

SIEMENS

SIMATIC

过程控制系统 PCS 7 APL 设计指南

编程手册

前言	1
基础知识	2
块图标的规范	3
面板的规范	4
APL 的设计规范	5
依照 APL 创建项目特定块	6
附录	A

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自带的文件说明，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®的都是西门子股份有限公司的注册商标。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。

目录

1	前言	7
2	基础知识	9
2.1	使用《APL 设计指南》	9
2.2	应用策略	11
3	块图标的规范	15
3.1	块图标的模板画面	15
3.2	块图标结构	16
3.3	安排块图标中画面对象需遵守的规则	17
4	面板的规范	19
4.1	面板的模板画面	19
4.2	面板视图的结构	20
4.3	安排面板中画面对象需遵守的规则	22
4.4	排列面板中的画面对象	25
4.4.1	排列显示和操作对象	25
4.4.2	组态模拟值的显示	26
4.4.3	组态模拟值显示的高亮显示	27
4.4.4	组态互锁	28
4.4.5	组态操作模式的显示和切换	29
4.4.6	组态限值显示的值范围	30
4.4.7	组态棒图	32
5	APL 的设计规范	35
5.1	操作和显示对象的规范	35
5.1.1	标签规范	35
5.1.2	颜色规范	36
5.2	视图规范	38
5.2.1	标准视图的结构	38
5.2.2	预览视图的结构	39
5.2.3	批生产视图的结构	41
5.2.4	限值视图的结构	41
5.2.5	参数视图的结构	42
5.2.6	斜坡视图的结构	43
6	依照 APL 创建项目特定块	45

6.1	简介	45
6.2	修改 AS 块	46
6.2.1	修改 AS 块（总览）	46
6.2.2	结构	46
6.2.2.1	作为结构的参数	46
6.2.2.2	构造一个结构并对其进行赋值	47
6.2.2.3	自定义数据类型 (UDT) 的使用	48
6.2.2.4	结构中元素的属性	49
6.2.3	状态字	52
6.2.4	备注视图	54
6.2.5	显示最差信号状态	55
6.2.6	显示测量单位	56
6.2.7	参数 OS_Perm/OS1Perm	57
6.2.8	Feature 参数	60
6.2.9	模拟值结构	61
6.2.10	连接 SIMATIC BATCH	63
6.2.11	仿真	64
6.2.12	“停止服务”功能	66
6.2.13	跳转到其它面板	69
6.2.14	维护发布	71
6.3	创建块图标	74
6.3.1	修改块图标（总览）	74
6.3.2	常规属性的描述	75
6.3.2.1	“Configurations”属性组	75
6.3.2.2	“System”属性组	76
6.3.2.3	“Trends”属性组	78
6.3.2.4	“Links”属性组	79
6.3.3	通过块图标操作	82
6.3.3.1	准备工作	82
6.3.3.2	调用操作对话框但不检查操作员权限	84
6.3.3.3	调用操作对话框前检查操作员权限	84
6.4	创建面板	86
6.4.1	生成现有面板的副本	86
6.4.2	修改面板（总览）	88
6.4.3	创建面板	89
6.4.3.1	画面 @PG_MyAPLFP.pdl 中的通用规范	89
6.4.3.2	画面 @PG_MyAPLFP_Overview.pdl 中的通用规范	90
6.4.3.3	画面 @PG_MyAPLFP_Standard.pdl 中的通用规范	90
6.4.3.4	更改视图选择	91
6.4.3.5	更改或分配操作员权限	94
6.4.3.6	模拟值的数字格式	95
6.4.4	面板对象的全局属性	97
6.4.4.1	对象的颜色梯度	97

6.4.4.2	TypeID 和 VersionID	98
6.4.4.3	ShowSelectionBorder.....	99
6.4.5	显示和操作模拟值.....	99
6.4.5.1	APL_ANALOG_OP_DISPLAY 模拟值显示	99
6.4.5.2	针对 APL_ANALOG_OP_DISPLAY 对象的模拟值操作	103
6.4.5.3	棒图.....	107
6.4.6	显示和操作二进制值	110
6.4.6.1	二进制值操作 APL_OP_BUTTON.....	110
6.4.6.2	复选框	115
6.4.6.3	使用文本框显示二进制值.....	119
6.4.6.4	信号状态显示.....	122
6.4.6.5	显示 OS_Perm 操作锁.....	123
6.4.7	创建自己的位图和状态显示	125
6.4.8	特殊要求的解决方案	126
6.4.8.1	在 V7 块图标的 ToolTipText 中显示块注释.....	126
6.4.8.2	多实例面板	128
A	附录	131
A.1	对象类型的类型 ID.....	131

前言

简介

“Advanced Process Library”(APL) 是用于实施自动化和过程控制解决方案的 PCS 7 标准库。 APL 包含功能块和相关的面板。 有关操作和监视用户界面的各种面板采用了统一的设计。

《APL 设计指南》提供了依照 APL 修改项目特定块的相关信息、设计规范和适用规则。 同时还可通过操作示例，对需要进行的相关修改进行说明。

在完成全部修改工作后，经修改的项目特定块与 APL 块的行为方式类似。

目标用户群

《APL 设计指南》主要面向要使用标准 APL 块以及希望依照 APL 的规范来修改项目特定块的人员。

说明

《APL 设计指南》的目的并不是帮助用户创建全新的技术库。

限制与要求

《APL 设计指南》中对有关块的创建和编程的文档进行了补充说明。 《APL 设计指南》仅包含有关 APL 特定的属性和功能信息。

使用《APL 设计指南》需要具备以下有关 PCS 7 的丰富知识：

- 创建块和面板
- 使用画面窗口技术
- 动态化对象属性
- 用 SCL 对块编程

有效性

《APL 设计指南》适用于以下版本的 PCS 7 及更高版本： PCS 7 V8.0。

- 对于基于 PCS 7 V8.0 的 APL 库所创建的项目特定的面板，仍然适用。
- 《APL 设计指南》主要介绍了对 PCS 7 模板画面及其所包含对象的运用。

内容

《APL 设计指南》是对编程手册《*SIMATIC; PCS 7 过程控制系统：块编程指南*》的补充。

《APL 设计指南》的结构和内容如下：

- 简介
包含有关此设计指南、目标用户群和相关要求的信息。
- 基础知识 (页 9)
包含有关设计指南的应用以及实施策略的实践型应用的更多信息。
- 块图标的规范 (页 15)
包含构建和组态项目特定块图标的相关规则。
- 面板的规范 (页 19)
包含构建和组态项目特定面板的相关规则。
- APL 的设计规范 (页 35)
包含依照 APL 设计块图标和面板的通用规范。
- 创建项目特定的块图标和面板
包含修改项目特定块的所需操作：
 - 修改 AS 块 (页 46) (创建结构和参数)
 - 创建块图标 (页 74)
 - 创建面板 (页 86)

基础知识

2.1 使用《APL 设计指南》

基于 APL 所创建的块的优势

基于 APL 创建的 PCS 7 项目具有以下优势：

- 通过将标准化块用于电机、阀门、闭环控制器和监视，可以使组态工作更为简便。
- 项目特定块将具有统一的设计。
- 从功能和操作方面来考虑，都可将项目特定块集成到 APL 中。
- 可依照 APL 创建项目特定的块。

使用《APL 设计指南》

通常按以下说明来使用本《APL 设计指南》：

项目特定库及其包含的工艺块和项目特定块都保存在现有项目中。在新项目中会将 APL 的标准块作为新工艺块与项目特定块一同使用。

- 项目特定块的程序代码和接口应尽可能保持不变。
- 因为相关面板与 APL 的设计不符，用户不得不进行相应的修改。

按照设计 APL 面板的规则和组态说明来进行必要的修改。《APL 设计指南》将提示用户如何执行如下操作：

- 应用颜色
- 安排画面对象
- 将面板连接到过程

AV 和 EventTs 块的功能

AV 和 EventTs 块的功能仅限于标准 APL 块。此功能不适用于用户块。

2.1 使用《APL 设计指南》

更新 APL 后丢失了对 APL 面板和 APL 块的项目特定修改内容

- 建议：
对于项目中项目特定的块，仅使用重命名后的 APL 块副本。此方法可避免在更新库时创建版本不一致的 APL。
- 如果修改当前的 APL 面板和 APL 块图标，那么在更新 APL 时新版本的面板和块图标将覆盖这些修改。

2.2 应用策略

初始状态

通常，项目特定的块在以下方面与 APL 块不同：

- 功能块接口的结构
- 块图标中的信息表示方式
- 面板的设计
- 面板中的功能

修改

- 功能块的程序代码将几乎保持不变。
- 需依照 APL 来修改项目特定块的功能和接口。
- 可通过以下方式创建项目特定的面板：
 - 例如，可使用现有 APL 面板的面板结构为基础来创建新面板。
 - 直接创建新的项目特定面板。

模板画面

以下模板画面包含用于项目特定类型的画面对象和块图标：

- "@PCS7ElementsAPL.PDL"
包含用于创建面板的画面对象。
- "@PCS7TypicalsAPL... .PDL"
包含用于自动放置和更新对象的阀门或电机的块图标和静态画面对象。
- "@TemplatesAPL... .pdl"
包含用于手动放置和更新对象的阀门或电机的块图标和静态画面对象。

说明

PCS 7 V8.0 中的 "@PCS7...APL... .PDL" 模板画面的版本

PCS 7 V8.0 包含以下几个版本的模板画面：

- "@PCS7TypicalsAPLV... .PDL" 的版本
 - @PCS7TypicalsAPLV7.PDL
 - @PCS7TypicalsAPLV8.PDL
- "@TemplatesAPLV... .PDL" 的版本
 - @TemplatesAPLV7.PDL
 - @TemplatesAPLV8.PDL

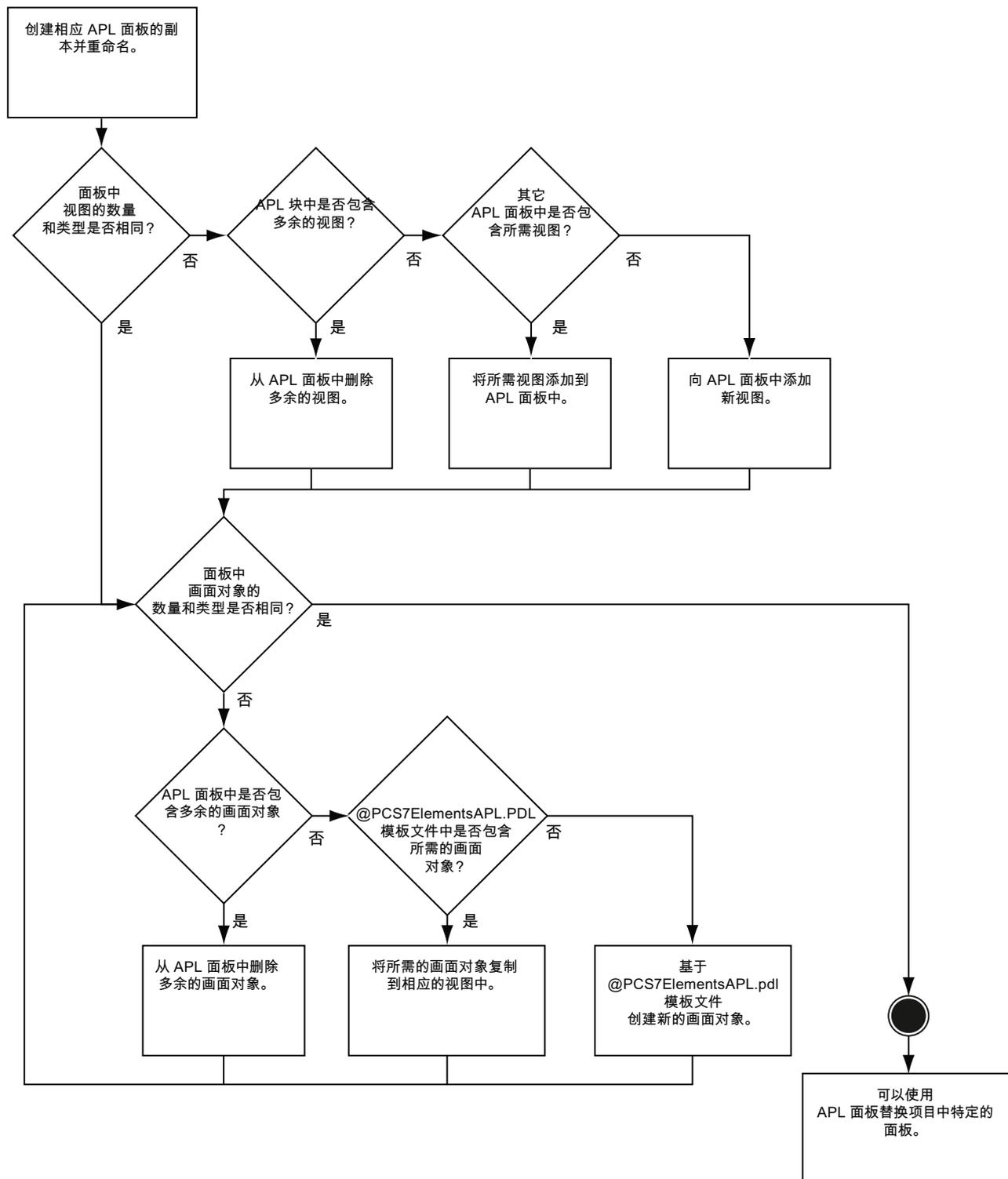
在手册中，使用了"@PCS7TypicalsAPLV8.PDL"和"@TemplatesAPLV8.PDL"模板画面的对象。

有关其他信息，请参见“面板的模板画面 (页 19)”一章。

修改项目特定的面板

- 在创建项目特定的面板前，请检查以下事宜：
 - 能否直接用 APL 块替换项目特定面板？
 - 如果存在功能上最接近项目特定面板的 APL 面板，则按以下流程图执行操作。
- 如果未找到功能上最接近项目特定面板的 APL 面板，可创建一个新面板。
为此，请使用 @PCS7ElementsAPL.pdl 模板画面中的画面对象。

流程图



建议

尽可能基于现有 **APL** 面板来创建项目特定面板。从模板中复制缺少的对象。只有在 **APL** 中找不到合适模板的情况下，再行创建新画面对象。

块图标的规范

3.1 块图标的模板画面

说明

"@PCS7TypicalsAPLV8.PDL" 和 "@TemplatesAPLV8.PDL" 模板画面包含预组态的块图标和静态画面对象。通过拖放操作，可将所需的块图标和画面对象应用于项目特定的模板画面。

说明

定期更新模板画面

只能将以下模板画面中的块图标用于过程画面：

- "@PCS7TypicalsAPLV8.PDL"
- "@TemplatesAPLV8.PDL"

其中包含的最新块图标总是符合正确的样式和设计规范。

内容

模板画面包含以下对象：

- 显示电机和阀门状态的静态符号
- 用于每个 APL 面板的各种不同块图标

参见

修改块图标（总览）（页 74）

3.2 块图标结构

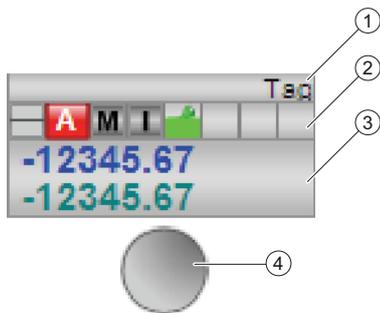
尺寸

块图标的尺寸如下：

- 宽（不包括符号）： 101 个像素或 115 个像素
- 宽（包括符号）： 115 个像素或 151 个像素
- 高： 变量

结构

下图显示了一个块图标结构的示例：



- ① 显示实例特定名称（必需）
- ② 块状态的状态显示（必需）
- ③ 模拟值显示（可选）
- ④ 块操作的符号（可选）
符号可放置在上方、下方、左侧或右侧。

参见

修改块图标（总览）(页 74)

3.3 安排块图标中画面对象需遵守的规则

在块图标中安排画面对象不需要遵守任何规则。

建议

在创建项目特定的块图标时，使用块图标模板画面中的块图标作为参考。

3.3 安排块图标中画面对象需遵守的规则

面板的规范

4.1 面板的模板画面

说明

面板的模板画面包含预组态的画面对象，可将这些画面对象拖放到项目特定的面板视图中。

面板的模板画面称为“@PCS7ElementsAPL.PDL”，默认情况下存储在 PCS 7 OS 项目的“GraCS”目录中。

说明

定期更新模板画面

仅将模板画面“@PCS7ElementsAPL.PDL”中的画面对象用于项目特定的面板。这种方法可确保在面板中始终使用样式和设计规范均正确的最新画面对象。

内容

模板画面包含以下对象

- 用于状态显示的文本框和按钮
- 用于调用二进制操作区的按钮
- 用于显示过程值的模拟值显示
- 用于在面板之间导航的按钮
- 用于标记的文本框
- 棒图
- 用于在面板视图之间导航的按钮
- 基本对象（有关更多信息，请参见 WinCC 信息系统）
- APL 的符号（有关更多信息，请参见 APL 在线帮助）

参见

排列显示和操作对象 (页 25)

4.2 面板视图的结构

尺寸

“Advanced Process Library”的每个面板视图都具有以下尺寸：

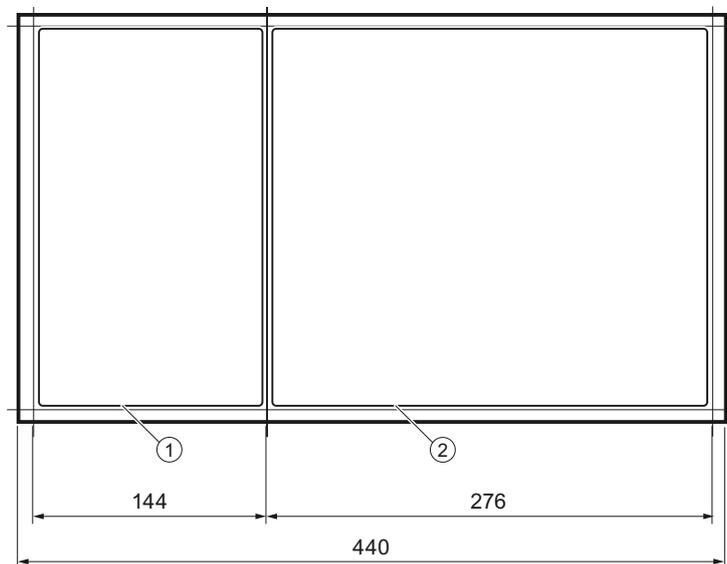
- 宽： 440 个像素
- 高： 变量
- 边距（上/下，左/右）： 10 个像素

指定的宽度对于包含“WinCC AlarmControl”或“WinCC TrendControl”的视图不适用。 在这种情况下，宽度取决于控件的宽度。

结构

“Advanced Process Library”的面板采用了带有边距的两栏式设计。 如果需要全新的视图，则可使用如下所示的基本设计。 APL 面板中所有视图的空面板视图均采用以下基本设计。

下图显示的是空面板视图网格，其尺寸以像素表示。



- ① “状态”区域
在此区域中可监视过程的状态信息。
- ② “显示和操作”区域
在此区域中，可监视过程值或使用单独的操作区域指定过程值。

坐标系和对象位置

画面上坐标系的原点位于左上角。将根据所选的矩形来计算对象的位置坐标。

例如，作为右侧栏标题的“静态文本”的坐标如下：

- 位置 X: 154
- 位置 Y: 10

有关此主题的更多信息，请参见“WinCC 信息系统”中的“使用画面”。

4.3 安排面板中画面对象需遵守的规则

画面对象的通用规范

通常，APL 的画面对象高 22 个像素。 APL 的面板视图使用以下画面对象类别：

- 文本框
- 操作和显示对象
- 分隔线

在面板视图中放置画面对象时，应遵守以下规则：

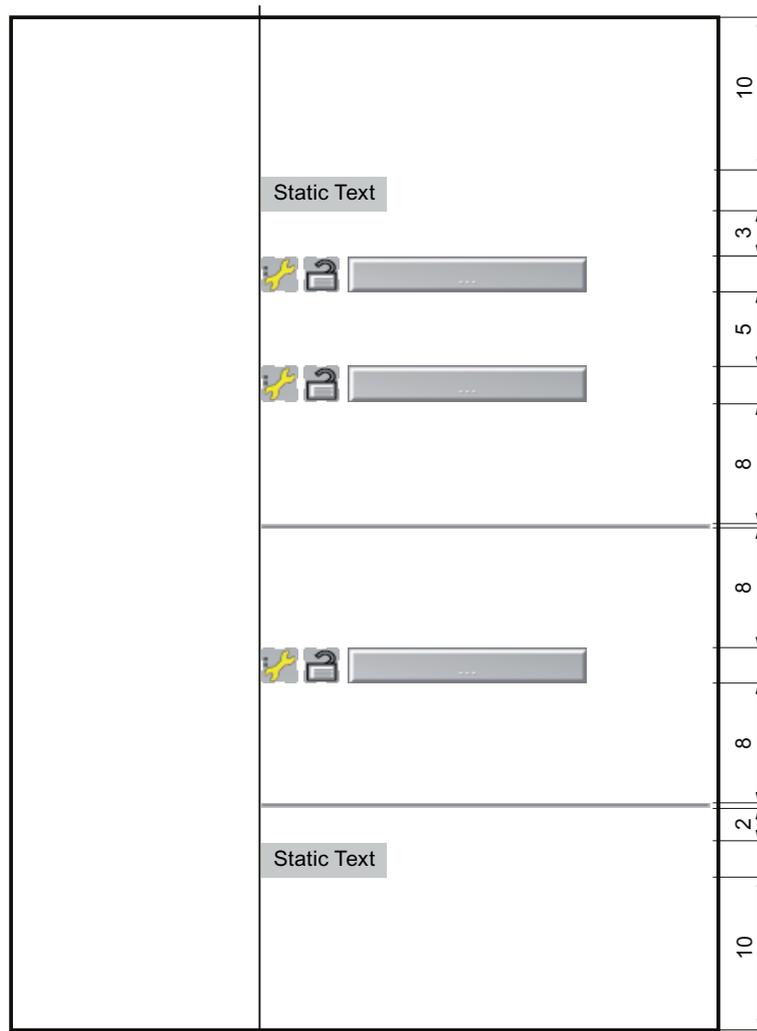
- 将文本框和/或操作和显示对象放在面板视图的开始位置。
- 不可将分隔线放在面板视图的开头或结尾位置上。
- 注意各画面对象与面板边距之间的距离。

画面对象之间的上/下距离规定

下表给出了各种画面对象类型之间的上下距离：

画面对象之间的距离	距离（像素）
分隔线与操作和显示对象之间	8
分隔线与文本框之间	2
文本框与操作和显示对象之间	3
显示和操作对象与显示和操作对象之间	5

下图以图解方式显示了在表格中给出的距离：



画面对象之间的左/右距离规则

画面对象之间的左右距离取决于相应面板视图的网格。

通常，两个画面对象之间的左右距离为 2 个像素。

4.3 安排面板中画面对象需遵守的规则

参见

- 标准视图的结构 (页 38)
- 预览视图的结构 (页 39)
- 批生产视图的结构 (页 41)
- 限值视图的结构 (页 41)
- 参数视图的结构 (页 42)
- 斜坡视图的结构 (页 43)
- 排列显示和操作对象 (页 25)

4.4 排列面板中的画面对象

4.4.1 排列显示和操作对象

简介

模板画面中的画面对象通常共同组成面板中的一个功能单元，例如，显示过程值及其允许限值的单元。

典型功能单元

《APL 设计指南》介绍了以下典型功能单元的所需画面对象及其在面板视图中的位置的列表：

- 模拟值显示
- 互锁
- 操作模式的显示和切换
- 限值显示的值范围
- 棒图

参见

组态模拟值的显示 (页 26)

组态互锁 (页 28)

组态操作模式的显示和切换 (页 29)

组态限值显示的值范围 (页 30)

组态棒图 (页 32)

安排面板中画面对象需遵守的规则 (页 22)

针对 APL_ANALOG_OP_DISPLAY 对象的模拟值操作 (页 103)

4.4 排列面板中的画面对象

4.4.2 组态模拟值的显示

应用

可使用模拟值显示来显示或输入模拟值。 如果还希望以图形方式来显示值，则必须组态棒图。 要在模拟值显示和棒图之间建立清晰的关系，可将棒图的颜色作为模拟值显示中边框的颜色。

下表列出了模板画面中的模拟值显示及其应用：

对象类型	在面板中的位置	框颜色 (HTML 颜色代码)	应用
APL_ANALOG_OP_DISPLAY	“显示和操作”区域	蓝色 (0000FF)	显示设定值 (包括权限检查)
APL_ANALOG_OP_DISPLAY2	“显示和操作”区域	绿色 (00B500)	显示以下值 (包括权限检查)： 表示较大的整数值，例如计数器块的值
APL_ANALOG_OP_DISPLAY3	“显示和操作”区域	深灰色 (808080)	显示斜坡的梯度 (包括权限检查)
APL_TIME_OP_DISPLAY2	“显示和操作”区域	红色 (FF0000)	显示持续时间
APL_ANALOG_OP_DISPLAY4	“显示和操作”区域	深灰色 (808080)	显示 PID 控制器的参数值

要求

- 图形编辑器已打开。
- "@PCS7ElementsAPL.pdl" 已打开。
- 面板视图已打开。

步骤

有关组态模拟值显示的信息，请参见章节“APL_ANALOG_OP_DISPLAY 模拟值显示 (页 99)”。

参见

排列显示和操作对象 (页 25)

面板视图的结构 (页 20)

组态模拟值显示的高亮显示 (页 27)

针对 APL_ANALOG_OP_DISPLAY 对象的模拟值操作 (页 103)

4.4.3 组态模拟值显示的高亮显示

简介

可通过以下方式在面板视图中高亮显示模拟值显示：

- 通过不同颜色进行高亮显示
在对模拟值显示进行操作时，其中的文字会通过颜色进行高亮显示。
- 钝化表示模拟值；例如，报警限值的限制视图；如果禁用，在面板中将以钝化方式显示。

说明

除模拟值显示外，通过颜色进行高亮显示还可用于“Checkbox”画面对象。

下表给出了用于空模拟值显示的对象：

对象类型	在面板中的位置	HTML 颜色代码	应用
APL_ANA_EMPT Y	“显示和操作”区域	深灰色 (808080)	空模拟值显示的显示方式取决于块输入或输出上的值
APL_ANA_EMPT Y2			
APL_ANA_EMPT Y3			

要求

- 图形编辑器已打开。
- "@PCS7ElementsAPL.pdl" 已打开。

4.4 排列面板中的画面对象

- 面板视图已打开。
- 已组态模拟值的显示。

步骤

按以下步骤组态模拟值显示的高亮显示：

- 要以颜色突出显示模拟值的显示，请选择相应的模拟值显示。
在组态对话框中，将“其它 > ShowSelectionBorder”属性设为“开”(On)。
- 要在已有的模拟值显示基础上再显示一个空模拟值显示，则可将空模拟值显示放置在
与现有模拟值显示相同的位置上。
组态相应的块输入或输出。

参见

组态模拟值的显示 (页 26)

ShowSelectionBorder (页 99)

APL_ANALOG_OP_DISPLAY 模拟值显示 (页 99)

4.4.4 组态互锁

应用

互锁是面板（例如电机）中与过程相关的禁用功能。

互锁的对象

对象类型	在面板中的位置	应用
APL_FACEPLATE3	“显示和操作”区域	执行锁定和解除锁定的按钮
APL_LOCK_SYMBOL	“显示和操作”区域	显示互锁状态
APL_QUALITY_CODE2 APL_QUALITY_CODE3	“显示和操作”区域	显示信号状态

要求

- 图形编辑器已打开。
- "@PCS7ElementsAPL.pdl" 已打开。
- 面板视图已打开。

步骤

按以下步骤组态互锁：

1. 从"@PCS7ElementsAPL.pdl"模板画面中将所需字段复制到面板视图中。
2. 设置以下值来放置互锁的对象：

对象	位置 X	位置 Y
互锁按钮	284	任意
表示互锁状态的符号	258	互锁按钮的“位置 Y”
表示信号状态的符号	234	互锁按钮的“位置 Y”

结果

互锁已组态完毕。

参见

排列显示和操作对象 (页 25)

面板视图的结构 (页 20)

4.4.5 组态操作模式的显示和切换

应用

显示和切换聚合操作的操作模式。

在面板中，电机或阀门的操作模式由一个文本框表示。并且提供了一个单独的操作区域来切换操作模式。通过组态附加按钮使操作员能打开此操作区域。

4.4 排列面板中的画面对象

要求

- 图形编辑器已打开。
- "@PCS7ElementsAPL.pdl" 已打开。
- 面板视图已打开。

步骤

按以下步骤组态操作模式的显示和/或切换：

1. 从"@PCS7ElementsAPL.pdl"模板画面中将所需字段复制到面板视图中。
有关“显示和操作”区域的各个属性的信息，请参见“使用文本框显示二进制值 (页 119)”部分。
2. 设置以下值来放置标签字段：

对象	位置 X	位置 Y
标签字段	154	任意
操作模式显示	284	标签字段的“位置 Y”
用于指定操作模式的按钮	401	标签字段的“位置 Y”

结果

操作模式切换已组态完毕。

参见

排列显示和操作对象 (页 25)

面板视图的结构 (页 20)

二进制值操作 APL_OP_BUTTON (页 110)

4.4.6 组态限值显示的值范围

应用

要向操作员显示过程值的允许范围，可在模拟值显示基础上组态限值显示。限值显示由另外两个字段组成，可放置在模拟值显示的上方和下方。

使用“Display range”对象实现限值显示。将名为“Display range”的对象放置在“显示和操作”区域中。

规则

以下规则适用于组态限值显示的值范围：

- 限值显示中最多可包括两个模拟值显示，例如，“设定值”(Setpoint) 和“回读值”(Read back value)。
- 遵守字段之间的以下垂直距离要求：
 - 限值显示和模拟值显示之间的距离：2 个像素
 - 两个模拟值显示之间的距离：5 个像素

要求

- 图形编辑器已打开。
- "@PCS7ElementsAPL.pdl" 已打开。
- 面板视图已打开。

步骤

按以下步骤组态限值显示的值范围：

1. 从"@PCS7ElementsAPL.pdl"模板画面中将所需对象复制到面板视图中。
2. 设置以下值来放置相应对象：

对象	位置 X	位置 Y
上限值显示	360	任意
模拟值显示	154	[上限值显示的“位置 Y”] + 24
用来指定下限值显示的按钮	360	[上限值显示的“位置 Y”] <ul style="list-style-type: none"> • + 48 (一个模拟值显示) 或者 • + 75 (两个模拟值显示)

4.4 排列面板中的画面对象

3. 通过每个限值显示的“其它 > 值”(Other > Value) 属性，对向限值显示提供值的变量进行组态。

结果：

对带限值显示的值规范进行组态。

4. 按照“针对 APL_ANALOG_OP_DISPLAY 对象的模拟值操作 (页 103)”部分所述为模拟值显示分配参数。

参见

排列显示和操作对象 (页 25)

组态模拟值的显示 (页 26)

面板视图的结构 (页 20)

4.4.7 组态棒图

应用

使用棒图以图形方式显示过程值。根据要显示的值来决定是使用垂直棒图还是使用水平棒图。

下表给出了模板画面中的棒图及其应用：

对象类型	在面板中的位置	棒图颜色 (HTML 颜色代码)	应用
APL_BAR_VERTIC_1	“状态”区域	绿色 (00B500)	显示以下值： <ul style="list-style-type: none"> 过程值 附加值 典型值：数量、温度
APL_BAR_VERTIC_2	“状态”区域	蓝色 (0000FF)	显示设定值
APL_BAR_GRADIENT_1	“状态”区域	灰色 (707070)	显示斜坡的梯度
APL_BAR_HORIZ_1	“显示和操作”区域	棕色 (D27A00)	显示调节变量
APL_BAR_HORIZ_2	“显示和操作”区域	深青色 (00CECA)	显示回读值
APL_BAR_TEXT_E	“状态”区域	蓝色 (0000FF)	显示外部设定值 ("E")

对象类型	在面板中的位置	棒图颜色 (HTML 颜色代码)	应用
APL_BAR_TEXT_R	“状态”区域	蓝色 (0000FF)	显示设定值斜坡的目标设定值 ("R")
APL_BAR_LINE_TEXT_E	“显示和操作”区域	棕色 (D27A00)	显示外部调节变量 ("E")
APL_BAR_LINE_TEXT_R	“显示和操作”区域	棕色 (D27A00)	显示斜坡目标值 ("R")

规则

以下规则适用于棒图：

- 组态每个棒图的相关模拟值字段：
 - 对于模拟值字段的边框，可使用棒图的颜色。
 - 将模拟值字段放置在棒图下方或旁边。
- 垂直棒图：
 - 在“状态”区域插入垂直棒图。
 - 居中放置垂直棒图。
 - 必要时，可将用于显示设定值斜坡的外部设定值以及目标设定值的对象放置在垂直棒图的左侧。
- 水平棒图：
 - 在“显示和操作”区域插入水平棒图。
 - 必要时，将显示设定值斜坡的外部设定值以及目标设定值的对象放置在水平棒图的上方。

要求

- 图形编辑器已打开。
- "@PCS7ElementsAPL.pdl" 已打开。
- 面板视图已打开。

4.4 排列面板中的画面对象

步骤

按以下步骤组态棒图：

1. 从“@PCS7ElementsAPL.pdl”模板画面中将所需棒图复制到面板视图中。
2. 通过以下设置可对齐两个棒图：

目标	操作
将垂直棒图以 2 个像素的间距居中放置。	<ul style="list-style-type: none"> • 设置“位置 X”的值： <ul style="list-style-type: none"> - 左侧棒图： 46 - 右侧棒图： 75 • 必要时，可将显示“E”和“R”放置在棒图的左侧。
将垂直棒图居中放置。	<ul style="list-style-type: none"> • 设置“位置 X”的值： 61
将水平棒图（一个或两个）以 2 个像素的间距左对齐放置。	<ul style="list-style-type: none"> • 设置“位置 X”的值： 154 • 设置“位置 Y”的值（棒图底部）： 棒图顶部的“位置 Y” + 24 个像素 • 垂直对齐棒图。 • 必要时，可将显示“E”和“R”放置在棒图的上方。

3. 如果已对显示“E”或“R”进行组态，则在“自定义 > BarGraphName”属性中输入每个对应棒图的对象名称。

结果：

棒图已放置完毕。

4. 按照“棒图 (页 107)”部分所述为棒图分配参数。

参见

排列显示和操作对象 (页 25)

组态模拟值的显示 (页 26)

组态限值显示的值范围 (页 30)

面板视图的结构 (页 20)

APL 的设计规范

5.1 操作和显示对象的规范

5.1.1 标签规范

简介

在面板中显示的所有文本或数字值都被视为是“标签”。以下对象包含文本和数字值：

- 静态文本
- 按钮
- 模拟值显示

规范

以下规范适用于面板中各种类型的文本：

字体 (FontName)	Arial
字体大小 (FontSize)	14 磅
粗体 (FontBold)	否
斜体 (FontItalic)	否
方向 (Orientation)	水平
Y 对齐 (AlignmentTop)	居中

对象特定的规范

以下对象特定的规范也适用：

按钮

下划线 (Underline)	否
X 对齐 (AlignmentLeft)	居中

填充图案规范

除颜色外，还可为背景定义填充图案。允许的填充图案类型如下：

- 透明
- 实心
- 图案

下表显示了块图标、面板视图和画面对象所允许的填充图案：

填充图案	允许的对象
0 	标签字段（静态文本）、模拟值显示 - 透明填充
1 	标签字段（静态文本）、模拟值显示 - 实心填充
5 	按钮
6 	垂直棒图
7 	水平棒图、模拟值显示、面板视图、块图标

5.2 视图规范

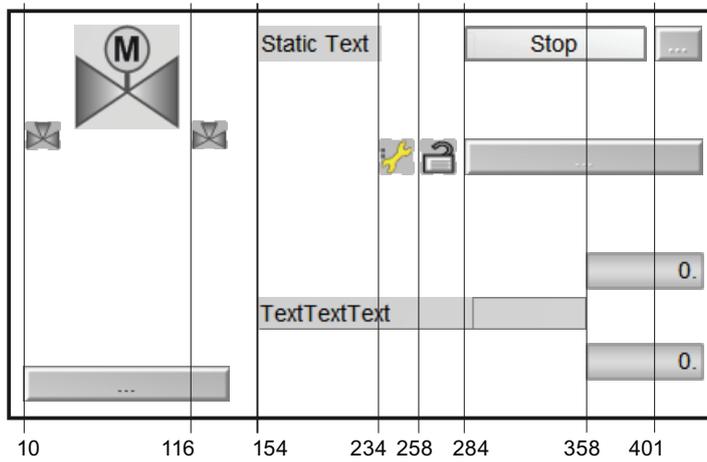
5.2.1 标准视图的结构

应用

操作员可在标准视图中监视和操作过程。

网格

下图显示了标准视图（包括网格的 X 坐标）的典型结构：



下表包含在网格中排列对象时的 X 坐标（像素）：

“位置 X”	说明	规则
10	“自动预览”符号表示的起始位置 状态框和按钮的起始位置	<ul style="list-style-type: none"> “状态”区域宽度为 144 个像素。 状态框和按钮的宽度为 128 个像素。 调整“自动预览”的符号表示，使其与状态框左对齐。 按以下方程式调整图形状态显示，使其相对于状态框居中放置： <ul style="list-style-type: none"> 位置 $X = ((128/2)+10) - (\text{对象宽度}/2)$ <p>如果有两个棒图，则两个棒图的对象宽度按以下方程式计算：</p> <ul style="list-style-type: none"> 对象宽度 = $(2 \times \text{宽度}) + 2$
116	安全位置符号表示的起始位置	<ul style="list-style-type: none"> 调整“安全位置”的符号表示，使其与状态框右对齐。
154	标签和标签化模拟值显示的起始位置	-
234	互锁符号的起始位置	<ul style="list-style-type: none"> 最多可组态两个宽度为 22 个像素的互锁符号。
258	互锁符号的起始位置	<ul style="list-style-type: none"> 从右向左放置互锁符号： <ul style="list-style-type: none"> 第一个互锁符号：位置 $X = 258$ 第二个互锁符号：位置 $X = 234$
284	按钮的起始位置	-
358	带上下限的字段的起始位置	-
401	用于调用操作区域的按钮的起始位置	-

参见

安排面板中画面对象需遵守的规则 (页 22)

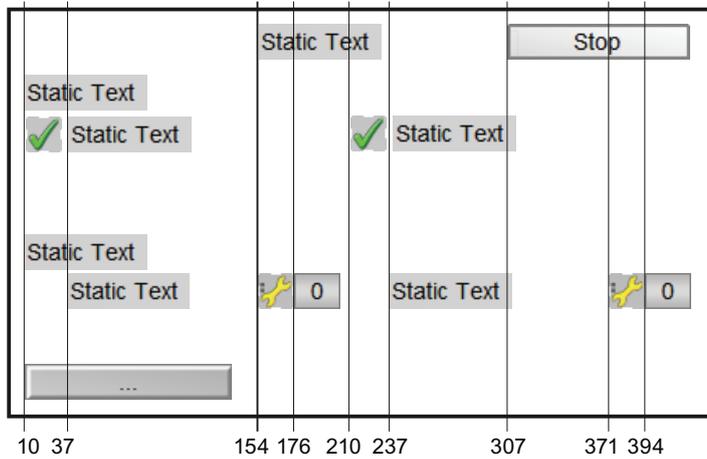
5.2.2 预览视图的结构

应用

预览会显示 OS 操作员可在整个块中操作的参数。

网格

下图显示了预览（包括网格的 X 坐标）的典型结构：



下表包含在网格中排列对象时的 X 坐标（像素）：

“位置 X”	说明	规则
10	标题的起始位置	-
10/210	状态显示（左/右栏）的起始位置	-
37/237	状态标签（左/右栏）的起始位置	-
154	标签和标签化模拟值显示的起始位置	-
154/371	信号状态显示（左/右栏）的起始位置	-
176/394	控制信号（左/右栏）的起始位置	-
307	按钮的起始位置	使用宽 113 个像素的按钮。

参见

安排面板中画面对象需遵守的规则 (页 22)

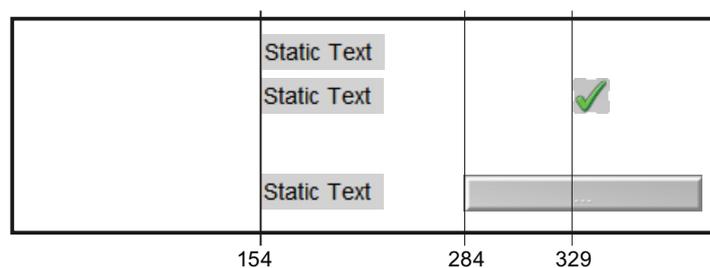
5.2.3 批生产视图的结构

应用

批生产视图可显示批生产处理的状态。

网格

下图显示了批生产视图（包括网格的 X 坐标）的典型结构：



下表包含在网格中排列对象时的 X 坐标（像素）：

“位置 X”	说明	规则
154	标签的起始位置	-
284	按钮的起始位置	-
329	状态显示的起始位置	-

参见

安排面板中画面对象需遵守的规则 (页 22)

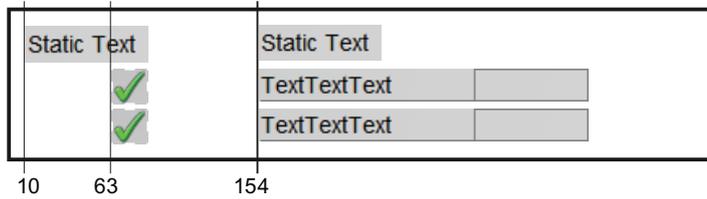
5.2.4 限值视图的结构

应用

操作员可在限值视图中指定限值。

网格

下图显示了限值视图（包括网格的 X 坐标）的典型结构：



下表包含在网格中排列对象时的 X 坐标（像素）：

“位置 X”	说明	规则
10	标题的起始位置	-
63	状态显示的起始位置	-
154	标题和标签化模拟值显示的起始位置	-

参见

安排面板中画面对象需遵守的规则 (页 22)

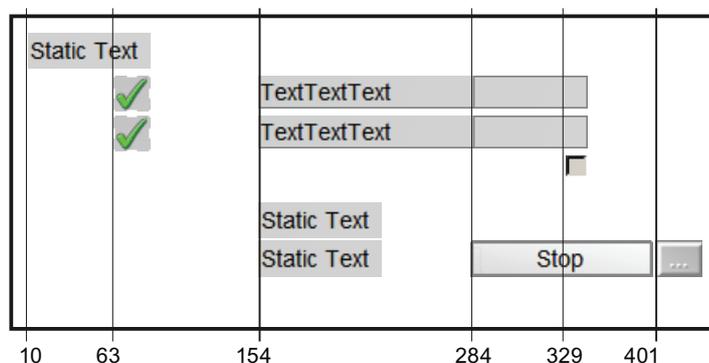
5.2.5 参数视图的结构

应用

在参数视图中，操作员可修改参数（例如，闭环控制器的参数）。

网格

下图显示了参数视图（包括网格的 X 坐标）的典型结构：



下表包含在网格中排列对象时的 X 坐标（像素）：

“位置 X”	说明	规则
10	标题的起始位置	-
63	状态符号的起始位置	-
154	标签化模拟值显示和标签的起始位置	
284	按钮的起始位置	使用宽 118 个像素的按钮。
329	复选框的起始位置	
401	用于调用操作区域的按钮的起始位置	

参见

安排面板中画面对象需遵守的规则 (页 22)

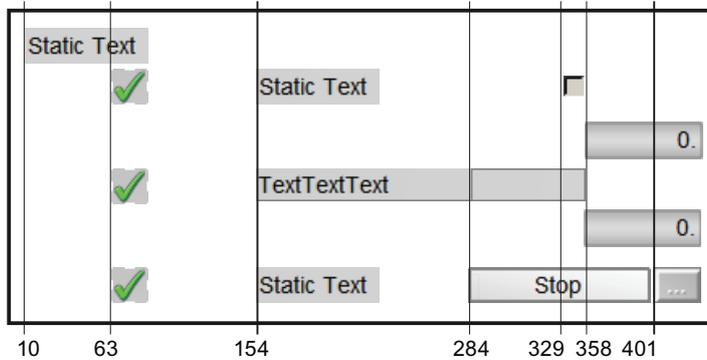
5.2.6 斜坡视图的结构

应用

可在斜坡视图中控制斜坡函数。

网格

下图显示了斜坡视图（包括网格的 X 坐标）的典型结构：



下表包含在网格中排列对象时的 X 坐标（像素）：

“位置 X”	说明	规则
10	标题的起始位置	-
63	状态符号的起始位置	-
154	标签和标签化模拟值显示的起始位置	-
284	按钮的起始位置	使用宽 118 个像素的按钮。
329	复选框的起始位置	-
358	带上下限的字段的起始位置	-
401	用于调用操作区域的按钮的起始位置	-

参见

安排面板中画面对象需遵守的规则 (页 22)

依照 APL 创建项目特定块

6.1 简介

下面部分提供了依照 APL 创建自定义块的相关信息。

要求

- 对 AS 块进行更改：
 - 依照 APL 修改 AS 块的输入和输出
 - 修改 APL 功能的块代码
- 使用以下对象作为块图标的来源：
APL 的标准块图标
- 使用以下对象作为面板的来源：
APL 的标准面板

AV 和 EventTs 块的功能

AV 和 EventTs 块的功能仅限于标准 APL 块。此功能不适用于用户块。

更新 APL 后丢失了对 APL 面板和 APL 块的项目特定修改内容

- 建议：
对于项目中的项目特定块，仅使用重命名后的 APL 块副本。此方法可避免在更新库时创建版本不一致的 APL。
- 如果修改当前的 APL 面板和 APL 块图标，那么在更新 APL 时新版本的面板和块图标将覆盖这些修改。

6.2 修改 AS 块

6.2 修改 AS 块

6.2.1 修改 AS 块（总览）

创建结构和参数

功能	部分
创建作为结构的参数	作为结构的参数 (页 46)
显示状态信息	状态字 (页 52)
创建备注视图中的文本	备注视图 (页 54)
确定对象的信号状态	显示最差信号状态 (页 55)
显示单位	显示测量单位 (页 56)
通过参数禁用或启用具体实例的操作	参数 OS_Perm/OS1Perm (页 57)
设计块行为方式	Feature 参数 (页 60)
根据用途设计模拟辅助值的可见性	模拟值结构 (页 61)
依照 APL 创建 SIMATIC BATCH 的接口	连接 SIMATIC BATCH (页 63)
在 APL 块中使用仿真	仿真 (页 64)
创建“停止服务”功能	“停止服务”功能 (页 66)
创建到其它面板的跳转	跳转到其它面板 (页 69)
创建维护发布	维护发布 (页 71)

6.2.2 结构

6.2.2.1 作为结构的参数

APL 块传送单独的参数和结构化参数。

参数的不同结构

APL 中的块存在以下参数类型：

- 属于互连的参数将作为结构来实现。

说明

例外

以下参数虽然互连，但并不作为结构来实现。

- PV_OutUnit
 - PV_Unit
 - EN
 - ENO
 - ANY
-

- 指定的参数不是结构。

说明

例外

以下参数作为结构来实现（无 Signal Status）：

- Feature
 - OS_Perm/OS1Perm
-

6.2.2.2 构造一个结构并对其进行赋值

APL 中主要使用了以下结构：

- 模拟值结构（见以下示例）
- 二进制值结构（与模拟值相同，结构元素 Value 的类型为 BOOL）

SCL 的语法

以下示例说明了如何声明“模拟值”结构的输入参数：

```
// 模拟值结构
PV:   STRUCT
      Value   : REAL := 0.0;      // Value
      ST      : BYTE := 16#80;   // Signal Status
END_STRUCT
```

以下示例说明结构赋值：

6.2 修改 AS 块

```
// 赋值
PV.Value    := X1 + X2;
PV.ST       := 16#78;
```

6.2.2.3 自定义数据类型 (UDT) 的使用

如果使用结构化参数，需将其创建为自定义数据类型 (UDTs)。

步骤

按以下示例脚本所示来创建 SCL 文件中的数据类型。将数据类型的符号名称定义如下：

- "Type AnaVal" 用于模拟值结构
或者
- "Type DigVal" 用于二进制值结构

```
// 示例脚本“UDT51”： 创建数据类型
TYPE AnaVal
  STRUCT
    Value    : Real := 0.0;      // Value
    ST       : Byte := 16#80;    // Signal Status
  END_STRUCT
END_TYPE
```

```
// 示例脚本“UDT61”： 创建数据类型
TYPE DigVal
  STRUCT
    Value    : Bool := false;   // Value
    ST       : Byte := 16#80;    // Signal Status
  END_STRUCT
END_TYPE
```

在编译 SCL 源代码之前，应先在符号表中为符号名称指定一个 UDT 编号。

说明

保留的 UDT 编号

保留 1 到 50 的 UDT 编号，供 APL 数据类型使用。

Status	Symbol	Address	Data type	Comment
39	AnaVal	UDT 51	UDT 51	
40	DigVal	UDT 61	UDT 61	
41				

图 6-1 示例： AnaVal = UDT 51 und DigVal = UDT 61

在编译 SCL 源文件时，系统将创建自定义数据类型并将其存储在块文件夹内：

Object name	Symbolic name	Created in language
UDT51	AnaVal	STL
UDT61	DigVal	STL

此时可按如下所示在块中创建结构参数

```
// 示例脚本： 作为结构的参数
VAR_INPUT
    PV           : AnaVal;
    FbkOpen     : DigVal;
```

在编译过程中，符号名称（例如 DigVal）将被替换为自定义数据类型。

说明

供应库

要对该块进行编译，只需要此块文件夹中自定义的数据类型。供应库中可以没有这些自定义数据类型。

6.2.2.4

结构中元素的属性

在编译 OS 时，APL 使用属性以变量形式创建结构中的各个元素。

属性均以前缀“S7_”开头。如果某属性引用了结构中的一个元素，则前缀后将跟随一个“x”，例如“S7_x”。

这些结构的语法保持不变。

规则

说明

检查语法

SCL 编译器不会对属性分支中 S7_x 属性的引号里面的内容进行语法检查。
 请确保语法正确，而且所列出的元素名称与“S7_x”属性中的一致。

以下规则适用于属性：

- 前缀为 "S7_" (无 x) 的属性

这些属性适用于整个结构。

示例： S7_visible, S7_dynamik, S7_contact.

语法与用于非结构化参数属性的语法相同。

- 前缀为 "S7_x" 的属性

这些属性始终引用结构中的元素。

其语法规则是，用逗号分隔分配给某个属性的结构元素。

示例： 请参见示例脚本“Struct_4” ⇒ PV_OpScale

```
- S7_m_c := 'true';
```

此表达式指定用户可以在 PCS 7 OS 中为该结构创建变量。

```
- S7_xm_c := 'High,true;Low,true;';
```

该表达式指定将 PV_OpScale 结构的 "High" 和 "Low" 元素创建为 PCS 7 OS 的变量。

Name
ACC_APL/ACC_APL.PV#Value
ACC_APL/ACC_APL.PV_Unit
ACC_APL/ACC_APL.PV_OpScale#High
ACC_APL/ACC_APL.PV_OpScale#Low

图 6-2 Struct_5

示例

```
// 示例脚本“Struct_4”：
// 将结构元素创建为变量
PV      {S7_dynamic := 'true';
         S7_visible   := 'true';
         S7_m_c       := 'true';
```

```

// 示例脚本“Struct_4”:
// 将结构元素创建为变量
    S7_xm_c      := 'Value,true;';
    S7_xshortcut := 'Value,true;'}
: STRUCT
    Value      : REAL := 0.0;      // Value
    ST         : BYTE := 16#80;    // Signal Status
END_STRUCT;

PV_OpScale {S7_m_c:='true';
    S7_xm_c      := 'High,true;Low,true;';
    S7_edit      := 'para';
    S7_xedit     := 'High,para;Low,para;'}
: STRUCT
    High        : REAL := 100.0;   // High Value
    Low         : REAL := 0.0;     // Low Value
END_STRUCT ;      // PV - Bar Display Limits for OS

```

以下表达式定义了曲线控件的测量范围:

- S7_xtrend:='Value,PV_OpScale.Low,PV_OpScale.High;

以下表达式指定在创建实例时创建短期归档的归档变量:

- S7_xarchive:='Value,shortterm;';

可用于结构的 S7_x 属性如下

属性	要求/属性
S7_xm_c	对于结构本身, S7_m_c 是必需的
S7_xshortcut	
S7_xstring_0	
S7_xstring_1	
S7_xedit	
S7_xqc	
S7_xarchive	(APL 块不使用)
S7_xtrend	
S7_contact	
S7_visible	
S7_dynamic	

6.2 修改 AS 块

6.2.3 状态字

状态字（双字）用于在 APL 面板和 APL 块图标中显示状态信息。

向 AS 块中添加这些状态字来使用 OS 对象。

建议：

使用 SCL 命令“AT”将二进制信息分配到双字中。使用“AT”命令可以不同的方式来显示数据块中的数据类型或这些数据类型的存储区域。

示例：

在以下示例中，已将“Status1”创建为 DWORD 类型的输出参数：

```
VAR_OUTPUT
```

```
    Status1 : DWORD;
```

```
    Status1Bits AT Status1 : ARRAY[0..31] OF Bool;
```

- "Status1Bits" 反映了双字“Status1”的存储区域，并且该双字采用了 Array of Bool 形式。
- 使用“Status1Bits”对该双字的各个位进行寻址，例如，使用以下语法：
Status1Bits[0] := True;

该指令将双字中的第 24 位设置为 TRUE。接下来的这部分内容说明了相应位的分配。

加载期间的操作

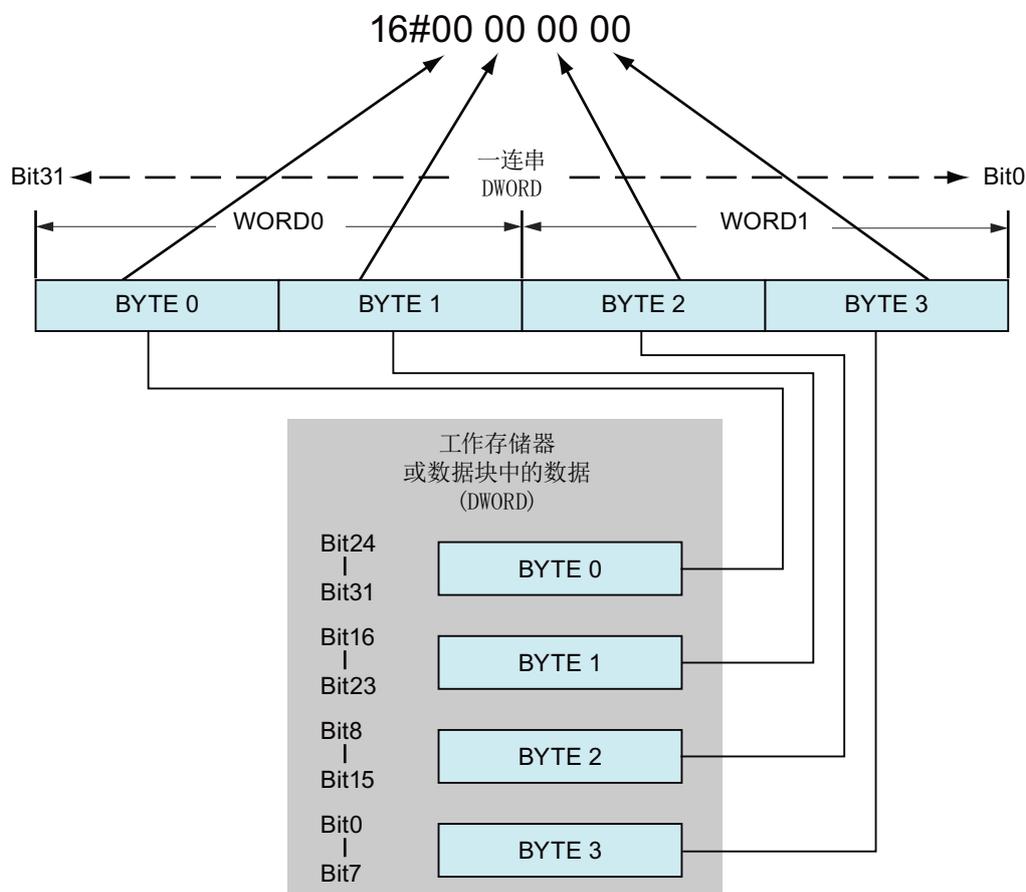
将双字从存储器加载到累加器中时，会按照如下方式交换字节：

- 交换双字中的 High WORD 和 Low WORD。
- 交换字中的 High Byte 和 Low Byte。

累加器中的结果：

- DB 中最高位地址区的字节是双字的第 0 位到第 7 位。
- DB 中最低位地址区的字节是双字的第 24 位到第 31 位。

“AT”命令总是使用数据块或存储器地址区按升序来指定其视图。



建议使用自定义数据类型 (UDT) 来处理双字中的各个位。

以下示例显示了自定义数据类型的结构：

```
// 示例脚本： 用于在双字中分配各个位的数据类型
TYPE SBit32inDWORD
STRUCT
    // highest Byte in accu                BYTE0 in memory
    X24 : BOOL;                          // 0.0
    X25 : BOOL;
    X26 : BOOL;
    X27 : BOOL;
    X28 : BOOL;
    X29 : BOOL;
    X30 : BOOL;
    X31 : BOOL;                          // 0.7
    // second highest Byte in accu        BYTE1 in memory
    X16 : BOOL;                          // 1.0
    X17 : BOOL;
```

6.2 修改 AS 块

```

// 示例脚本： 用于在双字中分配各个位的数据类型
X18 : BOOL;
X19 : BOOL;
X20 : BOOL;
X21 : BOOL;
X22 : BOOL;
X23 : BOOL; // 1.7
// third highest Byte in accu BYTE2 in memory
X8 : BOOL; // 2.0
X9 : BOOL;
X10 : BOOL;
X11 : BOOL;
X12 : BOOL;
X13 : BOOL;
X14 : BOOL;
X15 : BOOL; // 2.7
// lowest Byte in accu BYTE0 in memory
X0 : BOOL; // 3.0
X1 : BOOL;
X2 : BOOL;
X3 : BOOL;
X4 : BOOL;
X5 : BOOL;
X6 : BOOL;
X7 : BOOL; // 3.7

END_STRUCT
END_TYPE

```

寻址双字中的位

以下示例是对双字中的第 0 位进行寻址：

```

// 示例脚本： 寻址双字中的位
VAR_OUTPUT
    Status1 : DWORD;
    Status1Bits AT Status1 : SBits32inDWORD;
END_VAR
Begin
    Status1Bits.X0 := True;

```

6.2.4 备注视图

相应 AS 块的“AUTHOR”块属性必须以字符“AdvLib”开始。在编译 OS 时，将为备注视图创建 OS 内部变量。

备注视图需要的变量为：

- "xxx.#TextPermanent" 变量
- "xxx.#StatusPermanent" 变量

有关备注视图的更多信息，请参见高级过程库的在线帮助。

6.2.5 显示最差信号状态

FC 369 函数（符号名称 SelST16）确定 APL 块中的最差信号状态。该状态将在以下位置显示：

- 块图标中
- 面板总览中

步骤

在 AS 块中，将变量 ST_Worst 创建为输出参数，如示例脚本“ST_Worst”中所示。

可以接受该面板总览和块图标中的对象，而无须更改。

```
// 示例脚本“ST_Worst”： 创建变量
ST_Worst      {S7_visible:'false';s7_m_c:='true'}
               :BYTE := 16#80      ; // Worst Signal Status
```

函数的元素

下表给出了 SelST16 函数的输入参数：

参数	说明
InST	具有 16 个 BYTE 类型元素（b0 到 b15）的结构 此结构中最多可以应用 16 个信号状态。
Num	已使用 InST 元素的数量（整型）
SelPrio	优先级方案，最佳信号状态或最差信号状态 (Integer) 有关信息，请参见 APL 的在线帮助。

函数的返回值

6.2 修改 AS 块

参数	说明
RetVal	输入结构 InST 的最佳或最差 Signal Status 的输出

脚本

// 示例脚本： 声明临时变量

```
TempInST: STRUCT
b0 : BYTE;
b1 : BYTE;
b2 : BYTE;
b3 : BYTE;
b4 : BYTE;
b5 : BYTE;
b6 : BYTE;
b7 : BYTE;
b8 : BYTE;
b9 : BYTE;
b10 : BYTE;
b11 : BYTE;
b12 : BYTE;
b13 : BYTE;
b14 : BYTE;
b15 : BYTE;
END_STRUCT;
ST_Worst BYTE;
```

// 示例脚本： 函数的程序调用

```
TempInST.b0 := X1;
TempInST.b1 := X2;
TempInST.b2 := X3;
ST_Worst := SelSt16(InST := TempInST, Num := 3, SelPrio := 0);
```

6.2.6 显示测量单位

APL 自身具有 INTEGER 类型的参数可用于显示测量单位。

步骤

为想要显示的每个测量单位创建一个参数。

- 如果组态了符合标准 IEC 61158 的参数值，则将显示相应单位。APL 的在线帮助中提供了相应的表格。
- 如果值为 0，则显示在“s7_unit”属性中输入的文本。

```
// 示例脚本： 为测量单位创建参数
PV_Unit {
    S7_m_c:='true';
    S7_unit:='1/sec'
} : INT; //Unit
```

6.2.7 参数 OS_Perm/OS1Perm

使用输入参数 "OS_Perm" 或 "OS1Perm" 可以通过参数分配来禁用或启用特定实例的各项操作。

说明

如果 "OS_Perm" 不足，可以使用 APL 块中的参数 "OS1Perm"。

创建参数

对于 OS 可操作的块，建议将 "OS_Perm" 或 "OS1Perm" 参数创建为结构，如示例脚本 "OS_Perm" 中所示。

```
// 示例脚本 OS_Perm: 参数 "OS_Perm" 作为结构
OS_Perm {S7_visible:='false'}
:STRUCT
    Bit0: BOOL:=1; //1 = Operator can enable
                    accumulation
    Bit1: BOOL:=1; //1 = Operator can switch off
                    accumulation
    "
    "
    "
    Bit31: BOOL:=0; // Not used
END_STRUCT; // Operator permissions
ArrOS_Perm AT OS_Perm :
ARRAY[0..3] of BYTE;
```

6.2 修改 AS 块

编译 OS

必须将 "OS_PermOut" 和 "OS_PermLog" 参数创建为输出。在编译 OS 时，会将这些参数创建为变量：

- "OS_PermOut" 变量（输出双字）

"OS_PermOut" 输出指示通常情况下，是否在块中启用某个操作。

```
// 示例脚本: "OS_PermOut" 变量
OS_PermOut {S7_visible:='false';S7_m_c:='true'}
             :DWORD:=16#FFFFFFFF; //Parameterized Permissions
ArrOS_PermOut AT OS_PermOut : ARRAY[0..3] of BYTE;
```

- "OS_PermLog" 变量（输出双字）

"OS_PermLog" 输出指示通常情况下，是否在块中启用某个操作并且有条件地执行该操作。

在面板中，"OS_PermLog" 将用于对已使用的操作对象启用操作员控制。

```
// 示例脚本: "OS_PermLog" 变量
OS_PermLog {S7_visible:='false';S7_m_c:='true'}
            :DWORD:=16#FFFFFFFF; //Permissions with Process conditions
ArrOS_PermLog AT OS_PermLog : ARRAY[0..3] of BYTE;
```

结构的视图

"AT"命令提供了结构的视图：

结构	视图	形式	示例
OS_Perm	ArrOS_Perm	ARRAYof BYTE	请参见示例脚本 OS_Perm
OS_PermOut	ArrOS_PermOut	ARRAYof BYTE	请参见示例脚本 OS_PermOut
OS_PermLog	ArrOS_PermLog	ARRAYof BYTE	请参见示例脚本 OS_PermLog

使用这些视图可将 "OS_Perm" 输入结构传送到输出双字中。

```
// 示例脚本 "OS_PermLog":
// 将输入结构传送到输出双字中
//*****
// Copy Data from Structure OS_Perm to
//   DWORD OS_PermOut by means of AT command
//*****
ArrOS_PermOut[3]:=ArrOS_Perm[0];
ArrOS_PermOut[2]:=ArrOS_Perm[1];
```

```
// 示例脚本 "OS_PermLog":
// 将输入结构传送到输出双字中
ArrOS_PermOut [1] :=ArrOS_Perm[2];
ArrOS_PermOut [0] :=ArrOS_Perm[3];
```

说明

如果要使用 SCL 命令“AT”以位、字节或字的形式监视一个双字，请注意以下事项：

- 交换字中的 High Byte 和 Low Byte。
- 交换双字中的 High WORD 和 Low WORD。

要确保正确存储输出双字中结构的位，必须按示例脚本“OS_PermLog”中所示来分配这些位。

创建过程相关的使能端

“OS_PermLog”输出需要过程相关的使能端。

我们建议用户按照示例脚本 “OS_Perm3” 中所示来创建临时变量及这些变量的视图。

```
// 示例脚本 "OS_Perm3": OS_PermLog 的临时变量
// temporary Parameters for OS_PermLog
tmpDW_OS_Perm:DWORD;
arrtmpDW_OS_Perm AT tmpDW_OS_Perm: ARRAY[0..3] of BYTE;
sttmpDW_OS_Perm AT tmpDW_OS_Perm: STRUCT;
                                Bit0: BOOL;
                                Bit1: BOOL;
                                "
                                "
                                "
                                Bit31: BOOL;
                                END_STRUCT;
```

使用过程相关的使能端创建参数化使能端

示例脚本 “OS_Perm42” 说明了如何组合参数化使能端与过程相关的使能端。

通过使用“AT”命令，可将相应的位直接写入到临时双字“tmpDW_OS_Perm”中。在临时双字中交换了相应字节后，将对“OS_PermLog”输出双字执行写操作。

```
// 示例脚本 "OS_Perm42": 连接参数化使能端与过程相关的使能端
//*****
// OS_Perm and process related permissions
//*****
sttmpDW_OS_Perm.Bit0:=OS_Perm.Bit0 AND NOT HiAlmAct;
```

6.2 修改 AS 块

```
// 示例脚本 "OS_Perm42": 连接参数化使能端与过程相关的使能端
                                                                    //permission enable accumulation
sttmpDW_OS_Perm.Bit1:=OS_Perm.Bit1;    //no further process condition
    "
    "
    "
sttmpDW_OS_Perm.Bit31:=OS_Perm.Bit31;    //no further process condition

ArrOS_PermLog[3] := arrtmpDW_OS_Perm[0];
ArrOS_PermLog[2] := arrtmpDW_OS_Perm[1];
ArrOS_PermLog[1] := arrtmpDW_OS_Perm[2];
ArrOS_PermLog[0] := arrtmpDW_OS_Perm[3];
```

6.2.8 Feature 参数

使用“Feature”参数来设计块的行为方式。

步骤

创建以下结构:

```
// 示例脚本: 参数 "Feature"
Feature S7_visible:='false';
S7_xedit:='Bit0,para;Bit1,para;Bit2,para;Bit3,para;'}
:STRUCT
    Bit0: BOOL:=0; // common Featurebit not used in this block
    Bit1: BOOL:=0; // common Featurebit not used in this block
    Bit2: BOOL:=0; // common Featurebit not used in this block
    "
    "
    Bit5: BOOL:=0;
        // 1 = Alarm Output activ with 0,
        // individual block specific Feature Bit
    "
    "
    Bit31: BOOL:=0; // Not used
END_STRUCT; // Status of various features
```

说明

在程序部分对不同行为方式进行编程的位置, 需要考虑 "Feature" 参数的位。

6.2.9 模拟值结构

在面板中可显示模拟辅助值。为此，需要将“AnalogValue”类型的输入参数用作结构。可见性取决于输入参数是否已经互连。

显示模拟辅助值

必须判断相关 AS 块的 Signal Status (是否为 **16#FF**)，并将状态字中的此信息传送到 OS 中。

Signal Status 用于检测输入参数是否已经互连：

- **Signal Status = 16#FF**

输入参数没有互连且不会显示在面板中。

Signal Status 16#FF 是内部 APL 状态。

- **Signal Status 不等于 16#FF**

输入参数已经互连且将显示在面板中。

将“AnalogValue”类型的输入参数创建为结构

按以下示例脚本中所示创建输入参数。

在此示例脚本中，将通过状态字 1 的第 5 位来传送输入参数是否已经互连的信息。

说明

Signal Status 的预定义值必须为 **16#FF**。

```
// 示例脚本： 创建参数
VAR_INPUT
  Useranal  {S7_visible      := 'false';
             S7_m_c         := 'true';
             S7_xm_c        := 'Value,true;';
             S7_xqc         := 'Value,true;';
             S7_xshortcut   := 'Value,Auxiliary value1;'}
             : STRUCT
             Value          : REAL := 0.0;      // Value
             ST             : BYTE := 16#FF;    // Signal Status
END_STRUCT;                                     // User Analog Input 1

UA1unit    {S7_visible      := 'false';
             S7_m_c         := 'true';
             S7_unit        := ''}
```

6.2 修改 AS 块

```
// 示例脚本: 创建参数
                                :INT:=0;                                // Unit of UserAna1
END_VAR

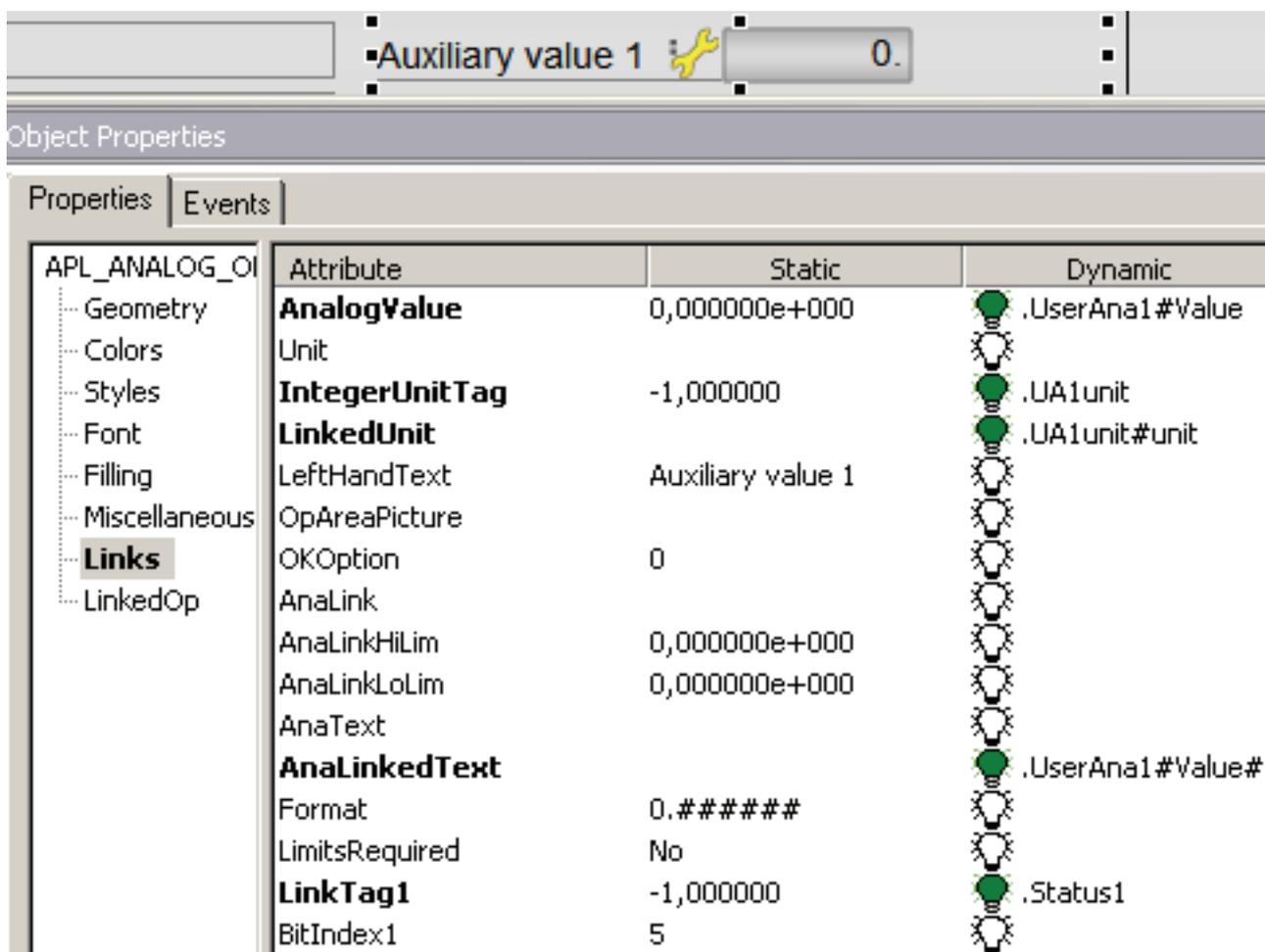
BEGIN
    dwOutStatus1.X5:=NOT (UserAna1.ST=16#FF);
```

复制对象

要在面板中显示辅助模拟值，请将以下对象从模板复制到面板视图中：

APL_ANALOG_OP_DISPLAY (@PCS7ElementsAPL.pdl; Display analog Value;
Setpoint)

- 将 AnalogValue 属性设置为 .UserAna1#Value。
- 按示例画面中所示，修改 "Links" 属性中的以下特性 (attribute)。
 - 删除示例画面中未显示的特性连接。
 - 将状态字设置为 "LinkTag1"，并在 "BitIndex1" 中组态适当的位。
 - 链接到 "LinkTag1" 属性的脚本通过“如改变”(If changed) 事件控制显示的可见性。
- 将“颜色”(Color) 属性中的 "LineColor" 特性设置为黑色以显示边框。



The screenshot shows the 'Object Properties' dialog for 'Auxiliary value 1'. The 'Links' property is selected, and the 'Dynamic' column shows the following values:

Attribute	Static	Dynamic
AnalogValue	0,000000e+000	.UserAna1#Value
Unit		.UA1unit
IntegerUnitTag	-1,000000	.UA1unit
LinkedUnit		.UA1unit#unit
LeftHandText	Auxiliary value 1	
OpAreaPicture		
OKOption	0	
AnaLink		
AnaLinkHiLim	0,000000e+000	
AnaLinkLoLim	0,000000e+000	
AnaText		
AnaLinkedText		.UserAna1#Value#
Format	0.#####	
LimitsRequired	No	
LinkTag1	-1,000000	.Status1
BitIndex1	5	

6.2.10 连接 SIMATIC BATCH

为符合 APL 的 SIMATIC BATCH 接口声明以下输入参数。

```
//SIMATIC BATCH 接口
```

```
BatchEn {S7_visible := 'false'} // 通过批生产配方启用控制器的远程操作
```

6.2 修改 AS 块

```
//SIMATIC BATCH 接口
:BOOL := false;
BatchID {S7_visible := 'false'; S7_m_c := 'true'} // 当前批次 ID (号)
:DWORD := 16#00;
BatchName {S7_visible := 'false'; S7_m_c := 'true'} // 当前批生产名称
:STRING [32];
StepNo {S7_visible := 'false'; S7_m_c := 'true'} // 批生产步号
:DWORD := 16#00;
Occupied {S7_visible := 'false'} // 被批生产占用
:BOOL := false;
```

6.2.11 仿真

以下选项可用于 APL 块中的仿真。

- 块外部仿真
- 块内部仿真

有关更多信息，请参见 APL 在线帮助。

在符合 APL 的块中进行块内部仿真时，需要考虑以下信息。

数字过程值仿真

该块必须有一个用于数字过程值仿真的“SimOn”输入参数。通过面板来操作该输入参数。

当具有数字过程值输入的块（例如，驱动器块）处于仿真状态时，将根据控制条件来仿真可能的反馈。

这就是总是要为反馈输入声明相应反馈输出的原因。

- 当仿真处于激活状态时，将通过控制条件得到这些反馈输出。
- 当仿真未处于激活状态时，这些反馈输出将被直接传送到反馈输入。

电机示例：

- 反馈输入“FbkRun”
- 反馈输出“FbkRunOut”

模拟过程值仿真

当具有模拟过程值输入的块处于仿真状态时，可以从面板中更改该过程值。使用以下示例脚本来仿真模拟过程值：

```
// 示例脚本： 模拟过程值仿真
VAR_INPUT
  SimOn   {S7_visible := 'false';
           S7_m_c := 'true'}
           :BOOL := FALSE;           //Simulation On/Off
  SimPV   {S7_visible := 'false';
           S7_m_c:= 'true'}
           :REAL :=0.0;           //Simulation Value
END_VAR
VAR_OUTPUT
  PV_Out   {S7_xqc:='Value,true;';
            S7_dynamic:='true';
            S7_m_c:= 'true';
            S7_xm_c:='Value,true;';
            S7_xshortcut:='Value,PV;';
            S7_xarchive:='Value,shortterm;';
            S7_xtrend:='Value,PV_OpScale.Low,PV_OpScale.High;'}
            :AnaVal; //Process Value (Analog Output)
END_VAR

// Code selection - Simulation
IF SimOn THEN
  PV_Out.Value := SimPV;
  PV_Out.ST := 16#60;
ELSE
  PV_Out.Value := PV.Value;
  SimPV := PV_Out.Value;
  PV_Out.ST := PV.ST;
END_IF;

dwOutStatus1.X6:=SimOn;
```

显示和操作

- 复制模板中的以下对象以在面板中显示和操作“启用/禁用仿真”功能。向面板视图添加这些对象。
 - APL_OP_BUTTON (@PCS7ElementsAPL.pdl; Open binary operation area; Normal)
 - APL_MULTI_TEXT2 (@PCS7ElementsAPL.pdl; Status display text; Text display)
 有关更多信息，请参见章节“使用文本框显示二进制值 (页 119)”。
- 从模板中复制以下对象以显示模拟值“PV”。向面板视图添加该对象：
 - APL_ANALOG_OP_DISPLAY (@PCS7ElementsAPL.pdl; 显示模拟值; 设定值)
 有关更多信息，请参见章节“针对 APL_ANALOG_OP_DISPLAY 对象的模拟值操作 (页 103)”。

说明

此处要确保“SimOn”的相应状态字/状态位与操作员控制使能端 (“LinkTag3”属性) 互连，而不是与“OS_PermLog”互连。

6.2.12 “停止服务”功能

使用“停止运行”操作模式禁用块的所有功能。不会生成任何进入消息或离开消息。只允许执行操作模式切换。

有关更多信息，请参见 APL 的在线帮助。

步骤

在块中创建接口和相关程序段，如 OutOfService_1 示例代码中所示。请注意以下与参数有关的信息。

示例脚本“OutOfService_1”

```
VAR_INPUT
  OnOp      {S7_m_c:='true'; S7_link:='false'; S7_visible:='false'}
            :BOOL := FALSE;          // 1=On Mode: On Mode by Operator
  OosOp     {S7_m_c:='true'; S7_link:='false'; S7_visible:='false'}
            :BOOL:=FALSE;          // 1=Oos Mode: Oos Mode by Operator
  OosLi     {S7_dynamic:='true'}
            :DigVal;              // 1=Oos Mode: Oos Mode by Field Signal
  OS_Perm   {S7_visible:='false'}
            :STRUCT
            Bit0: BOOL :=1;       // 1 = Operator can switch to On
```

```

示例脚本“OutOfService_1”
        Bit1: BOOL :=1;      // 1 = Operator can switch to OOS
        END_Struct;
END_VAR
VAR_OUTPUT
    OosAct  {S7_dynamic:='true';
            S7_m_c:='true';
            S7_xm_c:='Value,true;'}
            :DigVal;          // Out of service is active
    OnAct   {S7_dynamic:='true'}
            :STRUCT
            Value : BOOL := true;  // Value
            ST    : BYTE := 16#80; // Signal Status
            END_STRUCT;          // On Mode is active
END_VAR
VAR
    SxOosLi :BOOL := FALSE;
END_VAR

// CODE Selection
IF OnOp THEN
    OosAct.Value := false;
    OnOp := false;
ELSIF OosOp OR (OosLi.Value AND Feature.Bit1 AND NOT SxOosLi) THEN
    OosAct.Value := true;
END_IF;
OosOp := false;
SxOosLi := OosLi.Value;
OnAct.Value := NOT OosAct.Value;
IF OosAct.Value THEN // OutOfService conditions
    OS_PermLog := (16#1 AND OS_PermOut);
ELSE
    // Not in mode "Out of service"
END_IF

```

用于选择操作模式的参数

使用以下参数来选择块的操作模式：

- 参数“OnOp”可切换到“运行”(On) 操作模式
- 参数“OosOp”可切换到“停止运行”(Out of operation) 操作模式

说明

具有手动/自动操作模式切换功能的块

只允许在手动模式下切换到“停止运行”。

所需操作

6.2 修改 AS 块

1. 在 AS 块的状态字中为“停止运行”状态分配一个位。
在块图标中将显示维护状态。
复制 APL 块图标时，可能需要进行以下调整：
 - 状态字
 - 直接出现在扩展状态显示中的位
2. 在面板中显示维护状态。
必要时，可修改画面 @PG_MyAPLFP_Overview 中维护状态“停止运行”(Out of operation) 的状态显示：
 - 状态字
 - 为总览面板的扩展状态显示组态适当的位。

说明

示例脚本 OutOfService_1 说明了块的行为方式。

- 值在传送后会被冻结。
 - 除非切换回“运行”(On) 操作模式，否则将禁用所有操作。
-

有关更多信息，请参见章节“画面 @PG_MyAPLFP_Overview.pdl 中的通用规范 (页 90)”。

显示和操作

遵循以下段落中的描述，在面板中显示和操作“运行”/“停止运行”(“On” / “Out of operation”) 操作模式：

- 二进制值操作
“二进制值操作 APL_OP_BUTTON (页 110)”部分
- 多文本
“使用文本框显示二进制值 (页 119)”部分

在“停止运行”(Out of operation) 模式中，以下显示被设置为不可见：

- 块图标的组显示
其余显示以灰色背景出现。
在块图标中，该显示通过 "OosAct" 输出参数进行控制。
- 总览面板的组显示
当启用“停止服务”(Out of service) 时，一个空的组显示会叠加在状态字中有一个位的组显示之上。
为此，可以使用状态字中的第 3 位，或者将组显示对象的动态对话框修改成第 3 位。

6.2.13 跳转到其它面板

为跳转到其它面板，可在 APL 面板中创建跳转按钮。

步骤

复制跳转按钮并创建参数 "SelfP1" 和 "SelfP2" 。

跳转按钮：

- 跳转按钮 1 位于标准视图对象名称“csoFpBtnUser1”（对象类型 "APL_FACEPLATE"）中。
- 跳转按钮 2 位于预览对象名称“csoFpBtnUser2”（对象类型 "APL_FACEPLATE"）中。

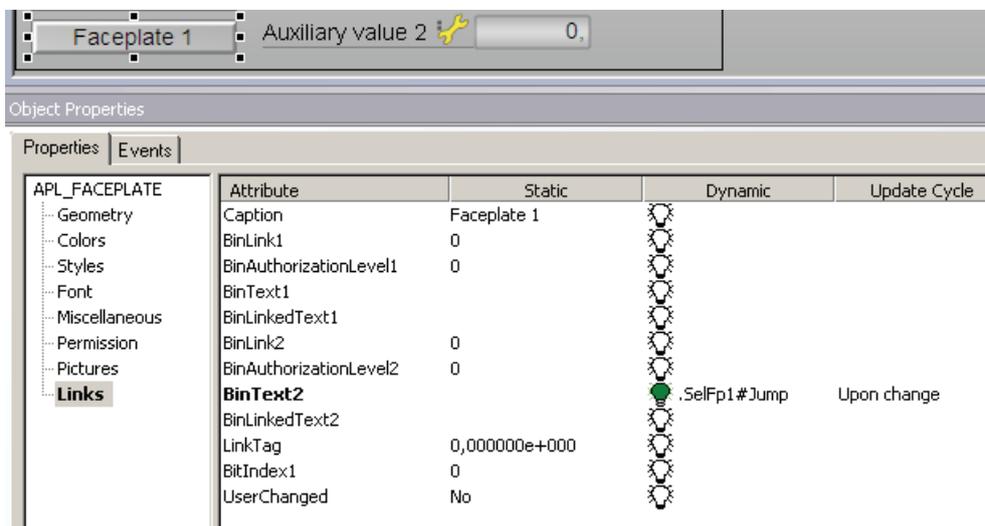


图 6-3 SelfP1_1

参数：

- 这些参数的类型为 ANY，从而可与另一个块的任何参数互连。
- 这些参数具有 CFC 属性 BLK_Jump。

编译 OS

- 如果按示例脚本“SelfP1_2”中所示对 BLK_Jump 属性进行设置，则在编译 OS 期间将为实例创建一个 OS 内部变量“xxx.SelfP1#Jump”。
- 如果参数已互连，则将创建包含互连块的 OS-tagname 条目的文本引用。

对该文本引用的引用包含在“xxx.SelfP1#Jump”变量中。

6.2 修改 AS 块

示例脚本： 引用文本引用的变量

```
SelFp1      {BLK_Jump:='1';S7_visible:='false'}  
            :ANY;                               //Select Faceplate1  
SelFp2      {BLK_Jump:='1';S7_visible:='false'}  
            :ANY;                               //Select Faceplate2
```

SelFp1_2

OS 块符号

无法将如 S7_string 或 S7_shortcut 之类的属性分配给 Anypointer。实例特定的按钮标签存储在块图标中：

属性 -> UserButtonText1、UserButtonText2

为 CFC 块中跳转按钮的标签指定文本

选项 1:

示例脚本： 创建 BOOL 类型的参数

```
SelFp1_3    {BLK_Jump:='1';  
            S7_m_c:='true';  
            S7_string_1:='Jump to xxx'  
            }  
            : BOOL; //Select Faceplate 3
```

SelFp1_3

1. 例如，创建 BOOL 类型的参数（而不是 Anypointer 类型），然后将跳转按钮标签的文本分配给 S7_string1 属性。

说明

创建了 BOOL 类型的参数后，将能够执行以下操作。

- 该参数只能与 BOOL 类型的参数互连。
 - 自 PCS7 V7.1 SP1 开始：
可将该参数与 BOOL 类型的 APL 结构互连。
-

选项 2:

示例脚本： 具有文本附加参数的 Anypointer

```
SelFp4      {BLK_Jump:='1';S7_visible:='false'} :ANY; //Select Faceplate 4  
SelFp4Tx    {S7_m_c:='true';
```

示例脚本： 具有文本附加参数的 **Anypointer**

```
S7_string_1:='Jump to xxx'
}
: BOOL; //Select Faceplate 4 text
```

1. 保持 **Anypointer** 不变并将跳转按钮标签的文本分配给附加参数，例如“**Selfp3Tx**”，如示例脚本中所示。
2. 按如下所示修改用户对象“**csoFpBtnUser1**”：

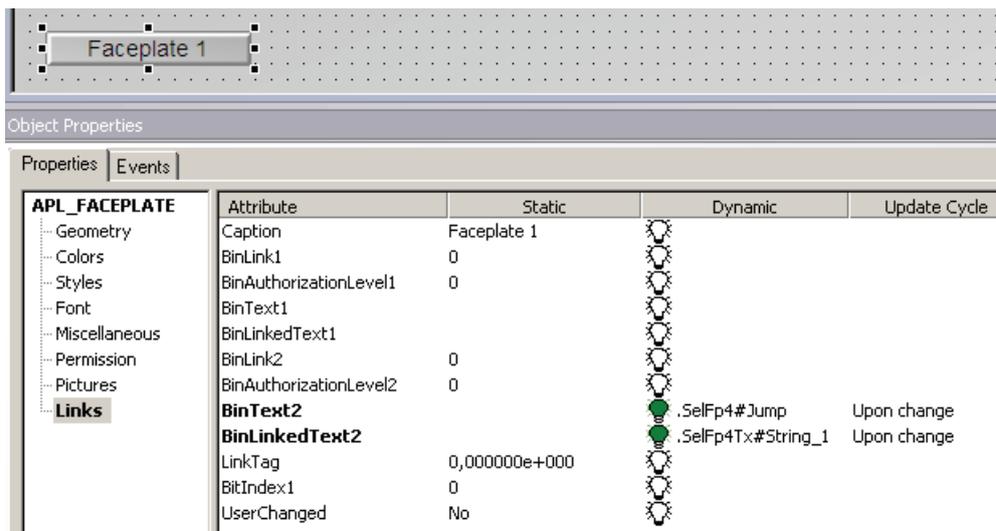


图 6-4 Selfp1_5

3. 更改脚本中连接到属性“**BinText2**”的以下行：

原始脚本

```
szCaption = GetPropChar(pszParent, "BlockiconCollection",
"UserButtonText1");
```

更改后的行：

更改后的脚本

```
pszCaption = GetPropChar(lpszPictureName, lpszObjectName,
"BinLinkedText2");
```

6.2.14

维护发布

维护发布用来指示需要执行维护、维修或校准的过程标签。

有关更多信息，请参见 **APL 在线帮助**。

需要 AS 块中的 "MS_RelOp" 输入参数。在“手动”操作模式下，块必须将 "MS_RelOp" 输入参数写入到 "MS_Release" 输出中。

6.2 修改 AS 块

示例脚本: OutOfService_1

```
VAR_INPUT
    MS_RelOp {S7_m_c := 'true'} :BOOL := FALSE;
END_VAR
VAR_OUTPUT
    MS_Release :DigVal;
END_VAR
Begin
    MS_Release.Value := MS_RelOp AND ManAct.Value;
    IF NOT ManAct.Value THEN
        MS_RelOp := false;
    END_IF;
Status1Bits.x31 := MS_RelOp
```

说明

请进行以下设置和互连:

- 为 "OS_Perm" 参数中相应位的操作创建操作使能端。
 - 将来自 "MS_Release" 的信息链接到相应的状态字。
-

显示和操作

- 复制模板中的以下对象以在面板中显示和操作“启用/禁用维护发布”功能。向面板视图中添加这些对象。
 - APL_OP_BUTTON (@PCS7ElementsAPL.pdl; Open binary operation area; Normal)
 - APL_MULTI_TEXT2 (@PCS7ElementsAPL.pdl; Status display text; Text display)
- 从模板中复制以下对象以显示模拟值“PV”。向面板视图中添加该对象：
 - APL_ANALOG_OP_DISPLAY (@PCS7ElementsAPL.pdl; Display analog Value; Setpoint)

有关更多信息，请参见章节“使用文本框显示二进制值 (页 119)”。

有关更多信息，请参见章节“针对 APL_ANALOG_OP_DISPLAY 对象的模拟值操作 (页 103)”。

说明

此处要确保“MS_RelOp”的相应状态字/状态位与操作员控制使能端 (“LinkTag3”属性) 互连，而不是与“OS_PermLog”互连。

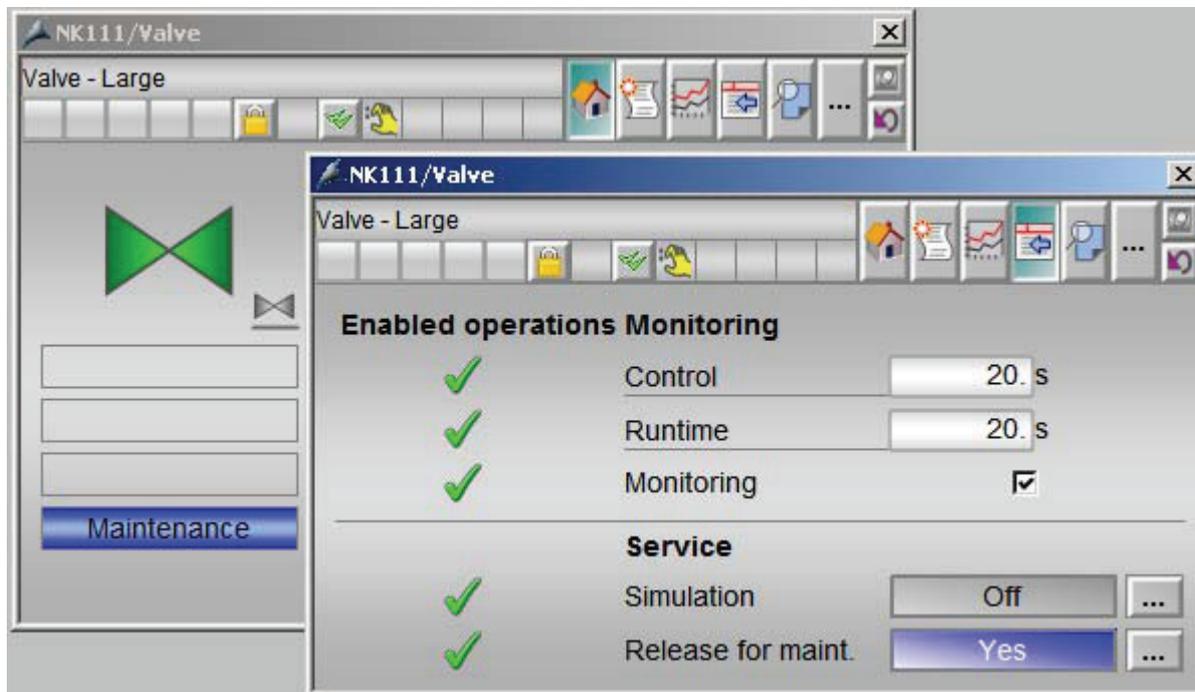


图 6-5 图形 MS_Release1

6.3 创建块图标

6.3 创建块图标

6.3.1 修改块图标（总览）

创建块图标

功能	部分
创建块图标	<ul style="list-style-type: none">• 准备工作 (页 82)• 组态操作对话框<ul style="list-style-type: none">– 直接调用操作对话框 (页 84)– 调用操作对话框前检查操作员权限 (页 84)

创建符合 APL 的块图标所用的基本过程与创建标准库的块图标所用的过程相同。

步骤

1. 将符合要求的块图标从 "@PCS7TypicalsAPLV8.PDL" 模板画面复制到您自己的画面中（例如 @PCS7TypicalsMyLib.pdl）。
2. 将参数分配给属性 "type" 和 "Servername"，以便能引用 AS 块和想要打开的面板。
3. 从快捷菜单中选择菜单命令“用户对象 > 编辑”(User object > Edit)。可以
向块图标中添加新对象或者移动现有对象等等。
4. 在组态对话框中将“speaking”指定为对象名称，方便识别各个对象的用途。

说明

当块图标的大小变化时，必须修改内部对象 "PolylineIcon" 的大小，因为它表示 "HighLightBlockIcon" 图标的边框。

5. 选择属于块图标中的所有对象并终止“编辑”(Edit) 模式。
6. 在组态对话框中，将想要动态化的新对象的属性应用到块图标的外部接口。
 - 如果多个状态显示访问同一个状态字，则将它们组合成块图标的的一个属性。
7. 将“speaking”指定为块图标的新属性名称。

说明

始终对属性的显示名称和名称使用相同的文本。

8. 退出组态对话框。
9. 将动态属性与新属性相连。如有必要，可修改副本模板中任何现有的动态属性。

参见

块图标模板画面 (页 15)

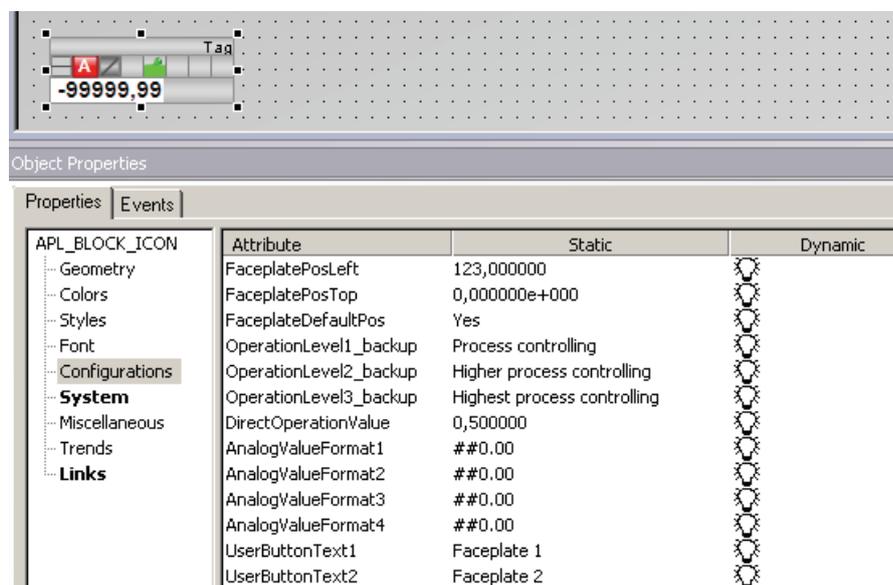
块图标结构 (页 16)

6.3.2 常规属性的描述

6.3.2.1 “Configurations”属性组

可在“Configurations”属性组中组态实例特定的设置。

“Configurations”属性组



属性	含义
FaceplatesDefaultPos	"FaceplatesDefaultPos" = No 可用来指定想要打开的面板的位置。
OperationLevel1 ... OperationLevel3	"OperationLevel1 ... 3" 用于设置实例特定的操作员权限级别。 可在三种操作员权限级别之间选择： 各种操作与这三个级别的分配关系将存储在面板中。

6.3 创建块图标

属性	含义
DirectOperationValue	DirectOperationValue 是块图标的属性。使用 DirectOperationValue 可为直接操作指定两个内部按钮的百分比值。使用 DirectOperationValue 乘以系数 5 可以自动计算出两个外部按钮。
AnalogValueFormat1 ... AnalogValueFormat4	AnalogValueFormat1 ... 4 用于设置块图标和面板的数字格式。
UserButtonText1 UserButtonText2	对于用来打开其它面板的按钮（跳转按钮），可通过 UserButtonText1..2 为其实例特定文本分配参数。

6.3.2.2 “System”属性组

“System”属性组包含面板的基本数据。

“System” 属性组

Attribute	Static	Dynamic	Update C...
TagNameDisplayed	Tag		
Type	@MonAnL/1		
TagName	Tagname		
Servername	PCS7 MonAnL Control		
SignificantMask	0		
EventQuitMask	3089	@EventQuit	Upon change
Relevant	Yes		
TagVisible	No		
TypeID	1023,000000		
VersionID	2,011060e+008		
HighlightBlockIcon	No		
OosNotActive	Yes		
ToolTipText		.#comment	Upon change
TagVisibleLink	No	@local::@APLShowTag	Upon change

属性	含义
TagNameDisplayed	"TagNameDisplayed" 是显示在块图标中的变量名。在工厂层级中，可以指定在面板中是显示完整 OS 变量名还是仅显示部分变量名。
Type	"Type" 是"@PCS7TypicalsAPLV8.PDL"模板画面中使用的引用，它引用该块图标中版本 1 的 "MonAnL" 块类型。 输入项目特定 FB 的名称，例如"MyAPL_BL"。
Tagname	"Tagname" 包含发送到面板中的实际 OS 变量名。

属性	含义
Servername	"Servername" 包含相应的面板类型。输入项目特定面板类型的名称，例如""PCS7 MyAPLFP Control""。
TypeID	TypeID 一直到 1999（包含在内）都预留给 APL 使用。

有关这些属性的更多信息，请参见《PCS 7 块编程指南》(PCS 7 Programming Instructions for Blocks)。

块图标上可用组显示的属性：

- 向外引导的组显示的新属性：
 - "EventQuitMask" 必须与"@EventQuit"变量互连。由于可调整 OS 项目编辑器中“容差”消息类别的行为，因此该设置是必需的。
 - "SignificantMask" 不可以更改。

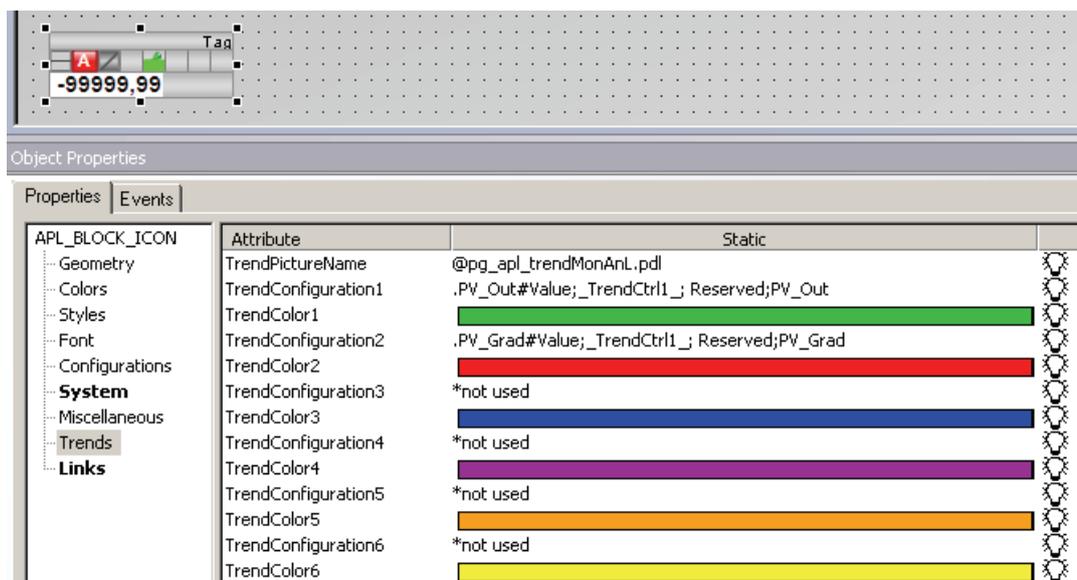
属性	含义
TypeID VersionID	<ul style="list-style-type: none"> • "TypeID" 是分配给 APL 块图标的编号。 建议： 为自定义的图标分配编号。最大到 2000（包含在内）的编号都预留给 APL 使用。 • "VersionID" 用于用户对象的版本控制。 版本以日期形式存储。 "BlockIcon2"画面中显示的"2.011060e+008"可按如下方式进行细分和解释： 双击"VersionID"。输入字段中将显示以下数字：201106012。此数字指示版本发布于 2011 年 6 月 1 日。最后一位数字指示递增值为 2。
ToolTipText	<p>"ToolTipText" 和变量名始终与同一个内部对象互连。这意味着 "ToolTipText" 将始终显示变量名。ToolTipText 可能没有分配参数。</p> <p>有关具体 ToolTipText 的更多信息，请参见“在 V7 块图标的 ToolTipText 中显示块注释 (页 126)”。</p>
TagVisibleLink	<p>以下版本中提供了 APL 的不同块图标：</p> <ul style="list-style-type: none"> • 永久显示 tagname 的版本 • 可在其中隐藏 tagname 的版本（通常是版本 5 和更高版本）。 <p>"@PCS7TypicalsAPLV8.PDL"模板文件具有一个"Show/Hide Tag"按钮。 该按钮作用于本地计算机变量"@local::@APLShowTag"，通过该变量可以集中显示/隐藏变量名。</p>

6.3 创建块图标

6.3.2.3 “Trends”属性组

可以在面板中组态“Trends”属性组的趋势视图。

“Trends” 属性组



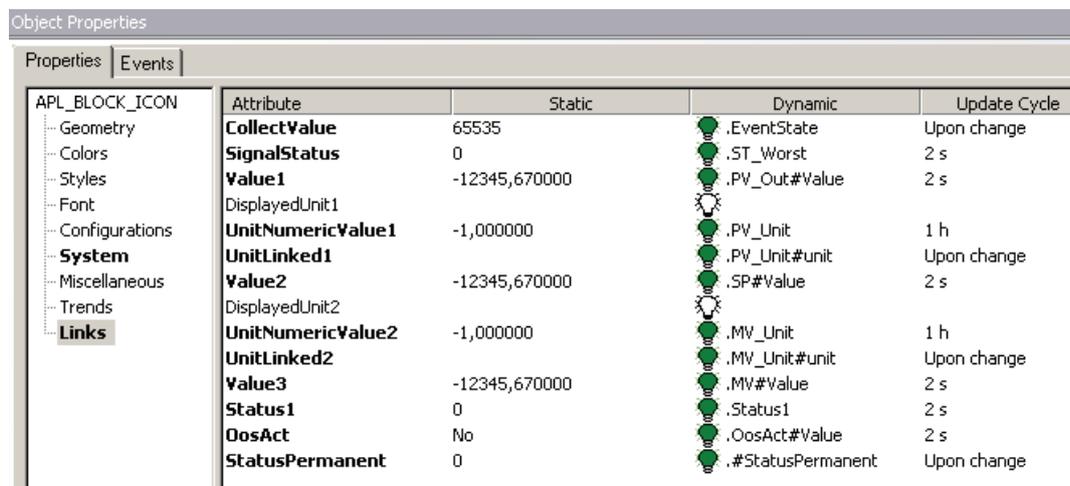
属性	含义
TrendPictureName	<p>使用 "TrendPictureName" 选择趋势视图的模板画面。</p> <p>上图包括“@pg_apl_trendMonAnL.pdl”模板画面，该画面包含两个趋势视图：</p> <ul style="list-style-type: none"> • 测量值的趋势视图 • 梯度显示窗口 <p>使用“@pg_apl_trend.pdl”模板画面可在一个趋势视图中显示最多 12 个趋势。</p>
TrendConfiguration 1	<p>在“TrendConfiguration1 到 12”中可以指定想要显示的值。输入以下信息并以分号进行分隔：</p>
...	<ul style="list-style-type: none"> • 变量 <p>例如， “.PV_Out#Value”</p>
TrendConfiguration 12	<ul style="list-style-type: none"> • 趋势控件 • 预留 • 轴和标尺的标签 <p>始终为“_TrendCtrl1_” 始终为“Reserved” "PV_Out"</p>

属性	含义
	<ul style="list-style-type: none"> 归档变量名 如果因现有趋势归档更改了 PH 并因此更改了测量点的更高层级标识 (HID)，必须始终指定归档变量名称。随后，归档变量名称在趋势归档中保持不变，并且无法再通过 HID 自动生成。在这种情况下，必须将归档变量名称指定为第五个参数。
TrendColor1 ... TrendColor12	使用 TrendColor1 ... 12 指定趋势的颜色。

6.3.2.4 “Links”属性组

可组态“Links”属性组中的动态变量。

“Links”属性组



属性	含义
SignalStatus	显示最差信号状态 在块图标中信号状态的显示通过两个状态显示来实现。
DisplayedUnit1	显示测量单位
UnitNumericValue	"DisplayedUnit1 包含测量单位的标签

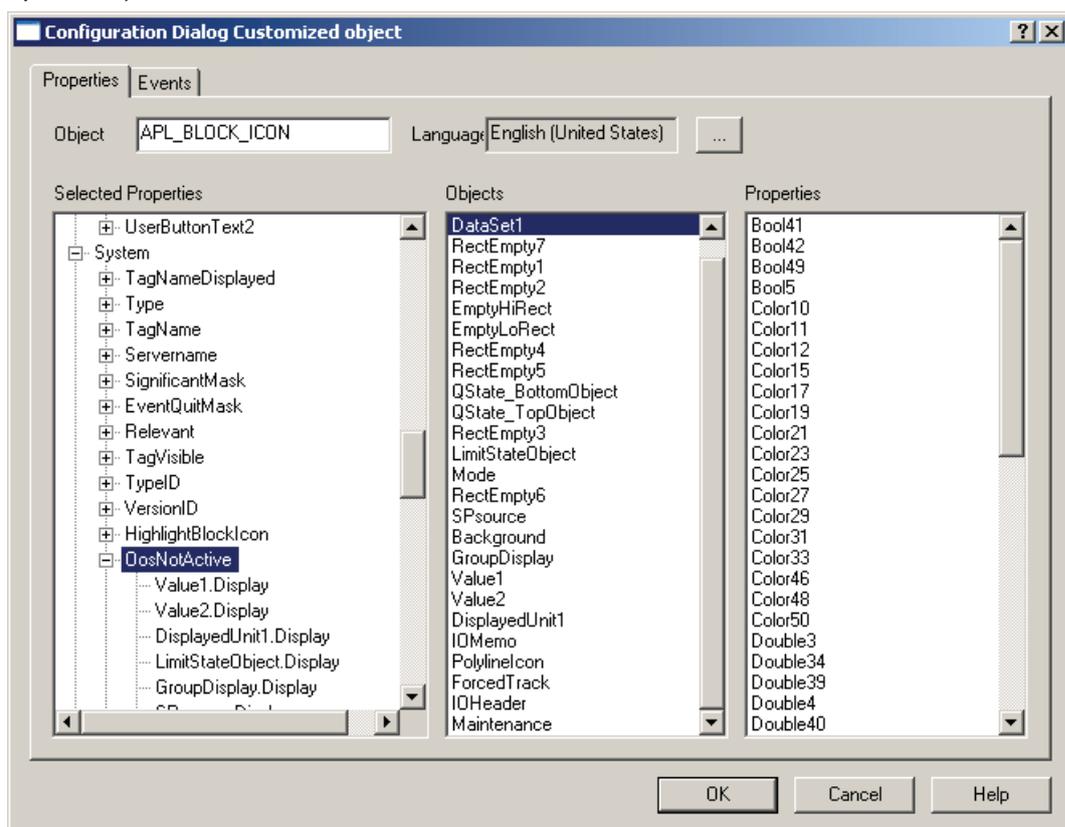
6.3 创建块图标

属性	含义
1 UnitLinked1	<p>"UnitNumericValue1" 用于显示测量单位，而且始终链接到整型变量 xx_Unit 。</p> <p>"UnitNumericValue1" 与一个脚本相连。该脚本具有以下功能：</p> <ul style="list-style-type: none"> 在“UnitNumericValue1 = 0”时 该脚本选择 S7_unit 属性的文本来显示测量单位。 在“UnitNumericValue1 > 0”时 该脚本从表中选择测量单位文本。该表格存储在资源 DLL 中。此资源 DLL 不可更改。 通过相应的整数来选择所需的文本。 <p>"UnitLinked1" 与引用 WinCC 文本词典的 OS 内部变量 ⇒ xx_Unit#unit 互连 在 OS 编译过程中，不支持 S7_xunit 属性。</p>
Status1	<p>显示各种二进制信息。"Status1" 作用于多个内部状态显示。 例如：操作模式或维护状态“进行中”(In progress)。</p>
OosAct	<p>指示块已“停止运行”(Out of operation)。该参数作用于多个内部对象。在 OosAct = "YES" 时，将隐藏这些对象。因此，在安装要显示的新对象时，应考虑这一点。</p>
StatusPermanent	<p>显示是否激活了提示。</p>

将对象切换为在“停止运行”操作模式下不可见

1. 选择块图标。
2. 使用快捷菜单打开组态对话框。
3. 选择“属性”(Properties) 选项卡

在 "@OosNotActive" 属性中列出了块图标的可见对象，在切换到“停止运行”(Out of operation) 操作模式时，这些对象将不可见。



说明

新内部可见对象的“显示”属性

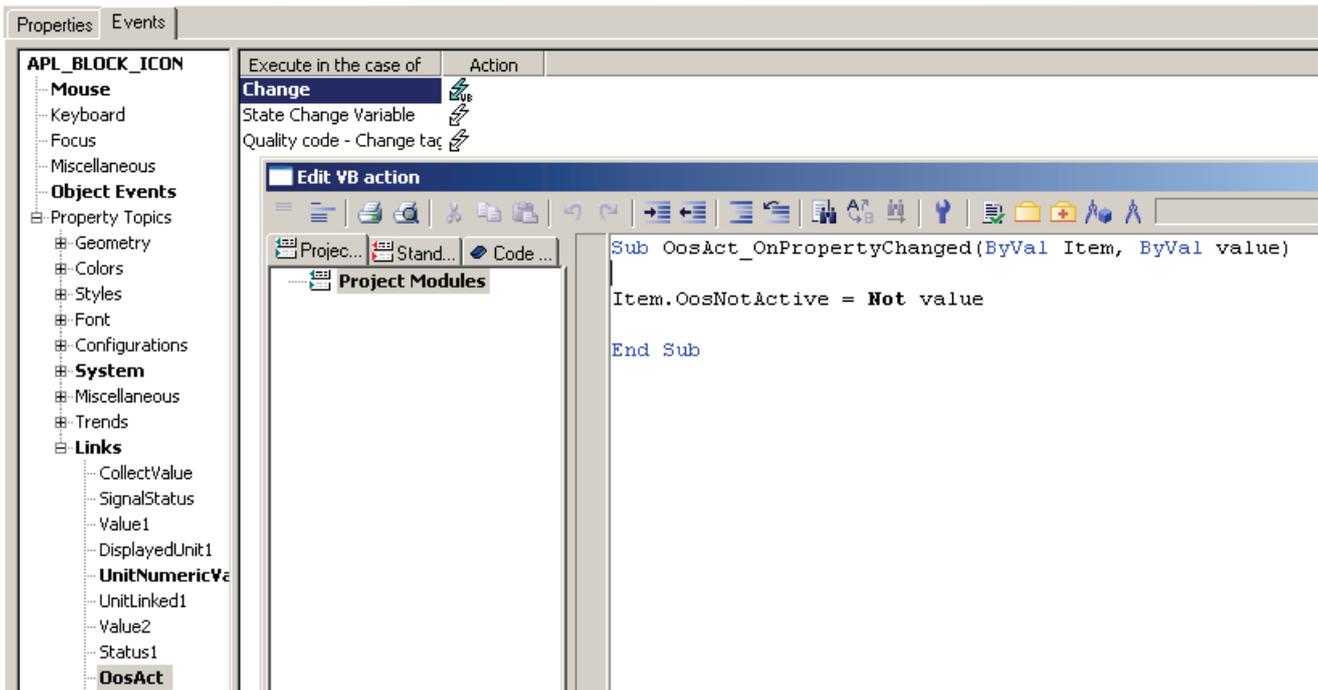
在用户对象组态对话框中，为新内部可见对象增加“显示”（“Visible”）属性。

6.3 创建块图标

“更改”事件的脚本

“更改”事件通过“OosAct”属性连接着一个脚本。此脚本会评估 "@OosNotActive" 属性并控制对象的可见性。

- 如果块不处于“停止运行”操作模式，则 "@OosNotActive" 的值为 "TRUE"。
- 如果块处于“停止运行”操作模式，则 "@OosNotActive" 的值为 "FALSE"。



6.3.3 通过块图标操作

6.3.3.1 准备工作

使用 APL 便可以通过块图标进行操作。

要通过块图标调用操作对话框，可通过脚本使事件“单击鼠标右键”(Release Right) 动态化。

启用事件

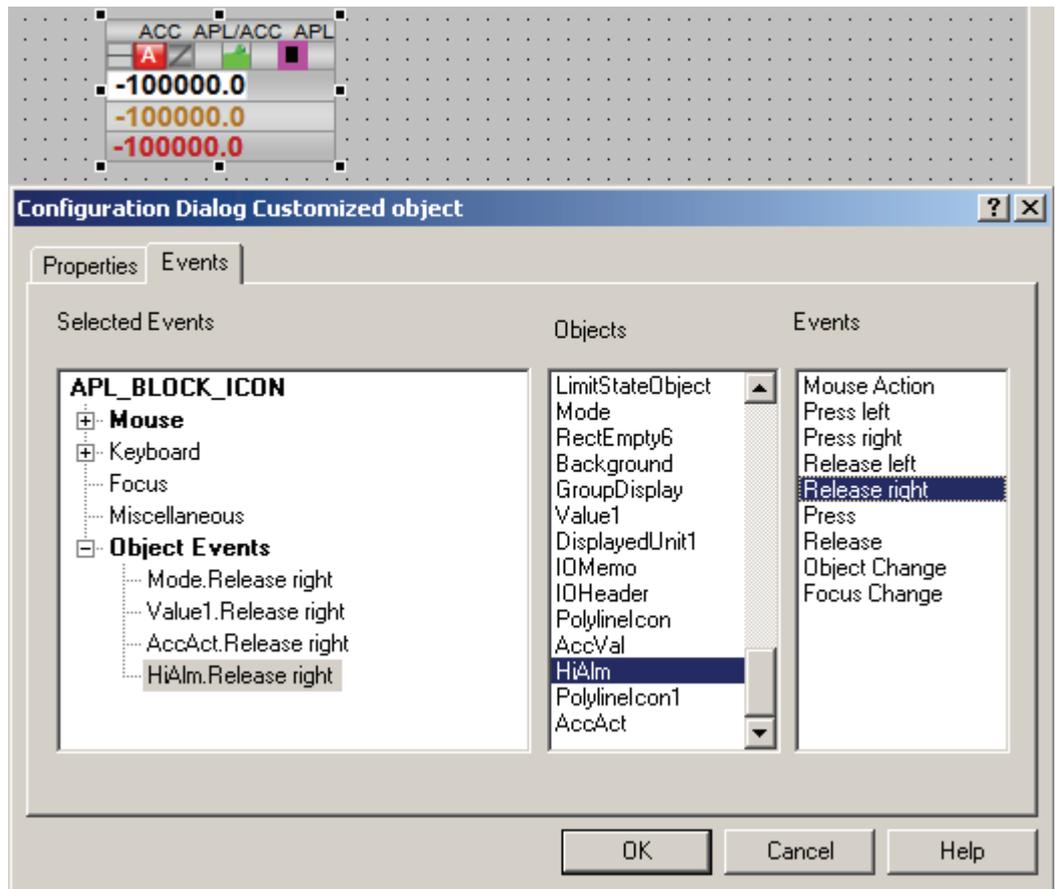
1. 打开块图标的组态对话框。
2. 选择“事件”(Events) 选项卡。

3. 在“对象”(Objects) 列表中选择要操作的对象。

示例：

HiAlm, 更改报警限值

4. 将“Release right”事件拖动到“对象事件”(Object Events) 选项卡中。



5. 退出组态对话框。

使用脚本使事件动态化

使用调用相应操作对话框的脚本，将所提供的事件动态化。以下部分对执行这些脚本的可能结果进行了说明：

- 调用操作对话框但不检查操作员权限
- 调用操作对话框前检查操作员权限

6.3 创建块图标

6.3.3.2 调用操作对话框但不检查操作员权限

建议

使用 APL 标准块图标的操作作为副本模板。

步骤

可按以下示例脚本中所示来调用操作对话框但不检查操作员权限：

- 对全局脚本“APL_OpenFaceplate”使用传送参数“1”来指定操作对话框调用。
- 在传送参数“MFPIC”中，指定要操作的对象所在的视图。
- 通过传送参数“MFOBJ”识别要操作的对象。

说明

不必对操作员权限进行检查，因为该检查已集成在操作对话框中。

示例脚本

调用操作对话框但不检查操作员权限

```
#include "apdefap.h"
void OnRButtonUp (char*lpzPictureName,
                  char*lpzObjectName,
                  char*lpzPropertyName,
                  UINT nFlags,
                  int x,
                  int y)
{
#define MFPIC "@PG_MyAPLFP_Standard.pdl"
#define MFOBJ "@csolOAnaHiAlm"

    APL_OpenFaceplate(lpzPictureName,lpzObjectName,1,MFPIC,MFOBJ);
//1: MF=OP Dialog
}
```

6.3.3.3 调用操作对话框前检查操作员权限

可按以下示例脚本中所示，在调用操作对话框前检查操作员权限：

- 该脚本将检查“Status1”状态字中的第 2 位。
- 只有在置位第 2 位（块切换到仿真状态）时，才会调用操作对话框。

示例脚本

调用操作对话框前检查操作员权限

```
#include "apdefap.h"
void OnRButtonUp
    (char*lpszPictureName,
    char*lpszObjectName,
    char*lpszPropertyName,
    UINT nFlags,
    int x, inty)
{
#define BIT(i,status) (((unsigned long)status & (1<<i))!=0)
#define MFPIC "@PG_PidConL_Standard.pdl"
#define MFOBJ "@csolOAnaPV"

    unsigned long value = (unsigned long)GetPropDouble(lpszPictureName,
        lpszObjectName, "Status1");
        BOOLxSimuOn=BIT(2, value);

        if (xSimuOn)
        {
            APL_OpenFaceplate(lpszPictureName, lpszObjectName, 1, MFPIC, MFOBJ);
        }
}
```

说明

如果使用面板的副本来创建此功能，则必须修改脚本中通过块图标实现的操作。由于面板视图的名称在输入后是固定不变的，因此必须修改脚本。

示例：

- 源代码：

```
#define MFPIC "@PG_PidConL_Standard.pdl"
```

- 更改为：

```
#define MFPIC "@PG_MyAPLFP_Standard.pdl"
```

(参见上面的代码示例)

6.4 创建面板

6.4.1 生成现有面板的副本

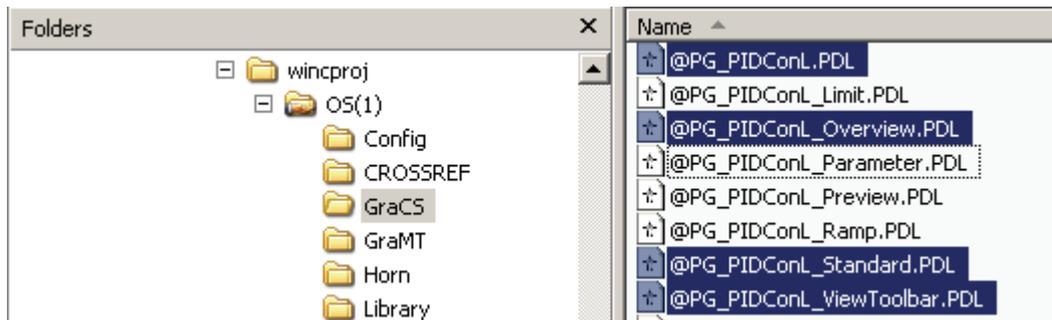
简介

此示例举例说明了如何基于 APL 标准控制器“PIDConL”创建项目特定的版本。

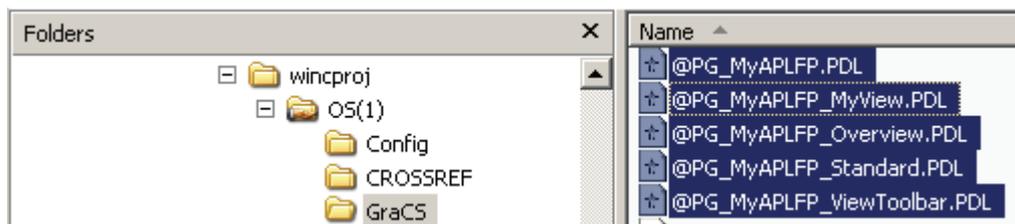
步骤

要依照 APL 创建面板，请按以下步骤操作：

1. 从项目模板中复制相应面板（例如，PIDConL）的文件。



2. 以相应的块名称或面板版本名称来保存这些文件。



示例中的视图

本示例中应用了以下视图

- 标准视图
- 报警视图
- 趋势视图
- 批生产视图

- 备注视图
- MyView（用户自己创建的视图）

创建了以下模板画面的副本：

	用户面板的新名称示例
基本画面， 面板的基本视图。	@PG_MyAPLFP.PDL
• 总览	@PG_MyAPLFP_Overview.PDL
• 视图选择	@PG_MyAPLFP_ViewToolbar.PDL
• 标准视图	@PG_MyAPLFP_Standard.PDL
• MyView（用户自己创建的视图）	@PG_MyAPLFP_MyView.PDL

全局视图

对于所有面板类型而言，以下全局视图都是只有一个。

全局视图	全局画面
报警视图	@PG_apl_message.pdf
趋势视图	@PG_apl_trend.pdf
批生产视图	@PG_apl_batch.pdf
备注视图	@PG_apl_memo.pdf

这些视图无需副本。在需要时，将直接从视图选择 @PG_MyAPLFP_ViewToolbar.PDL 中访问全局画面。

6.4 创建面板

6.4.2 修改面板（总览）

创建面板

功能	部分
创建副本	生成现有面板的副本 (页 86)
修改视图 <ul style="list-style-type: none"> • 标准视图 • 报警视图 • 趋势视图 • 批生产视图 • 备注视图 • MyView（用户自己创建的视图） 	<ul style="list-style-type: none"> • 修改基本视图“@PG_MyAPLFP.PDL” (页 89) <ul style="list-style-type: none"> – “Firstview”属性中的画面名称 – “ToolbarWindow”对象中的画面名称 – “ObjCollection”对象“BlockType”属性中的面板类型 • 修改总览画面“@PG_MyAPLFP_Overview.PDL” (页 90) <ul style="list-style-type: none"> – AS 块或面板中的互连 • 修改标准画面“@PG_MyAPLFP_Overview.PDL” (页 90) <ul style="list-style-type: none"> – 删除画面“@PG_MyAPLFP_Standard.pdl”中未使用的对象 • 修改视图选择“@PG_MyAPLFP_ViewToolbar.PDL” (页 91) <ul style="list-style-type: none"> – 更改视图选择
分配操作员权限	更改或分配操作员权限 (页 94)
指定面板中的数字格式	使用脚本为模拟值指定实例特定的数字格式 (页 95)
全局设置	面板对象的全局属性 <ul style="list-style-type: none"> • 对象的颜色梯度 (页 97) • TypeID 和 VersionID (页 98) • SelectionBorder (页 99)
插入面板对象	<ul style="list-style-type: none"> • 显示模拟值 (页 99)（例如： APL_ANALOG_OP_DISPLAY） • 针对 APL_ANALOG_OP_DISPLAY 对象的 模拟值操作 (页 103) • 操作二进制值 (页 110)（例如： APL_OP_BUTTON） • 使用文本框显示二进制值 (页 119) • 棒图 (页 107) • 复选框 (页 115) • 显示信号状态 (页 122) • 显示操作锁 (页 123)
准备面板之间的跳转	<ul style="list-style-type: none"> • 复制按钮并指定文本 (页 69)

6.4.3 创建面板

6.4.3.1 画面 @PG_MyAPLFP.pdl 中的通用规范

必须更改基本画面"@PG_MyAPLFP.PDL"中的属性。

说明

保存基本画面

在保存基本画面时，画面高度必须设置为“1”。

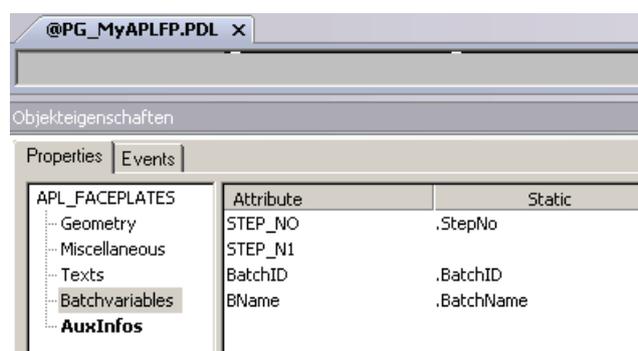
步骤

必须**始终**设置以下属性：

1. 在"@Faceplate"对象的 "Firstview" 属性中输入新画面名称，例如 "@PG_MyAPLFP_Standard.PDL"。
2. 在"ToolbarWindow"对象的 "PictureName" 属性中输入新画面名称，例如 "@PG_MyAPLFP_ViewToolbar.PDL"。
3. 在"ObjCollection"对象的 "BlockType" 属性中输入新面板类型的名称（服务器名称），例如"MyAPLFP"。

AS 块中没有任何用于 SIMATIC BATCH 的变量

如果 AS 块没有任何用于 SIMATIC BATCH 的变量，则请删除"@Faceplate"对象中的相应条目。



6.4 创建面板

6.4.3.2 画面 @PG_MyAPLFP_Overview.pdl 中的通用规范

如果在面板总览中使用控制器 (@PG_PIDConL_Overview.pdl) 模板的副本，则应修改 AS 块或画面 (*.PDL) 中的以下默认值：



- | | | |
|---|-----------|---------------------|
| ① | MSGLOCK | 互连到参数 STATUS2; BIT0 |
| ② | 最差信号状态 | 互连到参数 ST_Worst |
| ③ | 批生产被占用 | 互连到参数 STATUS1; BIT0 |
| ④ | 维护状态“进行中” | 互连到参数 STATUS1; BIT4 |

说明

为“最差信号状态”使用自定义参数名称

如果使用其它的参数名称替换参数名称 "ST_Worst"，则还必须修改彼此覆盖的两个状态显示中的参数名称。

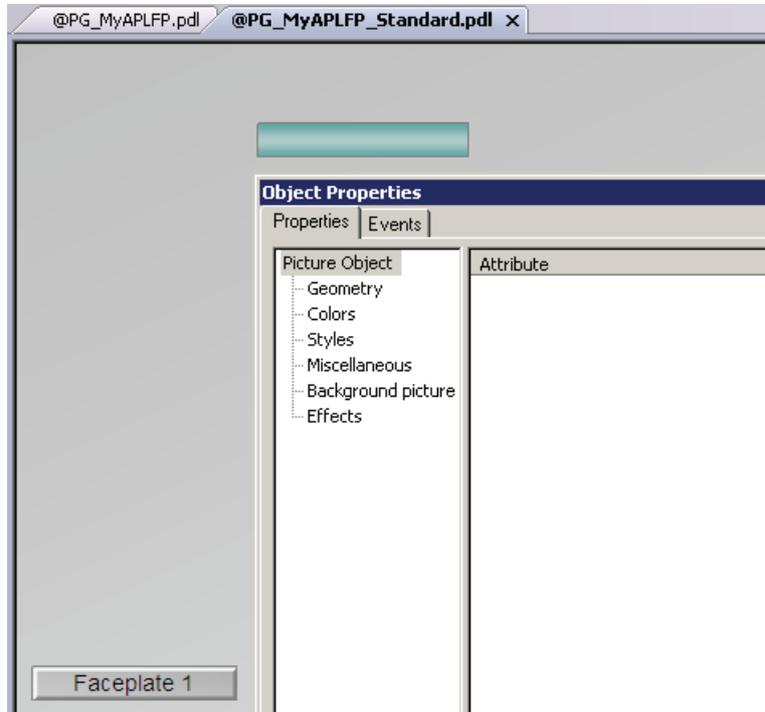
6.4.3.3 画面 @PG_MyAPLFP_Standard.pdl 中的通用规范

删除副本中除下图所示对象之外的所有其它对象。

以下对象应保持不变：

- "csoFpBtnUser1"，用于跳转到与 SelFp1 互连的块。
- "Level1"
- "Level2"
- "Level3"

- "rect_selectionBorder"
- "stUser"



6.4.3.4 更改视图选择

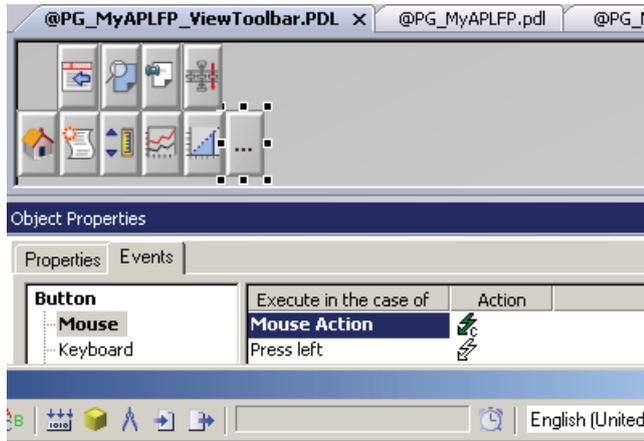
下图显示了示例画面"@PG_MyAPLFP_ViewToolbar.PDL"及其各个元素。

前景中的对象已经下移，因此会显示用于选择各种视图的元素。背景中的对象在顶行显示。

- 使用“STANDARD”对象（左下方）选择标准视图。此对象始终可见且可选。
“NEXT”对象（右下方）可激活“鼠标单击”的脚本，该脚本用于切换顶部对象和底部对象的显示。“NEXT”对象始终可见且可选。

下图显示了切换底部和顶部对象显示的脚本：

6.4 创建面板



```
//Button set 1
SetPropBOOL(lpszPictureName,"MESSAGE", "Visible", bButtonSet1);
SetPropBOOL(lpszPictureName,"LIMIT", "Visible", bButtonSet1);
SetPropBOOL(lpszPictureName,"TREND", "Visible", bButtonSet1);
SetPropBOOL(lpszPictureName,"RAMP", "Visible", bButtonSet1);

//Button set 2
SetPropBOOL(lpszPictureName,"PARAMETER", "Visible", bButtonSet2);
SetPropBOOL(lpszPictureName,"PREVIEW", "Visible", bButtonSet2);
SetPropBOOL(lpszPictureName,"MEMO", "Visible", bButtonSet2);
SetPropBOOL(lpszPictureName,"BATCH", "Visible", bButtonSet2);
```

- 按钮的对象名称决定了所调用视图的名称。

示例：

含有 "Name"=MyView 对象的按钮用于选择全新创建的视图 "@PG_MyAPLFP_MyView.PDL"。

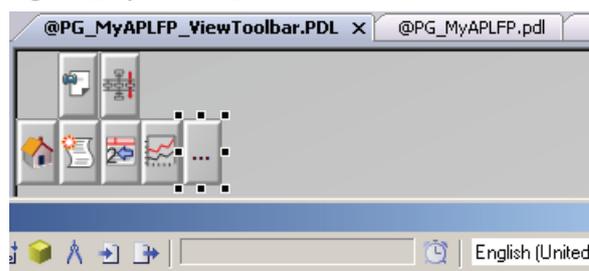
步骤

- 在按钮的“工具提示文本”(ToolTip Text) 属性中输入相应的文本，例如“MyView”。
- 用户可在“画面状态开启”(PictureStatus On) 和“画面状态关闭”(PictureStatus Off) 属性中，保存自定义的按钮画面。

模板画面包含一些匹配的按钮选项：



- 可按下图所示组态视图选择，请参考本设计指南中描述的示例。
 - 修改脚本中“NEXT”对象的条目。
 - 要确保在调用合成画面时显示上次的工具栏，则应修改 "@PG_MyAPLFP_ViewToolbar.PDL" 画面的“画面选择”事件脚本中相应的条目。



```
//Button set 1
SetPropBOOL(lpszPictureName,"MESSAGE", "Visible", bButtonSet1);
SetPropBOOL(lpszPictureName,"MyView", "Visible", bButtonSet1);
SetPropBOOL(lpszPictureName,"TREND", "Visible", bButtonSet1);

//Button set 2
SetPropBOOL(lpszPictureName,"MEMO", "Visible", bButtonSet2);
SetPropBOOL(lpszPictureName,"BATCH", "Visible", bButtonSet2);
```

结果

上图显示了前景中的对象。 这些对象显示用于选择以下视图的按钮：

- 标准视图
- 报警视图
- **MyView**（用户自己创建的视图）
- 趋势视图

单击“...”按钮可切换显示。(Next)。

6.4 创建面板

将显示用于选择视图的按钮：

- 备注视图
- 批生产视图

6.4.3.5 更改或分配操作员权限

在以下情况下，“stUser”对象会通过脚本来更新画面“@PG_MyAPLFP.pdl”中用户所列对象的可操作性：

- 在选择画面时
- 在切换用户（在面板未关闭的情况下）时
- 在更改本地操作员权限时

互连可操作对象的属性

还应互连可操作对象的以下属性：

- 对于 AS 使能端（例如“OS_PermLog”，更改属性“PermissionTag”
- 对于模拟显示，更改属性“LinkTag3”。

说明

如果这些属性的默认值为 16#FFFFFFFF 或 4294967295，则并不强制要求互连“PermissionTag”或“LinkTag3”属性。

修改脚本

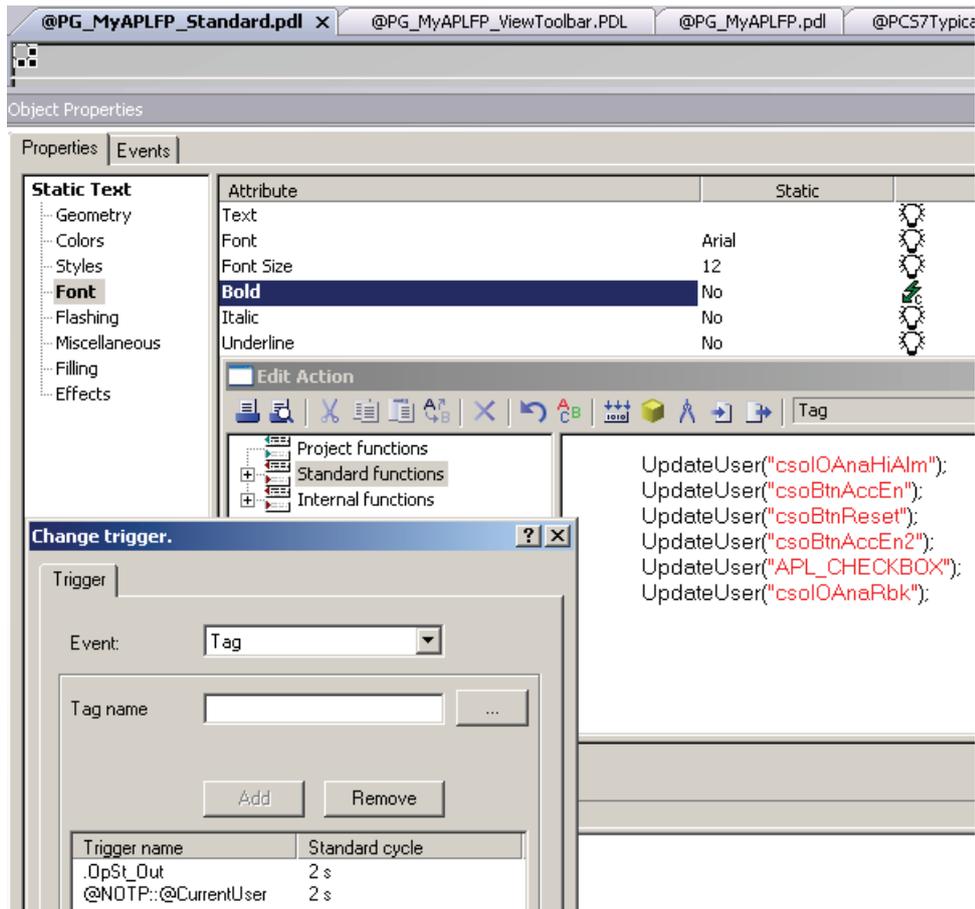
修改脚本中“stUser”对象的“粗体”(bold) 属性，如下所示：

- 列出脚本中的可操作对象。
- 将未使用的元素从原始文件中删除或将其注释掉。
- 直接对可操作对象设置操作员权限级别：
 - 对于模拟值操作，更改“OperationLevel”属性
 - 对于二进制值操作，更改“BinAuthorizationLevel1 .. x”属性

对象的操作员权限级别

- 1 = 过程操作
- 2 = 更高级过程操作
- 3 = 最高级过程操作

下图显示了带有可操作对象的脚本：



更多信息

有关 AS 使能端的参数分配的更多信息，请参见以下章节：

- “二进制值操作/PermissionTag”一章
- “模拟值操作/过程相关的使能端”一章。

6.4.3.6 模拟值的数字格式

使用 APL 中视图的画面选择脚本，组态特定实例中的数字显示格式（小数和小数位）。与直接连接相比，使用脚本解决方案效果会更好一些。

6.4 创建面板

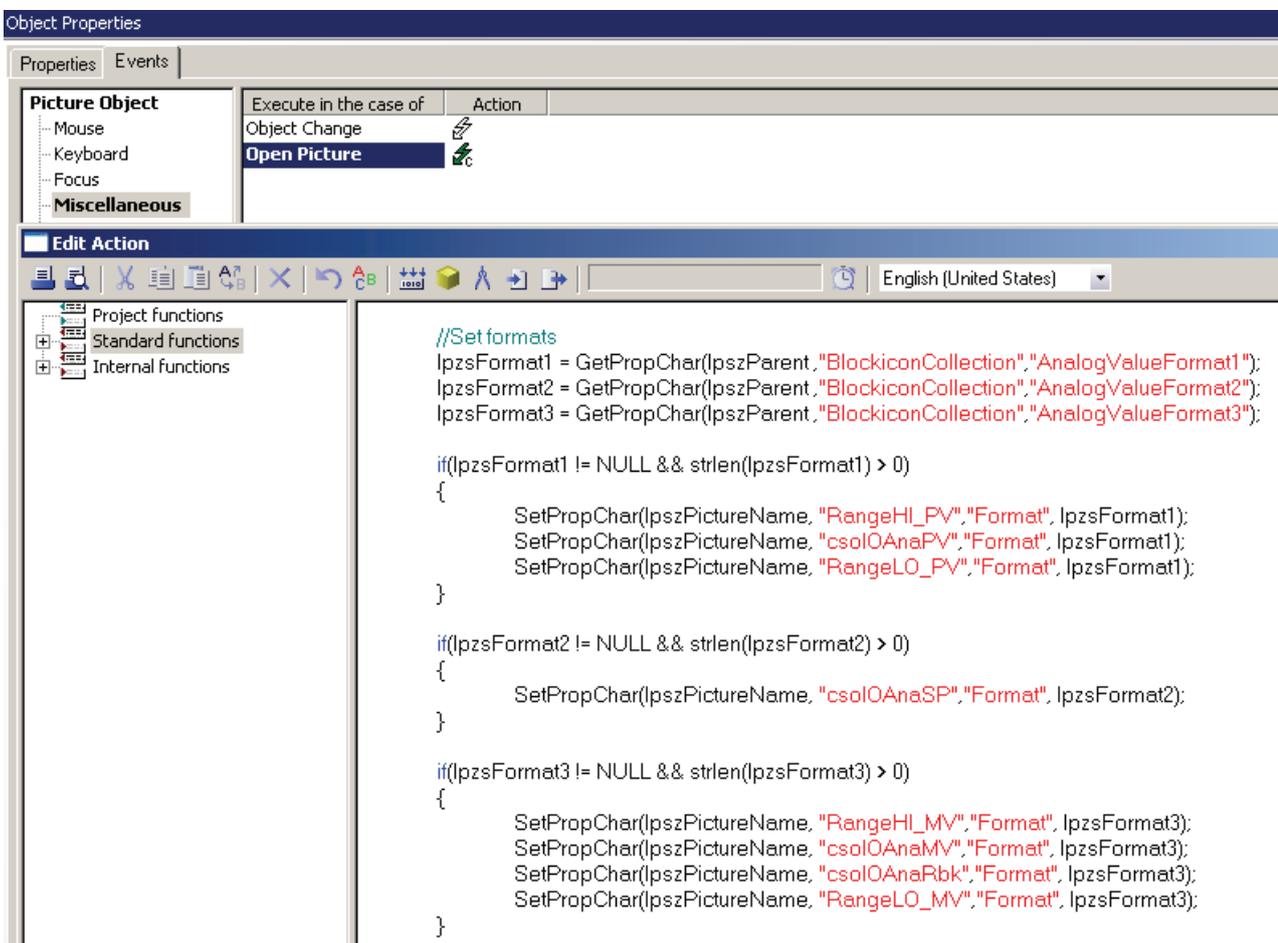
步骤

修改画面选择属性中操作的脚本，如下所示：

- 将未使用的元素从原始文件中删除或将其注释掉。
- 将要以指定格式显示的模拟值分配给脚本中相应的模拟值格式，例如 "AnalogValueFormat1"、"AnalogValueFormat2" 或 "AnalogValueFormat3"。

示例代码

下图显示了模拟值的格式化脚本：



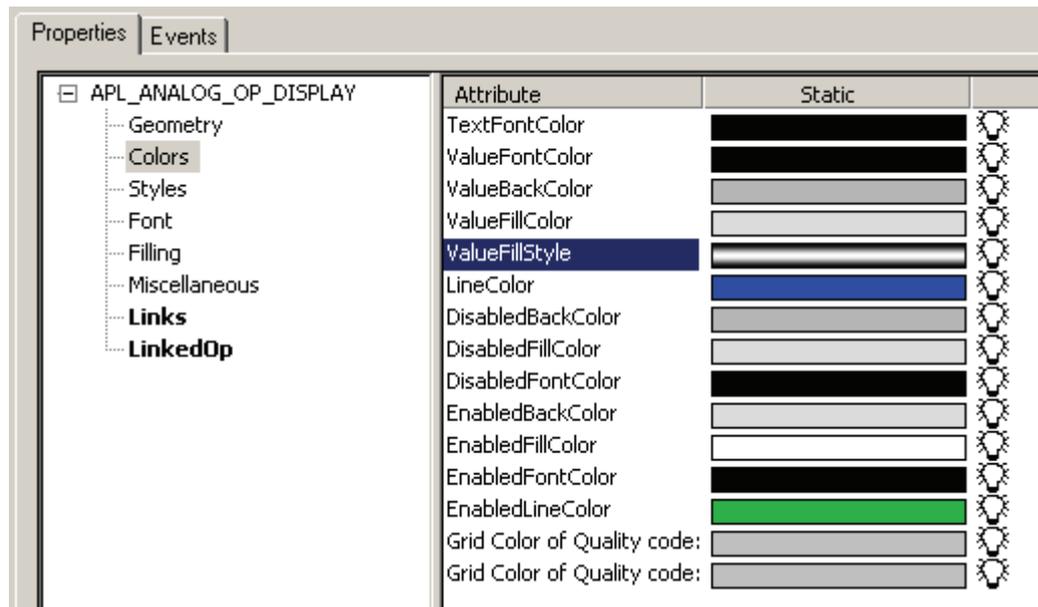
有关面板测试的建议

使用全局脚本诊断来测试已创建的面板。如此可轻松地识别出已不存在但脚本中仍在引用的对象。

6.4.4 面板对象的全局属性

APL 面板中用户对象的全局属性说明。

6.4.4.1 对象的颜色梯度



根据所使用的画面对象，在相应属性中使用 `ValueFillStyle = 7`。

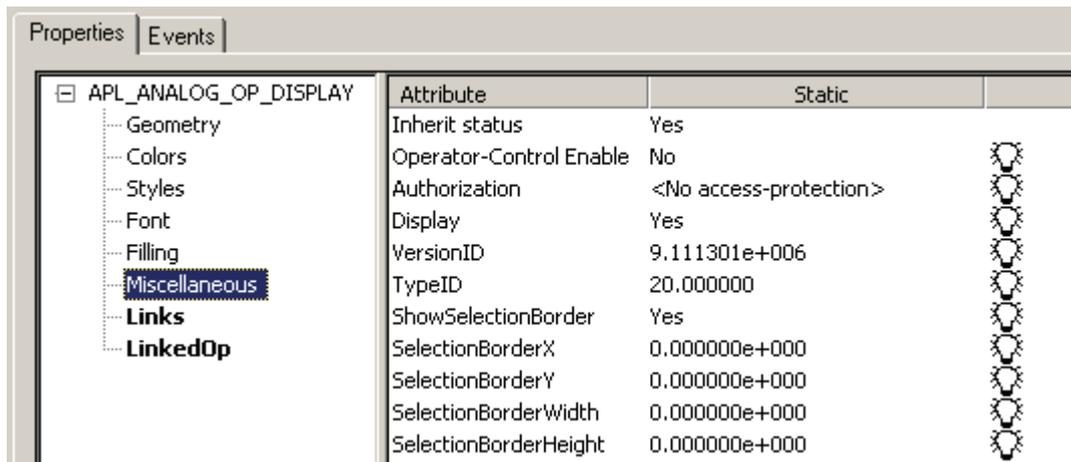
有关更多信息，请参见章节“颜色规范 (页 36)”。

6.4 创建面板

6.4.4.2 TypeID 和 VersionID

用户对象的属性

属性“VersionID”和“TypeID”是用户对象的属性。



属性 "VersionID"

属性 "VersionID" 用于对用户对象进行版本控制。

- 此处以日期形式存储版本。
- 上图中显示了相应的表示方式：9.111301+006 表示 11/13/2009。

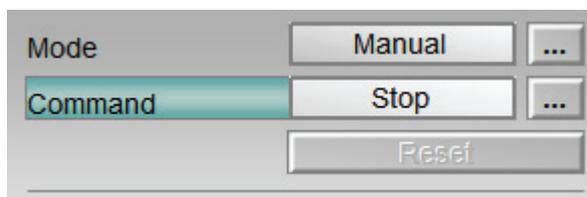
属性 "TypeID"

"TypeID"属性是分配给 APL 面板中所用用户对象的编号。

- 如果仅对结构中未更改的 APL 用户对象进行组态，则"TypeID"可保持不变。
- 如果更改对象（例如，包含新的内部对象），则需要指定新的"TypeID"。新的"TypeID"必须大于 2500，因为
- TypeID 一直到 1999（包含在内）都预留给 APL 使用。

6.4.4.3 ShowSelectionBorder

使用属性 "ShowSelectionBorder" = Yes 以在选中可操作功能时，赋予该功能一个彩色标记。



APL 标准对象的规范

对于 APL 标准对象，如果满足以下条件，则会在相对于被操作对象固定的位置上输入该标记：

- 尺寸保持不变。
- 为要选择的对象的 "Position X" 属性分配标准值“154”。
- 为以下从属属性分配值“0”：
 - "SelectionBorderX"
 - "SelectionBorderY"
 - "SelectionBorderWidth"
 - "SelectionBorderHeight"

具体设计

可通过“APL 标准对象的规范”部分指定的属性来为彩色标记指定具体的位置或尺寸。

6.4.5 显示和操作模拟值

6.4.5.1 APL_ANALOG_OP_DISPLAY 模拟值显示

模拟值显示和模拟值操作使用相同的对象。但参数分配和变量连接的步骤不同。

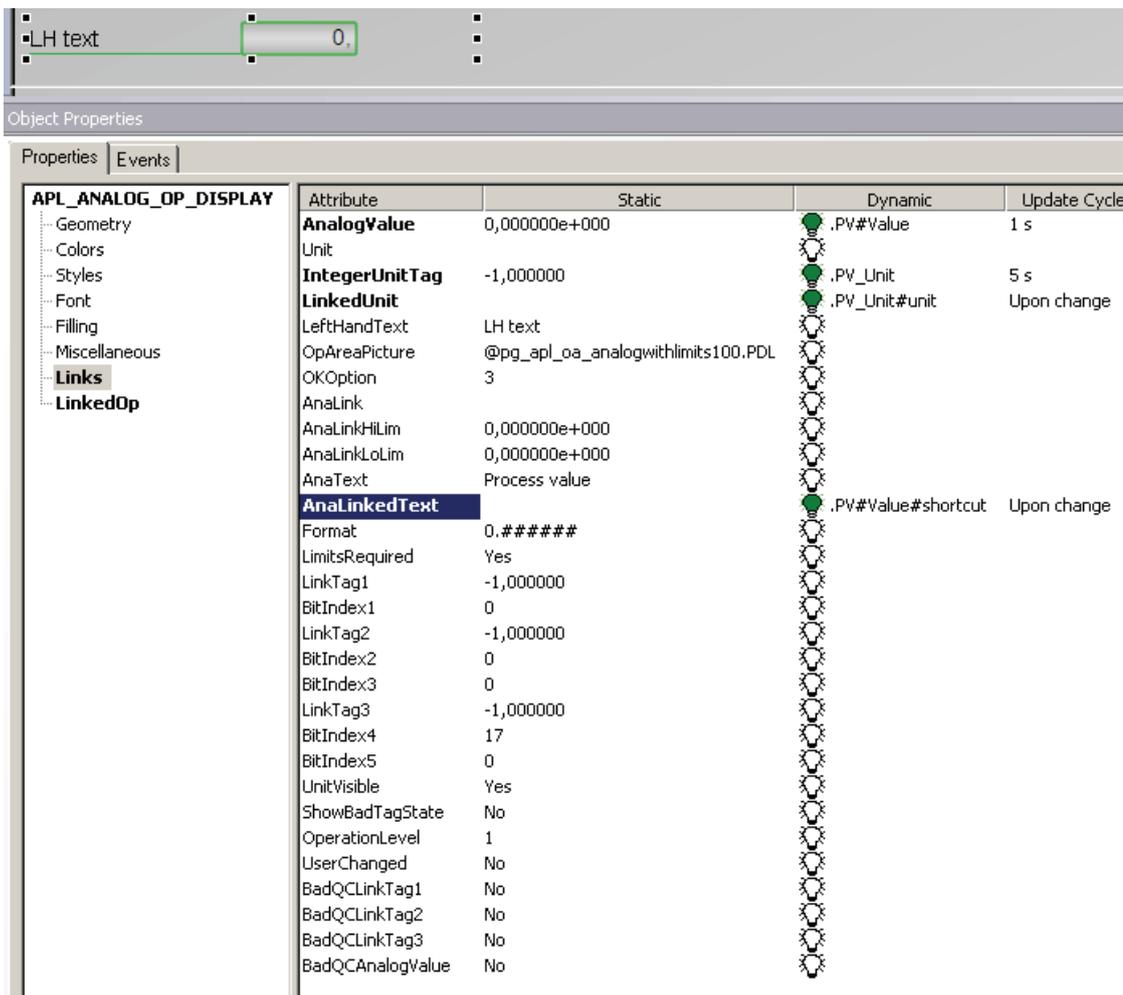
6.4 创建面板

为模拟值显示分配参数

1. 从模板画面中复制 APL_ANALOG_OP_DISPLAY(@PCS7ElementsAPL.pdl; Display analog Value; Setpoint) 对象，并将其插入面板视图中。
2. 在用于指定数字格式的画面选择脚本中输入对象名称。

有关更多信息，请参见章节“模拟值的数字格式 (页 95)”。

有关确定放置位置的信息，请参见章节“面板的规范 (页 19)”。



模拟值显示相关属性的简要说明。

随后将给出详细说明。

属性	简要说明
----	------

属性	简要说明
颜色 > LineColor	设置模拟值显示边距的颜色
链接 > AnalogValue	显示实际模拟值
链接 > Unit	显示单位
链接 > IntegerUnitTag	提供通过数字和表格来确定的单位 注： 有关此表格及符合“PROFIBUS 和 PROFINET 国际组织”的“过程设备的 PA 配置文件”规范的测量单位的相关信息，请参见 APL 在线帮助。
链接 > LinkedUnit	提供在数字 = 0 时，通过 S7_Unit 属性给出的单位
链接 > LeftHandText	未使用任何 S7_shortcut 时的显示标签
链接 > AnaLinkedText	使用了 S7_shortcut 时实例特定的显示标签
链接 > UnitVisible	隐藏测量单位

互连模拟值

- 实际模拟值与 "AnalogValue" 属性互连。
- 通过 "LineColor" 属性组态模拟值显示的彩色边距显示。

表示测量单位

在对象中有三个显示测量单位的属性。

属性	含义
Unit	“Unit”是测量单位的显示文本
IntegerUnitTag	“IntegerUnitTag”与整型变量（例如“PV_Unit”）互连。
LinkedUnit	“LinkedUnit”与引用文本词典的 OS 内部变量“PV_Unit#unit”互连。

6.4 创建面板

“IntegerUnitTag”事件的脚本

事件“IntegerUnitTag”中互连了执行以下操作的脚本：

- IntegerUnitTag = 0；此脚本传送“S7_unit”属性的文本。
- IntegerUnitTag > 0；此脚本通过整数值在测量单位表中搜索相应的文本。这个包含标准单位的表格将存储在 OS 中。

说明

如果在 SCL 源代码的结构中创建了单位属性“S7_xunit”，则在编译 OS 过程中将不会考虑该属性。

同时也不会创建 OS 内部变量。

使用 "UnitVisible" 属性可禁止测量单位的显示。

删除未使用的动态属性

必须删除以下属性中剩余的动态属性：

- "AnaLinkHiLim"
- "AnaLinkLo"
- "LimLinkOp"
- "LinkTag3"

设置 LinkTag3=0。然后便无法操作该对象。

标记模拟值的显示

可将 S7_Shortcut 属性与“AnaLinkedText”属性互连，从而创建实例特定的文本，例如“PV#Value#shortcut”。

对于 OS 内部变量的情况，建议始终将触发变量设置为“更改”。

- 如果 S7_Shortcut 中未输入任何文本，则显示 "LeftHandText" 属性中的输入文本作为标签。
- 如果 S7_Shortcut 中已输入文本，则将其显示为标签。

有关“AnaText”属性的信息，请参见章节“针对 APL_ANALOG_OP_DISPLAY 对象的模拟值操作 (页 103)”。

参见

组态模拟值显示的高亮显示 (页 27)

6.4.5.2 针对 APL_ANALOG_OP_DISPLAY 对象的模拟值操作

下面部分将介绍模拟值操作的参数分配。

6.4 创建面板

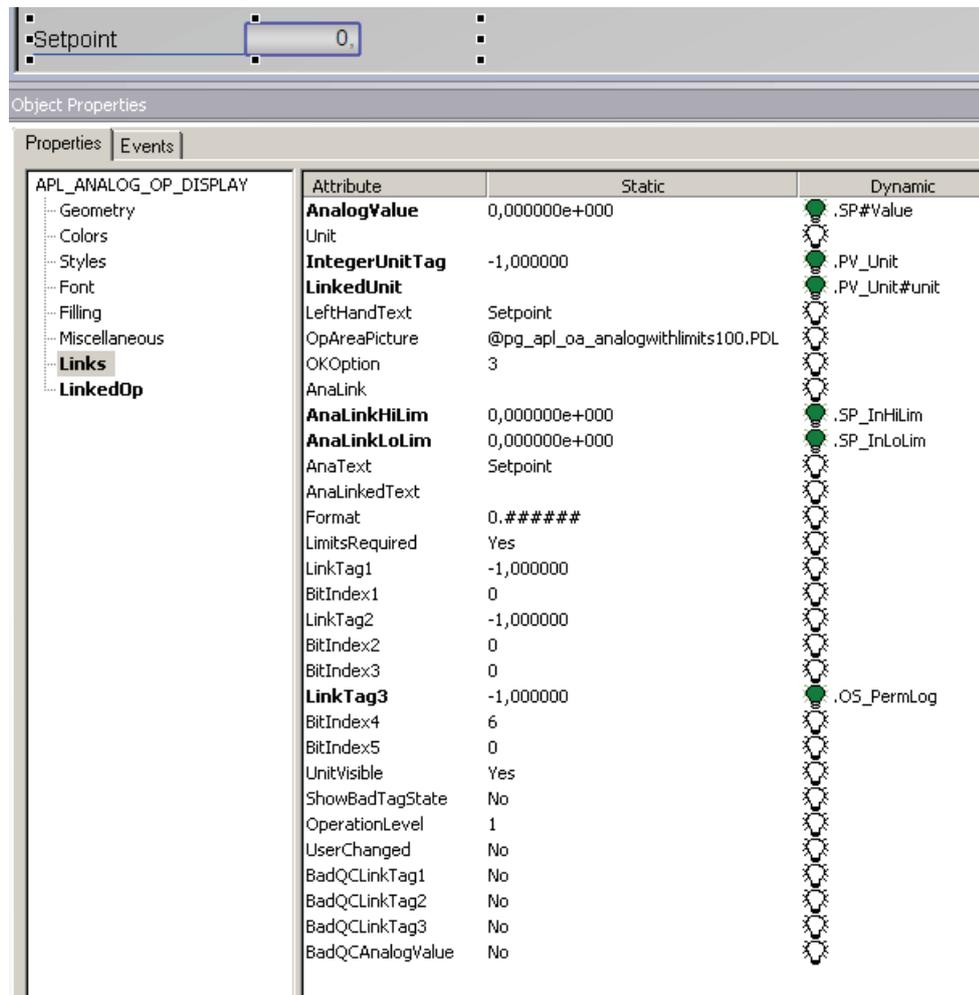
步骤

1. 从模板画面中复制 APL_ANALOG_OP_DISPLAY (@PCS7ElementsAPL.pdl; Display analog Value; Setpoint) 对象，并将其插入面板视图中。

有关确定放置位置的信息，请参见“ShowSelectionBorder (页 99)”。

2. 在用于指定数字格式的画面选择脚本中输入对象名称。
有关详细信息，请参见“模拟值的数字格式 (页 95)”一章

3. 在面板的“stUser”对象中输入对象名称，以在切换用户时更新操作员权限。



模拟值操作相关属性的简要说明

随后将给出详细说明。

属性	简要说明
Links > AnalogValue	显示实际模拟值
Links > Unit	显示单位
Links > IntegerUnitTag	提供通过数字和表格来确定的单位 注： 有关此表格及符合“PROFIBUS 和 PROFINET 国际组织”的“过程设备的 PA 配置文件”规范的测量单位的相关信息，请参见 APL 在线帮助。
Links > LinkedUnit	提供在数字 = 0 时，通过 S7_Unit 属性给出的单位
Links > LeftHandText	未使用任何 S7_shortcut 时的显示标签
Links > OpAreaPicture	选择操作区域（例如带有或不带滚动条）
Links > OkOption	在执行“正常”完成后计算输入值
Links > AnaLinkHiLim	度量滑块的范围上限和增量调节
Links > AnaLinkHiLim	度量滑块的范围下限和增量调节
Links > AnaText	用于操作日志的文本
Links > AnaLinkedText	使用了 S7_shortcut 时实例特定的显示标签
Links > Format	要显示的模拟值的数字格式
Links > LinkTag3	与过程相关的操作员权限 AS 变量
Links > BitIndex4 Bit	字中过程相关的操作员权限所在的地址。
Links > UnitVisible	隐藏测量单位
Links > OperationLevel	组态操作员权限级别
LinkedOp > LinkOp	要操作的模拟值

组态模拟值的显示

将要显示的模拟值与 "AnalogValue" 属性互连。

6.4 创建面板

模拟值的操作员控制

将要操作的参数与 "LinkOp" 属性互连。如果此参数与要显示的值相同，则再次互连这一参数。

有关测量单位、显示标签和格式化的信息，请参见“APL_ANALOG_OP_DISPLAY 模拟值显示 (页 99)”部分。

操作限值

"AnaLinkHiLim" 和 "AnaLinkLoLim" 属性用于指定模拟值操作的操作限值。它们对于滑块操作和增量操作起决定作用。

用于操作日志的文本

在 "AnaText" 属性中输入操作日志文本。必须在此处输入，否则操作日志中将缺少被操作的参数。

过程相关的使能端

使用 "LinkTag3" 属性可互连操作的过程相关使能端。

- **建议:**

- 使用 "OS_PermLog" 参数。

- 使用 "BitIndex4" 为在包含该使能端的双字中的位（从 Bit0 开始）分配参数。

- 如果不希望将变量连接到 "LinkTag3" 属性，则必须执行以下操作：

- 将 "LinkTag3" 中的值设置为从 -1 到 +1。

- 使用 "BitIndex4" 对 Bit0 寻址。

用户特定的使能端

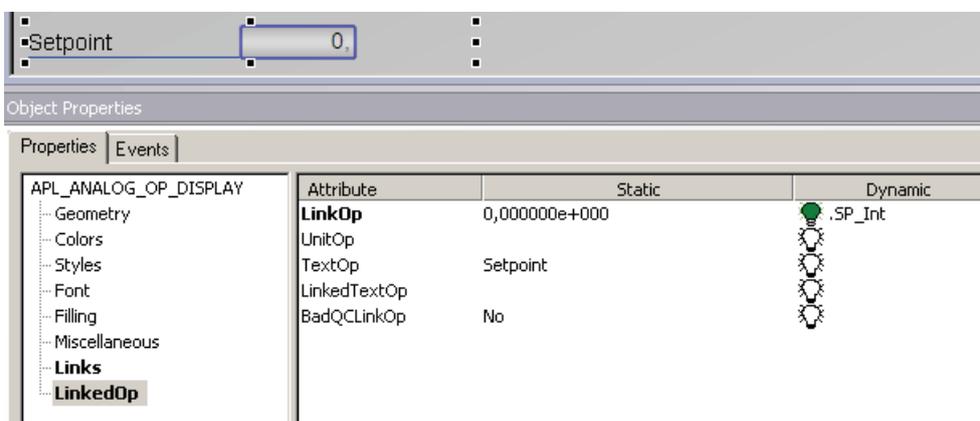
在 "OperationLevel" 属性中设置用户特定的权限级别。

对象的权限级别

- 1 = 过程操作
- 2 = 更高级过程操作
- 3 = 最高级过程操作

操作区域

- 通过 "OpAreaPicture" 属性选择操作区域的类型。
对于常规模拟值操作，可选择
 - @PG_APL_OA_Analog100 -> ，无测量范围
 - @PG_APL_OA_Analog101 -> ，有测量范围
 - @PG_APL_OA_AnalogWithLimits -> ，有测量范围、滑块、增量操作。
- 使用 "OkOption" 属性来确定操作执行后的行为。对于简单模拟值操作，该属性始终为 3。
- 在 "LinkedOp" 选项卡（如下图所示）中，只有 "LinkOp" 属性比较重要，其余属性保留即可。



参见

画面 @PG_MyAPLFP_Standard.pdl 中的通用规范 (页 90)

排列显示和操作对象 (页 25)

组态模拟值的显示 (页 26)

6.4.5.3 棒图

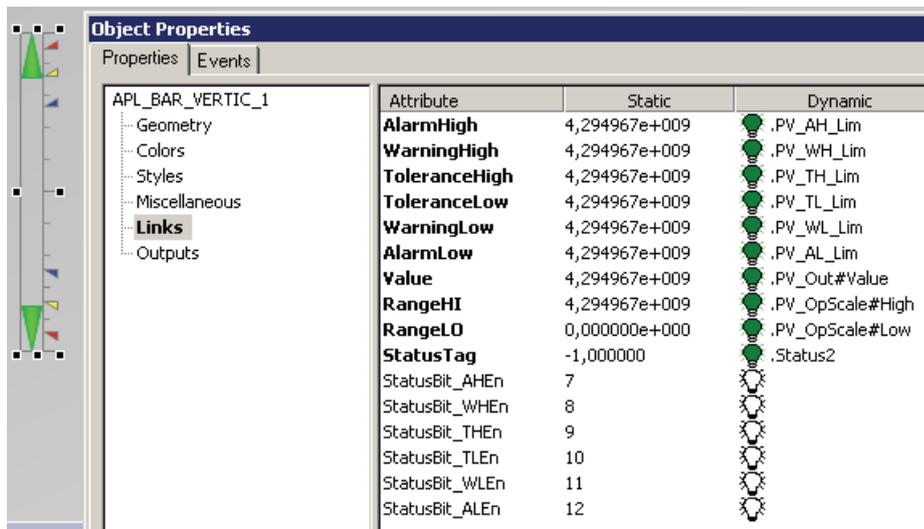
以下部分包含 APL 面板中所用棒图的属性（可通过参数进行设置）：

- 最多有三个限值对的实际值垂直棒图
- 有一个限值对的设定值垂直棒图
- 水平棒图

6.4 创建面板

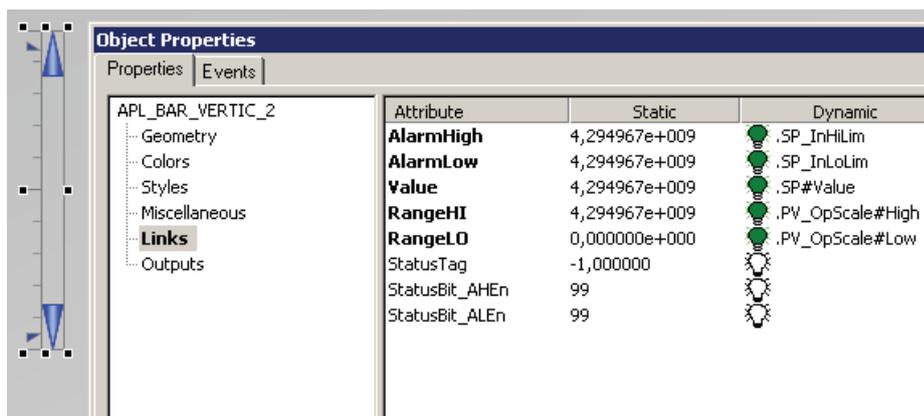
最多有三个限值对的实际值垂直棒图

1. 从模板画面中复制 "APL_BAR_VERTICAL_1"(@PCS7ElementsAPL.pdl; Bar graph display) 对象，并将其插入面板视图中。
2. 指定一个有意义的对象名称。



有一个限值对的设定值垂直棒图

1. 从模板画面中复制 "APL_BAR_VERTICAL_2"(@PCS7ElementsAPL.pdl; Bar graph display) 对象，并将其插入面板视图中。
2. 指定一个有意义的对象名称。



原则上，此处可使用与 APL_BAR_VERTICAL_1 棒图相同的操作步骤。

不同的是，这里的棒图为蓝色，仅显示的一个限值对，且不可通过事件来控制该限值对的显示/隐藏。

使用属性来显示/隐藏限值对

若要显示或隐藏限值对，需执行以下操作：

1. 对 "StatusTag" 属性应用一个状态字。
2. 通过 "StatusTag" 属性创建事件脚本。

可使用 APL_BAR_VERTIC_1(@PCS7ElementsAPL.pdl; Bar graph display) 对象的相应脚本作为副本模板。

水平棒图

1. 从模板画面中复制 APL_BAR_HORIZ_1 或 APL_BAR_HORIZ_2 (@PCS7ElementsAPL.pdl; Bar graph display) 对象，并将其插入面板视图中。
2. 指定一个有意义的对象名称。



除了颜色和水平位置外，水平棒图与 APL_BAR_VERTIC_1 棒图完全相同。

棒图的属性

属性	含义	设置
棒图	“值”(Value) 属性	使用“值”(Value) 属性来互连要在棒图中显示的模拟值。
	“颜色”(Colors) 对话框	颜色表示可在“颜色”(Colors) 对话框中进行指定。棒图的标准颜色：绿色
指定测量值范围	“RangeHI”和“RangeLO”属性	使用“RangeHI”和“RangeLO”属性来互连要显示的测量值。

6.4 创建面板

属性	含义	设置
限值	限值的属性	<p>可使用以下属性在棒图中显示限值：</p> <ul style="list-style-type: none"> • “AlarmHigh” • “WarningHigh” • “ToleranceHigh” • “AlarmLow” • “Warninglow” • “ToleranceLow” <p>系统按如下方式表示这些属性：</p> <ul style="list-style-type: none"> • 红色箭头表示报警 • 黄色箭头表示警告 • 蓝色箭头表示容差
“StatusTag”属性	“StatusTag”属性	<p>使用“StatusTag”属性互连状态字。</p> <ul style="list-style-type: none"> • 基本属性“StatusBit_xxxx”指定状态字中相应位的位置。 示例： “StatusBit_0”表示第 0 位 • 可通过事件控制来指定是否显示限值标记。 示例： AS 块中状态字“Status2”的第 7 位为 1 时，棒图中将显示限值标记“报警上限”(Alarm High)。

参见

组态棒图 (页 32)

6.4.6 显示和操作二进制值

6.4.6.1 二进制值操作 APL_OP_BUTTON

下面部分将介绍二进制值操作的参数分配。

步骤

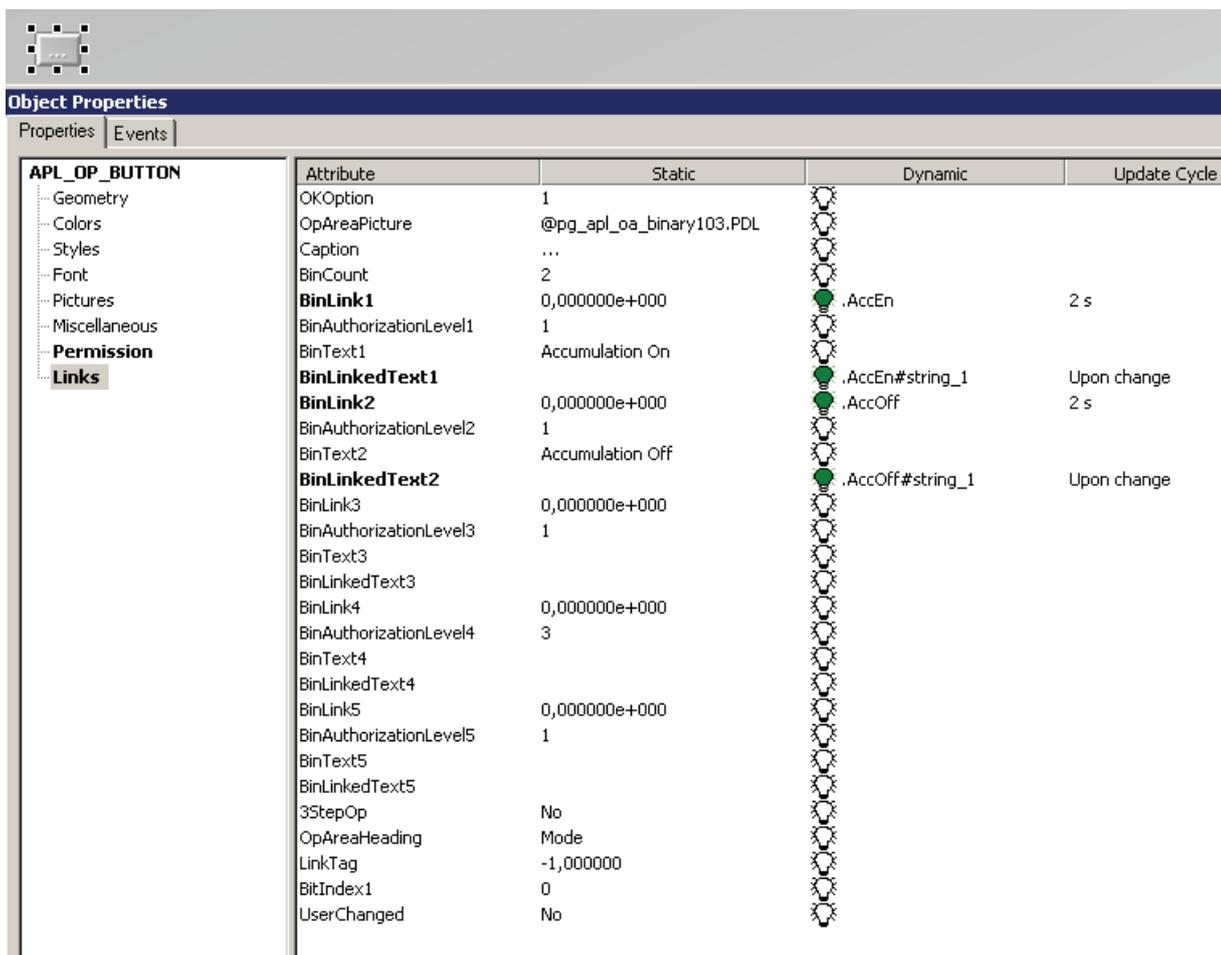
1. 从模板画面中复制 **APL_OP_BUTTON** (@PCS7ElementsAPL.pdl; Open binary operation area; Schaltfläche unter "Normal") 对象，并将其插入面板视图中。有关确定放置位置的信息，请参见“ShowSelectionBorder (页 99)”。

2. 指定一个有意义的对象名称，例如“csoBtnAccEn”。

3. 在面板的“stUser”对象中输入对象名称，以在切换用户时更新操作员权限。

有关更多信息，请参见章节“画面 @PG_MyAPLFP_Standard.pdl 中的通用规范 (页 90)”。

6.4 创建面板



二进制值操作相关属性的简要说明。

随后将给出详细说明。

属性	简要说明
Links/ OkOption	在执行“正常”完成后计算输入值
Links/ OpAreaPicture	选择操作区域（例如，一行或两行）
Links/ Caption	选择按钮的标签
Links/ BinCount	要操作的二进制元素的个数
Links/ BinLink1	二进制元素 1 的互连
Links/ BinAuthorizationLevel1	二进制元素 1 的用户权限
Links/ BinText1	用于面板中二进制元素 1 的固定文本

属性	简要说明
Links/ BinLinkedText1	通过 S7_string 为二进制元素 1 指定的实例特定文本
Links/ OpAreaHeading	操作区域标签
Permission/ PermissionTag	过程相关使能端的变量 (OS_PERM)
Permission/ BinPermBitIndex1	字中用于设置二进制元素 1 的过程相关操作员权限的位地址

通过 **APL_OP_BUTTON** 对象，用户可创建最多带五个按钮的操作对话框。

OkOption

使用 "OkOption" 属性可指定以下几项：

- 写入的变量类型
- 操作执行后的行为

参数分配

- OkOption = 1

在选择相应的按钮时，逻辑“1”将被写入与 BinLink1 到 BinLink5（最大）相连的二进制变量中。

- OkOption = 2

用于写入逻辑值“0”和“1”的二进制变量。在此情况下，BinLink1 和 BinLink2 将互连相同的变量。

- 逻辑“1”将写入与 BinLink1 相连的变量中。
- 逻辑“0”将写入与 BinLink2 相连的变量中。
- BinLink3 到 5 在此情况中并没有任何意义。

- OkOption = 4

仅限于对象 APL_OP_BUTTON3

- 将整型变量与 BinLink1 互连。
- 在“AnalogValue”属性中组态的值将写入与 BinLink1 互连的整型变量中。

- OkOption = 5

取反值将写回与 BinLink1 相连的二进制值中（切换功能）。

6.4 创建面板

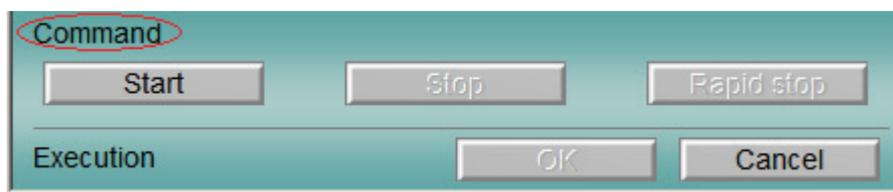
OpAreaPicture

通过此属性，可组态带三个或五个按钮的操作区域。

- "@pg_apl_oa_binary103.PDL"，最多带三个按钮的操作区域
- "@pg_apl_oa_binary105.PDL"，最多带五个按钮的操作区域

操作区域的标题

通过 "OpAreaHeading" 属性来组态操作区域的标题。



BinCount

在此处可组态按钮数量以及与 BinLink1 到 BinLink5 相连的变量连接数量。

BinAuthorizationLevel1 .. 5

使用 "BinAuthorizationLevel1 .. 5" 对各种操作设置单独的操作员权限级别。

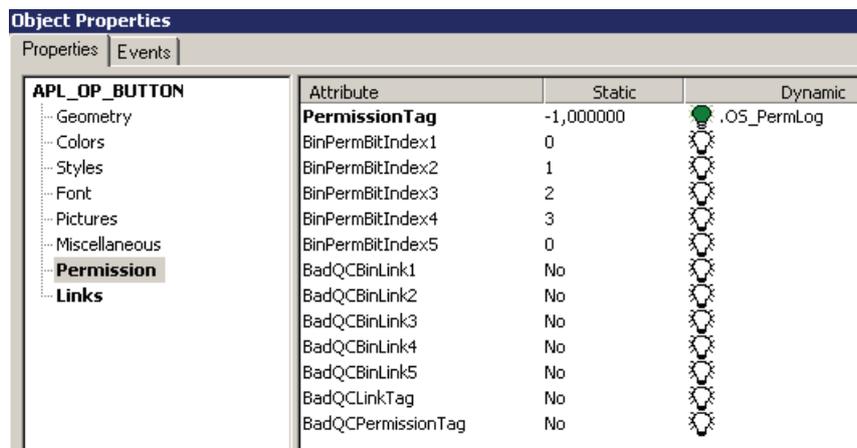
BinText1 .. 5

使用 "BinText1 .. 5" 组态按钮标签。

BinLinkedText1 .. 5

如果将相应的内部 OS 变量与 "BinLinkedText1 .. 5" 互连，则在块中组态的 S7_String0/1 可使用该按钮标签。

如果未在 S7_string 中输入任何文本，则将使用在 BinText1 .. 5 中输入的文本。



PermissionTag

过程相关使能端与此属性相连。此处通常会互连双字 "OS_PermLog"。

针对具体操作，使用 "BinPermBitIndex1 .. 5" 将参数分配给双字中的相应位（从 Bit0 开始）。

6.4.6.2 复选框

以下部分介绍了添加复选框的方法。

步骤

1. 从模板画面中复制 APL_CHECKBOX (@PCS7ElementsAPL.pdl; Check box) 对象，并将其插入面板视图中。
有关确定放置位置的信息，请参见“ShowSelectionBorder (页 99)”。
2. 指定一个有意义的对象名称，例如“APL_CHECKBOX_Reset”。
3. 在画面 @PG_MyAPLFP_Standard.pdl 的“stUser”对象中输入对象名称，以在切换用户时更新操作员权限。

有关更多信息，请参见章节“画面 @PG_MyAPLFP_Standard.pdl 中的通用规范 (页 90)”。

6.4 创建面板

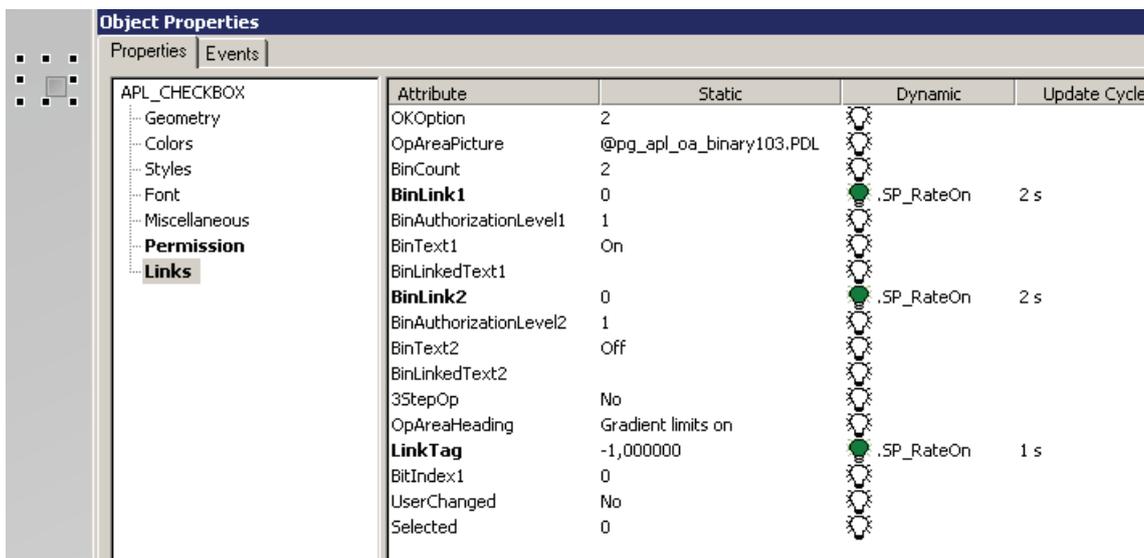


图 6-6 CheckBox1

复选框相关属性的简要说明。

随后将给出详细说明。

属性	简要说明
Links/ OkOption	在执行“正常”完成后计算输入值（1 或 2）
Links/ OpAreaPicture	选择操作区域（例如，一行或两行）
Links/ BinCount	要操作的元素个数（此处始终为 2）
Links/ BinLink1	二进制元素 1 的互连
Links/ BinAuthorizationLevel1	二进制元素 1 的用户权限
Links/ BinText1 Fester	用于面板中二进制元素 1 的文本
Links/ BinLinkedText1	通过 S7_string 为二进制元素 1 指定的实例特定文本
Links/ OpAreaHeading	操作区域标签
Links/ LinkTag	用于显示复选框中复选标记的变量
Links/ BitIndex1	LinkTag 中存在状态字时的位位置
Permission/ PermissionTag	过程相关使能端的变量 (OS_PERM)
Permission/ BinPermBitIndex1	字中用于设置二进制元素 1 的过程相关操作员权限的位地址

属性

属性	说明
OKOption	<p>可通过 "OKOption" 属性使用 APL 复选框。属性 "BinLink1" 和 "BinLink2" 将通过使用 "OKOption" 属性来求值。可进行以下设置：</p> <ul style="list-style-type: none"> 属性 "OKOption" =1 将两个不同的变量与属性 "BinLink1" 和 "BinLink2" 互连。 将向这些变量写入逻辑 1。 属性 "OKOption" =2 将同一个变量（例如 SP_RateOn，见 CheckBox1 图形）与属性 "BinLink1" 和 "BinLink2" 互连。 <ul style="list-style-type: none"> BinLink1 表示设置为布尔变量的逻辑 1 BinLink0 表示设置为布尔变量的逻辑 0
BinAuthorizationLevel1 BinAuthorizationLevel2	可通过 "BinAuthorizationLevel1 ..2" 属性来为各种操作单独设置操作员权限级别。
LinkTag	使用 "LinkTag" 属性互连用于显示复选框中复选标记的变量，逻辑 1 表示设置复选标记。
BitIndex1	如果已将状态字互连到 LinkTag，则可使用 "BitIndex1" 属性来参数化状态字中相应位（从 Bit0 开始）。
BinText1 BinText2	使用 BinText1 .. 2" 属性组态按钮标签。
BinLinkedText1 BinLinkedText2	<ul style="list-style-type: none"> 如果将相应的 OS 内部变量与 "BinLinkedText1 .. 2" 互连，则该按钮标签可来自在块中组态的 S7_string0/1。 如果在块中没有为 "S7_string0/1" 输入任何文本，则使用在 BinText1 .. 2" 属性中输入的文本。

Object Properties				
Properties Events				
	Attribute	Static	Dynamic	Update Cycle
APL_CHECKBOX	PermissionTag	-1,000000	 .OS_PermLog	1 s
-- Geometry	BinPermBitIndex1	12		
-- Colors	BinPermBitIndex2	12		
-- Styles	BadQCBinLink1	No		
-- Font	BadQCBinLink2	No		
-- Miscellaneous	BadQCLinkTag	No		
-- Permission	BadQCPermissionTag	No		
-- Links				

6.4 创建面板

PermissionTag

"PermissionTag" 属性用于互连过程相关的使能端。

- **建议:**
 - 使用双字 "OS_PermLog"。
 - 使用 "BinPermBitIndex1 .. 2" 为包含使能端的双字中的位（从 Bit0 开始）分配参数。
 - BinPermIndex1 是变量的逻辑 1 设置对应的操作员权限。
 - BinPermIndex2 是变量的逻辑 0 设置对应的操作员权限。
- 如果不希望将变量连接到 "PermissionTag" 属性，则必须执行以下操作：
 - 将 "PermissionTag" 中的值设置为从 -1 到 +1。
 - 通过 "BinPermBitIndex1 .. 2" 对 Bit0 寻址

6.4.6.3 使用文本框显示二进制值

步骤

1. 从模板画面中复制 **APL_MULTI_TEXT2** (@PCS7ElementsAPL.pdl; Status display text; Manual) 对象，并将其插入面板视图中。
2. 指定一个有意义的对象名称。

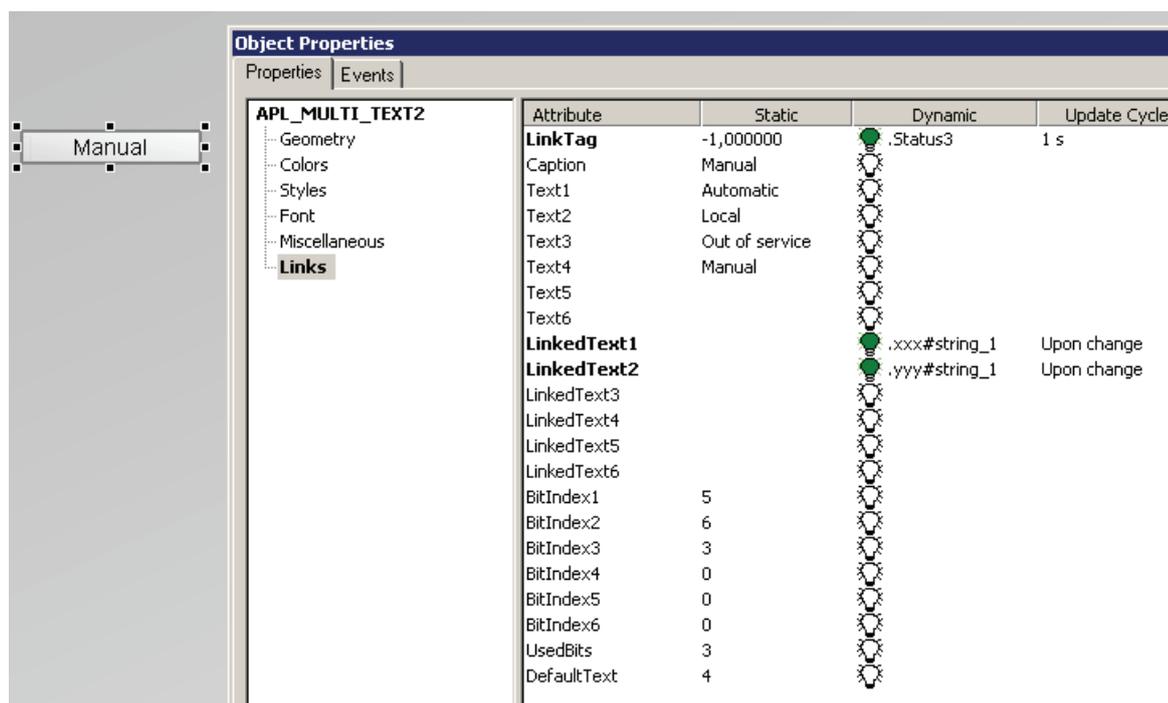


图 6-7 MultiText2_1

文本显示相关属性的简要说明。

随后将给出详细说明。

属性	简要说明
Links/ LinkTag	应用状态字
Links/ Caption	实际显示的文本
Links/ Text1 .. 6	面板中作为替代的参数化文本
Links/ LinkedText1..6	作为替代的实例文本（仅适用于特定版本）
Links/ BitIndex1..6	状态字中用于替代文本的位的分配
Links/ UsedBits	从 BitIndex1 开始已使用位的位数

6.4 创建面板

属性	简要说明
Links/ DefaultText	未对任何位进行置位时的默认值
Colors/ State1BackColor	背景色 Text1
Colors/ State1FillColor	Text1 的背景填充图案
Colors/ State1FontColor	Text1 的文本颜色

此对象的基本设计意图是让用户可根据状态字中的位在文本显示中显示最多六种不同的文本。

属性	简要说明
LinkTag	将相应的状态字与 "LinkTag" 属性互连。
BitIndex1 ...BitIndex6	通过 "BitIndex1 ..6" 属性参数化状态字中显示“Text1”到“Text6”的相应位（从 Bit0 开始）。
LinkedText1... LinkedText6	如果将已组态文本的“string0/string1”类型字符串与 "LinkedText1 ... 6" 属性互连，则将显示此实例特定的文本。 通过对象 APL_MULTI_TEXT6 和 APL_MULTI_TEXT 提供。
UsedBits	"UsedBits" 属性用于指定替代文本显示的数量。可从 BitIndex1 开始确定状态字中的已用位。
DefaultText	使用属性 "DefaultText" 确定所寻址的位都不为 1 时显示的文本。允许值为 1 到 6。 示例： "DefaultText" = 4 以及 "UsedBits" = 3 如果通过 BitIndex1 .. 3 寻址的位都不为 1，则将显示通过 Text4 组态的文本。

在“Colors”选项卡中组态各个文本的颜色。

APL_MULTI_TEXT2	Attribute	Static
Geometry	BackColor	
Colors	FillColor	
Styles	FontColor	
Font	BorderColor	
Miscellaneous	State1BackColor	
Links	State1FillColor	
	State1FontColor	
	State2BackColor	
	State2FillColor	
	State2FontColor	
	State3BackColor	
	State3FillColor	
	State3FontColor	
	State4BackColor	
	State4FillColor	
	State4FontColor	
	State5BackColor	
	State5FillColor	
	State5FontColor	
	State6BackColor	
	State6FillColor	
	State6FontColor	

附加文本显示

- **APL_MULTI_TEXT6**

可显示互连变量。必要时，将修改显示文本的字体大小。如果互连变量为空，则使用标准文本。

- **APL_MULTI_TEXT**

与 **APL_MULTI_TEXT6** 相似，但还会增加一个外部对象以适应此位置的显示文本。

例如，外部对象可以是与文本一同显示的状态显示。

可在 **ArrowDisplay** 选项卡的“**ArrowObjectName**”属性中组态外部对象的对象名称。

- 需要考虑的是，如果应用了此对象，则必须存在已组态的外部对象，否则便会在处理脚本的过程中出现错误。
- 对于互连的文本，外部对象将切换为不可见。

- **APL_MULTI_TEXT4**

此显示是透明的，即也可不显示任何文本。

- **APL_MULTI_TEXT3**

此显示是透明的，即也可不显示任何文本。不同的版本都以同一种颜色显示，这意味着运行时间得到了优化。

6.4 创建面板

参见

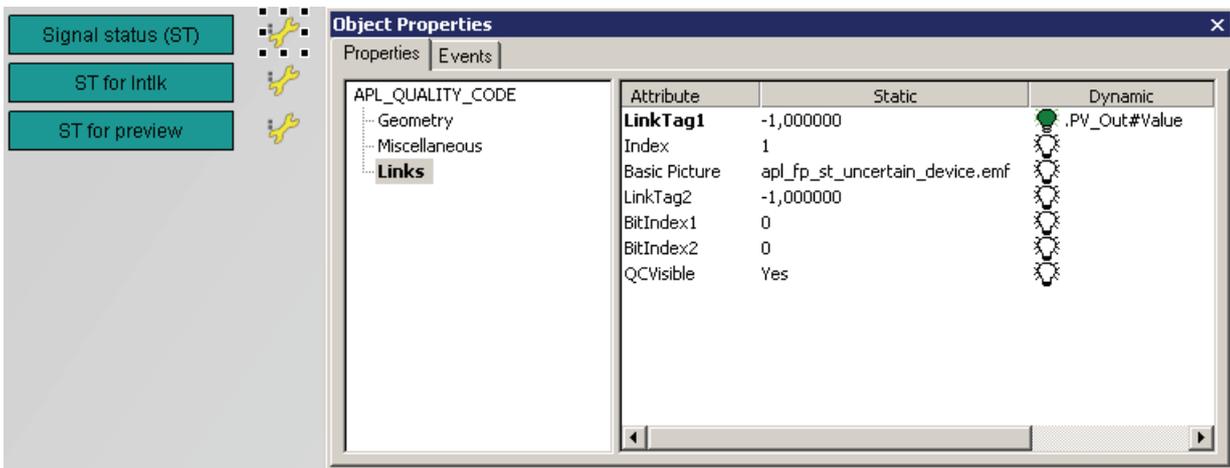
组态操作模式的显示和切换 (页 29)

6.4.6.4 信号状态显示

以下部分介绍了信号状态显示：

步骤

1. 从模板画面中复制 APL_QUALITY_CODE<x>(@PCS7ElementsAPL.pdl; Symbol status display) 对象。将该对象插入面板视图中。
2. 指定一个有意义的对象名称。



3. 互连要通过“LinkTag1”属性显示其信号状态的变量，并将对象放置在视图中。

模板画面中的面板

模板画面包含以下用于显示信号状态的面板：

- **APL_QUALITY_CODE1**

互连要通过“LinkTag1”属性显示其 Signal Status 的变量，并将对象放置在视图中。

- **APL_QUALITY_CODE2**

此 Signal Status 显示主要适用于互锁块，因为在此处 Signal Status 16#60 被显示为蓝色背景中的白色 B。

互连要通过“LinkTag1”属性显示其 Signal Status 的变量，并将对象放置到视图中。

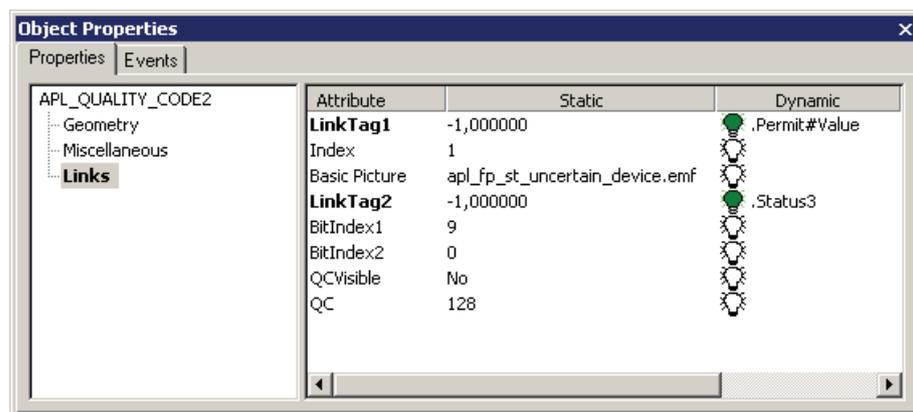
此外，还可通过此版本指定一般情况下是否显示符号。

应用示例：

如果未互连任何输入，则不显示任何内容。

– “LinkTag2”属性用于互连状态字。

– 使用“BitIndex1”寻址状态字中的相应位。



- **APL_QUALITY_CODE3**

与 APL_QUALITY_CODE2 显示类似，但在 Signal Status 16#60 位置会显示黄色的手掌图标。

6.4.6.5 显示 OS_Perm 操作锁

使用以下对象来显示对象的操作员控制使能端状态：

- 绿色勾号 -> 操作由 AS 块启用。
- 灰色勾号 -> 因过程所限而无法执行操作
- 红色叉号 -> 操作由 AS 块禁用。

6.4 创建面板

步骤

1. 从模板画面复制“StatusOSPerm_1”对象。 将对象插入面板视图中。
该对象是扩展状态显示。
2. 指定一个有意义的对象名称。

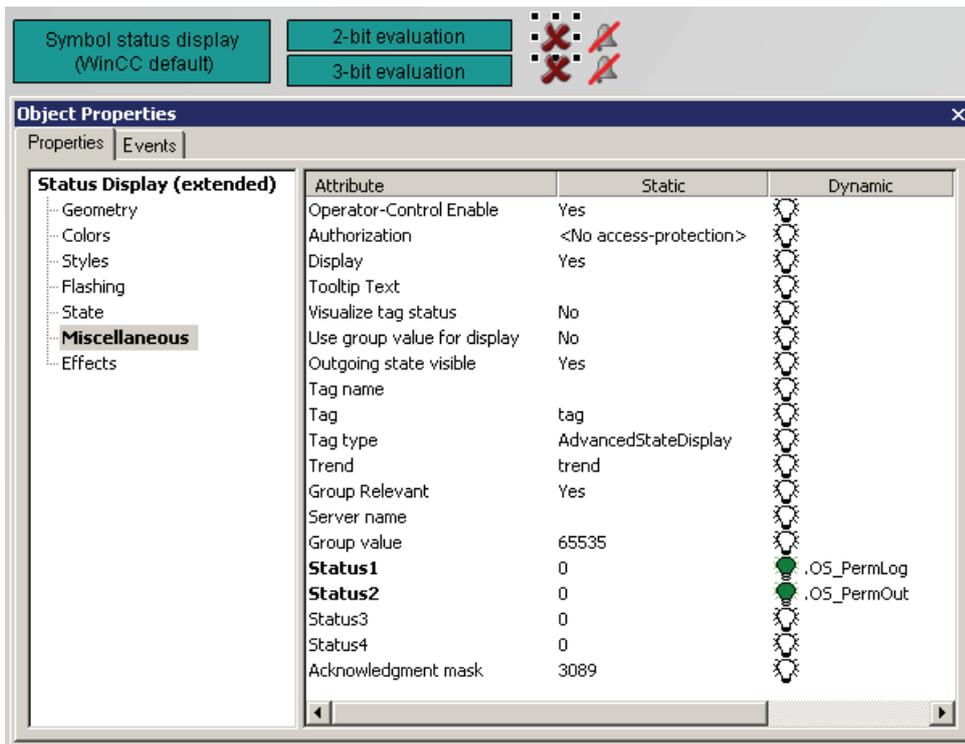


图 6-8 OS_Perm5

3. 将以下属性与参数互连（见图“OS_Perm5”）：
 - “Status1”属性与“OS_PermLog”参数
 - “Status2”属性与“OS_PermOut”参数
4. 在扩展状态显示的组态对话框中，将参数分配给“OS_PermLog”和“OS_PermOut”的相应位（见图“OS_Perm6”）。

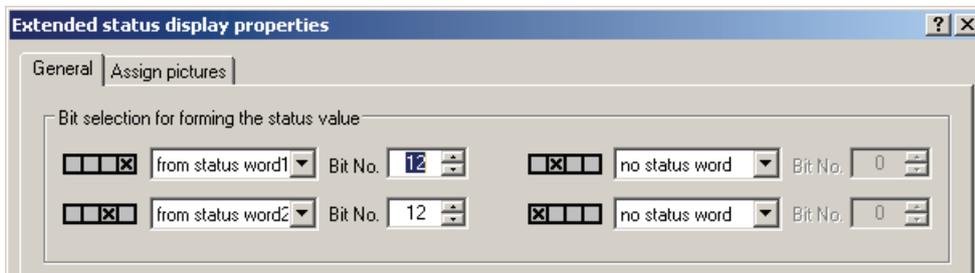


图 6-9 OS_Perm6

“OS_Perm7”示例

显示“StatusOSPerm_2”对象时，可将整个显示切换为不可见（见图“OS_Perm7”）。为此，应将状态字中的相应位互连到“Status3”并在扩展状态显示的组态对话框中对该位进行组态。

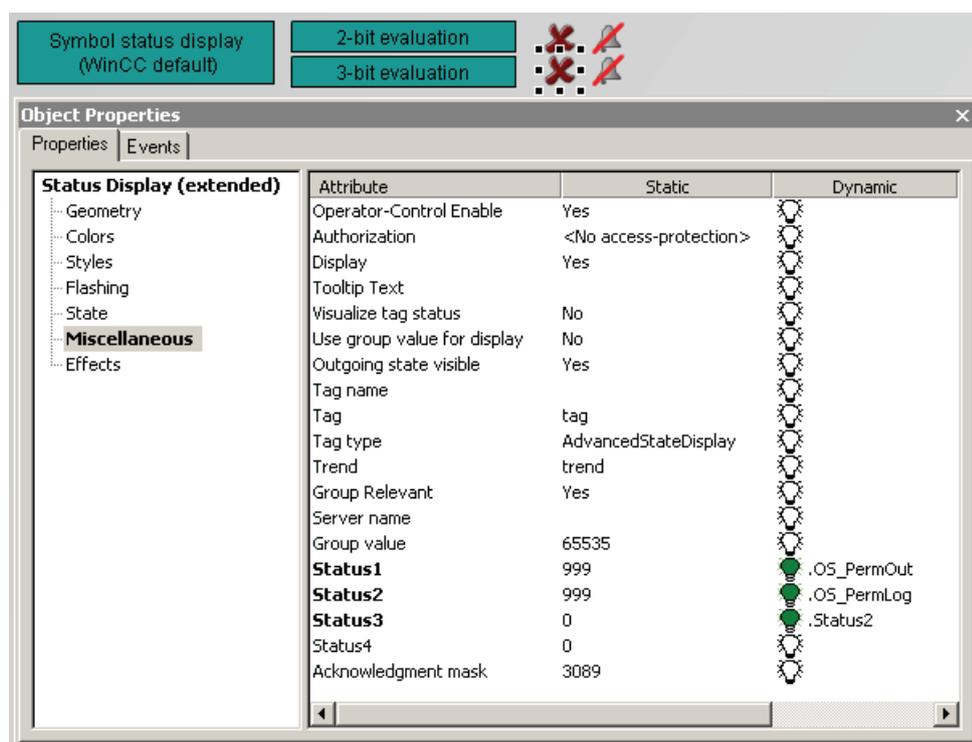


图 6-10 OS_Perm7

6.4.7 创建自己的位图和状态显示

已经使用 Visio 2007 创建了 APL 面板和块图标的位图或 EMF 文件，并以 Visio VSD 格式将它们存储为数据源。

6.4 创建面板

文件和格式

将 VSD 文件存储为可在 PCS 7 中使用的“EMF+”文件。

- 该“EMF+”格式提供以下功能：
 - 改进的 3D 视图
 - 改进的颜色梯度
 - 允许修改块图标的大小
- 在 Windows 中，无法区分“EMF+”文件与 EMF 文件 (*.emf)。
- 如果没有原始 VSD 文件，则无法使用 Visio 2007 修改存储成 EMF 文件的文件。

6.4.8 特殊要求的解决方案

6.4.8.1 在 V7 块图标的 ToolTipText 中显示块注释

默认情况下，tagname 显示为 APL 块图标版本 7 的 ToolTipText。

块注释可显示为 ToolTipText。下面描述了需要进行的更改。

在示例中，我们使用了 @PCS7TypicalsAPLV7.PDL (MonAnI - with Header) 模板画面中的 APL 块图标的副本 "MonAnI5"。

步骤

1. 选择块图标。
2. 使用快捷菜单（鼠标右键）打开组态对话框。
3. 在 Selected Properties 列表中选择 "Systems/ToolTipText" 属性。
4. 删除从属对象中的 "IOTagname.OutputValue"。
5. 在“对象”(Objects) 列表中，选择用户对象（“ToolTipText2”画面中的“MonAnI5”对象）。

- 从“属性”(Properties) 列表中将“工具提示文本”(Tooltip Text) 条目拖放到左侧列表“Selected Properties”的“ToolTipText”属性中。

结果如下图所示。

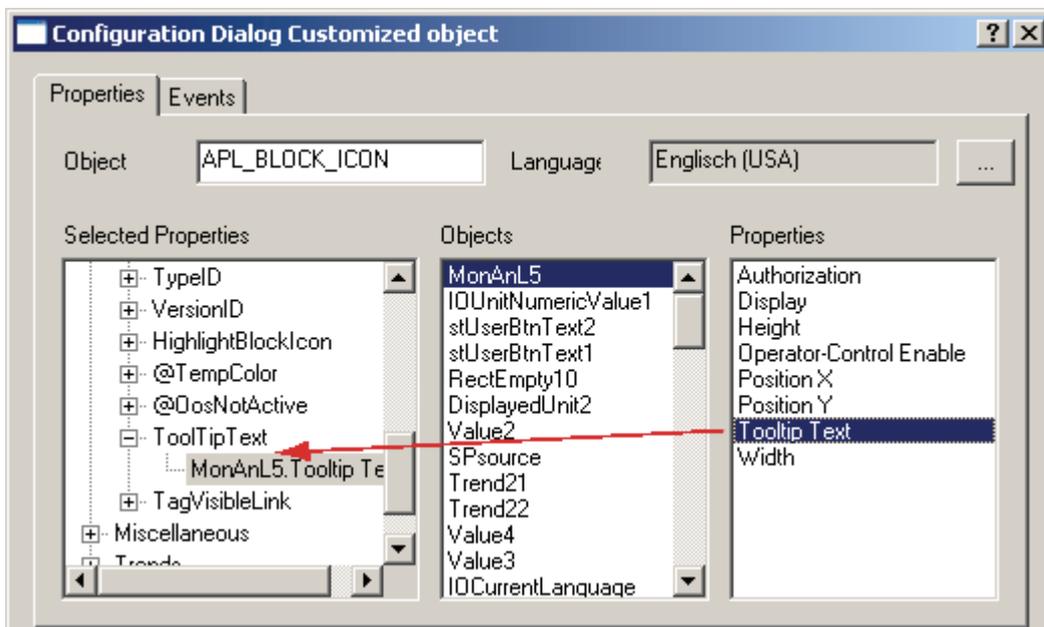


图 6-11 ToolTipText2

- 单击“确定”(OK)。
- 在块图标的属性中组态以下内容：
将“系统”(Systems) 文件夹中“ToolTipText”属性的文本“.#comment”组态为动态显示
请参见图“ToolTipText3”

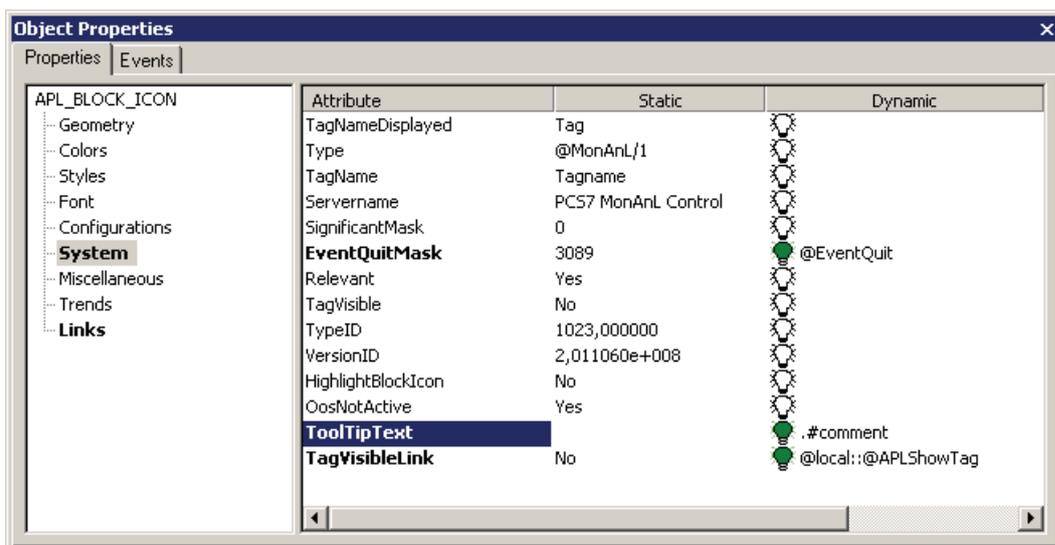


图 6-12 ToolTipText3

6.4 创建面板

6.4.8.2 多实例面板

多实例面板是用于访问 CFC 图表中不同块实例的一种面板。

设置标准库多实例面板的步骤

在标准库中，@Faceplate 对象具有 MULTI_INSTANCE 属性，该属性被设置为 TRUE。在这种情况下，块名称随后会在 OS 变量名中被截断。在多实例面板中，必须在所有变量连接的参数名称中将相应块名称补充完整。

例如，/Controller.PV_IN. 替代标准面板中的 .PV_IN

APL 不支持这种类型的多实例面板技术。

设置 APL 多实例面板的步骤

以下方法适用于 APL 面板。也可以将该方法用于标准库。

要求

- 想要寻址的块位于同一个 CFC 图表中。
- 只有一个具有消息的工艺块，并且其它块无需显示消息。
- 其它块的块名称包含具有消息的工艺块的名称 + 扩展名（请参见下面的示例图“MultiInstanceFP_1”）。

步骤

1. 在同一个 CFC 图表中创建待寻址的所有块。
2. 根据具有消息的块的名称形成其它块的名称。 该名称包含扩展名。

示例

示例图 MultiInstanceFP_1 包含以下块：

- 具有消息的块：
 - 电机块，“MOT_”。
- 不含消息的块：
 - 运行时间计数器，“MOT_OH”
 - 开关周期计数器，“MOT_SC”

在多实例面板中显示消息

多实例面板无法显示其它块的内部消息。 举例来说，这意味着无法在多实例面板中显示运行时间计数器和开关周期计数器中存在的任何消息。

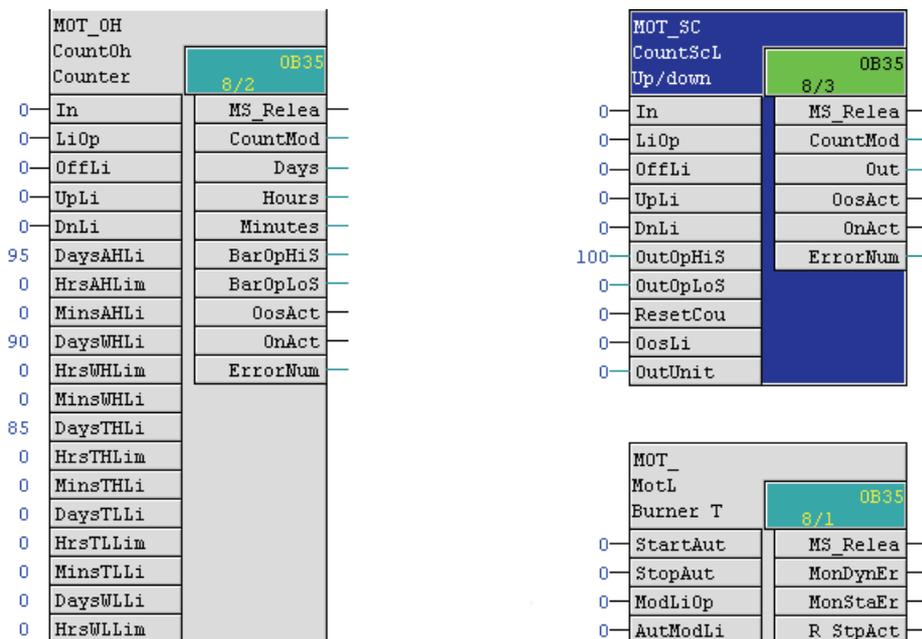


图 6-13 MultInstanceFP_1

连接面板中的变量

- 对于电机：
不需要对变量连接进行更改。
- 对于运行时间计数器和开关周期计数器：
必须在面板中的变量连接上添加相应的扩展名。

示例

OH.days 或 SC.Out, 另请参见 MultInstanceFP_2 图形。

6.4 创建面板

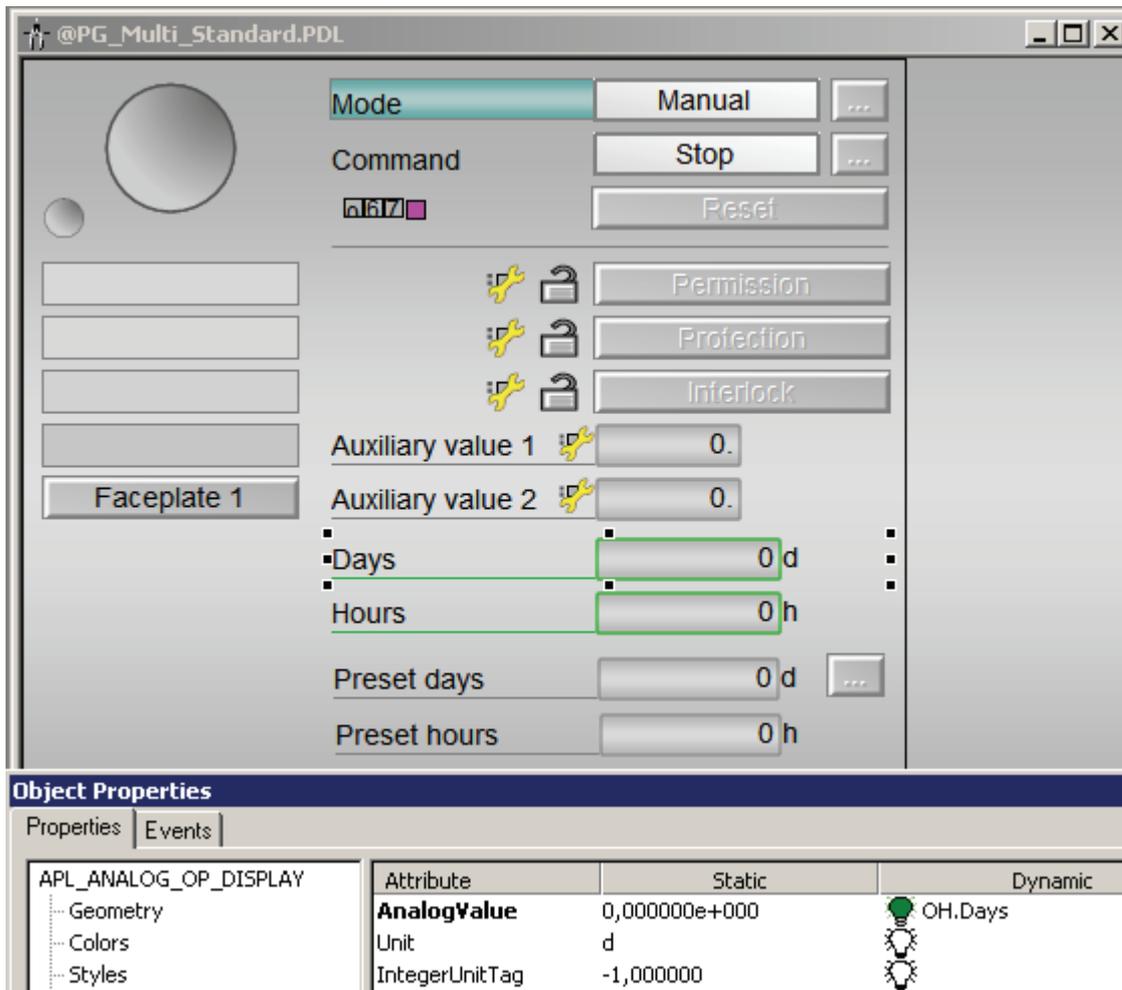


图 6-14 MultInstanceFP_2

A.1 对象类型的类型 ID

概述

对象类型	TypeID
@Faceplate	1
APL_ANA_EMPTY	21
APL_ANA_EMPTY2	26
APL_ANA_EMPTY3	25
APL_ANALOG_OP_DISPLAY	20
APL_ANALOG_OP_DISPLAY2	24
APL_ANALOG_OP_DISPLAY3	27
APL_ANALOG_OP_DISPLAY4	28
APL_BAR_GRADIENT_1	74
APL_BAR_HORIZ_1	72
APL_BAR_HORIZ_2	73
APL_BAR_LINE	84
APL_BAR_LINE_TEXT_E	82
APL_BAR_LINE_TEXT_R	83
APL_BAR_TEXT_E	80
APL_BAR_TEXT_R	81
APL_BAR_VERTIC_1	71
APL_BAR_VERTIC_2	70
APL_BAR_VERTIC_3	75
APL_BLOCK_ICON	3
APL_CHECKBOX	40
APL_FACEPLATE	50

A.1 对象类型的类型 ID

对象类型	TypeID
APL_FACEPLATE2	51
APL_FACEPLATE3	52
APL_INTLK_DISPLAY	100
APL_INTLK_FIRSTIN	102
APL_INTLK_OUTPUT	103
APL_INTLK_OUTPUT2	104
APL_INTLK_STATUS	101
APL_LOCK_SYMBOL	60
APL_MULTI_TEXT	10
APL_MULTI_TEXT10	19
APL_MULTI_TEXT11	120
APL_MULTI_TEXT2	11
APL_MULTI_TEXT3	12
APL_MULTI_TEXT4	13
APL_MULTI_TEXT5	14
APL_MULTI_TEXT6	15
APL_MULTI_TEXT7	16
APL_MULTI_TEXT8	17
APL_MULTI_TEXT9	18
APL_OBJ_COLLECTION	2
APL_OP_BUTTON	30
APL_OP_BUTTON2	31
APL_OP_BUTTON3	32
APL_OP_BUTTON4	33
APL_OP_BUTTON5	34
APL_OP_OBJECT	5
APL_OP_PERMISSION	6
APL_PIN_BUTTON	4
APL_QUALITY_CODE	61

对象类型	TypeID
APL_QUALITY_CODE2	62
APL_QUALITY_CODE3	63
APL_RANGE_DIFF	91
APL_SYMBOL_MOTL	114
APL_SYMBOL_MOTREVL	111
APL_SYMBOL_MOTSPDCL	112
APL_SYMBOL_MOTSPDL	113
APL_SYMBOL_VLVMOTL	110
APL_TIME_OP_DISPLAY	22
APL_TIME_OP_DISPLAY2	23
APL_TIME_OP_DISPLAY3	29
APL_TIME_RANGE	90
APL_TIME_RANGE2	92

A.1 对象类型的类型 ID