

InTouch[®] HMI 可视化指南

Invensys Systems, Inc.

修订版 A

上次修订日期：2007 年 8 月 6 日



版权声明

© 2007 Invensys Systems, Inc. 版权所有。保留所有权利。

保留所有权利。未经 Invensys Systems, Inc. 事先书面明确同意，不得通过任何手段（电子、机械、影印、录制或其它方式）复制、传输本文档中的任何部分，或是将其存储到检索系统。使用本文档所含信息不需承担任何相关的版权或专利责任。虽然在编制本文档的过程中已采取一切预防措施，但错误或疏漏在所难免，出版商与作者对此概不承担任何责任。对由于使用本文档所含信息而导致的任何损害，亦不承担任何赔偿责任。

本文档中的内容如有变更，恕不另行通知，这些内容亦不代表 Invensys Systems, Inc. 一方的承诺。本文所述软件系在遵守许可协议或保密协议的前提下提供。本软件的使用或复制必须遵守这些协议中的各项条款。

Invensys Systems, Inc.
26561 Rancho Parkway South
Lake Forest, CA 92630 U.S.A.
(949) 727-3200

<http://www.wonderware.com>

对产品文档如有任何意见或建议，请发送电子邮件到 productdocs@wonderware.com。

商标

本文所提及且已知为商标或服务标志的所有专用名词均已采用适当的首字母大写形式。Invensys Systems, Inc. 无法证实此类信息的准确性。在本文档中使用某个专用名词不应视为会影响任何商标或服务标志的有效性。

Alarm Logger、ActiveFactory、ArchestrA、Avantis、DBDump、DBLoad、DT Analyst、FactoryFocus、FactoryOffice、FactorySuite、FactorySuite A²、InBatch、InControl、IndustrialRAD、IndustrialSQL Server、InTouch、MaintenanceSuite、MuniSuite、QI Analyst、SCADAAlarm、SCADASuite、SuiteLink、SuiteVoyager、WindowMaker、WindowViewer、Wonderware 以及 Wonderware Logger 均为 Invensys plc 及其子公司与附属公司的商标。所有其它品牌可能是其相应所有者的商标。

目录

欢迎.....	9
文档惯例.....	9
技术支持.....	10
第 1 章 WindowMaker: 您的开发环境.....	11
设置 WindowMaker 首选项.....	12
使用屏幕网格与标尺.....	15
将对象对齐到网格	15
使用标尺.....	15
平移与缩放	16
使用略图窗口进行平移和缩放	16
使用鼠标滚轮进行缩放与平移	17
平移与缩放限制.....	17
管理工具栏	18
使用应用程序浏览器.....	19
在应用程序浏览器中浏览	20
将应用程序添加到应用程序浏览器	20
使用调色板	21
打开调色板	21
创建自定义颜色	22
导入与导出自定义颜色	23
设置字体缺省值.....	24

使用全屏幕模式	24
窗口对话框的功能	24
鼠标快捷方式	26
使用方向键移动对象	27
查看许可证信息	27
第 2 章 应用程序窗口	29
创建应用程序窗口	30
导入与导出窗口	31
设置要在运行时出现的窗口	32
修改应用程序窗口	32
打开、保存以及关闭窗口	33
创建窗口的副本	34
删除窗口	34
第 3 章 WindowMaker 对象	35
简单对象	36
创建线条与图案	36
创建按钮	37
创建多边线与多边形	37
创建文本	37
复杂对象	38
单元与符号	39
将对象组合成单元	39
将对象组合成符号	40
共同的操作	40
选择对象	41
移动对象	42
对齐对象	43
层叠对象	43
控制水平与垂直间距	44
翻转对象与单元	44
调整对象大小	45
旋转对象	45
更改文本外观	46
更改线条与轮廓	47
更改填充	47

删除对象.....	48
撤消更改.....	48
所有对象的特殊操作.....	49
剪切、复制以及粘贴对象	49
剪切、复制以及粘贴对象链接	49
创建对象的副本	50
特殊对象的特殊操作.....	51
调整多边线与多边形对象的形状.....	51
使用位图容器	52
定义位图透明度.....	53
更改圆角长方形的圆角半径.....	54
替换对象文本	54

第 4 章 设置对象动画效果..... 55

两种类型的动画链接.....	56
数据显示动画	56
创建值显示	56
创建移动.....	58
创建旋转.....	59
设置大小动画效果	61
设置颜色动画效果	63
设置填充级别动画效果	66
使对象闪烁.....	68
启用可见性	69
使对象失效	70
配置工具提示	71
放置触控窗口	72
数据输入动画	75
启用离散输入	76
启用模拟输入	77
启用字符串输入.....	78
启用游标.....	79
启用按钮.....	81
打开与关闭窗口	82
配置屏幕键盘	83
常见动画任务	88
选择标记或属性.....	88
创建键盘快捷键.....	93

更改标记名引用	94
转换占位符标记	95

第 5 章 向导97

使用向导	98
向导类型	98
将向导添加到工具栏	99
粘贴向导实例	99
配置向导	100
对向导执行标准操作	100
安装与删除向导	100
趋势对象	102
Windows 控件向导	103
创建与配置 Windows 控件	103
创建文本框	104
创建列表框	105
创建组合框	106
创建复选框	107
创建单选按钮组	107
编写 Windows 控件脚本	109
获取或设置控件值	109
启用或禁用用户输入控件	111
动态隐藏 Windows 控件	112
对组合框中的项目进行添加与删除	113
从文件中加载列表项或将列表项保存到文件	116
在组合框或列表中查找项目	119
在组合框或列表中使用项目索引	120
统计列表框或组合框项目数	122
获取或设置列表项的值	123
获取列表项的名称	125
加载文本框的内容	126
检查文本框是否为只读	128
获取或设置复选框的标签	129
理解 Windows 控件错误消息	130

第 6 章 ActiveX 控件.....	133
使用 ActiveX 控件.....	134
配置 ActiveX 控件.....	136
给 ActiveX 控件命名.....	137
ActiveX 控件上的标准操作	137
安装与删除 ActiveX 控件.....	138
 索引.....	 141

欢迎

本文介绍如何使用 **InTouch WindowMaker** 创建称为窗口的 **HMI** 应用程序屏幕。您可以在窗口中使用从简单线条到复杂符号的各种图形对象。在运行时，应用程序会根据您在图形对象与生产数据之间设置的链接呈现动画效果。

您可以联机查看本文，也可以使用 **Adobe Acrobat Reader** 的打印功能来打印本文的部分或全部内容。

本指南假设您了解如何使用 **Microsoft Windows**，包括浏览菜单、在应用程序之间切换，以及在屏幕上移动对象。如需有关这些任务的帮助，请参阅 **Microsoft** 联机帮助。

文档惯例

本文采用以下惯例：

惯例	用于
首字母大写	路径与文件名。
粗体	菜单、命令、对话框名称以及对话框选项。
等宽字体	代码范例与显示文本。

技术支持

Wonderware 的“技术支持”部门提供多种技术支持方案，帮助解答有关 Wonderware 产品及其实施方案的任何疑问。

在与“技术支持”部门联系之前，请参阅本文中相关的章节，以寻求问题的可能解决方案。如果需要联系技术支持以获取帮助，请准备好以下信息：

- 使用的操作系统的类型与版本。
- 有关如何重现问题的详细说明。
- 看到的错误消息的准确内容。
- Log Viewer 或任何其它诊断应用程序提供的任何相关输出列表。
- 为解决问题所作的尝试及其结果的详细说明。
- 如果遇到仍然存在的已知问题，请提供指定给该问题的“Wonderware 技术支持”案例号。

第 1 章

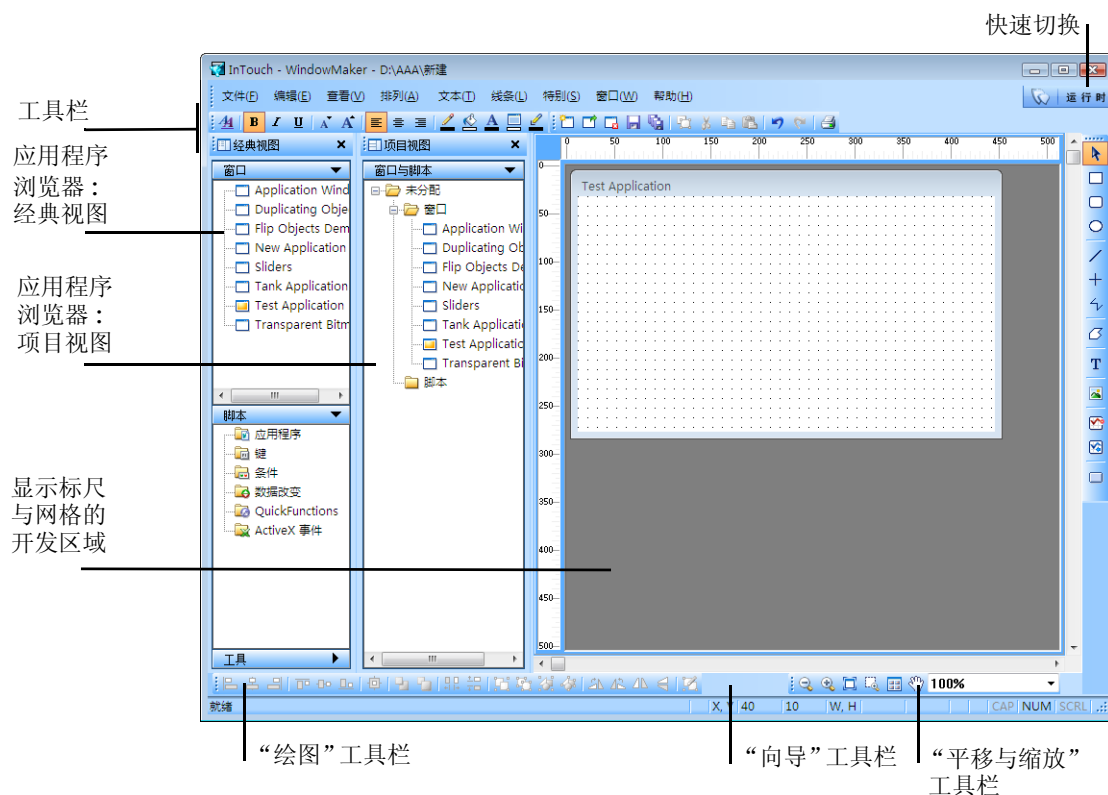
WindowMaker: 您的开发环境

WindowMaker 是用于创建 InTouch 应用程序的开发环境。

该开发环境的主要部分包括:

- 工具栏与状态栏
- 经典视图, 它显示窗口与脚本。
- 项目视图, 它按照另一种格式显示窗口与脚本。
- 开发区域
- 快速切换按钮, 用于打开 WindowViewer

下图显示这个环境：



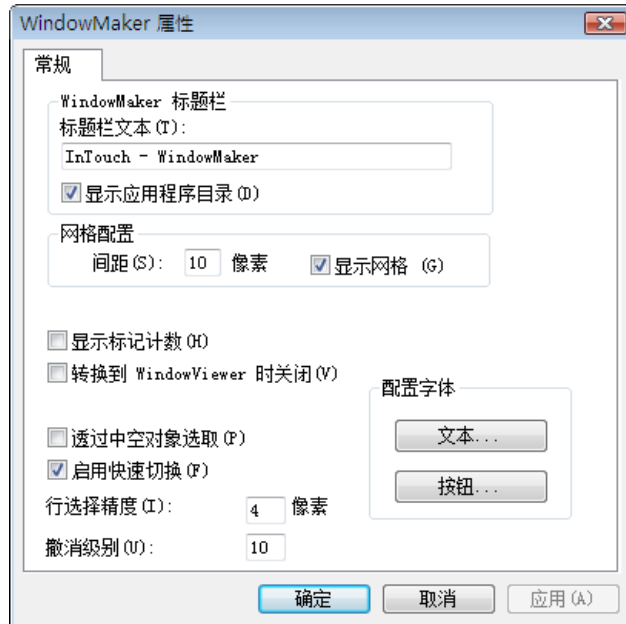
设置 WindowMaker 首选项

通过使用“WindowMaker 属性”对话框，可以配置会影响 WindowMaker 行为的首选项与选项。您可以：

- 更改标题栏文本。
- 显示网格或关闭网格。
- 更改网格上像素之间的间距。
- 显示标记计数。
- 更改文本与按钮的缺省字体。
- 设置行选择的精度。
- 设置某个选项以便在切换到 WindowViewer 时关闭 WindowMaker。
- 设置某个选项以透过中空对象进行选取。
- 启用从 WindowMaker 快速切换到 WindowViewer 的功能。
- 设置撤销级数。

要设置 WindowMaker 的属性

- 1 在**特别**菜单上，指向**配置**，然后单击**WindowMaker**。此时出现**WindowMaker 属性**对话框。



- 2 在**WindowMaker 标题栏**区域中，配置标题栏的外观。执行以下任何操作：
 - 在**标题栏文本**框中，输入要在设计期间显示在标题栏中的文本。
 - 选择**显示应用程序目录**复选框，以便在标题栏中包含应用程序文件夹的路径。
- 3 在**网格配置**区域中，配置背景网格。执行以下任何操作：
 - 在**间距**框中，输入网格坐标之间的像素数量。
 - 选择**显示网格**复选框以显示网格。
- 4 配置窗口的其它属性 执行以下任何操作：
 - 选择**显示标记计数**复选框，以便在菜单栏中显示“标记名字典”中的标记名数量。如果有许多标记，则显示标记计数时，可能会影响“标记名字典”的性能。
如果使用大小受限的“标记名字典”创建应用程序，此项功能会非常有用。标记名计数不包括远程标记名引用或系统标记。单击**特别**菜单上的**更新使用计数**，可以找出远程标记名引用的使用情况。
 - 选择**转换到 WindowViewer 时关闭**复选框，以便在启动 WindowViewer 时自动关闭 WindowMaker。
此选项旨在节约有限的内存。如果内存足够，并且经常 在 WindowViewer 与 WindowMaker 之间切换，请不要选择此选项。

选择**转换到 WindowViewer 时关闭**时，同时会选择 **WindowViewer 属性**对话框中**常规**属性选项卡上对应的**关闭 WindowViewer** 命令。

- 选择**透过中空对象选取**复选框，以便选择中空对象后面的对象。

这样可达到类似这样的目的：不必将框架置后，便能选择框架内的对象。

- 选择**启用快速切换**复选框，以使用“快速切换”在 WindowMaker 与 WindowViewer 之间切换。

快速切换是出现在 WindowMaker 右上角的**运行时**字样。在 WindowViewer 中，它是**开发**字样。

使用快速切换时，在切换到 WindowViewer 之前，WindowMaker 自动保存对所有打开的窗口所作的全部更改。

- 在**行选择精度**框中，输入光标可以远离某行多少个像素但仍能选中它。

在大多数情况下，缺省设置 4 的效果就很好。

- 在**撤消级别**框中，输入要保留的**撤消**与**恢复**级数。

最多可以有 25 级。如果输入零，则会关闭撤消 / 恢复功能。

一个级别代表一个操作。创建新的窗口或是打开现有的窗口时，**撤消**与**恢复**堆栈都是空的。关闭窗口时，两个堆栈都会清空。

- 5 在**配置字体**区域中，单击**文本**或**按钮**，以设置缺省的文本字体或按钮字体。选择缺省字体，然后单击**确定**。

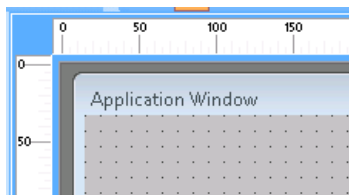
通过使用**字体**工具栏，可以覆盖任何窗口中的这些缺省值。

- 6 单击**确定**。

- 7 重新启动 WindowMaker 以应用所作的任何更改。

使用屏幕网格与标尺

您可以显示网格与标尺来帮助排列与对齐对象。



将对象对齐到网格

通过选择**对齐网格**，可以将对象对齐到预定义的网格点。

缺省条件下，网格设置为 10 个像素，并且设置为在启动 WindowMaker 时可见。网格的像素间隔可以在 **WindowMaker 属性**对话框中进行配置。

要查看网格，必须选择 **WindowMaker 属性**对话框中的**显示网格**，并且选择**排列**菜单上的**对齐网格**。

要配置网格

- 1 在**特别**菜单上，指向**配置**，然后单击 **WindowMaker**。此时出现 **WindowMaker 属性**对话框。
- 2 在**间距**框中，输入坐标之间间隔的像素数量。
- 3 如果希望在选择**对齐网格**时看到网格，请选择**显示网格**复选框。
如果没有选择**显示网格**，则在选择**对齐网格**时，窗口中看不见网格。

使用标尺

使用标尺帮助精确对齐窗口中的对象。标尺沿着开发环境窗口的顶部和一侧显示。

要显示或隐藏标尺

- 1 在**查看**菜单上，单击**标尺**。
- 2 重复此步骤隐藏标尺。

平移与缩放

通过放大与缩小，可以更仔细地查看正在编辑的元素，从而确保各个对象都能精确排列或是放在正确的位置上。

缺省条件下，“平移与缩放”工具栏出现在屏幕的右下角。同其它工具栏一样，它可以浮动或固定在其它位置上。

您可以：

- 在 100% 到 500% 的范围内进行放大与缩小。
- 使用橡皮圈工具缩放特定的区域。
- 按特定的百分比缩放窗口。
- 单击并拖动以平移窗口。
- 返回到正常的缺省视图。

要显示或隐藏“平移与缩放”工具栏

- ◆ 在**查看**菜单上，单击**平移与缩放**。

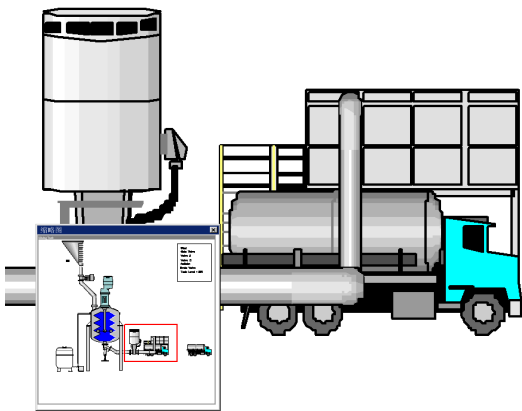
使用略图窗口进行平移和缩放

放大应用程序窗口的细部时，略图窗口显示该细部与整个应用程序之间的关系。

缩略图窗口让您既可以总览开发区域，又可以观察其细部。

通过单击**略图**按钮，可以显示或隐藏**缩略图**窗口。

红色长方形是窗口中缩放区域的边界。



- 拖动红色长方形可以显示窗口的不同部分。
- 单击窗口的不同区域可以将长方形移到该区域。
- 调整长方形的大小可以更改显示区域的缩放级别。

对于不能缩放的对象，如 ActiveX 控件，略图视图会显示白色的长方形框。

使用鼠标滚轮进行缩放与平移

如果鼠标带有滚轮，则通过按 CTRL 键并滚动滚轮，可以更改图像的缩放级别。



- 滚动鼠标滚轮时，每一格相当于按 20% 更改缩放级别。
- 将光标放到 InTouch 窗口中，然后按鼠标滚轮，也可以在窗口中浏览。按鼠标滚轮时，会出现带有四个箭头的图标。通过移动鼠标在窗口中浏览。

平移与缩放限制

平移与缩放操作不会应用于以下控件：

- ActiveX 控件
- 分布式报警对象
- 16 笔趋势
- SPC Pro 对象
- 文本框
- 复选框
- 列表框
- 组合框
- “单选按钮组”对象

如果在视图缩放到超过 100% 时，这些控件中的某一个位于可见区域中，则在该控件占据的区域中，会出现一个使用该控件名的长方形。

管理工具栏

任何工具栏都可以从**查看**菜单进行显示或隐藏。

任何工具栏都可以从它的缺省固定位置移到开发窗口中的任何其他位置。浮动工具栏有标题栏，而且大小也可以更改。

启动 WindowMaker 时，所有的工具栏都会显示出来。

您可以通过拖动工具栏来浮动或固定工具栏。显示原本隐藏的固定工具栏时，它会重新出现在上次它在窗口中所固定的位置上。

要显示或隐藏工具栏

- ◆ 在**查看**菜单上，单击工具栏名。

要更改浮动工具栏的大小

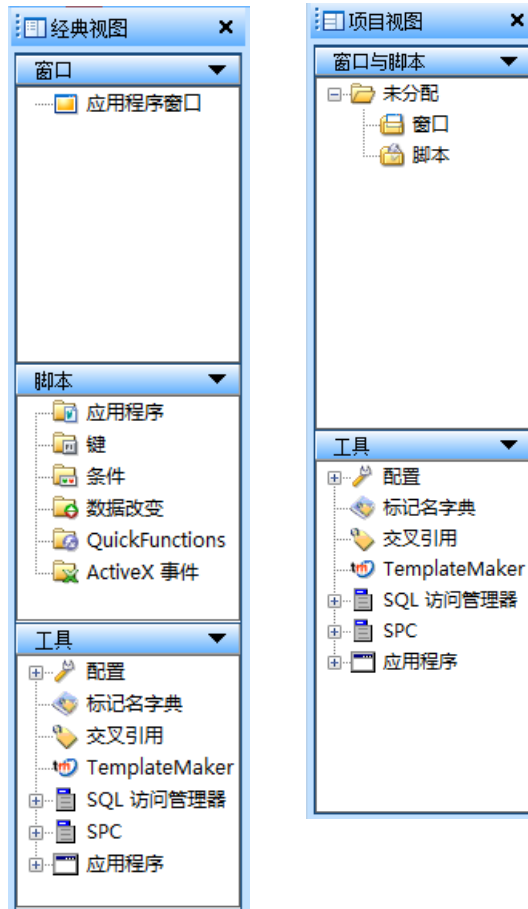
- 1 将光标移到工具栏的任何一个边缘。此时光标变成一个双向箭头。
- 2 拖动工具栏的边缘以调整工具栏的大小。
在移动光标的过程中会出现一个框，指出释放鼠标按钮时工具栏的大小。

要一次隐藏所有的工具栏

- ◆ 在**查看**菜单上，单击**全部隐藏**。

使用应用程序浏览器

“应用程序浏览器”有两种视图：**经典视图**与**项目视图**。这些视图显示 InTouch 应用程序窗口和可用的工具。和其它工具栏一样，它们也可以处于打开、关闭、固定或浮动状态。



通过这些视图，可以访问所有的应用程序窗口、脚本、配置菜单、“标记名字典”以及向导。

请勿将 WindowViewer (view.exe) 添加到**应用程序浏览器**。启动 WindowViewer 的正确方法是：单击**文件**菜单上的 **WindowViewer**，或是单击工具栏中的**运行时**进行快速切换。

在应用程序浏览器中浏览

您可以在两个“应用程序浏览器”工具栏的任何一个中展开或折叠文件夹。

应用程序视图显示其它已安装的应用程序。

要展开或折叠“应用程序浏览器”中的文件夹

- 1 双击文件夹或图标以展开并显示组成员。
- 2 双击成员以打开该成员。

将应用程序添加到应用程序浏览器

WindowMaker 的“应用程序浏览器”可以从 WindowMaker 中启动其它应用程序。例如，您可以一边开发应用程序，一边运行“I/O 服务器”并配置它。您也可以启动第三程序，如 Windows 的“记事本”、“写字板”、Microsoft Excel、Microsoft Word、Microsoft 的“画图”等等。

您还可以配置“应用程序浏览器”以打开特定的文件，如文档或电子表格。

要将新的应用程序添加到“应用程序浏览器”

- 1 在经典视图的工具窗格中，使用鼠标右键单击应用程序，然后单击**新建**。此时出现**应用程序属性**窗口。



- 2 在**名称**框中，输入应用程序的名称。
- 3 在**命令行**框中，输入应用程序的完整路径。单击省略号按钮可浏览到应用程序。
在**命令行**框中，可以给应用程序添加命令行参数。
- 4 在**启动方式**列表中，单击希望应用程序在启动后的出现方式。
- 5 单击**确定**。此时应用程序添加到**应用程序浏览器**中的“应用程序”下。现在您随时可以从 WindowMaker 中运行该应用程序。

使用调色板

通过使用调色板，可以将颜色应用于线条、长方形、圆角长方形、椭圆、多边形、多边形以及文本的静态与动态属性。此外，您也可以为窗口选择背景颜色，以及为位图选择透明色（以便能够查看位图后面的对象）。

调色板提供广泛的颜色供您选择，可选的颜色多达 1670 万种。可用的颜色数量可能会受视频卡功能的限制。

您也可以：

- 定义和添加自定义颜色。
- 导入在其它 Windows 应用程序中创建的自定义调色板。
- 将调色板导出到其它 Windows 应用程序。

打开调色板

每次单击对话框中的颜色方块，或单击某个颜色工具以便将线条、填充或文本颜色应用于所选的对象时，便会出现调色板。

要打开调色板

- 1 单击对话框中的颜色方块。此时出现**标准调色板**。



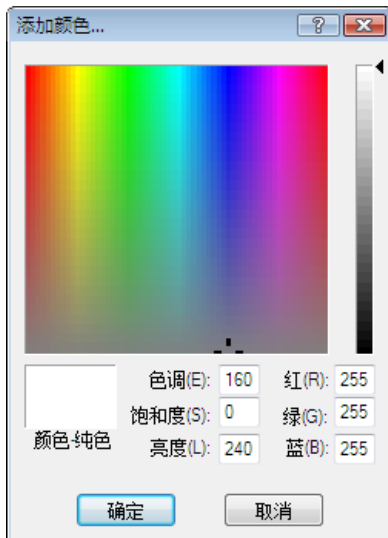
- 2 单击右箭头以访问传统颜色。
- 3 单击要使用的颜色。此时关闭调色板，并应用所选的颜色。

创建自定义颜色

您可以创建由自定义颜色构成的调色板。

要创建自定义颜色

- 1 打开调色板。
- 2 在自定义调色板区域中，使用鼠标右键单击其中的一个空白方块，然后单击**编辑自定义颜色**。此时出现**添加颜色**对话框。



- 3 执行以下任何操作：
 - 单击颜色显示区域中的任何位置并使用游标调整颜色。要指定不含白色或黑色成分的 100% 的颜色，请按 ALT + O。
 - 在红、绿以及蓝框中输入一些值，以定义一种颜色。您可以通过观察颜色表来对这些值进行试验。注意色调、饱和度以及亮度的值也会随之改变。
 - 在色调、饱和度以及亮度框中输入一些值，以定义一种颜色。更改这些值中的任何值时，红、绿、蓝的值也会相应改变。

“色调”是离散的颜色值，其中 0 是红色、60 是黄色、120 是绿色、180 是青色、200 是洋红色、240 是蓝色。

“饱和度”是指定的色调中的颜色数，最大值为 240。

“亮度”是颜色的明亮程度。

- 4 在**颜色 - 纯色**框中查看产生的颜色。

如果显示器设置为显示 256 色，则**颜色 - 纯色**框可能会显示两种颜色。右侧显示所选的颜色作为纯色显示时的效果。左侧显示经过抖动处理的颜色，即使用 256 色中的两种来显示特定颜色时的近似效果。

- 5 单击**确定**。

您也可以使用吸管工具来创建自定义颜色。

备注 使用此项功能创建透明位图。

要使用吸管工具选择自定义颜色

- 1 打开**调色板**。
- 2 使用鼠标右键单击调色板底部**自定义调色板**部分的某个空白方块。
- 3 单击吸管工具，然后单击希望添加的颜色。

导入与导出自定义颜色

如果定义了自定义的调色板，则可以从 InTouch 应用程序中导出它，然后将它导入另一个 InTouch 应用程序。

要导入自定义的调色板

- 1 打开**调色板**。
- 2 单击**自定义调色板**向下箭头。
- 3 单击**加载调色板**。此时出现标准的 Windows **打开**对话框。
- 4 找到并选择包含所需颜色定义的 .pal 调色板文件。
- 5 单击**打开**。此时调色板文件中包含的颜色都加载到**自定义调色板**中。

要导出自定义的调色板

- 1 打开**调色板**。
- 2 单击**自定义调色板**向下箭头。
- 3 单击**导出调色板**。此时出现标准的 Windows **另存为**对话框。
- 4 为调色板文件指定名称，然后单击**保存**。

设置字体缺省值

您可以为文本对象与带有文本的按钮对象设置字体缺省值。

提示 通过使用工具栏来自定义窗口或按钮文本，可以覆盖任何窗口中的这些缺省值。

要设置字体缺省值

- 1 在**特别菜单**上，指向**配置**，然后单击 **WindowMaker**。此时出现 **WindowMaker 属性**对话框。
- 2 在**配置字体**区域中，单击**文本**或**按钮**，以设置缺省的文本字体或按钮字体。选择缺省字体，然后单击**确定**。
通过使用**格式**工具栏中的**字体**工具，可以在任何窗口中覆盖这些缺省值。
- 3 单击**确定**。

使用全屏幕模式

全屏幕模式隐藏除了打开的窗口和浮动工具栏之外的所有程序元素。

要在打开或关闭全屏幕之间切换

- ◆ 在**查看**工具栏上，单击**全屏幕**按钮，以从正常模式切换到全屏幕模式。
此时**查看**工具栏自动更改为**恢复**工具栏，并浮动在顶部。

窗口对话框的功能

使用**文件**菜单打开、保存、关闭、删除窗口，或是创建窗口的副本时，适合所选命令的所有窗口的名称都出现在一个列表中。

提示 单击**详细资料**从列表视图更改为详细资料视图。详细资料视图包含在**窗口属性**对话框中给窗口所作的任何**注释**。

要以多列格式查看窗口与详细资料

1 单击**详细资料**按钮。此时出现一个对话框，显示窗口列表。



- 2 选择一个窗口，再次单击窗口以清除它。
- 要选择所有的窗口，请单击**全选**。
 - 要清除所有的窗口，请单击**全部清除**。
 - 要打开所选窗口，请单击**确定**。

鼠标快捷方式

以下快捷方式可用于打开对话框与执行其它常见任务。

要访问 WindowMaker 中项目的菜单命令

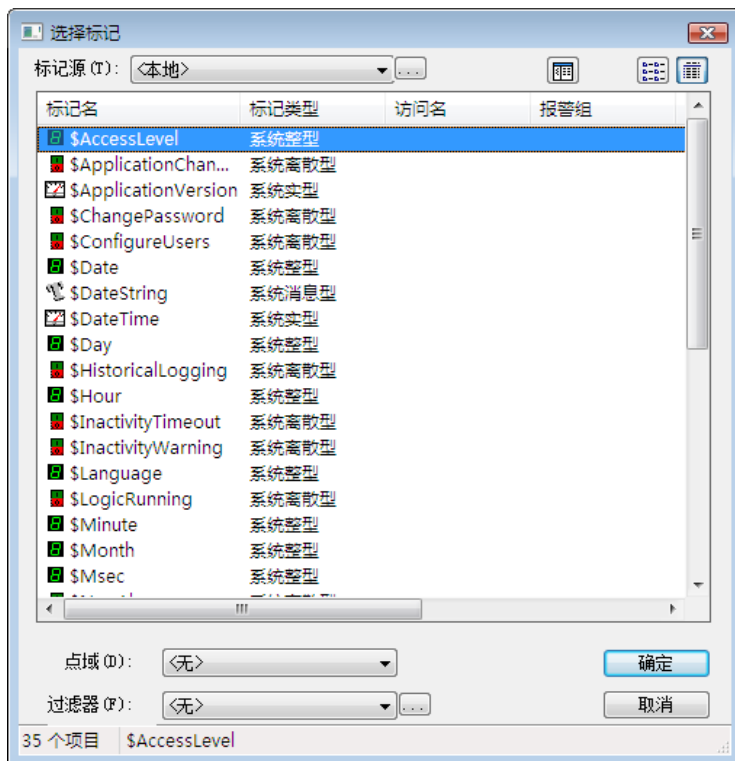
- ◆ 使用鼠标右键单击该项目。这些项目包括图形对象、视图中的文件夹名，等等。

要打开“动画链接”对话框

- ◆ 双击对象或符号。

要打开“选择标记”对话框（标记浏览器）

- ◆ 双击链接定义对话框中空白的表达式输入区域。此时出现选择标记对话框。



要访问标记点域

- ◆ 在任何“标记名”或“表达式”输入框中，输入标记名加上一个英文句点，然后双击英文句点的右侧。您也可以仅输入一个英文句点，然后双击它的右侧。此时出现选择域名对话框，其中显示所有的标记名点域。

要在“标记名字典”中打开标记名定义

- ◆ 双击标记名。

使用方向键移动对象

在 WindowMaker 中，可以使用**方向键**来移动一个或一组所选的对象。

使用方向键移动对象时，对象的移动距离取决于是否显示网格。

显示网格时，对象移动多少个像素取决于网格间距，此间距在 WindowMaker 属性对话框中设置。缺省设置是网格点之间相距十个像素。

显示网格时：

- 按方向键一次，对象移动一个网格点。
- 按 **SHIFT** + 方向键，对象移动两个网格点。
- 按 **CTRL** + 方向键，对象移动四个网格点。

不显示网格时：

- 按方向键一次，对象移动一个像素。
- 按 **SHIFT** + 方向键，对象移动 10 个像素。
- 按 **CTRL** + 方向键，对象移动 50 个像素。

查看许可证信息

您可以查看关于 InTouch 软件的信息，如：

- 版本号。
- 序列号。
- 许可证过期日期（如果适用）。

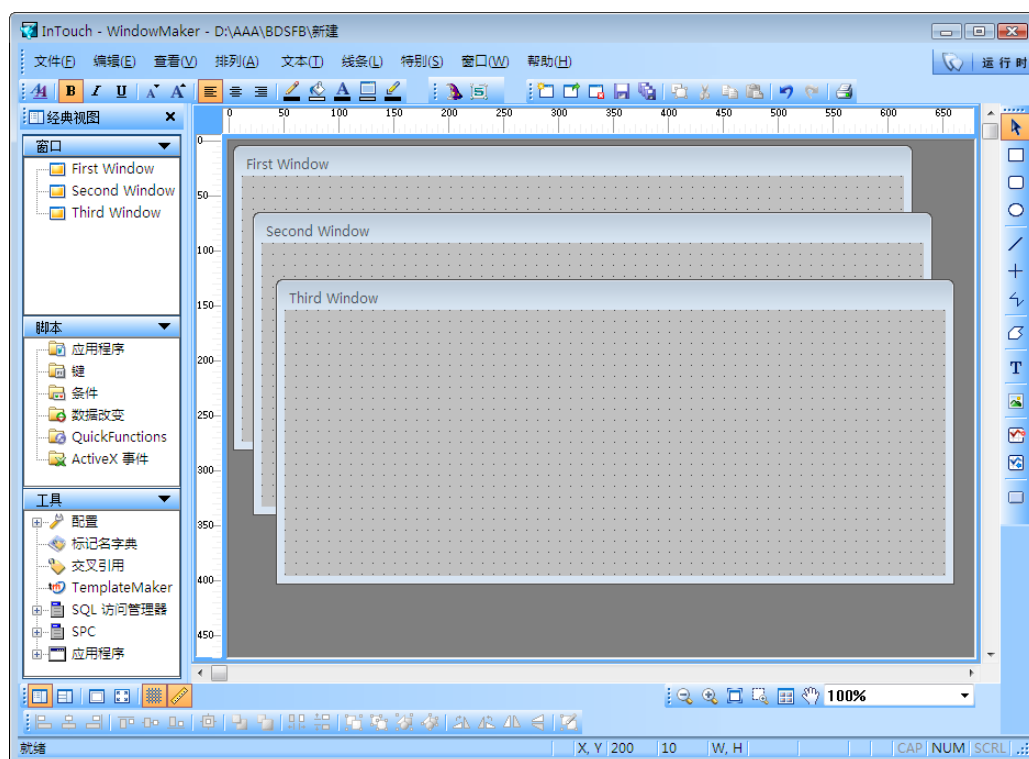
要查看许可证信息

- 1 在**帮助**菜单上，单击**关于**。此时出现 **About InTouch WindowMaker**（关于 InTouch WindowMaker）对话框。
- 2 单击 **View License**（查看许可证）打开许可证管理器，以查看关于许可证的信息，或是管理许可证。
- 3 单击 **View License Agreement**（查看许可协议）以阅读许可协议或查看专利信息。要查看专利信息，必须有 Internet 连接。

第 2 章

应用程序窗口

应用程序窗口是一个容器，它包含用于给生产过程建模的一个或多个图形。例如，您可以在窗口中使用某个单元来显示设备。另一个窗口可以显示与该单元有关的一组报警信息。



您可以创建任意数量的窗口，并且可以定义各种窗口属性，如背景颜色、屏幕位置、窗口标题，等等。

创建应用程序窗口

创建新的应用程序窗口时，其缺省设置采用先前创建的或当前活动的窗口的设置。

窗口名最多可以包含 32 个字符，并且可以包含键盘上除英文引号之外的任何字符。创建窗口时，必须提供的只有窗口名。所有其它项目都是可选的。

您可以给窗口包含注释，但它仅用于提供信息。该注释出现在文件列表中，但应用程序并不使用它。

缺省条件下，窗口的尺寸值设置为先前创建的窗口的尺寸。如果通过拖动窗口边框来手工更改窗口的大小，这些值也会自动修改。

如果在应用 **Windows 经典** 主题的系统上创建一个应用程序，然后在应用 **Windows XP** 主题的系统上运行该应用程序，则可能出现 InTouch 窗口底部的一些像素显示不全的现象。这是由于在 **Windows XP** 主题中 Windows 的开始菜单栏更高的缘故。

要创建新的窗口

- 1 在文件菜单上，单击**新建窗口**。此时出现**窗口属性**对话框。



- 2 配置基本的窗口属性。执行以下操作：

- 在**名称**框中，输入标识窗口的唯一名称。
- 在**注释**框中，输入要与窗口关联的任何注释。注释必须为 50 个字符或者更少。
- 单击**窗口颜色**框，以选择窗口的背景颜色。

- 3 在**窗口类型**区域中，配置窗口在运行时的打开方式。
 - 单击**替换**，则窗口出现在屏幕上时，与它相交的任何窗口都自动关闭。
 - 单击**覆盖**，则窗口出现在当前打开的窗口的上方。它可以比自己所覆盖的窗口更大。覆盖窗口关闭时，它后面的任何窗口都会重新出现。单击重叠窗口后面的窗口的任何可见部分，便可以将该窗口置前，并使之成为活动窗口。
 - 单击**弹出**，则窗口始终出现在所有其它窗口的上方。弹出窗口通常需要用户作出响应才能删除。
- 4 在**框式样**区域中，配置窗口周围的边框。
 - 单击**单一**，可以得到具有三维边框且带有标题栏与大小控制的窗口。选择**标题栏**复选框可以包含标题栏。
 - 单击**加倍**，可以得到具有三维边框且不带标题栏的窗口，此时如果没有选择大小控制，则窗口大小也无法调整。
 - 单击**无**，可以得到不带边框与标题栏的窗口，此时如果没有选择大小控制，则窗口大小也无法调整。
- 5 选择**大小控制**复选框，则用户可以在运行时调整窗口的大小。
- 6 在**位置大小**区域中，指定窗口的位置与大小。执行以下操作：
 - 在**X 位置**框中，输入设计区域的左侧边缘与正在定义的窗口的左侧边缘之间的像素数量。
 - 在**Y 位置**框中，输入设计区域的顶部边缘与正在定义的窗口的顶部边缘之间的像素数量。
 - 在**窗口宽度**与**窗口高度**框中，输入以像素计得窗口宽度与高度。
- 7 单击**确定**。

导入与导出窗口

您可以在 InTouch 应用程序之间导入和导出窗口及其内容。如需有关详细信息，请参阅 *InTouch® HMI 应用程序管理与扩展指南* 中的第 4 章“导出与导入标记定义、窗口以及脚本”。

设置要在运行时出现的窗口

“起始”窗口是用户从图标或菜单命令直接启动 WindowViewer 时，出现在 WindowViewer 中的窗口。

如果使用运行时快速切换来启动 WindowViewer，则不会出现起始窗口。

通过在脚本中使用 ShowHome() 函数，可以在运行时随时显示起始窗口。

要设置起始窗口

- 1 在**特别**菜单上，指向**配置**，然后单击 **WindowViewer**。此时出现 **WindowViewer 属性**对话框。
- 2 单击**起始窗口**选项卡。
- 3 选择要在 WindowViewer 启动时打开的一个或多个窗口。
- 4 单击**确定**。

修改应用程序窗口

开发应用程序时，可以随时修改窗口的属性。

要修改应用程序窗口的属性

- 1 使用鼠标右键单击**经典**或**项目**视图中的窗口名，然后单击**属性**。此时出现**窗口属性**对话框。
- 2 根据需要进行更改。如需有关窗口选项的详细信息，请参阅第 30 页的“创建应用程序窗口”。
- 3 单击**确定**。

打开、保存以及关闭窗口

在开发应用程序期间，只要计算机内存足够，便可以打开任意多的窗口。

打开窗口

要打开窗口

- 1 在**文件**菜单上，单击**打开窗口**。此时出现**要打开的窗口**对话框，它列出应用程序中所有窗口的名称。
- 2 执行以下操作之一：
 - 要打开一个窗口，请双击相应的窗口名。
 - 要打开多个窗口，请选择要打开的各个窗口对应的复选框，然后单击**确定**。

保存窗口时，与该窗口关联的所有图形、QuickScript、属性等同样会保存下来。

保存窗口

要保存窗口

- 1 在**文件**菜单上，单击**保存窗口**。此时出现**要保存的窗口**对话框，它列出所有窗口的名称。
- 2 选择需要保存的窗口。
- 3 单击**确定**。

关闭经过修改的窗口时，程序会提示保存所作的更改。

关闭窗口

要关闭窗口

- 1 在**文件**菜单上，单击**关闭窗口**。此时出现**要关闭的窗口**对话框，它列出当前打开的所有窗口的名称。
- 2 选择窗口名旁边的复选框。
- 3 单击**确定**。

创建窗口的副本

有非常类似的过程需要模拟与控制时，您可能会希望创建窗口的副本，然后按照第二个过程或单元对该副本进行自定义。

您可以随窗口一起创建与该窗口关联的所有图形、QuickScript、属性等的副本。

在开始之前，要创建副本的窗口必须处于打开状态，并至少保存过一次。一次只能创建一个窗口的副本。

要创建窗口的副本

- 1 在**文件**菜单上，单击**窗口另存为**。此时出现**要在新名称下保存的窗口**对话框，它列出当前打开的所有窗口的名称。
- 2 选择窗口名旁边的复选框。
- 3 在**新名**框中，输入新窗口的名称。
- 4 单击**确定**。

删除窗口

要节约计算机的储存空间时，或是“应用程序浏览器”中的窗口列表太长以致于无法管理时，可以删除未使用的窗口。

注意 确保没有误删窗口。已删除的窗口无法使用**撤消**来恢复。

要删除窗口

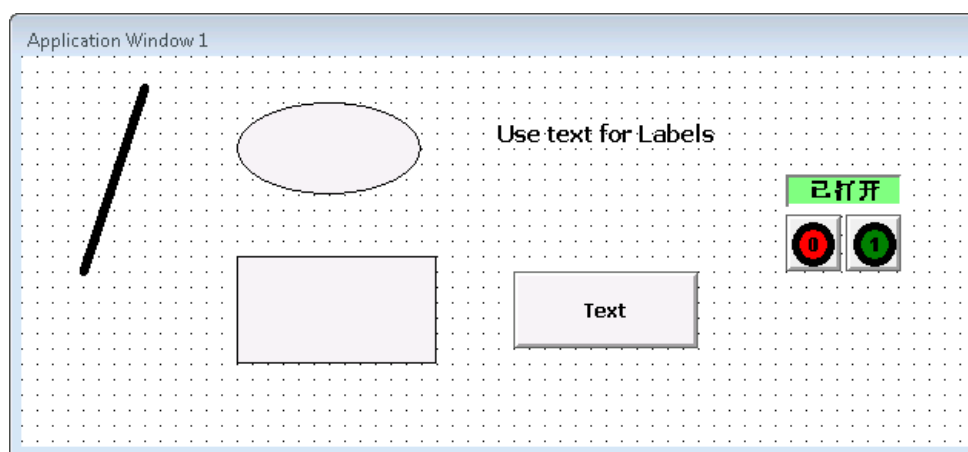
- 1 在**文件**菜单上，单击**删除窗口**。此时出现一个对话框，它列出所有窗口的名称。
- 2 选择要删除的窗口名，然后单击**确定**。出现消息时，单击**是**。
- 3 单击**确定**。

第 3 章

WindowMaker 对象

图形对象是您构建的人机界面 (HMI) 应用程序的关键部分。

在构建应用程序的过程中，您可以创建简单的对象、将简单的对象合并成更复杂的对象，以及使用一些预先定义好的复杂对象。



如果正在使用 InTouch 第 10 版或更新的版本来创建新的图形，则 “ArchestrA 符号” 或许更适合您的应用程序。如需有关详细信息，请参阅 *InTouch® HMI 与 ArchestrA® 集成指南* 中的第 1 章 “关于 InTouch HMI 与 ArchestrA 集成”。

简单对象

您可以创建以下类型的简单对象：






- 线条
- 图案
- 文本
- 按钮

每种简单对象都有一些控制其外观的属性。

- 线条颜色与粗细
- 填充颜色
- 高度
- 宽度
- 方向

创建线条与图案

下表介绍如何执行基本的绘图任务。绘图按钮位于“绘图”工具栏。

要绘制	单击	按钮
线条	“线条”按钮	
水平或垂直线条	“水平 / 垂直线”按钮	
长方形	“长方形”按钮	
圆角长方形	“圆角长方形”按钮	
备注 要调整圆角长方形的圆角半径，请参阅第 54 页的“更改圆角长方形的圆角半径”。		
圆或椭圆	“椭圆”按钮。按住 SHIFT 键可以绘制圆。	

创建按钮

您可以使用按钮来创建与应用程序交互的点。此过程与创建简单的绘图对象非常类似。

如需有关创建多边形的详细信息，请参阅第 37 页的“创建多边形与多边形”。

要创建按钮



- 1 在**绘图**工具栏上，单击**按钮**按钮。
- 2 单击并拖动，以放置按钮及调整其大小。
- 3 编辑缺省的按钮文本。执行以下操作：
 - a 使用鼠标右键单击该按钮，然后单击**替换字符串**。
 - b 在**新字符串**框中，输入按钮的文本。
 - c 单击**确定**。

创建多边形与多边形

绘制多边形与绘制线条略有不同。

要创建多边形或多边形



- 1 在**绘图**工具栏上，单击**多边形或多边形**按钮。
- 2 单击应用程序窗口以设置第一个点。
- 3 在应用程序窗口上再次单击，以设置更多的点来定义多边形或多边形。
- 4 通过双击设置最后一个点。

创建文本

您可以使用文本来给应用程序中的可视化项目贴标签。

创建文本时，文本的格式设置与 **WindowMaker 属性**对话框中的设置相匹配。您可以更改所选文本的外观。如需有关详细信息，请参阅第 46 页的“更改文本外观”。

输入多行文本时，它们会变成可以独立移动与编辑的对象。您也可以将多个文本对象合并成一个符号，将它们当作一个组进行编辑。

要创建文本



- 1 在**绘图**工具栏上，单击**文本**按钮。
- 2 在要开始文本的位置单击。
- 3 输入文本，然后按 **ENTER** 键。此时出现新的文本行。

复杂对象

复杂对象提供比简单对象更多的功能。复杂对象有以下类型。

复杂对象	描述
单元	结合成单一整体的两个或更多个对象（包括符号或其它单元）。您可以使用单元来创建虚拟设备，如游标控制器。对于创建要与不同的标记关联的多个设备，单元非常有用。
符号	结合在一起并视为单个对象的一组简单对象（如线条、图案以及文本）。应用于符号的任何属性更改都会影响该符号中的所有组成对象。符号不得包含位图、按钮、单元、向导或趋势。
SmartSymbol	已经转换成可复用图形模板的 InTouch 单元。在应用程序窗口中可以放置一个或多个 SmartSymbol 模板的实例。对模板所作的任何更改都会传播到这些实例。如需有关详细信息，请参阅 <i>InTouch® HMI SmartSymbol 指南</i> 中的第 1 章“关于 SmartSymbol”。
ArchestrA 符号	使用“ArchestrA 集成开发环境” (IDE) 中的“符号编辑器”创建的功能非常丰富的图形。如需有关详细信息，请参阅 <i>InTouch® HMI 与 ArchestrA® 集成指南</i> 中的第 1 章“关于 InTouch HMI 与 ArchestrA 集成”。
位图容器	可以导入图像（如照片、图画以及屏幕截图）的对象。您可以旋转位图，也可以给它设置透明背景。如需有关详细信息，请参阅第 52 页的“使用位图容器”。
趋势对象	多个标记的实时或历史数据值随时间而变化的图表。如需有关详细信息，请参阅第 102 页的“趋势对象”。
向导	只需针对应用程序进行选择、放置以及配置的预先构建的对象。如需有关详细信息，请参阅第 97 页的“向导”。
ActiveX 控件	在应用程序中运行的软件组件。WindowMaker 同时支持 Wonderware 的与第三方的 ActiveX 控件。如需有关详细信息，请参阅第 134 页的“使用 ActiveX 控件”。

单元与符号

您可以将多个对象合并成两种不同类型的单个整体：单元与符号。单元可包含任何对象。符号只能包含简单对象。符号不得包含单元。

要确定特定的对象是单元还是符号，请双击该对象。

- 如果是单元，则打开**替换标记名**对话框；如果单元不包含标记名，则出现“替换名”警告消息。
- 如果是符号或简单的图形对象，则打开**动画链接选择**对话框。

关于单元

使用单元可以合并多个元素，并在它们之间保持固定的间距关系。您也可以使用单元来四处移动多个元素，也可以将它与其它图形对齐。

要更改单元中的元素，必须分解该单元，对元素进行更改，然后重新将这些元素合并成单元。

您可以给单元中的元素设置动画效果，但是无法给单元设置动画效果。单元的大小也无法调整。

关于符号

您可以给符号与简单对象设置动画效果。您也可以使用符号给复杂图形的某些部分设置动画效果。

如果有多个所选对象包含链接，则无法制作符号。

如果将两个符号合并成一个新符号，则原始的符号结构会丢失。如果分解新符号，则它会分解成每个原始符号的各个单独组成部分。两个原始符号都会丢失。

将对象组合成单元

您可以将符号、位图、趋势、按钮、向导以及其它单元合并成一个单元。如果单元中包含符号，则与该符号关联的所有动画链接都会保持不变。

在 SmartSymbol 中使用单元之后，如果分解该 SmartSymbol，则无法调整该单元的大小。

要创建单元

- 1 选择要包含的对象。
- 2 在**排列**菜单上，单击**制作单元**。

要分解单元

- 1 选择单元。
- 2 在**排列**菜单上，单击**分解单元**。

将对象组合成符号

符号不得包含位图、按钮、单元、向导或趋势。如果所选对象之一上附有动画链接，则这些链接会附加到新符号上。

要创建符号

- 1 选择要包含的对象。
- 2 在**排列**菜单上，单击**制作符号**。

要分解符号

- 1 选择符号。
- 2 在**排列**菜单上，单击**分解符号**。

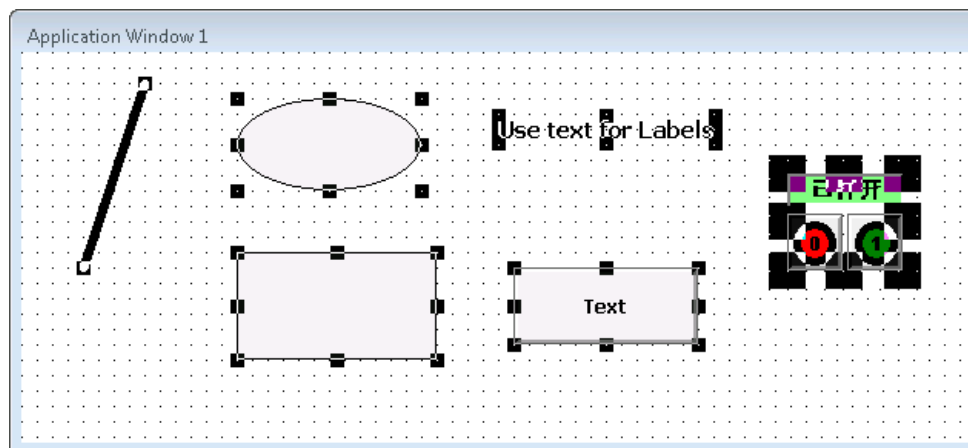
共同的操作

使用鼠标右键单击对象可以看到一个菜单，它显示可应用于该对象的有效命令或动作。您可以：

- 选择对象
- 移动对象
- 对齐对象
- 排列对象
- 层叠对象
- 撤消更改
- 翻转对象
- 翻转符号
- 调整对象大小
- 旋转对象
- 更改字体
- 更改线条或轮廓
- 更改填充
- 删除对象
- 控制水平与垂直间距

选择对象

修改对象之前，必须先选择它。选中对象时，它的周边会出现手柄。这些手柄可用于调整对象的大小及 / 或更改对象的形状。



要选择活动窗口中的所有对象

- ◆ 在**编辑**菜单上，单击**全选**，或者按 **F2**。



要选择一个对象

- ◆ 单击**选择模式**按钮，然后单击要选择的对象。

要取消选择一个对象

- ◆ 单击窗口的空白区域。

要选择多个对象

- ◆ 单击**选择模式**按钮，选择第一个对象，然后按住 **SHIFT** 键并单击其它对象。

要选择一组对象

- ◆ 单击**选择模式**按钮，然后在对象周围拖出一个框。此时会选择完全包含在该长方形内的所有对象。

要取消选择一组对象中特定的一个或多个对象

- ◆ 按住 **SHIFT** 键并单击对象。

移动对象

对象可以通过以下方式移动：

- 拖动对象。
- 使用键盘上的方向键。
- 在状态栏的框中输入窗口坐标。

移动对象时，请注意状态栏中坐标的变化情况。

要通过拖动来移动对象

- ◆ 选择对象并拖动它。

使用方向键移动对象时，对象的移动距离取决于是否显示网格。

显示网格时，对象移动多少个像素取决于网格间距，此间距在 WindowMaker 属性对话框中设置。缺省设置是网格点之间相距十个像素。

显示网格时，

- 按方向键一次，对象移动一个网格点。
- 按 SHIFT + 方向键，对象移动两个网格点。
- 按 CTRL + 方向键，对象移动四个网格点。

不显示网格时，

- 按方向键一次，对象移动一个像素。
- 按 SHIFT + 方向键，对象移动十个像素。
- 按 CTRL + 方向键，对象移动 50 个像素。

要使用方向键移动对象

- ◆ 选择对象，然后
 - 按方向键。
 - 按 SHIFT + 方向键。
 - 按 CTRL + 方向键。

要使用状态栏移动对象

- 1 选择对象。
- 2 在状态栏中输入 X 与 Y 坐标。
- 3 按 Enter 键。

X, Y	188	121
------	-----	-----

对齐对象

您可以根据对象的左侧或右侧边缘、中心、中心点、顶部、中间或底部来对齐它们。

通过使用菜单命令或按钮，可以按多种方式进行对齐。

选择	或单击	实现
左对齐		以最左端对象的左边缘为基准，对齐各个对象的左边缘。
水平居中对齐		以对象组的垂直中心线为基准，对齐各个对象的垂直中心线。
右对齐		以最右端对象的右边缘为基准，对齐各个对象的右边缘。
上对齐		以最顶端对象的顶部边缘为基准，对齐各个对象的顶部边缘。
垂直居中对齐		以对象组的水平中心线为基准，对齐各个对象的水平中心线。
下对齐		以最底端对象的底部边缘为基准，对齐各个对象的底部边缘。
中心点对齐		以对象组的中心点为基准，对齐各个对象的中心点。

要对齐对象

- 1 选择多个对象。
- 2 在排列菜单上，指向**对齐**，然后单击适当的对齐命令。

层叠对象

您可以将对象放置在其它对象的前面或后面。

要将一个对象置于另一对象之后

- 1 选择对象。
- 2 在排列菜单上，单击**置后**。



要将一个对象置于另一对象之前

- 1 选择对象。
- 2 执行以下操作之一：
 - 在排列工具栏上，单击**置前**。
 - 在排列菜单上，单击**置前**。
 - 按 SHIFT + F9。



控制水平与垂直间距

您可以在最左端与最右端的所选对象之间沿水平方向分布对象。

您也可以在最顶部与最底部的所选对象之间控制垂直间距。

要水平或垂直分布对象

1 选择对象。

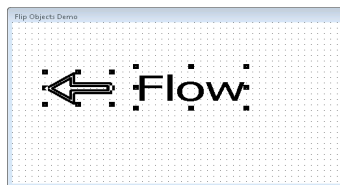
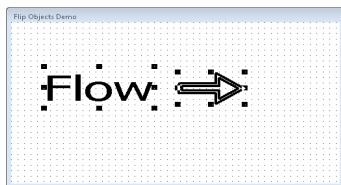


2 在排列工具栏上，单击**水平间距**或是单击**垂直间距**。

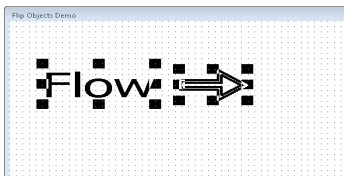
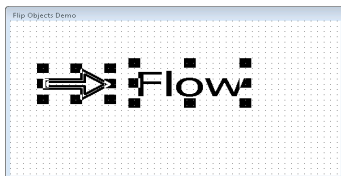
翻转对象与单元

大多数对象都可以水平或垂直翻转。您可以翻转单个对象或一组对象。

翻转对象相当于将它变换成自己的镜像。文本无法翻转。



翻转单元并不会将单元变成镜像。只有对象组中单元的位置会进行镜像处理。



比较单元翻转前、后的位置。翻转的是位置而不是内容。

要翻转对象或单元

1 选择对象。



2 在排列工具栏上，单击**水平翻转**，或单击**垂直翻转**。

调整对象大小

对象的大小可以使用两种方法来调整。您可以拖动它，或是指定精确的宽度与高度。

如果打开对齐网格功能，则在按比例调整大小期间，对象会与网格对齐。这会导致垂直与水平尺寸之间的比率稍有偏差。要避免此偏差，请关闭对齐网格功能。

要通过拖动来调整对象的大小

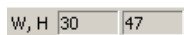
- 1 选择对象，然后将箭头光标指针置于手柄中央。
- 2 拖动手柄以调整对象的大小。

要按比例调整对象的大小

- ◆ 选择对象，然后按 **SHIFT** 键并拖动。

要按尺寸调整对象的大小

- 1 选择对象。
- 2 在状态栏上的 **W, H** 框中，输入宽度与高度尺寸。



旋转对象

大多数对象（包括符号、文本与位图）都可以旋转。单元无法旋转。

对象可以顺时针或逆时针旋转 360 度，增量为 90 度。

在 WindowMaker 中旋转对象与在运行时或是在 WindowViewer 中动态旋转对象是两回事。在 WindowViewer 中，通过将对象链接到方向动画来旋转对象。

要旋转对象

- 1 选择对象。
- 2 在排列工具栏上，单击顺时针旋转，或是单击逆时针旋转。



更改文本外观

您可以在创建文本之前，通过预先设置缺省值来设置字体的外观；也可以在创建文本之后，再更改字体的外观。如需有关设置字体缺省值的详细信息，请参阅第 24 页的“设置字体缺省值”。

对于显示动态值的文本对象而言，文本对齐方式属性设置非常重要。对齐方式确定各个变长字段在运行时如何显示。

例如，如果在居中对齐或右对齐的文本字符串的末尾显示一个数值，则每当显示的数字位数发生改变时，整个文本字符串（包括数值）都会居中对齐或右对齐。

通过使用菜单命令或按钮，可以按多种方式配置文本。

作用	单击	按钮
更改字体、样式、颜色或文本大小	字体	 
使文本变为粗体	粗体	
使文本变为斜体	斜体	
给文本添加下划线	下划线	
减少或增加字体大小	字体缩小或字体放大	 
更改对齐方式	左对齐、居中或右对齐	

要配置文本外观

- 1 选择文本对象。
- 2 在文本菜单上，单击适当的文本命令。

更改线条与轮廓

您可以更改轮廓化对象周围的线条或轮廓的颜色，以及它们的线型或宽度。轮廓化对象包括填充图案（如椭圆、长方形、多边形）、位图以及其它导入的图像。

更宽的线条在运行时需要更长的时间来绘制。虚线与点划线的宽度只能是一个像素。

要设置线条外观的缺省设置

- 1 单击窗口的空白区域。
- 2 在**线条**菜单上，选择线条宽度或线型。
- 3 在**格式**工具栏上，单击**线颜色**工具。
- 4 选择一种颜色。



要更改线条的颜色

- 1 选择一个线条、一组线条或是具有轮廓的对象。
- 2 在**格式**工具栏上，单击**线颜色**工具。
- 3 选择一种颜色。



要更改线条或轮廓的样式或宽度

- 1 选择对象。
- 2 在**线条**菜单上，单击线条的样式或宽度。

要删除轮廓

- 1 选择对象。
- 2 在**线条**菜单上，单击**无线**。

更改填充

填充的图案是由线条包围着的图案。长方形、圆角长方形、圆、椭圆以及多边形都是填充图案的示例。

要更改对象的填充颜色

- 1 选择对象。
- 2 在**格式**工具栏上，单击**填充颜色**工具。
- 3 选择一种颜色。



要设置填充图案的缺省颜色

- 1 单击窗口的空白区域。
- 2 在**格式**工具栏上，单击**填充颜色**工具。
- 3 选择一种颜色。



删除对象

您可以删除一个或多个对象。

要删除对象

- ◆ 执行以下操作之一：
 - 使用鼠标右键单击对象，然后单击**清除**。
 - 选择对象，然后按 **Delete** 键。

撤消更改

WindowMaker 会记录对每个窗口所作的编辑与格式更改。缺省条件下，WindowMaker 支持 10 级撤消 / 恢复，其中每级代表一个动作。您可以将 WindowMaker 设置为保留多达 25 级动作。您也可以通过将撤消 / 恢复的级数设置为零来关闭撤消 / 恢复功能。

如果关闭窗口，则会清除记录的所有操作。

要撤消命令

- ◆ 在**编辑**菜单上，单击**撤消**。

要恢复命令

- ◆ 在**编辑**菜单上，单击**恢复**。

要设置撤消 / 恢复级数

- 1 在**特别**菜单上，指向**配置**，然后单击 **WindowMaker**。此时出现 **WindowMaker 属性**对话框。
- 2 在**撤消级别**框中，输入级数。

所有对象的特殊操作

您可以修改与操纵简单的对象。您可以：

- 剪切、复制以及粘贴对象。
- 剪切、复制以及粘贴对象链接。
- 创建对象的副本。

剪切、复制以及粘贴对象

WindowMaker 中的剪切、复制以及粘贴操作同其它基于 Windows 的应用程序中的操作大体相同，但也有一些需要注意的重大区别。

剪切、复制或粘贴对象时，同时也会剪切、复制或粘贴与对象相关的属性与动画链接。

粘贴的所有对象在粘贴之后仍保持选中状态，您可以通过移动来调整它们的位置。

要剪切对象

- ◆ 使用鼠标右键单击对象，然后单击**剪切**。

要复制对象

- ◆ 使用鼠标右键单击对象，然后单击**复制**。

要粘贴对象

- 1 使用鼠标右键单击窗口中的空白区域，然后单击**粘贴**。此时光标变为弯头符号。
- 2 按住鼠标左键。此时光标变为虚线长方形，大小等于所复制的对象。
- 3 拖动该长方形以指定对象的位置。
- 4 释放鼠标按钮。

剪切、复制以及粘贴对象链接

“剪贴板”是剪切或复制的链接的临时储存区。

- “剪贴板”只存储最近的剪切或复制操作对应的链接。
- 您可以将链接粘贴到支持“剪贴板”中的链接的任何对象或符号。
- 如果对象不支持所粘贴的链接（例如，文本对象上的线条颜色链接），则不会粘贴该链接。
- 如果选择多个对象进行粘贴，则这些链接会粘贴到所有的对象。

要剪切、复制、粘贴以及清除链接

- ◆ 使用鼠标右键单击对象，指向**链接**，然后单击适当的命令。

创建对象的副本

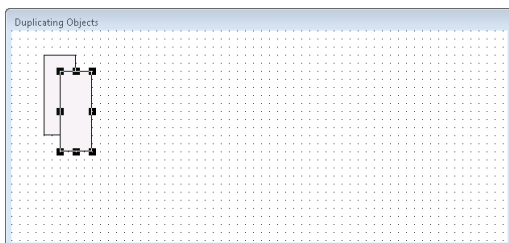
给对象创建副本同复制对象及其动画链接类似，此外也有一个优点，即多次给对象创建副本时，偏移距离与方向同样也会重复。

如果在没有取消选择的情况下移动通过创建副本得到的对象，则再次创建副本时，第三个副本的偏置距离和方向与前两个副本之间的相同。

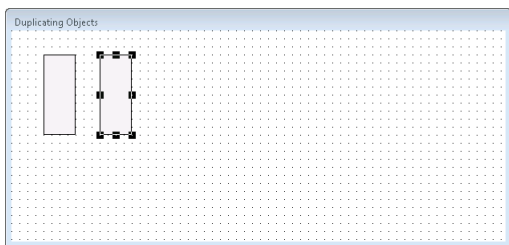
您可以根据需要重复这个操作程序任意多次。

要创建对象的副本

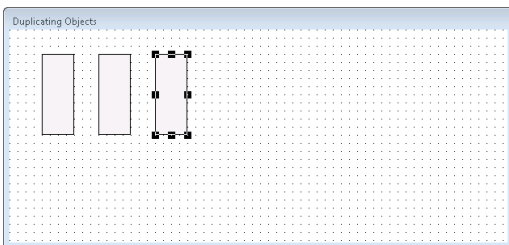
- 1 使用鼠标右键单击对象，然后单击**重复**。此时会复制对象并将它粘贴到与原始对象存在一定偏移的位置。



- 2 保持选择所重复的对象，然后将它拖到一个不同的位置。



- 3 同样，在没有取消选择重复的对象的情况下，使用鼠标右键单击该对象，然后再次单击**重复**。此时出现对象的第三个副本，它的相对位置与前两个副本的相同。



特殊对象的特殊操作

以下对象类型包含一些可以编辑的独特属性：

- 多边线与多边形
- 位图容器
- 位图透明度
- 圆角长方形
- 对象文本

调整多边线与多边形对象的形状

您可以调整多边线与多边形的形状。

要调整多边线与多边形的形状

- 1 选择对象。
- 2 执行以下操作之一。
 - 在**编辑**菜单上，单击**改变对象外形**。
 - 使用鼠标右键单击该对象，然后单击**改变对象外形**。
 - 在**排列**工具栏上，单击**改变对象形状**按钮。



此时每个形状定义顶点都变成一个手柄。

- 3 拖动手柄以调整形状。

要给多边形添加顶点

- 1 选择对象。
- 2 执行以下操作之一。
 - 在**编辑**菜单上，单击**添加顶点**。
 - 使用鼠标右键单击对象，然后单击**添加顶点**。
- 3 单击多边形的边缘，然后拖动顶点以更改多边形的形状。

要删除多边形的顶点

- 1 选择对象。
- 2 执行以下操作之一。
 - 在**编辑**菜单上，单击**删除顶点**。
 - 使用鼠标右键单击对象，然后单击**删除顶点**。
- 3 单击多边形的顶点，此时会删除该顶点，多边形的形状也随之发生更改。

使用位图容器

位图容器是用于将图形对象（如图片、屏幕截图以及图画）导入应用程序的对象。

可用的文件类型有 .bmp、.jpeg、.jpg、.pcx 以及 .tga。

导入位图时，位图自动填充位图容器；但也可以将它调整为原始的大小与比例。

位图可以按 90 度的增量旋转。

位图可以包含在单元，但无法包含在符号中。

通过使用 WindowMaker，可以将比 WindowViewer 所能加载的更多位图放入窗口。如果需要将大量的位图放入窗口，在发行应用程序之前，请务必在 WindowViewer 中测试该窗口。

要导入位图图像



- 1 在**绘图**工具栏上，单击**位图**工具。此时光标变成十字形。
- 2 拖动光标以绘制位图容器。
- 3 在**编辑**菜单上，单击**导入图像**。此时出现**选择图像文件**对话框。
- 4 选择图像文件名，然后单击**打开**。

要使位图变为原始大小

- 1 选择图像。
- 2 在**编辑**菜单上，单击**位图 - 原始大小**。

要粘贴位图图像



- 1 将图形复制到 Windows 的“剪贴板”。
- 2 单击**位图**工具，并在窗口中绘制一个位图容器。
- 3 使用鼠标右键单击该位图容器，然后单击**粘贴位图**。

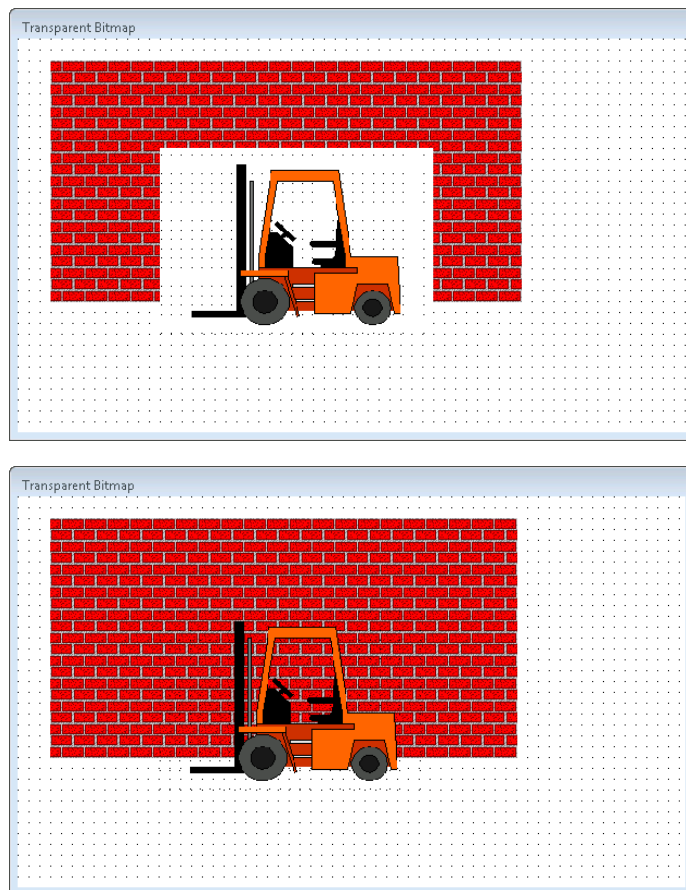
要编辑位图

- 1 选择位图。
- 2 在**编辑**菜单上，单击**编辑位图**。此时打开 Microsoft 的“画图”并显示该位图。
- 3 在 MS 的“画图”中编辑位图。
- 4 保存并关闭 MS 的“画图”。

定义位图透明度

在位图中定义一种透明色时，可以透过使用透明色的所有位置看到窗口背景或位图后面的任何对象。

每幅图像只能定义一种透明色。



要创建透明位图



1 选择位图，单击**格式**工具栏上的**透明颜色**按钮，以打开透明色调色板。

2 使用鼠标右键单击**自定义调色板**中的一个颜色方块。此时出现**添加颜色**对话框。



3 单击吸管工具。

4 单击位图中要变为透明色的颜色。此时该颜色会复制到调色板中所选的颜色方块。

5 单击该颜色方块以便将透明颜色应用于位图。图像中使用该颜色的所有像素都变为透明的。

更改圆角长方形的圆角半径

您可以增加与 / 或减少 “圆角长方形” 的圆角半径。

要增加或减少圆角对象的半径

- 1 选择对象。
- 2 在**编辑**菜单上，单击**放大半径**或**缩小半径**。

替换对象文本

您可以编辑包含文本的对象（如符号、单元或按钮）的文本。

更改文本字符串时，它会保留原始的所有属性，包括字体、样式以及颜色等。文本格式也适用于数值。

要更改对象中的文本

- 1 选择包含文本的对象或按钮。执行以下操作之一：
 - 在**特别**菜单上，单击**替换字符串**。
 - 使用鼠标右键单击文本对象，指向**替换**，然后单击**替换字符串**。
- 2 在**新字符串**框中，输入新的字符串，然后单击**确定**。

要更改一系列文本对象中的部分文本

- 1 选择所有的文本对象。
- 2 在**特别**菜单上，单击**替换字符串**。
- 3 单击**替换**。此时出现**替换文本**对话框。
- 4 在**旧文本**框中，输入要替换的部分文本。
- 5 在**新文本**框中，输入新的文本字符串。
- 6 单击**确定**。此时新的文本字符串替换掉所选的所有对象中的旧文本字符串。

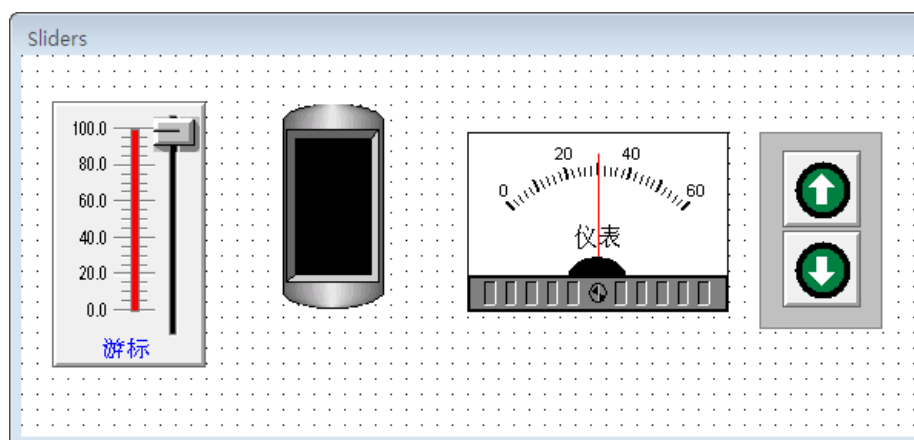
第 4 章

设置对象动画效果

通过使用动画链接，可以给对象或符号设置动画效果。动画链接将标记或表达式的值连接到对象或符号。

例如，可以：

- 创建显示罐中液位的游标或罐符号。
- 创建仪表，显示一定范围的值。
- 创建可供操作员控制的触摸屏符号。



两种类型的动画链接

动画链接有两种基本类型：显示链接与触动链接。

- 显示链接用于给操作员显示信息。显示链接的例子有：更改颜色，更改填充级别、水平或垂直移动，以及使对象闪烁。
- 触动链接可供操作员向系统进行输入。触动链接的例子有：游标，或用于响应操作员输入的按钮。

您可以为对象或符号定义多个链接。通过组合各种链接，您可以创建几乎任何屏幕动画效果。

数据显示动画

数据显示动画只用于向操作员显示信息。这些动画不允许操作员进行输入。

创建值显示

使用值显示文本对象显示标记的值。这可用于显示填充级别、开 / 关状态或报警消息等对象。

您可以使用三种类型值显示链接中的任何一种来显示运行时的消息。

值显示类型	显示
离散	离散值，如开或关。
模拟	模拟表达式的值，如填充级别或速度。
字符串	字符串表达式的值，如 “Fill Level = 100”。

在表达式中，您可以使用长达 1023 个字符。如果需要更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

消息会出现在原始文本对象所在位置，使用为该对象设置的字体、大小、颜色、对齐方式及链接的属性。字段的原始内容不会影响运行时的消息。

要创建离散值显示链接

- 1 使用鼠标右键单击文本对象，然后单击**动画链接**。此时出现**动画链接**对话框。
- 2 在**值显示**区域中，单击**离散**。此时出现**输出 -> 离散表达式**对话框。

- 3 在**表达式**框中，输入离散标记名或等于离散值的表达式。例如：
Cooling_Pump
- 4 在**打开消息**框中，输入在表达式的值等于 1、true、on 或 yes 时要出现的消息。例如：
Pump is ON
- 5 在**关闭消息**框中，输入在表达式的值等于 0、false、off 或 no 时要显示的消息。例如：
Pump is OFF
- 6 单击**确定**。

要创建模拟值显示链接

- 1 使用鼠标右键单击文本对象，然后单击**动画链接**。此时出现**动画链接**对话框。
- 2 在**值显示**区域中，单击**模拟**。此时出现**输出 -> 模拟表达式**对话框。

- 3 在**表达式**框中，输入模拟（整型或实型）标记名或等于模拟值的表达式。例如：
Tank_CV*0.06
- 4 单击**确定**。

要创建字符串值显示链接

- 1 使用鼠标右键单击文本对象，然后单击**动画链接**。此时出现“动画链接”对话框。
- 2 在**值显示**区域中，单击**字符串**。此时出现**输出 -> 字符串表达式**对话框。

- 3 在**表达式**框中，输入消息标记名或使用消息标记的表达式。
例如：
“The Tank Level is:” + Text(TankLevel, ” #”)
- 4 单击**确定**。

创建移动

通过使用位置链接，可以使对象在运行时移动。您可以根据模拟标记或表达式值的变化，让对象沿水平、垂直或同时沿两个方向移动。例如，随着贮料罐液位的升高或降低，指示器上下移动。

要创建水平移动

- 1 将对象放到屏幕上的开始位置。
- 2 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 3 在**位置**区域中，单击**水平**。此时出现**水平位置**对话框。

属性	值	水平移动
在左端(D):	0	向左(O): 0
在右端(G):	100	向右(H): 100

- 4 在**表达式**框中，输入模拟标记名或等于模拟值的表达式。
- 5 在**属性**区域，配置对象移动的距离。执行以下操作：
 - a 在**在左端**框中，输入对象位于最左端时的模拟标记的值。
 - b 在**在右端**框中，输入对象位于最右端时的模拟标记的值。
 - c 在**向左**框中，输入对象应向起始位置左侧移动的像素数。
 - d 在**向右**框中，输入对象应向起始位置右侧移动的像素数。
- 6 单击**确定**。

要创建垂直移动

- 1 将对象放到屏幕上的开始位置。
- 2 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 3 在**位置**区域中，单击**垂直**。此时出现**垂直位置**对话框。



- 4 在**表达式**框中，输入模拟标记名或等于模拟值的表达式。
- 5 在**属性**区域中，执行以下操作：
 - a 在**在顶端**框中，输入对象位于最顶端时的模拟标记的值。
 - b 在**在底端**框中，输入对象位于最底端时的模拟标记的值。
 - c 在**向上**框中，输入对象应从起始位置向上移动的像素数。
 - d 在**向下**框中，输入对象应从起始位置向下移动的像素数。
- 6 单击**确定**。

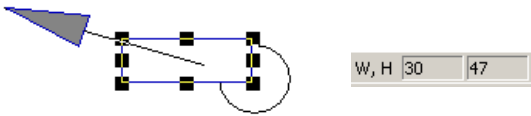
创建旋转

通过使用方向链接，可以根据模拟标记值的变化，让对象绕中心点旋转。例如，随着压强的升高或降低，指针可以绕刻度盘旋转。

方向链接使用对象或符号的中心作为旋转的缺省中心。您可以偏移旋转的中心。

Archestra 图形不支持方向链接。

提示 从对象的中心到旋转中心点绘制一个临时的长方形。现在，您可以从状态栏的 W、H 框中读取 X、Y 偏移大小（以像素计）。



要创建方向链接

- 1 使用鼠标右键单击对象，然后单击**动画链接**。此时出现**动画链接**对话框。
- 2 在其它区域中，单击**方向**。此时出现**方向 -> 模拟值**对话框。

方向 -> 模拟值

表达式(E): 确定(O)

取消

属性

最大逆时针旋转时的值(V): 逆时针旋转(C):

最大顺时针旋转时的值(M): 顺时针旋转(W):

清除(L)

旋转中心相对于对象中点的偏移量

X位置(X): Y位置(Y):

- 3 在**表达式**框中，输入模拟标记名或等于模拟值的表达式。
- 4 在**属性**区域中，执行以下操作：
 - a 在**最大逆时针旋转时的值**框中，输入对象要旋转到最大逆时针位置的表达式值。
 - b 在**最大顺时针旋转时的值**框中，输入对象要旋转到最大顺时针位置的表达式值。
 - c 在**逆时针旋转**框中，输入到达**最大逆时针旋转时的值**时，对象逆时针旋转的度数。
 - d 在**顺时针旋转**框中，输入到达**最大顺时针旋转时的值**时，对象顺时针旋转的度数。
- 5 在**旋转中心相对于对象中点的偏移量**区域中，执行以下操作：
 - a 在**X 位置**框中，输入旋转中心点的水平偏移量。输入距对象中心点的偏移量（以像素计）。
 - b 在**Y 位置**框中，输入旋转中心点的垂直偏移量。输入距对象中心点的偏移量（以像素计）。
- 6 单击**确定**。

设置大小动画效果

通过使用对象大小链接，可以根据模拟标记或表达式的值来改变对象的高度与 / 或宽度。

例如，压力指示器的值随着压力的升高而增大，或者传送带上的对象通过变得更大而呈现为向观察者移动。

通过使用动画的参考位置，对象大小链接不仅可以控制对象的大小，而且可以控制对象大小变化的方向。

要创建对象大小高度链接

- 1 使用鼠标右键单击对象，然后单击**动画链接**。此时出现**动画链接**对话框。
- 2 在**对象大小**区域中，单击**高度**。此时出现**对象高度 -> 模拟值**对话框。



对象高度 -> 模拟值

表达式(E):

属性

最大高度值(M): 100 最大百分比高度(H): 100

最小高度值(N): 0 最小百分比高度(E): 0

参考位置

☐ 顶端(T) ☐ 中间(D) ☒ 底端(B)

确定 取消 清除(L)

- 3 在**表达式**框中，输入模拟标记名或等于模拟值的表达式。
- 4 在**属性**区域中，执行以下操作：
 - a 在**最大高度值**框中，输入对象达到最大高度时的标记或表达式的值。
 - b 在**最小高度值**框中，输入对象达到最小高度时的标记或表达式的值。
 - c 在**最大百分比高度**框中，输入在标记名或表达式达到**最大高度值**框中设置的值时，对象占原始高度的百分比。百分比数字以对象的绘制大小的百分比来表示。绘制大小始终是 100%。
 - d 在**最小百分比高度**框中，输入在标记名或表达式达到**最小高度值**框中设置的值时，对象占原始高度的百分比。百分比数字以对象的绘制大小的百分比来表示。绘制大小始终是 100%。
- 5 选择对象扩展的**参考位置**。
 - 选择**顶端**以使对象从顶端往下扩展。
 - 选择**中间**以使对象从中心点向两边扩展。
 - 选择**底端**以使对象从底端往上扩展。
- 6 单击**确定**。

要创建对象大小宽度链接

- 1 使用鼠标右键单击对象，然后单击**动画链接**。此时出现**动画链接**对话框。
- 2 在**对象大小**区域中，单击**宽度**。此时出现**对象宽度 -> 模拟值**对话框。



- 3 在**表达式**框中，输入模拟标记名或等于模拟值的表达式。
- 4 在**属性**区域中，执行以下操作：
 - a 在**最大宽度值**框中，输入对象达到最大宽度时的标记或表达式的值。
 - b 在**最小宽度值**框中，输入对象达到最小宽度时的标记或表达式的值。
 - c 在**最大百分比宽度**框中，输入在标记名或表达式达到**最大宽度值**框中设置的值时，对象占原始宽度的百分比。百分比数字以对象的绘制大小的百分比来表示。绘制大小始终是 100%。
 - d 在**最小百分比宽度**框中，输入在标记名或表达式达到**最小宽度值**框中设置的值时，对象占原始宽度的百分比。百分比数字以对象的绘制大小的百分比来表示。绘制大小始终是 100%。
- 5 选择对象宽度扩展的**参考位置**。
 - 选择**左端**以使对象从左端扩展。
 - 选择**中间**以使对象从中心点向两边扩展。
 - 选择**右端**以使对象从右端扩展。
- 6 单击**确定**。

设置颜色动画效果

通过使用颜色链接，可以给任何对象设置颜色变化动画效果。变化可以基于模拟或离散标记的值、模拟或离散表达式的值，或是离散或模拟报警状态。

您可以使用三种颜色链接来设置对象的动画效果。

- 线条颜色
- 填充颜色
- 文本颜色

对于这三种颜色链接而言，有四种类型的表达式可以控制颜色变化。

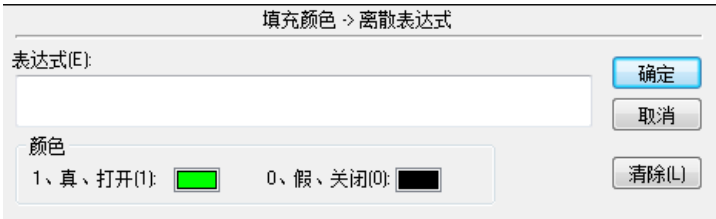
表达式类型	更改颜色的根据
离散	离散标记或表达式的值。
模拟	模拟标记或表达式的值。您可以定义十种颜色来表示不同的值。
离散报警	标记、“报警组”或“组变量”的报警状态。
模拟报警	模拟标记、“报警组”或“组变量”的报警状态。您可以定义五种颜色来表示五种报警条件。

警告！ 使用模拟报警链接时，如果该链接是来自 InTouch 7.11 以前的版本所创建且未转换的应用程序的远程标记，则对象不会进入报警状态。

所有离散颜色链接都是按照相同的方法创建的。以下操作程序介绍如何创建填充颜色链接。

要创建离散填充颜色链接

- 1 使用鼠标右键单击对象，然后单击**动画链接**。此时出现**动画链接**对话框。
- 2 在**填充颜色**区域中，单击**离散**。此时出现**填充颜色 -> 离散表达式**对话框。



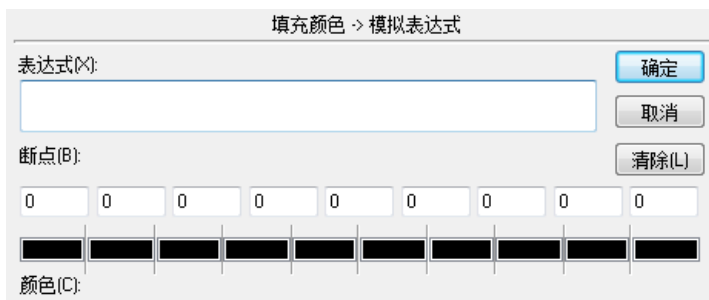
- 3 在**表达式**框中，输入离散标记名或等于 true 或 false 值的离散表达式。

离散表达式可以包含模拟标记。例如，`TankLevel >= 75`。
在本例中，"TankLevel" 变量的值大于或等于 "75" 时，对象的填充颜色会改变。

- 4 在**颜色**区域中，单击每个颜色框以打开调色板。选择要给每个状态使用的颜色。
- 5 单击**确定**。

要创建模拟表达式颜色链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**填充颜色**区域中，单击**模拟**。此时出现**填充颜色 -> 模拟表达式**对话框。



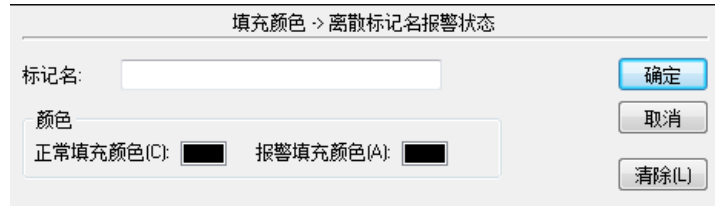
- 3 在**表达式**框中，输入模拟标记名或等于模拟值的表达式。
- 4 在**断点**区域中，执行以下操作：
 - 指定对象将在该处改变颜色的断点值。

提示 您不必使用十个不同的颜色。例如，如果只希望对象更改颜色三次，便只需输入三个值，然后给其余的值使用相同的颜色。如果需要更多样化的范围，请回顾 **ArchestrA** 符号的模拟填充功能。如需有关详细信息，请参阅 *InTouch® HMI 与 ArchestrA® 集成指南* 中的第 1 章“关于 InTouch HMI 与 ArchestrA 集成”。

- 在**颜色**区域中，选择用于每个断点的颜色。
- 5 单击**确定**。

要创建离散报警状态颜色链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**填充颜色**区域中，单击**离散报警**。此时出现**填充颜色 -> 离散标记名报警状态**对话框。



- 3 在**标记名**框中，输入要与对象关联的离散标记名。
- 4 在**颜色**区域中，给每种报警状态选择颜色。
- 5 单击**确定**。

要创建模拟报警状态颜色链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**填充颜色**区域中，单击**模拟报警**。此时出现**填充颜色 -> 模拟标记名报警状态**对话框。



- 3 在**标记名**框中，输入要与对象关联的模拟标记名。

- 4 在**报警类型**区域中，从三种报警类型之中选择要与该对象关联的类型。

报警类型	最多可用
值	五种颜色显示值报警的状态。
偏差	三种颜色显示偏差报警的状态。
ROC（变化率）	两种颜色显示变化率报警的状态。

- 5 在**颜色**区域中，给每种报警状态选择一种颜色。
- 6 单击**确定**。

设置填充级别动画效果

您可以使用填充百分比链接来改变对象的填充级别。填充百分比基于模拟标记或表达式的值。您可以创建水平填充、垂直填充或同时创建这两种填充。

例如，您可以使用垂直填充链接显示罐中的液位，或使用水平填充链接显示过程的进度。

水平填充百分比与垂直填充百分比链接的创建方法相同。

要创建填充百分比链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**填充百分比**区域中，执行以下操作之一：
- 单击**垂直**，此时出现**垂直填充 -> 模拟量**对话框。

垂直填充 -> 模拟量

表达式(E):

确定

取消

属性

最大填充值(A):

100

最大填充百分比(B):

100

最小填充值(C):

0

最小填充百分比(D):

0

方向(D)

☒ 向上

☐ 向下

背景颜色(E):

清除(L)

- 单击**水平**，此时出现**水平填充 -> 模拟量**对话框。

水平填充 -> 模拟量

表达式(E):

属性

最大填充值(A): 100 最大填充百分比(P): 100

最小填充值(I): 0 最小填充百分比(N): 0

方向(D)

☐ 向左 ☒ 向右

背景颜色(B):

确定 取消 清除(L)

- 在**表达式**框中，输入模拟标记名或等于模拟值的表达式。
- 在**属性**区域中，执行以下操作：
 - 在**最大填充值**框中，输入将使对象填充到最大级别的表达式的值。
 - 在**最小填充值**框中，输入将使对象填充到最小级别的表达式的值。
 - 在**最大填充百分比**框中，输入在表达式达到**最大填充值**框中设置的级别时，对象填充的百分比 (0-100)。
 - 在**最小填充百分比**框中，输入在表达式达到**最小填充值**框中设置的级别时，对象填充的百分比 (0-100)。
- 在**方向**区域中，单击填充的方向。
- 在**背景颜色**框中，选择对象未填充部分的颜色。
 - 实际填充颜色是绘制对象时选择的颜色。
 - 如果将垂直填充百分比与水平填充百分比链接同时链接到相同的对象，则最后选择的颜色是背景颜色。
- 单击**确定**。

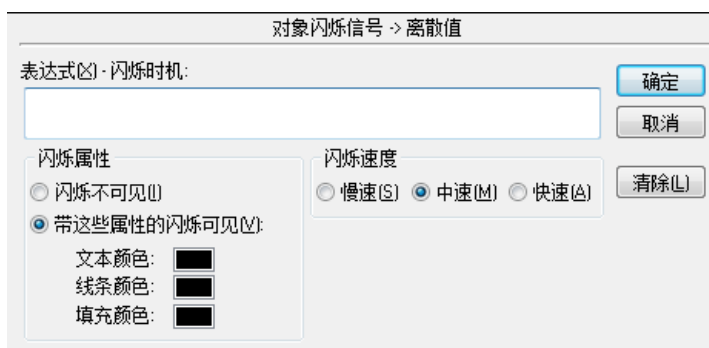
使对象闪烁

通过使用闪烁链接，可以创建根据标记的值进行闪烁的动画对象。例如，您可以创建一个对象，在某种设备打开时，或达到报警设定点时，它呈红色闪烁。

提示 离散表达式可以包含模拟标记名。例如，`TankLevel > 75`。在本例中，当 `TankLevel` 标记的值大于 `75` 时，对象开始闪烁。

要创建闪烁链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**其它区域**中，单击**闪烁**。此时出现**对象闪烁信号 -> 离散值**对话框。



- 3 在**表达式 - 闪烁时机**框中，输入离散标记名或等于离散值的表达式。
- 4 在**闪烁属性**区域中，执行以下操作：
 - 单击**闪烁不可见**，将对象的闪烁方式设置为在窗口中消失然后重新出现。
 - 单击**带这些属性的闪烁可见**，将对象设置为保持可见，而在激活时改变颜色。
 - 单击**文本颜色**、**线条颜色**或**填充颜色**框，以选择用于对象的这些部分的颜色。此时出现调色板。

备注 如果选择与对象的填充颜色相同的填充闪烁颜色，则对象好像没有闪烁。

- 5 在**闪烁速度**区域中，设置对象的闪烁速度。单击**慢速**、**中速**或**快速**中的一个。
- 6 单击**确定**。

要设置 WindowMaker 的闪烁频率

- 1 在**特别菜单**上，指向**配置**，然后单击 **WindowViewer**。此时出现 **WindowViewer 属性**对话框。
- 2 在**闪烁频率**区域中，输入用于三种速度的毫秒数。

备注 对这些设置所作的任何更改都是全局性的，因此会影响应用程序中所有的闪烁速度。

3 单击**确定**。

启用可见性

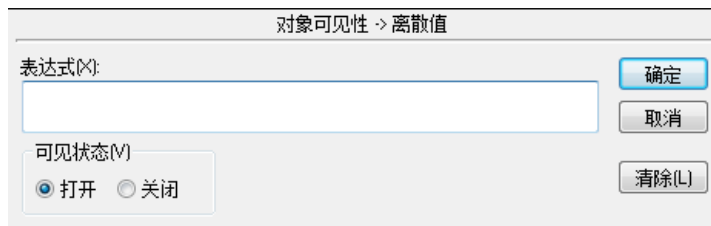
您可以使用可见性链接创建根据不同标记的值来隐藏对象的链接。通过使用可见性链接，您可以：

- 对象朝错误的方向移动时，通过隐藏它们，可以创造出对象只朝一个方向移动的印象。
- 创建正在移动的对象已经停止的印象。
- 使对象（如报警或错误消息）仅在激活时变得可见。

提示 离散表达式可以包含模拟标记，例如 `TankLevel >= 75`。在本例中，当标记 `TankLevel` 的值大于或等于 75 时，便可以在窗口中看到对象。

要创建可见性链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**其它**区域中，单击**可见性**。此时出现**对象可见性 -> 离散值**对话框。



- 3 在**表达式**框中，输入离散标记名或等于离散值的表达式。
- 4 选择对象的**可见状态**。如果选择**关闭**，则表达式的值为真时对象不可见。如果选择**打开**，则表达式的值为真时对象可见。
- 5 单击**确定**。

使对象失效

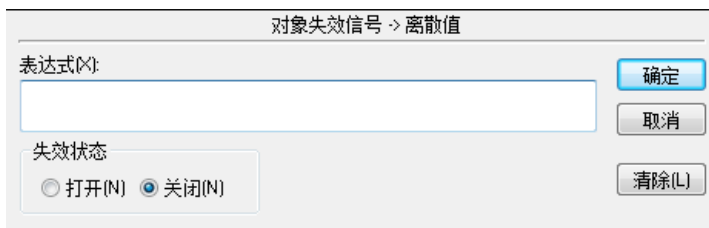
您可以使用失效链接在应用程序上施加另一层安全机制。例如，您可以根据操作员访问级别或姓名来使触控对象失效。或者，如果无人登录，则可以禁用按钮，以防外人擅自操作。

“打开”的失效状态表示只要表达式为真，该对象或按钮的触控功能便会关闭且处于不活动状态。

提示 离散表达式可以包含模拟标记名。例如，`TankLevel >= 75`。在本例中，当变量“TankLevel”的值大于或等于 75 时，对象失效。

要创建失效链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**其它区域**中，单击**失效**。此时出现**对象失效信号 -> 离散值**对话框。



- 3 在**表达式**框中，输入离散标记名或等于离散值的表达式。
- 4 在**失效状态**区域中，执行以下操作之一：
 - 选择**打开**，以便将失效状态设置为在离散标记或表达式为真期间对象无法激活。
 - 选择**关闭**以便移除失效状态，这样，在离散标记或表达式为真期间便允许对象运行。
- 5 单击**确定**。

配置工具提示

通过使用工具提示链接，可以创建“工具提示”，向用户提供屏幕上对象的有关信息。在指针移到对象上时，工具提示出现；而指针移开时，工具提示则消失。工具提示出现的持续时间以及出现的位置都由操作系统决定。

您可以为工具提示设置表达式或静态文本。

- 创建静态工具提示，以便在工具提示每次出现时显示相同的消息。
- 创建表达式工具提示，以便每次出现工具提示时，对表达式进行求值并显示为工具提示文本。

例如，在下面的示例表达式中，文本显示为 msgTooltipTag01 消息标记的当前值。

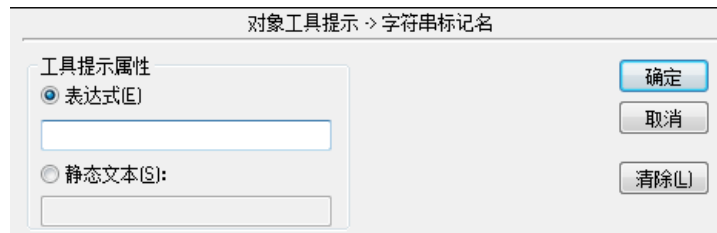
msgTooltipTag01

在本例中，字符串之后跟有 iTemp 标记的当前值，结果作为工具提示文本出现：

"Current temp. is " + StringFromIntg (iTemp,10)

要创建工具提示链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时打开**动画链接**对话框。
- 2 在**其它**区域中，单击**工具提示**。此时出现**对象工具提示 -> 字符串标记名**对话框。



- 3 在**工具提示属性**区域中，选择**表达式**或**静态文本**。
 - 如果选择**表达式**，请输入等于消息值的表达式。这可以是一个简单的消息标记名，也可以是一个较复杂的表达式。
 - 如果选择**静态文本**，请输入用作工具提示文本的静态消息，最多 131 个字符。
- 4 单击**确定**。

放置触控窗口

您可以使窗口在运行时出现在相对于触控对象的精确位置上。例如，操作员可以选择对象以查看状态、名称或其它与该对象链接的数据。在操作员选择对象时，通过单击或悬停鼠标，可以使窗口出现在指定的位置中。

通过使用带系统只读标记 `$ObjHor` 与 `$ObjVer` 的脚本函数 `ShowAt()` 或 `ShowTopLeftAt()`，可以相对于对象来放置窗口。您还可以在这些函数中使用固定的位置。

如果 Windows 的**显示属性**设置为 **Windows XP** 主题，则此功能在特定的情况下不稳定。

语法与以下类似：

```
ShowTopLeftAt (windowname, $ObjHor, $ObjVer);
```

其中

windowname: 要打开的窗口的名称。

\$ObjHor: 所选对象中心的水平位置。

\$ObjVer: 所选对象中心的垂直位置。

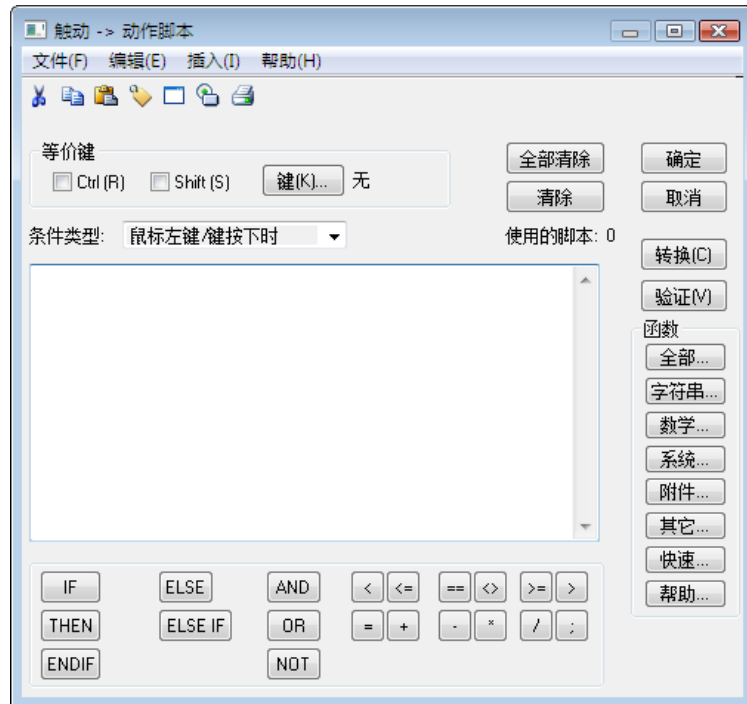
新窗口的左上角出现在所选对象的中心。

类似的脚本函数将打开窗口，使其中心位于所选对象的中心位置。语法与以下类似：

```
ShowAt (windowname, $ObjHor, $ObjVer);
```


要在所选对象的位置打开窗口

- 1 设计、命名并创建要出现的窗口。
- 2 使用鼠标右键单击对象以触发窗口，然后单击**动画链接**。此时出现**动画链接**对话框。
- 3 在**触动按钮**区域中，选择**动作**。此时出现**触动 -> 动作脚本**对话框。



- 4 根据之前定义的语法输入以下脚本之一：
`ShowTopLeftAt (windowname, $ObjHor, $ObjVer);`
 或者
`ShowAt (windowname, $ObjHor, $ObjVer);`
- 5 在**条件类型**框中，单击鼠标动作以打开窗口。
- 6 单击**确定**。

\$ObjHor 系统标记

包含对象（拥有焦点）中心的水平像素位置。

类别

系统

用法

\$ObjHor

数据类型

整型（只读）

另请参阅

\$ObjVer

\$ObjVer 系统标记

包含对象（拥有焦点）的中心的垂直像素位置。

类别

系统

用法

\$ObjVer

数据类型

整型（只读）

另请参阅

\$ObjHor

数据输入动画

要创建可供操作员交互操作的对象，请使用触动链接。触动链接可供操作员在系统中输入数据。例如，操作员可以使用键盘登录，打开或关闭阀门，输入新的报警设定点，或是开始或停止某个过程，等等。

触控对象拥有焦点时，它周围出现一个框。用户将光标移到该对象上，或是按 **TAB** 或方向键将焦点移到该对象时，对象便获得焦点。

如果期望用户使用 **TAB** 键来选择触控对象，请尝试将它们按水平模式排列。按 **Tab** 键从窗口顶端开始，从左到右，然后向下，将焦点从一个对象移到另一个对象。

如果触动链接对象包含顶端重叠放置的几个文本对象，则只显示顶端的文本对象。

要激活触控对象，操作员可以单击它、按指定的等价键、在对象框架出现时按 **Enter** 键，或是实际触摸它（如果使用触摸屏显示设备）。

您可以创建九种类型的用户输入触动链接：

触动链接	动作
用户输入	• 离散
	• 模拟
	• 字符串
游标	• 垂直
	• 水平
按钮	• 离散值
	• 动作
	• 显示窗口
	• 隐藏窗口

文本字段用于输入时，文本会在按键时出现在屏幕上。

如果不希望在输入时出现文本，请选择该链接的配置面板中的**仅输入**选项。

启用离散输入

您可以设计操作员输入触动链接，来将离散标记的值从一个状态更改成另一个状态。例如，要打开或关闭某个泵，请使用离散链接。

要创建离散输入链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**触动链接**区域中的**用户输入**下，选择**离散**。此时出现**输入 -> 离散标记名**对话框。

输入 -> 离散标记名

标记名(G):

等价键

☐ Ctrl (T) ☐ Shift (S) 无

给用户的消息(M):

置位提示(E) 打开消息(N):

复位提示(R) 关闭消息(F):

☐ 仅输入(I)

- 3 在**标记名**框中，输入离散标记的名称。
- 4 作为可选项，在**等价键**区域中，指定链接的等价键。如需有关详细信息，请参阅第 93 页的“创建键盘快捷键”。
- 5 配置离散输入的详细资料。执行以下操作：
 - 在**给用户的消息**框中，输入要出现在**输入**对话框中的消息。
 - 在**置位提示**与**复位提示**框中，输入要出现在按钮上的消息，操作员可以单击这些按钮来打开或关闭离散值。
 - 在**打开消息**与**关闭消息**框中，输入在值打开或关闭时要出现在与该对象关联的文本字段中的消息。
- 6 选择**仅输入**复选框，以避免输入内容出现在与对象关联的文本字段中。
- 7 单击**确定**。

启用模拟输入

通过使用模拟输入链接，您可以创建操作员用于输入实数值（如报警设定点或传送带速度）的对象。

要创建模拟输入链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**触动链接**区域中的**用户输入**下，选择**模拟**。此时出现**输入 -> 模拟标记名**对话框。

- 3 在**标记名**框中，输入模拟标记的名称。
- 4 在**等价键**区域中，指定链接的等价键。如需有关详细信息，请参阅第 93 页的“创建键盘快捷键”。
- 5 配置模拟输入的详细资料。执行以下操作：
 - 如果希望在屏幕上显示数字小键盘来输入新的值，请在**数字小键盘?**区域中单击**是**。在**给用户的信息**框中，输入希望出现在数字小键盘中的提示消息。
 - 在**最小值**与**最大值**框中，输入标记的最小值与最大值。
- 6 选择**仅输入**复选框，以避免输入内容出现在与对象关联的文本字段中。
- 7 单击**确定**。

启用字符串输入

通过使用字符串输入链接，您可以创建供用户输入字符串（如批次名、操作员 ID 或口令）的对象。

要创建字符串输入链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**触动链接**区域中的**用户输入**下，选择**字符串**。此时出现**输入 -> 字符串标记名**对话框。

- 3 在**标记名**框中，输入消息标记的名称。
- 4 作为可选项，配置等价键与 / 或键盘。
 - 在**等价键**区域中，指定链接的等价键。如需有关详细信息，请参阅第 93 页的“创建键盘快捷键”。
 - 如果希望显示键盘来输入新的值，请在**数字小键盘?**区域中单击**是**。在**给用户的消息**框中，输入要出现在键盘中的消息。
- 5 在**回显字符?**区域中，选择在用户输入字符串时输入框中是否出现字符。
 - 单击**是**以便在输入框中显示输入的文本。
 - 单击**否**以便不显示输入的文本。
 - 单击**口令**以使用“掩码”字符代替输入的文本。在**口令字符**框中，输入掩码字符。选择**加密**复选框以加密口令。

重要 口令加密仅在 InTouch HMI 环境中有效。如果希望将字符串传递给外部安全系统（例如操作系统或 SQL Server 数据库），请不要给字符串加密。外部安全系统无法读取加密的口令字符串，因此访问会失败。

- 6 选择**仅输入**复选框，以避免输入内容出现在与对象关联的文本字段中。
- 7 单击**确定**。

启用游标

通过使用游标触动链接，您可以创建供用户来回拖动的对象。用户移动对象时，它会改变与之链接的标记的值。您可以将对象链接到水平或垂直游标上。

在将对象链接到游标上时，请设置参考位置（即对象上的一点，光标用它来锁定对象）。

您可以在一个对象上同时使用水平与垂直链接，这样这两个模拟标记的值便会同时改变。

要创建水平游标链接

- 1 使用鼠标右键单击对象，然后单击**动画链接**。此时出现**动画链接**选择对话框。
- 2 在**游标**区域中，单击**水平**。此时出现**水平游标**对话框。



The image shows a dialog box titled "水平游标" (Horizontal Slider). It contains the following fields and controls:

- 标记名(T):** A text input field for the marker name.
- 属性 (Attributes):**
 - 在左端(L):** A numeric input field with the value "0".
 - 在右端(R):** A numeric input field with the value "100".
 - 向左(F):** A numeric input field with the value "0".
 - 向右(B):** A numeric input field with the value "100".
- 参考位置(O):** A section with three radio buttons: **左端** (selected), **中间**, and **右端**.
- Buttons:** "确定" (OK), "取消" (Cancel), and "清除(C)" (Clear).

- 3 在**标记名**框中，输入模拟标记的名称。
- 4 在**属性**区域中，执行以下操作：
 - a 在**在左端**框中，输入游标位于最左端时的标记的值。
 - b 在**在右端**框中，输入游标位于最右端时的标记的值。
 - c 在**向左**框中，输入游标可向左移动的像素数。
 - d 在**向右**框中，输入游标可向右移动的像素数。
- 5 在**参考位置**区域中，单击对象上光标将锁定的位置。
- 6 单击**确定**。

要创建垂直游标链接

- 1 使用鼠标右键单击对象，然后单击**动画链接**。此时出现**动画链接**对话框。

- 2 在**游标**区域中，单击**垂直**。此时出现**垂直游标**对话框。



垂直游标对话框的界面如下：

- 标题：垂直游标
- 标记名(G): 输入框
- 确定按钮
- 取消按钮
- 清除(C)按钮
- 属性区域：

	值		垂直移动
在顶端(T):	0	向上(U):	0
在底端(B):	100	向下(D):	100
- 参考位置(O):
 - ☐ 顶端
 - ☐ 中间
 - ☒ 底端

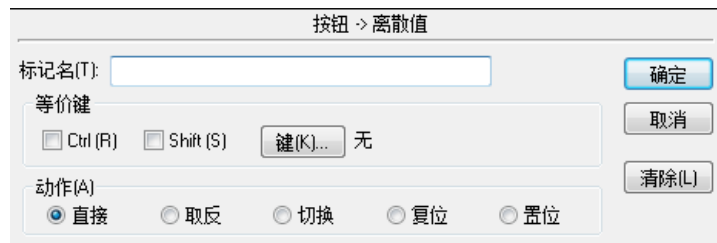
- 3 在**标记名**框中，输入模拟标记的名称。
- 4 在**属性**区域中，执行以下操作：
- a 在**在顶端**框中，输入游标位于最顶端时的标记的值。
 - b 在**在底端**框中，输入游标位于最底端时的标记的值。
 - c 在**向上**框中，输入游标可向上移动的像素数。
 - d 在**向下**框中，输入游标可向下移动的像素数。
- 5 在**参考位置**区域中，单击对象上光标将锁定的位置。
- 6 单击**确定**。

启用按钮

通过使用触动按钮链接，您可以创建启动动作脚本的触控对象。动作脚本可以将标记设置为指定的值、启动与控制其它应用程序、执行函数等。

要创建离散值触动链接

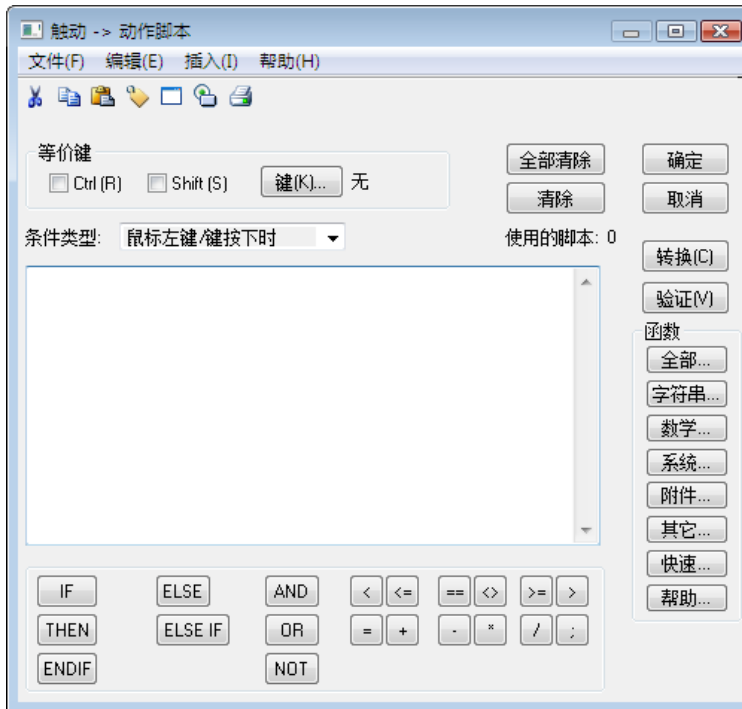
- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**触动按钮**区域中，单击**离散值**。此时出现**按钮 -> 离散值**对话框。



- 3 在**标记名**框中，输入离散标记名。
- 4 单击**键**以指定链接的等价键。
- 5 在**动作**区域中，单击以下类型之一：
 - 单击**直接**以便只要按下或按住按钮，便将值设置为 1。释放按钮时，该值自动复位成 0。
 - 单击**取反**以便在按下或按住按钮时，将值设置为 0。释放按钮时，该值自动复位成 1。
 - 单击**切换**以反转离散标记的状态。
 - 单击**复位**以便将该值设置为 0。
 - 单击**置位**以便将该值设置为 1。
- 6 单击**确定**。

要创建动作脚本链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现“动画链接”对话框。
- 2 在**触动按钮**区域中，单击**动作**。此时出现**触动 -> 动作脚本编辑器**。



- 3 在**条件类型**列表中，选择一种脚本类型。如需有关动作脚本类型的详细信息，请参阅 *InTouch® HMI 脚本与逻辑指南* 中的第 3 章“脚本触发器”。

备注 如果将等价键链接指定给动作按钮和用于某个键脚本的同一个键，则所指定的等价键链接的优先级高于该键脚本。

- 4 在**脚本编辑器**窗口中，输入对象激活时要执行的脚本。
- 5 单击**确定**。

打开与关闭窗口

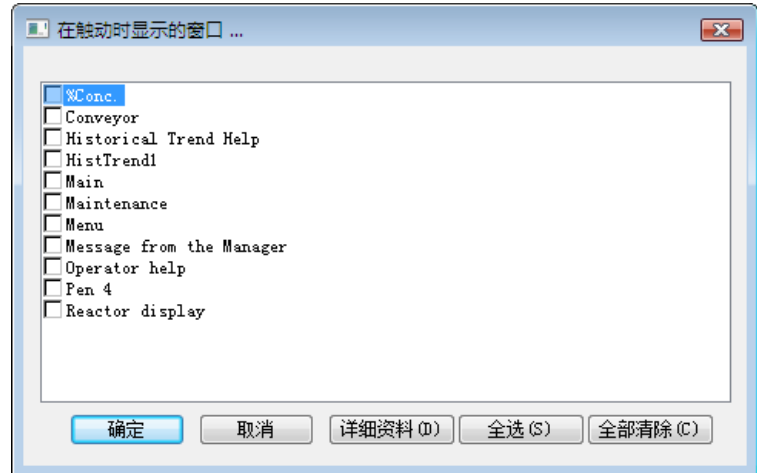
通过使用显示窗口与隐藏窗口链接，可以创建打开与关闭其它 InTouch 窗口的触动链接。

您可以给对象编程以便一次打开多个窗口。不过如果将链接编程为打开多个窗口，请注意任何窗口交叉与各窗口类型。

如果其中一个打开的窗口是替换类型的窗口，并与另一个打开的窗口交叉，则另一个窗口会在打开之前关闭。

要创建显示（或隐藏）窗口链接

- 1 使用鼠标右键单击对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 2 在**触动按钮**区域中，单击**显示窗口**或**隐藏窗口**。此时出现在**触动时显示的窗口**或在**触动时隐藏的窗口**对话框。



- 3 选择要打开或隐藏的窗口。
- 4 单击**确定**。

配置屏幕键盘

在计算机没有连接键盘的情况下，屏幕键盘可供操作员进行输入。

您可以指定三种屏幕键盘之一。

- 标准 InTouch 键盘或数字键盘。这是缺省键盘。
- Windows 系统键盘。Windows 键盘是功能齐全的 QWERTY 型键盘，带有功能键、打印屏幕键、数字锁定键、方向键等。
- 可调整大小的键盘或数字键盘。这种键盘的大小可以在运行时调整。

通过在脚本中使用 `DialogStringEntry()` 与 `DialogValueEntry()` 函数，您可以打开键盘。如需有关详细信息，请参阅第 85 页的“`DialogStringEntry()` 函数”与第 86 页的“`DialogValueEntry()` 函数”。

要配置屏幕键盘类型

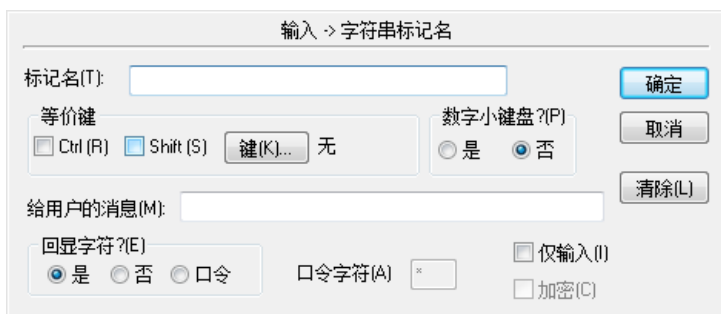
- 1 在特别菜单上，指向**配置**，然后单击 **WindowViewer**。此时出现 **WindowViewer 属性**对话框。
- 2 在**键盘**区域中，选择需要的键盘类型。



- 3 如果选择**可调整大小的键盘**，请单击**选项**以选择键盘的字体、位置、尺寸等属性。
- 4 单击**确定**。

要显示屏幕键盘

- 1 配置屏幕键盘的类型。
- 2 使用鼠标右键单击文本对象，然后选择**动画链接**。此时出现**动画链接**对话框。
- 3 在**触动链接**区域中的用户输入下，选择**字符串**。此时出现**输入 -> 字符串标记名**对话框。



- 4 在**数字小键盘?**区域中，选择**是**。
- 5 单击**确定**。

DialogStringEntry() 函数

在屏幕上显示字母数字键盘，供操作员在“标记名字典”中更改消息标记的当前字符串值。

类别

其它

语法

```
[Result=]DialogStringEntry(MessageTag_Text, UserPrompt_Text);
```

参数

MessageTag_Text

要修改的消息标记的名称。此值是一个字符串值。指定带英文引号括起的标记名，或使用不带英文引号的 .Name 点域。您还可以将消息标记用作指针。

UserPrompt_Text

要在键盘顶部显示的用户消息。

返回值

返回以下整数值之一：

- 0 = 已按过“取消”。
- 1 = 已按过“确定”。
- 1 = 内部错误。
- 2 = 无法启动。
- 3 = 未定义标记名。
- 4 = 标记名并非“消息”型。
- 5 = 无法写入。

附注

此函数主要用在包含触摸屏的应用程序中。

示例

```
Errmsg=DialogStringEntry(MyMessageTag.Name, "Enter a new string...");
```

```
Errmsg=DialogStringEntry("MyMessageTag","Enter a new string...");
```

例如，以下脚本打开一个字母数字键盘，在键盘顶部显示消息 "Enter a new string..." 期间，允许修改 MyMessageTag：

```
MessageTagX="MyMessageTag"; {assign the string MyMessageTag  
(which is actually the tagname to be modified) to the Memory  
Message tagname MessageTagX}
```

```
MessageDisplay="Enter a new string..."; {assign the new message  
string to the Memory Message tagname MessageDisplay}
```

```
Errmsg=DialogStringEntry(MessageTagX, MessageDisplay); {quotes  
are not required because MessageTagX was defined as a Message  
tagname}
```

另请参阅

DialogValueEntry()

DialogValueEntry() 函数

在屏幕上显示数字小键盘，供用户更改离散、整型或实型标记名的当前值。

类别

其它

语法

```
[Result=] DialogValueEntry(ValueTag_Text, LowLimit, HighLimit,  
    UserPrompt_Text);
```

参数

ValueTag_Text

要修改的离散、整型或实型标记的名称。此值是一个字符串值。指定带英文引号括起的标记名，或使用不带英文引号的.Name 点域。您还可以将消息标记用作指针。

LowLimit

标记可以使用的最小值。（这应该是大于或等于标记名的最小值、最小原始值或最小工程单位的定义，具体视适用情况而定）。

HighLimit

标记可以使用的最大值。（这应该是大于或等于标记名的最大值、最大原始值或最大工程单位的定义，具体视适用情况而定）。

UserPrompt_Text

要在数字键盘顶部显示的用户消息。

返回值

返回以下整数值之一：

0 = 已按过“取消”。

1 = 已按过“确定”。

-1 = Highlimit<=Lowlimit。

-2 = 无法启动。

-3 = 未定义标记名。

-4 = 标记名不是“离散”、“整型”或“实型”。

-5 = 写入失败。

附注

此函数主要用在包含触摸屏的应用程序中。

示例

```
Errmsg=DialogValueEntry(MyIntegerTag.Name,  
    MyIntegerTag.MinEU, MyIntegerTag.MaxEU, "Enter a new  
    value...");
```

```
Errmsg=DialogValueEntry("MyIntegerTag", -100, 100, "Enter a new
value...");
```

例如，以下脚本弹出一个数字键盘，在数字键盘顶部显示消息 "Enter a new value..." 期间，允许修改 MyIntegerTag，其最大与最小极限值分别是 -100 与 100：

```
TagnameX="MyIntegerTag"; {assign the string MyIntegerTag (which is
actually the tagname to be modified) to the Memory Message
tagname TagnameX}
```

```
Min=-100; {assign the minimum value allowed for the tagname to the
Memory Real/Integer tagname Min}
```

```
Max=100; {assign the minimum value allowed for the tagname to the
Memory Real/Integer tagname Max}
```

```
MessageDisplay="Enter a new value..."; {assign the new message string
to the Memory Message tagname MessageDisplay}
```

```
Errmsg=DialogValueEntry(TagnameX, Min, Max, MessageDisplay);
{quotes are not required because TagnameX was defined as a
Message tagname. By assigning a Discrete, Integer or Real tagname
to TagnameX, the function will modify that assigned tagname}
```

另请参阅

DialogStringEntry()

常见动画任务

常见动画任务包括选择标记、创建键盘快捷键、更改标记名引用以及替换占位符标记。

选择标记或属性

使用**选择标记**对话框（也称为标记浏览器）来选择：

- 本地或远程 InTouch 应用程序中定义的标记。
- 使用 Attribute Browser（属性浏览器）来选择 ArchestrA 对象属性。

双击任何需要输入标记名的文本框，以打开**选择标记**对话框。

选择 InTouch 标记

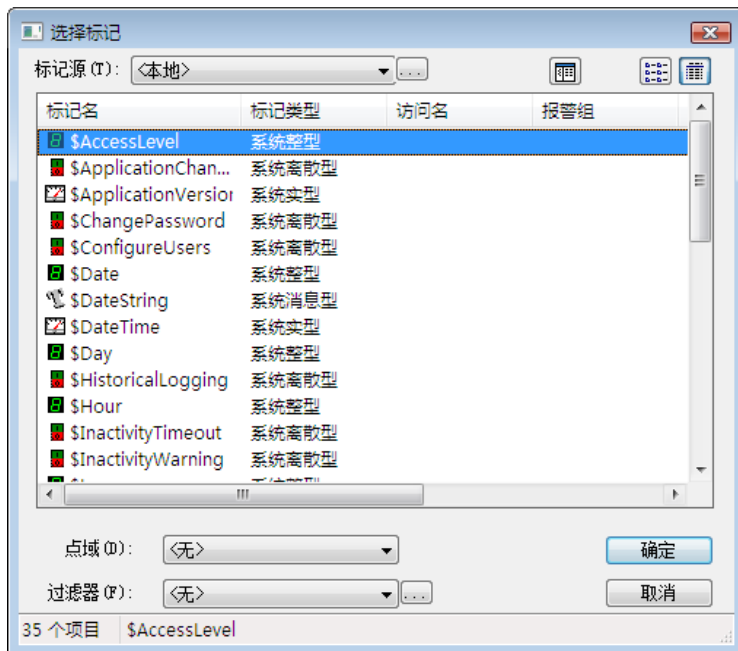
您可以选择在本地或远程 InTouch 应用程序中定义的标记。在支持“标记名字典”界面的任何标记源中，都可以创建对标记的引用。

例如，远程标记引用可供您的应用程序从“I/O 服务器”访问数据，而不必在本地“标记名字典”中创建标记。

您可以为选择的每个 InTouch 标记设置点域。点域可以访问、监视及修改标记属性。如果不选择点域，则使用 .Value 点域。如需有关点域的详细信息，请参阅 *InTouch® HMI 数据管理指南* 中的第 4 章“使用标记点域来查看或更改标记属性”。

要选择 InTouch 标记

- 1 双击任何需要输入标记名的文本框，以打开**选择标记**对话框。



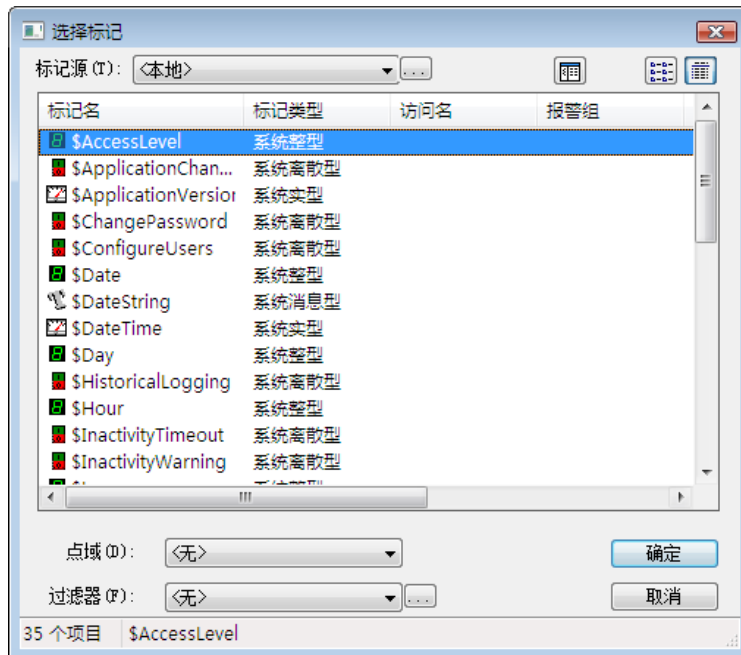
- 2 在**标记源**列表中，单击标记源的名称，或是单击浏览按钮以定义要使用的新标记源。
如需有关定义标记源的详细信息，请参阅 *InTouch® HMI 数据管理指南* 中的第 5 章“使用 I/O 进行数据访问”。
- 3 在**过滤器**列表中，单击过滤器以减少在窗口中显示的标记名数。要定义过滤器，请单击省略号按钮。如需有关详细信息，请参阅第 90 页的“创建标记过滤器”。
- 4 在窗口中选择一个标记名。
您可以更改**选择标记**对话框中标记名的显示方式。如需有关详细信息，请参阅第 92 页的“更改选择标记对话框中的视图”。
- 5 在**点域**列表中，单击一个点域以将它附加到所选的标记名上。
点域可以访问、监视及修改标记属性。如果不选择点域，则使用 .Value 点域。单击**确定**。

选择 ArchestrA 对象属性

您可以选择与 ArchestrA 对象关联的属性。为此，必须将包含对象的 Galaxy 添加为 InTouch 应用程序的数据源。如需有关数据源的详细信息，请参阅 *InTouch® HMI 数据管理指南* 中的第 5 章“使用 I/O 进行数据访问”。

要选择对象属性

- 1 双击任何需要输入标记名的文本框，以打开**选择标记**对话框。



- 2 在**标记源**列表中，单击使用 Galaxy 的标记源的名称，或是单击浏览按钮以定义要使用的新标记源。此时出现 **Attribute Browser**（属性浏览器）对话框。
- 3 使用 **Attribute Browser**（属性浏览器）来查找与选择 ArchestrA 对象属性。如需有关详细信息，请参阅“Wonderware 应用程序服务器”文档。
- 4 单击**确定**以关闭 **Attribute Browser**（属性浏览器）。属性引用出现在要求使用标记名的文本框中。

备注 要从 **Attribute Browser**（属性浏览器）返回到**选择标记**对话框，请单击 **Attribute Browser**（属性浏览器）右下角的 **Back**（返回）按钮。

创建标记过滤器

如果“标记名字典”中包含大量的标记，则查找动画的标记会非常麻烦。例如，您可能只希望查看指定给特定“访问名”或“报警组”的标记。您可以配置一个过滤器，仅使一部分标记显示在**选择标记**对话框中。

过滤器中可以使用以下通配符表达式。

- 多字符通配符是星号 (*)。例如, "Asyn*" 将搜索以字符 "Asyn" 开头的所有标记名。
- 单字符通配符是问号 (?)。例如, "Tag?" 过滤器搜索以 "Tag" 开头的所有四个字符的标记名。“Tag*” 过滤器搜索以 “Tag” 开头的所有标记名。
- 过滤器可以使用有效的标记名字符与两个通配符的任意组合。有效的标记名字符是: A-Z、a-z、0-9、!、@、-、#、\$、%、_ 及 &。

要定义搜索过滤器

- 1 在**选择标记**对话框中, 单击**过滤器**列表旁边的省略号按钮。此时出现**定义标记过滤器**对话框。



- 2 在**过滤器名**框中, 输入过滤器名。



- 3 在**过滤器选项**区域中，配置过滤器准则。执行以下任何一项操作：
- 在**标记名**框中，输入标记名。
 - 在**标记类型**框中，输入标记类型。
 - 在**访问名**框中，输入本地访问名。
 - 在**报警组**框中，输入报警组名。
 - 在**注释**框中，输入注释表达式。
- 4 单击**确定**。此时**过滤器名**出现在**选择标记**对话框的**过滤器**列表中。您可以选择过滤器以显示符合该过滤器准则的标记。

要删除搜索过滤器

- 1 在**过滤器名**框中，单击过滤器。
- 2 单击**删除**。

更改选择标记对话框中的视图

选择标记对话框有三种不同的视图：列表视图、详细视图及目录树视图。

要查看	单击	描述
列表视图	列表视图按钮 	根据每个标记名的类型，不同的小图标会出现在标记名的旁边。
详细视图	详细视图按钮 	您会看到相同的图标与标记名，还有“标记类型”、“访问名”、“报警组”以及“注释”。通过单击列的标题给列表排序。
目录树视图	目录树视图图标 	目录树视图以两种视图显示标记名。您可以访问任何 SuperTag 模板中的成员标记名。

创建键盘快捷键

通过使用等价键链接，可以指定使用键盘上某个特定的键来激活某些动画链接。仅当包含此链接的对象可见或被选定时，等价键才可操作。如果该对象包含可见性或失效链接，则在该对象失效时，等价键不会激活。

您可以在多个窗口中定义相同的键。不过，只有最近打开的窗口中定义的键才会是活动的。在窗口重叠的情况下，顶部窗口的键才是活动的。

备注 如果将活动窗口中的任何对象或动作按钮指定给用于键动作脚本的同一个键，则活动窗口中该键的等价键链接的执行优先级高于键动作脚本。

支持等价键的动画链接出现在链接对话框中的**等价键**区域。

键链接只支持功能键 1-16。如果使用包含 16 个以上功能键的自定义键盘，则需要从厂商处获取设备驱动程序，以允许访问系统上的扩展功能键。

要将键指定给链接

- 1 根据正在配置的链接类型，打开相应的**动画链接**对话框。
- 2 如果希望操作员在按下等价键的同时按住 Ctrl 与 Shift 之一或同时按住这两者，请选择 **Ctrl** 与 / 或 **Shift**。
- 3 单击**键**。此时出现**选择键**对话框。
- 4 单击要指定给链接的键。此时再次出现**动画链接**对话框，并在**键**按钮旁边显示所选键的名称。
- 5 单击**确定**。

更改标记名引用

给对象创建副本时，所得副本是原件的完全拷贝，包括链接、动画、脚本，等等。不过，如果要在副本对象上使用不同的标记，则必须替换标记。

您可以选择与替换任何对象的标记，也可以选择多个对象并同时替换它们的标记。

如果系统许可证支持有限数目的标记名，则将本地标记转换为远程标记引用，以减少本地“标记名字典”中定义的标记数。

替换标记对全部标记与引用都有效。

要将一个标记替换为另一个标记

- 1 选择与要更改为另一个标记的标记关联的对象。
- 2 在**特别**菜单上，单击**替换标记**。此时出现**替换标记名**对话框。
- 3 在**新名**框中，输入新的标记名。
 - 如果双击**新名**框中的标记，则在“标记名字典”中出现它的定义。
 - 如果清除该标记名，然后双击空白框，则出现**选择标记**对话框。
- 4 单击**确定**。与该对象关联的标记会自动改变。

转换占位符标记

如果从当前应用程序导入或导出一个窗口或 QuickScript 时，所有与该窗口或 QuickScript 关联的标记都会随窗口一起转移。

本地标记不会自动添加到应用程序数据库中。相反，它们会自动标为占位符标记。

远程标记引用不会受到影响，它们不会标为占位符标记。

您必须将占位符标记转换为现有的本地标记，在新的应用程序标记名字典中定义它们，或使它们成为远程标记引用。

请注意标记之前的占位符：**?d:**、**?i:**、**?m:** 及 **?r:**。它们指明了标记原先定义的类型。

占位符符号	标记名类型
d	离散
i	整型
m	消息
r	实型

备注 远程标记引用不会显示为占位符，而是显示为远程标记引用，例如：**PLC2:Temperature**。

在**替换标记名**对话框中，您可以使用多种方法将占位符标记转换为本地标记。如需有关详细信息，请参阅 *InTouch® HMI 应用程序管理与扩展指南* 中的第 4 章“导出与导入标记定义、窗口以及脚本”。

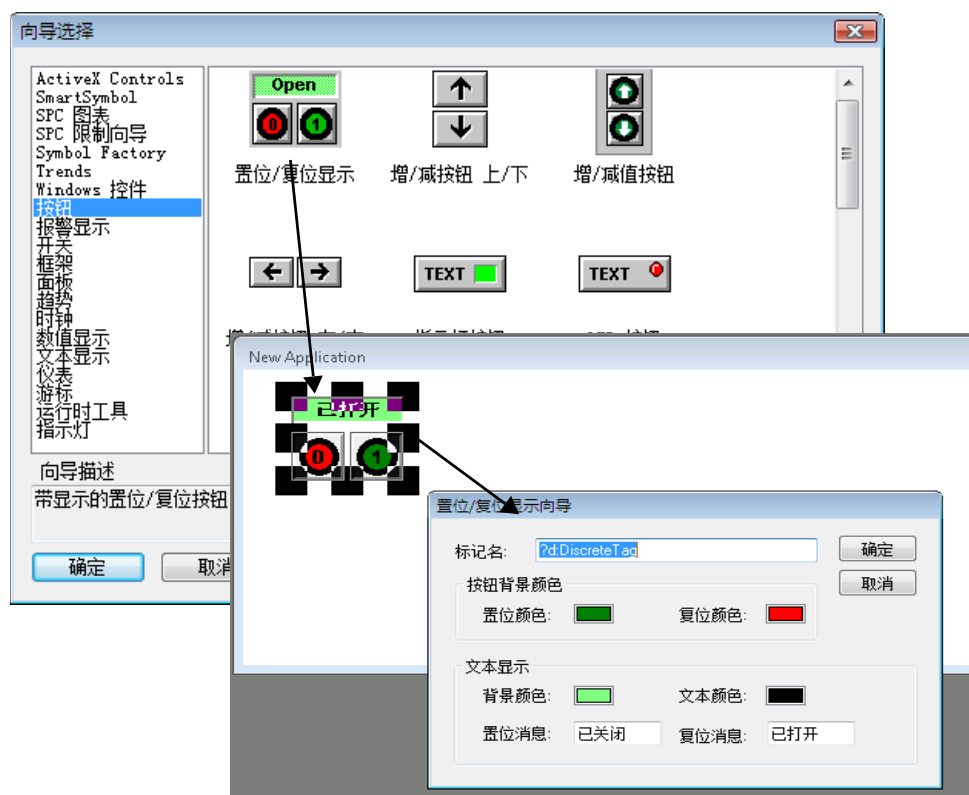
提示 如果对标记名进行了手工转换，不再需要原始的“标记名字典”中定义的原始标记，则可以更新标记名使用计数，然后删除未使用的标记。

通过从另一个应用程序导入窗口或 QuickScript，并将与动画链接或 QuickScript 关联的所有标记名转换为远程标记名引用，则不必在本地“标记名字典”定义单个标记名，即可创建一个应用程序，立刻接收来自数以百计的远程标记名的数据。

第 5 章

向导

向导是一组预先设计、构建及编程的对象，您只需要选择这些对象、将它们放置到应用程序中并进行相应的配置即可。



使用向导

通过使用向导，不必花费时间绘制去对象的各个单独组件、输入对象的值范围或是给对象设置动画效果。

- 您可以从“向导 /ActiveX 工具栏”中选择向导。
- 您可以通过在配置对话框中输入标记与 QuickScript 来配置向导。
- 将所选的向导粘贴到窗口中，然后双击它时，会出现配置对话框。

例如，对于游标向导，配置项目包括要使用的标记名、游标的最小与最大范围标签以及填充颜色，等等。提供所需的配置信息之后，该向导便已经可以使用。

您也可以自行开发复杂的向导来提供“幕后”操作。这些操作可能包括创建完整的显示窗口、创建或转换数据库、导入 AutoCAD 图形，以及配置其它应用程序。

在创建自己的向导之前，应该先研究一下一些 ArchestrA 符号，这些符号不仅提供向导功能，而且还不必编程。

向导类型

向导的类别显示在向导选择对话框中。



“趋势对象”与“Windows 控件向导”是有一些独特参数的特殊向导。如需有关详细信息，请参阅第 102 页的“趋势对象”与第 103 页的“Windows 控件向导”。

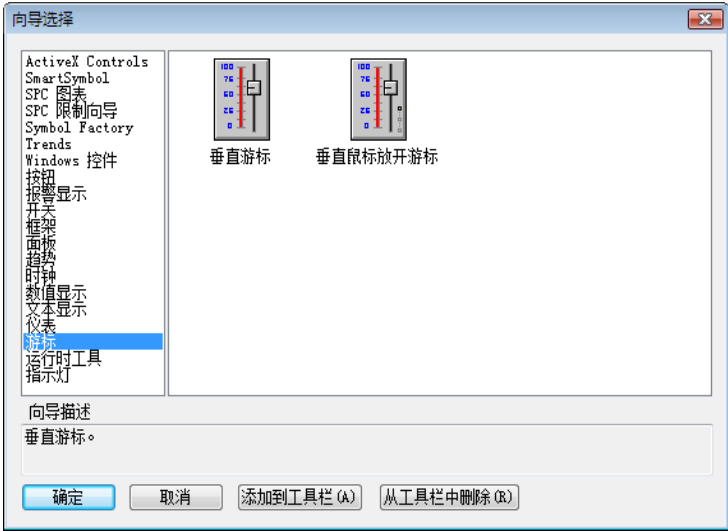
将向导添加到工具栏

您可以将常用的向导添加到“向导 /ActiveX 工具栏”以便快速访问。

要将向导添加到向导 /ActiveX 工具栏



- 1 单击向导 /ActiveX 工具栏中的向导按钮。此时出现向导选择对话框。



- 2 在左侧窗格中，选择向导类别，如游标。
- 3 在右侧窗格中，选择向导，然后单击添加到工具栏。此时向导按钮出现在工具栏中。

要从工具栏中删除向导



- 1 单击向导 /ActiveX 工具栏中的向导按钮。此时出现向导选择对话框。
- 2 单击从工具栏中删除。此时出现从工具栏删除向导对话框。
- 3 选择要删除的向导，然后单击确定。

粘贴向导实例

您可以将向导实例放入窗口中。

要将向导放入窗口



- 1 单击向导 /ActiveX 工具栏中的“向导”按钮。此时出现向导选择对话框。在左侧窗格中，选择向导类别。
- 2 在右侧窗格中，选择向导。
- 3 单击确定。此时对话框关闭，光标变成弯头符号。
- 4 单击要放置该向导的位置。

配置向导

在应用程序中放置向导之后，双击该向导以配置其属性。此时出现一个根据所选的向导而定制的属性对话框。

如需有关每种特定类型的向导的详细信息，请参阅向导的“帮助”（如果有）。

对向导执行标准操作

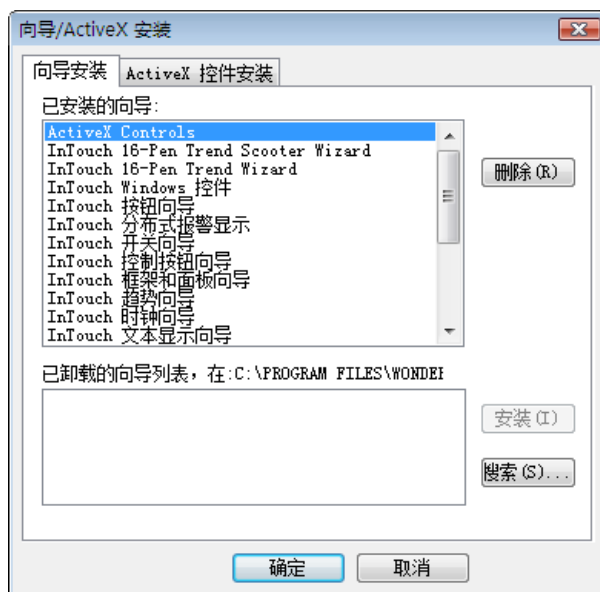
像处理其它对象那样，您可以使用相同的方法来剪切、复制、粘贴、删除向导，以及给向导创建副本，结果也与其它对象相似。

安装与删除向导

您必须给 WindowMaker 安装向导，以便在应用程序中使用它。从 WindowMaker 中删除向导时，它并没有从计算机上彻底删除。

要安装向导

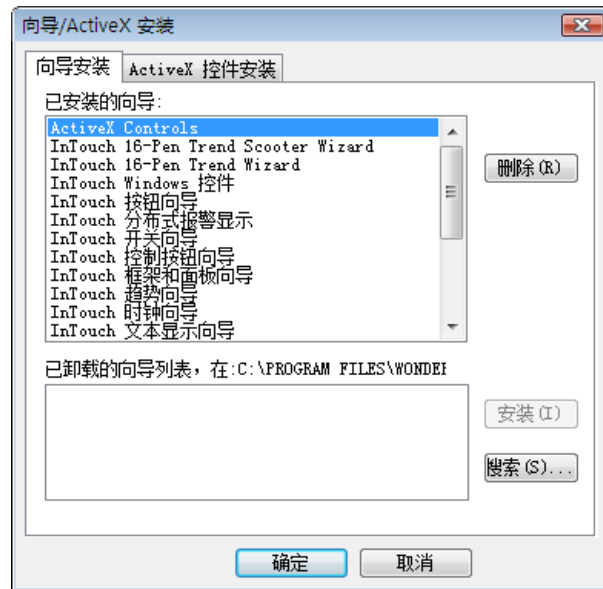
- 1 在**特别**菜单上，指向**配置**，然后单击**向导 /ActiveX 安装**。此时出现**向导 /ActiveX 安装**对话框。



- 2 请从**已卸载的向导列表**中选择向导，然后单击**安装**。

要删除向导

- 1 在**特别**菜单上，指向**配置**，然后单击**向导 /ActiveX 安装**。此时出现**向导 /ActiveX 安装**对话框。



- 2 在**已安装的向导**列表中，选择要删除的向导，然后单击**删除**。此时出现一个消息框，要求确认删除。

提示 通过在按住 **SHIFT** 或 **CTRL** 键的同时进行单击，可以选择多个向导。

- 3 单击**是**以删除向导。该向导出现在**已卸载的向导**列表中。

要从其它目录导入向导

- 1 在**特别**菜单上，指向**配置**，然后单击**向导 /ActiveX 安装**。此时出现**向导 /ActiveX 安装**对话框。
- 2 单击**搜索**。此时出现**浏览文件夹**对话框。
- 3 浏览到包含要安装的向导的目录，然后单击**确定**。此时再次出现**向导安装**对话框，且导入的向导出现在**已卸载的向导**列表中。

趋势对象

趋势对象是绘制标记值随时间变化的一些向导。

趋势对象主要有三种类型：

- “实时”趋势，最多使用四个标记进行实时图表绘制。
- “历史”趋势，最多使用过去一个时段内的八个标记进行图表绘制。
- “16 笔趋势”，最多使用十六个标记绘制实时与历史数据图表。

对于可以在窗口中放置的趋势对象（实时或历史），没有数量上的限制。

使用以下项目配置趋势对象：

- 时间跨度
- 值范围
- 网格分辨率
- 时间标签与值标签的位置
- 笔与颜色

在使用趋势向导之前，必须为每个标记启用记录功能以便进行跟踪，并且也要在 InTouch 应用程序中启用记录功能。

要启用标记的记录功能

- 1 从**标记名字典**中选择标记，然后选择**记录数据**。
- 2 如果之前没有这样做，则需要在 InTouch 应用程序中启用记录功能。
 - a 在**特别菜单**上，指向**配置**，然后单击**历史记录**。此时出现**历史记录属性**对话框。
 - b 选择**启用历史记录**复选框，然后单击**确定**。

如需有关配置与使用趋势对象的详细信息，请参阅 *InTouch® HMI 数据管理指南* 中的第 10 章“绘制标记数据的趋势”。

Windows 控件向导

“Windows 控件向导”是一组用户界面对象，如下拉列表、组合框、单选框或复选框。

“Windows 控件向导”可以向用户显示一组预定义的选项。例如，您可能会创建一个下拉列表，列出过程、配方或操作员 ID。您也可以启用和禁用指定的控件。您甚至可以加载与修改下拉列表的内容。

通过 QuickScript 函数而不是动画链接表达式，可以访问“Windows 控件向导”的运行时属性。

“Windows 控件向导”具有一些类似于 *InTouch* 标记点域的属性。它们既可以是可读写的，也可以是只读的。有些属性可以在开发时访问，有些则在运行时访问。它们使用 `ControlName.x` 的形式进行标识，其中 *x* 是属性。

例如，如果 Windows 控件的 `.Visible` 属性等于 0，则该控件在窗口中不可见。同 *InTouch* 标记类似，`.Value` 是“Windows 控件向导”的缺省属性。

备注 如果使用 *Archestra* 符号，则可以使用一组基于 .NET 的更强大灵活的 Windows 控件。

创建与配置 Windows 控件

创建与配置“Windows 控件向导”时，请注意以下事项：

- “Windows 控件向导”不能相互重叠。
- 每个“Windows 控件向导”都有唯一的控件名，这并不添加到标记计数中。
- 指定给列表框或组合框的标记初始值并不初始化列表框或组合框的值。您必须在脚本中使用 `SetPropertyX` QuickScript 函数，以指定需要与缺省值不同的初始值。

提示 您可以像操作其它向导一样，将“Windows 控件向导”粘贴到窗口中。要取得最佳显示效果，请为 Windows 控件选择灰色背景。如果背景颜色无法设置为灰色，请在“Windows 控件向导”背后放置一个灰色的“面板向导”。

对于每个“Windows 控件向导”，必须指定一个字母数字控件名，其中首字符是字母。下划线允许使用，但不允许使用其它特殊字符。例如，允许使用“`Checkbox_1`”，但不允许使用“`Checkbox#1`”。

您可以使用 *InTouch* QuickScript 配置“Windows 控件向导”。

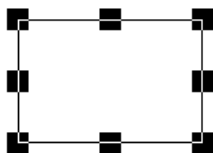
创建文本框

您可以在应用程序中使用文本框，以供操作员输入文本字符串。

要创建与配置文本框

1 创建与放置文本框。执行以下操作：

- a 在**向导选择**对话框中，选择 **Windows 控件向导**。此时出现各个控件向导图标。
- b 双击文本框图标。此时再次出现应用程序窗口，光标变成左弯头符号。
- c 在应用程序窗口中单击要放置该向导的位置。此时出现拐角处带浓黑色手柄的文本框向导。



- d 根据应用程序拖动该向导并调整其大小。

2 双击该向导。此时出现**文本框控件**对话框。



3 配置该对话框。执行以下操作：

- a 在**控件名**框中输入控件名，如 *TextBox_1*。
- b 在**标记名**框中输入内存消息标记名，如 *New_Value*。
- c 在**属性**区域中，选择**允许输入**与**可见**。

4 单击**确定**。此时**文本框控件**对话框关闭。

创建列表框

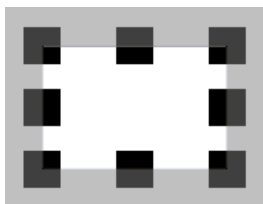
您可以创建应用程序，使用户可以从选项列表中选择项目。

将列表框添加到应用程序时，可以将控件放置在屏幕上，设置属性以配置该列表，然后编写可能需要的任何脚本。

在运行时，列表框可以从文件加载项目，或直接从键盘输入。如需有关详细信息，请参阅第 109 页的“编写 Windows 控件脚本”。

要创建列表框

- 1 创建并放置列表框控件。执行以下操作：
 - a 在向导选择对话框中，选择 **Windows 控件**。
 - b 双击列表框图标。此时再次出现应用程序窗口，光标变成左弯头符号。
 - c 在应用程序窗口中单击要放置该控件的位置。此时出现列表框控件。



- 2 双击该控件。此时出现**列表框控件**对话框。



- 3 配置控件。执行以下操作：
 - a 在**控件名**框中输入控件名，如 *ListBox_1*。
 - b 在**标记名**框中输入内存消息标记名，如 *LB1_Value*。
 - c 在**属性**区域，配置控件的显示方式与功能。
- 4 单击**确定**。

创建组合框

您可以创建应用程序使用组合框让用户从选项列表中选择项目。组合框是将文本框与列表框综合到一起的一种 Windows 控件。

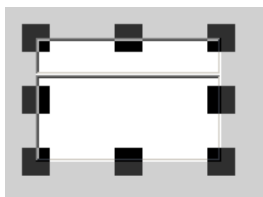
将组合框添加到应用程序时，可以将控件放置在屏幕上，设置属性以配置组合框，然后编写可能需要的任何脚本。

在运行时，组合框可以从文件加载项目，也可以直接从键盘输入。如需有关详细信息，请参阅第 109 页的“编写 Windows 控件脚本”。

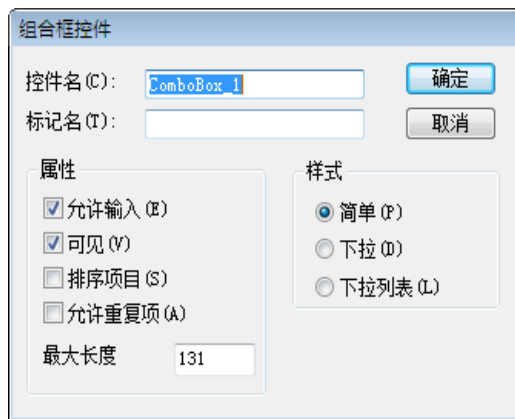
要创建组合框

1 创建并放置组合框控件。执行以下操作：

- a 在**向导选择**对话框中，选择 **Windows 控件**。
- b 双击组合框图标。此时再次出现应用程序窗口，光标变成左弯头符号。
- c 在应用程序窗口中单击要放置该控件的位置。此时出现组合框控件。



2 双击控件。此时出现**组合框控件**对话框。



3 配置控件。执行以下操作：

- a 在**控件名**框中输入控件名，如 *ComboBox_1*。
- b 在**标记名**框中输入内存消息标记名，如 *CB1_Value*。
- c 在**属性**区域，配置控件的显示方式与功能。
- d 在**样式**区域中，选择组合框的类型。

4 单击**确定**。

创建复选框

复选框控件允许操作员选择选项。

要配置复选框控件

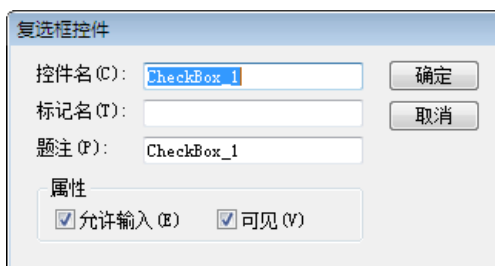
1 创建复选框 执行以下操作：

- a 在**向导选择**对话框中，选择 **Windows 控件向导**。此时出现各个控件向导图标。
- b 双击复选框图标。此时再次出现应用程序窗口，光标变为“左”弯头符号。
- c 在应用程序窗口中单击要放置该向导的位置。此时出现复选框向导。



- d 拖动并调整向导大小。

2 双击该向导。此时出现**复选框控件**对话框。



3 配置向导。执行以下操作：

- a 输入控件名。
- b 输入离散标记名，或双击空标记名框以显示**选择标记**对话框，然后选择标记。
- c 输入显示在按钮表面的标题。

4 单击**确定**。

创建单选按钮组

用户必须从多个选项中选择其一的时候，使用单选按钮。用户选择一个选项时，会取消之前所选的选项。

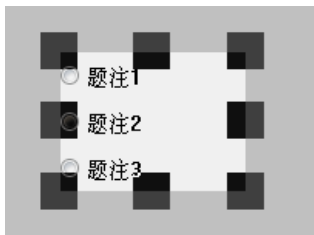
您可以为用户创建多个选项作为单选按钮组。每个单选按钮具有一个题注，并向脚本提供唯一值。

您可以仅将整型标记指定给单选按钮控件。

要创建单选按钮组

1 创建“单选按钮”控件向导。执行以下操作：

- a 在**向导选择**对话框中，选择 **Windows 控件** 向导。此时出现各个控件向导图标。
- b 双击“单选按钮组”图标。此时再次出现应用程序窗口，光标变成左弯头符号。
- c 在应用程序窗口中单击要放置该向导的位置。此时出现“单选按钮组”控件向导，显示三个单选按钮。



- d 根据应用程序拖动该向导并调整其大小。

2 双击该向导。此时出现**单选按钮组控件**对话框。



3 配置向导。执行以下操作：

- a 输入控件名。
- b 输入要连接到此控件的整型标记名。
- c 输入要显示的按钮数。
- d 输入每个按钮的题注。
- e 设置布局与属性。

4 单击**确定**。

编写 Windows 控件脚本

您可以在脚本中使用 QuickScript 函数以：

- 获取或设置控件的值。
- 启用、禁用或隐藏控件。
- 使用组合框、列表框、文本框以及复选框中的项目。

运行时属性可以是可读写的或只读的，具体取决于属性本身。

使用 `GetPropertyX()` 与 `SetPropertyX()` 函数来控制或检索这些属性。

获取或设置控件值

`.Value` 属性是所有 InTouch “Windows 控件向导” 的缺省属性。

对此属性所作的更改与 InTouch 标记名以及 “Windows 控件向导” 同步。

.Value 点域

所有 InTouch Windows 控件向导的缺省属性。对此属性所作的更改与 InTouch 标记名以及 Windows 控件向导同步。

类别

Windows 控件

用法

M、I、D 分别是 `GetProperty` 与 `SetProperty` 函数的内存 (Memory)、整型 (Integer)、离散 (Discrete) 形式。

```
[ErrorNumber=]GetPropertyM("ControlName[.Value]", Tagname);
```

```
[ErrorNumber=]SetPropertyM("ControlName[.Value]", Value);
```

```
[ErrorNumber=]GetPropertyI("ControlName[.Value]", Tagname);
```

```
[ErrorNumber=]SetPropertyI("ControlName[.Value]", Value);
```

```
[ErrorNumber=]GetPropertyD("ControlName[.Value]", Tagname);
```

```
[ErrorNumber=]SetPropertyD("ControlName[.Value]", Value);
```

参数

ControlName

Windows 控件的名称，例如 `ChkBox_4`。

Tagname

要写入属性值的标记名。

[.Value]

此属性是可选的。如果未指定，函数总是假设使用 `.Value` 属性。

Value

处理函数时，要写入的实际值，或用于存放要写入的属性值的有效 InTouch 标记名（与要写入的属性的类型相同）。

附注

指定给列表框或组合框的标记名初始值无法用于初始化列表框或组合框的值。

此点域在开发与运行时为可读写。如果通过将标记名关联到列表框或组合框来访问 .Value 点域，则此点域是只读的。如果将 .Value 点域指定给复选框、单选按钮、文本框，则它是可读写的。开发时指定的值用作运行时的缺省值。

数据类型

对于文本框、列表框、组合框，为消息（可读写）。

对于单选按钮，为整型（可读写）。

对于复选框，为离散（可读写）。

适用范围

文本框、列表框、组合框、复选框、单选按钮。

示例

以下语句将单选按钮对象 "RadioButton_1" 的 .Value 点域的值设置为 4:

```
SetPropertyI("RadioButton_1.Value", 4);
```

另请参阅

GetPropertyM()、 SetPropertyM()、 GetPropertyI()、
SetPropertyI()、 GetPropertyD()、 SetPropertyD()

启用或禁用用户输入控件

使用 `.Enabled` 属性确定控件对象是否可以响应操作员生成的事件。

.Enabled 点域

确定控件对象是否可以响应用户生成的事件。

类别

Windows 控件

用法

```
[ErrorNumber=] GetPropertyD("ControlName.Enabled",  
    Tagname);
```

```
[ErrorNumber=] SetPropertyD("ControlName.Enabled",  
    Discrete);
```

参数

ControlName

Windows 控件的名称。例如， *ChkBox_4*。

Tagname

离散型标记名，用于存放所请求的属性。

Discrete

离散值或离散型标记名，处理函数时用于存放在要写入的值。
对于离散值：

0 = 禁用控件。

1 = 启用控件。

附注

此属性在开发与运行时都是可读写的。

数据类型

离散（可读写）

适用范围

文本框、列表框、组合框、复选框、单选按钮。

示例

以下语句禁用列表框对象 "ListBox_1"。

```
SetPropertyD("ListBox_1.Enabled", 0);
```

另请参阅

`GetPropertyD()`、`SetPropertyD()`

动态隐藏 Windows 控件

使用 `.Visible` 属性来确定窗口中的 Windows 控件是否可见。

.Visible 点域

确定窗口中 Windows 控件是否可见。

类别

Windows 控件

用法

```
[ErrorNumber=]GetPropertyD("ControlName.Visible", Tagname);
```

```
[ErrorNumber=]SetPropertyD("ControlName.Visible", Number);
```

参数

ControlName

Windows 控件的名称。例如， *ListBox_1*。

Tagname

标记名（与返回值的类型相同），处理函数时用于存放属性值。

Number

离散值或离散型标记名，处理函数时用于存放要写入的值。对于离散值：

0 = 控件不可见。

1 = 控件可见。

附注

此属性在开发与运行时都是可读写的。

数据类型

离散（可读写）

有效值

适用范围

文本框、列表框、组合框、复选框、单选按钮。

示例

以下语句隐藏文本框 "TextBox_1"。

```
SetPropertyD("TextBox_1.Visible",0);
```

另请参阅

GetPropertyD()、 SetPropertyD()

对组合框中的项目进行添加与删除

使用以下脚本函数对组合框与列表中的项目进行添加与删除

脚本函数	效果
wcAddItem()	将某个项目添加到列表框或组合框的列表末尾。如果启用了排序功能，则在添加项目之后给列表排序。
wcInsertItem()	将项目添加到列表框或组合框的列表的指定位置。
wcDeleteItem()	删除列表框或组合框的列表中指定的位置上的项目。
wcDeleteSelection()	从列表或组合框中删除当前所选的项目。
wcClear()	从列表或组合框中删除所有项目。

wcAddItem() 函数

将某个项目添加到列表框或组合框的列表末尾。如果启用了排序功能，则在添加项目之后给列表排序。

类别
Windows 控件

语法
[ErrorNumber=]wcAddItem("ControlName", "MessageTag");

参数
ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

MessageTag
要显示的消息字符串。实际的字符串或消息标记名。

附注
如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围
列表框与组合框。

示例
在打开包含列表框的窗口（使用显示时窗口 QuickScript）时，以下语句将消息字符串的内容添加到列表框中。

```
wcAddItem("ListBox_1", "Chocolate");  
wcAddItem("ListBox_1", "Vanilla");  
wcAddItem("ListBox_1", "Strawberry");
```

另请参阅
wcInsertItem()

wcInsertItem() 函数

将指定的字符串插入列表框或组合框列表中指定的位置上。与 `wcAddItem()` 函数不同，`wcInsertItem()` 函数并不给列表排序，即便它是作为排序的列表框或组合框来创建的。

类别

Windows 控件

语法

```
[ErrorNumber=]wcInsertItem("ControlName", ItemPosition,  
    "Message");
```

参数

ControlName

Windows 控件对象的名称。例如，`ListBox_1`。实际的字符串或消息标记名。

ItemPosition

与要添加的项目的位置对应的数字。如果此参数为 `-1`，则将字符串添加到列表的末尾。任何数字或整型标记名。

Message

包含要插入到 `ItemPosition` 所指出的位置的字符串。实际的字符串或消息标记名。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句将 "Blueberry" 这个新项目插入到列表框中的第四个位置上（从上至下）。

```
wcInsertItem("ListBox_1", 4, "Blueberry");
```

另请参阅

`wcAddItem()`

wcDeleteItem() 函数

从列表框或组合框列表中指定的位置上删除某个项目。

类别

Windows 控件

语法

```
[ErrorNumber=]wcDeleteItem("ControlName", ItemPosition);
```

参数

ControlName

Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

ItemPosition

与项目的位置对应的数字。任何数字或整型标记名。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句删除列表中的第三个项目：

```
wcDeleteItem("ListBox_1", 3);
```

wcDeleteSelection() 函数

从列表中删除当前所选的项目。

类别

Windows 控件

语法

```
[ErrorNumber =]wcDeleteSelection("ControlName");
```

参数

ControlName

Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句删除列表框中当前所选的项目：

```
wcDeleteSelection("ListBox_1");
```

wcClear() 函数

从列表框或组合框中删除所有的项目。

类别
Windows 控件

语法
[ErrorNumber=]wcClear(" *ControlName*");

参数
ControlName
Windows 控件对象的名称。例如， ListBox_1。实际的字符串或消息标记名。

附注
如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围
列表框与组合框。

示例
在动作脚本运行时，以下语句清除列表框中的所有的项目：
wcClear("ListBox_1");

从文件中加载列表项或将列表项保存到文件

使用以下脚本函数从文件中将一些项目加载到组合框或列表中，或是将组合框或列表中的项目保存到文件。

脚本函数	效果
wcLoadList()	将文件中的一些新项目加载到列表框或组合框的内容中。
wcSaveList()	将列表框或组合框的内容保存到文件。

wcLoadList() 函数

将文件中的一些新项目加载到列表框或组合框。

类别
Windows 控件

语法
[ErrorNumber=]wcLoadList(" *ControlName*", " *Filename*");

参数
ControlName
Windows 控件对象的名称。例如， ListBox_1。实际的字符串或消息标记名。

Filename

包含文件的名称。如果没有将完整的路径名作为消息参数的一部分来提供，则函数会在应用程序目录中查找消息文件。实际的字符串或消息标记名。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

如果使用外部文件来填充列表与组合框，则它们必须采用特定的格式并包含特定的信息。格式：

```
ControlType, ListCount
ListItem, ItemIndex
ListItem, ItemIndex
:
:
ListItem, ItemIndex
```

ControlType 为 COMBOBOX 或 LISTBOX。

例如，您希望将某个列表文件加载到组合框，它包含三个可供选择的项目，并且这些项目没有指定的项目数据。文件的格式是：

```
COMBOBOX, 3
Chocolate, 0
Vanilla, 0
Strawberry, 0
```

COMBOBOX 是控件类型。列表计数为 3: Chocolate、Vanilla、Strawberry。然后 Chocolate 被列为第一个项目或位置 1。Vanilla 是位置 2、Strawberry 是位置 3。每个项目的数据值都是 0。

如需有关项目数据的详细信息，请参阅第 124 页的“wcSetItemData() 函数”。

适用范围

列表框与组合框。

示例

以下语句将适当格式的列表（位于 c:\wclist.txt 中）加载到组合框中：

```
wcLoadList("Combobox_1", "c:\wclist.txt");
```

另请参阅

wcAddItem()、wcSaveList()

wcSaveList() 函数

将列表框或组合框中的项目保存到文件。

类别

Windows 控件

语法

```
[ErrorNumber=]wcSaveList("ControlName", "Filename");
```

参数

ControlName

Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

Filename

包含文件的名称。如果该文件不存在，则创建它。实际的字符串或消息标记名。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句将列表框中的当前项目保存到文件（c:\newlist.txt）中：

```
wcSaveList("ListBox_1", "c:\newlist.txt");
```

另请参阅

wcLoadList()、wcSetItemData()

在组合框或列表中查找项目

使用 `wcFindItem()` 函数在列表框或组合框中搜索指定的项目。如果找到项目，此函数将相应的位置作为第四个参数返回给某个整型标记名。

`wcFindItem()` 函数

确定列表框或组合框中第一个同所提供的消息字符串匹配的项目所对应的位置。

类别

Windows 控件

语法

```
[ErrorNumber=]wcFindItem ("ControlName", "MessageTag",  
CaseSens, Tagname);
```

参数

ControlName

Windows 控件对象的名称。例如，`ListBox_1`。实际的字符串或消息标记名。

MessageTag

要比较的消息字符串。实际的字符串或消息标记名。

CaseSens

确定字符串比较的类型。它可以是离散值或标记名。以下值有效：

0 = 不区分大小写。1 = 区分大小写。

Tagname

返回匹配项位置的整型标记。如果未找到匹配项，则返回 -1。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围

列表框、组合框

示例

假设 `ListBox_1` 这个列表框包含 "ItemA"、"ItemB"、"ItemC"，则此函数将以下这些值返回给 `Result`：

```
wcFindItem("ListBox_1", "ItemB", 0, Result);
```

返回 2

```
wcFindItem("ListBox_1", "Itemb", 1, Result);
```

返回 -1

```
wcFindItem("ListBox_1", "itemc", 0, Result);
```

返回 3

```
wcFindItem("ListBox_1", "XYZ", 0, Result);
```

返回 -1

在组合框或列表中使用项目索引

使用以下点域处理列表框或组合框中的项目索引。

点域	效果
.TopIndex	列表框中最顶端项目的整型索引。
.NewIndex	通过 <code>wcAddItem()</code> 或 <code>wcInsertItem()</code> 函数添加到列表框或组合框的最后一个项目的整型索引（标记名）。
.ListIndex	列表中当前所选项目的索引（标记名或数字）。

.TopIndex 点域

设置或读取列表框中最顶端项目的整型索引。

类别

Windows 控件

用法

```
[ErrorNumber=]GetPropertyI("ControlName.TopIndex", Tagname);  
[ErrorNumber=]SetPropertyI("ControlName.TopIndex", Number);
```

参数

ControlName

Windows 控件的名称。例如， *ListBox_1*。

Tagname

整型标记名，处理函数时用于存放属性值。

Number

定义列表框中最顶端项目的索引号。可以是整数值或整型标记名，也可以是提供整数值的表达式。

附注

此属性仅在运行时可用。

数据类型

整型（可读写）

适用范围

列表框。

示例

以下语句将列表框对象 "ListBox_1" 的 TopIndex 值设置为 14:

```
SetPropertyI("ListBox_1.TopIndex", 14);
```

另请参阅

GetPropertyI()、 SetPropertyI()、 .ListIndex、 .NewIndex

.NewIndex 点域

返回通过 `wcAddItem()` 或 `wcInsertItem()` 添加到列表框或组合框的最后一个项目的整型索引（标记名）。

类别

Windows 控件

用法

```
[ErrorNumber=]GetPropertyI("ControlName.NewIndex", Tagname);
```

参数

ControlName

Windows 控件的名称。例如， *ListBox_4*。

Tagname

标记名，包含添加到列表或组合框的最后一个项目的整型索引。对于空白列表，返回值为 -1。

附注

此属性仅在运行时可用。

数据类型

整型（只读）

适用范围

列表框与组合框。

示例

以下语句检索 "ListBox_1" 列表框中最近添加的项目的索引，并将该值写入内存整型标记名 `NewItemIndex`。

```
GetPropertyI("ListBox_1.NewIndex", NewItemIndex);
```

另请参阅

`GetPropertyI()`、`wcAddItem()`、`wcInsertItem()`、`.ListIndex`、`.TopIndex`

.ListIndex 点域

设置或读取列表中当前所选项目的索引（标记名或数字）。

使用列表框时，索引 -1 指出当前未选择任何项目。

使用组合框时，索引 -1 指出用户已经将新文本输入到控件的文本输入区域中。

语法

```
[ErrorNumber=]GetPropertyI("ControlName.ListIndex", Tagname);
```

```
[ErrorNumber=]SetPropertyI("ControlName.ListIndex", Number);
```

参数

ControlName

Windows 控件的名称。例如， `ListBox_4`。

Tagname

要写入当前所选项目的索引的标记名。

Number

定义列表中某个特定项目的索引号。

附注

索引号定义列表中的某个特定项目。使用 `.ListIndex` 点域来设置或确定列表或组合框中当前所选项目的索引 °f

此属性仅在运行时可用。

数据类型

整型（读或写）

适用范围

列表框与组合框。

示例

此语句检索 "ListBox_1" 列表框中当前所选的项目，并将该值写入内存整型标记名 `MyListBoxIndex` 中。

```
GetPropertyI( "ListBox_1.ListIndex",  
    MyListBoxIndex );
```

另请参阅

`GetPropertyI()`、`SetPropertyI()`、`.NewIndex`、`.TopIndex`

统计列表框或组合框项目数

`.ListCount` 点域包含列表框或组合框中项目的数量。

.ListCount 点域

读取列表框或组合框中项目的数量。

类别

Windows 控件

语法

```
[ErrorNumber=]GetPropertyI("ControlName.ListCount", Tagname);
```

参数

ControlName

Windows 控件的名称。

Tagname

有效的标记名，包含列表中项目的整型计数。

附注

此属性仅在运行时可用。

数据类型

整型（只读）

适用范围

列表框与组合框。

示例

以下语句检索 *ListBox_1* 列表框中的项目数量，并将该值写入内存整型标记 *MyListBoxCount* 中。

```
GetPropertyI("ListBox_1.ListCount", MyListBoxCount);
```

另请参阅

GetPropertyI()、.ListIndex

获取或设置列表项的值

使用 `wcGetItemData()` 函数来查找与项目索引所确定的列表项关联的整数值。

使用 `wcSetItemData()` 函数将整数值指定给项目索引指定的列表中的项目。这会将一个数字指定给字符串。

wcGetItemData() 函数

读取与 `ItemIndex` 参数所确定的列表项关联的整数值。

类别

Windows 控件

语法

```
[ErrorNumber=]wcGetItemData("ControlName", ItemIndex, Tagname);
```

参数

ControlName

Windows 控件对象的名称。例如，`Listbox_1`。实际的字符串或消息标记名。

ItemIndex

与项目的位置对应的数字。任何数字或整型标记名。

Tagname

实型或整型标记名的实际名称。从函数返回时，`wcGetItemData()` 函数将该项目对应的数值赋予此标记名。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句检索与列表框中的第五个项目关联的数值，并将它返回给 `ItemValue` 整型标记名：

```
wcGetItemData("ListBox_1", 5, ItemValue);
```

如果给列表中的第五个项目指定了整数值 4500，则 `ItemValue` 标记名包含 4500。

另请参阅

wcSetItemData()

wcSetItemData() 函数

将项目的整数值（Number 参数）指定给列表中 ItemIndex 参数所指定的项目。此函数允许将数字指定给字符串。

类别

Windows 控件

语法

```
[ErrorNumber=]wcSetItemData("ControlName", ItemIndex, Number);
```

参数

ControlName

Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

ItemIndex

指定要编辑的列表项的整数值。任何数字或整型标记名。

Number

代表项目数据的整数值。任何数字或整型标记名。

附注

您可以使用“记事本”之类的程序创建包含各个项目的完整列表，然后使用一个函数调用来加载它们。根据 wcSaveList() 函数的要求设置列表格式。

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

使用 wcGetItemData() 函数返回与列表项关联的值（项目数据）。此标记名参数包含返回的数值。此参数可以是直接写入实际设备的 I/O 整型标记名。

示例

某配方包含三种成分：面粉、糖、盐。面粉的质量是 4500 克、糖是 1500 克、盐是 325 克。通过使用由所选配方 (tagname, RecipeName) 触发的数据改变脚本，将值指定给每个列表框项目。

```
wcSetItemData("ListBox_1", 1, 4500); {set 1st item in the list
(flour)=4500}
wcSetItemData("ListBox_1", 2, 1500); {set 2nd item in the list
(sugar)=1500}
wcSetItemData("ListBox_1", 3, 325); {set 3rd item in the list
(salt)=325}
```

另请参阅

wcLoadList()、wcSaveList()、wcGetItemData()

获取列表项的名称

使用 `wc.GetItem()` 函数返回与列表框或组合框中相应的项目索引关联的项目字符串。

`wc.GetItem()` 函数

返回一个字符串，包含与列表框或组合框中的 `ItemIndex` 所对应的项目的内容。

类别

Windows 控件

语法

```
[ErrorNumber=]wc.GetItem("ControlName", ItemIndex, Tagname);
```

参数

ControlName

Windows 控件对象的名称。例如，`ListBox_1`。实际的字符串或消息标记名。

ItemIndex

与项目的位置对应的数字。任何数字或整型标记名。

Tagname

消息标记名。从函数返回时，`wc.GetItem` 函数将该项目所对应的数据放入此标记名。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句将组合框中的第十个项目的字符串值返回给 `ListSelection` 消息标记：

```
wc.GetItem("Combobox_1", 10, ListSelection);
```

如果列表中的第十个项目是 "Vanilla"，则 `ListSelection` 标记包含字符串 "Vanilla"。

加载文本框的内容

使用 `wcLoadText()` 函数从文件加载文本框的内容。使用 `wcSaveText()` 函数将文本框的内容保存到文本文件中。

备注 如果给标记名定义了最大长度，则只能从文本框内容中将该数量的字符给标记。如果未给文本框指定标记，其内容最多可以包含 65535 个字符。

wcLoadText() 函数

使用文件的内容替换文本框的内容。

类别

Windows 控件

语法

```
[ErrorNumber=]wcLoadText("ControlName", "Filename");
```

参数

ControlName

Windows 控件对象的名称。例如， `Listbox_1`。实际的字符串或消息标记名。

Filename

包含文件的名称。如果未将完整的路径名作为消息参数的一部分来提供，此函数会在应用程序目录中查找文件。实际的字符串或消息标记名。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围

文本框。

示例

在打开包含文本框的窗口（“显示时”窗口脚本）时，以下语句将一个文本文件 (`c:\InTouch.32\readme.txt`) 加载到文本框中：

```
wcLoadText("Textbox_1", "c:\InTouch.32\readme.txt");
```

wcSaveText() 函数

将文本框中包含的文本保存到指定的文件中。如果文件不存在，则创建它。如果存在，则必须是可读写的。

类别

Windows 控件

语法

```
[ErrorNumber=]wcSaveText("ControlName", "Filename");
```

参数

ControlName

Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

Filename

包含目标文件的名称。如果不提供完整的路径名，则文件保存到应用程序目录中。如果该文件存在，则它会被覆盖。如果该文件不存在，则创建它。随后可以使用 wcLoadText() 函数将产生的文件加载到文本框对象中。实际的字符串或消息标记名。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围

文本框。

示例

在动作脚本运行时，以下语句将文本框中所输入的当前信息保存到文件 (c:\InTouch.32\newtext.txt) 中：

```
wcSaveText("Textbox_1", "c:\InTouch.32\newtext.txt");
```

另请参阅

wcLoadText()

检查文本框是否为只读

使用 `.ReadOnly` 点域来确定文本框的内容是只读还是可读写的。

`.ReadOnly` 点域

确定文本框的内容是只读还是可读写的。

类别

Windows 控件

用法

```
[ErrorNumber=]GetPropertyD("ControlName.ReadOnly", Tagname);
```

参数

ControlName

Windows 控件的名称。例如, *Textbox_1*。

Tagname

离散标记名, 处理函数时用于存放属性值。

0 = 文本框的内容为可读写

1 = 文本框的内容为只读

附注

此属性在开发与运行时都可用。

数据类型

离散 (只读)

适用范围

文本框。

示例

以下语句检索 "TextBox_1" 文本框的只读状态:

```
GetPropertyD("TextBox_1.ReadOnly", A_Tagname);
```

另请参阅

`GetPropertyD()`, `SetPropertyD()`

获取或设置复选框的标签

.Caption 点域定义复选框的消息文本。

.Caption 点域

确定要使用复选框显示的消息。

类别

Windows 控件

用法

```
[ErrorNumber=]GetPropertyM ("ControlName.Caption", Tagname);  
[ErrorNumber=]SetPropertyM ("ControlName.Caption", "Message");
```

参数

ControlName

Windows 控件的名称。例如, *ChkBox_4*。

Tagname

消息标记名, 用于存放所请求的属性。

Message

用英文引号括起的消息字符串。

附注

此属性在开发与运行时都是可读写的。

数据类型

消息 (可读写)

适用范围

复选框。

示例

此语句将复选框对象 "CheckBox_1" 的题注设置为 "Blue Paint Option"。

```
SetPropertyM("CheckBox_1.Caption","Blue Paint Option");
```

另请参阅

GetPropertyM(), SetPropertyM()

理解 Windows 控件错误消息

只要给出错误号，`wcErrorMessage()` 便会返回描述该错误的字符串消息。它适用于列表框、文本框、组合框、单选钮及复选框。

Window 控件功能会根据 `QuickScript` 函数的处理结果返回一些值。返回值用于错误诊断。您可以将这些值指定给整型标记名。例如：

```
ErrorNumber = wcGetItem("ControlName", Number, Tagname);
```

在此脚本中，`ErrorNumber` 是包含所返回的错误值的整型标记。函数的返回值可以传递给 `wcErrorMessage()`。

`wcErrorMessage()` 将返回该错误的字符串描述。例如：

```
ErrorMsg = wcErrorMessage(ErrorNumber);
```

在此脚本中，`ErrorMsg` 是一个消息型标记，它包含所返回的错误文本。下表列出错误值及其定义。

错误消息	定义
0	成功
-1	一般错误
-2	可用内存不足
-3	属性为只读
-4	指定的项目已经存在
-5	未知对象名
-6	未知属性名
-x	未知的错误

wcErrorMessage() 函数

返回描述错误的消息字符串。

类别

Windows 控件

语法

```
ErrorMessage=wcErrorMessage(ErrorNumber);
```

参数

ErrorMessage

消息标记名。

ErrorNumber

所有 Windows 控件函数返回的数字。任何数字或整型标记名。

附注

如需返回的错误号列表，请参阅第 130 页的“理解 Windows 控件错误消息”。

适用范围

列表框、文本框、组合框、复选框、单选钮。

示例

如果在加载列表时发生错误，则在 **ErrorDescription** 消息标记名中显示该错误的文本描述。在本例中，通过给 **ErrorDescription** 标记名指定一个“值显示”动画链接来显示错误消息。

在“显示时”窗口的 QuickScript 中：

```
ErrorNumber=wcLoadList("ListBox_1","c:\recipe.txt");  
ErrorDescription=wcErrorMessage(errornumber);
```

您可以将此函数与所有 Windows 控件函数结合使用，以显示错误消息：

```
ErrorNumber=wcAddItem("ListBox_1","AM_4A4356");  
ErrorMsg=wcErrorMessage(ErrorNumber);
```


第 6 章

ActiveX 控件

ActiveX 控件是一些独立的软件组件，它们可以给 InTouch 应用程序带来额外的功能。

您可以在 InTouch 应用程序中使用多种类型的 ActiveX 控件。

- InTouch HMI 包含用于报警的 ActiveX 控件。
- 其它 Wonderware 产品（如 ActiveFactory）则提供用于操纵和分析数据的控件。
- 您可以使用第三方的 ActiveX 控件。
- 您可以使用 Visual Basic 或 C 语言来开发自己的 ActiveX 控件。

您可以在 InTouch 应用程序中使用任意数量的 ActiveX 控件。您可以：

- 选择 ActiveX 控件并将它粘贴到任何应用程序窗口。
- 调整控件大小（如果控件支持调整大小）。
- 剪切、复制、粘贴、删除 ActiveX 控件，以及为其创建副本。
- 对齐 ActiveX 控件：左侧、右侧、顶部、底部，以及中心点。
- 将 ActiveX 控件添加到“向导 /ActiveX 工具栏”。
- 在创建单元时合并 ActiveX 控件及其它对象。
- 使用特定 ActiveX 控件所支持的属性、方法及事件。

InTouch 应用程序不支持以下类型的 ActiveX 控件：

- 无窗口控件
- 单框架位置或区域框
- 容器
- 数据控件
- 调度对象

InTouch 应用程序仅支持这些数据类型：离散（布尔）、整型（32 位数字）、实型（32 位 IEEE 浮点数表示法），以及消息（最长 131 个字符的字符串）。不支持的数据类型包括变量、指针、数组、结构，以及参数化的属性。

ActiveX 控件不能同其它的 InTouch 对象（如窗口控件或图形对象）重叠。一个窗口上的 ActiveX 控件太多会导致系统性能降低。

使用 ActiveX 控件

您可以选择一个 ActiveX 控件，将它粘贴到窗口中，然后再双击它。此时出现它的配置对话框。

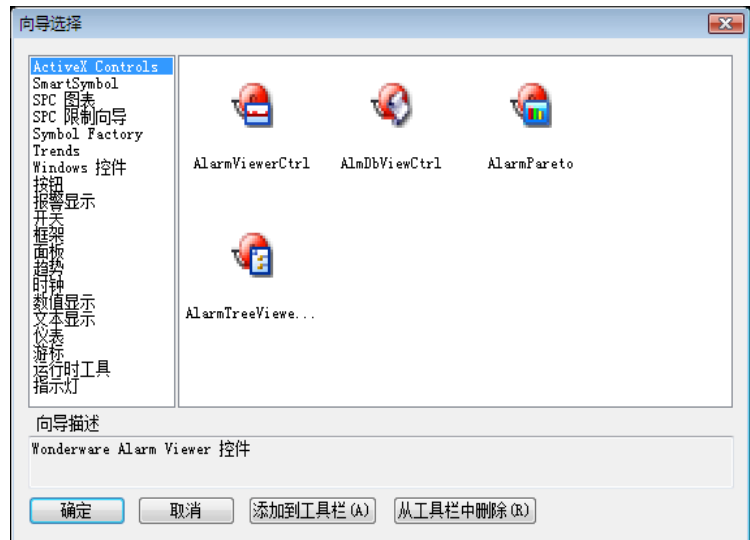
配置 ActiveX 控件时，可以给它指定一个唯一的控件名。随后在脚本中可以引用此控件名。

要在 InTouch 应用程序中使用 ActiveX 控件

- 1 安装 ActiveX 控件。请参阅第 138 页的“安装与删除 ActiveX 控件”。
- 2 选择 ActiveX 控件并将它粘贴到 InTouch 窗口中。
- 3 配置 ActiveX 控件属性，并将它们绑定到标记。
- 4 将 ActiveX 事件关联到 ActiveX 事件脚本。
- 5 在 ActiveX 事件脚本或其它 InTouch QuickScript 中，调用 ActiveX 方法并设置 ActiveX 控件属性。

要在窗口中放置 ActiveX 控件

- 1 单击向导 /ActiveX 工具栏中的“向导”工具。此时出现向导选择对话框。



- 2 在向导列表中，单击 **ActiveX Controls**（ActiveX 控件）类别。此时所有可用的 ActiveX 控件都出现在显示区域。
- 3 双击要使用的 ActiveX 控件。此时对话框关闭，光标变为弯头符号。
- 4 单击要粘贴 ActiveX 控件的位置。

要将 ActiveX 控件添加到工具栏

- 1 单击向导 /ActiveX 工具栏中的向导工具。此时出现向导选择对话框。
- 2 选择要添加的 ActiveX 控件。
- 3 单击添加到工具栏。

要从工具栏中删除 ActiveX 控件

- 1 单击向导 /ActiveX 工具栏中的向导工具。此时出现向导选择对话框。
- 2 单击从工具栏中删除。此时出现从工具栏删除向导对话框。
- 3 选择要删除的 ActiveX 控件。
- 4 单击确定。

配置 ActiveX 控件

您可以为特定的 ActiveX 控件配置的属性取决于该 ActiveX 控件自身。所有属性都有一个缺省值。

在配置 ActiveX 控件的属性之前，必须将它粘贴到 InTouch 窗口中。

- 在将 ActiveX 控件粘贴到窗口时，将生成缺省的控件名，如 Calendar1。您可以在脚本中引用此控件名。ActiveX 控件必须在打开的应用程序窗口中运行，运行的脚本才能引用它。
- 您可以将 ActiveX 控件属性指定给 InTouch 标记。您必须将每种属性类型指定给一个相应的 InTouch 标记类型。

ActiveX 控件有三个标准选项卡：**控件名**、**属性**以及**事件**。



使用**事件**选项卡将脚本指定给可用的控件事件，如用户双击鼠标时。

任何其它选项卡及其配置都是控件独有的，并取决于控件的属性。例如，有些控件可能要求您配置颜色与字体，而其它的则可能不具备这些属性。

给 ActiveX 控件命名

在以下情况中，会创建一个新的控件实例，并给它指定一个独特的名称：

- 选择**编辑**菜单上的**重复**。
- 选择**编辑**菜单上的**剪切**或**复制**，然后选择**粘贴**。
- 选择**文件**菜单上的**窗口另存为**。
- 单击**撤消**，然后单击**恢复**。
- 导入包含控件的窗口。

ActiveX 控件的名称必须唯一。如果试图给控件使用一个已经存在的名称，则会显示一条错误消息。

要给 ActiveX 控件命名

- 1 将 ActiveX 控件粘贴到开发窗口中。
- 2 双击控件。此时出现该控件的**属性**对话框。
- 3 单击**控件名**选项卡，然后在**控件名**框中给该 ActiveX 控件输入名称。

ActiveX 控件上的标准操作

像对任何其它 InTouch 对象那样，您也可以在 ActiveX 控件上执行各种标准的操作。如需有关详细信息，请参阅第 49 页的“所有对象的特殊操作”。

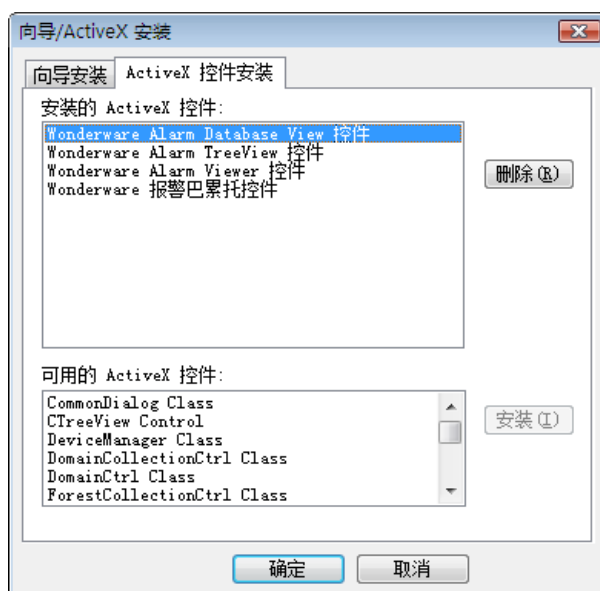
安装与删除 ActiveX 控件

即便已经在计算机上安装了某个 ActiveX 控件，还是必须将此控件安装到 WindowMaker 上以使 InTouch HMI 了解到这个情况。

从 WindowMaker 中删除某个控件时，它并没有从计算机上彻底删除。它只是从内存中删除掉，不再起什么作用。

要安装 ActiveX 控件

- 1 在特别菜单上，指向配置，然后单击向导 /ActiveX 安装。此时出现向导 /ActiveX 安装对话框。
- 2 单击 ActiveX 控件安装选项卡。此时出现 ActiveX 控件安装属性页。



- 3 在安装的 ActiveX 控件列表中，选择要安装在可用的 ActiveX 控件列表内的控件，然后单击安装。

提示 要选择多个控件，请使用 SHIFT 键或 CTRL 键。

- 4 单击确定。

要删除 ActiveX 控件

- 1 在**特别**菜单上，指向**配置**，然后单击**向导 /ActiveX 安装**。此时出现**向导 /ActiveX 安装**对话框。
- 2 单击 **ActiveX 控件安装**选项卡。此时出现 **ActiveX 控件安装**属性页。
- 3 在**安装的 ActiveX 控件**列表中，选择要从应用程序中删除的控件，然后单击**删除**。此时出现一条消息，要求确认是否删除。

提示 要选择多个控件，请使用 **SHIFT** 键或 **CTRL** 键。

- 4 单击**是**以删除该控件。此时该控件移到**可用的 ActiveX 控件**列表中。
- 5 单击**确定**。

索引

符号

\$ObjHor 系统标记 72, 74

\$ObjVer 系统标记 72, 74

A

ActiveX 控件

安装与删除 138

标准操作 137

从工具栏中删除 135

定义 38

放到窗口上 135

关于 133

命名 137

配置 136

使用 134

添加到工具栏 135

ArchestrA 符号的定义 38

按钮

创建 37

缺省字体 14

替换字符串 37

B

报警类型 66

标记

查找与选择 88

启用记录功能 102

替换 94

显示标记计数 13

转换占位符 95

标记过滤器 90

C

.Caption 点域 129

参考位置 61, 62

层叠对象 43

长方形按钮 36

撤消级数 14

触动链接 56, 75

创建按钮 81

创建单选按钮 107

创建动作脚本链接 82

创建多边形 37

创建方向链接 60

创建可见性链接 69

创建离散输入链接 76

创建离散值显示 57

- 创建模拟数据显示 57
- 创建模拟输入链接 77
- 创建图案 36
- 创建线条与图案 36
- 创建应用程序窗口 30
- 创建游标 79
- 创建值显示 56
- 创建字符串输入链接 78
- 创建字符串值显示 58
- 窗口
 - 保存 33
 - 撤消更改 48
 - 创建 30
 - 创建副本 34
 - 打开 33, 82
 - 大小控制 31
 - 导出 31
 - 导入 31
 - 放置触控窗口 72
 - 高度 31
 - 关闭 33, 82
 - 宽度 31
 - 框式样 31
 - 类型 31
 - 名称 30
 - 起始 32
 - 删除 34
 - 设置类型与外观 32
 - 设置为在运行时出现 32
 - 位置大小 31
 - 显示隐藏 83
 - 修改应用程序窗口 32
 - 选择背景颜色 30
 - 在所选对象的位置上打开 73
- 窗口类型
 - 弹出 31
 - 覆盖 31
 - 替换 31
 - 替换、覆盖或弹出 31
- 垂直线按钮 36
- 垂直移动 59
- 错误消息
 - 定义 130
 - 关于 130

D

- DialogStringEntry() 函数 85
- DialogValueEntry() 函数 86
- 打开窗口 82
- 大小高度链接 61
- 大小控制 31
- 大小宽度链接 62
- 单选按钮组控件 108
- 单元
 - 创建 39
 - 定义 38
 - 确定 39
 - 使用 39
- 导入位图 52
- 点域
 - .Caption 点域 129
 - .Enabled 点域 111
 - .NewIndex 点域 121
 - .ReadOnly 点域 128
 - .TopIndex 点域 120
 - .Value 点域 109
 - .Visible 点域 103, 112
- 调色板 21
 - 打开 21
 - 导入与导出 23
 - 使用 21
 - 自定义 22
- 调整对象大小 45
- 动画
 - 数据输入 75
 - 数据显示 56
- 动画链接的类型 56
- 动画任务 88
- 对话框的功能 24
- 对齐对象 43
- 对齐网格 15
- 对象
 - 报警类型 66
 - 层叠 43
 - 创建副本 50
 - 垂直移动 59
 - 大小高度链接 61
 - 大小宽度链接 62

- 调整大小 45
- 调整多边线与多边形的形状 51
- 对齐 43
- 对齐网格 15
- 翻转 44
- 复杂 38
- 简单 36
- 剪切、复制以及粘贴 49
- 控制水平与垂直间距 44
- 趋势 102
- 删除 48
- 闪烁 68
- 失效 70
- 使用方向键移动 27
- 水平移动 58
- 特殊操作 49
- 替换对象 54
- 透过中空对象选取 14
- 选择 41
- 旋转 45, 59
- 移动 42, 58
- 组合成单元 39
- 组合成符号 40
- 对象的垂直间距 44
- 对象的水平间距 44
- 对象的水平与垂直间距 44

E

- .Enabled 点域 111

F

- 翻转对象或单元 44
- 符号
 - 创建或分解 40
 - 定义 38
 - 合并与分解 39
 - 确定 39
 - 使用 39
- 复选框
 - 创建 107
 - 获取或设置标签 129

G

- 给多边形添加或删除顶点 51

- 更改标记名引用 94
- 更改填充 47
- 更改线条与轮廓 47
- 更改圆角长方形的圆角半径 54
- 更新使用计数 13
- 工具栏
 - 更改大小 18
 - 显示或隐藏 18
- 工具提示
 - 创建 71
 - 静态文本或表达式 71
 - 配置 71
- 工作区
 - 自定义 12
- 关闭窗口 82

H

- 函数

- DialogStringEntry() 函数 85
- DialogValueEntry() 函数 86
- ShowAt() 函数 72
- ShowTopLeftAt() 函数 72
- wcAddItem() 函数 113
- wcClear() 函数 116
- wcDeleteItem() 函数 115
- wcDeleteSelection() 函数 115
- wcErrorMessage() 函数 131
- wcFindItem() 函数 119
- wcGetItem() 函数 125
- wcGetItemData() 函数 123
- wcInsertItem() 函数 114
- wcLoadList() 函数 116
- wcLoadText() 函数 126
- wcSaveList() 函数 118
- wcSaveText() 函数 127
- wcSetItemData() 函数 124

- 绘图按钮 36

J

- 简单对象 36
- 键盘快捷键 93
- 剪切、复制以及粘贴对象链接 49
- 将对象置后 43
- 将对象置前 43

K

控件

- 获取与设置值 109
- 启用或禁用用户输入 111

快速切换 14

框式样 31

- 单一 31
- 加倍 31
- 无 31

L

链接

- 按钮 81
- 触动 56, 75
- 创建离散报警状态 65
- 创建离散填充颜色 63
- 创建链接 59
- 创建模拟报警状态颜色 65
- 创建模拟表达式颜色 64
- 创建移动 58
- 打开或关闭窗口 82
- 大小链接 61
- 单元中的符号 39
- 动画 26, 39
- 动画对象 55
- 动画类型 56
- 动作脚本 82
- 方向 59
- 工具提示 71
- 键盘快捷键 93
- 剪切、复制或粘贴 49
- 剪贴板限制 49
- 可见性 69
- 离散输入 76
- 离散值触动链接 81
- 模拟输入 77
- 屏幕键盘 83
- 闪烁 68
- 失效 70
- 填充百分比 66
- 位置链接 58
- 无法制作符号 39
- 显示 56

颜色 63

- 颜色链接类型 63
- 用户输入类型 75
- 游标创建 79
- 值显示类型 56
- 字符串输入 78

联系技术支持 10

列表

- 保存到文件 116
- 查找项目 119
- 从文件加载 116
- 获取或设置列表项的值 123
- 获取列表项的名称 125
- 使用项目索引 120

列表框

- 创建 105

略图 16

N

- .NewIndex 点域 121

P

配置屏幕键盘 83

平移与缩放

- 显示或隐藏工具栏 16
- 限制 17

Q

起始窗口 32

启用或禁用用户输入 111

趋势对象 38, 102

全屏幕模式 24

R

- .ReadOnly 点域 128

S

ShowAt() 函数 72

ShowTopLeftAt() 函数 72

SmartSymbol 定义 38

闪烁对象 68

闪烁属性 68

设置大小动画效果 61

设置动画效果

- 大小 61
- 填充级别 66
- 颜色 63
- 设置闪烁频率 68
- 设置颜色动画效果 63
- 失效链接 70
- 鼠标快捷方式 26
- 数据输入
 - 离散 76
 - 模拟输入 77
 - 字符串输入 78
- 数据输入动画 75
- 数据显示
 - 创建离散值 57
 - 创建模拟值 57
 - 创建字符串值 58
- 动画 56
- 值 56
- 水平线按钮 36
- 水平移动 58

T

- .TopIndex 点域 120
- 替换标记 94
- 替换字符串
 - 创建按钮 37
 - 文本对象 54
- 填充百分比 66
- 填充级别
 - 百分比 66
 - 设置动画效果 66
- 椭圆按钮 36

V

- .Value 点域 109
- .Visible 点域 103, 112

W

- wcAddItem() 函数 113
- wcClear() 函数 116
- wcDeleteItem() 函数 115
- wcDeleteSelection() 函数 115
- wcErrorMessage() 函数 131
- wcFindItem() 函数 119

- wcGetItem() 函数 125
- wcGetItemData() 函数 123
- wcInsertItem() 函数 114
- wcLoadList() 函数 116
- wcLoadText() 函数 126
- wcSaveList() 函数 118
- wcSaveText() 函数 127
- wcSetItemData() 函数 124
- WindowMaker 11
 - 撤消级别 14
 - 更改标题栏 13
 - 关于 27
 - 快速切换 14
 - 首选项 12
 - 转换时关闭 13
- Windows 控件的脚本函数 109
- Windows 控件向导
 - 创建 103
 - 关于 103
 - 配置 103
 - 隐藏 112
- Windows XP 30, 72
- 网格
 - 对齐 15
 - 间距 13, 15
 - 显示 13, 15
- 位图
 - 创建透明位图 53
 - 导入 52
 - 定义透明度 53
 - 粘贴、编辑 52
- 位图容器
 - 定义 38
 - 使用 52
- 文本
 - 创建 37
 - 更改字体 46
 - 缺省字体 14
- 文本框
 - 创建 104
 - 加载或保存内容 126
 - 检查是否只读 128
- 文档惯例 9

X

- X 位置 31
- 系统标记
 - \$ObjHor 72, 74
 - \$ObjVer 72, 74
- 下拉列表 103
- 显示或隐藏标尺 15
- 显示链接 56
- 显示隐藏窗口 83
- 线条按钮 36
- 向导
 - 安装与删除 100
 - 创建 Windows 控件 103
 - 从工具栏中删除 99
 - 导入 101
 - 定义 38
 - 类型 98
 - 配置 100
 - 配置 Windows 控件 103
 - 使用 98
 - 添加到工具栏 99
 - Windows 控件 103
 - 粘贴实例 99
- 行选择精度 14
- 选择标记对话框 92
- 旋转 59
- 旋转对象 45

Y

- Y 位置 31
- 颜色链接
 - 表达式类型 63
 - 创建离散报警状态链接 65
 - 创建离散填充链接 63
 - 创建模拟报警状态链接 65
 - 创建模拟表达式链接 64
 - 定义 63
- 移动
 - 创建 58
 - 垂直 59
 - 水平 58
- 用于 Windows 控件的脚本函数 109
- 圆按钮 36
- 圆角长方形按钮 36
- 运行时 14

Z

- 粘贴向导实例 99
- 字体
 - 更改 46
 - 配置 14
 - 设置缺省值 24
- 组合框
 - 查找项目 119
 - 创建 106
 - 使用项目索引 120
 - 添加与删除项目 113