中的数字。

下列示例显示了 TrailNum 函数产生的数值：

函数	结果
TrailNum("42")	42
TrailNum("Anzahl42")	42
TrailNum("Anzahl0042")	42
TrailNum("Anzahl-42")	42
TrailNum("Minimum42_Maximum84")	84

下列示例显示了 TrailNum 函数在 SiVArc 表达式中的使用方式。

在 TIA Portal 中对符号名称为“SG19_FG97_ST090+IR001_FB”的函数块编程。

SiVArc 表达式	结果
"MyBlock_"&TrailNum(ModuleBlock.SymbolicName)	“MyBlock_1”
"MyBlock_"&TrailNum(ModuleBlock.SymbolicName[0])	“MyBlock_19”

如果不指定字符串索引，则会输出字符串值中的最后一个数字。

8.4.22 “Trim”函数

Trim 函数

Trim 函数可删除字符串首尾的所有空格。将返回生成的字符串。

如果字符串首尾均没有空格，则字符串将原样返回。

函数	结果
Trim("ButtonText")	“ButtonText”
Trim("ButtonText")	“ButtonText”
Trim("ButtonText")	“ButtonText”
Trim("ButtonText")	“ButtonText”

8.4.23 “UBound”函数

UBound 函数

UBound 函数可将数组作为参数并返回可能的最大索引。

函数	结果
UBound(Split("SG19_FG97_ST090", "_"))	2
UBound(Split("SG19_FG97, "_"))	1
UBound(Split("", "."))	0

8.5 运算符

可以在 SiVArc 表达式中使用以下运算符。

注意运算符区分大小写。一方面，对于逻辑运算符和位运算符而言，这与运算符本身有关。另一方面，必须考虑设置的与比较运算符有关的字符串情况，例如，比较两个字符串以检查名称是否相同。

算术运算符

算术运算符	示例	结果
+	4+2	6
-	4-2 -4+2	2 -2
*	4*2	8
/	4/2	2

关系运算符

关系运算符	示例	结果
=	4=4 4=2	True False
<> （“不等于”）	4<>4 4<>2	False True

关系运算符	示例	结果
>	4>2	True
	2>4	False
>=	4>=2	True
	4>=4	True
<	4<2	False
	2<4	True
<=	4<=2	False
	4<=4	True

逻辑运算符

逻辑运算符	示例	结果
And	True And True	True
	True And True	False
	False And False	False
Or	True Or True	True
	True Or False	True
	False Or False	False
Not	Not True	False
	Not False	True

按位运算符

按位运算符	示例	结果
And	16 And 16	16
Or	8 Or 4	12
Xor	3 Xor 1	2

用于字符串序列的运算符

并置运算符	示例	结果
&	"Tool"&"Bar"	ToolBar

字符串的优先级

下表说明了在 SiVArc 表达式中使用多个运算符时处理运算符的优先顺序。 1 具有最高优先级。

运算符	Not -（一元）	*, /	+, -	&	=, <> >, >= <, <=	And	Or	Xor
优先级	1	2	3	4	5	6	7	8

使用括号来更改处理顺序。

8.6 字符串索引

用法

包含一个字符串的子串通过 _ 字符分隔。要访问子串，可使用索引运算符 `[]`。

子串从 0 开始计数。可通过索引运算符中的编号对子串进行访问。

示例

在 TIA Portal 中，变量“FB_Name”使用值“SG19_FG97_ST090+IR001_FB”进行定义。

SiVArc 表达式中的字符串索引	结果
`FB_Name[0]`	SG19
`FB_Name[1]`	FG97
`FB_Name[2]`	ST090+IR001
`FB_Name[3]`	FB

8.7 If 条件

在 SiVArc 表达式中使用 If 运算符设定逻辑条件。

If 运算符

If 运算符具有下列语法：

```
If(<condition>, <thenExpression>, <elseExpression>)
```

`<condition>`	布尔或整数
`<thenExpression>`	当 `<condition>` 为 `True` 或非 `0` 整数值时会生成
`<elseExpression>`	当 `<condition>` 为 `False` 或 `0` 时会生成

也可嵌套条件，在 If 条件中使用另一个 If 条件。

示例

If 条件	结果
`If(True, "On", "Off")`	On
`If(0, "On", "Off")`	Off
`If(42, "On", "Off")`	On
`If(4>2, If(False, 4, 2), 42)`	2

8.8 PLC 变量支持的数据类型

SiVArc 支持 PLC 通过 WinCC V13.1 显示在 HMI 设备上的所有基本数据类型。

SiVArc 也支持结构化数据类型 ARRAY、STRUCT 和 UDT。

基本数据类型

名称	数据类型
BOOL	布尔值
BYTE	8 位二进制和十六进制数字
CHAR	ASCII 字符
DINT	双精度整数，有符号整数
DTL	日期和时间 （年-月-日-小时:分钟:秒:纳秒）
DWORD	32 位二进制和十六进制数字
DATE	增量为 1 天的 IEC 日期

名称	数据类型
DATE_AND_TIME	日期和时间 （年-月-日-小时:分钟:秒；定点数）
INT	整数，有符号整数
LDT	日期和时间 （年-月-日-小时:分钟:秒）
LINT	
LREAL	
LTIME	
LTIME_OF_DAY	
LWORD	
REAL	实数 （IEEE 浮点数）
S5TIME	S5T# 格式的时间段，增量为 10 ms 的 Step7 时间
SINT	
STRING	字符串
TIME	IEC 格式的时间段，增量为 1 s 的 IEC 时间，有符号整数
TIME_OF_DAY	增量为 1 ms 的时间
UDINT	
UINT	
ULINT	
USINT	
WORD	16 位二进制和十六进制数字
WString	长度可变的 Unicode 字符串
WChar	16 位的 Unicode 字符

结构化数据类型

SiVArc 支持结构化 PLC 变量，和所有针对 WinCC 发布的相关元素。生成期间，SiVArc 根据 PLC 变量创建结构化外部变量和元素。变量和元素自动连接到 PLC 变量及其元素。

名称	数据类型
ARRAY	数组
ARRAY DB	
ARRAY DB STRUCT	
STRUCT	结构
UDT	自定义数据类型 （PLC 数据类型）

说明

PLC 数据类型 (UDT) 的条件

如果 PLC 数据类型为结构化数据类型（STRUCT 或 UDT）的数组时，SiVArc 会在 WinCC 中将数组分解为多个此数据类型的单个变量。如果 PLC 数据类型包含元素形式的结构化数据类型的数组，这些数组在"HMI 变量"(HMI tags) 编辑器中将显示为结构化元素。

8.9 面板支持的系统函数

系统函数

可在 SiVArc 事件中使用以下系统函数，具体取决于为哪个 HMI 设备而生成：

系统函数	RT Advanced	RT Professional
ActivateScreen	x	x
DecreaseTag	x	x
IncreaseTag	x	x
InvertBit	x	x
InvertBitInTag	x	x
SetBit	x	x
SetBitInTag	x	x
SetTag	x	x

系统函数	RT Advanced	RT Professional
ResetBit	x	x
ResetBitInTag	x	x
ActivateScreenInScreenWindow	---	x
ActivatePreviousScreen	x	---
ShiftAndMask	x	---

8.10 在 SiVArc 编辑器中编辑视图

简介

用户可以在该编辑器或生成概览中对 SiVArc 规则进行过滤和排序，而不会影响生成顺序。如有必要，可一直存储新布局，直到下次启动 TIA Portal。用户在所有 SiVArc 编辑器中按列来对视图分组。这种情况下需禁用过滤功能。

当列表进行过滤或排序时，可以继续编辑 SiVArc 规则或创建新规则。有效的过滤标准应用于新规则和编辑后的规则。

说明

过滤后的编辑器中的新规则

如果在过滤后的编辑器中创建新规则，那么该新规则即为在最底部显示的规则的副本。如果列表根据 SiVArc 规则的名称排序，那么新的 SiVArc 规则不显示。

针对视图对编辑器的内容进行过滤

当“组”(Group) 按钮被禁用时，就可以对编辑器的内容进行过滤。

要在编辑器中过滤 SiVArc 规则，请按以下步骤操作：

1. 在编辑器的工具栏中，单击“过滤”(Filter) 按钮。
   过滤行将显示在编辑器标题下。
2. 打开所需列过滤单元格中的选择对话框。
3. 在选择对话框中，选择希望显示在编辑器中的对象。
   根据选择过滤各项规则。

要隐藏过滤行，请再次单击“过滤”(Filter) 按钮。

针对视图对编辑器的内容进行排序

当“组”(Group) 按钮被禁用时，就可以对编辑器的内容进行排序。

当列表显示为已过滤后，还可对 SiVArc 规则重新排序，反之亦然。

要在编辑器中对 SiVArc 规则进行排序，请按以下步骤操作：

- 单击显示画面排序所依照的列标题。
  显示画面按照所选列以字母顺序降序排列。当规则编辑器中包含组时，组中的规则也根据此列进行排序。

保存排序和过滤

要将规则的过滤或排序保存到下次启动 TIA Portal，请按照以下步骤操作：

- 在编辑器工具栏中，单击“保存窗口设置”(Save window settings) 按钮。
  当下次打开 TIA Portal 时，SiVArc 规则会根据上一次的设置排列和过滤。

重新分组显示画面

当显示画面首次打开时，其中的内容根据第一列分组显示。

要在编辑器中将内容重新分组，请按照以下步骤进行操作：

1. 要激活分组功能，请单击“组”(Group) 按钮。
   “组”(Group) 按钮随即会显示为按下状态。
2. 单击想要为其内容对显示画面进行分组的列标题。
   所有 SiVArc 规则或 SiVArc 对象均根据显示画面中所选列的内容分组。

8.11 画面图例

概述

符号和抽象显示按照如下方式在 SiVArc 文档的示意图中使用：

符号/显示	说明
	函数块
	OB1 中的实例（OB1 中的调用）

符号/显示	说明
	OB1 中的网络
	函数块类型（库类型）
	函数块类型（库类型）的实例
	数据块
SiV	SiVArc 组态
< ... >	访问数据块的 SiVArc 表达式
< ... >	访问函数块的 SiVArc 表达式
< ... >	访问网络条目的 SiVArc 表达式

工具提示

9

9.1 UMAC

9.1.1 CTHMISivarc

参见

创建 SiVArc 规则 (页 170)

SiVArc 规则 (页 166)

SiVArc 中的用户管理控制 (UMAC) (页 28)

索引

S

T

U

V

按

背

编

变

表

布

参

支

主

注

专

组