Wonderware® FactorySuite™ InTouch®

用户指南

修订版

2001 年 6 月

Wonderware Corporation

版权所有。未经 Wonderware 公司的事先书面许可，不得以电子、机械、影印、录制或其它的任何方式复制、传输本说明文件的任何部分或将其储存在检索系统中。使用此处包含的信息不应承担版权或专利责任。虽然在本说明文件编制过程中采取了一切预防措施，错误或疏漏在所难免，出版商和作者不对此承担任何责任，亦不对因使用此处包含的信息而引起的任何损害负责。

本说明文件的信息如有变更，恕不另行通知，亦不代表 Wonderware 公司一方的承诺。本说明文件所述的软件在遵守许可证或许可协议的前提下提供。本软件的使用或复制须遵从这些协议规定的条款。

© 2001 Wonderware 公司。版权所有。

100 Technology Drive
Irvine, CA 92618
U.S.A.
(949) 727-3200
http://www.wonderware.com

商标

本书所有作为商标或服务标志的专门名词均采用大写字母印刷。Wonderware 公司并不保证本书信息的准确性。使用本书中的任何术语不应视为影响任何商标或服务标志的有效性。

Wonderware、FactorySuite 和 InTouch 是 Wonderware 公司的注册商标。

WindowMaker、WindowViewer、SQL Access Manager、Recipe Manager、SPCPro、DBDump、DBLoad、HDMerge、HistData、Wonderware Logger、Alarm Logger、InControl、InBatch、IndustrialSQL、FactoryOffice、FactoryFocus、License Viewer、SuiteLink、SuiteVoyager 和 NetDDE 是 Wonderware 公司的商标。

目录

欢迎使用 InTouch xi

InTouch 特性 x
系统要求 xiv
安装 InTouch xv
关于本手册 xvi
技术支持 xvii
您的 FactorySuite 许可证 xviii
第一次运行 InTouch xviii
使用 InTouch 应用程序管理器 xx
    应用程序管理器工具 xxv

第 1 章 - WindowMaker 程序元素 1-1

WindowMaker GUI 1-2
应用程序浏览器 1-3
    在应用程序浏览器中导航 1-4
    添加应用程序到应用程序浏览器 1-10
WindowMaker 工具栏 1-13
    使用浮动/固定工具栏 1-13
WindowMaker 标尺 1-23
WindowMaker 状态栏 1-24
WindowMaker 调色板 1-25
    使用标准调色板 1-25
    创建自定义调色板 1-26
弹出式菜单 1-29
通用窗口对话框特性 1-32
其它鼠标快捷方式 1-35
快捷方式和加速键 1-36
用箭头键移动对象 1-37
使用 WindowMaker 帮助 1-38

第 2 章 - 使用 WindowMaker 2-1

简单对象 2-2
复杂对象 2-3
自定义您的开发环境 2-6
使用 WindowMaker 窗口 2-10
    创建新窗口 2-11
    创建一个隐藏标题栏和菜单栏的窗口 2-15
    打开窗口 2-16
    保存窗口 2-16
    关闭窗口 2-17
    删除窗口 2-17
    复制窗口 2-18
    导出窗口 2-19
    导入窗口 2-22
使用图形对象 2-24
    选择和调整对象大小 2-26
    撤消对象编辑 2-27
    复制对象 2-29
    剪切对象到 Windows 剪贴板 2-30
    复制对象到 Windows 剪贴板 2-30

从 Windows 剪贴板粘贴对象 2-31
剪切和粘贴对象链接 2-32
删除对象 2-32
增大或减小圆角对象的半径 2-33
调整多边线或多边形的形状 2-34
排列窗口对象 2-36
对齐对象 2-36
对象分层 2-39
控制水平和垂直间距 2-40
旋转对象 2-41
对象镜像 2-42
创建单元和符号 2-43
使对象对齐网格 2-45
使用图像和位图 2-46
从 Windows 剪贴板粘贴位图 2-49
编辑位图 2-50
创建透明位图 2-51
使用文本对象 2-53
格式化文本对象 2-53
显示数值 2-54
编辑文本对象 2-57
替换部分文本对象 2-58
使用线条和轮廓 2-60
使用向导 2-61
InTouch 窗口控件向导 2-66
使用 InTouch 窗口控件向导 2-67
配置窗口控件向导 2-73
窗口控件向导属性 2-74
窗口控件向导函数 2-75
使用 ActiveX 控件 2-76
配置 ActiveX 控件 2-81
配置 ActiveX 控件属性 2-83
使用 ActiveX 控件方法 2-86
使用 ActiveX 控件事件参数 2-89
重复使用 ActiveX 事件脚本 2-93
导入 ActiveX 事件脚本 2-95
自定义运行时环境 2-96
设置 WindowViewer 的通用属性 2-96
设置 WindowViewer 的窗口配置属性 2-100
选择 WindowViewer 的起始窗口 2-103
将 WindowViewer 作为 NT 服务运行 2-104
使用 InTouch 安全性 2-106
使用安全内部标记名 2-107
配置操作员的安全级别 2-108
更改安全登录口令 2-109
登录到应用程序 2-110
创建自定义安全登录窗口 2-111
注销应用程序 2-111
自动注销系统 2-112

第 3 章 - 创建分布式应用程序 3-1
网络结构 3-2
独立式应用程序 3-2
基于客户端的结构 3-3
基于服务器的结构 3-4
网络应用程序开发 (NAD) 3-5

配置网络资源 3-6
配置文件的 UNC 路径 3-6
Wonderware SuiteLink 通讯协议 3-7
网络疑难排解 3-8
InTouch 的公共数据源配置 3-10
InTouch 访问名 3-10
InTouch 应用程序的 NAD 配置 3-17
自定义 NAD 更新功能 3-20
手动通知客户应用程序改变 3-21
应用程序复制过程 3-21
应用程序编辑锁定 3-21
动态分辨率转换 (DRC) 3-22
使用多监视器系统 3-24
分布式应用程序与时区 3-25
分布式报警 3-26
分布式历史记录 3-26

第 4 章 - 标记名字典 4-1
标记名字典特殊功能 4-2
标记名类型 4-3
内存类型标记名 4-3
I/O 类型标记名 4-4
其它类型标记名 4-5
SuperTag 4-8
扩展的标记名支持 4-9
定义新的标记名 4-10
定义标记名详细资料 4-15
定义内存离散标记名详细资料 4-15
定义内存模拟标记名详细资料 4-16
定义内存消息型标记名详细资料 4-18
定义 I/O 离散型标记名详细资料 4-19
定义 I/O 模拟标记名详细资料 4-21
定义 I/O 消息标记名详细资料 4-24
定义 SuperTag 成员标记名详细资料 4-25
定义标记名报警条件 4-26
报警约束标记名 4-26
定义离散标记名报警条件 4-28
定义模拟型标记名的报警条件 4-30
标记名报警注释 4-33
创建 InTouch SuperTag 4-34
InTouch SuperTag 语法 4-35
创建 SuperTag 模板结构 4-36
创建间接 SuperTag 4-45
编辑 SuperTag 模板和成员标记名 4-47
定义 SuperTag 模板实例 4-48
创建 SuperTag 的其它方法 4-51
使用标记名字典创建 SuperTag 4-52
远程标记名引用 4-54
创建标记名服务器应用程序 4-57
使用远程标记名引用 4-59
动态地址引用 (DRA) 4-60
使用动态引用 4-61
使用 IOSetItem 函数改变引用 4-62
使用 .ReferenceComplete 验证引用 4-62
在本地标记名中使用间接标记 4-63
在远程引用中使用间接标记名 4-64

标记浏览器 4-67
标记浏览器选择模式 4-68
标记浏览器视图 4-70
定义标记源 4-73
定义标记浏览器过滤器 4-76
InTouch 交叉引用实用程序 4-78
查看交叉引用搜索结果 4-80
改变交叉引用搜索标准 4-83
保存交叉引用文件 4-88
打印交叉引用文件 4-89
打印标记名字典详细资料 4-90
从字典中删除标记名 4-92
更新使用计数 4-93
删除不再使用的标记名 4-93
显示标记名使用计数 4-94
替换标记名 4-96
转换占位符标记名 4-97
将标记名转换为远程引用 4-98
定标 I/O 标记名 4-104
仪器故障监视 4-104
内部系统 $Tagnames 4-105
标记名点域 4-108
模拟型标记名的地址位域 4-120
标记名字典实用程序 4-121
DBDump 实用程序 4-121
DBLoad 实用程序 4-124
创建数据库输入文件 4-125
数据库输入文件格式 4-125
创建 SuperTag 实例 4-126
数据库输入文件操作模式 4-129
创建数据库记录模板 4-132

第 5 章 - 创建动画链接 5-1

通用动画链接功能 5-2
对象类型对话框 5-2
动画链接选择对话框 5-3
指定等价键 5-3
应用颜色链接 5-5
访问标记浏览器 5-6
访问标记名点域 5-7
动画对象 5-9
创建触动链接 5-10
创建用户输入触动链接 5-11
创建游标触动链接 5-15
创建触动按钮触动链接 5-17
创建显示链接 5-22
创建颜色链接 5-23
创建对象大小链接 5-30
创建位置链接 5-32
创建填充百分比链接 5-34
创建其它链接 5-36
创建值显示链接 5-42

第 6 章 - 在 InTouch 中创建 QuickScript..... 6-1

InTouch QuickScript..... 6-2
使用 InTouch QuickScript 编辑器..... 6-3
QuickScript 编辑器工具栏..... 6-4
QuickScript 编辑器一般步骤..... 6-4
应用程序脚本..... 6-11
窗口脚本..... 6-13
键脚本..... 6-15
触动按钮动作脚本..... 6-17
指定脚本的等价键..... 6-19
条件脚本..... 6-20
数据改变脚本..... 6-22
ActiveX 事件脚本..... 6-24
QuickFunction..... 6-27
异步 QuickFunction 脚本..... 6-33
使用局部变量..... 6-36
创建 FOR-NEXT 循环脚本..... 6-38
脚本编辑样式和语法..... 6-44
表达式和脚本的基本语法..... 6-44
简单脚本..... 6-45
复杂脚本..... 6-51
导入 QuickScript..... 6-54
打印脚本..... 6-58
脚本函数..... 6-59
字符串函数..... 6-59
数学函数..... 6-61
系统函数..... 6-62
其它函数..... 6-64
脚本编辑器错误消息..... 6-70
窗口控件和分布式报警的错误消息..... 6-75

第 7 章 - 报警/事件..... 7-1

简介..... 7-3
支持非 InTouch 报警供应器..... 7-4
报警背景知识..... 7-5
报警与事件..... 7-7
报警摘要与报警历史..... 7-8
标准报警子系统、主/从报警与分布式报警子系统..... 7-9
分布式报警与 InTouch 旧有标准式报警的比较..... 7-10
终端服务报警支持..... 7-11
支持其它报警源..... 7-12
报警类型..... 7-13
事件类型..... 7-14
报警优先级..... 7-15
报警组..... 7-16
分布式报警组列表..... 7-21
报警确认模型..... 7-24
扩展的报警摘要..... 7-25
使用扩展的摘要报警..... 7-25
发布/预订机制..... 7-26
报警供应器..... 7-26
报警接收器..... 7-26
报警数据存储..... 7-27
标记名报警配置..... 7-28
报警可见性控制..... 7-30

配置报警系统 7-33
报警/事件通用属性 7-33
分布式报警显示 7-35
选择和配置报警查询收藏夹 7-39
分布式报警显示准则 7-42
创建分布式报警显示 7-43
配置分布式报警显示 7-44
分布式报警显示通用属性 7-44
分布式报警消息格式 7-47
分布式报警显示颜色属性 7-50
动态控制显示类型 7-52
在报警确认函数后添加注释 7-53
使用分布式显示监视本地报警 7-54
显示报警统计 7-55
确认本地报警 7-56
分布式报警属性和函数 7-57
报警点域 7-63
从旧 InTouch 标准报警系统移植到 分布式报警系统 7-70
从旧的主/从报警系统移植到分布式 报警系统 7-71
Alarm Printer 公用程序 7-72
报警打印日期/时间标签 7-72
使用一个或多个 Alarm Printer 7-73
使用 Alarm Printer 7-73
Alarm Printer 工具栏 7-74
配置 Alarm Printer 7-75
运行 Alarm Printer 的报警查询 7-81
Alarm DB Logger 公用程序 7-82
记入 SQL 数据库 7-83
使用 Alarm DB Logger 公用程序 7-84
Alarm DB Logger 配置 7-85
Alarm DB Logger 查询配置 7-86
Alarm DB Logger 高级设定配置 7-87
报警数据库视图 ActiveX 控件 7-89
安装 7-90
卸载 7-90
属性 7-91
上下文相关菜单选项 7-93
颜色页 7-104
字体页 7-110
数据库页 7-111
“选择”标签 7-114
“查询”标签 7-123
属性页 7-128
通用属性 7-128
方法和事件 7-130
方法 7-130
事件页 7-133
热备份与同步 7-134
有关热备份对的注意事项 7-135
热备份配置 7-136
设置报警记录的关键码域 7-142
映射报警记录 7-144
热备份使用实例 7-147
同步实例 7-152
Alarm DB Purge/Archive 公用程序 7-154
清除/归档通用属性配置 7-154
清除/归档数据库配置 7-157

清除/归档配置 7-158
Alarm DB Restore 公用程序 7-160
恢复数据库配置 7-160
恢复文件名配置 7-162
AlarmSuite History Migration 实用程序 7-164
分布式报警数据库视图 7-166
视图与存储过程简介 7-166
视图与存储过程列的定义 7-167
报警历史数据库视图 7-169
事件历史数据库视图 7-170
报警与事件历史数据库视图 7-171
报警计数器数据库存储过程 7-173
在 Enterprise Manager 中查看存储过程 定义 7-174
事件计数器数据库存储过程 7-175
AlarmSuite 报警记录数据库视图 7-176

第 8 章 - 实时趋势和历史趋势 8-1
实时趋势 8-2
创建实时趋势 8-2
配置实时趋势 8-3
历史趋势 8-6
创建历史趋势 8-7
配置历史趋势 8-8
使用历史趋势向导 8-12
记录标记名 8-15
配置历史记录属性 8-16
在运行时配置历史趋势 8-20
“平均/散点”历史趋势计算 8-25
在运行时更新历史趋势 8-26
配置历史趋势打印 8-27
历史趋势点域 8-29
历史 QuickScript 函数 8-31
分布式历史系统 8-33
使用分布式历史系统 8-34
分发您的应用程序 8-35
配置分布式历史供应器列表 8-36
配置远程历史供应器 8-39
使用标记浏览器访问远程历史供应器 8-40
动态配置远程历史供应器 8-41
配置分布式历史记录 8-42
创建历史趋势指示器 8-43
历史趋势和夏令时 8-46
HistData 实用程序 8-53
HistData 数据库 8-53
在 InTouch 中使用 HistData 8-56
在 Excel 中使用 HistData 8-60
常见 HistData 错误消息 8-61

第 9 章 - I/O 通讯 9-1
支持的通讯协议 9-2
Wonderware SuiteLink 9-3
InTouch I/O 地址命名惯例 9-4
InTouch I/O 地址 9-5
InTouch 访问名 9-6
在 InTouch 中定义 I/O 项目 9-10
监视 I/O 对话的状态 9-13

使用 IOStatus 主题名 9-13
监视 I/O 服务器通讯状态 9-17
监视多个输入设备状态 9-19

第 10 章 - Terminal Services for InTouch.....10-1
简介 10-2
终端服务器概念 10-4
用户环境 10-5
终端服务优点 10-6
终端服务高级客户端 10-8
优点 10-8
安装 10-9
如何使用 10-9
保护基于 Web 的应用程序 10-11
已知问题和限制 10-14
服务器硬件要求 10-16
RDP 客户端的硬件要求 10-18
规划终端服务安装 10-19
安装终端服务 10-20
安装 Terminal Services for InTouch 10-22
在终端环境下测试您的应用程序 10-23
Terminal Services for InTouch QuickScript 函数 10-24
Wonderware Logger 控制台和客户端文件 10-25
建立终端会话 10-26
连接到终端服务器 10-26
结束终端会话 10-26
其它应用程序问题 10-27
配置客户端设置 10-27

第 11 章 - InTouch 应用程序发布器.....11-1
发布 InTouch 应用程序 11-2
用多种分辨率发布应用程序 11-6

附录 A - InTouch Windows NT 服务概览..... A-1
为什么要使用 Windows NT 服务？ A-2
在桌面和系统环境下运行 Windows NT 服务 A-3
创建主用户帐号 A-4
配置系统权限 A-5
配置 InTouch 服务 A-7
手动安装/删除 NT 服务 A-11
配置 NT 服务的启动选项 A-12
InTouch 服务之间的相关性 A-13
注销 Windows NT A-13
InTouch 服务疑难解答 A-14
InTouch 服务无法安装或启动 A-15
InTouch 服务的注册表项 A-16

词汇表 G-1

索引 I-1

欢迎使用 InTouch

欢迎使用 Wonderware InTouch，这一创建Microsoft Windows 2000 和 Windows NT 操作系统下的人机界面 (HMI) 应用程序的最简便快捷的方法。InTouch 是Wonderware FactorySuite 的一个组成部分。InTouch 广泛应用于包括食品加工、半导体、石油、汽车、化工、制药、纸浆和造纸、交通等在内的全球众多纵向市场。

有关所支持的操作系统版本的详细信息，请参阅“系统要求”。

通过运用 InTouch，您可以创建强大的、功能齐全的应用程序，充分利用 Microsoft Windows 的关键功能，包括ActiveX 控件、OLE、图形和网络等。InTouch也可以通过添加自定义 ActiveX 控件、向导、常规对象以及创建 InTouch QuickScript 等来进行充分地扩展。

InTouch 包括三个主要程序，它们是 InTouch 应用程序管理器、WindowMaker 和WindowViewer。此外，InTouch 还包括诊断程序 Wonderware Logger 。

InTouch 应用程序管理器用于组织您所创建的应用程序，也可以用于将 WindowViewer 配置成一个 NT 服务程序，为基于客户机和基于服务器的结构体系配置网络应用程序开发 (NAD)，以及配置动态分辨率转换 (DRC) 和（或）分布式报警。DBDump 和 DBLoad 数据库实用程序也是从应用程序管理器启动的。

WindowMaker是一个开发环境，在这个开发环境中可以使用面向对象的图形来创建富于动画感的触控式显示窗口。这些显示窗口可以连接到工业 I/O 系统和其它 Microsoft Windows 应用程序。

WindowViewer 是用于显示在 WindowMaker 中创建的图形窗口的运行时环境。WindowViewer 执行 InTouch QuickScript、执行历史数据的记录和报告、处理报警记录和报告，并且可以充当 DDE 和 SuiteLink 通讯协议的客户机和服务器。

目录

- InTouch 特性
- 系统要求
- 安装InTouch
- 关于本手册
- 技术支持
- 您的 FactorySuite 许可证
- 第一次运行 InTouch
- 使用 InTouch 应用程序管理器

InTouch 特性

InTouch 包括以下特性：

增强的分布式报警系统*

InTouch 的通知系统能通知操作人员有关过程和系统的情况。该系统支持过程报警和系统事件的显示、记录和打印。过程报警是指过程情况的警告，而系统事件则指系统正常状态下的消息。

InTouch 支持分布式报警系统，允许显示由本地 InTouch 应用程序及其它网络 InTouch 应用程序的报警系统产生的报警和事件。并可在本地 InTouch 节点或从网络上的远程节点确认报警。

Terminal Services for InTouch*

终端服务是包括在 Microsoft Windows 2000 Server 操作系统中的可配置服务，能够从服务器统一运行基于 Windows 的 32 位应用程序。终端服务允许计算机同时作为瘦客户机和全功能的个人计算机进行操作，是对分布式计算模式的扩展。计算机可以继续发挥其在现有网络中的作用，同时还可充当瘦客户机来模拟 Windows 2000 Professional 桌面。

扩展的标记名支持*

InTouch 标记名数据字典可支持多达 61,405 个标记。InTouch 60K 标记名许可证允许最大数目的标记名引用。如果安装了 60K 标记名许可证，则每次在窗口或 QuickScript 中激活标记名时，许可证强制逻辑所使用的标记名引用计数会增大；每次引用窗口或 QuickScript 关闭和（或）终止时，引用计数会减小。

在 InTouch 7.11 之前的版本中，在一个应用程序进程内，设计时声明的标记名总数加上有关的远程标记名引用数目必须始终小于或等于 60K。InTouch 7.11 60K 标记名许可证允许应用程序设计人员声明超过 60K 的设计时加远程标记名引用数目。当远程标记名分别进入或退出应用程序进程范围时，系统会动态地增大或减小总的标记名引用计数。

注意：在许可证允许少于最大标记名引用数的情况下，一旦标记名在许可限额中被引用和计数，即使标记名不再使用，限额中的计数也不会减少。

---

* InTouch 7.11 新版本功能

应用程序浏览器

分层结构的应用程序浏览器为您提供了改进的导航能力。例如，它能显示您所创建的所有窗口的名称，当双击某个窗口名称时，此窗口便会打开。当右击某个窗口名称时，屏幕会出现一个菜单，显示您可以执行的各种命令，您可以使用它们来打开窗口、保存窗口、打开窗口的 QuickScript 及其属性对话框等等。应用程序浏览器还让您可以在当前的应用程序中快速访问所有的 InTouch QuickScript 类型，包括：键、条件、数据改变、QuickFunction、ActiveX 脚本、所有配置命令、标记名字典、标记名交叉引用实用程序以及 SuperTag TemplateMaker。应用程序浏览器允许您添加快捷方式，用以启动其它 FactorySuite 程序或第三方应用程序。应用程序浏览器的显示是可选择的。

运行在 Windows 2000 和 Windows NT 操作系统上的应用程序

您在 Windows 2000 或 Windows NT 操作系统下创建的应用程序可以相互交换。这些程序不必转换就可以在其中任一操作系统下运行。

有关所支持的操作系统版本的详细信息，请参阅“系统要求”。

ActiveX 容器

InTouch 是一种 ActiveX 容器。它允许您安装任何第三方 ActiveX控件并且可以在任何应用程序窗口下使用。为方便访问已安装的ActiveX 控件，您可以将其添加到 WindowMaker 的**向导/ActiveX 工具栏**中。通过使用 ActiveX 控件，您可以从 InTouch QuickScript处理所有控件事件、控件方法以及控件属性。您也可以将 ActiveX 控件属性直接与 InTouch 标记名相关联。

远程标记名引用

远程标记名引用允许您从远程数据源中访问数据，而不必在本地标记名字典中建立标记名。远程标记名可以使用 Microsoft DDE 或Wonderware SuiteLink 协议来引用大部分 I/O 数据源中定义的数据。例如， I/O 数据源可以是 Microsoft Excel 或远程视图节点。您可以从任何 InTouch 应用程序导入图形窗口，然后将窗口的占位符标记名转换为远程标记名引用，从而建立不含本地标记名字典的客户应用程序。

SuperTag TemplateMaker

SuperTag TemplateMaker 允许您创建、修改和删除自定义的SuperTag 模板。SuperTag 模板可以最多定义 64 个成员。一个SuperTag 模板可以是另一 SuperTag 模板的成员，即是说 SuperTag 模板最多可以有两层嵌套。成员的行为与常规 InTouch 标记名完全一样，并且可用于 InTouch QuickScript 和动画链接中。与所有标记名**点域**一样，成员也支持趋势和报警。

标记浏览器

标记浏览器允许您从任意 FactorySuite 应用程序或支持InTouch标记名字典接口的任意其它标记名来源选择标记名**点域**，它是编辑标记名字典的主要工具。

仪器故障监视

InTouch 支持三个标记名**点域**（**.RawValue**、**.MinRaw** 和 **.MaxRaw**），您可以在 InTouch QuickScript中使用它们来监视仪器数值，以确定其是否超出范围、需要校准或出现故障。

QuickFunction

QuickFunction 是您创建的 InTouch QuickScript，您可以从其它 QuickScript或动画链接表达式中对其进行调用。QuickFunction 支持参数和返回值。从其它QuickScript 或表达式调用QuickFunction 允许您一次性建立 QuickFunction，然后再多次重复使用它。使用 QuickFunction 减少了应用程序的维护工作，因为无论有多少个其它脚本或动画链接调用 QuickFunction，您只需维护 QuickFunction 本身就可以了。只需修改一个 QuickFunction，您就可以自动更新其它许多 QuickScript 或表达式。

异步 QuickFunction

QuickFunction 可以配置为异步。异步功能在WindowMaker 开发环境中配置，而在WindowViewer 运行时环境下执行。在主WindowViewer 进程运行的同时，异步 QuickFunction 在后台运行。这使得 WindowViewer 可以从主程序流中，将 SQL 数据库调用和 FOR NEXT 循环等耗时性操作分开运行。当通过异步 QuickFunction 执行此类耗时性操作时，所有动画链接和其它 InTouch 功能可同时保持活动状态。

标记名交叉引用

标记名交叉引用实用程序允许您确定标记名和 SuperTag 的使用情况，以及特定标记名用在哪个窗口或 QuickScript 中。为方便起见，在您执行其它任务时，标记名交叉引用实用程序可以在WindowMaker中保持打开状态。它还允许您查看标记名所在的任何 QuickScript 或 QuickFunction。

本地变量

InTouch QuickScript 和QuickFunction 支持本地变量的使用，以存储临时结果和创建包含中间脚本值的复杂计算。在 QuickScript 和 QuickFunction 中运用本地变量，并不会减少您的授权标记名计数。

作为 NT 服务运行的 WindowViewer

WindowViewer 可以作为 NT 服务程序运行，这就为关键 InTouch 组件提供了 NT 服务能力，例如历史记录、提供报警和提供 I/O 数据等。此服务能力保证了在操作系统登录和注销时确保 WindowViewer 连续操作。另外一项功能是当电源出现故障或机器关闭并重启时自动地启动 WindowViewer。这就为在无人工作站中启动 WindowViewer 提供了保障，同时又不致影响 NT 操作系统的安全性。

分布式历史记录

分布式历史趋势系统允许您为趋势图表的每支笔动态地指定不同的历史文件数据源。这使得操作员可以在同一趋势下查看本机 InTouch 历史和 IndustrialSQL 历史。

动态分辨率转换

您现在可以在一种屏幕分辨率下开发应用程序，在另一种分辨率下运行该应用程序，而不致影响原来的应用程序。应用程序也可以不按显示器分辨率而按用户定义的分辨率运行。

动态地址引用

数据源引用可以改变，用单个标记名就可以引用多个数据源。

网络应用程序开发

新的远程开发功能可适应大型、多节点安装系统的要求，包括从一个开发工作站更新网络上的所有节点。

FactoryFocus

FactoryFocus 是 InTouch 的只读运行时版本。它允许监管人员实时查看 HMI 应用程序进程。因为数据不能改变，所以只读版本的系统更加安全。要使用 InTouch FactoryFocus，您不必更改 InTouch 应用程序。

InTouch FactoryFocus 只充当客户机的作用，它不能使用 DDE 或 SuiteLink 写入数据，也不能插入数据到 Excel之类的程序中。报警可以查看但不能确认。FactoryFocus 不能充当 I/O 服务器来向客户机发出请求。象动画链结、标记名、实时和历史趋势等功能也是只读的。

InTouch 的其它特色包括：

可与超过 300 个 I/O 服务器连接。

作为低成本的过程查看工具，价格远低于全功能的 HMI 方案。

由 I/O 服务器提供的 I/O 型标记名的 VTQ（包含相关时间戳和质量的数据值）。

用户可以使用 **HTSelectTag()** 函数，在运行时选择过去记录的任何标记名。

Wonderware SuiteLink 协议。SuiteLink 允许在客户应用程序与服务器应用程序之间传递应用程序命令（读、写和更新）及其相关数据。

可用 Wonderware NetDDE 轻松连网。

实时应用程序进程查看。

具备标准 Windows 2000/NT 的图形用户界面格式。

鼠标右键功能在整个 WindowMaker 内均受支持 ，可快速访问常用命令。

浮动和固定工具栏

可自定义的调色板，提供 1.67 千万种颜色支持（颜色支持仅受显卡性能的限制）。

Windows 2000 和 Windows NT 操作系统对长文件名的支持。

系统要求

要运行 InTouch，我们建议下列的硬件和软件配置：

- 带 Pentium 100或更高处理器的任何 IBM 兼容个人计算机。
- 至少 100 兆的可用硬盘空间。
- 至少 32 MB 的随机存储器 (RAM)。

注意：我们建议每 5K 标记名使用 5MB 的 RAM。例如，32K 的标记名要求 32MB的 RAM 支持；60K 的标记名要求 128MB 的 RAM 支持。

- SVGA 显示适配器（建议2MB RAM）。

注意：建议使用 800x600 或更高的分辨率来查看 WindowMaker，以确保清楚看见所有对话框。

- 定位装置，例如鼠标、轨迹球和触摸屏。
- Microsoft Windows 2000 或 Windows NT 4.0 SP6a（或以上版本的）操作系统
- 如使用 Windows 2000 操作系统，为了实施 InTouch 分布式功能，必须安装和运行 Wonderware NetDDE。

注意：Wonderware FactorySuite InTouch 第 7.11 版（或以上版本）不支持Microsoft Windows NT 3.x或 Microsoft Windows for Workgroups 或 Windows 9x 操作系统。

安装 InTouch

InTouch 使用 Wonderware FactorySuite 安装程序进行安装 。InTouch 在 Microsoft Windows 2000 或 Windows NT 操作系统下运行。安装程序会创建所需的目录，然后将文件从光盘复制到您的硬盘。

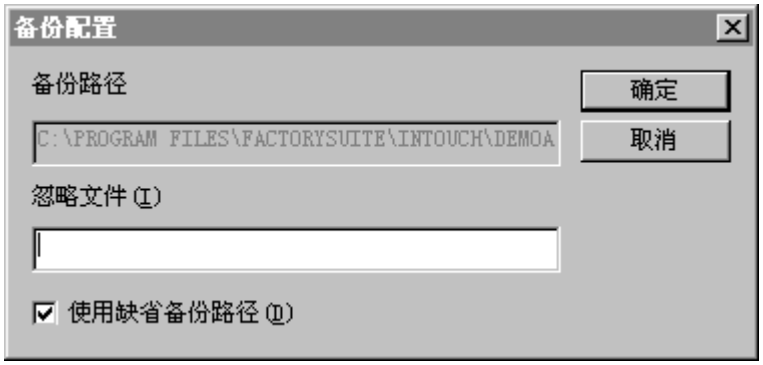
注意：有关完整的 InTouch 安装说明，请阅读 InTouch 版本注释 (ITRELNOTES.TXT)。有关完整的 FactorySuite 安装说明，请阅读 FactorySuite 版本注释 (FSRELNOTES.TXT)。此外，您的联机 *《FactorySuite 系统管理员指南》* 也为您提供了 FactorySuite 软件包中大部分产品的详细安装说明。

从以前的 InTouch 版本升级

所有的 InTouch 版本都可以从以前的版本轻易地升级，您只需要在对话框要求您确认升级时，确认您的应用程序已经备份妥当。当您尝试在 WindowMaker 或 WindowViewer 中打开一个用先前版本的 InTouch 创建的应用程序时，屏幕会出现这两个对话框。

备份旧应用程序

当您尝试打开一个旧应用程序时，WindowMaker 会检测出它是一个旧应用程序，并显示“**备份配置**”对话框来提示您在转换之前先行备份：



要更改缺省备份路径（<应用程序目录>\Bak），先关闭“**使用缺省备份路径**”选项，然后在“**备份路径**”框中，输入用于保存应用程序备份的现有目录路径。如果目录不存在，您必须先创建一个目录，然后才能继续备份。

在“**忽略文件**”框中，您可以指定备份时您要忽略的任何文件（缺省条件下，系统会备份应用程序中的所有文件）。输入每个文件名，文件名之间用分号 (;) 隔开。

- 您可以在文件名中使用标准通配符（'*' 和 '?'）。

关于本手册

本手册由逻辑上连贯的多个章节组成，就建立 InTouch 应用程序的各方面问题分别进行介绍。本书用“程序化”格式写成，通过连续的步骤阐述如何执行大多数的功能和任务。

如果您阅读的是本手册的联机版，则可以在看到绿色文本时，单击该文本来跳到相关的章节。如果您要在跳转后回到原来的章节，可以使用所提供的“后退”选项。

这些“提示”可以告诉您如何用更简便快捷的方法来完成某项功能或任务。

要熟悉 WindowMaker 开发环境及其工具， 请阅读第 1 章“WindowMaker 程序元素”。要了解如何使用窗口、图形对象、向导和 ActiveX 控件等元素，请阅读第 2 章“使用 WindowMaker”。

有关运行时环境 (WindowViewer) 的详细说明，请参阅*《InTouch 运行时用户指南》*。

此外， *《InTouch 参考指南》*为您提供了有关 InTouch QuickScript 语言和函数、系统标记名以及标记名**点域**的深入参考资料。

有关附加程序 SPC Pro 的详细信息，请参阅您的*《SPC Pro 用户指南》*。

有关附加程序“配方管理器”的详细信息，请参阅您的*《配方管理器用户指南》*。

有关附加程序“SQL 访问管理器”的详细信息，请参阅您的*《SQL 访问管理器用户指南》*。

《FactorySuite 系统管理员指南》 也为您提供了有关 FactorySuite 中的一般组件、系统要求、网络连接注意事项、产品集成、技术支持等方面的完整信息。

FactorySuite 软件包还提供了所有 FactorySuite 组件的联机版手册。

注意：要查看或打印联机手册，您必须安装 Adobe Acrobat Reader（4.0 或以上版本）。

假设

本手册假设您：

已经熟悉 Windows 2000 和（或）Windows NT 操作系统工作环境。

懂得如何使用鼠标、Windows菜单、选择选项和访问联机帮助。

有使用编程语言和宏语言的经验。最好理解一些编程概念，如变量、语句、函数和方法等。

推荐参考读物

为获得有关建立有效人机界面的深入信息，我们建议您阅读：

The Windows Interface: An Application Design Guide, Microsoft Press, 1992.

Dreyfuss, Henry. *Symbol Sourcebook: An Authoritative Guide to International Graphic Symbols.* Van Nostrand Reinhold, 1984.

Laurel, Brenda. *The Art of Human-Computer Interface Design*. Addison-Wesley, 1990.

Norman, Donald A. *The Design of Everyday Things*. Doubleday, 1990.

Tufte, Edward. *The Visual Display of Quantitative Information*. Graphics Press, 1983.

Chappell, David. *Understanding Active X and OLE - A Guide for Developer's and Managers*. Microsoft Press, Strategic Technology Series 1996.

技术支持

Wonderware 技术支持提供许多支持选项，来回答有关 Wonderware 产品及其实施的问题。

在联系技术支持前，请先参考*《InTouch 用户指南》*中的有关章节，以寻求您使用 InTouch 系统时所遇问题的可能解决方法。如果您觉得有必要求助于技术支持，请提供以下信息：

1. 您的软件序列号。
2. 您所运行的 InTouch 版本。
3. 您所使用的操作系统类型和版本。例如，Microsoft Windows NT 第 4.0 SP5 版（或以上版本的）工作站。
4. 描述所遇系统错误消息的准确用词。
5. 来自 Wonderware Logger、Microsoft Diagnostic utility (MSD) 或任何其它诊断应用程序的任何相关输出列表。
6. 您所尝试的解决方法的细节和结果。
7. 有关如何重现问题的细节。
8. 如果所遇问题是老问题，请提供指定的 Wonderware 技术支持案例号。

有关技术支持的详细信息，请参阅您的联机*《FactorySuite 系统管理员指南》*。

您的 FactorySuite 许可证

要查看您的 FactorySuite 系统许可证信息，可以从 WindowMaker 帮助菜单的“**关于**”对话框中启动许可证查看程序进行查看。

- **要打开许可证实用程序：**

1. 在 WindowMaker 的“**帮助**”菜单上，单击“**关于**”。
2. 单击“**查看许可证**”。“许可证实用程序 **- LicView**”对话框出现。

有关许可证查看实用程序的详细信息，请参阅您的*《FactorySuite 系统管理员指南》*。

第一次运行 InTouch

您第一次运行 INTOUCH.EXE 时，系统会自动创建 INTOUCH.INI 文件。此文件包含您的应用程序的系统缺省配置。当您配置应用程序时，您的设置将写入 INTOUCH.INI 文件中。

一旦您自定义了应用程序，在创建新的应用程序时， 就可以将自定义的 INTOUCH.INI文件复制到新应用程序的目录中。这样在创建一个新的应用程序时，就不必每次重新设置您的自定义参数。

有关自定义应用程序的详细信息，请参阅第 2 章“使用 WindowMaker”。

-  **要首先运行 InTouch：**

1. 启动 InTouch 程序 (intouch.exe)。“**欢迎使用 InTouch 应用程序管理器**”对话框出现。
2. 单击“**下一步**”，“**欢迎使用 InTouch 应用程序管理器**”对话框再次出现，显示启动目录的缺省路径。例如 **C:\programfiles\factorysuite\intouch**。
3. 要指明不同的目录，请在输入框中输入目录路径，或单击“**浏览**”来寻找目录。
4. 单击“**完成**”。
5. “**InTouch - 应用程序管理器**”出现，并自动搜索您计算机中的当前 InTouch 应用程序。一旦找到应用程序，则会在对话框中显示该应用程序名称及图标。

➢ **要创建新的应用程序：**

1. 在“**文件**”菜单上，单击“**新建**”，或单击工具栏上的“**新建**”工具。“**创建新应用程序**”向导出现。
2. 单击“**下一步**”，“**创建新应用程序**”向导再次出现。缺省条件下，系统将显示 InTouch 目录的路径，后跟 "**NewApp**"。
3. 在输入框中，键入用以创建应用程序的目录路径，或单击“**浏览**”来寻找该目录。
4. 单击“**下一步**”。如果您指定的目录不存在，一个消息对话框会出现，询问您是否要创建一个。单击“**确定**”。“**创建新应用程序**”向导对话框再次出现。
5. 在“**名称**”框中，为新应用程序图标输入一个唯一的名称，当应用程序在“**InTouch 应用程序管理器**”窗口中列出时，将显示该图标。
6. 在“**描述**”框中，输入应用程序的描述文字（描述框是一个可选项，输入的描述内容最长可达 255 个字符）。
7. 单击“**完成**”。“**InTouch - 应用程序管理器**”重新出现，并显示图标及您为新应用程序指定的名称。



8. 要打开某个应用程序，选择该应用程序并单击鼠标右键，然后在“**文件**”菜单中单击您要用于该应用程序的程序名；或者从列表中选择该应用程序，然后在工具栏中单击 WindowMaker 工具（新应用程序不能执行 WindowViewer）。

   要快速打开应用程序，双击其图标，或者在选择应用程序后按 **Enter** 键。

使用 InTouch 应用程序管理器

您可以使用 InTouch 应用程序管理器来创建新应用程序，在 WindowMaker 或 WindowViewer 中打开现有的应用程序、删除应用程序，以及运行 InTouch DBDump 和 DBLoad 标记名字典实用程序。

有关 DBDump 和 DBLoad 程序的详细信息，请阅读第 4 章“标记名字典”。

➢ **要运行 InTouch 应用程序管理器：**

启动 InTouch 程序 (intouch.exe)。“InTouch 应用程序管理器”对话框出现：



当您从列表中选择某个应用程序时，其名称及描述文字将出现在屏幕底部的框中。如果您右击描述框，一个菜单将出现，显示您可以应用到所选文本的命令。

当您选择某个应用程序时并单击鼠标右键时，会出现一个菜单，从此菜单也可以执行 InTouch 应用程序管理器的多个菜单命令。例如：



要对应用程序图标重命名，在列表中右击该应用程序，然后单击“**重命名**”。键入新名称，然后按 **Enter** 键。

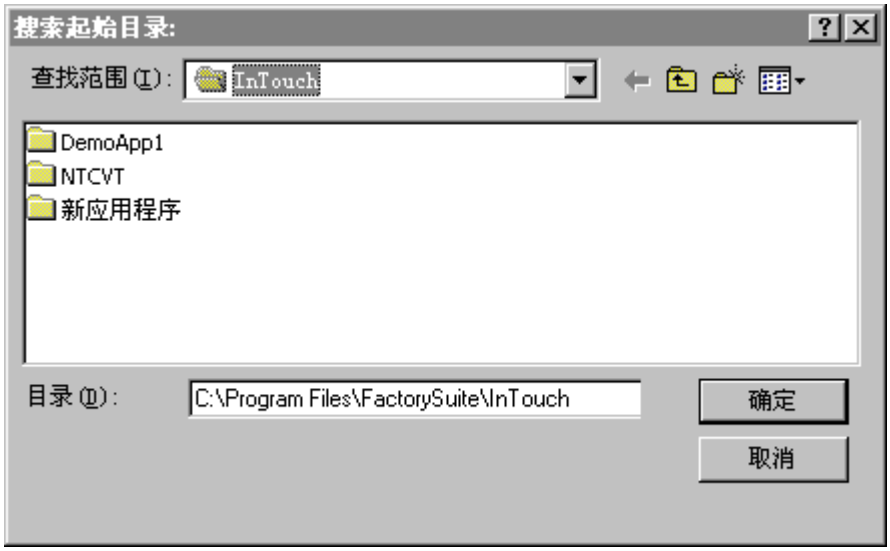
要删除某个应用程序图标，在列表中右击该应用程序，然后单击“**删除**”。屏幕出现一个要求您确认删除的对话框。单击“**是**”从窗口中删除应用程序，或单击“**否**”取消删除。

注意：如果您从列表中删除某个应用程序，它不删除您的文件或应用程序目录。

➢ **要查找应用程序：**

1. 在“**工具**”菜单上，单击“**查找应用程序**”。“**启动目录搜索**”对话框出现：

   要快速查找某个应用程序，在窗口空白区域单击鼠标右键，然后在弹出式菜单中单击“**查找应用程序**”。



2. 找到您要搜索的应用程序目录，然后单击“**确定**”。

   InTouch 应用程序管理器重新出现，显示选定目录中找到的所有应用程序的图标。

- **要查看应用程序属性：**

1. 从列表中选择应用程序。
2. 在“**文件**”菜单上，单击“**属性**”。“**属性**”对话框出现：



- **要查看节点属性：**

  在“**工具**”菜单上，单击“**节点属性**”。“**节点属性**”对话框出现：

  - 要快速访问“**节点属性**”对话框，右击窗口的任意空白区域，然后单击“**节点属性**”。



您可以使用此对话框来配置下列项目：

作为 NT 服务运行的WindowViewer

网络应用程序开发 (NAD)

动态分辨率转换 (DRC)

有关详细信息，请参阅第 3 章“建立分布式应用程序”。

➢ **要配置应用程序管理器的显示窗口：**

1. 在“**查看**”菜单上，单击您要应用的命令，或右击任意列标题，或单击应用程序管理器窗口的空白区域，或在所出现的弹出式菜单上单击详细资料（不是应用程序名）。



2. 指向“**视图**”，将出现下面的子菜单：



3. 选择您要应用的命令。

有关显示命令的详细信息，请参阅“应用程序管理器工具”。

应用程序管理器工具

缺省条件下，当初次运行 InTouch 时，将显示应用程序管理器的工具栏和状态栏。

- **要隐藏工具栏：**

  在“**查看**”菜单上，选择“**工具栏**”。重复此步骤可再次显示工具栏。

- **要隐藏状态栏：**

  在“**查看**”菜单上，选择“**状态栏**”。重复此步骤可再次显示状态栏。

下面简要介绍应用程序管理器工具栏上的每个工具：

工具	描述
	执行“**文件**”菜单上的“**新建**” 命令，以创建一个新的应用程序。
	执行“**文件**”菜单上的“**WindowMaker**” 命令，以打开 WindowMaker 中选定的应用程序。 要在 WindowMaker 中快速打开应用程序，双击其图标，或选择应用程序，然后按 ENTER 键。
	执行“**文件**”菜单上的“**WindowViewer**”命令，以打开 WindowViewer 中选定的应用程序。
	执行“**文件**”菜单上的“**DBLoad**”命令，以运行用于加载标记名字典输入文件的 DBLoad 实用程序。
	执行“**文件**”菜单上的“**DBDump**”命令，以运行用于提取应用程序标记名字典的 DBDump 实用程序。 有关 DBDump 和 DBLoad 程序的详细信息，请参阅第 4 章“标记名字典”。
	执行“**查看**”菜单上的“**大图标**”命令，以显示所列应用程序的大图标。
	执行“**查看**”菜单上的“**小图标**”命令，以显示所列应用程序的小图标。
	执行“**查看**”菜单上的“**列表**”命令，以将对话框改为列表视图模式。
	执行“**查看**”菜单上的“**详细资料**”命令，以将对话框改为详细视图模式。
	执行“**工具**”菜单上的“**节点属性**”命令，以打开用于设置以下计算机属性的“**节点属性**”对话框： 作为 NT 服务运行的 WindowViewer 网络应用程序开发 (NAD) 动态分辨率转换 (DRC) 有关详细信息，请参阅第 3 章“创建分布式应用程序”。

第 1 章

WindowMaker 程序元素

WindowMaker 是 InTouch 的开发环境。WindowMaker 图形用户界面符合 Windows 2000 和 Windows NT GUI 标准。WindowMaker 支持浮动和固定工具栏，遍布整个程序的鼠标右键菜单可快速访问常用命令及可提供 1.67 千万种颜色支持的可自定义调色板（颜色支持仅受显卡性能的限制）。

WindowMaker 的应用程序浏览器，为您提供了一种强有力的图形方法来浏览和配置 InTouch 应用程序。它使您可以轻易地访问 WindowMaker 最常用的命令和功能，例如所有的 Windows 命令、所有的配置命令和所有的 InTouch QuickScript 编辑器。此外，应用程序浏览器会显示所有安装的附加程序，例如 SQL 访问管理器、SPC Pro 和配方管理器，并为您提供了一种可自定义的应用程序启动工具。

为了在 HMI 配置、I/O 服务器配置和控制配置之间快速切换，您可以配置应用程序浏览器来启动任何其它 FactorySuite 程序或者 Windows 程序。

目录

- WindowMaker GUI
- 应用程序浏览器
- WindowMaker 工具栏
- WindowMaker 标尺
- WindowMaker 状态栏
- WindowMaker 调色板
- 弹出式菜单
- 通用窗口对话框特性
- 其它鼠标快捷方式
- 快捷方式和加速键
- 用箭头键移动对象
- 使用 WindowMaker 帮助

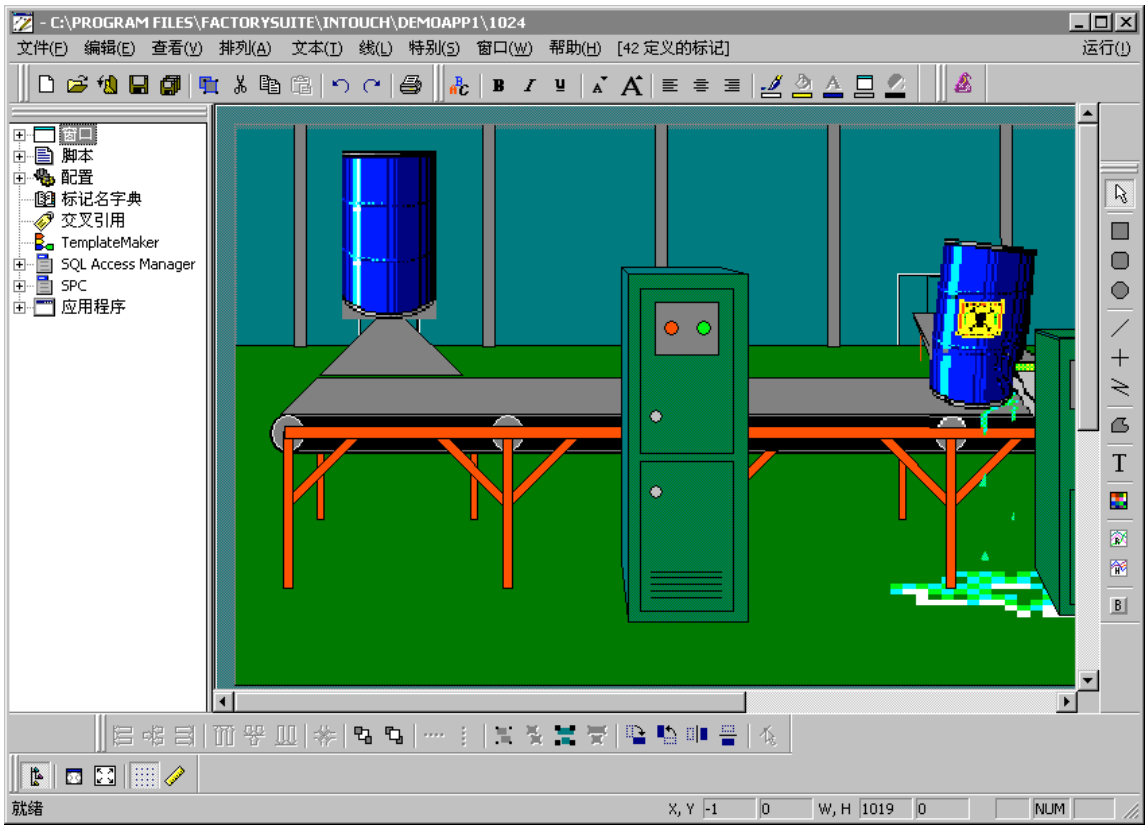
WindowMaker GUI

WindowMaker 支持 Windows 2000 和 Windows NT 操作系统图形用户界面 (GUI) 标准，包括右击鼠标支持、浮动和固定工具栏、下拉菜单、上下文相关帮助等等。

WindowMaker 的开发环境是可配置的。在缺省状态下，当您最初打开 WindowMaker 时，绝大部分可用的程序元素会自动显示，包括所有工具栏、应用程序浏览器和状态栏。然而，在 WindowMaker 窗口下您可以显示或隐藏任何程序元，也可以把工具栏和应用程序浏览器移动到您希望的任何位置。您也可以显示可选标尺，打开或关闭窗口上的可视网格。

有关移动工具栏的详细信息，请参阅“使用浮动/固定工具栏”。

下图说明了 WindowMaker 开发环境的程序元。



当您创建一个新的应用程序并且第一次运行 WindowMaker 时，其程序元将如上图所示自动出现在缺省配置中。

许多工具只有当打开窗口并且在窗口中选定对象时才会被激活。当一个工具未被激活时，其功能对当前窗口状态或所选对象将不适用。

当您关闭 WindowMaker 时，浮动或固定工具栏的位置和大小、应用程序浏览器以及 WindowMaker 窗口尺寸的首选项都会保存起来。当您以后再运行 WindowMaker 时，将保留上次使用的大小和位置。

应用程序浏览器

WindowMaker 的应用程序浏览器是应用程序的分层结构图形视图工具。它显示应用程序中已配置的项目并让您可以轻易地访问这些项目。它还让您可以快速访问 WindowMaker 的许多最常用命令和功能。

注意：您可以配置应用程序浏览器以启动任何其它 FactorySuite 程序或 Windows 程序。这个强大的功能使您能够在 HMI 配置、I/O 服务器配置和控制配置之间快速切换。

不要在应用程序浏览器中添加 WindowViewer (VIEW.EXE)。如果添加了 WindowViewer，那么在 WindowMaker 中创建的新窗口可能会与 WindowViewer 中的窗口不同步。启动 WindowViewer 的正确方法是执行“**文件**”菜单上的“**WindowViewer**”命令，或者单击 WindowMaker 工具栏中的“**运行时**”快速开关。

与所有 WindowMaker 的工具栏一样，应用程序浏览器可以“停放”在 WindowMaker 窗口的任意边缘或“漂浮”在 WindowMaker 窗口中的任何地方。

当您将应用程序浏览器停放在 WindowMaker 窗口的任意边缘时，它会相应地自动调整其大小，并在必要时显示滚动条。当您将应用程序浏览器漂浮在 WindowMaker 窗口中时，它会显示标题栏。与所有 WindowMaker 工具栏一样，当应用程序浏览器漂浮显示时，您可以改变其大小。

有关固定/浮动应用程序浏览器的详细信息，请参阅“使用浮动/固定工具栏”。

如果您右击应用程序浏览器的标题栏，会出现下列菜单：



有关此菜单的详细信息，请参阅“弹出式菜单”。

有关应用程序浏览器中的右击菜单功能的详细信息，请参阅“在应用程序浏览器中导航”。

➢  **要显示/隐藏应用程序浏览器：**

1. 在“**查看**”菜单上，单击“**应用程序浏览器**”（当您初次启动 WindowMaker 时，缺省情况下将显示应用程序浏览器）。
2. 重复第一步，关闭应用程序浏览器。

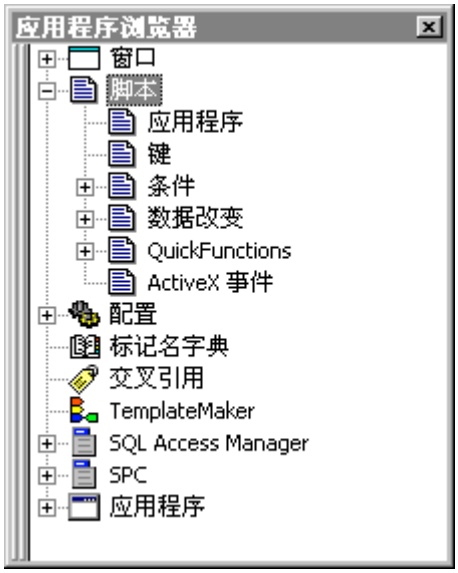
- 要快速隐藏应用程序浏览器，单击“**视图**”工具栏上的应用程序浏览器工具。

  当应用程序浏览器漂浮在 WindowMaker 窗口中时，要快速隐藏应用程序浏览器，单击标题栏上的 ☒ 按扭或者，或者右击标题栏，然后在菜单上单击“**隐藏**”。当您再次显示应用程序浏览器时，它会重新显示上一次在窗口中的大小和位置。

在应用程序浏览器中导航

您可以展开或折叠在应用程序浏览器分层结构图形视图中列出的组。例如，如果您双击某个组，视图将展开并显示这个组的成员。如果您双击某个成员，则将打开该成员。例如，在“**Windows**”组中，如果您双击某个成员窗口名，这个窗口将会打开。如果您双击“**标记名字典**”，“**标记名字典**”对话框将会出现。

- 所有包含成员的组，其前面都会显示一个 ⊞。您可以单击 ⊞ 来快速展开该组并浏览其成员。同样，您可以单击 ⊟ 来折叠组并隐藏其成员。
  例如：



下面的章节将向您简要介绍在执行所述动作时，应用程序浏览器中列出的每个组的行为。

项目	动作	描述
窗口	双击，或单击 ⊞	展开视图以显示应用程序中所有现有的窗口名。
	双击，或单击 ⊟	折叠视图。
	右击	打开弹出式菜单： 单击“**新建**”可打开“**窗口属性**”对话框。
窗口名	双击	打开窗口。
	右击	出现一个您可以应用到选定窗口的弹出式命令菜单。 
脚本	双击，或单击 ⊞	展开视图以显示所有的 InTouch QuickScript 类型。
	双击，或单击 ⊟	折叠视图。
应用程序	双击	在应用程序脚本编辑器中打开应用程序脚本。
	右击	**注意**：出现一个“**打开**”按钮。单击按钮以打开应用程序脚本编辑器。
键	单击 ⊞	展开视图以显示应用程序中的所有键脚本（按所指定的键列出）。
	单击 ⊟	折叠视图。
	双击	打开键脚本编辑器（如果存在脚本，则打开最近编辑过的脚本）。
	右击	出现一个弹出式菜单： 单击“**新键**”可打开键脚本编辑器（如果存在脚本，则打开最近编辑过的脚本）。 单击“**打开**”可打开最近编辑过的脚本。

项目	动作	描述
键脚本名	双击	在键脚本编辑器中打开脚本。
	右击	出现一个弹出式菜单： 单击“新键”可打开键脚本编辑器（您右击的脚本将打开）。 单击“**编辑**”可打开选定的脚本。 单击“**删除**”可删除选定的脚本。
条件	单击 ⊞	展开视图以显示应用程序中的所有条件脚本。
	单击 ⊟	折叠视图。
	双击	打开条件脚本编辑器（如果存在脚本，则打开最近编辑过的脚本）。
	右击	出现一个弹出式菜单 单击“**新键**”可打开条件脚本编辑器（如果存在脚本，则打开最近编辑过的脚本）。 单击“**打开**”可打开最近编辑过的脚本。 单击“**列出依据**”可打开一个子菜单： 单击“**名称**”以按其条件列出脚本 单击“**描述**”以按其注释列出脚本
条件脚本名	双击	在条件脚本编辑器中打开脚本。
	右击	出现一个弹出式菜单： 单击“**新键**”可打开条件脚本编辑器（您右击的脚本将打开）。 单击“**编辑**”可打开选定的脚本。 单击“**删除**”可删除选定的脚本。
数据改变	单击 ⊞	展开视图以显示应用程序中的所有数据改变脚本（按所指定的标记名列出）。
	单击 ⊟	折叠视图。
	双击	打开数据改变脚本编辑器（如果存在脚本，则打开最近编辑过的脚本）。
	右击	出现一个弹出式菜单： 单击“**新建**”可打开数据改变脚本编辑器（如果存在脚本，则打开最近编辑过的脚本）。 单击“**打开**”可打开最近编辑过的脚本。

项目	动作	描述
数据改变脚本名	双击	在数据改变脚本编辑器中打开脚本。
	右击	出现一个弹出式菜单： 单击“**新建**”可打开数据改变脚本编辑器（您右击的脚本将打开）。 单击“**编辑**”可打开选定的脚本。 单击“**删除**”可删除选定的脚本。
QuickFunction	单击 ⊞	展开视图以显示应用程序中的所有 QuickFunction（按函数名列出）。
	单击 ⊟	折叠视图。
	双击	打开 QuickFunction 编辑器（如果存在 QuickFunction，则打开最近编辑过的 QuickFunction）。
	右击	出现一个弹出式菜单： 单击“**新建**”可打开 QuickFunction 编辑器（如果存在 QuickFunction，则打开最近编辑过的 QuickFunction）。 单击“**打开**”可打开最近编辑过的 QuickFunction。
QuickFunction 名称	双击	在 QuickFunction 编辑器中打开 QuickFunction。
	右击	出现一个弹出式菜单： 单击“**新建**”可打开 QuickFunction 编辑器（您右击的脚本将打开）。 单击“**编辑**”可打开选定的 QuickFunction。 单击“**删除**”可删除选定的 QuickFunction。
ActiveX 事件	单击 ⊞	展开视图以显示应用程序中的所有 ActiveX 事件脚本（按事件名列出）。
	单击 ⊟	折叠视图。
ActiveX 事件脚本名	双击	在 ActiveX 事件脚本编辑器中打开脚本。
	右击	出现一个弹出式菜单： 单击“**编辑**”可打开选定的脚本。 单击“**删除**”可删除选定的脚本。

项目	动作	描述
配置	双击，或单击 ⊞	展开视图以显示 WindowMaker 的多个配置命令以及“**向导/ActiveX 安装**”命令。
	双击，或单击 ⊟	折叠视图。
配置项目名	双击	打开相应的对话框。
	右击	出现一个“**打开**”按钮。单击按钮可打开该项目的相应对话框。
标记名字典	双击	打开“**标记名字典**”对话框，显示最近修改过的标记名定义。否则，显示缺省 **$AccessLevel** 系统标记名。
	右击	出现一个“**打开**”按钮。单击按钮可打开“**标记名字典**”对话框，显示最近修改过的标记名定义。否则，显示缺省 **$AccessLevel** 系统标记名。
交叉引用	双击	打开“**交叉引用**”实用程序。
	右击	出现一个“**打开**”按钮。单击按钮可打开“**交叉引用**”实用程序。
TemplateMaker	双击	打开 SuperTag **TemplateMaker** 实用程序。
	右击	出现一个“**打开**”按钮。单击按钮可打开 SuperTag **TemplateMaker** 实用程序。
附加程序	双击，或单击 ⊞	展开视图以显示附加程序的配置命令。
	双击，或单击 ⊟	折叠视图。
程序名	双击	打开相应的对话框。
	右击	出现一个“**打开**”按钮。单击按钮可打开命令的相应对话框。 **注意**：附加程序只有在安装后才会出现在应用程序浏览器中。
应用程序	双击，或单击 ⊞	展开视图以显示您可以从 WindowMaker 启动的所有其它应用程序。
	双击，或单击 ⊟	折叠视图。
	右击	出现一个“**新建**”按钮。单击按钮可在应用程序浏览器中添加一个应用程序。

项目	动作	描述
应用程序名	双击	不退出 WindowMaker 而启动应用程序。
	右击	出现一个弹出式菜单： 单击“**新建**”可打开一个空白“应用程序属性”对话框。使用此对话框可添加新的应用程序。 单击“**运行**”可启动应用程序。 单击“**删除**”将从应用程序浏览器中删除应用程序。 单击“**属性**”可打开选定应用程序的“应用程序属性”对话框。

添加应用程序到应用程序浏览器

WindowMaker 应用程序浏览器最强大的功能之一是从 WindowMaker 启动其它 FactorySuite 及第三方 Windows 应用程序。

例如，您可以在开发应用程序的同时，运行 I/O 服务器程序并对其进行配置。您可以启动常用的第三方 Windows 程序，例如 Windows 记事本、写字板、Microsoft Excel、Microsoft Word 和 Microsoft Paint 等。

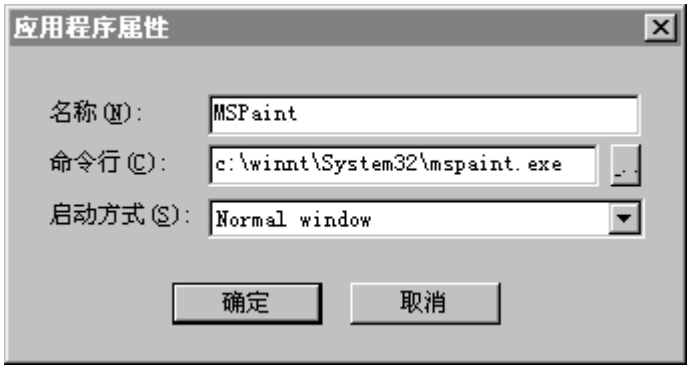
- InTouch 附加程序、SQL 访问管理器、SPC Pro 和配方管理器等在安装时会自动添加到应用程序浏览器中。

警告：不要在应用程序浏览器中添加 WindowViewer (VIEW.EXE)。如果添加了 WindowViewer，那么在 WindowMaker 中创建的新窗口可能会与 WindowViewer 中的窗口不同步。启动 WindowViewer 的正确方法是执行“**文件**”菜单上的“**WindowViewer**”命令，或者单击 WindowMaker 工具栏中的“**运行时**”快速开关。

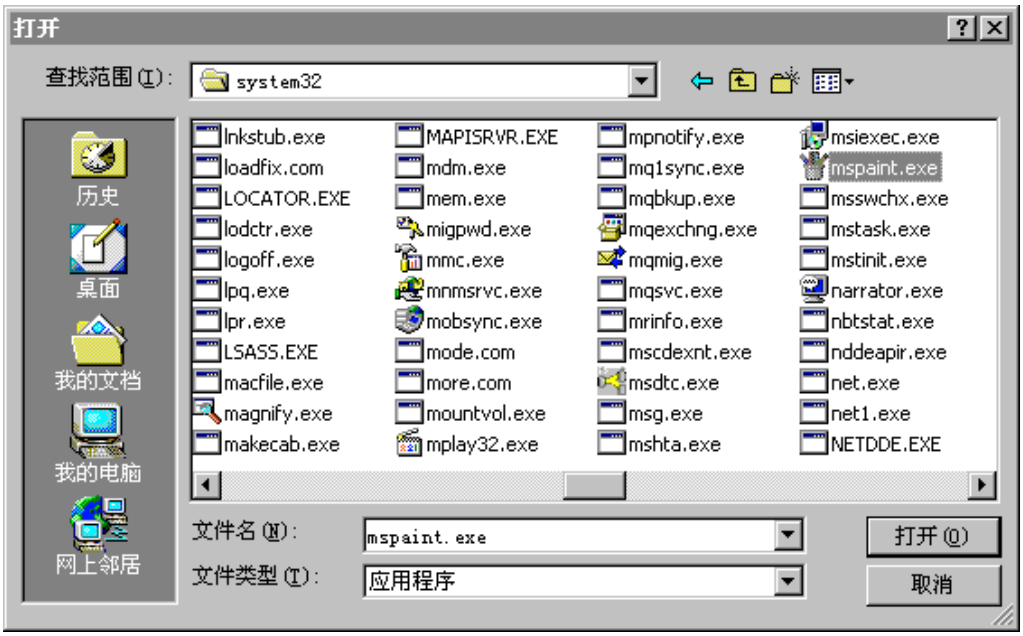
您也可以配置应用程序浏览器，以在程序中打开特定的文档或电子表格。例如，如果您选择了某个 Microsoft Word 文档或 Microsoft Excel 电子表格，则当您在应用程序浏览器中双击该应用程序的图标时，应用程序会启动并自动显示您选定的文档或电子表格。这些文档会显示其原先创建的应用程序的图标，或配置为关联应用程序的 .exe。

➢ **要在应用程序浏览器中添加一个应用程序：**

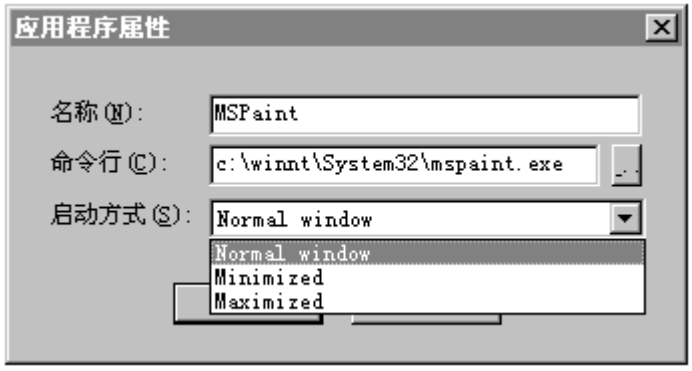
1. 显示应用程序浏览器。
2. 右击“**应用程序**”，出现一个“**新建**”按钮。
3. 单击“**新建**”，“**应用程序属性**”对话框出现：



4. 在“**名称**”框中，输入您要显示在应用程序浏览器中的应用程序名称。
5. 在“**命令行**”框中，输入应用程序的完整路径，或单击 (...) 按钮。“**打开**”对话框出现：



6. 找到应用程序，然后单击“**打开**”。“**应用程序属性**”对话框再次出现：



- 您可以在“**命令行**”字段中，为应用程序添加可选的命令行参数。

7. 单击“**启动方式**”箭头，选择从 WindowMaker 运行应用程序的方式。

8. 单击“**确定**”。

应用程序即被添加到应用程序浏览器的“**应用程序**”下方。现在，您可以随时从 WindowMaker 运行此应用程序。

WindowMaker 工具栏

WindowMaker 工具栏上的工具按公共功能分组。例如，“**排列**”工具栏包含了可用于快速应用“**排列**”菜单上的大部分命令的工具。

如果您将光标置于工具上，一个工具提示框会出现，显示此工具的名称。例如：



使用浮动/固定工具栏

WindowMaker 工具栏具有“浮动和固定”功能。这意味着您可以将任何工具栏从其缺省“固定”位置移动到 WindowMaker 窗口的任意边缘，或将其放在 WindowMaker 窗口顶部的工具栏区域。您也可以从位于窗口边缘的固定位置移动固定工具栏，以让其浮动在窗口中。浮动工具栏带有允许您改变其大小的标题栏。

- 与其它工具栏一样，应用程序浏览器也可以固定或浮动在窗口中的任何位置，并且其大小也可以改变。

有关应用程序浏览器的详细信息，请参阅“应用程序浏览器”。

➢ **要改变固定工具栏在窗口中的位置：**

1. 单击工具栏上的“灰色条”，或单击固定工具栏的空白区域。
2. 按住鼠标左键不放，将工具栏拖离窗口边缘、工具栏区域或 WindowMaker 窗口的任意边缘。
3. 把工具栏移动到窗口另一边或工具栏区域的新位置。
   - 如果您将水平固定工具栏移动到 WindowMaker 窗口的左边或右边，则它在固定时会自动变成缺省的垂直形状。同样，如果您将垂直工具栏移到窗口的顶部或底部，则它在固定时会变成缺省的水平形状。
4. 释放鼠标，工具栏就固定在新位置上了。

注意：当工具栏固定后，您不能改变其大小，也不能访问其右击菜单。

➢ **要显示/隐藏固定工具栏：**

1. 在“**查看**”菜单上，选择工具栏的名称（当初次启动 WindowMaker 时，缺省条件下会显示所有的工具栏）。
2. 重复第一步以切换显示/隐藏。

   当您再次显示被隐藏的固定工具栏时，它会再次出现在窗口中上次的固定位置上。



➢ **要浮动固定工具栏：**

1. 单击工具栏上的“灰色条”，或单击固定工具栏的空白区域。
2. 按住鼠标按钮不放，将工具栏从其固定位置拖到 WindowMaker 窗口中的新位置。
3. 释放鼠标按钮，工具栏将出现并显示如下：



   您可以改变浮动工具栏的大小。

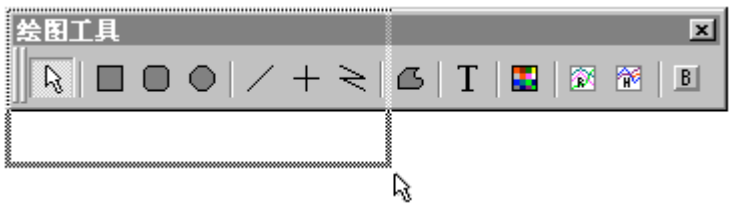
   如果您将浮动工具栏固定在 WindowMaker 窗口的左边或右边，它会在固定时自动变成缺省的垂直形状。同样，如果您将工具栏移动到 WindowMaker 窗口的顶部或底部，它会在固定时变成缺省的水平形状。

   当您固定一个大小已改变的工具栏时，也会发生这种情况。但是，当您在窗口中再次浮动工具栏时，它会恢复其原来的浮动大小。

➢ **要改变浮动工具栏的大小：**

1. 将鼠标移动到工具栏的任一边缘之上。光标将变成一个双向箭头。
2. 单击工具栏边缘，按住鼠标按钮不放并拖动鼠标，以改变工具栏大小。

   当您移动鼠标时，会出现一个方框，指明您释放鼠标时的工具栏大小。例如：



3. 当工具栏达到所需尺寸时，释放鼠标。

   当您右击浮动工具栏的标题栏时，会出现一个菜单，显示您可以应用到工具栏的命令。

有关此菜单的详细信息，请参阅“弹出式菜单”。

➢ **要隐藏/显示浮动工具栏：**

1. 要隐藏浮动工具栏，在“**查看**”菜单上选择工具栏名称，或者右击工具栏的标题栏，然后单击菜单上的“**隐藏**”。

   要快速隐藏工具栏，单击工具栏标题栏的 按钮。

2. 要显示已隐藏的浮动工具栏，在“**查看**”菜单上，选择工具栏的名称。

   工具栏会以先前的大小重新出现在先前的位置上。

➢ **要一次隐藏所有工具栏：**

1. 在“**查看**”菜单上，单击“**隐藏全部**”，或者在“**视图**”工具栏中单击“全部隐藏/恢复”工具。应用程序浏览器的所有工具栏都会隐藏起来。

2. 重复第一步以再次显示所有工具栏。

   在“**查看**”菜单上，所有显示的工具栏其名称前面都有一个复选标记。

通用工具栏

“**通用**”工具栏包括用于执行“**文件**”菜单上的大多数窗口命令的工具，以及“**编辑**”菜单上的 Microsoft Windows 剪贴板工具。



当您右击某个打开窗口的空白区域，或者在应用程序浏览器中的“**窗口**”下右击某个窗口名时，所出现的菜单也会包含大多数的这些窗口命令。

以下简要介绍每个工具：

工具	描述
	执行“**文件**”菜单上的“**新建窗口**”命令，以打开“**窗口属性**”对话框来创建一个新窗口。
	执行“**文件**”菜单上的“**打开窗口**”命令，以打开“**打开窗口**”对话框，列出您可以选择打开的现有窗口名。
	执行“**文件**”菜单上“**关闭窗口**”命令，以打开“**关闭窗口**”对话框，列出您可以选择关闭的所有已打开窗口名。
	执行“**文件**”菜单上的“**保存窗口**”命令，以打开“**保存窗口**”对话框，列出自上次保存以来修改过的所有已打开窗口名。
	自动保存所有自上次保存以来修改过的已打开窗口。此工具不会要求您逐一确认每个窗口。它会自动保存所有修改过的窗口。
	执行“**编辑**”菜单上的“**复制**”命令，以复制窗口中当前选定的对象。
	执行“**编辑**”菜单上的“**剪切**”命令，以从窗口中剪下当前选定的对象，并将其复制到 Windows 剪贴板上。
	执行“**编辑**”菜单上的“**复制**”命令，以复制当前选定的对象，并将其复制到 Windows 剪贴板上（被复制的对象不会从窗口中删除）。
	执行“**编辑**”菜单上的“**粘贴**”命令，以粘贴任何剪下或复制到 Windows 剪贴板的对象（光标会变成粘贴模式。在窗口中单击，可以将复制或剪切对象粘贴到窗口中）。
	执行“**编辑**”菜单上的“**撤消**”命令，以取消最后一次应用到对象的动作或命令。
	执行“**编辑**”菜单上的“**恢复**”命令，以取消最后一次应用到对象的撤消动作或命令。 撤消/恢复的缺省层数设置为 10。您可以在“**WindowMaker 属性**”对话框中将“**撤消的级别**”增大为 25。要访问此对话框，在应用程序浏览器的“**配置**”下，双击“**WindowMaker**”，或者在“**特殊**”菜单下，指向“**配置**”，然后单击子菜单上的“**WindowMaker**”。
	执行“**文件**”菜单上的“**打印**”命令，以打开“**WindowMaker 打印输出**”对话框，用于打印数据库和窗口信息，以及 QuickScript。

向导/ActiveX 工具栏

“**向导/ActiveX**”工具栏在缺省条件下只包含用于访问向导“**选择**”对话框的向导工具。但是，您可以将任何已安装的向导或 ActiveX 控件添加到这个工具栏中。例如：



以下简要介绍向导工具：

工具	描述
	显示“**向导选择**”对话框，用于选择要粘贴到窗口的向导。

格式工具栏

“**格式**”工具栏包括用于执行“**文本**”菜单上的大多数文本对象格式命令的工具。它也包含用于访问调色板的工具，以选择线、文本、填充色、窗口背景和透明对象颜色等。



以下简要介绍每个工具：

工具	描述
	执行“**文本**”菜单上的“**字体**”命令，以打开“**字体**”对话框来选择字体、字形和大小。
	执行“**文本**”菜单上的“**粗体**”命令，以便对选定的单个或多个文本字符串和数值字段应用“**粗体**”字形。
	执行“**文本**”菜单上的“**斜体**”命令，以便对选定的单个或多个文本字符串和数值字段应用“**斜体**”字形。
	执行“**文本**”菜单上的“**下划线**”命令，以便对选定的单个或多个文本字符串和数值字段应用“**下划线**”字形。
	执行“**文本**”菜单上的“**字体缩小**”命令，以缩小任何字体的大小。要使用此命令，在选择文本字符串后单击“**文本**”工具栏上的此工具。
	执行“**文本**”菜单上的“**字体放大**”命令，以放大任何字体的大小。要使用此命令，在选择文本字符串后单击“**文本**”工具栏上的此工具。
	执行“**文本**”菜单上的“**左对齐**”命令，以使选定的单个或多个文本字符串和数值字段左对齐。
	执行“**文本**”菜单上的“**居中**”命令，以使选定的单个或多个文本字符串和数值字段居中对齐。
	执行“**文本**”菜单上的“**右对齐**”命令，以使选定的单个或多个文本字符和数值字段右对齐。
	打开调色板来选择线对象或对象轮廓的颜色。
	打开调色板来选择对象的填充色。
	打开调色板来选择文本对象的颜色。
	打开调色板来选择窗口的背景色。
	打开调色板来选择位图对象的透明颜色。

绘图对象工具栏

“绘图对象”工具栏包括用于绘制画形对象的所有工具，您既可以在窗口中画简单的图形对象，如长方形、椭圆、线或文本对象，也可以画复杂的对象，如实时趋势、历史趋势、位图和带标签的三维按钮：



以下简要介绍每个工具：

工具	描述
	用于选择窗口对象的选择器模式。
	用于画长方形或正方形的长方形工具。
	用于画圆角长方形或圆角正方形的圆角长方形工具。
	用于画椭圆或圆的椭圆工具。
	用于以任意角度画线条的线条工具。
	用于画水平线或垂直线的线条工具。
	用于画多边线的线条工具。
	用于画多边形对象的多边形工具。
	用于输入文本对象的文本工具。
	用于画位图容器的位图工具，以便直接从 Windows 粘贴板或下面任一文件类型粘贴位图：.BMP、.JPG、.PCX 或 .TGA。
	用于绘制实时趋势对象的实时趋势工具。
	用于绘制历史趋势对象的历史趋势工具。
	用于绘制带标签的三维按钮的按钮工具。

视图工具栏

“**视图**”工具栏包括用于执行“**查看**”菜单上的大部分窗口命令的工具。这些命令用于控制 WindowMaker 窗口的状态。



以下简要介绍每个工具：

工具	描述
	切换打开和关闭“**查看**”菜单上的“**应用程序浏览器**”命令，以显示/隐藏应用程序浏览器。
	切换打开和关闭“**查看**”菜单上的“**隐藏全部**”命令，以隐藏/显示所有固定工具栏。 当全部隐藏模式被激活时，WindowMaker 的总体大小保持不变。要返回正常模式，单击浮动**视图工具栏**上的“**全部隐藏/恢复**”工具，或单击“**查看**”菜单上的“**全部隐藏**”。 在全部隐藏模式下，所有浮动工具栏都可看见，**视图工具栏**会自动漂浮在 WindowMaker 的上方。如果您在全部隐藏模式下固定任一浮动工具栏，则此模式会自动终止。
	切换打开和关闭“**查看**”菜单上的“**全屏幕**”命令，以使显示模式从正常视图转为全屏。 要返回到正常模式，单击浮动**视图工具栏**上的全屏工具，或单击“**查看**”菜单上的“**全屏幕**”。 在全屏模式下，除了打开的窗口和浮动工具栏外，所有的 WindowMaker 程序元均隐藏起来。**视图工具栏**会自动漂浮在 WindowMaker 的上方。 在全屏模式下，客户区域的坐标保持不变。例如，左上角坐标仍为 (0,0)。全屏模式在最大化客户区域后会自动设定坐标、隐藏标题栏和菜单栏，并调整客户区域来模仿视图的全屏模式。
	切换打开和关闭“**排列**”菜单下的“**对齐网格**”命令，以显示/隐藏用于使对象对齐的可见网格。使用“**排列**”菜单下的“**对齐网格**”命令。 如果没有选定“**WindowMaker 属性**”对话框中的“**对齐网格**”选项，则此工具将不起作用。
	切换打开和关闭“**查看**”菜单上的“**标尺**”命令，以显示/隐藏标尺。

有关标尺的详细信息，请参阅“WindowMaker 标尺”。

排列工具栏

“**排列**”工具栏包括用于执行“**排列**”菜单上的大多数对象排列命令的工具。



以下简要介绍每个工具：

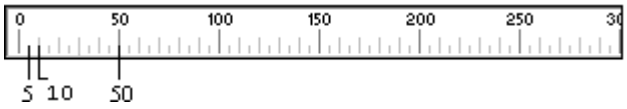
工具	描述
	执行“**排列/对齐**”子菜单上的“**左对齐**”命令。以最左侧选定对象的左边缘为基准，对齐所有选定对象的左边缘。
	执行“**排列/对齐**”子菜单上的“**水平居中对齐**”命令。以选定对象组的中心线为基准，对齐所有选定对象的中心线。
	执行“**排列/对齐**”子菜单上的“**右对齐**”命令。以最右侧选定对象的右边缘为基准，对齐所有选定对象的右边缘。
	执行“**排列/对齐**”子菜单上的“**顶部对齐**”命令。以最顶端选定对象的顶边缘为基准，对齐所有选定对象的顶边缘。
	执行“**排列/对齐**”子菜单上的“**垂直居中对齐**”命令。以对象组的中心为基准，对齐所有选定对象的中心。
	执行“**排列/对齐**”子菜单上的“**底部对齐**”命令。以最底端选定对象的底边缘为基准，对齐所有选定对象的底边缘。
	执行“**排列/对齐**”子菜单上的“**中心点对齐**”命令。以选定对象组的中心点为基准，对齐所有选定对象组的中心点。
	执行“**排列**”菜单上的“**置后**”命令，将所有选定对象置于所有未选定对象之后。
	执行“**排列**”菜单上的“**置前**”命令，将所有选定对象置于所有未选定对象之前。
	执行“**排列**”菜单上的“**水平间距**”命令，使所有选定对象均匀地水平分布在最左端与最右端的选定对象之间。
	执行“**排列**”菜单上的“**垂直间距**”命令，使所有选定对象均匀地垂直分布在最顶端与最底端的选定对象之间。
	执行“**排列**”菜单上的“**制作符号**”命令，将多个对象合并成一个称为符号的整体。
	执行“**排列**”菜单上的“**分解符号**”命令，将一个符号分解成不同的元件。
	执行“**排列**”菜单上的“**制作单元**”命令，将多个选定对象合并成一个称为单元的整体。在合并单元时，每个单元会被保留，所以当分解合并单元时，仍可恢复原始单元。

工具	描述
	执行“**排列**”菜单上的“**分解单元**”命令，以分解选定的单元。在合并单元时，每个单元会被保留，所以当分解合并单元时，仍可恢复原始单元。
	执行“**排列**”菜单上的“**顺时针旋转**”命令，使选定对象顺时针旋转 90 度。
	执行“**排列**”菜单上的“**逆时针旋转**”命令，使选定对象逆时针旋转 90 度。
	执行“**排列**”菜单上的“**水平翻转**”命令，以水平翻转选定的对象。
	执行“**排列**”菜单上的“**垂直翻转**”命令，以垂直翻转选定的对象。
	打开“**编辑**”菜单上的“**改变对象形状**”命令，以改变多边线或多边形的形状。

WindowMaker 标尺

您可以使用 WindowMaker 标尺来精确对齐窗口中的对象。

小刻度标志的间距为 5 个象素；中等刻度标志的间距为 10 个象素；带数字刻度标志的间距为 50 个象素。例如：



标尺的 10 和 50 个象素间距的增量相当于您按住 SHIFT 或 CTRL 键并按向上、向下、向左、向右箭头键时，选定对象所移动的象素距离。

例如，如果您要一次将对象移动 10 个象素，可在按住 SHIFT 键的同时，按下相应的箭头键；要将对象一次移动 50 个象素，可在按住 CTRL 键的同时，按下相应的箭头键。

如果您选择对象并只按箭头键，则对象一次将移动 1 个象素。

当您需要精确地对齐和调整位置时，这些功能将会很有用处。

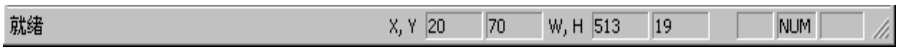
有关使用箭头键的详细信息，请参阅“用箭头键移动对象”。

➢  **要显示/隐藏标尺：**

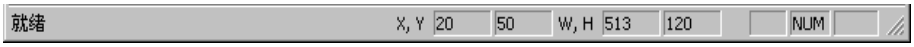
1. 在“**查看**”菜单上，单击“**标尺**”，或者单击“**查看**”菜单上的标尺工具。
2. 重复第一步可隐藏标尺。

WindowMaker 状态栏

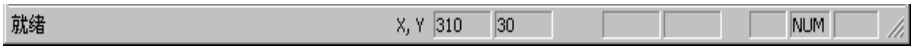
当您选择窗口对象时，WindowMaker 状态栏会显示对象的左上角 X 和 Y 象素坐标，以及对象的象素高度和宽度。例如：



当您选择多个对象时，状态栏会显示最左端对象的左边缘坐标 (X) 和最顶端对象的顶边缘坐标 (Y)，同时还显示整个组的宽和高。例如：



当您单击窗口的空白显示区域，状态栏会显示窗口中光标当前位置的 X 和 Y 坐标。例如：



- **要显示/隐藏状态栏：**

  1. 在“**查看**”菜单上，单击“**状态栏**”。
  2. 重复第一步可隐藏“**状态栏**”。

WindowMaker 调色板

WindowMaker 调色板提供多达 1.67 千万种颜色支持（颜色支持仅受显卡性能的限制）。缺省条件下，调色板为您提供了范围广泛的颜色选择。但是，您可以创建自己的自定义调色板。您的自定义调色板可以加载到 WindowMaker 调色板或者从 WindowMaker 调色板导出。

使用标准调色板

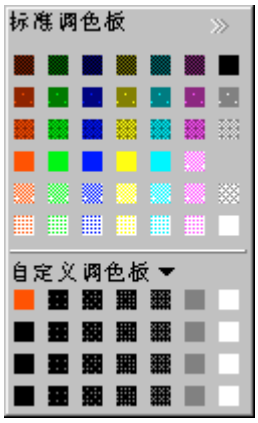
WindowMaker 调色板用于在线条、长方形、圆角长方形、椭圆、多边线、多边形以及文字的静态和动态属性中添加颜色，也可用于选择窗口的背景色以及位图的透明颜色，以便查看位图后面的对象。

有关透明位图的详细信息，请参阅第 2 章“使用 WindowMaker”。

每当您单击对话框中的颜色方块，或者单击某个颜色工具并将线条、填充色或文本颜色应用到选定对象时，就会出现调色板。

➢ **要使用标准调色板：**

1. 在“**标准调色板**”区域中单击您要使用的颜色，以选择标准颜色（调色板将关闭，同时应用您所选定的颜色）。



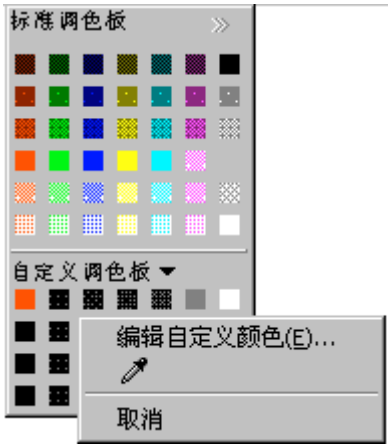
2. 单击右角的“>>”，从 InTouch 的 32 种典型颜色（InTouch 第 7.0 版之前的调色板颜色）中选择一种。

创建自定义调色板

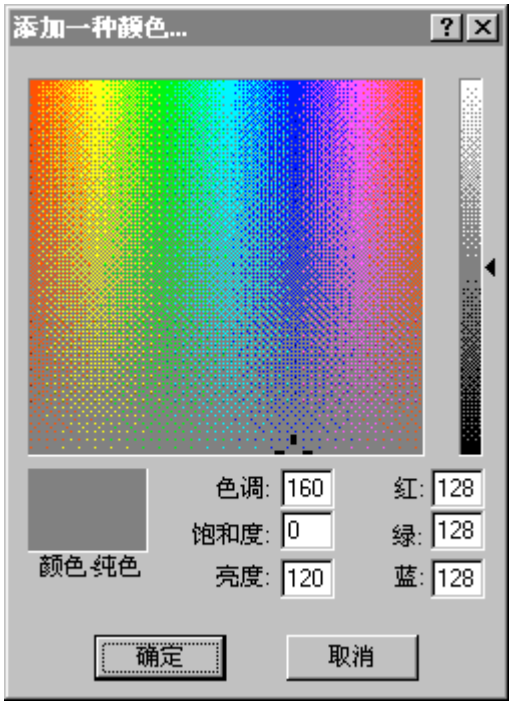
WindowMaker 调色板允许您自定义颜色并将其添加到您的调色板中，它还允许您导入其它 Windows 应用程序中创建的调色板，并将其添加到标准调色板中。您也可以将自定义调色板导出到其它 Windows 应用程序中。

➢ **要创建自定义颜色：**

1. 打开调色板。
2. 右击调色板底部“**自定义调色板**”区域中的某个空白方块。出现下面的菜单：



3. 单击“**编辑自定义颜色**”。“**添加一种颜色**”对话框出现：



4. 单击表中的任意位置，使用对话框右边的滑块来调整颜色属性。

色调、饱和度和亮度

您可以使用色调、饱和度和亮度的组合来定义任何一种颜色。 。色调是彩色转盘的值，其中 0 是红色、60 是黄色、120 是绿色、180 是青色、200 是深红色、240 是蓝色；饱和度是指定色调中的颜色数量，最大为 240。亮度指颜色的亮度。如果您改变其中的任意值，则红、绿、蓝比例也会相应改变。

试用不同颜色的最简单方法，是按住鼠标不放，然后在颜色表中移动光标。

红、绿、蓝

您可以使用红、绿、蓝不同比例的组合来定义任何一种颜色，在颜色表中可以看到改变这些值的效果。如果您改变这些值，色度、饱和度和亮度值也将相应改变。

在使用“**色度**”、“**饱和度**”、“**亮度**”或“**红**”、“**绿**”、“**蓝**”比例来定义颜色时，您可以在“**颜色|实心**”框中查看该颜色，以确保所定义的颜色符合要求。

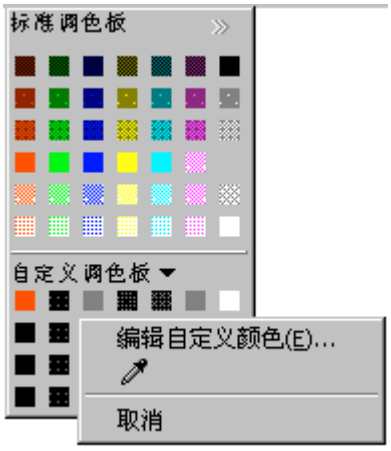
左边的“**颜色**”框显示指定颜色的黑色和白色数量；_右边的“**实心**”框显示您选择不含黑白色的 100% 彩色时的颜色样式。要调整颜色，可使用对话框右边的滑块；要指定不含黑白色的 100% 彩色，可按 ALT + O。

5. 单击“**确定**”。您选定的颜色将添加到您原来在调色板中单击的方块中。

➢ **要用吸管工具来选择自定义颜色：**

1. 打开调色板。

2. 右击调色板底部“**自定义调色板**”区域中的某个空白方块。出现下面的菜单：

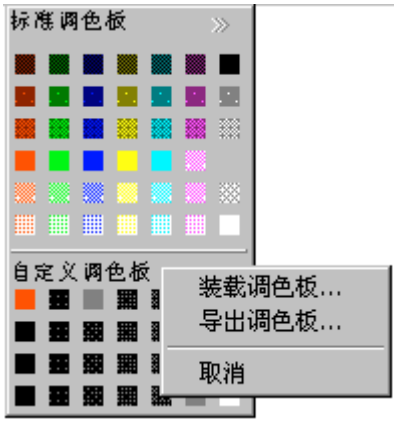


3. 单击吸管工具，然后单击您要添加到调色板的“**自定义调色板**”区域中的颜色。您可以选择 WindowMaker 窗口内或 WindowMaker 之外任意位置的任何颜色。此功能主要在创建透明位图时使用。

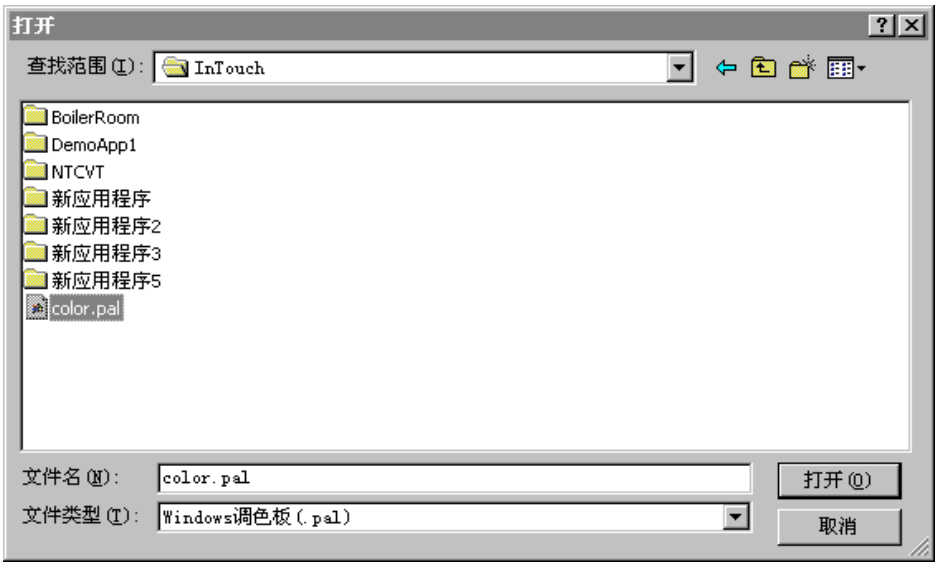
有关创建透明位图的详细信息，请参阅第 2 章“使用 WindowMaker”。

➢ **要导入自定义调色板：**

1. 打开调色板。
2. 单击“**自定义调色板**”箭头。出现下面的菜单：



3. 单击“**装载调色板**”。出现标准 Windows 的“**打开**”对话框：



4. 找到并选择调色板 (.PAL) 文件，然后单击“**打开**”或者双击文件名。调色板中包含的颜色将加载到调色板的“**自定义调色板**”区域中。

➢ **要导出自定义调色板：**

1. 打开调色板。
2. 单击“**自定义调色板**”箭头，然后单击菜单上的“**导出调色板**”（如上所示）。
3. 出现标准的 Windows“**另存为**”对话框。指定用以保存调色板的文件名，然后单击“**保存**”。

   调色板必须用 .PAL 扩展名进行保存。

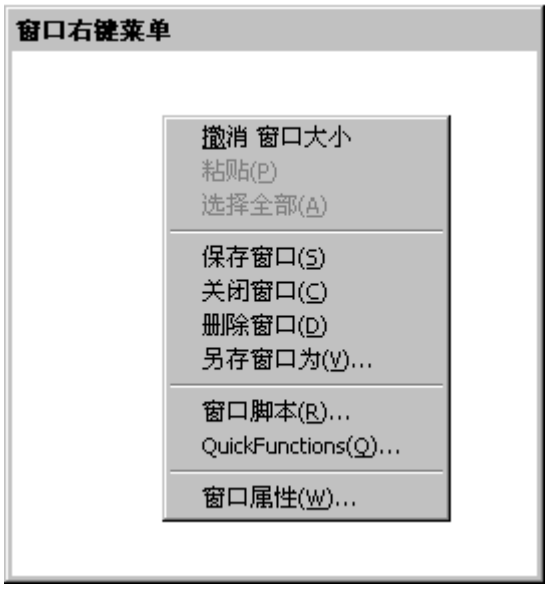
弹出式菜单

InTouch 支持鼠标右键功能，可显示窗口和图形对象的常用命令。弹出式菜单包含可应用于对话框、文本框、动画链接标记名或表达式框中的选定文本的命令，在 QuickScript 窗口中也支持弹出式菜单。您不必使用标准菜单来寻找您要使用的命令，而只需在应用程序浏览器中右击窗口、对象、对话框、文本框或组及其成员。

要关闭右击功能，在您的 INTOUCH.INI 文件中添加一行：**oldrightmousebehavior**=1 。

➢ **要访问窗口右击菜单：**

1. 右击窗口的空白区域。出现下面的菜单：



2. 单击菜单上您要使用的命令（不适用于当前窗口状态的命令将不被激活）。

➢ **要访问图形对象右击菜单：**

1. 右击窗口中的对象。出现下面的菜单：



2. 单击菜单上您要使用的命令（不适用于当前对象状态的命令将不被激活）。

   如果某个命令存在“子菜单”，则会显示一个箭头。要选择子菜单中的命令，先指向初始右击菜单中的命令，然后单击子菜单中您要应用到对象的命令。

➢ **要访问对话框文本右击菜单：**

1. 在任意 WindowMaker 对话框中，右击文本框。出现下面的菜单：



2. 单击您要应用于选定文本的命令（不适用于当前文本状态的命令将不被激活）。当选定部分文本或者没有选定文本时，“**全部选定**”命令将变成活动状态。

➢ **要访问浮动工具栏右击菜单：**

1. 右击浮动工具栏的标题栏。出现下面的菜单：

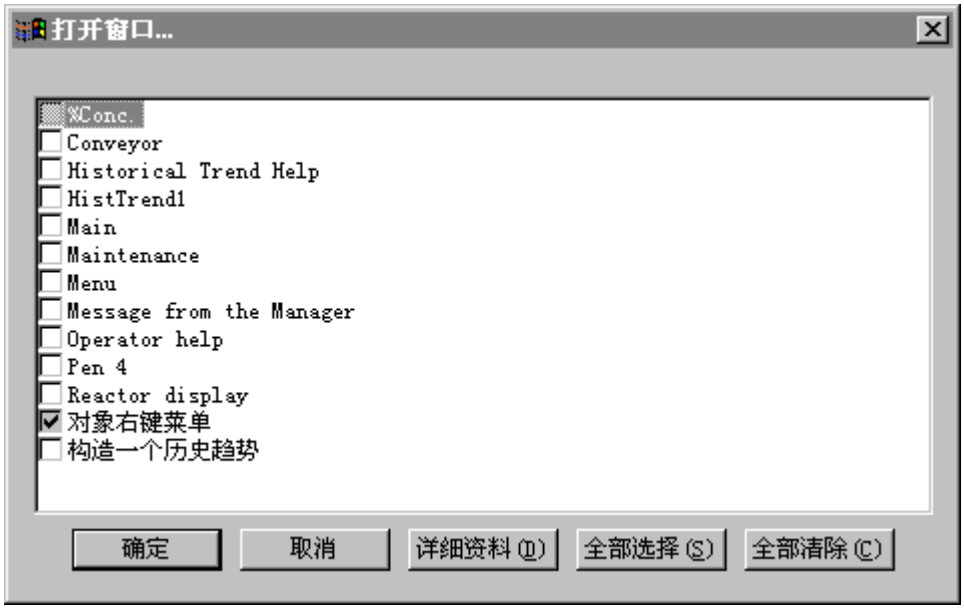


2. 单击菜单上您要应用于工具栏的命令。

通用窗口对话框特性

当您使用 WindowMaker“**文件**”菜单上的命令打开、保存、关闭、删除或复制窗口时，所显示的对话框都很相似，它们有许多共同的特性。为避免重复说明如何执行这些操作，本章节向您介绍这些对话框的公共特性。

当您右击一个打开窗口的空白区域，或者单击“**文件**”菜单上的打开、保存、关闭、删除或另存为窗口命令时，缺省条件下，所选命令的相应对话框将出现在“列表视图”中。这意味着适用于选定命令的所有窗口名将出现在一个连续列表中。例如：



- 当窗口名数目超过缺省的列表空间时，会出现一个水平滚动条。

  单击“**详细资料**”以将“列表视图”改为详细视图。

当您选择详细视图时，屏幕将以多列格式显示窗口及其详细资料。显示的详细资料包括您在“**窗口属性**”对话框中输入的任何窗口注释、窗口类型，以及窗口最后一次修改的日期和时间。例如：



在详细视图中，您可以选择任何一个没有打开的窗口，方法单击其行上的任何部分，而不只是复选框（整行将辉亮显示。再次单击选定的窗口，可以取消选定）。当窗口名数目超过缺省列表空间时，会出现一个垂直滚动条。

要按详细资料类型对列表排序，可以单击该详细资料的标题栏。详细视图的排序顺序如下所示：

列	排序顺序
名称	字母
注释	字母
类型	覆盖、替换、然后弹出
上一次修改	最早的日期/时间（顶端）到最近的日期/时间（底端）

- 每次单击列标题时，列表的排列顺序将从升序变为降序。例如，如果列表当前以升序排列，当您单击某个列标题时，列表将按降序对选定的列重新排序。

要使列表返回缺省显示，单击列标题最左端的小方框。

要改变列大小，将光标置于每个详细资料列标题的垂直分隔线上。当光标变成 "I" 形时，单击并拖曳该标题到您需要的宽度。

要迅速地自动调节列大小，可双击该列的右垂直分隔线。

要打开选定的窗口，单击“**确定**”。

要取消选择并关闭对话框，单击“**取消**”。

要使对话框返回“列表视图”，单击“列表”。

要选择所有列出的窗口，单击“**全部选定**”。

要清除所有选定的窗口，单击“**全部清除**”。

其它鼠标快捷方式

双击任何对象或符号，可以自动执行所选对象或符号的“**动画链接**”命令（位于“**特殊**”菜单上）。

有关动画链接的详细信息，请参阅第 5 章“创建动画链接”。

双击“链接定义”对话框内的空白表达式输入字段，可以启动标记浏览器，列出应用程序标记名字典中定义的所有标记名。

在“链接定义”对话框的表达式输入域中的圆点（.）之后双击，将显示“**选择域名**”对话框，其中包含所有标记名**点域**的全局列表。

双击动画链接标记名或表达式中的标记名，将打开标记名字典中的该标记名定义。

双击动画标记名或表达式中的 SuperTag 父模板名，将打开此 SuperTags 成员标记名的标记名字典详细对话框。

关于标记浏览器和标记名点域的详细信息，请参阅第 4 章“标记名字典”。

右击一个打开窗口的空白区域，任意 WindowMaker 对话框中的文本框或图形对象上将显示一个可应用于窗口、文本或对象的命令菜单。右击浮动工具栏上的标题栏，也将显示一个您可以应用于该工具栏的命令菜单。

快捷方式和加速键

InTouch 提供了许多鼠标和键盘快捷方式来访问一些常用功能。只要菜单命令包含键盘快捷方式，则它会显示在菜单中此命令的右边。此外，所有命令都可以用一个以 ALT 键开始的三键序列加以执行。第二个键是菜单名中带下划线的字符，第三个键是命令中带下划线的字符。

例如，您可以用序列 “alt + fn” 来执行 “文件” 菜单上的 “新建窗口” 命令。

要执行右击菜单上的命令，您可以输入命令中带下划线的字符。

要执行子菜单上的命令，您必须依次按下三个键。例如，要执行 “对齐” 子菜单上的 “中心对齐” 命令，您需要按 alt + aac 或 ctrl+F5。



注意：当您选择一个后跟 **(...)** 的菜单命令时，会出现一对话框，您必须选择和输入更多的信息，才能完成此命令。

用箭头键移动对象

在 WindowMaker 中，您可以使用上、下、左、右箭头键，使选定的对象或对象组朝所按箭头键的方向一次移动一个象素。当需要精确地进行对齐和位置调整时，此特性将很有用处。要一次移动 10 个像素，可在按住 SHIFT 键的同时按下箭头键。要一次移动 50 个像素，可在按住 CTRL 键的同时按下箭头键。

在 WindowViewer 中，当您使用箭头键时，系统使用一定的算法从一个触控对象移动到另一个对象。例如，左光标箭头键向左移动的宽度与所选对象的高度相等。如果它在运动路线内与另一触控对象相交，后者将取代当前选定对象的地位。如果没有遇到任何其它触控对象，运动路线将在屏幕左边界绕回，并继续试图与另一对象相交。如果没有遇到其它对象，则原始对象将保持选定状态。

所有箭头键的工作方式均相同。向上和向下箭头键使用选定对象的宽度作为其运动路径的宽度。

- 如果您的应用程序用户只使用光标键在您的应用程序窗口中导航，则必须认真地设计窗口中的触控对象布局，以确保其运动路径相交。

在 WindowViewer 中，也可以使用 TAB 键在窗口中的触控对象之间转换（但是并不保证跳格的顺序）。

只要在设计应用程序时小心，加上理解箭头键的作用方式，您就能够创建一个不用鼠标的应用程序。

使用 WindowMaker 帮助

WindowMaker 支持上下文相关帮助，如下所示：

要获得已打开对话框的帮助信息，请按 F1 键。相应的 Windows 联机帮助主题将出现。

要获得各种 QuickScript 函数的帮助，请在 QuickScript 编辑器中单击“**帮助**”，或者在“**插入**”菜单上指向“**函数**”，然后单击“**帮助**”。一个对话框会出现，列出所有可用的 QuickScript 函数。单击您要获得帮助的函数。函数的相应 Windows 联机帮助主题将出现。

要获得有关 InTouch 软件的信息，例如您使用的版本号、序列号、许可证截止日期等，可以在 WindowMaker 的“**帮助**”菜单上，单击“**关于**”。“**关于 InTouch**”对话框将出现：



单击“**查看许可证**”启动许可证查看实用程序，获得有关您的 FactorySuite 许可证的信息。

有关许可证查看实用程序的详细信息，请参阅您的*《FactorySuite 系统管理员指南》*。

第 2 章

使用 WindowMaker

通过设置 WindowMaker 和 WindowViewer 的各种属性，您可以自定义应用程序的功能和最终外观。例如，您可以指定 WindowViewer 中的可用菜单，也可以在应用程序的标题栏中显示公司名称等等。

本章向您介绍如何配置 WindowMaker 和 WindowViewer、使用 WindowMaker 窗口、编辑和排列图形对象，以及如何安装和使用向导及 ActiveX 控件。

目录

- 简单对象
- 复杂对象
- 自定义您的开发环境
- 使用 WindowMaker 窗口
- 使用图形对象
- 排列窗口对象
- 使用图像和位图
- 使用文本对象
- 使用线条和轮廓
- 使用向导
- InTouch 窗口控件向导
- 使用 ActiveX 控件
- 配置 ActiveX 控件
- 自定义运行时环境
- 将 WindowViewer 作为 NT 服务运行
- 使用 InTouch 安全性

简单对象

WindowMaker 有四种基本类型的简单对象：线条，填充图案、文本和按钮。每种简单对象类型都包含影响其外观的属性。这些属性包括线条颜色、填充色、高度、宽度和方向等，并且可以是静态，也可以是动态。静态属性在应用程序运行期间保持不变。动态属性与表达式值相链接，因此表达式值的变化会导致属性的变化。例如，一个对象的填充色可以与一个离散表达式的值相链接。根据表达式的状态，当表达式为真时，填充色显示为一种颜色，当表达式为假时，填充色显示另一种颜色。简单对象的大多数属性都可以设为动态。一个对象可以具有一个以上的动态属性。动态属性可以任意组合，以达到所需的结果。下面简要介绍 WindowMaker 的简单对象类型：

对象	描述
线条	取决于线条类型，一个线条对象可以由一个或多个线段组成。颜色是线条可以设置动画效果的唯一属性。宽度和样式不能设置动画，而只能简单地指定其缺省属性。
填充图案	填充图案是由线条包围的闭合内部区域组成的二维对象。填充图案的例子有矩形、圆角矩形、圆、椭圆和多边形等。填充图案的属性包括：线颜色、线宽、线样式、填充色、颜色填充百分比、高度、宽度、位置、可见性、方向和大小。
文本	文本是由字符串或单行组成的对象。文本对象的属性包括：字体、字型、颜色、**粗体**、下划线、*斜体*、对齐方式、可见性和位置等。
按钮	使用 WindowMaker“绘图对象工具栏”上的按钮工具，可以创建任意大小的三维按钮。按钮表面的缺省“文本”字符串可以用“特别”菜单上的“替换字符串”命令进行编辑。 按钮可以附加许多链接类型，例如动作脚本、键脚本、模拟或离散值输入或输出链接。如果按钮附加了输入或输出链接，则按钮上会显示文本字符串值。

复杂对象

除了简单对象外，InTouch 也支持存在相当差异的各种复杂对象。下面简要介绍 WindowMaker 的复杂对象类型：

对象	描述
位图	位图工具用于复制和粘贴位图到您的应用程序中。一旦被粘贴到应用程序窗口，位图就可以旋转，也可以定义为透明背景，以漂浮在其它对象之上。
趋势	趋势工具有二种：一种用于创建显示实时数据的趋势，另一种用于创建显示历史数据的趋势（可从历史记录文件检索）。实时和历史趋势二者都可以进行设置，以显示特定时间内多个标记名的图形表象。
符号	符号是简单对象（线条、填充图案和文本）的组合，符号被视为一个单一对象。应用于符号的任何属性改变，无论是 WindowMaker 中的静态属性改变，还是 WindowViewer 中的动态属性改变，都会影响符号的所有组件对象。 例如，如果您创建的泵符号由两个圆和两个矩形组成，您随后为符号定义了一个填充色链接，则填充色将应用于所有四个对象。同样，在 WindowMaker 中，选择填充色也会改变所有组件对象的填充色。 对于相同的属性，符号的组件对象可以具有不同的值，条件是在将对象合并成符号之前，这些属性不同，并且在变成符号之后这些属性没有改变。符号不能包含位图、按钮、单元、报警或趋势。

对象	描述
单元	单元是两个或更多对象、符号或其它单元的集合，它们相互结合形成一个单一的整体。单元在各自的图形元素之间保持固定的空间关系。一个单元的每个组件都有其自身的链接。单元用于创建虚拟设备，如滑杆控制器。 要创建一个单元，可以通过选定两个或更多的对象、符号和（或）单元，然后执行“**排列**”菜单上的“**制作单元**”命令。 一旦一组对象被设置成单元，其内部细节如颜色、大小和动画链接等，将不能再改变。改变单元外观或方向的唯一方法是使用“**排列**”菜单上的“**分解单元**”命令将其“分开”。 在 WindowViewer 中，通过链接操作可改变单元组件的属性。您可以对单元执行复制、拷贝、粘贴、对齐和更改间距等操作，但单元不能改变大小，而必须先分解，在改变大小后再重新设置成单元。在创建与不同标记名关联的多个相似设备时，单元非常有用。
向导	向导可以在应用程序开发阶段为您节省大量的时间。它们易于使用和配置。如果您在选择向导并将其粘贴到窗口后双击它，其配置对话框将出现，您随后就可以开始配置。 例如，对于“游标”向导，配置项目包括激活标记名、游标的最小和最大范围标签以及填充色等。一旦输入所需的配置信息，就可以开始使用向导。通过使用向导，您不必再花时间绘制对象的各个组件，输入对象的值域，或设置对象动画。 您可以开发“复杂”向导来提供“后台”类型的操作。这些操作包括创建完整的显示窗口、创建或转换数据库、导入 AutoCad 制图和配置其它应用程序，如 InTouch 配方管理器和 SPC 附加程序等。 **注意**：熟练的 "C" 程序员可以使用 Wonderware 的扩展工具套件来开发自定义向导。此工具套件可从分销商处获得。 有关向导的详细信息，请参阅“使用向导”。

对象	描述
ActiveX 控件	WindowMaker 支持 ActiveX 控件，最简单形式的控件是与您的应用程序对话或运行于您的应用程序内的微型应用程序。WindowMaker 支持 Wonderware FactorySuite 组件中包括的所有 ActiveX 控件，例如所有的 InTrack ActiveX 控件。WindowMaker 也支持如 Office97 中安装的第三方 ActiveX 控件。 安装 InTouch ActiveX 控件的方法与安装任何其它向导一样。如果需要，您可以添加常用的 ActiveX 控件到 WindowMaker 的“**向导/ActiveX 工具栏**”中。 如果您在选定 ActiveX 控件并将其粘贴到窗口后双击它，其配置对话框将出现。在配置 ActiveX 控件时，您可以为其指定一个唯一的控件名以便可以在脚本中引用它（当您最初添加控件时会建立一个缺省名称）。 所有的 ActiveX 控件都具有关联的属性、方法和事件。您可以将 ActiveX 属性与相应数据类型的标记名相关联。您可以通过 InTouch QuickScript 函数执行 ActiveX 方法。您可以将 ActiveX 事件与事件发生时所执行的 ActiveX 事件脚本相关联。换句话说，您可以使用 InTouch QuickScript 函数来处理控件事件、调用控件方法和控件属性。 在运行时，您在 WindowMaker 中定义的标记名和 QuickScript 将控制 ActiveX 控件的行为。

自定义您的开发环境

您可以对自定义 WindowMaker 的许多属性进行设置。例如，您可以自定义应用程序的标题栏文本，使之包含公司名称。您可以设置网格的象素间距等。

➢ **要设置 WindowMaker 的属性：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**WindowMaker**”，或者在应用程序浏览器的“**配置**”下双击“**WindowMaker**”。
“**WindowMaker 属性**”对话框出现并显示“**通用**”属性页：

   您也可以在应用程序浏览器的“**配置**”下，右击“**WindowMaker**”，然后单击“**打开**”。



   如果您右击任意对话框中的文本输入框，将出现一个菜单，显示可应用于选定文本的命令。

2. 在“**标题栏文本**”框中，输入运行时应用程序的标题栏中出现的标题。例如：

   ABC Company, Paint APP1

   注意：如果您使用“推广版许可证”，则不能更改标题栏。

   有关 FactorySuite 授权的详细信息，请参阅您的*《FactorySuite 系统管理员指南》*。

3. 如果您要在标题栏中包括应用程序的目录路径，请选择“**显示应用程序目录**”。例如：

   ABC Company, Paint APP1 - C:\DEMOAPP1

4. 在“**垂直间距**”框中，输入对齐网格坐标间隔的象素数。

5. 如果您要在打开 WindowMaker 的“对齐网格”功能时，在窗口中显示网格，请选择“**显示网格**”。

   如果您不选择“**显示网格**”，那么当您打开对齐网格时将不能看见网格。

    单击“**视图**”工具栏上的“对齐网格”工具，或者在“**排列**”菜单上，单击“**对齐网格**”，打开或关闭对齐网格功能。

   对象会按左上角的网格对齐。如果您选定多个对象，则会按组中第一个选定对象的左上角来对齐对象。

6. 如果您希望在启动 WindowMaker 时，自动启动 FactorySuite Wonderware Logger 应用程序，请选择“**启动 Wonderware 日志**”。

   注意：在 Windows NT 操作系统上，Wonderware Logger 的操作稍有不同。

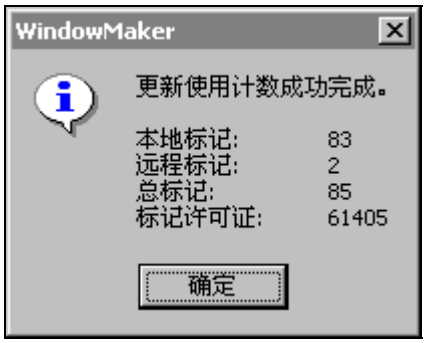
   有关 Wonderware Logger 的详细信息，请参阅您的联机*《FactorySuite 系统管理员指南》*。

7. 如果您要在 WindowMaker 菜单栏上显示标记名字典中已定义的标记名数，可以选择“**显示标记计数**”。

   - 如果您使用有限大小的标记名字典来创建应用程序，此功能将非常有用。

注意：显示标记名计数会严重影响到 WindowMaker 标记名字典的性能。

显示的标记名计数不包括远程标记名引用。要确定远程标记名引用的使用情况，在“**特别**”菜单上，执行“**更新使用计数**”命令。一旦系统完成更新使用计数，将出现下面的对话框：



有关更新使用计数的详细信息，请参阅第 4 章“标记名字典”。

8. 如果您希望在启动 WindowViewer 时自动关闭 WindowMaker，可以选择“**转换到 WindowViewer 时关闭**”。

注意：如果内存足够并且您经常在 WindowViewer 与 WindowMaker 之间切换，请不要选择此选项。

   - 当您选择此选项时，将同时选定位于“**WindowViewer 属性**”对话框的“**通用**”属性页上的“**关闭 WindowViewer**”选项。

9. 如果您要在 WindowMaker 和 Scrapbook+ 之间复制和粘贴对象，请选定“**启用剪贴簿菜单项**”。

注意：要使用这些菜单命令，您的计算机必须安装 Windows Scrapbook+ 程序。此选项供旧 InTouch 系统使用。FactorySuite 及由 Wonderware 售出的 Scrapbook+ 不支持此选项。

---

10. 如果您希望能够选择“中空”对象后面的对象，请选定“**透过中空对象中选取**”。

    - 如果您选定此选项，然后画四条线并将其连成一个包围另一对象的框架，您可以选定框架内的对象而不必将框架置后。

11. 如果您希望在 WindowMaker 与 WindowViewer 的菜单栏上显示两个程序之间的快速切换开关，请选择“**激活快速切换**”。

    - 如果您选定了此选项，则在 WindowMaker 中，快速切换开关为屏幕右上角显示的“**Runtime**”字样；在 WindowViewer 中则显示为“**Development**”字样。要在两个程序之间快速切换，请单击其快速切换开关。

    注意：当您使用快速切换开关时，WindowMaker 会在 WindowViewer 启动之前，自动保存对所有打开窗口所作的修改。

12. 在“**行选择精度**”框中，输入光标可以远离行但仍能选定行的象素数

    - 大多数情况下，使用缺省设置值 4 就足够了。

13. 在“**撤消的级别**”框中，输入您要保存的撤消/恢复层数。您最多可以选定 25 层。如果您输入 0，撤消/恢复功能将被关闭。

    一层对应一个动作。当您创建一个新窗口或打开一个现有窗口时，“撤消”和“恢复”堆栈是空的。当保存窗口时两个堆栈都会被清空。

    有关撤消/恢复的详细信息，请参阅“撤消对象编辑”。

14. 单击“**确定**”。

注意：当您修改上述任意设置后，必须重新启动 WindowMaker，才能使所作的更改生效。

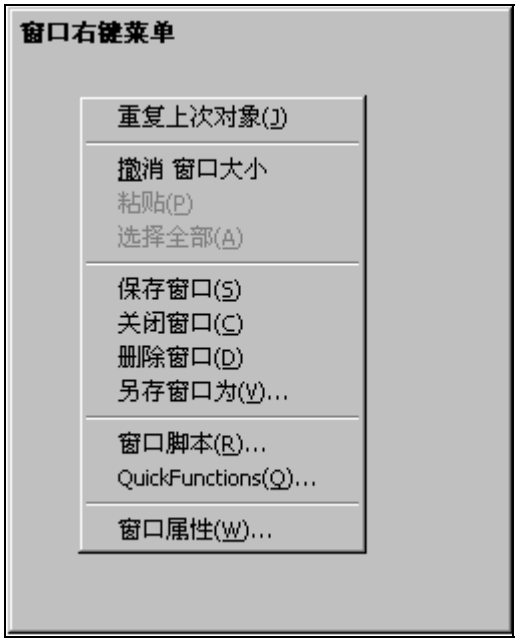
使用 WindowMaker 窗口

您的 InTouch 应用程序更像是用于保存图形和文本对象的一系列窗口。当您在 WindowMaker 中创建新窗口时，程序会要求您定义该窗口的某些属性，例如背景色、标题和屏幕位置等。您也可以创建 QuickScript，根据窗口是打开、显示还是关闭加以执行。

本章节包含您创建、打开、保存、关闭、删除和复制窗口时需要遵循的步骤。

- **通用工具栏**”包含可用于快速应用“**文件**”菜单上的大多数窗口命令的工具。

  要快速访问可应用于窗口的各种命令，右击打开窗口的空白区域，然后单击右击菜单上的相应命令。例如：



创建新窗口

➢  **要创建一个新窗口：**

1. 在“**文件**”菜单上，单击“**新建窗口**”，或者单击“**通用工具栏**”上的“**新建窗口**”工具。“**窗口属性**”对话框出现：

   ☞ 要快速创建一个新窗口，在应用程序浏览器中右击“**Windows**”，然后单击“**新建**”。



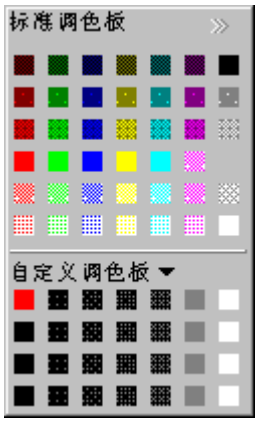
注意：缺省条件下，此对话框中的设置会反映先前创建的窗口的属性，或者如果您在 WindowMaker 中打开窗口时选定了此命令，则此设置会反映活动窗口的属性。如果活动窗口带有窗口脚本，则会出现一个消息框，询问您是否要将窗口脚本复制到新窗口。

   ☞ 如果您右击任意对话框中的文本输入框，将出现一个菜单，显示可应用于选定文本的命令。

2. 在“**名称**”框中，输入您要在新窗口的标题栏中显示的名称。名称最长可达 32 个字符，它可以包含嵌套空格、标点符号和键盘上的任何其它字符，但引号 (") 除外。

3. 在“**注释**”框中，输入您要与窗口关联的任何其它注释（可选）。此信息仅出于文档化目的，并不用于应用程序中。

4. 单击“**窗口颜色**”框，选定窗口的背景色。调色板出现：



如果不需要更改，单击当前颜色选项或者按 ESC 键关闭调色板。

有关使用颜色调色板的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

5. 单击您要用作窗口背景的颜色。

6. 选择您要使用的“**窗口类型**”。窗口类型有三种：

替换 当此类型窗口出现在屏幕上时，它会自动关闭与其相交的任意窗口，包括弹出类型和其它替换类型的窗口。

覆盖 出现在当前显示窗口的上方，可以比其所覆盖的窗口更大。当覆盖窗口关闭后，所有被之覆盖的窗口将重新显示。单击被覆盖窗口的任意可见部分，可以将该窗口置前并激活。

弹出 类似于覆盖窗口，其区别在于弹出式窗口总是位于所有其它打开窗口的上方（即使单击另一窗口）。弹出式窗口通常需要用户做出回应才能移除。

您可以通过再次打开其“**窗口属性**”对话框来改变窗口类型。方法有三种：

打开窗口，然后在“**Windows**”菜单上，单击“**窗口属性**”。

在应用程序浏览器的“**Windows**”下，右击窗口名，然后单击“**属性**”（如果执行命令时窗口没有打开，它会自动在对话框后面打开）。

打开窗口，右击窗口的空白区域，然后单击“**窗口属性**”。

7. 选择窗口的“**框式样**”。式样有三种：

   单一 带标题栏和“**大小控制**”的三维边界窗口。

   加倍 不带标题栏和“**大小控制**”的三维边界窗口，不能调整大小。

   无 没有边界、不带“**大小控制**”的窗口，不能调整大小（如果带“**大小控制**”，将变成一个可调整大小的三维边界窗口）。

8. 如果您希望窗口带标题栏，请选择“**标题栏**”。通过在标题栏上单击并拖动鼠标，标题栏也可以用于移动窗口。

注意：如果窗口带有标题栏，您将不能为“**框式样**”选定“**加倍**”或“**无**”。

---

9. 如果您希望用户能够在 WindowViewer 中改变窗口大小，请选择“**大小控制**”。

10. 在“**维数**”组中，输入每个窗口坐标的象素位置：

    X 轴定位 WindowMaker 设计区域的左边缘与所定义的窗口左边缘之间的象素数目。

    Y 轴定位 WindowMaker 设计区域的顶边缘与所定义的窗口顶边缘之间的象素数目。

    窗口宽度 窗口宽度的象素值。

    Windows 根据您的显示器来限制窗口的最小宽度值。例如：对于标准的 VGA 显示器，最小值为 102 个象素。

    窗口高度 窗口高度的象素值。

    Windows 根据您的显示器来限制窗口的最小高度值。例如：对于标准的 VGA 显示器，最小值为 26 个象素。

注意：缺省条件下，这些输入框的值设置为先前创建的窗口的尺寸。如果在 WindowMaker 中通过窗口边线手动调整窗口大小，则这些值会被自动修改。

---

11. 单击“**脚本**”访问“**窗口脚本**”编辑器。您可以对窗口应用三种脚本类型：

显示时	在窗口最初显示时执行一次。
在显示期间	在窗口显示期间，按指定的频率连续执行。
隐藏时	在窗口隐藏时执行一次。

注意：如果您对活动窗口应用窗口脚本，然后创建一个新窗口，则活动窗口的脚本可以复制到新窗口。一个消息对话框会出现，询问您是否要复制窗口脚本。

- 如果您后来决定需要对打开的窗口应用脚本，可以右击打开窗口的空白区域，然后单击“**窗口脚本**”。如果窗口没有打开，请在应用程序浏览器中，双击“**窗口**”，显示所有的窗口名。右击窗口名，然后单击“**窗口脚本**”。

有关创建窗口脚本的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

创建一个隐藏标题栏和菜单栏的窗口

WindowMaker 的设计区域是标题和菜单栏下方窗口框架内的整个区域。在 WindowViewer 中，设计区域变成显示区域。指定的 X=0 和 Y=0 位置始终为标题栏和菜单栏下方的左上角。标题栏和菜单栏均为 19 象素高并且位于设计区域上方。例如，如果 WindowMaker 被最大化，而您使用的是 1024x768 视频显示，则可见的设计区域为 1024x730（768 象素减去标题栏和菜单栏 各 19 象素，得出可见设计区域象素值 730）。如果 WindowViewer 配置为显示标题栏和菜单栏，则 WindowViewer 的显示区域与 WindowMaker 完全一样，包括标题和菜单栏下方的整个屏幕。

为充分利用被标题和菜单栏占用的空间，您可以设计一个隐藏标题栏和菜单栏的应用程序。当标题栏和菜单栏隐藏时，窗口的左上角将对应屏幕上的一个新位置。这就增加了可见显示区域，为您提供了更多的显示空间。如果您按照这种方式配置 WindowViewer，则所有窗口会自动上移，窗口底部会出现一个缺口。要填充此缺口，您需要将窗口 Y 坐标设置为负值来增加窗口高度。此操作的结果是：在 WindowMaker 中，窗口位于标题栏和菜单栏的下方，而在 WindowViewer 中，窗口位于标题栏和菜单栏的上方。

您可以利用此项技术，在 WindowViewer 中用一个弹出式窗口来覆盖标题栏和菜单栏。当您需要看见标题和菜单栏时，您也可以随时创建一个触动链接按钮或 QuickScript 来隐藏此弹出式窗口。此外，通过应用安全口令，您可以限制特定用户隐藏窗口和访问菜单，包括退出 WindowViewer 的能力。

注意：推广版的 InTouch 许可证不允许隐藏标题栏。因此，如果应用程序是在推广版的 InTouch 许可证下开发的，并且 WindowViewer 的配置是在标准运行时许可证下查看应用程序时选定“**隐藏标题栏**”，则标题栏将根据配置而被隐藏，并且应用程序中的所有窗口将向上移动。

打开窗口

在开发应用程序期间，只要您的计算机内存足够，您可以打开任意多个窗口。

➢  **要打开窗口：**

1. 在“**文件**”菜单上，单击“**打开窗口**”，或者单击“**通用工具栏**”上的“打开窗口”工具。“**打开窗口**”对话框出现，列出应用程序中的所有窗口名。

   - 要快速打开一个窗口，在应用程序浏览器中，双击“**窗口**”打开应用程序中所有窗口名的列表，然后双击所需窗口名。您也可以右击窗口名，然后单击“**打开**”。

2. 单击您要打开的窗口名旁边的复选框。

   - 缺省条件下，所有当前打开的窗口均已选定。

3. 单击“**确定**”关闭对话框并打开选定的窗口。

保存窗口

一旦创建窗口，您需要在关闭窗口或退出应用程序之前保存窗口。所有与窗口关联的图形、QuickScript 和属性等均将被保存。

➢  **要保存窗口：**

1. 在“**文件**”菜单上，单击“**保存窗口**”，或者单击“**通用工具栏**”上的“保存窗口”工具。“**保存窗口**”对话框出现，列出所有需要保存的窗口名。

   - 要快速保存一个窗口，在应用程序浏览器中，右击窗口名，然后单击“**保存**”。您也可以右击窗口的任意空白区域，然后单击“**保存窗口**”。

      要快速保存所有当前打开的窗口，单击“**通用工具栏**”上的“**保存所有窗口**”工具，或者单击“**文件**”菜单上的“**全部保存**”命令。

2. 单击您要保存的窗口名旁边的复选框。

3. 单击“**确定**”关闭对话框并保存选定的窗口。

关闭窗口

如果您试图关闭一个自上次保存以来修改过的窗口，程序会在 WindowMaker 关闭窗口之前，提示您保存所做的修改。

➢  **要关闭窗口：**

1. 在“**文件**”菜单上，单击“**关闭窗口**”，或者单击“**通用工具栏**”上的“关闭窗口”工具。“**关闭窗口**”对话框出现，列出所有当前打开的窗口名。

   要快速关闭一个窗口，在应用程序浏览器中，右击窗口名，然后单击“**关闭**”。您也可以右击窗口的任意空白区域，然后单击“**关闭窗口**”。

2. 单击您要关闭的窗口名旁边的复选框。
3. 单击“**确定**”关闭对话框并关闭选定的窗口。

删除窗口

➢ **要删除窗口：**

1. 在“**文件**”菜单上，单击“**删除窗口**”。“**删除窗口**”对话框出现，列出所有当前打开的窗口名。

   要快速删除一个窗口，在应用程序浏览器中，右击窗口名，然后单击“**删除**”。您也可以右击窗口的任意空白区域，然后单击“**删除窗口**”。

2. 单击您要删除的窗口名旁边的复选框。
3. 单击“**确定**”关闭对话框并删除选定的窗口。

注意：除非您已经备份，否则不能恢复被删除的窗口。程序会提示您确认删除选定的每个窗口名。

复制窗口

如果您想要复制一个现有的窗口，则需复制的窗口必须打开。

➢ **要复制窗口：**

1. 在“**文件**”菜单上，单击“**窗口另存为**”。“**在新名下要保存的窗口**”对话框出现，列出所有当前打开的窗口名。

   要快速复制窗口，在应用程序浏览器中，右击窗口名，然后单击“**另存为**”。您也可以右击窗口的任意空白区域，然后单击“**窗口另存为**”。

2. 单击您要复制的窗口名旁边的复选框（只能选定一个窗口名）。“**保存窗口**”对话框出现：



3. 在“**新名**”框中，为新窗口输入一个有效的名称。
4. 单击“**确定**”关闭对话框并创建复制窗口。

导出窗口

当需要创建或维护一个库应用程序或需要在另一应用程序中快速创建远程标记名引用时，导出窗口将非常有用。要将窗口从一个 InTouch 应用程序移动到另一应用程序，您必须使用“**文件**”菜单上的“**导出窗口**”命令。

有关远程标记名引用的详细信息，请参阅第 4 章“标记名字典”。

警告：如果您试图用任何其它复制方法复制 InTouch 窗口文件，例如使用文件管理器或者 Windows 资源管理器的复制命令，您可能会损坏应用程序的标记名字典！

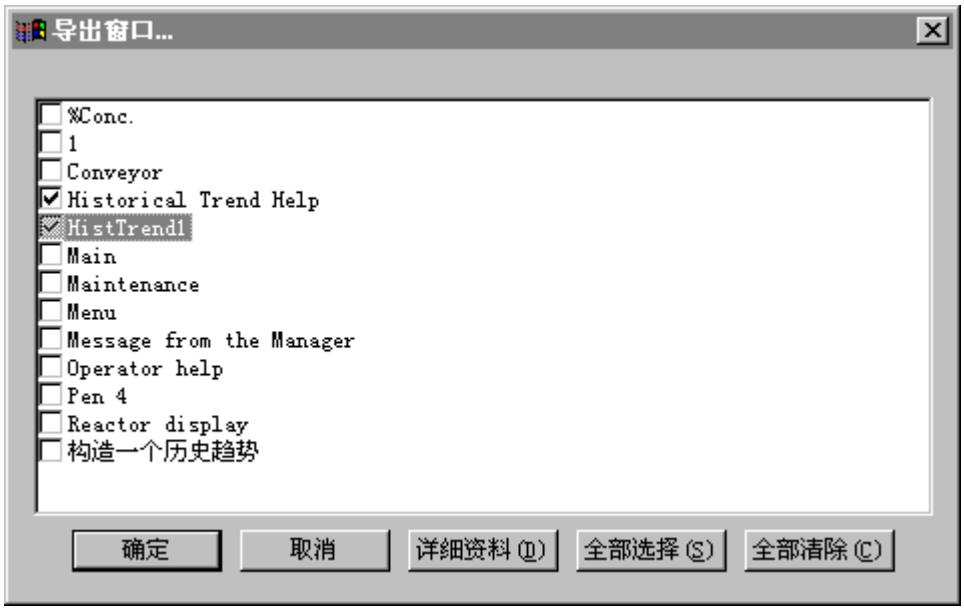
➢ **要导出窗口：**

1. 关闭当前应用程序中的所有窗口。
2. 在“**文件**”菜单上，单击“**导出窗口**”。“**导出到目录**”对话框出现：



3. 找到并选择窗口导出到的应用程序目录（文件夹），然后单击“**确定**”。

4. “**导出窗口**”对话框出现：



5. 选择您要导出的窗口。

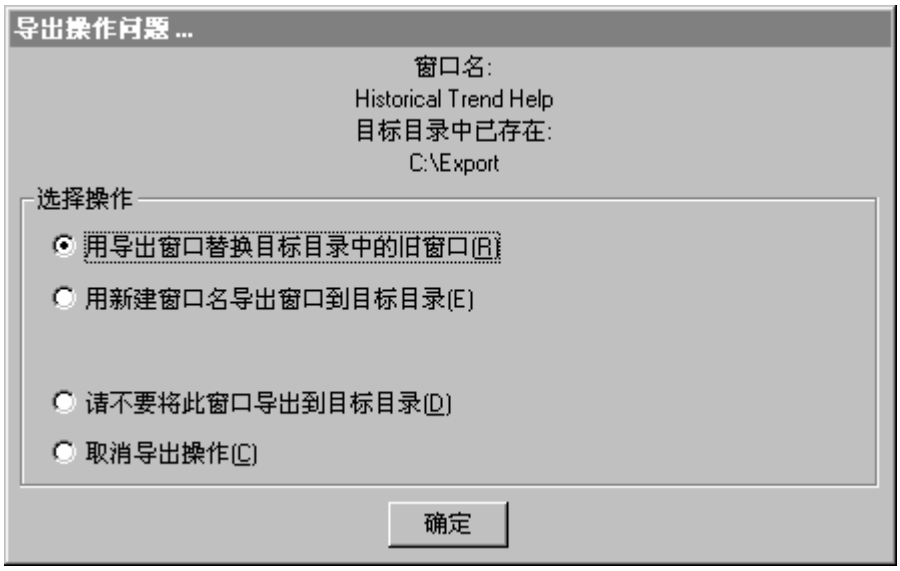
6. 单击“**确定**”。导出操作开始。

注意：导出窗口时，与该窗口关联的所有对象和动画链接将随窗口一起导出。然而，与窗口中的对象关联的标记名将转换成“占位符”标记名。使用占位符标记名，可以避免当目标应用程序的标记名字典不包含相同标记名时可能出现的任何问题。

有关转换占位符标记名的详细信息，请参阅第 4 章“ 标记名字典”。

导出操作问题

如果在导出窗口时系统遇到问题，屏幕会出现“**导出操作问题**”对话框：



在“**选择操作**”组中，选择您要采取的措施，然后单击“**确定**”。

导入窗口

将窗口从一个 InTouch 应用程序导入当前应用程序，可以使您节省大量开发时间，同时还为您提供一种建立远程标记名引用的简便快捷方法。它允许您重复使用先前创建的窗口、对象和窗口脚本。在将窗口从一个 InTouch 应用程序移动到另一应用程序时，您必须使用“**文件**”菜单上的“**导入**”命令。

有关远程标记名引用的详细信息，请参阅第 4 章“标记名字典”。

注意：如果您试图用任何其它移动方法移动 InTouch 窗口文件，例如使用文件管理器或者 Windows 资源管理器的移动命令，您可能会损坏应用程序的标记名字典！

➢ **要导入窗口或 QuickScript：**

1. 关闭当前应用程序中的所有窗口。
2. 在“**文件**”菜单上，单击“**导入**”。“**浏览文件夹**”对话框出现：



3. 找到并选择包含您要导入的窗口的应用程序目录（文件夹），然后单击“**确定**”。

4. 出现下面的对话框：



5. 选择您要导入的项目，然后单击“**选择**”。一个对话框出现，让您选择要导入的窗口或 QuickScript。

6. 在选定要导入的窗口或 QuickScript 后，单击“**导入**”。系统会自动开始将选定项目导入到当前应用程序。

注意：要导入一个窗口脚本，您必须导入整个窗口。当您导入窗口时，与该窗口关联的所有对象和链接将随窗口一起导入。但是，与窗口中的对象关联的标记名（以及所导入脚本中使用的标记名）将转换成“占位符”标记名。

当导入的脚本或窗口中的标记名转换成占位符标记名时，每个标记名的开头会添加三个索引字符。例如，当导入一个离散型标记名时，该标记名的开头会有三个字符“**?d:**”。当导入一个长度为 30、31 或 32 字符的标记名时，每个标记名开头仍会添加这三个索引字符。但是，增加这三个字符不会截短现有标记名的长度。例如，占位符标记名的 32 字符标记名会增加到 35 字符。这三个增加的空间仅分配用于占位符标记名。标准标记名不支持这种标记名长度的增加。

有关转换占位符标记名的详细信息，请参阅第 4 章“标记名字典”。

注意：当从包含 SuperTag 的应用程序导入窗口时，只有所导入窗口中实际使用的 SuperTag 实例会导入到新应用程序中。整个 SuperTag 模板结构不会被导入。例如，如果应用程序定义了数百个 SuperTag 成员标记名，其中只有 50 个用于导入窗口中，那么只有这 50 个标记名会被导入。

使用图形对象

一旦您在应用程序中创建了新的窗口，就可以使用图形对象来丰富其内容。WindowMaker 提供了许多工具来编辑和排列您绘制和粘贴到应用程序的各种图形对象。本章节向您介绍对图形对象执行各种编辑功能时使用的步骤。

对象	描述
绘图对象工具栏	包含用于创建窗口图形的绘图工具。
视图工具栏	包含标尺工具，用于显示标尺来帮助您对齐窗口中的图形对象。此工具栏还包含用于隐藏或显示应用程序浏览器、工具栏或窗口中的可见网格的工具。它还包含切换全屏模式的工具。
通用工具栏与格式工具栏	通过单击这两栏中的工具，您可以将“**编辑**”和“**文本**”菜单上的许多命令快速应用到图形对象中。
排列工具栏	单击此栏上的工具，您可以快速应用“**排列**”菜单上的对齐命令工具。
向导/ActiveX 工具栏	通过添加反复使用的向导或 ActiveX 控件，您也可以自定义此工具栏。

有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

如果您右击一个对象，会出现一个菜单，显示可用于选定对象的有效命令或动作。例如：



注意：右击菜单上的命令会变化，它们取决于您所选定的对象类型。

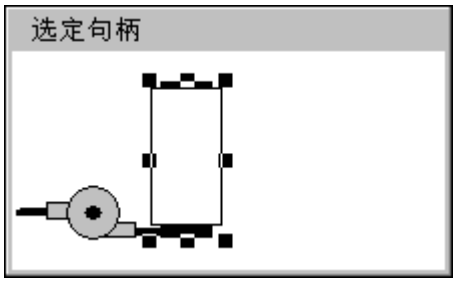
选择和调整对象大小

如果您在绘制对象后单击对象，其周围会出现几个小方框。这些方框称为“句柄”。您可以使用这些句柄来调整对象大小和（或）改变其形状。

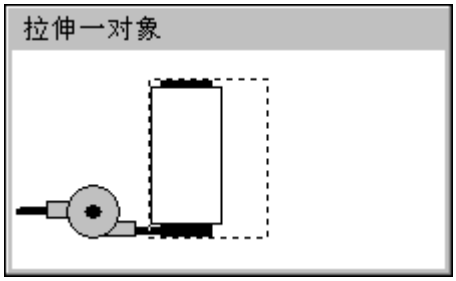
“选定”这个词是 WindowMaker 图形编辑中的一个重要概念。如果对象周围出现句柄，则表示对象已经被“选定”。直接单击对象就可选定对象。单击窗口的空白区域可以取消选定该窗口中的任何当前选定对象。一般说来，您执行的任何命令都可应用于所有选定对象，只要命令对该对象有效。

➢ **要改变对象大小：**

1. 选择对象，然后将箭头光标指针置于“句柄”中央。

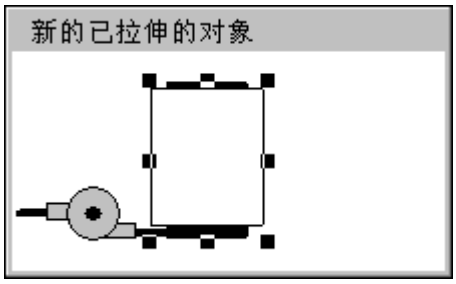


2. 按住鼠标左键不放，同时拖动句柄以缩放对象。



取决于选定的句柄，您可以朝任意方向缩放对象。如果您使用四个角句柄之一来改变对象大小，则对象会同时在垂直和水平方向上改变大小。

3. 释放鼠标按钮，对象即被调整为新的尺寸。



 如果尺寸不正确，请在“**编辑**”菜单上，单击“**撤消**”，或者单击“**通用工具栏**”上的撤消工具，然后重试。

有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

关于撤消和恢复编辑的详细信息，请参阅“撤消对象编辑”。

选择窗口中的所有对象

要选择活动窗口中的所有对象，在“**编辑**”菜单中，单击“**全部选定**”，或按 F2 键，或者右击打开窗口的空白区域，然后单击“**全部选定**”。

选择多个对象

要选择多个对象，在选择第一个对象后，按住 SHIFT 键，单击您要选择的其它对象。要从一组选定对象中删除某个特定对象，按住 SHIFT 键，单击您要撤消选定的对象。

选择对象组

把光标移动到窗口的空白区域。按下鼠标按钮并拖动鼠标，一个带小手光标的虚线选择矩形框出现。拖动鼠标直至完全包围您要选定的所有对象。释放鼠标按钮，所有完全位于矩形框内的对象将被选定。

撤消选定对象组

如果您在按住 SHIFT 键的同时画一个选择矩形框，则将撤消选定所有包围的对象。这个技巧也可以用来启动另一个对象上面的选择矩形框。

如果您在按住 SHIFT 键的同时单击鼠标左键，则在移动光标时将不能拖动光标下的对象，而是画一个选择矩形框。

撤消对象编辑

WindowMaker 可以跟踪您所作的编辑和格式更改。您可以配置 WindowMaker，使之支持最多 25 层撤消/恢复操作。您也可以通过将层数设置为零 (0)，来禁用撤消/恢复功能。缺省条件下，WindowMaker 设置为支持 10 层，其中每一层代表一个动作。

注意：当您创建新窗口或打开现有窗口时，“撤消”和“恢复”堆栈是空的。当您保存窗口时，这两个堆栈也将被清空。

关于配置撤消/恢复层数的详细信息，请参阅“自定义您的开发环境”。

“**撤消**”和“**恢复**”命令位于“**编辑**”菜单中。通过右击对象也可以访问这些命令。这些命令会动态改变，以反映可应用的最后一次动作。例如，如果您移动了某个对象，随后决定恢复其在窗口中的初始位置，则可以右击窗口的空白区域，单击“**撤消移动**”，或者在“**编辑**”菜单上，单击“**撤消移动**”。

 撤消和恢复工具位于“**通用工具栏**”内。

您可以使用撤消和恢复操作来反转应用到对象的动作或命令。您可以撤消和恢复以下动作或命令：

动作/命令	**支持的操作**
基本	创建、选择、撤消选择、移动和调整对象大小、线条、填充图案、文本和窗口颜色、窗口移动和调整窗口大小
编辑	复制、剪切、拷贝、删除、粘贴位图和调整位图 – 原始大小，全部选定、链接剪切、链接复制、链接粘贴、链接清除、放大半径、缩小半径、改变对象形状、添加顶点、删除顶点
排列	置后、置前、对齐（所有命令）、水平间距、垂直间距、顺时针旋转、逆时针旋转、水平翻转、垂直翻转、制作符号、分解符号、制作单元、分解单元
文本	所有操作（大小、样式、间距、对齐）
线条	所有操作（宽度和样式）
特别	动画链接（双击对象）、替换字符串

复制对象

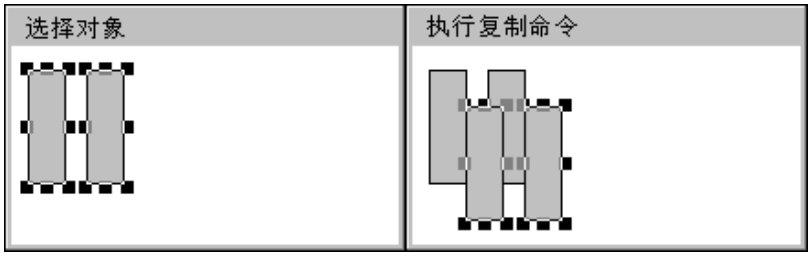
➢ **要重绘以前绘制的对象：**

1. 绘制对象。
2. 右击对象，然后单击“**重复上次对象**”。
3. 单击鼠标左键，然后再次绘制相同的对象。

➢  **要复制对象：**

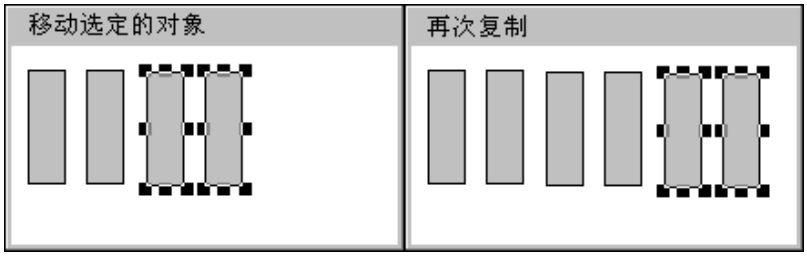
1. 选择您要复制的对象。
2. 在“**编辑**”菜单上，单击“**复制**”，或者单击“**通用工具栏**”上的“复制选项”工具。

   有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。



要快速复制对象，右击对象，然后单击“**复制**”。

如果您在没有撤消选定的情况下移动所复制的对象，则再次复制时，第二次复制（及所有后续复制）会自动偏移第一次复制时移动的距离。例如：

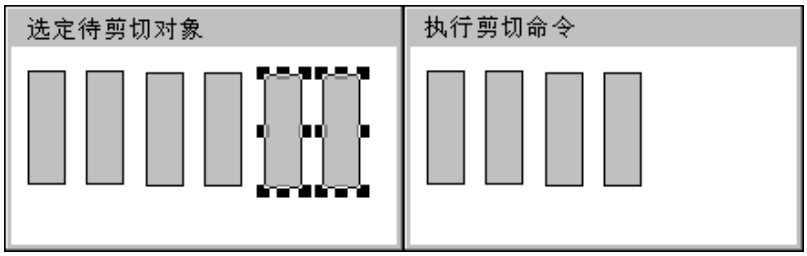


您可以根据需要重复任意多次此过程。

剪切对象到 Windows 剪贴板

➢  **要剪切对象：**

1. 选择您要剪切的对象。
2. 在“**编辑**”菜单上，单击“**剪切**”，或者单击“**通用工具栏**”上的“剪切到剪贴板”工具。



要快速剪切对象，右击对象，然后单击“**剪切**”。

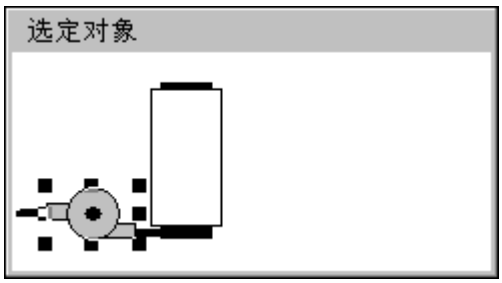
注意：当您剪切对象时，该对象会从窗口被移除并复制到 Windows 剪贴板上。该对象的属性和动画链接也将同时复制。

复制对象到 Windows 剪贴板

➢  **要复制对象：**

1. 选择要复制的对象。
2. 在“**编辑**”菜单上，单击“**复制**”，或者单击“**通用工具栏**”上的“复制到剪贴板”工具。

有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。



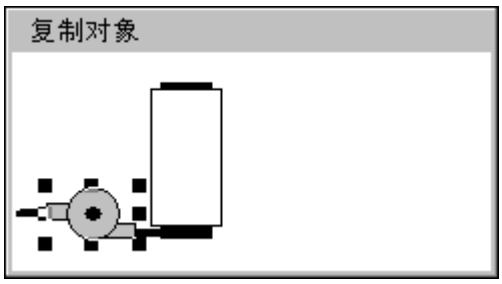
要快速复制对象，右击对象，然后单击“**复制**”。

注意：当您复制对象时，对象并没有从窗口中被移除，而是被复制到 Windows 剪贴板上。对象的属性和动画链接也将同时复制。

从 Windows 剪贴板粘贴对象

➢  **要从 Windows 剪贴板粘贴对象：**

1. 复制或剪切对象：

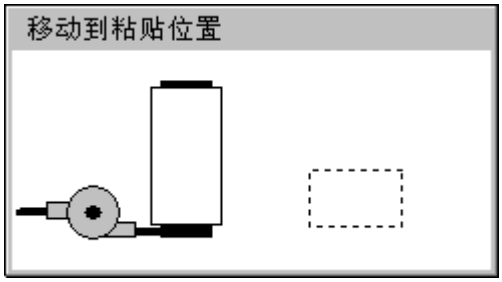


2. 在“**编辑**”菜单上，单击“**粘贴**”，或者单击“**通用工具栏**”上的“从剪贴板粘贴”工具。

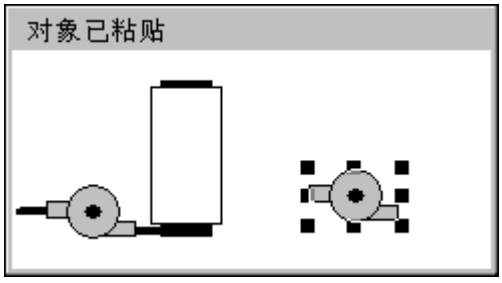
   要快速粘贴所复制的对象，右击对象，然后单击“**粘贴**”。

3. 光标将变成一个弯头符号。

4. 按住鼠标左键，出现一个所复制对象大小的虚线矩形框。将矩形框拖到对象要粘贴到的窗口中的位置。



5. 释放鼠标以粘贴对象：



   所有粘贴的对象在粘贴后仍保持选定状态，您可以通过移动来调整其位置。

注意：当您将对象复制或剪切到 Windows 剪贴板时，所有的对象属性和动画链接也将随之一起复制。如果您随后将该对象粘贴到窗口，其所有属性将保持原样。

剪切和粘贴对象链接

WindowMaker 的链接粘贴缓冲区是一个临时存储区，用于存储您从一个对象剪切或复制的链接（缓冲区只存储您最近的剪切或复制动作的链接）。您可以将链接粘贴缓冲区中存储的链接粘贴到任何对象或符号。如果您选择多个对象，则该链接将粘贴到每个不同的对象上。

如果粘贴的链接对于对象没有表现值，例如文本对象上的线条颜色链接，则不会粘贴该链接。

有关动画链接的详细信息，请参阅第 5 章“创建动画链接”。

➢ **要剪切、复制、粘贴和清除链接：**

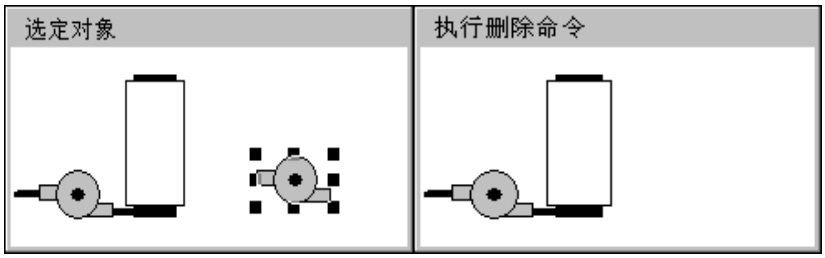
1. 选择链接命令要应用到的对象。
2. 在“**编辑**”菜单上，指向“**链接**”，然后单击相应的命令。

   要快速访问链接命令，右击对象，指向“**链接**”，然后单击相应的链接命令。

删除对象

➢ **要删除对象：**

1. 选择您要删除的对象。
2. 在“**编辑**”菜单上，单击“**清除**”。



要快速删除对象，右击对象，然后单击“**清除**”，或在选定对象后按 DEL 键。

注意：被删除的对象不会复制到 Windows 剪贴板。

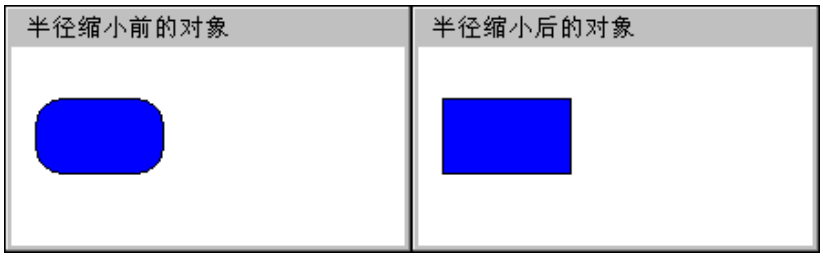
增大或减小圆角对象的半径

您可以增大和（或）减小使用圆角矩形工具绘制的任何对象的角半径。

➢ **要增大（或减小）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**”）。

   要快速增大或减小半径，右击圆角对象，然后单击相应的命令。
3. 重复此命令，直至半径增大到所需大小。例如：

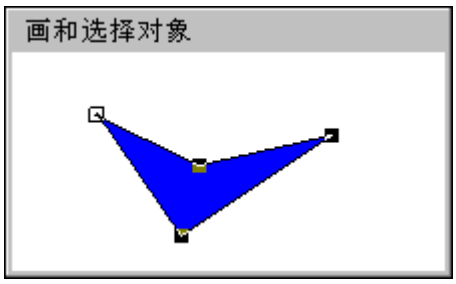


   您也可以使用键盘上的快捷键 Shift+加号（数字小键盘上的 + 号）来增大半径，或 Shift+减号（数字小键盘上的 - 号）来减小半径。每次按下这些键的组合时，将执行相应的命令。如果您按住键不放，命令将持续执行，直至对象半径不能再增减为止。

调整多边线或多边形的形状

➢  **要调整多边线或多边形的形状：**

1. 选择多边线或多边形对象。

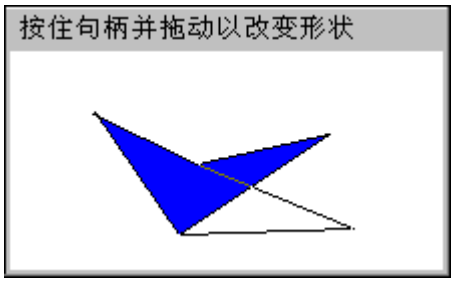


   - 您绘制对象时鼠标单击的每个“点”将重新显示为“句柄”。

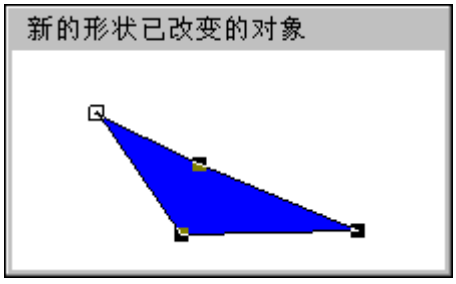
2. 在“**编辑**”菜单上，单击“**改变对象形状态**”，或者单击“**排列工具栏**”上的“改变对象形状”工具。

   - 要快速调整多边线或多边形的形状，右击对象，然后单击相应的命令。

3. 要快速调整对象形状，抓住句柄并将其拖到所需位置。

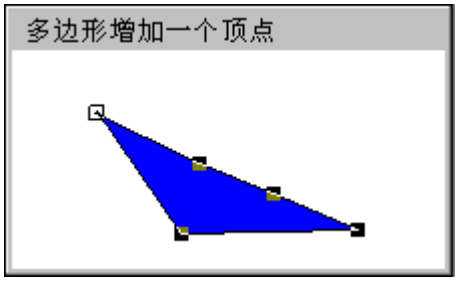


4. 释放鼠标，对象即被调整为新的形状。



➢ **要添加或删除多边线或多边形上的“顶点”：**

1. 选择多边线或多边形对象。
2. 在“**编辑**”菜单上，单击“**添加顶点**”或“**删除顶点**”，然后单击新顶点要添加到的对象位置，或者单击您要删除的顶点。



要在多边线或多边形上快速添加（或删除）顶点，右击对象，然后单击相应的命令。

排列窗口对象

WindowMaker 为您提供了许多用于排列窗口对象的工具。本章节向您介绍 WindowMaker 提供的各种排列工具。

- 用“**排列工具栏**”中包含的工具，您可以对选定的对象快速应用“排列”菜单上的大部分命令。例如：



有关“**排列工具栏**”的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

对齐对象

您可以按对象的左或右边缘、中心、中心点、顶端、中部或底端来对齐对象。

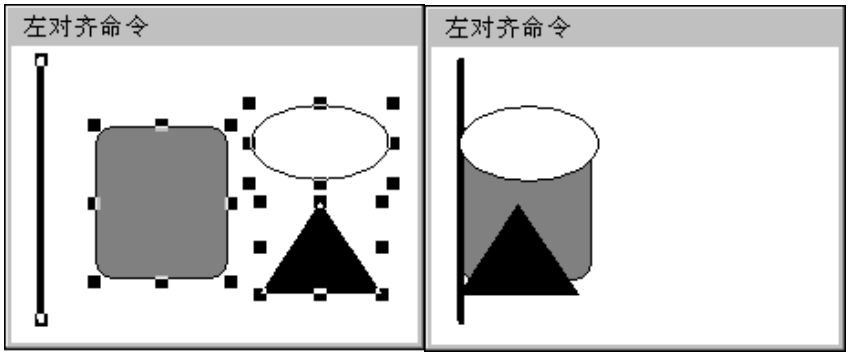
➢ **要对齐所有选定的对象：**

1. 选择对象。
2. 在“**排列**”菜单上，指向“**对齐**”，然后单击相应的对齐命令。选定的对象将根据您的选择对齐。

下面的例子向您讲解每个对齐命令的操作方式：



“**左对齐**”以组中最左端对象的左边缘对齐为基准，对齐所有选定对象的左边缘。

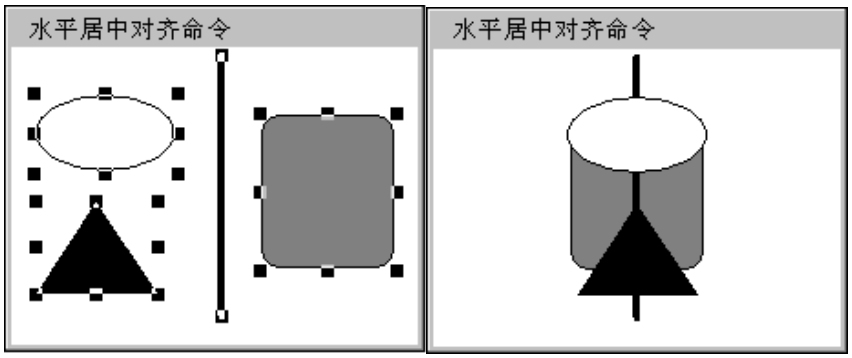


- 要快速对齐对象，选择对象，然后单击“**排列工具栏**”上的相应工具。

有关工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

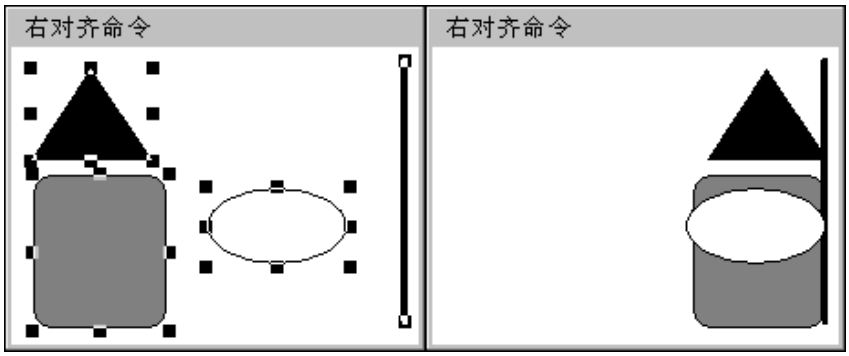


“**中心对齐**”以组中的垂直中心线为基准，对齐所有选定对象：



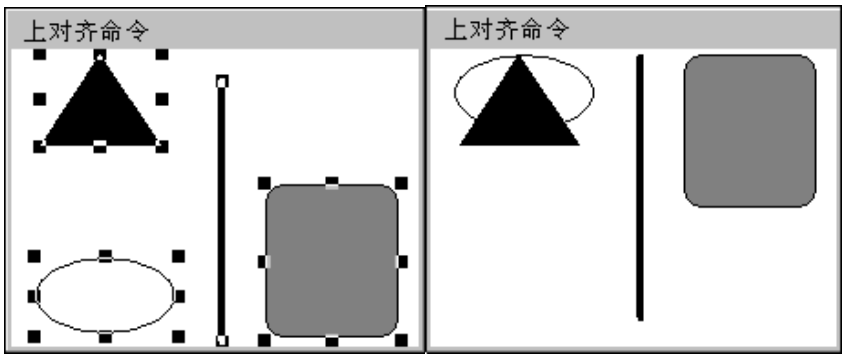


“**右对齐**”以组中最右端对象的右边缘为基准，对齐所有选定对象的右边缘：



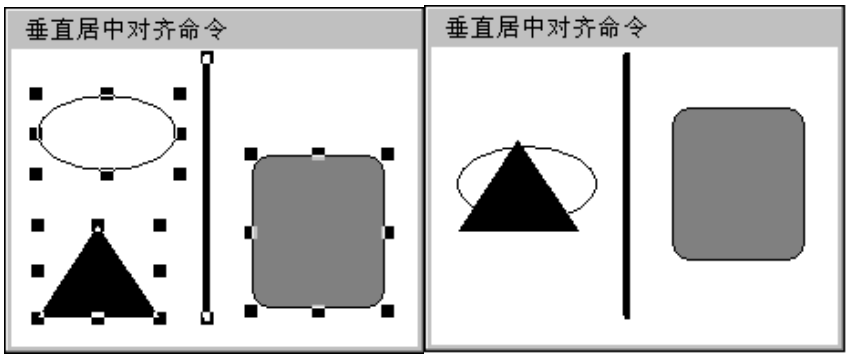


“**顶部对齐**”以组中最顶端对象的顶边缘为基准，对齐所有选定对象的顶边缘：



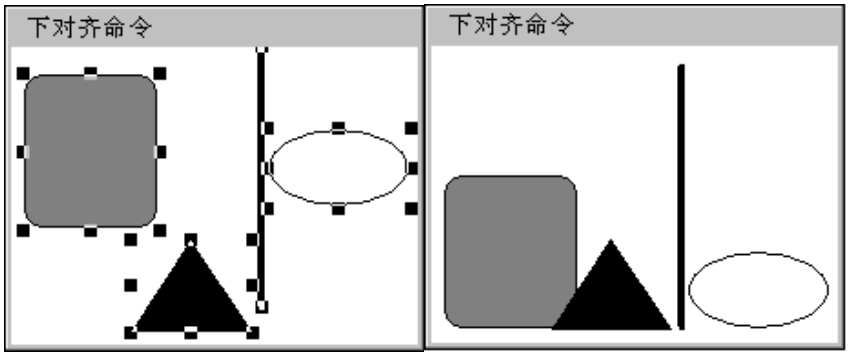


“**垂直居中对齐**”以组的中间位置为基准，对齐所有选定对象的中部：



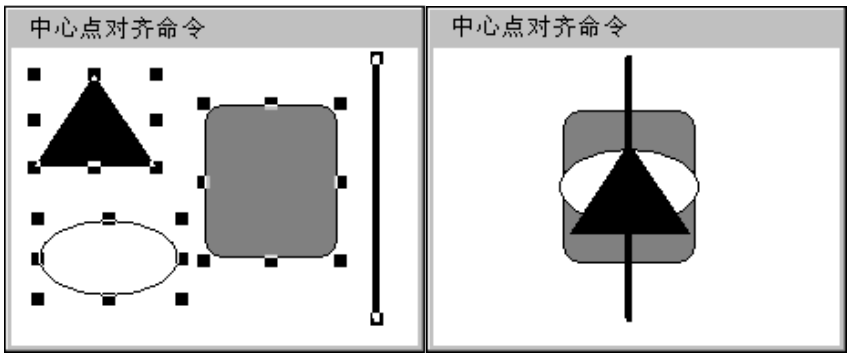


“**底部对齐**”以组中最底端对象的底边缘为基准，对齐所有选定对象的底边缘：





“**中心点对齐**”以组的中心点为基准，对齐所有选定对象的中心点：

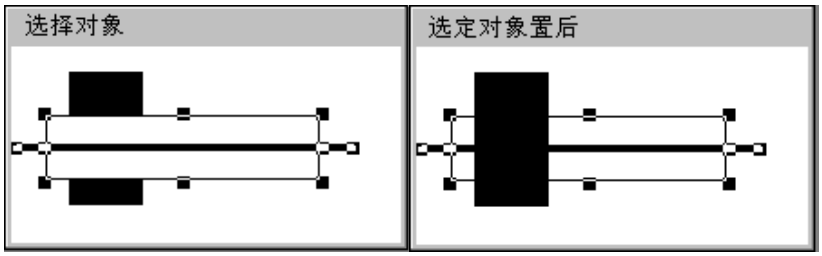


对象分层

您可以通过将对象置前或置后来对窗口中的对象分层。

➢  **要将一个对象置于另一对象之后：**

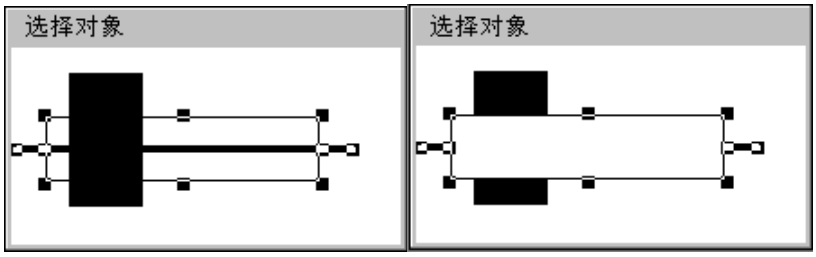
1. 选择对象。
2. 在“**排列**”菜单上，单击“**置后**”，或单击“**排列工具栏**”上的“置后”工具。选定对象即被设置在窗口中未选定对象之后：



要快速将对象置后，右击对象，指向“**前/后**”，然后单击“**置后**”。

➢  **要将一个对象置于另一对象之前：**

1. 选择对象。
2. 在“**排列**”菜单上，单击“**置前**”，或单击“**排列工具栏**”上的“置前”工具。选定对象即被设置在窗口中未选定对象之前：

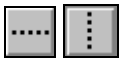


请注意最后的对象被置于最前端。

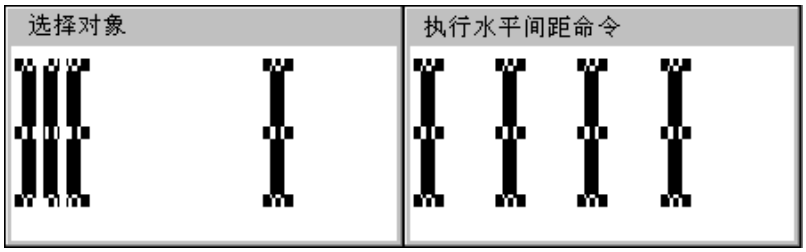
要快速将对象置前，右击对象，指向“**前/后**”，然后单击“**置前**”。

控制水平和垂直间距

您可以在最左端与最右端选定对象之间沿水平方向均匀地分布对象。您也可以控制最顶端与最底端选定对象之间的垂直间距。

➢  **要水平或垂直地均匀分布对象：**

1. 选择对象。
2. 在“**排列**”菜单上，单击“**水平间距**”（或“**垂直间距**”），或单击“**排列工具栏**”上的相应工具。选定对象即被设置为在相隔最远的两个对象之间均匀分布。例如：



有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

旋转对象

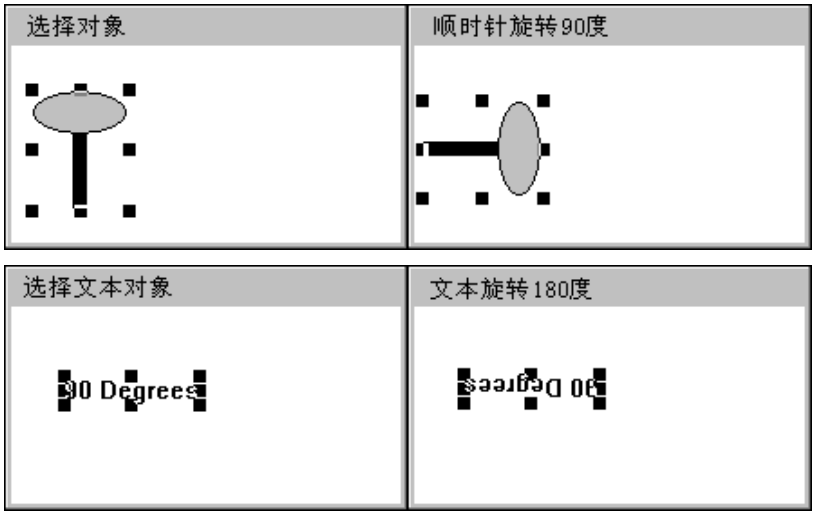
在 WindowMaker 中，您可以旋转大部分对象，包括位图、JPEG、PCX、TGA 图像和文本对象。对象能以 90 度的增量顺时针或逆时针旋转 360 度（即 90 度、180 度、270 度和 360 度）。对象附带的任何链接也将随之一起旋转。您不能旋转单元，但是可以旋转符号。

注意：在 WindowMaker 中旋转对象与在运行时动态地旋转对象无关。在 WindowViewer 中，对象通过链接“**方向**”动画链接来旋转。文本对象不能在 WindowViewer 中旋转。但是，位图或图像可以通过指定“**方向**”动画链接来旋转。



➢ **要使选定对象旋转 90 度：**

1. 选择对象。
2. 在“**排列**”菜单上，单击“**顺时针旋转**”（或“**逆时针旋转**”）。选定对象将按选定方向旋转 90 度。

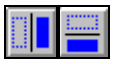


要使对象旋转 180 度，重复此过程。要使对象旋转 270 度，执行此过程两次，以此类推。

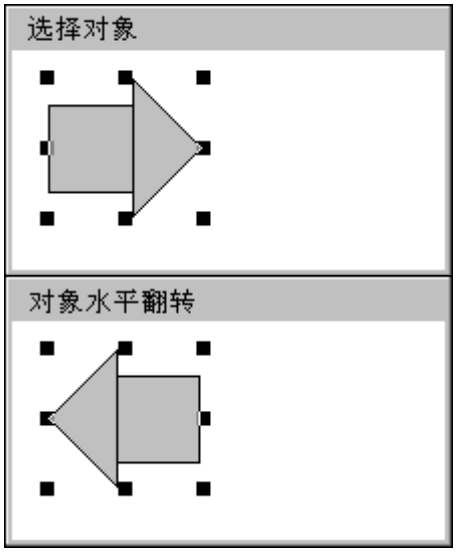
- 要快速旋转对象，右击对象，指向“**旋转/翻转**”，然后单击相应的命令。

对象镜像

您可以水平或垂直地翻转大部分的 WindowMaker 对象，包括位图、JPEG、PCX 和 TGA 图像（文本只能旋转而不能翻转）。在翻转对象时，您可以将其变换成水平或垂直的镜像。对象附带的任何链接将随之一起翻转。

➢  **要翻转选定的对象：**

1. 选择对象。
2. 在“**排列**”菜单上，单击“**水平翻转**”（或“**垂直翻转**”），或者选择“**排列工具栏**”上的相应工具。选定对象将翻转。例如：



要快速旋翻对象，右击对象，指向“**旋转/翻转**”，然后单击相应的命令。

有关工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

创建单元和符号

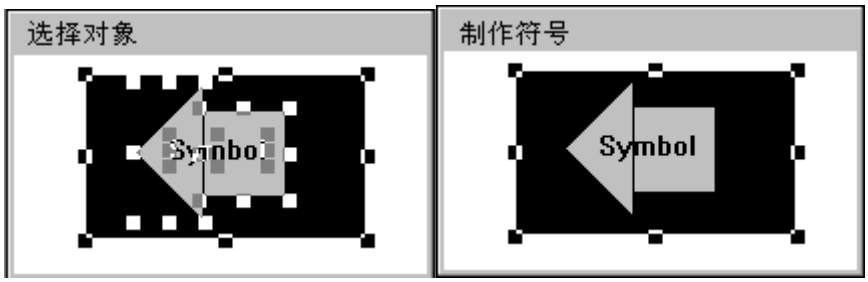
您可以将多个对象合并成两种不同的单一单元类型：单元和符号。多个单元可以合并成一个单一单元。单元是在不同图形元素之间保持固定空间关系的对象。单元内的单个组件（另一单元除外）可以设置动画效果。单元不能调整大小，也不能附加动画链接。但是，符号可以附带动画链接，而单元可以包括符号。与单元内的符号或对象关联的所有动画链接都不能更改。在单元被分解成不同组件之前，单元内的对象属性如文本、字体、行宽、半径及相对位置等不能调整大小。

- 双击单元，将出现“**替换标记名**”对话框（不是对象和符号的动画链接选择对话框）。

有关替换标记名的详细信息，请参阅第 4 章“标记名字典”。

注意：当合并单元时，每个单元都将保留，所以当合并单元被分解时，将恢复为原始单元。

符号可由多个符号和（或）多个简单对象组成，如下图所示：



如果某个选定对象附带动画链接，则该链接会附加到新符号上（如果链接粘贴缓冲区包含链接，程序会询问您是否要将该链接粘贴到新符号上）。

注意：如果多个选定对象包含链接，则您不能制作符号。如果您将两个符号合并成一个新符号，则原始符号的结构将丢失。因此，如果您分解新符号，它将会分解成各个原始符号的不同组件。两个原始符号将丢失。

有关动画链接的详细信息，请参阅第 5 章“创建动画链接”。

➢  **要创建符号或单元：**

1. 选择您要纳入单元或符号的对象。
2. 在“**排列**”菜单上，单击“**制作单元**”（或“**制作符号**”），或单击“**排列工具栏**”上的相应工具。

   要快速创建单元或符号，选择所有对象。右击选定对象之一，指向“**单元/符号**”，然后单击相应的命令。

➢  **要分解符号或单元：**

1. 选择符号或单元。
2. 在“**排列**”菜单上，单击“**分解单元**”（或“**分解符号**”），或单击“**排列**”工具栏上的相应工具。

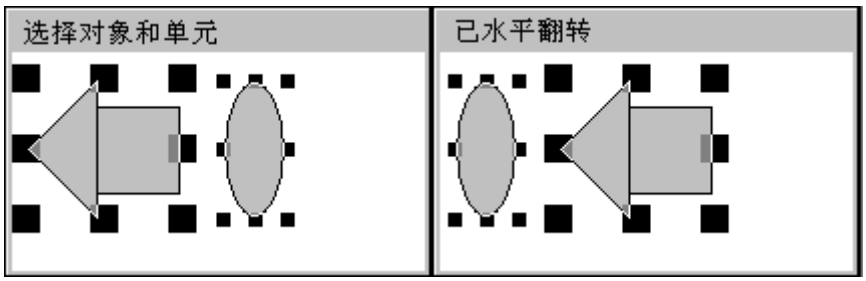
   如果符号定义了链接，则这些链接会自动保存到链接粘贴缓冲区。

   要快速分解单元或符号，右击单元或符号，指向“**单元/符号**”，然后单击相应的命令。

有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

翻转单元

当您翻转单元时，单元并没有被映射。映射的只有对象组中的单元位置。例如：



比较单元翻转前后的位置（左边）及其朝向。它的位置翻转了，但内容没有翻转。椭圆对象发生了映射。这同样适用于单元在垂直方向上的翻转。

使对象对齐网格

当您把在窗口中排列对象时，打开网格功能可以使图形按网格左上角的象素间隔对齐。如果您选定多个对象，则会按组中第一个选定对象的左上角来对齐对象。

 单击“**视图**”工具栏上的“对齐网格”工具，以打开或关闭网格对齐功能，或在“**排列**”菜单上，单击“**对齐网格**”。

- 缺省条件下，网格设置为 10 个像素大小，并且在您初次启动 WindowMaker 时显示。您可以通过“**WindowMake 属性**”对话框来配置网格的象素间隔。

有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

➢ **要配置网格：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**WindowMaker**”，或者在应用程序浏览器的“**配置**”下，双击“**WindowMaker**”。
   “**WindowMaker 属性**”对话框出现：
2. 在“**间距**”框中，输入显示网格坐标间隔的象素数。
3. 如果您要在打开 WindowMaker 的“对齐网格”功能时，在窗口中显示网格，请选择“**显示网格**”。
   - 如果您不选择“**显示网格**”，则在打开对齐网格功能时窗口将不会显示网格。

有关配置网格的详细信息，请参阅“自定义您的开发环境”。

使用图像和位图

在其它 Windows 程序中创建的所有图形对象，如图形、屏幕抓图、AutoCad 制图、JPEG、PCX 和 TGA 文件类型等，必须粘贴到 WindowMaker 的位图容器中。

WindowMaker 将位图视为单一对象。因此，您不能对位图中的单个元素设置动画，也不能将位图放入符号中。但是您可以将位图纳入单元中。

在 WindowMaker 中，您可以旋转位图、JPEG、PCX 和 TGA 图像。它们能以 90 度的增量顺时针或逆时针旋转 360 度（即 90 度、180 度、270 度和 360 度）。位图附带的任何链接将随之一起旋转。

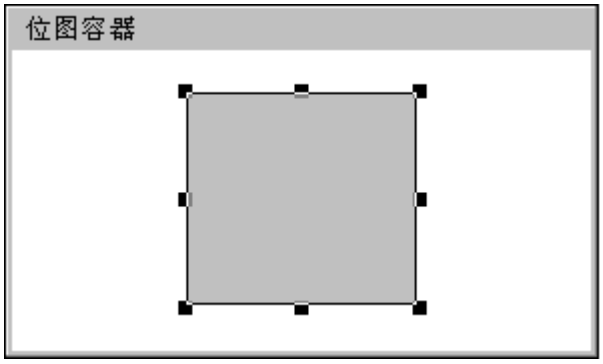
注意：在 WindowMaker 中旋转位图与在运行时动态地旋转位图无关。在 WindowViewer 中，位图或图像通过链接“**方向**”动画链接来旋转。

您也可以用透明颜色定义位图，这样您就可以让其漂浮在其它对象上。当您用透明颜色定义位图时，使用透明颜色的所有位置将显示窗口背景色或位图后面的任何对象（每个位图只能使用一种透明颜色）。

有关透明位图和图像的详细信息，请参阅“创建透明位图”。

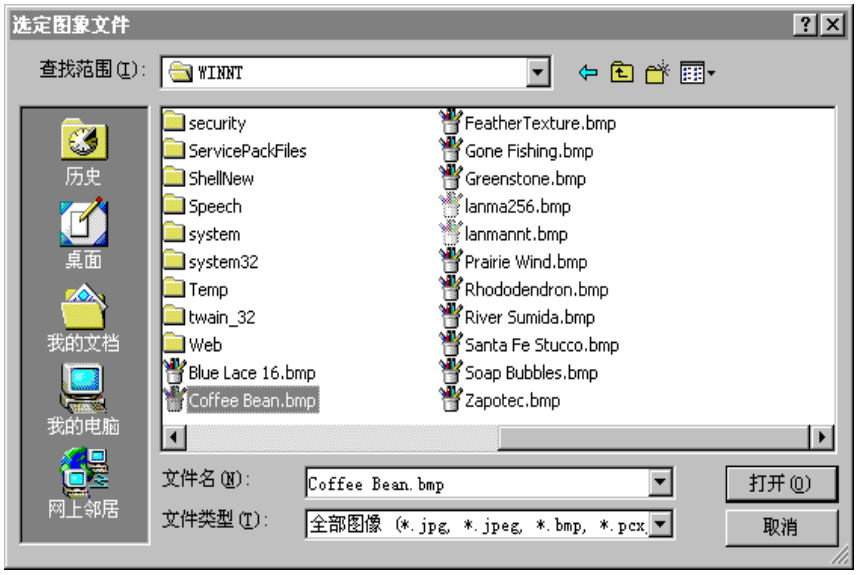
➢ **要导入位图或 JPEG、PCX 或 TGA 文件类型：**

1. 单击位图工具（光标将变成十字形），然后在窗口中画一个位图容器（任意大小）。
2. 选择位图容器：

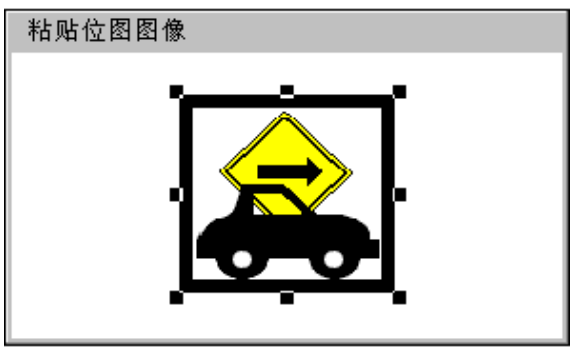


3. 在“**编辑**”菜单上，单击“**导入图像**”。“**选定图像文件**”对话框出现：

   要快速粘贴图像，右击位图容器，然后单击“**导入图像**”。



4. 找到并选择您要作为位图导入的 .BMP、.PCX、.TGA 或 .JPG 文件，然后单击“**打开**”或双击图像文件名。图像将被粘贴到位图容器中：



5. 要使用位图的原始大小，选择位图，然后在“**编辑**”菜单上，单击“**位图 - 原始大小**”。位图将设置为其原始尺寸。

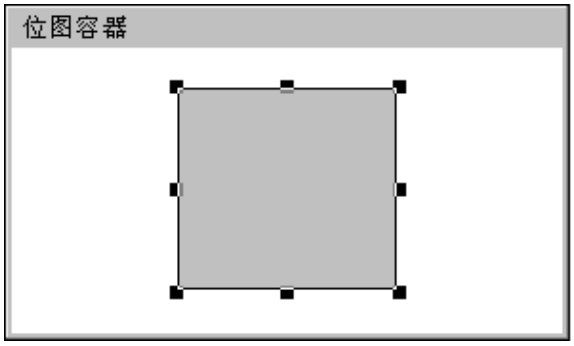
   要快速调整位图大小，右击位图，然后单击“**位图 - 原始大小**”。



从 Windows 剪贴板粘贴位图

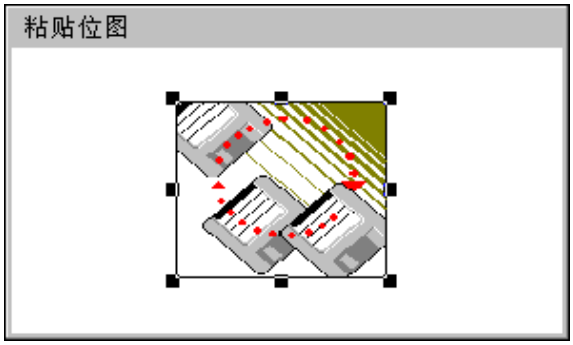
➢ **要将位图从 Windows 剪贴板粘贴到窗口：**

1. 将图形复制到 Windows 剪贴板。例如，显示图形，在按 PRINT SCRN 键的同时按下 ALT 键，将其复制到 Windows 剪贴板。
2. 单击位图工具（光标将变成十字形），然后在窗口中画一个位图容器（任意大小）。
3. 选择位图容器：

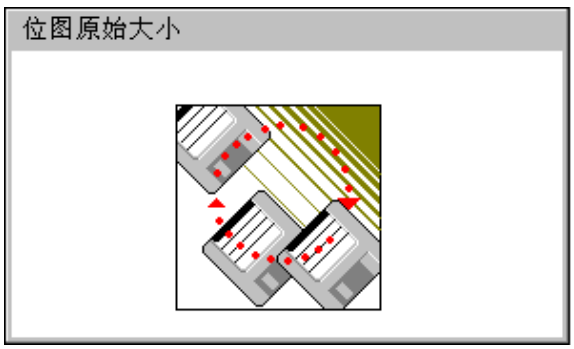


4. 在“**编辑**”菜单上，单击“**粘贴位图**”。位图将从 Windows 剪贴板粘贴到位图容器：

   要快速粘贴位图，右击位图容器，然后单击“**粘贴位图**”。



5. 要使用位图的原始大小，选择位图，然后在“**编辑**”菜单上，单击“**位图 - 原始大小**”。位图将设置为其原始尺寸。



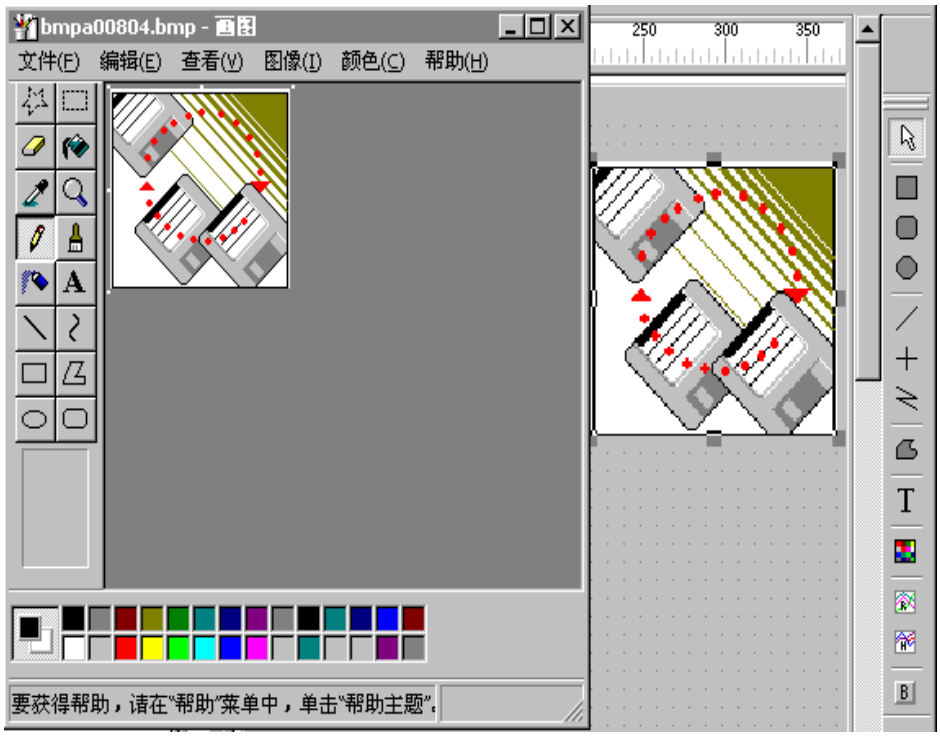
   要快速调整位图大小，右击位图，然后单击“**位图 - 原始大小**”。

编辑位图

您可以直接从 WindowMaker 编辑位图。当您在 WindowMaker 中编辑位图时，将启动与 .bmp 扩展名关联的应用程序。例如，如果位图是使用 MSPaint 创建的，则在 WindowMaker 中编辑位图时，MSPaint 会自动启动，如下所述：

➢ **要在 WindowMaker 中使用 Windows Paint 编辑位图：**

1. 选择位图。在“**编辑**”菜单上，单击“**编辑位图**”。Microsoft Paint 打开并显示位图：



要快速编辑位图，右击位图，然后单击“**编辑位图**”。

2. 在 MS Paint 中编辑位图。
3. 退出 MS Paint 以返回 WindowMaker。

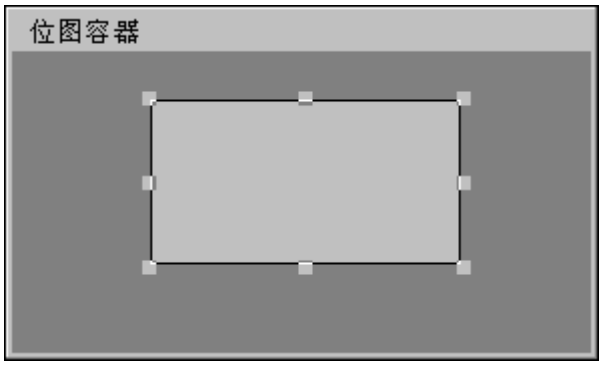
注意：在使用原来的应用程序编辑位图时，您将不能使用 WindowMaker。要返回 WindowMaker，退出原始应用程序。

创建透明位图

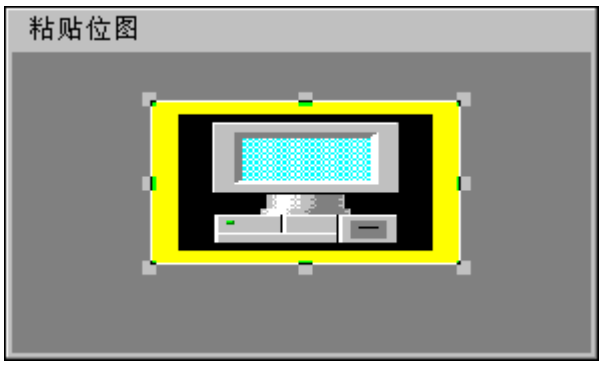
您可以用透明颜色定义位图或图像，这样就可以让其漂浮在其它对象上。当您用透明颜色定义位图或图像时，使用透明颜色的所有位置将显示窗口的背景色或位图后面的任何对象。

➢ **要创建透明位图：**

1. 单击位图工具（光标将变成十字形），然后在窗口中画一个位图容器（任意大小）。
2. 选择位图容器：

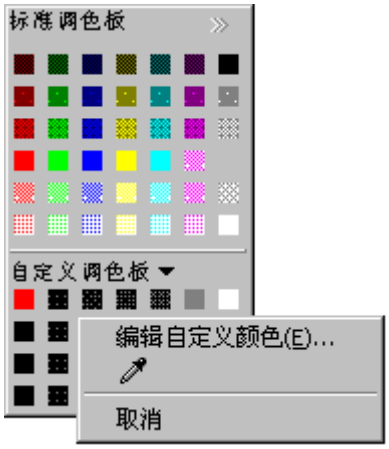


3. 右击位图容器，然后单击“**粘贴位图**”（如果您已经将图形复制到 Windows 剪贴板）。否则，单击“**导入图像**”（找到并选择要打开的 .BMP、.PCX、.TGA 或 .JPG 文件）。位图图像将粘贴到位图容器中：



4. 右击位图，然后单击“**位图 - 原始大小**”，使位图恢复其原始大小。
5. 选定位图，单击“**格式**”工具栏上的“透明颜色”工具，以打开透明调色板。

6. 右击调色板底部“**自定义调色板**”区域中的空白颜色方块。“**编辑自定义颜色**”对话框出现。



7. 单击吸管工具（“**编辑自定义颜色**”对话框将关闭）。

8. 单击位图中您要设置为透明的颜色。该颜色将复制到透明调色板中选定的颜色方块上。

9. 单击颜色方块以将透明颜色应用到位图上：

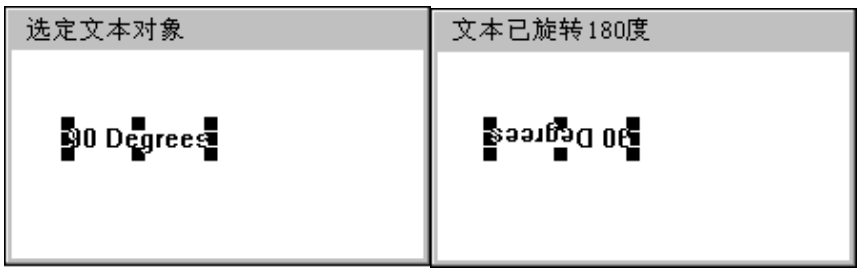


在本例中，我们使位图的宽边区域透明。因此，位图现在可以显示出窗口的背景色。如果对象位于现在的透明区域之后，则也可以在位图上显示出来。

注意：一个位图只能应用一种透明颜色。

使用文本对象

在 WindowMaker 中，您可以改变任意选定文本对象的字体、字型、字体大小、对齐方式和旋转。您也可以按 90 度的增量使文本对象顺时针或逆时针旋转 360 度（90 度、180 度、270 度和 360 度）。例如：



注意：在 WindowMaker 中旋转文本对象与在运行时动态地旋转对象无关。**“方向”**动画链接不适用于文本对象。因此，文本对象不能在 WindowViewer 中旋转。

使用**“格式工具栏”**上包含的工具，您可以对选定的对象快速应用**“文件”**菜单上的大部分命令。例如：



有关格式工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

格式化文本对象

所有 WindowMaker 文本命令都是针对单个或多个文本字符串选项及数值域进行的。如果在执行**“文本”**菜单上的命令时没有选定文本对象，则该命令会自动应用于**“格式工具栏”**上的相应文本工具的缺省设置，以及**“绘图对象工具栏”**上的文本工具的缺省设置之上。

文本对齐方式属性设置对于用于输出动态值的文本对象特别重要。对齐方式决定在运行时如何显示不同长度的域。

例如，如果您要在居中或右对齐的文本字符串末尾显示数值，则每当显示的数字位数改变时，整个文本字符串（包括值）将会再次居中或右对齐。

有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

显示数值

文本对象也可用于显示静态或动态数值。通过对文本对象附加**“触动链接用户输入 - 模拟”**或**“值显示-模拟”**动画链接，您可以显示模拟（整型或实型）标记名的数值。

要确定模拟量的显示格式，可使用下面的四个字符：

0	零
#	数字或磅符号
,	逗号
.	句点或小数点

下面举例说明模拟量的字段格式。

字符	**描述**
#	显示任意整数，例如： **1234** 显示为 **1234**（只需一个 # 号）
0.0	强制使用一个前导零和一个小数位。例如： **.1** 显示为 **0.1** **77.1** 显示为 **77.1**
00000	强制使用前导零，例如： **123** 显示为 **00123** **1234** 显示为 **01234** **12345** 显示为 **12345**
#,##0.0	如有需要，插入逗号和前导零，以及一个小数位。例如： **1234.56** 显示为 **1,234.6** **123.4** 显示为 **123.4**
0,000.0	强制使用逗号、前导零和一个小数位，例如： **12.3** 显示为 **0,012.3**

注意：如果您在格式中使用零，则其后面必须跟零。小数点右边所有位数必须始终为零。例如，000.00 正确，而 #0#0.0# 则不正确。

- 所有常规文本格式均适用于数值，包括字体、大小、颜色、对齐方式及粗体。

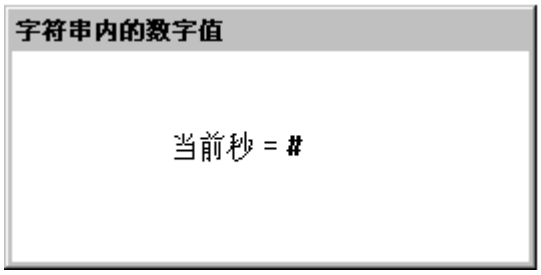
➢ **要创建文本对象：**

1. 单击“**绘图对象工具栏**”上的“文本”工具。
2. 单击窗口并输入文本字符串。

   要快速访问可应用于文本对象的各种命令，右击文本对象，然后单击相应的命令。

➢ **要在字符串内显示数值：**

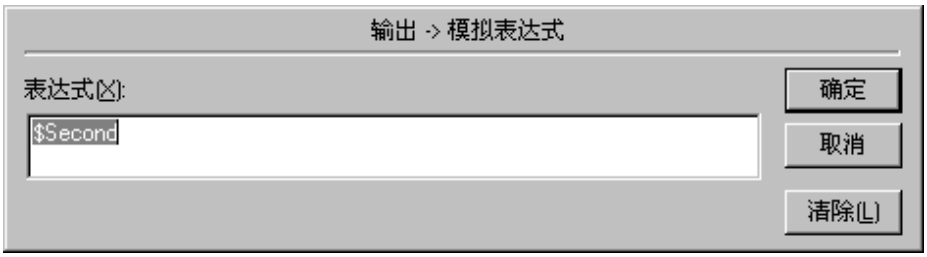
1. 单击文本工具，然后用前面所述的有效数字格式之一在窗口中输入文本对象。例如：



2. 选择对象，然后在“**特别**”菜单上，单击“**动画链接**”或双击文本对象。动画链接选择对话框出现。

   要快速访问对话框，右击文本对象，然后单击“**动画链接**”。

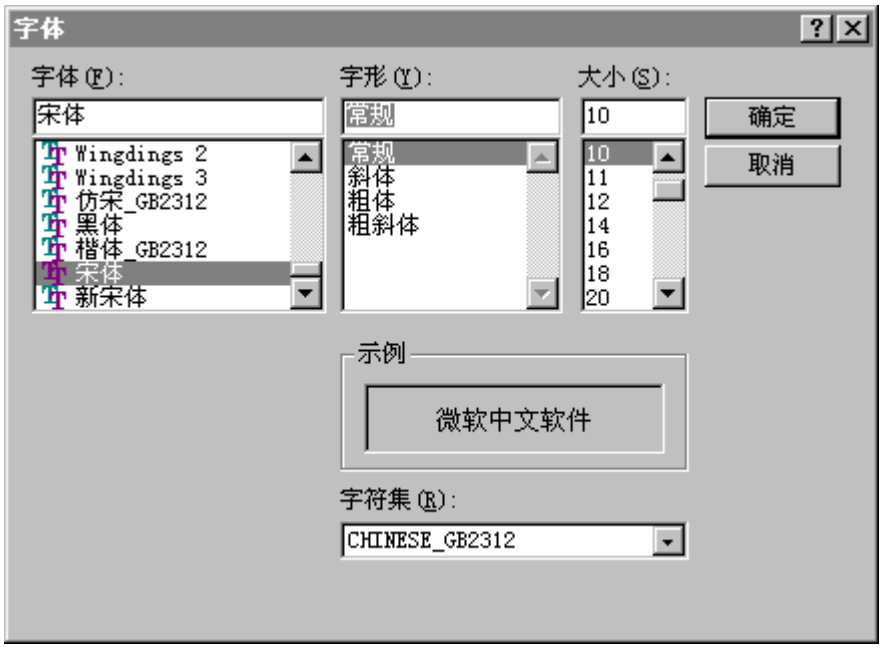
3. 在“**值显示**”区域中，单击“**模拟**”。“**输出量->模拟表达式**”对话框出现：



4. 在“**表达式**”框中，输入模拟标记名或表达式（在本例中，使用系统标记名 **$Second**）。
5. 单击“**确定**”。
6. 单击菜单栏右上角的“**运行**”快速切换开关（或使用快捷键 ALT+！）切换到 WindowViewer，或者在“**文件**”菜单上，单击“**WindowViewer**”。
7. 如果您使用本实例，则会看到文本字符串中的磅 (#) 符号位置显示当前系统秒值（0-59 之间）。
8. 单击菜单栏右上角的“**开发**”快速切换开关（或使用快捷键 ALT + !）返回 WindowMaker，或者在“**文件**”菜单上，单击“**WindowMaker**”。

➢ **要改变字符串的字体、字形和大小：**

1. 选择文本字符串，然后在“**文本**”菜单上，单击“**字体**”，单击“**格式工具栏**”上的“字体”工具。标准的 Windows“**字体**”对话框出现：



2. 从“**字体**”列表中选择所需字体（字体名将显示在“**字体**”字段中）。一旦选定字体，可供选择的字形和大小将出现在“**字形**”和“**大小**”字段中。在选定字体大小时，相应字形和大小的字体样本会出现在“**示例**”字段中（参见上面的实例）。

3. 单击“**确定**”。

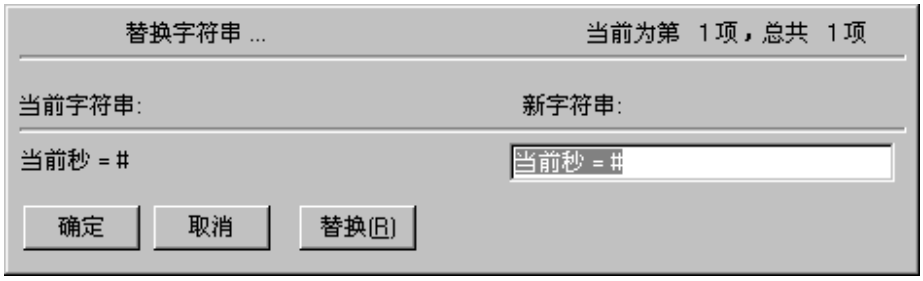
注意：字体磅值会根据所选字体的可用磅值范围来放大或缩小。WindowMaker 的缺省字体是系统字体，并且不能改变大小。在改变大小前需先选择 Windows True-Type 字体。

编辑文本对象

➢ **要更改对象中的文本：**

1. 选择包含文本的对象或按钮。
2. 在“**特别**”菜单上，单击“**替换字符串**”。“**替换字符串**”对话框出现：

   ☞ 要快速访问此对话框，右击文本对象，指向“**替换**”，然后单击“**替换字符串**”。



有关模拟输入/显示链接的详细信息，请参阅第 5 章“创建动画链接”。

3. 在“**新字符串**”框中，输入新字符串，然后单击“**确定**”。

   ☞ 您也可以对符号或单元中包含的字符串使用此命令，以更改用“按钮”工具所画的按钮上的标签。

   在更改文本字符串时，它会保留其所有原始属性，包括字体、字形和颜色等。所有常规文本格式也适用于数值。

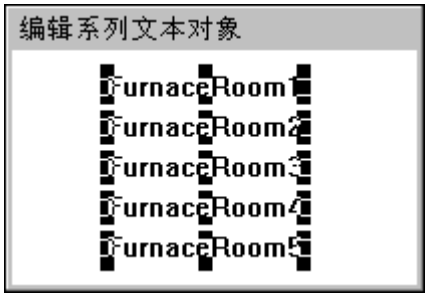
   您也可以同时选择和编辑多个字符串对象。

替换部分文本对象

您可以更改文本对象的部分文本，InTouch 会自动对使用相同文本的所有选定文本对象作出更改。

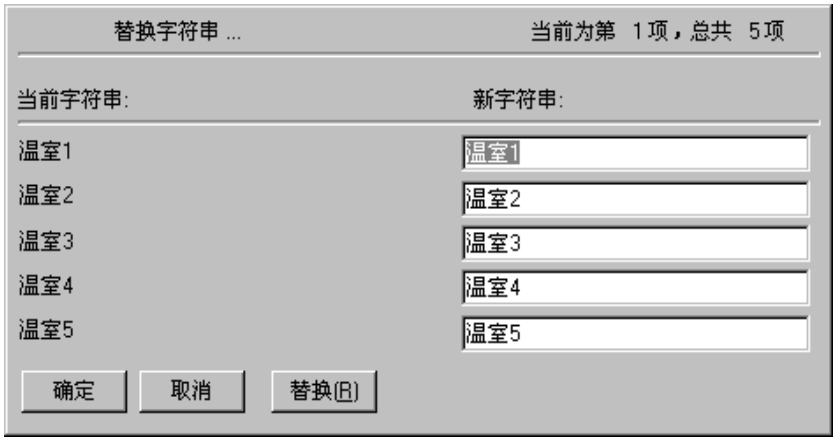
➢ **要更改一系列文本对象中的部分文本：**

1. 选择所有文本对象。



2. 在“**特别**”菜单上，单击“**替换字符串**”。“**替换字符串**”对话框出现：

   要快速访问此对话框，右击文本对象，指向“**替换**”，然后单击“**替换字符串**”。

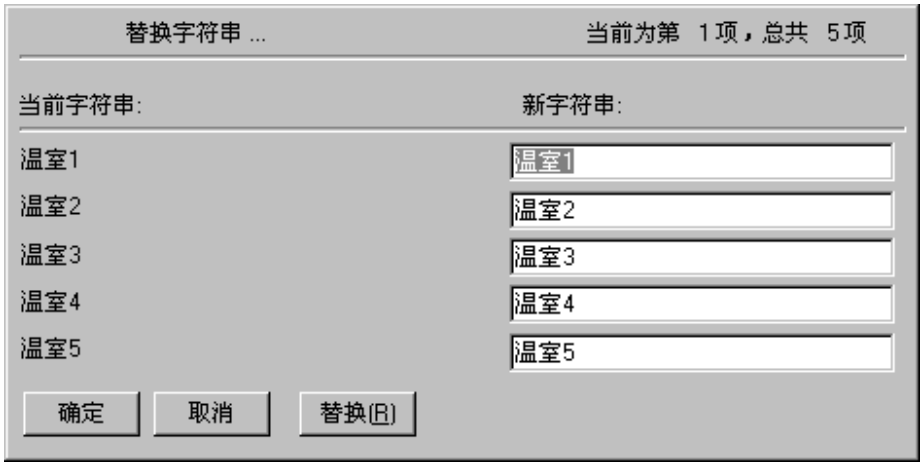


   如果您右击任意对话框中的文本输入框，将出现一个菜单，显示可应用于选定文本的命令。

3. 单击“**替换**”。“**替换文本**”对话框出现：



4. 在“**旧文本**”框中，输入需要替换的部分字符串。
5. 在“**新文本**”框中，输入替换文本。
6. 单击“**确定**”。“**替代字符串**”对话框重新出现，显示对所选文本字符串所作的更改。



7. 单击“**确定**”。所有选定的文本对象将自动被修改。

使用线条和轮廓

您可以改变线条对象的样式和宽度，包含椭圆形、长方形、多边形、位图或图像的轮廓。您可以对单个或多个选定对象应用线条样式或宽度的更改。

“**线**”菜单分成两个部分。上面的部分包含线宽，下面的部分包含线样式。例如：



- **要应用线命令：**

选择对象，然后在“**线**”菜单上，单击所需的线样式或宽度。

- 如果您在选择线样式或宽度时没有选择对象，此所作更改将应用到“**向导工具栏**”中所有线工具的缺省设置中。

注意：您只能改变实线的宽度。虚线只能是固定的单个象素宽。宽线在运行时需要更长的时间才能绘出。

- **要删除对象的轮廓：**

选择对象，然后在“**线**”菜单上，单击“**无线**”。对象的轮廓即被删除。

使用向导

向导可以在应用程序开发阶段为您节省大量的时间，并且易于使用和配置。要配置向导，先安装向导，在“**向导选择**”对话框中选择向导，将其粘贴到窗口中，然后双击它。屏幕将出现相应的配置对话框（假定该向导可进行配置）。

例如，如果您要使用游标向导，则需要配置的项目包括标记名效果、游标的最大与最小范围标签以及填充色等等。通过使用向导您可以节省大量的开发时间，因为您不必再花时间单独绘制对象的各个组件、设置对象的值范围，或设置对象的动画效果。

FactorySuite InControl 程序包括下列五个可以纳入 InTouch 窗口的向导。这些向导可实现 InControl 与 InTouch 之间简单而有效的相互作用。

向导	描述
InControl 项目	启动 InControl 项目。此向导针对指定项目启动 InControl，允许操作员使用所有的 InControl 功能来编辑、编译、下载和运行该项目中的程序。
配置运行时引擎	启动运行时引擎并选择其运行节点。
InControl 模式	用于将当前下载到运行时引擎的程序设置为指定模式（运行、暂停、单步）。
InControl 编辑	启动项目中的单个程序。此向导在指定程序中的指定行，于开发环境下启动 InControl。您可以使用开发环境下的所有可用工具来编辑、编译、下载和运行程序。
InControl 运行时添加标记	将 InTouch 标记名与 InControl 符号（变量）相关联。

➢ **要安装或删除向导：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**向导 / ActiveX 安装**”，或者在应用程序浏览器中，双击“**向导 / ActiveX 安装**”。“**向导 / ActiveX 安装**”对话框出现，并显示“**向导安装**”属性页：

   在应用程序浏览器中，您也可以右击“**向导 / ActiveX 安装**”，然后单击“**打开**”。



注意：可同时安装的最多向导数为 43。

2. 在“**已安装的向导**”列表中，选择您要从应用程序中删除的向导，然后单击“**删除**”。屏幕出现一个要求您确认删除的对话框。

注意：仅当向导显示在“**已安装的向导**”列表中时，“**删除**”按钮才被激活。

   要选择一组向导，单击列表中的第一个向导，在按住 SHIFT 键的同时单击您要选择的最后一个向导。如此将选定列表中第一个和最后一个选项之间的所有向导。要选择列表中的多个不连续向导，单击第一个向导，在按住 CTRL 键的同时单击下一个向导。重复此步骤，选定您要选择的所有向导。

3. 单击“**是**”以删除向导。被删除的向导将移至“**已卸载的向导列表**”中。

注意：向导仅被移除，而没有被删除。但是它将不能再载入内存。

4. 要安装向导，在“**已卸载的向导列表**”中选定向导，然后单击“**安装**”。

注意：仅当向导显示在“**已卸载的向导列表**”中时，“**安装**”按钮才被激活。

5. 如果您要从另一目录安装向导，单击“**搜索**”。“**搜索向导文件**”对话框出现：

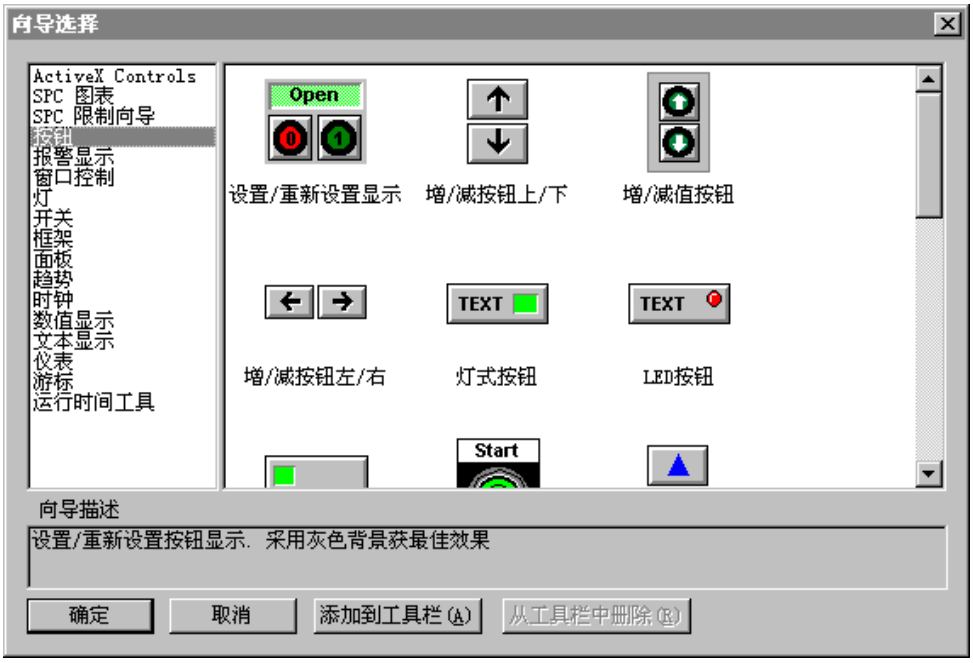


6. 找到包含您要安装的向导的目录，然后单击“**确定**”。向导安装对话框重新出现。

   所有找到的向导将会出现在“**已卸载的向导列表**”中，您可以如前所述开始安装。

➢  **要将向导放入窗口：**

1. 单击“**向导/ActiveX 工具栏**”上的向导对话工具，“**向导选择**”对话框出现：



2. 在向导列表中，单击您要使用的向导类别。

   该类别中的所有可用向导将出现在显示区域中。例如，如果您选择“**按钮**”，则所有可用的按钮向导将立刻出现在显示区域。

3. 选择您要使用的向导，然后单击“**确定**”或双击该向导。对话框关闭，窗口重新出现。

   要在“**向导/ActiveX 工具栏**”中添加向导，单击“**添加到工具栏**”。一旦您将向导添加到“**向导/ActiveX 工具栏**”，就可以随时选择该向导并将其粘贴到您打开的窗口中。

注意：您可以添加到工具栏中的向导数目取决于您的系统资源。

---

4. 当您返回窗口时，光标将变成弯头符号 。单击您要粘贴向导的窗口位置。

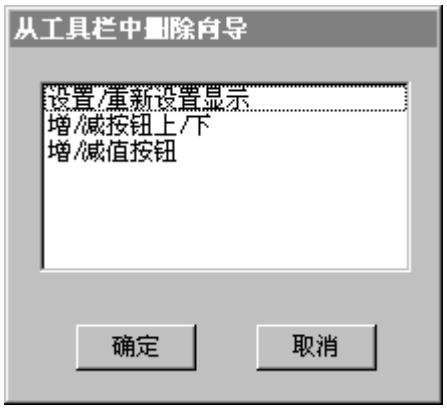
5. 双击向导以进行配置（如果适用）。

注意：可以使用某些工具栏功能来直接修改适用的向导。例如，“**缩小字体**”工具、“**线颜色**”工具和“**填充色**”工具等等。

有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

➢  **要从工具栏删除向导：**

1. 单击“**向导/ActiveX 工具栏**”上的向导对话工具，“**向导选择**”对话框出现：
2. 单击“**从工具栏中删除**”，“**从工具栏中删除**”对话框出现：



3. 选择您要从工具栏删除的向导。
4. 单击“**确定**”。

InTouch 窗口控件向导

窗口控件向导是复杂的对象。与一般向导不同，它们通过 InTouch QuickScript 来提供增强功能。控件向导可用于编辑数据对象和操作员输入。窗口控件向导也具有 InTouch 标记名**点域**，其某些属性可以在开发或运行时通过使用 QuickScript 函数来访问。

您可以在 InTouch 应用程序中使用窗口控件向导，以在运行时显示文本/数据、收集用户输入或为用户提供选项。选项形式可以是列表框、复选框、组合框和单选（选项）按钮。您可以使用文本框来显示或输入文本/数据。

在配置窗口控件向导时，您必须指定一个**控件名**来识别控件。当您执行窗口控件 QuickScript 函数时，InTouch 将使用**控件名**来识别控件。因此，您还必须在 QuickScript 函数中指定**控件名**参数。例如：

```
SetPropertyD ( "ControlName.Property", Discrete );
```

有关使用窗口控件 QuickScript 函数的详细信息，请参阅 *《InTouch 参考指南》*。

控件名不会计入应用程序标记名计数中，并且对于每个控件必须是唯一的。虽然对标记名不作要求，但它们对于有效使用控件却是必不可少的。例如，如果选定项目未自动指定给标记名，以使之可以让 InTouch 访问，则在列表框中选择项目是没有意义的。

窗口控件属性（类似于标记名**点域**）可在开发 (WindowMaker) 和运行时 (WindowViewer) 修改。它们也支持可在运行时处理的特定 QuickScript 函数，以修改列表、下载文件、禁用控件等等。

注意：为确保正常运行，窗口控件对象不能互相重叠。为此，在 WindowMaker 中选择对象并确保对象的句柄没有接触另一对象。

接受键盘输入的窗口控件不支持 SuiteVoyager。

赋予列表框或组合框的标记名初始值不能用于初始化列表框或组合框的值。

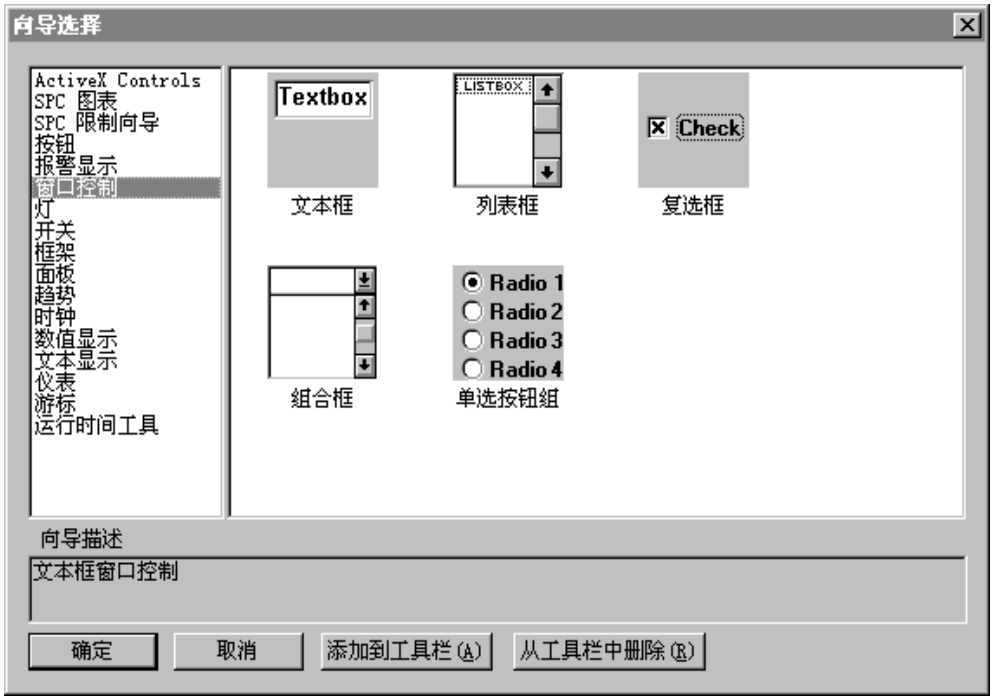
有关使用窗口控件**点域**的详细信息，请参阅您的 *《InTouch 参考指南》*。

使用 InTouch 窗口控件向导

可用的窗口控件向导包括：文本框、复选框、组合框、列表框、单选（选项）按钮等。窗口控件向导也具有**点域**和 QuickScript 函数，可用于在运行时动态地对其进行控制。

- 与其它向导一样，窗口控件向导可以粘贴到窗口中。

有关粘贴向导的详细信息，请参阅“将向导放入窗口”。



- 要获得最佳的窗口控件三维效果，为您的窗口选择灰色背景。如果窗口不能被设成灰色，请在窗口控件向导后面放置一个灰色“窗格向导”。

窗口控件使用准则

使用窗口控件向导时切记遵循下列准则：

1. 窗口控件向导只有在不与其它窗口控件向导或其它常规图形对象重叠时，才能正常工作。

   - 要核实窗口控件向导没有与任何其它对象重叠，请在 WindowMaker 中选定该向导。确认其选择句柄不与屏幕上的其它图形对象接触。

2. 使用窗口控件向导时务必谨慎小心。

注意：在一个窗口中放置 10 到 20 个窗口控件向导不仅不美观，而且难以浏览显示。当必须使用多个窗口控件向导时，我们建议您调用其它带窗口控件向导的对话框。

文本框控件向导



文本框控件向导是一个多功能控件，可用于接受用户输入或显示文本，如记事本文件（仅限于 ASCII 平面文件）。您可以配置文本框使之允许用户输入，或设置为只读以只用于显示。您只能为文本框控件向导指定**消息**型标记名。

文本框控件向导的 QuickScript 实例：

```
wcLoadText("TextBox_1",FileName);

wcSaveText("TextBox_1",FileName);
```

注意：如果标记名定义为最大长度，则只能从文本框内容将该字符数赋值给标记名。如果文本框未指定标记名，则其内容最长可达 65,535 个字符。

列表框控件向导



列表框控件向导向用户显示一个选项列表。这些选项垂直分布在单个列中。如果项目数超过列表框中可显示的数目，控件上会自动出现滚动条。列表框要求用户从列表中进行选择而不允许用户输入。您只能为列表框控件向导指定**消息**型标记名。

下面是列表框控件向导的 QuickScript 实例：

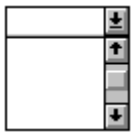
```
IF (ItemToAdd == "") THEN
    Show "Cannot Add Blank";
ELSE
    wcAddItem("ListBox_1",ItemToAdd);
    {获得刚添加项目的索引。}
        {因为列表已排序，我们不能对新项目位置作任何假设。}
    GetPropertyI("ListBox_1.NewIndex",ListBox_NewIndex);
    {现在，设置由用户在屏幕上指定的项目数据。}
        {从现在起，此项目将与此数据相关联。}
        {它允许您将数字与字符串相关联；字符串显示在列表中。}
    wcSetItemData("ListBox_1",ListBox_NewIndex,
       ListBox_ItemData);
    {由于我们刚添加项目，所以需更新“NumItems”变量。}
    GetPropertyI("ListBox_1.ListCount",ListBox_NumItems);
ENDIF;
```

列表框和组合框控件向导使用内部统一编号系统（项目索引），对列表中的每个项目自动编号。例如，列表中的第一项编号为 1，第二项编号为 2，以此类推。项目索引是一个 32 位整数，作为窗口控件“项目”的一个参数使用。

注意：在对列表框和组合框使用 **wcLoadList()** 和 **wcSaveList()** 时，必须提供特定的格式和信息。

有关窗口控件 QuickScript 函数的详细信息，请参阅您的 *《InTouch 参考指南》*。

组合框控件向导



组合框控件向导结合了文本框和列表框的特点。选项垂直地显示在单个列中。如果项目数超过列表框中可显示的数目，控件上会自动出现滚动条。组合框控件向导允许用户通过输入文本或从列表中选择项目来进行选择。您只能为组合框控件向导指定**消息**型标记名。

组合框有三种类型：

类型	描述
简单	组合框只显示列表。要显示列表框中的所有项目，列表框必须画得足够大。如果超出可显示项目，会自动插入一个垂直滚动条。简单的组合框允许用户输入列表中未列出的选项，或显示与所输入字母匹配的列表中的第一项。如果未找到匹配项，则显示列表的顶部。
下拉	此组合框允许用户直接输入文本，或单击箭头来打开选项列表。与简单组合框一样，此控件允许用户输入列表中未列出的选项，或显示与所输入字母匹配的列表中的第一项。如果未找到匹配项，则显示列表的顶部。
下拉列表	此组合框类似于简单列表框，即会显示可供选择的选项列表。与列表框不同的是，列表仅在单击箭头后才显示。此控件类型可用于保留屏幕空间。

下面是组合框控件向导的 QuickScript 实例：

```
wcAddItem("ComboBox_1", UserMessage );
```

此处：*UserMessage* 是赋给字符串输入链接的标记名。当操作员输入一条新消息，然后单击链接到“**按下时**”QuickScript 的此动作按钮时，该消息会在组合框中显示为控件名“ComboBox_1”。

复选框控件向导



复选框指明某个条件是开/关、真/假或是/否。复选框彼此独立，允许用户同时选择或撤消选择任何多个复选框。复选框返回一个离散值。如果没有选定，则返回 0；如果选定，则返回 1。您只能为复选框控件向导指定**离散**型标记名。

下面是复选框控件向导的 QuickScript 实例：

```
{ Clear any previous machine }

Machine = "";

IF (Cutter_Selected) THEN
   Machine = Machine + "Cutter";

ENDIF;

IF (Mixer_Selected) THEN
   Machine = Machine + "Mixer";

ENDIF;
```

此处：*Cutter_Selected* 是赋给第一个复选框向导中的控制名“Checkbox_1”的标记名；*Mixer_Selected* 是赋给第二个复选框向导中的控制名为 Checkbox_2”的标记名；

Machine 是赋给用于显示选定复选框名称的字符串输出链接的标记名。

单选（选项）按钮控件向导



单选或选项按钮提供一组选项供用户选择。与复选框不同，单选钮相互关联。选择一个单选钮会立即清除组中的所有其它按钮。单选钮返回一个整型值。单选钮控件的值对应于选定的单选钮。

例如，如果选定“Radio 1”选项，则当前值为 1；如果选定“Radio 4”选项，则值为 4。您只能为单选钮控件向导指定**整型**标记名。

下面是单选（选项）按钮控件向导的 QuickScript 实例：

```
SelectedMachine=1;
```

此处：*SelectedMachine* 是赋予控件“RadioButtonGroup1”的单选钮向导的整型标记名。这是一个窗口“**显示时**”QuickScript，它将 *SelectedMachine* 标记名的值置为 1（设置缺省值为初次显示窗口时选择第一个单选钮）。当操作员选择另一单选钮时，*SelectedMachine* 的值将根据所选按钮而改变。例如，如果单选钮组具有 4 个选项而操作员选择了第三个按钮，则 *SelectedMachine* 的值将置为 3。

配置窗口控件向导

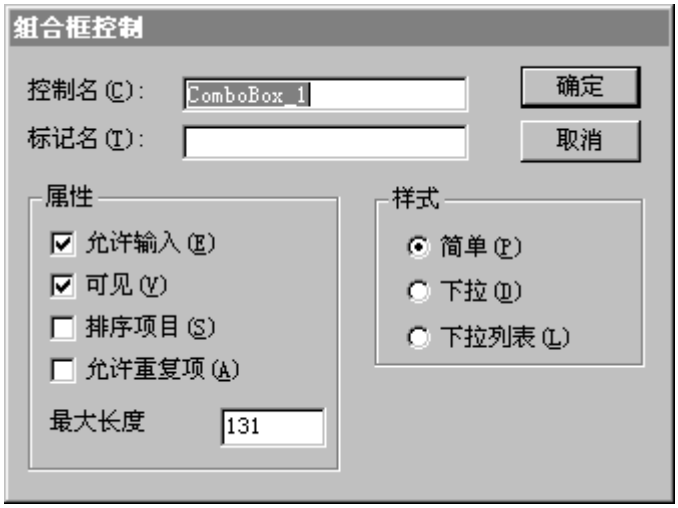
每个窗口控件向导根据其预期功能都具备独特的配置对话框。对话框中显示的选项都是可配置属性，这些属性不能由其它 WindowMaker 工具提供。但是，象颜色、字体和大小这样的属性可以通过相应的 WindowMaker 工具进行修改。

大部分的窗口控件向导都支持 QuickScript 函数。例如，您可以创建一个数据改变 QuickScript 来下载和清除列表，在列表中添加和删除项目等。

有关窗口控件 QuickScript 函数及**点域**的详细信息，请参阅您的*《InTouch 参考指南》*。

➢ **要配置窗口控件向导：**

1. 将向导粘贴到您的窗口中。
2. 双击向导，其相应的配置对话框出现。例如：



如果您右击任意对话框中的文本输入框，将出现一个菜单，显示可应用于选定文本的命令。

3. 在“**控制名**”框中，输入用于识别窗口控件的唯一名称。

   为确保运行时正确操作，您必须为每个窗口控件向导指定一个唯一的“**控制名**”。当您执行窗口控件 QuickScript 函数时，WindowViewer 使用“**控制名**”来识别控件。“**控制名**”必须以字母字符开头（第一个字母字符后可使用下划线或数字）并且不能包含任何特殊字符。

4. 您可以选择在配置窗口控件时指定**标记名**。但是，一旦指定标记名，其值将自动设置为控件的 **.Value** 属性（即列表框中选定项目的索引）。
5. 输入所有其它必要项目，并为您当前配置的特定窗口控件设置所有适用参数。
6. 单击“**确定**”。

窗口控件向导属性

Windows 控件向导具有与 InTouch 标记名**点域**类似的属性，即可以是读写或只读。某些属性可在开发时访问，而某些属性则在运行时访问。它们可以标识为 *ControlName.x* ，此外 *x* 指属性。

例如，如果窗口控件的 **.Visible** 属性等于零，则此控件在窗口中将不可见。与 InTouch 标记名相类似，**.Value** 是窗口控件向导的缺省属性。

在 WindowMaker 中，窗口控件向导属性，如文本字体、大小和颜色等可以通过使用相应的 WindowMaker 工具栏或菜单命令来进行修改。那些工具栏或菜单命令不支持的属性可在向导的配置对话框中定义。窗口控件向导的其它属性是动态的，在运行时为读写或只读。这类似于 InTouch 标记名（**点域**）的运行时属性，如 **.Value** 和 **.Name**。与 InTouch 标记名不同的是，窗口控件向导的运行时属性可通过 QuickScript 函数，而不是动画链接表达式来访问。

取决于属性的不同，运行时属性可以是读写或只读。必须使用 **GetProperty()** 和 **SetProperty()** QuickScript 函数来控制或检索这些属性。下面简要描述各个窗口控件属性。

属性	描述
.Caption	决定复选框显示的“消息”。
.Enabled	决定控件对象是否可对应操作员生成的事件。
.ListCount	决定列表框或组合框中的项目数。
.ListIndex	决定列表中当前选定项目的对应索引（*标记名*或*数字*）。 **注意：**索引是定义列表中的特定项目的数字。当使用列表框时，索引 –1 表示当前没有选定项目；当使用组合框时，索引 –1 表示用户已在控件的文本输入字段中输入新文本。
.NewIndex	返回通过 **wcAddItem()** 或 **wcInsertItem()** 函数添加到列表框或组合框的最后一个项目的对应整型索引（*标记名*）。
.ReadOnly	决定文本框的内容是只读还是读写。
.TopIndex	决定列表框最顶端项目的对应整型索引。
.Value	所有 InTouch 窗口控件向导的缺省属性。此属性更改在 InTouch 标记名和窗口控件向导中保持同步。
.Visible	决定窗口控件在窗口中是否可见。 窗口控件向导属性不会出现在“**选择域名**”对话框中。

例如：

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

此处：

参数	描述
ControlName	为窗口控件向导配置的“**控件名**”，例如 CheckBox_1，或报警对象名，例如 AlmObj_1。
.Property	窗口控件或报警对象属性。
Tagname	在函数处理时存储属性值的有效 InTouch 标记名（将返回相同类型）。

有关窗口控件向导的详细信息，请参阅“InTouch 窗口控件向导”。

窗口控件向导函数

下面简要介绍可在 InTouch Windows 控件向导中使用的 InTouch QuickScript 函数。

函数	描述
wcAddItem	在列表框或组合框中加入指定字符串。
wcClear	从列表框或组合框中删除所有项目。
wcDeleteItem	删除与列表框或组合框中的项目索引参数关联的项目。
wcDeleteSelection	从列表中删除当前选定的项目。适用于列表框和组合框。
wcErrorMessage	对于给定的错误号，wcErrorMessage() 返回描述该错误的一个字符串消息。适用于列表框、文本框、组合框、单选钮和复选框。
wcFindItem	决定列表框或组合框中与指定字符串匹配的第一个项目的对应索引。
wcGetItem	返回与列表框或组合框中的相应索引相关联的项目字符串的值属性。
wcGetItemData	检索与列表框或组合框中的列表项目关联的整型值。
wcInsertItem	在列表框或组合框中插入字符串。
wcLoadlist	用新项目替代列表框或组合框的内容。
wcLoadText	用新字符串替代文本框的内容。
wcSavelist	用列表对象中的项目替代文件名的内容。
wcSaveText	将文本框中包含的文本保存到文件名。
wcSetItemData	对列表框中的项目指定一个整型值。

使用 ActiveX 控件

ActiveX 控件的前身为 OLE 控件或 OCX，是以标准方式执行特定功能的独立软件组件，它为可重复使用的组件定义了标准接口。ActiveX 控件不是独立的应用程序，而是置入控制容器内的服务器。要使用 ActiveX 控件，必须将其置于 ActiveX 容器内。InTouch 是一种 ActiveX 容器。Microsoft VisualBasic 和 Internet 浏览器也是 ActiveX 容器。

ActiveX 控件与 InTouch 向导几乎完全一样，不同的是它给 InTouch 应用程序带来了引人注目的新功能。您可以使用 Visual Basic、Microsoft VC++ 或其它第三方开发工具来创建 ActiveX 控件，也可以从第三方购买某些特殊功能的 ActiveX 控件。这些控件以 OCX 的形式封装。Wonderware 的 FactorySuite InTrack 组件也为您提供了几种 ActiveX 控件。此外，IndustrialSQL 的 ActiveTrend 允许您在 InTouch 内运行 IndustrialSQL 趋势分析程序（或一个函数子集），当在另一应用程序发生事件时，ActiveEvent 允许您通知 IndustrialSQL 事件子系统。

ActiveX 控件有三个主要组成部分：*属性*、*方法*和*事件*。属性非常类似于您可以修改的变量，例如 Calendar.day、Control.height 等。方法则类似于您可以从容器调用的脚本函数调用。

例如，**Browser.Navigate("http://www.wonderware.com")**, **Engine.start()**。事件通过 ActiveX 容器发生。例如，
Control.click (shift). **FileViewer.DoubleClick (name)** 等等。

InTouch 允许您访问 ActiveX 控件属性、方法和事件。您可以将这些属性与 InTouch 标记名相关联，或者通过编写 InTouch QuickScript 来访问它们。

注意：为使 ActiveX 事件脚本正常工作，为之创建脚本的 ActiveX 控件必须载入内存。如果关闭包含 ActiveX 控件的窗口，则其 ActiveX 事件脚本，或包含与该 ActiveX 控件相关联的脚本函数的任何其它 InTouch QuickScript 将不能正常执行。

您可以在 InTouch 应用程序中使用一个或多个 ActiveX 控件。InTouch 允许您方便地选择 ActiveX 控件、将其粘贴到任何应用程序窗口，以及添加到“**向导/ActiveX 工具栏**”中。您也可以将 ActiveX 事件脚本从一个应用程序导入另一应用程序。

➢ **要在 InTouch 中使用 ActiveX 控件：**

1. 安装您要使用的 ActiveX 控件。
2. 选择并粘贴 ActiveX 控件到 WindowMaker 窗口。
3. 配置 ActiveX 控件的属性并为其指定标记名。
4. 将 ActiveX 事件与 ActiveX 事件脚本关联。
5. 调用 ActiveX 方法，并在 ActiveX 事件脚本或其它 InTouch QuickScript 中设置 ActiveX 控件属性。

您可以在 WindowMaker 中对 ActiveX 控件执行下列编辑操作：

- 调整 ActiveX 控件的大小（如果控制支持大小调整）。
- 复制、剪切、拷贝、粘贴和删除 ActiveX 控件。
- 对 ActiveX 控件使用所有对齐命令（左、右、顶、底、中心点对齐）。
- 将 ActiveX 控件添加到“**向导/ActiveX 工具栏**”。
- 在创建单元时，可随其它对象一起添加 ActiveX 控件。
- 可使用 WindowMaker 菜单命令及相应的工具栏工具来直接修改许多 ActiveX 属性。例如，缩小字体、线颜色、填充色等。

InTouch 不支持下列 ActiveX 控件类型。

- 窗口无关控件
- 单框架布置（分组框）
- 容器
- 数据控件
- 发送对象
- 数组、Blob、对象、变量类型

➢ **要安装或删除 ActiveX 控件：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**向导 / ActiveX 安装**”，或者在应用程序浏览器中，双击“**向导 / ActiveX 安装**”。“**向导 /ActiveX 安装**”对话框出现。

   - 在应用程序浏览器中，您也可以右击“**向导 / ActiveX 安装**”，然后单击“**打开**”。

2. 单击“**ActiveX 控件安装**”标签以激活“**ActiveX 控件安装**”属性页：



3. 在“**已安装的 ActiveX 控件**”列表中，选择您要从应用程序中删除的控件，然后单击“**删除**”。屏幕出现一个要求您确认删除的交互式消息框。

   - 要选择一组控件，单击您要选择的第一个控件，在按住 SHIFT 键的同时，选择所需的最后一个控件。如此将选定介乎其间的所有控件。要选择列表中不连续的多个控件，单击第一个控件，在按住 CTRL 键的同时，单击下一个控件。

4. 单击“**是**”以删除控件。被删除的控件将移至“**可用 ActiveX 控件**”列表中。

注意：控件仅被移除，而没有被删除，但是它将不能再载入内存。也就是说，它将不能再正常工作。

---

5. 要安装 ActiveX 控件，在“**可用 ActiveX 控件**”列表中选择该控件，然后单击“安装”。

注意：仅当控件显示在“**可用 ActiveX 控件**”列表中时，“**安装**”按钮才被激活。

6. 单击“**关闭**”。

➢ **要将 ActiveX 控件放入窗口：**

1. 单击“**向导/ActiveX 工具栏**”上的向导对话工具，“**向导选择**”对话框出现：



2. 在向导列表中，单击“**ActiveX 控件**”类别。所有可用的 ActiveX 控件将出现在显示区域。

3. 选择您要使用的 ActiveX 控件，然后单击“**确定**”，或者双击该控件。对话框关闭，窗口重新出现。

   要在“**向导/ActiveX 工具栏**”中添加 ActiveX 控件，单击“**添加到工具栏**”。一旦将控件添加到“**向导/ActiveX 工具栏**”中，您就可以随时选择该控件并将其粘贴到您打开的窗口中。

注意：可添加到工具栏中的 ActiveX 控件数取决于您的系统资源。

4. 当您返回窗口时，光标将变成弯头符号 。单击窗口中您要粘贴 ActiveX 控件的位置。

5. 双击控件以配置其属性。

有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

➢  **要从工具栏中删除 ActiveX 控件：**

1. 单击“**向导/ActiveX 工具栏**”上的“**向导对话**”工具，“**向导选择**”对话框出现：
2. 单击“**从工具栏中删除**”，“**从工具栏中删除向导**”对话框出现：



3. 选择您要从工具栏中删除的 ActiveX 控件。
4. 单击“**确定**”。

配置 ActiveX 控件

当您将 ActiveX 控件粘贴到 InTouch 窗口时，您必须配置其属性以与 InTouch 相互作用。您必须对每个控件命名，以便可以从 InTouch QuickScript 引用。当您粘贴 ActiveX 控件时，系统会生成一个缺省控件名，例如 Calendar1（此控件名通用于您的 InTouch 应用程序）。

您必须将 ActiveX 控件属性指定给 InTouch 标记名。每种属性类型必须指定给相应的 InTouch 标记名类型。

➢ **要命名 ActiveX 控件：**

1. 将 ActiveX 控件粘贴到 WindowMaker 窗口中。
2. 双击控件，或右击控件，然后单击“**属性**”。控件的相应“**属性**”对话框出现：

注意：每个 ActiveX 控件的“属性”对话框对于控件是唯一的。显示的标签数取决于特定控件的属性。某些 ActiveX 控件可能要求您配置更多的属性。例如某些控件可能要求您配置其颜色和字体，而其它一些控件则可能不需要这些属性。但是，对于所有的 ActiveX 控件，InTouch 均包括三个标签：**控制名**、**属性**和**事件**。例如：



3. 单击“**控件名**”标签，然后在“**控件名**”框中输入 ActiveX 控件的唯一名称。

   您必须为 InTouch 应用程序中使用的每个 ActiveX 控件定义一个唯一的名称。脚本函数使用控件名来识别控件。例如：

```
#Calendar1.day = Tag1;

#Calendar1.year = 1997;
```

注意：缺省条件下，控件名由控件的 ProgID 确定。ProgID 是在计算机中安装 ActiveX 控件时输入系统注册表的名称。当将该控件的实例加入 InTouch 应用程序时，程序从系统注册表读取控件的 ProgID，并在其后附加一个索引号，结果就生成了控件名如 **Calendar1**。

如果您使用缺省控件名，程序会在下列情况下创建控件的一个新实例并为其指定一个唯一名称：

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

更改 ActiveX 控件名

您最好不要更改 ActiveX 控件名，但在某些情况下您可能需要更改控件名。假定您使用缺省控件名，如 **Calendar1**。后来您在删除此 ActiveX 控件后，又重新创建了一个。InTouch 会对控件名作自动递增处理。在本例中，重复的 ActiveX 控件名为 **Calendar2**。为了使您现有的脚本能够识别此新控件，您必须将新控件从 **Calendar2** 重命名为 **Calendar1**。

ActiveX 控件必须具有唯一名称。当您编辑控件名并单击“**确定**”或“**应用**”时，该控件名会与现有的控件名表进行比较。如果表中不存在该名称，则控件名会被更改并保存（如果已经存在该名称，则会出现一个错误消息框）。此时，您必须为控件指定一个唯一名称。.

配置 ActiveX 控件属性

特定 ActiveX 控件的可配置属性由 ActiveX 控件的设计者决定。每个 ActiveX 控件的“**属性**”属性页会显示三列：**属性**、**范围**和**相联标记**。**属性**和**范围**列是只读的。**相联标记**列用于将 InTouch 标记名与“**属性**”列中的相应属性相关联。

注意：当您单击“**范围**”列中的某个项目时，会出现一个箭头，您可以单击它来浏览此项目的可能值列表。列表区域中的项目只能查看而不能更改。

➢ **要配置 ActiveX 控件的属性：**

1. 单击 ActiveX 控件“**属性**”对话框中的“**属性**”标签，激活“**属性**”属性页：



2. 单击“**相联标记**”列中的各个单元格，输入相应属性的标记名。

   如果您输入了一个标记名字典中没有定义的标记名，系统会提示您现在定义一个。

   如果您双击空白单元格，或者单击 ... 按钮，标记浏览器会出现，显示选定标记源的标记名。双击您要使用的标记名，或选择该标记名，然后单击“**确定**”。标记名会自动插入到单元格中。

注意：标记名初始化与创建 ActiveX 控件之间会出现时间间隔问题，由于此时间间隔问题，很难保证标记名的初始值等于 ActiveX 控件中的关联属性值。

为解决此问题，需要建立启动事件与 InTouch QuickScript 之间的联系。在 QuickScript 中，可应用逻辑将标记名值设置为控件属性。这可以通过使用赋值语句来实现。例如：

```
#ThisControl.Property = MyTag;
```

一旦事件发生，QuickScript 即开始执行，控件属性与标记名值将启动同步。因此，属性与标记名之间会出现常规通知。

有关标记浏览器的详细信息，请参阅第 4 章“标记名字典”。

3. 一旦您指定标记名，右击标记名左边的单元格可以选择标记名与相应属性的关联方向（连续双击可在各种关联方向选项之间切换，关联方向选项如下所述）。

   - “**相联标记**”列的每个单元格实际包含两个字段：“关联方向选择”和“标记名项目”。ActiveX 控件决定关联方向，属性类型决定必须使用的标记名类型。

   您可以选择单向或双向关联。但是，如果您选择的关联方向对此属性或标记名无效，控件将自动地作相应改变。例如，如果您选择 ⇦，当标记名的值改变时，其关联属性将相应地改变。取决于属性和标记名之间的潜在联系，下列符号的特定子集将出现。

例如，将属性与可写标记名关联，与将相同属性与只读标记名关联相比，您会看到一个不同的关联集合。选择相应的关联符号，如下所示：

 标记名设置关联属性的值。

 此符号表示属性是只读的，标记名不能改变属性的值。

 属性设置关联标记名的值。

 此符号表示标记名是只读的，属性不能改变标记名的值。

 可从标记名或属性二者设置值（标记名优先）。

 标记名和属性都是只读的。

 标记名可改变属性的值，但属性不能改变标记名的值。属性不能改变标记名的值，是因为属性是非绑定的，或者标记名是只读的。

 属性可以改变标记名的值，但标记名不能改变属性的值。标记名不能改变属性的值，是因为属性是只读的。

4. 单击“**确定**”。

注意：您也可以通过 ActiveX 事件脚本和（或）其它 InTouch QuickScript 来访问或改变属性。所有的 ActiveX 脚本函数用 #（磅）符号注明。访问 ActiveX 属性的有效语法是：

```
#ControlName.PropertyName
```

实例：

```
#Calendar1.Day = 29;
Tag1 = #Calendar1.year;
```

有关详细信息，请参阅“配置 ActiveX 控件”。

注意：在更改 ActiveX 属性页内容时，无论您是在属性页上单击“应用”还是“取消”，所做更改都会立即生效。在控件属性页上更新的属性可能不会反映在 InTouch ActiveX 属性页中。在标签之间来回切换会更新属性页。

InTouch 不支持 "SelectionChange" 事件。

使用 ActiveX 控件方法

ActiveX 控件方法类似于 ActiveX 控件属性。控件方法在运行时 (WindowViewer) 激活。ActiveX 控件方法通过 ActiveX 事件脚本和（或）其它 InTouch QuickScript 进行访问。

注意：为使 ActiveX 事件脚本正常工作，为之创建脚本的 ActiveX 控件必须载入内存。如果关闭包含 ActiveX 控件的窗口，则其 ActiveX 事件脚本，或包含与该 ActiveX 控件相关联的脚本函数的任何其它 InTouch QuickScript 将不能正常执行。

➢ **要使用 ActiveX 方法和（或）属性：**

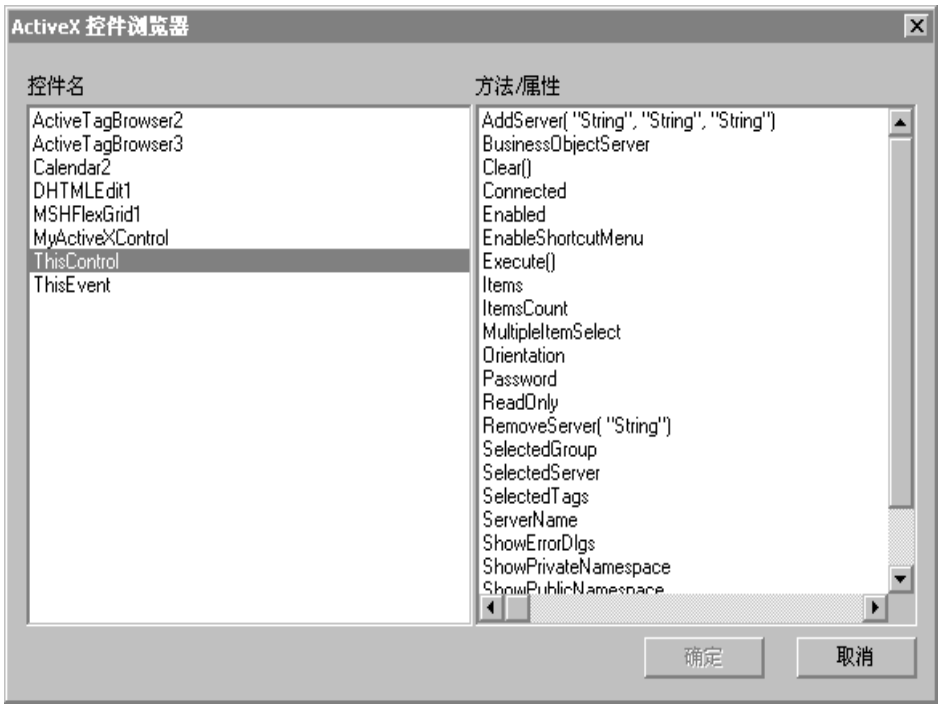
1. 在 ActiveX 控件的“**属性**”对话框中，单击“**事件**”标签以激活“**事件**”属性页：



2. 双击“**脚本**”列中的空白单元格。“**ActiveX 事件脚本**”编辑器出现：



3. 在“**插入**”菜单上，单击“**ActiveX**”。“**ActiveX 控制浏览器**”出现：



4. 在“**控制名**”列表中，选择您要访问其方法和属性的 ActiveX 控件。

应用程序当前使用的所有 ActiveX 控件均将列出。

注意：如果您选择“**ThisControl**”而不是实际的“控制名”，则显示的是应用程序中当前选定的 ActiveX 控件的方法和属性。通过选择“**ThisControl**”而不是实际的控件名，您可以创建通用 ActiveX 事件脚本函数。随后，您可以将这些函数复制和粘贴到任何其它 ActiveX 事件脚本或其它 InTouch QuickScript 中，而不需要在新的脚本中改变控件名。例如：

```
#ThisControl.Navigate  ("http:\\www.wonderware.com");
#ThisControl.Navigate(URL);   { where URL is a
tagname}
```

“**ThisControl**”只能通过 ActiveX 事件脚本，而不能通过任何其它 InTouch QuickScript 来访问。

“**ThisControl**”用于编写可在相同控件类型的任何实例中重复使用的事件脚本的一般引用。当选定“**ThisControl**”时，属性和方法的相同列表显示为该特定控件名的列表。

---

5. 在“**方法/属性**”列表中，选择您要在脚本中使用的方法或属性。

   - 方法是列表中包含括号的项目。例如 **Display()**。

6. 单击“**完成**”。选定的控件名和方法或属性将自动插入您的脚本中。

   - 在所有其它 InTouch QuickScript 类型中，ActiveX 控件的方法和属性也通过“**插入**”菜单来访问的。

使用 ActiveX 控件事件参数

通过设计特定的动作并将其与事件关联，您可以在运行时 (WindowViewer) 执行 ActiveX 控件。例如，如果您的 ActiveX 控件带有错误事件处理器，您可以创建一个 ActiveX 事件 QuickScript，在出错时显示一个包含错误消息的窗口。ActiveX 事件脚本可支持事件动作，您可以将指定的事件脚本与每个事件关联。

➢ **要使用 ActiveX 事件参数：**

1. 双击您要为之创建 ActiveX 事件脚本的 ActiveX 控件。所选 ActiveX 控件的“**属性**”对话框出现。
2. 单击“**事件**”标签以激活“**事件**”属性页：



3. 在“**事件**”列中，选择您要与 ActiveX 事件脚本相关联的事件。
4. 在“**脚本**”列的相应单元格中，为您要创建的 ActiveX 事件脚本输入一个唯一的名称，然后双击此名称，或单击“**确定**”。将出现下面的消息框：



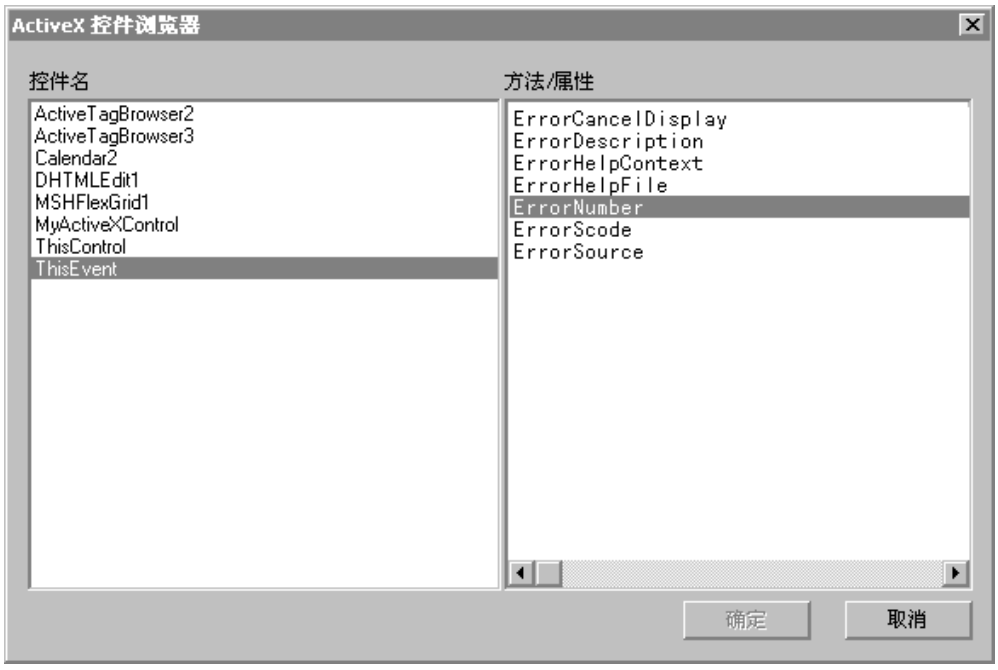
单击“**确定**”。ActiveX 事件脚本编辑器出现，显示您在“**名称**”框中输入的名称（见下面的例子）。如果您双击一个空白的**脚本**单元格，则当 ActiveX 事件脚本编辑器出现时，您必须为 ActiveX 事件脚本输入一个名称。

- 如果您要使用的 ActiveX 事件脚本已经存在，单击 ... 按钮。“**选择 ActiveX 脚本**”对话框出现，列出应用程序中的所有现有 ActiveX 事件脚本。

  有关详细信息，请参阅“重复使用 ActiveX 事件脚本”。



5. 在“**插入**”菜单上，单击“**ActiveX**”。“**ActiveX 控制浏览器**”出现：



6. 在“**控制名**”列表中，选择“**This Event**”以访问所选事件的参数。在本例中，选定的事件是“**Error**”。

注意：“**ThisEvent**”只能通过 ActiveX 事件脚本，而不能通过任何其它 InTouch QuickScript 来访问。您必须选择“**This Event**”来访问 ActiveX 控件的事件参数。

事件参数包含有用的信息。例如，在出现按键事件后，您想要了解这样的问题：“我们都知道按了键，但到底按的是哪个键呢？”如果此信息很重要，则控件开发员可能添加一个事件参数来存储键代码。要访问此参数，您需要选择“**ThisEvent**”，然后选择可能由 **keypressedkeycode** 调用的参数。需要特别注意的是大多数的 ActiveX 事件都不具有参数。

事件不一定会在运行时传递参数。事件参数可以通过使用“**ThisEvent**”关键字来访问。例如：

```
MyActiveXErrorNumber = #ThisEvent.ErrorNumber;
```

此处：# 表示这是一个 ActiveX 脚本函数。“**ThisEvent**”与 ActiveX 控件的“**事件**”属性页中选定的事件相关联，“**ErrorNumber**”是由选定事件传递的参数。

7. 在“**方法/属性**”列表中，选择您要在 ActiveX 事件脚本中使用的方法或属性。

8. 单击“**完成**”。选定的控件名（在本例中为“**This Event**”）及选定的事件参数将自动插入到光标位置所在的脚本中。例如：



注意：ActiveX 事件脚本引用由下列几部分组成：

表示“我是一个 ActiveX 脚本引用”的数字符 (#)。

控件名。

分隔符 (.)，将控件名与所调用的属性或方法分开。

属性、方法或事件参数。

9. 单击“**确定**”保存 ActiveX 事件脚本并关闭脚本编辑器。ActiveX 控件的“**属性**”对话框重新出现。
10. 单击“**确定**”关闭“**属性**”对话框，或者您可以继续创建 ActiveX 事件脚本。

重复使用 ActiveX 事件脚本

只有相同事件和相同类型的 ActiveX 控件才能重复使用 ActiveX 事件脚本。例如，按下鼠标事件可能是数百个 ActiveX 控件中的一个库存事件。但是，在 ActiveX ControlA 中为按下鼠标所写的 ActiveX 事件脚本不能被 ActiveX ControlB 中的按下鼠标事件所重复使用，除非这两个控制是相同的类型。

➢ **要重复使用 ActiveX 事件脚本：**

1. 双击您要为之重复使用现有 ActiveX 事件脚本的 ActiveX 控件。所选 ActiveX 控件的“**属性**”对话框出现。
2. 单击“**事件**”标签以激活“**事件**”属性页：



3. 在相应事件的“**脚本**”列中，单击 ... 按钮。“**选择 ActiveX 脚本**”对话框出现：



此对话框将仅显示为相同类型 ActiveX 控件和相同选定事件所写的 ActiveX 事件脚本。

例如，假定您要为第二个 ActiveX Calendar 控件的“Click”事件创建一个 ActiveX 事件脚本。您已经在应用程序中创建了其它两个 ActiveX 事件脚本，分别为 Click1 和 Click2，Click1 是为一个不同的 ActiveX Calendar 控件的“Click”事件而创建的，而 Click2 是为一个 ActiveX InSQL 趋势控件的“Click”事件而创建的。当您单击 ... 按钮，并且出现“**选择 ActiveX 脚本**”对话框时，它将仅显示 Click1 的脚本，因为它是为相同的 ActiveX 控件和相同事件而创建的。

4. 选择您要使用的 ActiveX 事件脚本，然后单击“**确定**”。

   所选脚本的名称会自动插入到您先前单击 ... 按钮的“**脚本**”单元格中。例如：



5. 单击“**确定**”关闭“**属性**”对话框，或者继续创建 ActiveX 事件脚本。

导入 ActiveX 事件脚本

将 ActiveX 事件脚本从一个 InTouch 应用程序导入您的当前应用程序，可以为您节省大量的开发时间。当您将 ActiveX 事件脚本从一个 InTouch 应用程序移动到另一程序时，您必须使用 WindowMaker“**文件**”菜单中的“**导入**”命令。

注意：当您将 ActiveX 事件脚本从一个应用程序导入另一应用程序时，所有 ActiveX 事件脚本均将导入。此外，为使导入的 ActiveX 事件脚本能够在新应用程序中正常运行，原先为之创建脚本的相同 ActiveX 控制和相同事件也必须用于新应用程序中，并且必须载入到内存中。如果关闭包含 ActiveX 控件的窗口，则其 ActiveX 事件脚本，或包含与该 ActiveX 控件相关联的脚本函数的任何其它 InTouch QuickScript 将不能正常执行。

有关导入脚本的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

自定义运行时环境

与 WindowMaker 一样，您可以设置许多属性来自定义运行时环境 (WindowViewer)。例如，您可以设置闪烁对象的闪烁速度、系统闲置超时及警告值、当 WindowViewer 从其图标或菜单命令启动时自动打开的窗口等等。

设置 WindowViewer 的通用属性

➢ **要设置 WindowViewer 的属性：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**WindowViewer**”，或者在应用程序浏览器的“**配置**”下，双击“**WindowViewer**”。
“**WindowViewer 属性**”对话框出现并显示“**通用**”属性页。

   在应用程序浏览器中，您也可以右击“**WindowViewer**”，然后单击“**打开**”。



- 如果您右击任意对话框中的文本输入框，将出现一个菜单，显示可应用于选定文本的命令。

2. 如果您希望在启动 WindowViewer 时自动启动 FactorySuite Wonderware Logger，请选择“**启动 Wonderware 日志**”。

注意：在 Windows NT 操作系统上，Wonderware Logger 的操作稍有不同。

有关 Wonderware Logger 的详细信息，请参阅您的联机《*FactorySuite 系统管理员指南*》。

3. 如果您希望 WindowViewer 作为一个图标而不是窗口启动，请选择“**启动后只显示图标**”。

   - 仅当您使用 WindowViewer 来为其它 I/O 互连应用程序收集数据时，才可选择此选项。

4. 如果您希望在启动 WindowMaker 时自动关闭 WindowViewer，请选择“**关闭 WindowViewer**”。

   - 如果内存足够，并且您使用快速切换开关在 WindowViewer 和 WindowMaker 之间转换，请不要选择此选项。

     快速切换选项在“**WindowMaker 属性 – 通用**”对话框中选定。

5. 当您选定此选项时，“**WindowMaker 属性/通用**”属性页中的“**转换到 WindowViewer 时关闭**”选项也将自动选定。

6. 如果您希望在从 WindowViewer 转换到 WindowMaker 时自动关闭所有打开的窗口，选择“**关闭所有打开的窗口**”。

   - 选择此选项可以释放系统内存。

7. 如果内存不足，请选择“**总是从磁盘加载窗口**”。

   - 选择此窗口，应用程序窗口将从磁盘加载，并且在关闭时将不会保存到 RAM 内存中。

8. 在“**所需最少可用内存**”框中，输入您要为其它 Windows 应用程序保留的可用内存千字节数。

9. 选择“**优化内存**”以明显加快图形的更新速率。选择此选项也将明显地提高文本字段的更新速率。

   - 如果系统内存不足，请不要启用此选项。

10. 在“**警告**”框中，输入在系统离散标记名 **$InactivityWarning** 置为 1（真）之前无操作员活动（鼠标或击键）的时间（以秒计）。

    - 您可以在条件 QuickScript 中使用 **$InactivityWarning** 来显示一个窗口，警告操作员即将从系统注销。如果操作员在指定的超时范围内，单击鼠标、击键或者使用其它指针设备执行任一动作，则其将不会注销。**$InactivityWarning** 与计时器将重置。

11. 在“**超时**”框中，输入系统离散标记名 **$InactivityTimeout** 置为 1（真）之前无操作员活动（鼠标或击键等）的时间（以秒计）。当 **$InactivityTimeout** 为真时，系统会将登录的操作员用户名视为保留名“None”，并将安全标记名 **$AccessLevel** 置为 0。

    - 您可以在在条件 QuickScript 中使用 **$InactivityTimeout** 来显示一个窗口，告诉操作员他将从应用程序注销。

    “**超时**”功能可独立于“**警告**”功能来使用，但是，为使两个系统标记名正常操作，“**超时**”值必须大于“**警告**”值。

    例如，“**警告**”在 30 秒闲置后变为真，“**超时**”在此后 15 秒（总共 45 秒）闲置后才变为真。

12. 在“**计时间隔**”框中，输入 InTouch 用于检查其内部计时器的速度间隔。

    注意：此设置控制应用程序**运行期间**、窗口**显示期间**、条件**为真/假期间**、键和触动按纽动作**按下期间**，QuickScript 的执行速度。

    在 Windows NT 操作系统中，脚本的执行速度间隔不能超过 10 毫秒，在 Windows 2000 操作系统中不能超过 50 毫秒。

    有关详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

13. 在“**更新时间变量**”框中，输入您希望 WindowViewer 更新时间系统标记名，如 $Msec, $Second, $Minute 等的频率（以毫秒为单位）。

    - 我们建议使用缺省设置 1000 毫秒。但是，您可以输入 0 来阻止更新所有时间变量。

14. 如果您要在 WindowViewer 中选定触控对象时让其发出蜂鸣，请选择“**对象接触时蜂鸣**”。

15. 如果您要加快趋势对象的更新速度，请选择“**更新所有趋势"Fast"**”。

    注意：仅当您完全确认没有对象覆盖您的运行时趋势对象时，才可选择此选项。如果您在有对象覆盖当前趋势对象时选定此选项，它将不能正常绘制。

    ---

16. 如果您希望在每次执行 QuickScript 时在 Wonderware Logger 程序中写入一条消息，请选择“**调试脚本**”。

    - 如果您在“**WindowViewer 属性/窗口配置**”属性页中选择“**调试**”菜单选项，则可以在运行时从 WindowViewer 的“**特别**”菜单打开或关闭此命令。

17. 仅当您使用由 InTouch 第 3.26 版或更早的版本开发的国际应用程序时，才可选择“**使用旧的 SendKeys**”（这是一个旧选项，FactorySuite 已不再使用）。

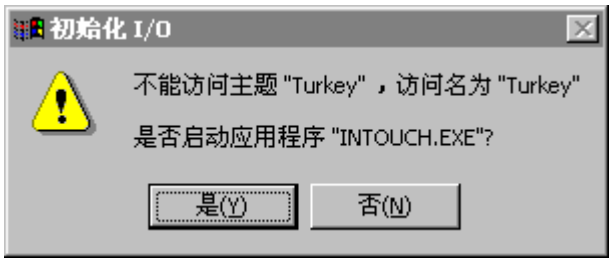
注意：“**使用旧的 SendKeys**”是一个旧选项，它不再用于 FactorySuite。

---

18. 在“**慢**”、“**中等**”和“**快速**”框中，输入您要用于闪烁动画链接的速度（以毫秒为单位）。

19. 在“**I/O 重试开始**”框中，输入 InTouch 尝试重新连接到 I/O 服务器之前等待的秒数（如果 InTouch 第一次就成功连接到 I/O 服务器，则“**I/O 重试开始**”框将失效）。

20. 如果您希望在每次启动 WindowViewer 而您试图与之通讯的服务器没有运行时显示一个对话框，请选择“**启动本地服务器**”选项。例如：



单击“**是**”以启动服务器，单击“**否**”忽略此消息并关闭对话框。

21. 单击“**确定**”保存您的属性设置并关闭对话框。

注意：在修改上述任意参数后，您必须重新启动 WindowViewer 以使所做的更改生效。

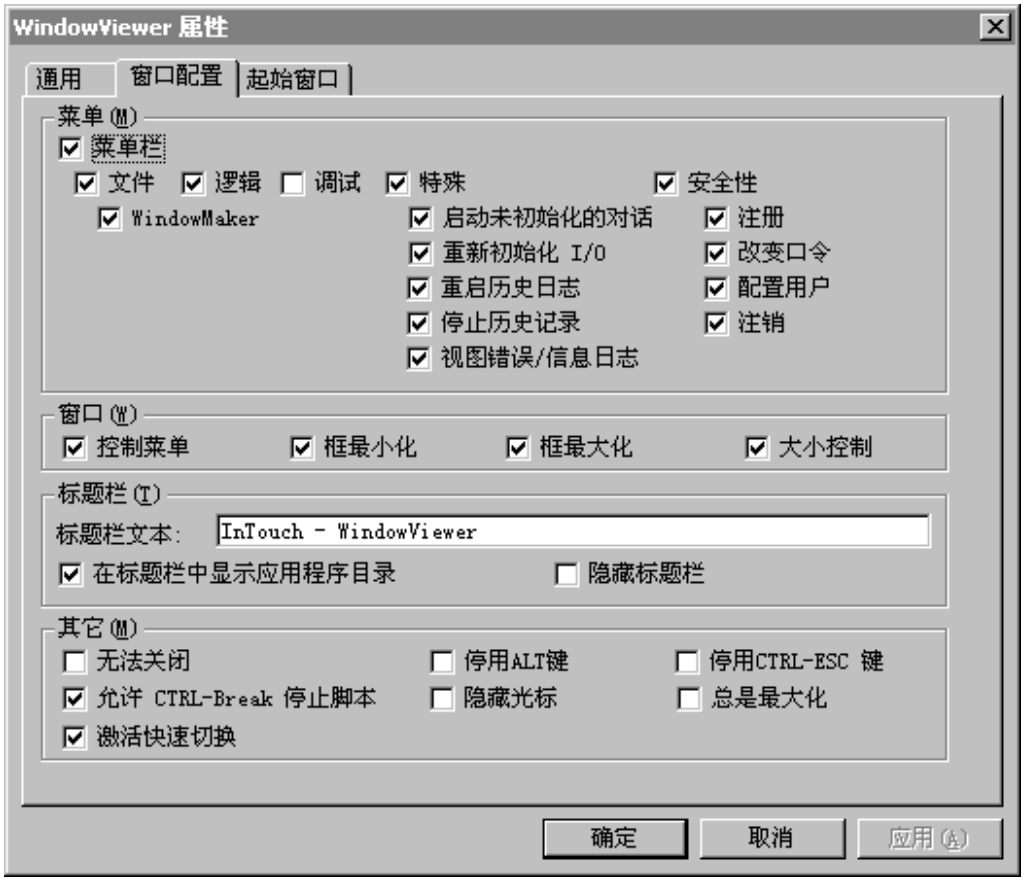
设置 WindowViewer 的窗口配置属性

➢ **要配置 WindowViewer 程序窗口：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**WindowViewer**”，或者在应用程序浏览器的“**配置**”下，双击“**WindowViewer**”。
“**WindowViewer 属性**”对话框出现。

   在应用程序浏览器中，您也可以右击“**WindowViewer**”，然后单击“**打开**”。

2. 单击“**窗口配置**”标签：



如果您右击任意对话框中的文本输入框，将出现一个菜单，显示可应用于选定文本的命令。

3. 在菜单组中，选择您希望在运行时显示的菜单和命令。

   缺省条件下，在运行 WindowViewer 时将显示菜单栏。清除“**菜单栏**”选项以防止菜单栏显示。

4. 如果您想要阻止操作员切换到 WindowMaker 程序，请清除“**WindowMaker**”命令（选择此选项不影响快速切换到 WindowMaker）。

5. 如果您想要防止操作员在运行时启动或停止所有 QuickScripts 的执行，请清除“**逻辑**”菜单。

注意：您可以使用系统标记名 **$LogicRunning** 来允许操作员启动和停止所有的 QuickScript。

如果您选择了“**允许 CTRL-Break 停止脚本**”选项（稍后介绍），则无论“**逻辑**”菜单是否显示，操作员都可以停止所有 QuickScripts 的执行。

您不能停止当前执行的异步 QuickFunction。但是，您可以阻止新的异步 QuickFunction 的执行。

---

6. 仅当您需要“调试”应用程序时，才可选择“**调试**”菜单。

7. 选择运行时可用的“**窗口**”控件。

注意：要隐藏应用程序右上角的关闭按钮（**X** 按钮），您必须清除“**控制菜单**”选项（也称为系统菜单）。

---

8. 在“**标题栏文本**”框中，输入在运行时应用程序的标题栏中出现的标题。例如：

   ABC Company, Paint APP1

注意：如果您使用“推广版许可证”，则不能更改标题栏。

有关 FactorySuite 授权的详细信息，请参阅您的*《FactorySuite 系统管理员指南》*。

---

5. 如果您要在标题栏中包括应用程序目录路径，请选择“**在标题栏中显示应用程序目录**”。例如：

   ABC Company, Paint APP1 - C:\DEMOAPP1

6. 如果您要在运行时隐藏应用程序的标题栏，请选择“**隐藏标题栏**”。

7. 如果您要防止操作员关闭 WindowViewer，请选择“**无法关闭**”。

注意：要隐藏应用程序右上角的关闭 (X) 框，您还必须清除“**控制菜单**”选项（也称为系统菜单）。

8. 如果您希望操作员能够在运行时视情形按 CTRL + BREAK 键序列来终止所有 QuickScripts 的执行，请选择“**允许 CTRL-Break 停止脚本**”。

注意：您不能停止当前执行的异步 QuickFunction。但是，您可以阻止新的异步 QuickFunction 的执行。

9. 如果您想要禁用 ALT 键并阻止操作员使用 ALT + 加速键来执行菜单命令，例如，使用 ALT + FX 来退出应用程序，请选择“**停用 ALT 键**”。

注意：要隐藏应用程序右上角的关闭 (X) 框，您还必须清除“**控制菜单**”选项（也称为系统菜单）。

10. 如果您因为使用触摸屏而要防止在运行时显示光标，请选择“**隐藏光标**”。
11. 如果您要防止操作员通过访问 Windows 的“**开始**”菜单来关闭和（或）切换应用程序，请选择“**停用 CTRL-ESC 键**”。
12. 如果您希望 WindowViewer 程序总是保持最大化，请选择“**总是最大化**”。
13. 单击“**确定**”以保存设置并关闭对话框。

注意：在修改上述任意参数后，您必须重新启动 WindowViewer 以使所做的更改生效。

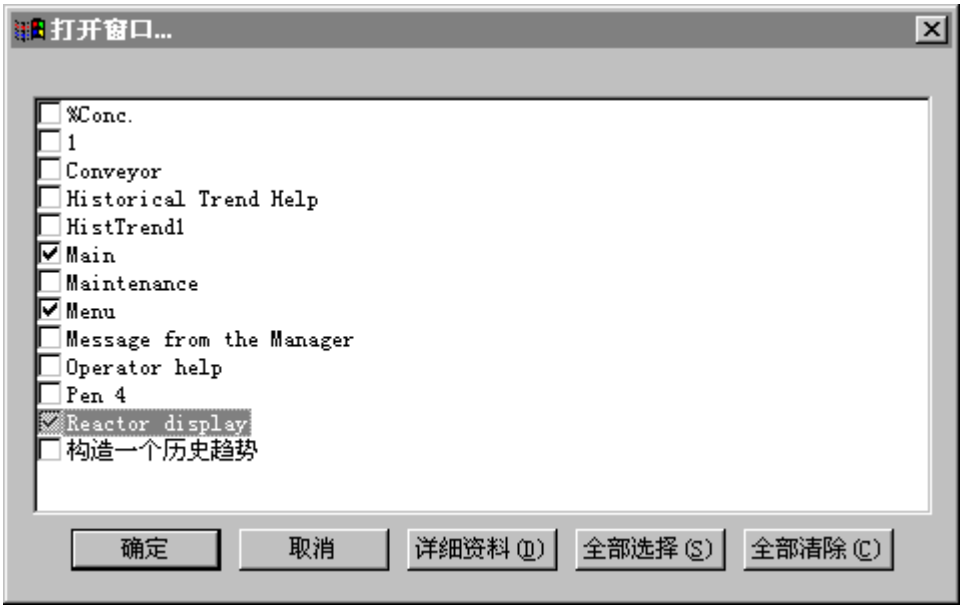
选择 WindowViewer 的起始窗口

➢ **要选择 WindowViewer 的缺省启动窗口：**

1. 在“**特殊**”菜单上，指向“**配置**”，然后单击“**WindowViewer**”，或者在应用程序浏览器的“**配置**”下，双击“**WindowViewer**”。
“**WindowViewer 属性**”对话框出现。

   在应用程序浏览器中，您也可以右击“**WindowViewer**”，然后单击“**打开**”。

2. 单击“**起始窗口**”标签：



3. 选择当 WindowViewer 直接启动时您希望自动打开的窗口。

注意：选择起始窗口不会影响到您用快速切换开关来启动 WindowViewer。当您直接从其图标或菜单命令启动 WindowViewer 时，起始窗口会自动打开。

---

4. 单击“**确定**”。

将 WindowViewer 作为 NT 服务运行

如果您使用 InTouch 7.0（或以上版本），则您可以非常轻松地创建客户机/服务器配置。您可以配置某个节点作为服务器节点。此服务器节点可以存储标记名字典和历史记录数据、执行 InTouch QuickScript、提供报警机制和 I/O 数据。任何客户机节点随后可以从服务器节点检索此信息并显示图形。

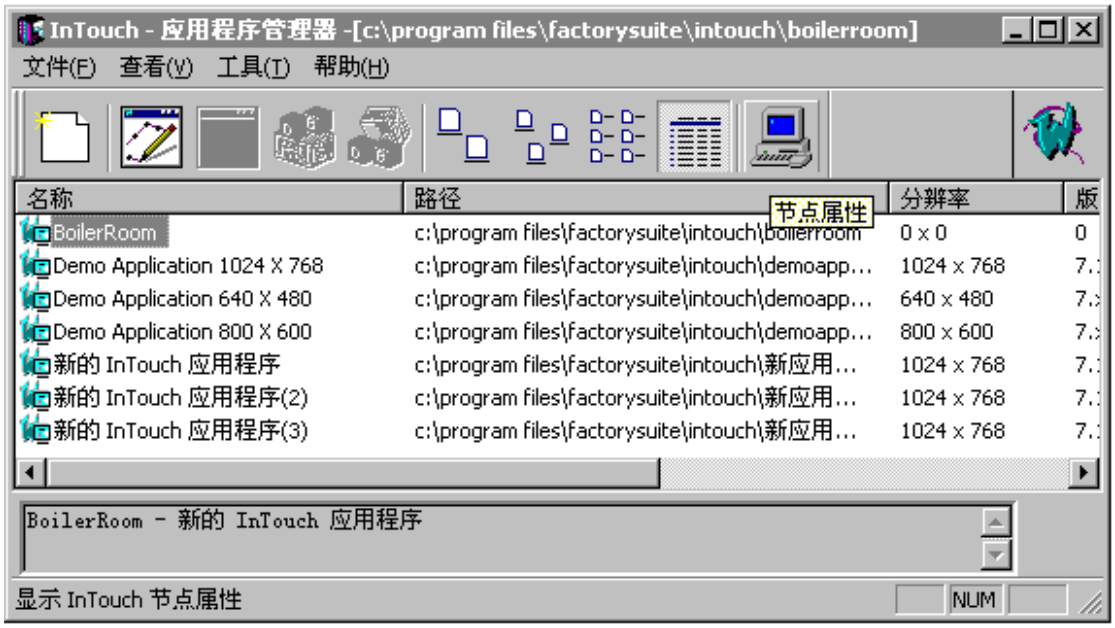
将 WindowViewer 作为 NT 服务运行，您可以充分利用 NT 服务所提供的所有功能。例如，在操作员注销后连续作业以及在系统启动时间自动启动，而不需要操作员干预。这样，就可以实现在无人工作站中启动 WindowViewer，而又不影响 NT 操作系统的安全性。

有关 Windows NT 服务的详细信息，请参阅附录 A“InTouch Windows NT 服务概览”。



➢ **要配置 WindowViewer 作为 NT 服务启动:**

1. 启动 InTouch 程序 (intouch.exe)，“**InTouch 应用程序管理器**”对话框出现：



2. 单击“**节点属性**”工具，或者在“**工具**”菜单上，单击“**节点属性**”。“**节点配置**”对话框出现并显示“**应用程序开发**”属性页：

   要快速访问“**节点属性**”对话框，右击显示窗口的空白区域，然后单击“**节点属性**”。

注意：当在应用程序管理器窗口中选定应用程序时选择“**文件**”菜单上的“**属性**”命令，将可以显示该应用程序的“**属性**”对话框。



3. 选择“**启动 WindowViewer 作为 NT 服务**”选项来配置 WindowViewer，使其自动作为 NT 服务运行。

4. 单击“**确定**”。

注意：如果 WindowViewer 设置为 NT 服务并随后直接启动（从其图标或 Windows 开始菜单等），则在 WindowViewer 显示窗口之前约有 15 秒的延迟。此延迟是因为 WindowViewer 尝试连接到 NT 服务控制管理器而造成的。如果未能连接到控制管理器，WindowViewer 会显示下面的消息框：



如果单击“**是**”，WindowViewer 将作为应用程序，而不是 NT 服务来启动。如果单击“**否**”，用于启动 WindowViewer 的命令将被取消。

假如您通过关闭“**启动 WindowViewer 作为 NT 服务**”选项来停止 WindowViewer 作为 NT 服务运行，则 WindowViewer 服务将被自动卸载。但是，它仍可以作为应用程序运行。

有关 Windows NT 服务的详细信息，请参阅附录 A“InTouch Windows NT 服务概览”。

使用 InTouch 安全性

在应用程序中应用安全性是一个可选项。但是，通过在应用程序中应用安全性，您可以控制操作员允许执行的特定功能，这可以通过将这些功能与内部标记名相链接来实现。此外，当您在应用程序中建立安全性时，可以创建审查跟踪线索，将操作员与其登录到系统后所发生的所有报警/事件相联系。

安全性的基本概念是操作员“登录”到应用程序，然后输入其用户名和口令。因此，您必须为每个操作员配置用户名、口令和访问级别。

注意：Microsoft 操作系统安全性与 InTouch 安全性之间没有关联。

当您创建一个新的应用程序时，缺省条件下，用户名设为“Administrator”并享有访问级别 9999（此级别可访问所有的安全性命令）。一旦您在安全性列表中添加新的用户名并重新启动 WindowMaker 或 WindowViewer，缺省用户名将自动设为“None”而访问级别变为“0”（此级别不允许访问 WindowMaker 和 WindowViewer 中的配置用户命令）。因此，为了以后能够访问安全性用户列表，您必须为系统管理员配置一个访问级别等于或高于 9000 的用户名。

一旦操作员登录到应用程序，操作员的口令和访问级别将与其所访问功能相链接的内部安全标记名进行核对，经验证一致后方能访问任何受保护的功能。

例如，通过指定登录操作员的访问级别必须高于 2000，您可以控制对某窗口的访问或某对象的可见性等。

注意：通过执行 WindowViewer“**特别**”菜单（如果“**特别**”菜单已显示）“**安全性**”下的“**登录**”菜单命令，操作员可以登录到应用程序；或者，您可以用链接到内部安全标记名的触控输入对象来创建一个自定义登录窗口。

有关内部安全标记名的详细信息，请参阅您的联机 *《InTouch 参考指南》*。

- 用于建立应用程序安全性的命令位于 WindowMaker 和 WindowViewer “**特别**”菜单上的“**安全性**”之下。安全性命令用于登录到应用程序或从应用程序注销、更改口令，以及配置有效用户名、口令和访问级别的列表。

使用安全内部标记名

一旦您在应用程序中实施安全性，则可以在按钮、动画链接表达式或 QuickScript 等对象中使用两个安全内部标记名，来控制是否允许登录操作员执行特定的功能：

标记名	类型	有效值	访问
$AccessLevel	整型	0-9999	只读
$Operator	消息型	最大 16 个字符	只读

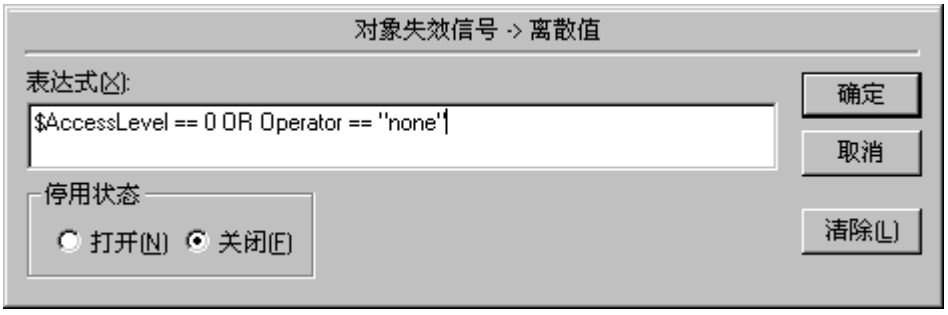
例如，要根据登录用户的访问级别来显示对象，您可以在可见性动画链接表达式中使用下面的语句：

```
$AccessLevel >= 2000;
```

Or, a QuickScript can be bounded by an IF statement:

```
IF $Operator == "DayShift" THEN
    Show "Control Panel Window";
    {and other lines that only execute for the DayShift
       Operator}
ENDIF;
```

通过使用“**失效**”动画连接，您也可以根据内部安全标记名的值来控制对象的触动功能。例如：



通过使用上面的表达式，则在无人登录的情况下，对象或按钮将被禁用，从而防止被擅自篡改。

配置操作员的安全级别

➢ **要配置应用程序操作员的安全级别：**

1. 在“**特别**”菜单上，指向“**安全性**”，然后单击“**配置用户**”。“**配置用户**”对话框出现：



   - 如果您右击任意对话框中的文本输入框，将出现一个菜单，显示可应用于选定文本的命令。

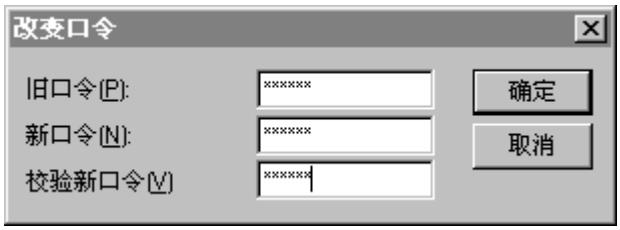
2. 在“**用户名**”字段中，输入您要指定给操作员的名称。
3. 在“**口令**”字段中，输入口令（最长 15 个字符）。
4. 在“**访问级别**”字段中，输入级别值（最低 = 0，最高 = 9999）。
5. 单击“**添加**”以将用户名添加到安全性列表。
6. 要修改现有的用户名，从“**用户名**”列表中选择所需用户名。输入更改内容，单击“**更新**”以接受更改。
7. 要删除用户名，从列表中选定该用户名，然后单击“**删除**”。

注意：“**None**”和“**Administrator**”名称是保留的，只有口令 (Wonderware) 或管理员才可以更改。一旦为应用程序配置了用户名，您应该更改“管理员”用户名的口令，因为它很可能已泄露给系统中的大多数用户。管理员的缺省访问级别为 (9999)，这是最高级别，可以访问包括“配置用户”菜单命令在内的所有内容。

更改安全登录口令

➢ **要更改操作员口令：**

1. 在“**特别**”菜单上，指向“**安全性**”，然后单击“**改变口令**”，“**改变口令**”对话框出现：



   - 如果您右击任意对话框中的文本输入框，将出现一个菜单，显示可应用于选定文本的命令。

2. 在“**旧口令**”字段中，输入旧口令。
3. 在“**新口令**”字段中，输入新口令（最长 16 个字 符）。
4. 在“**校验新口令**”字段中，再次输入新口令。
5. 单击“**确定**”。

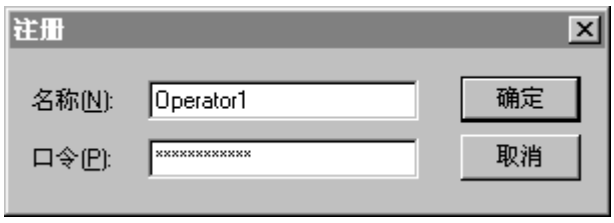
   - 为防止他人看见操作员口令，输入信息在屏幕上显示为星号。

注意：如果您不准备在 WindowViewer 中显示“**特别**”菜单，可以建立一个离散型按钮并将其链接到 **$ChangePassword** 内部标记名，通过将 **$ChangePassword** 标记名设置为 1 来显示“**改变口令**”对话框。一旦对话框显示出来，操作员就可以更改其口令了。

登录到应用程序

➢ **要“登录”到应用程序：**

1. 在“**特别**”菜单上，指向“**安全性**”，然后单击“**注册**”。“**注册**”对话框出现：



如果您右击任意对话框中的文本输入框，将出现一个菜单，显示可应用于选定文本的命令。

2. 在“**名称**”框中，输入您的用户名。
3. 在“**口令**”框中，输入您的口令。
4. 单击“**确定**”。

如果键入的信息不正确或无效，会出现一消息框，指示登录失败。

如果登录成功，“**$AccessLevel**”内部标记名将设置为其预定值（在安全性用户列表中配置）。

创建自定义安全登录窗口

如果 WindowViewer 中没有显示“**特别**”菜单，您可以创建一个自定义登录窗口来让操作员登录到应用程序。

➢ **要创建自定义登录窗口：**

将 **$OperatorEntered** 和 **$PasswordEntered** 系统标记名与用户输入对象相链接，或者在 QuickScript 中使用它们来设置“用户名”和“口令”（这两个标记名是仅用于写操作的内部消息型标记名）。

例如：

```
Set the User Name string into ->     $OperatorEntered
Set the User Password string into -> $PasswordEntered
```

如果输入有效，“**$AccessLevel**”和“**$Operator**”内部标记名将设定为预定值（在安全性用户列表中配置）。

此外，如果 WindowViewer 中没有显示“**特别**”菜单，您可以将“**用户输入 - 离散**”按钮与“**$ChangePassword**”标记名相链接，以显示“**改变口令**”对话框并允许操作员更改其口令。当操作员单击按钮时，“**$ChangePassword**”标记名的值将置为 1 并且“**改变口令**”对话框会出现；当操作员关闭对话框时，系统会将值重置为 0（这是一个仅用于写操作的系统离散型标记名）。

您也可以将“**用户输入-离散**”按钮与“**$ConfigureUsers**”标记名相链接，以允许经授权的操作员通过访问“**配置用户**”对话框来编辑安全用户名列表。当操作员单击按钮时，“**$ConfigureUsers**”标记名的值将置为 1 并且“**配置用户**”对话框会出现；当操作员关闭对话框时，系统会将值重置为 0（这是一个仅用于写操作的系统离散型标记名）。

注销应用程序

➢ **要注销应用程序：**

在“**特别**”菜单上，指向“**安全性**”，然后单击“**注销**”。

当执行此命令时，“用户名”将重置为“None”，访问级别变为“0”。

您可以配置应用程序，使得操作员在未执行任何操作的指定时间后，自动注销该操作员。

有关详细信息，请参阅“自动注销系统”。

自动注销系统

通过使用警告和超时设置，您可以配置应用程序，使得操作员在未执行任何操作的指定时间后，自动注销该操作员。

➢ **要配置闲置活动：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**WindowViewer**”，或者在应用程序浏览器的“**配置**”下，双击“**WindowViewer**”。
“**WindowViewer 属性**”对话框出现并显示“**通用**”属性页。

   在应用程序浏览器中，您也可以右击“**WindowViewer**”，然后单击“**打开**”。



2. 在“**警告**”框中，输入在系统离散标记名 **$InactivityWarning** 置为 1（真）之前无操作员活动（鼠标或击键）的时间（以秒计）。

   - 您可以在条件 QuickScript 中使用 **$InactivityWarning** 来显示一个窗口，警告操作员即将从系统注销。如果操作员在指定的超时范围内，单击鼠标、击键或者使用其它指针设备执行任一动作，则其将不会注销。**$InactivityWarning** 与计时器将重置。

3. 在“**超时**”框中，输入系统离散标记名 **$InactivityTimeout** 置为 1（真）之前无操作员活动（鼠标或击键等）的时间（以秒计）。当 **$InactivityTimeout** 为真时，系统会将登录的操作员用户名视为保留名“None”，并将安全标记名 **$AccessLevel** 置为 0。

   - 您可以在条件 QuickScript 中使用 **$InactivityTimeout** 来显示一个窗口，告诉操作员他已从应用程序注销。

     “**超时**”功能可独立于“**警告**”功能来使用，但是，为使两个系统标记名正常操作，“**超时**”值必须大于“**警告**”值。

     例如，“**警告**”在 30 秒闲置后变为真，“**超时**”在此后 15 秒（总共 45 秒）闲置后才变为真。

4. 单击“**确定**”。

   - 您可以在条件 QuickScript 中使用 **$InactivityWarning** 来显示一个窗口，警告操作员即将从系统注销。如果操作员在指定的超时值到达之前单击鼠标、击键或者使用触摸屏，他将不会被注销（**$InactivityWarning** 和计时器将被重置）。

5. 在“**超时**”框中，输入系统离散标记名 **$InactivityTimeout** 置为 1（真）之前无操作员活动（鼠标或击键等）的时间（以秒计）。当 **$InactivityTimeout** 为真时，系统会将登录操作员的用户名视为保留名“None”，并将安全标记名 **$AccessLevel** 置为 0。

   - 您可以在条件 QuickScript 中使用 **$InactivityTimeout** 来显示一个窗口，告诉操作员他已从应用程序注销。

     “**超时**”功能可独立于“**警告**”功能来使用，但是，为使两个系统标记名正常操作，“**超时**”值必须大于“**警告**”值。

     例如，“**警告**”在 30 秒闲置后变为真，“**超时**”在此后 15 秒（总共 45 秒）闲置后才变为真。

第 3 章

创建分布式应用程序

InTouch 的设计支持独立式和分布式两种应用程序。独立式应用程序指的是每个监视系统只使用一个操作员界面 (OI)，如锅炉部件控制。独立式应用程序一般较容易配置，对网络的要求很低甚至不需要网络，并且只需简单的维护。相反，分布式应用程序要复杂得多，通常包含好几层网络。典型的分布式应用程序具有一个中央开发工作站，采用集中式数据存储并且有一群与中央站相互作用的*客户站*，这些客户站也同时相互作用。

InTouch 提供了许多功能来极大地简化分布式应用程序的建立和维护。其中最强大的功能之一是“网络应用程序开发”(NAD)。NAD 允许多个*客户站*共同拥有一个应用程序的备份而不限制对这个应用程序的开发。InTouch NAD 在应用程序变化时也会向这些客户站提供自动通知。

本章向您介绍如何使用 InTouch 的分布式功能、您可以运用的不同结构以及它们各自的优缺点。

目录

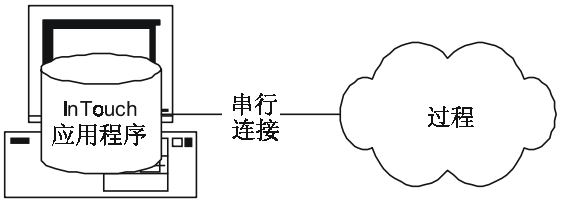
- 网络结构
- 网络应用程序开发 (NAD)
- 配置网络资源
- 网络疑难排解
- InTouch 的公共数据源配置
- InTouch 应用程序的 NAD 配置
- 动态分辨率转换 (DRC)
- 分布式应用程序与时区
- 分布式报警
- 分布式历史记录

网络结构

InTouch 是一个可高度配置的软件包，可以根据您的应用需要，采用各种方式灵活地进行设置。本节概述 InTouch 提供的各种结构及其各自的优缺点，同时还将论及报警和历史记录等各种应用程序组件，这些系统将在后面的章节中向您做更深入的介绍。

独立式应用程序

独立式应用程序对每个监视过程只使用单个操作员界面，通常由一个充当主操作员界面 (PC) 的未连网个人计算机 (PC) 组成。这台计算机通过直接连接，如串行电缆与工业流程进行连接。



开发 / 视图

在这种结构中，计算机上只安装单个 InTouch 应用程序。如果需要进行开发工作，可以直接在这台计算机上开发应用程序。它也可以复制到另一台计算机上，在修改后再复制回原来的计算机。独立式结构虽然不是一种网络结构，但为了完整起见，本章也将向您介绍。

优点

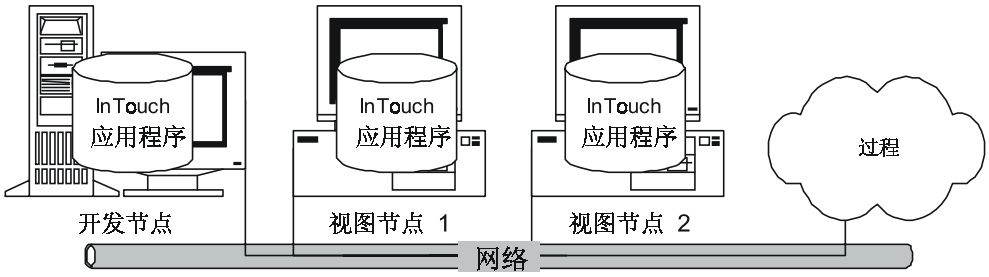
- 易于维护

缺点

- 局限于单个节点

基于客户端的结构

基于客户端的结构是第一种网络结构，它直接从独立式结构发展而来，为运行 WindowViewer 和 NetDDE（视图节点）的每台计算机提供一个 InTouch 应用程序的独特备份。这个应用程序可以安装到每台计算机的硬盘或者网络服务器的一个特别位置上。在下面的实例中，应用程序将在开发节点上开发和测试，然后复制到每个视图节点。



由于每个视图节点具有相同的应用程序备份，每个视图节点对应用程序所引用的任何数据源的访问也必须完全相同。这些数据源可以是 I/O 服务器、SQL 数据库、DOS 文件等。如果使用一个中央数据源，如网络共享式 I/O 服务器，则每个视图节点需与共享服务器保持单独对话，这就增大了网络负载。因而，如果预计网络繁忙的话，最好考虑每个节点上使用单个的 I/O 服务器。

基于客户端的结构在应用程序维护方面有利也有弊。由于每个节点都有各自的应用程序备份，应用程序开发节点对该应用程序的编辑和修改没有限制，此外还可以在不影响运行过程的情况下在该节点进行修改和测试。此方法的缺点在于将修改后的应用程序分配给视图节点将很费力。必须在本地关闭每个视图节点，并且必须在复制新应用程序后重新启动视图。

优点

- 应用程序开发没有限制。
- 由于每个节点可以自足，因而具有内在冗余度。
- 可以使用的视图节点数目没有限制。

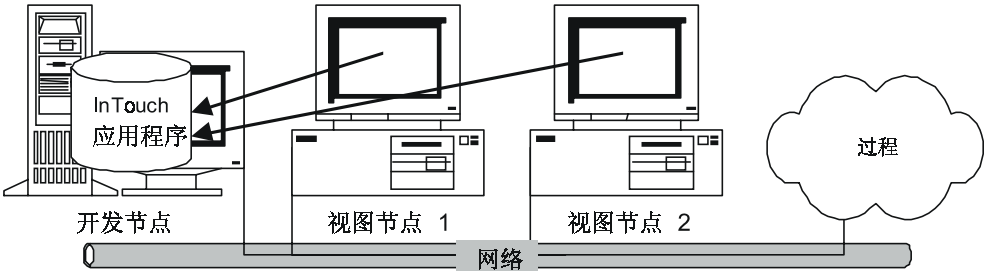
缺点

- 分布应用程序困难
- 所有节点必须对同一数据源进行相同的访问

注意：这种结构已经被 NAD 结构取代，后者将在本章稍后介绍。此处提及这种结构只是为了让您对网络结构有一个完整的概念。

基于服务器的结构

基于服务器的结构允许几个视图节点共享共同的 InTouch 应用程序。在下面的实例中，两个视图节点从开发节点访问同一应用程序。每个视图节点必须在网络软件中创建一个逻辑驱动器，并将其映射到开发节点的共享网络驱动器上。每个视图节点还必须向 InTouch 程序注册共享的应用程序。



与基于客户端的应用程序一样，每个视图节点对应用程序所引用的任何数据源的访问必须完全相同。通过使用脚本组合来获得节点名，并根据此名称来改变每个数据源位置，系统为您提供了几种自定义数据源位置的方法。

有关详细信息，请参阅“InTouch 的公共数据源配置”。

这种网络结构允许应用程序变化及 WindowViewer 重新启动时更新视图节点。

优点

- 只需维护单个应用程序
- 当应用程序变化时视图节点自动更新

缺点

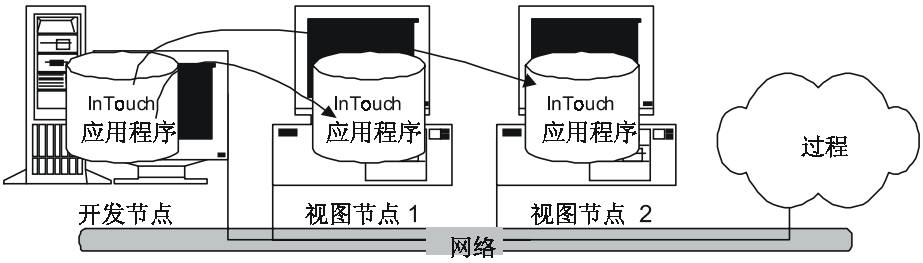
- 应用程序开发受到限制
- 没有冗余度来应付开发站崩溃的情形
- 所有节点必须具有相同的屏幕分辨率

注意：这种结构已经被 NAD 结构取代，后者将在本章稍后介绍。此处提及这种结构只是为了让您对网络结构有一个完整的概念。

网络应用程序开发 (NAD)

网络应用程序开发 (NAD) 是一个结合了基于客户端和基于服务器结构优点的结构。NAD 针对应用程序更新提供自动通知，并将更新后的应用程序自动分发到视图节点。

在 NAD 结构中，应用程序的主备份在中央网络位置维护，与基于服务器的结构一样，每个视图节点可下载该网络应用程序，但它不是从服务器运行应用程序，而是将应用程序复制到用户定义的位置并从那里运行。这就提供了基于客户的冗余优点。在下面的实例中，两个视图节点都在开发节点上注册了主应用程序，但是实际上是从其各自的硬盘中运行。



当视图节点复制并运行一个主应用程序时，它会自动地监视主备份的变化。这些变化由主应用程序目录中的标帜指明。当应用程序开发者在编辑该应用程序的同时使用 WindowMaker“**特别**”菜单上的“**通知客户**”命令，此标帜将手动设置。当此标帜改变时，每个视图节点将由一个用户自定义动作来指定该节点应做出的反应。此动作可以是忽略标帜、自动关闭以及重新启动视图节点，后者将重新载入主应用程序。

注意：如果配置为将历史数据写入主应用程序节点的“应用程序目录”，则所有 NAD 节点会尝试将其历史数据写入主应用程序。为避免出现此问题，需在每个 NAD 节点上配置将历史数据写入本地目录，而**不是**主应用程序节点。

优点

- 只需维护单个应用程序
- 当应用程序改变时自动通知视图节点
- 每个视图节点都可以定义对应用程序更新所作的反应
- 应用程序开发没有限制

缺点

- 如果将一个较大的、复杂的应用程序分发给许多节点，则在最初下载时会明显减慢系统响应时间，更新将被优化。
- 限制了不同应用程序在不同节点上运行的灵活性
- 应用程序传输对较慢的网络或串行连接可能造成困难

配置网络资源

InTouch 为分布式应用提供了许多可配置选项。本节通过实例向您介绍如何设置和使用这些选项来分发 InTouch 应用程序。

配置文件的 UNC 路径

InTouch 支持对应用程序目录输入、配置项目和分布式报警使用通用命名惯例 (UNC)。UNC 允许直接访问基于网络的文件而不需要创建映射驱动器。每个 UNC 地址由三部分组成：节点、共享和路径，组成形式为 **\\Node\Share\Path**。节点指的是包含文件共享的计算机节点名称。共享指的是指定给该计算机上的共享目录的逻辑名称。路径是指有关共享的特定文件的常规 DOS 路径。

注意：在使用 Wonderware 的 SuiteLink 协议时，“**节点名**”限于最长 15 个字符。

在通过 UNC 访问文件之前，您必须在需要访问的计算机上创建文件共享。您可以共享整个驱动器，或只共享它的一个目录，甚至可以定义该共享的安全性。无论采用何种方法，您都必须创建在 UNC 地址中使用的共享名称。有关创建文件共享的详细信息，请参阅您的 Windows 操作系统手册。

一旦创建共享，您就可以使用 UNC 地址在通常需要输入文件路径的地方引用该驱动器。例如， InTouch 程序允许对 InTouch 应用程序位置输入 UNC 路径或标准 DOS 路径 。

例如，假定您有一台网络名为“EngineRm”的计算机并且与根驱动器“C:\”共享，共享名为“Root”。要设置一个到“C:\InTouch.32\Apps\Boiler”应用程序的 UNC 路径，您必须使用下面的 UNC：

\\EngineRm\Root\InTouch.32\Apps\Boiler

如果“Boiler”目录自身已共享为“Boiler”，则 UNC 可以简化为：

\\EnginerRm\Boiler

如果共享的是路径，则不需要指明路径。

注意：如果您需要写入一个由 UNC 地址引用的文件，则此共享必须是一个读/写共享，即使它位于本地节点上。如果您创建一个口令保护的共享，则除非您首先设置网络驱动器映射，否则将不能使用 UNC 来访问共享。您可以使用 Windows 资源管理器从远程节点设置驱动器映射。

Wonderware SuiteLink 通讯协议

Wonderware 的通讯协议 SuiteLink 随 Wonderware FactorySuite 一起提供。Wonderware SuiteLink 使用基于 TCP/IP 的协议。SuiteLink 专门设计用于满足象数据完整性、高吞吐量以及方便诊断这样的工业需要。Microsoft Windows NT 和 Windows 2000 均支持这种通讯标准。

SuiteLink 不是 DDE、FastDDE 或 NetDDE 的替代。客户机与服务器之间的每次连接取决于您的网络状况。SuiteLink 是专门为高速工业应用而设计的，它具有下列特点：

数值时间质量 (VTQ) 在所有传送给识别 VTQ 的客户机的数据值上均会插入一个时间戳和质量指示器。

对数据吞吐量、服务器加载、计算机资源消耗以及网络传输的广泛诊断可以通过 Microsoft Windows NT 操作系统性能监视器来实现。这个特点对于规划和维护分布式工业网络是至关重要的。

应用程序之间可维持一致的高数据容量，而不管这些应用程序是位于单一节点上还是分布在多个节点上。

注意：网络传输协议是使用 Microsoft 标准 Winsock 接口的 TCP/IP。

有关使用 SuiteLink 的详细信息，请参阅第 9 章“I/O 通讯”。

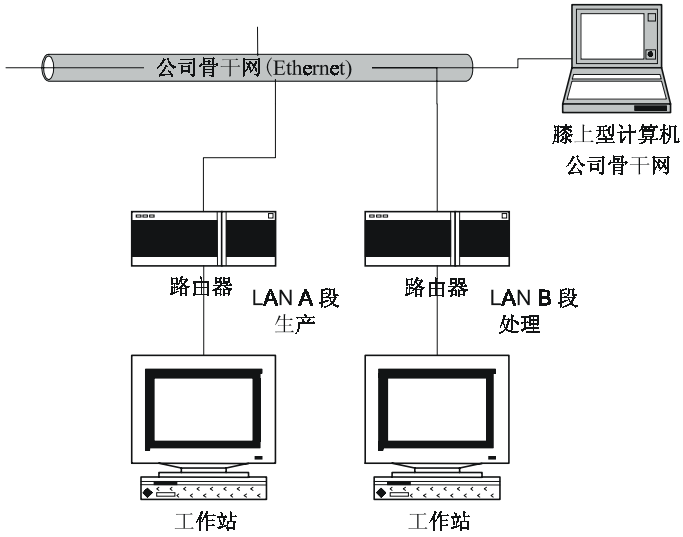
网络疑难排解

在使用 InTouch 的分布式网络特点时必须考虑几个关键的网络结构限制。实施大型 InTouch 节点网络时首先要意识到所有 Ethernet 连接全都相同。很多 MIS 部门使用一种称为“路由器”的设备将其网络分段。这些路由器充当交通警察，调节从一个 Ethernet 网段到另一网段的传输流量。路由器具有“过滤”某些类型的网络流量和地址的能力。

如果您使用的是 NetBEUI，您可能无法连接到另一建筑或不同城市的远程 I/O 服务器，即使网络已经连接妥当。这通常是因为路由器设置为滤除 NetBEUI 流量。一个解决办法是重新设定路由器，使之允许通过这种流量。但是，重新设定路由器有一个缺点，那就是 NetBEUI 传输会流向更多的网络用户，从而增加其它网络段的负担。

注意：Wonderware 的 SuiteLink 协议不能与 NetBEUI 一起使用。SuiteLink 协议要求所引用的所有计算机名称不超过 15 个字符长。

网络分段实例



为避免出现此情况，应改用 TCP/IP 协议并将路由器设置为从一个路由器自动路由到下一个路由器。路由器必须配置为允许双向数据流动（A 到 B 和 B 到 A），以尽可能减少通过路由器进入其它网络段的传输量。通过使用 TCP/IP，您将可以使用很多种网络服务（如 Internet、Frame Relay、ISDN 等）进行远距离对话，从而提高网络的性能。选用 TCP/IP 的另一个关键原因是，TCP/IP 是 PLC 供应商们所选择的协议。

Microsoft 使得切换到 TCP/IP 的工作变得非常容易。Windows NT 服务器随带面向 Windows 2000 或 Windows NT 4.x（或以上版本）的网络客户软件，并具备内置的 TCP/IP 支持。Windows NT 服务器还具有管理动态分配 TCP/IP 地址和名称的能力。

Microsoft 的主要网络产品 Windows NT 服务器也对网络 InTouch 节点提出了一些要求。

例如，每当您试图通过另一个网域检索 InTouch 文件（如历史记录文件）时，您必须提供口令并且必须在该网域中具备帐号才能进行连接。创建这些帐号以及频繁地中断工作来输入口令使得检索操作几乎无法进行。此时，您可以使用两个办法来避免此问题：

1. 将所有计算机分布在同一网域中，但是太多计算机分布在同一网域会引发管理问题。
2. 在网域之间建立“信任关系”，允许一个网域中的计算机共享其它网域的资源，而不用在每次连接时建立附加帐号和提供口令。

可以看出，后一种设置更加可行，而且容易设置和管理。此外，系统还提供其它的高级 NT 网域结构模型，您需要根据类似的约束条件进行实施。

建议您在这种情况下，就整个网络实施的设计和拓扑结构问题请教 Microsoft NT 系统专家。

InTouch 的公共数据源配置

InTouch 允许您使用几种不同的结构来创建应用程序。无论采用何种结构，您都需要考虑应用程序将要访问的数据源及其访问方式。

每种结构都有一个共同特性，那就是应用程序可以在每个视图节点上运行，就如同应用程序实际位于该节点上一样。初看起来这似乎不成问题，但是切记要注意 InTouch 应用程序可能包含的数据源引用。典型的数据源包括访问名、SQL 连接字符串或配方文件。

上述各种数据源都是通过引用地址进行检索的，如配方文件中的 D:\PROCESS\RECIPE.CSV，或 SQL 连接中的 DSN＝PROCDB。这些地址虽然可以被开发应用程序的计算机识别，但当该应用程序被复制并运行于一个没有 D:驱动器或者名为 PROCDB 的注册 ODBC 数据源的视图节点上时，它们可能变得毫无意义。

如果您计划将应用程序分发到多个节点，则需要考虑数据源地址的影响。为此，您可以采用两种基本方式：

1. 在每个视图节点上创建数据源的相同备份，或
2. 对数据源只使用全局地址。

以下章节将介绍这些选项，它们与两个主要的数据源有关：远程数据和文件访问。

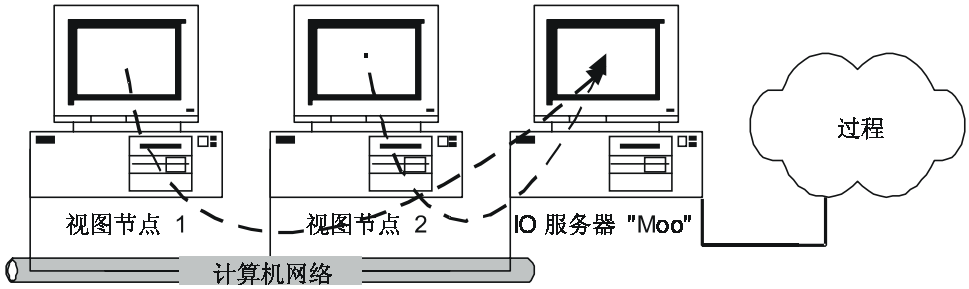
InTouch 访问名

InTouch 使用访问名来引用实时 I/O 数据。每个访问名相当于一个 I/O 地址，可以包含节点、应用程序和主题。在分布式应用程序中，I/O 引用可以设置为指向网络 I/O 服务器的全局地址，或指向本地 I/O 服务器的本地地址。

有关访问名的详细信息，请参阅第 9 章“I/O 通信”。

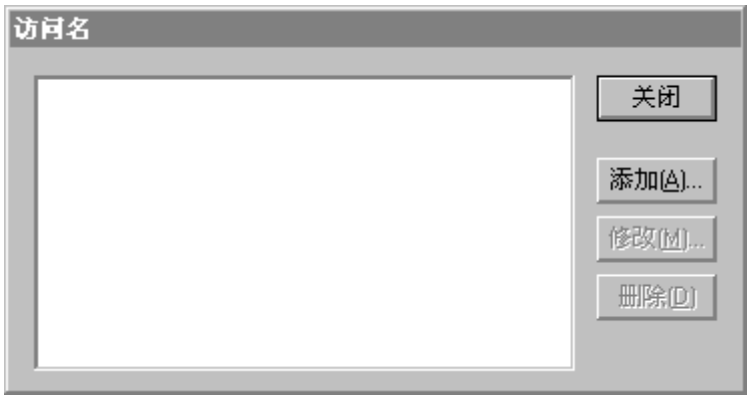
指向 I/O 数据源的全局地址

指向 I/O 数据的全局地址允许所有视图节点共享一个公共的基于网络的 I/O 服务器。这可以防止使用多个 I/O 服务器所需的成本，但容错能力会降低并可能导致整体性能降低。在下面的实例中，两个视图节点分别运行相同应用程序的备份，同时引用相同的 I/O 数据源。由于每个应用程序使用与数据源完全相符的 I/O 地址，因而所有的引用均指向同一个 I/O 服务器。



➢ **要建立此配置：**

1. 在“**特别**”菜单上，单击“**访问名**”，或者在应用程序浏览器的“**配置**”下，双击“**访问名**”。“**访问名**”对话框出现：



2. 单击“**添加**”，“**添加访问名**”对话框出现：



3. 在“**访问名**”框中，输入 **PLC1**。
4. 在“**节点名**”框中，输入 **Moo**（节点名前不能加 \\）。
5. 在“**应用程序名**”框中，输入 **Genius**。
6. 在“**主题名**”框中，输入 **PLC1**。

注意：您可以定义任意“**访问名**”，但“**节点名**”、“**应用程序名**”和“**主题名**”必须引用计算机的 I/O 服务器。为简便起见，建议您使用与主题名相同的名称。

---

7. 选择您所使用的协议。
8. 单击“**确定**”。“**访问名**”对话框重新出现，并在列表中显示新的访问名：

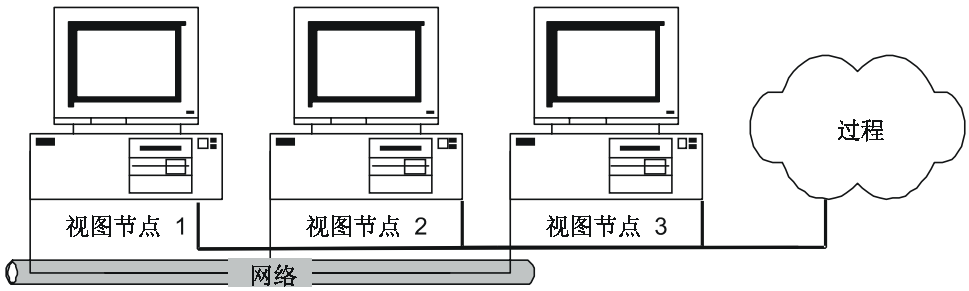


9. 单击“**关闭**”。

指向 I/O 数据源的本地地址

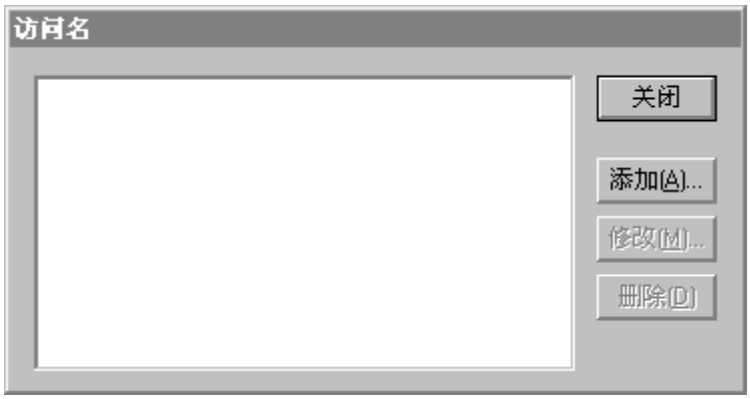
当每个视图节点具有自己的 I/O 服务器时，您可以使用指向 I/O 数据源的本地地址。这种结构能够提供容错操作，当网络中断时每个视图节点可以独立运行。在下面的实例中，两个视图节点分别运行相同应用程序的备份，并引用其各自的 I/O 数据源。由于每个应用程序使用数据源的本地 I/O 地址，因此每个引用都指向本地 I/O 服务器。

使用这种方法会显著地增大过程网络的负载。这是因为必须单独处理每个节点的请求，三个节点将三倍于单个节点产生的流量。



➢ **要建立此配置：**

1. 在“**特别**”菜单上，单击“**访问名**”，或者在应用程序浏览器的“**配置**”下，双击“**访问名**”。“**访问名**”对话框出现：



2. 单击“**添加**”，“**添加访问名**”对话框出现：



3. 在“**访问名**”框中，输入 **PLC1**。
4. 将“**节点名**”框保留空白。
5. 在“**应用程序名**”框中，输入 **Genius**。
6. 在“**主题名**”框中，输入 **PLC1**。

注意：您可以定义任意“**访问名**”，但“**节点名**”、“**应用程序名**”和“**主题名**”必须引用计算机的 I/O 服务器。为简便起见，建议您使用与主题名相同的名称。

---

7. 选择您所使用的协议。
8. 单击“**确定**”。“**访问名**”对话框重新出现，并在列表中显示新的访问名：



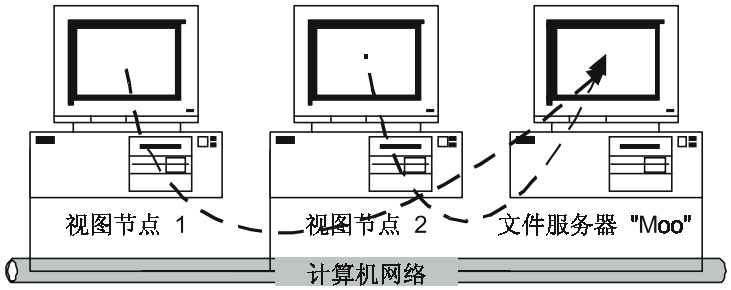
9. 单击“**关闭**”。

文件访问

InTouch 使用 DOS 文件、FAT 或 NTFS 来读取和写入引用数据。某些程序如配方管理器会非常频繁地调用文件。在分布式应用程序中，文件引用可以设置为指向网络文件服务器的全局地址，或指向本地文件的本地地址。

指向文件数据源的全局地址

指向文件数据的全局地址允许所有视图节点共享一个公共的基于网络的文件集合。这就提供了文件的单源维护，但其容错性能要比本地备份差。在下面的实例中，两个视图节点分别运行相同应用程序的备份，同时引用相同的配方文件。由于每个应用程序使用的驱动器号与文件的完全相符网络路径相映射，因此所有引用都指向相同的文件。



- **要建立此配置：**

  建立网络驱动器与包含引用文件的共享路径的映射。在用于检索配方文件的脚本中，输入下列语句：

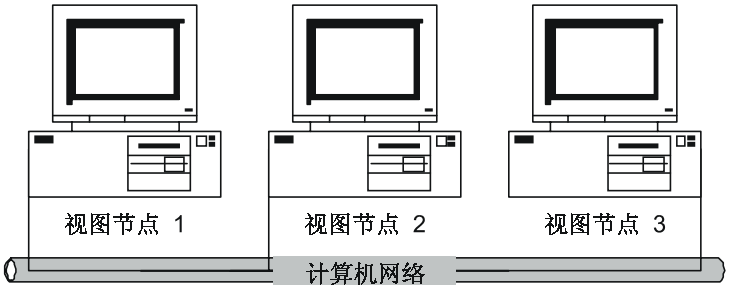
```
RecipeSelectRecipe("G:\Directory\Recipe.CSV",
"review", "RecipeName");
```

  此处，“G:\”是 指向 **\\Moo\Share** 的映射驱动器号。每个视图节点必须单独配置，以映射此相同的“G:\”驱动器 。

  有关详细信息，请参阅“配置文件的 UNC 路径”。

指向文件数据源的本地地址

当每个视图节点包含本地文件时，可以使用文件数据的本地地址。由于网络中断时各个视图节点可以独立运行，因此这种结构可提供容错操作，但它要求文件的任何变化必须复制到所有视图节点。在下面的实例中，两个视图节点分别运行相同应用程序的备份，并且引用本地的配方文件。因为每个应用程序使用文件的本地地址，所以每个引用都指向本地文件。



- **要建立此配置：**

使用本地地址（例如 **C:\Directory**）来直接引用文件。在用于检索配方文件的脚本中，输入下列语句：

```
RecipeSelectRecipe("C:\Directory\Recipe.CSV",
"review", "RecipeName");
```

此处，“**C:**\”是本地驱动器。

每台机器的本地目录“**C:**\directory”下必须保存文件“Recipe.csv”的一个副本。如果修改了文件，文件必须再次复制到每台机器上。由于此配置维护相当困难，因此任何文件访问都只能是“只读”的，并且不允许对本地文件进行修改。

InTouch 应用程序的 NAD 配置

网络应用程序开发 (NAD) 是一个结合基于客户端和基于服务器结构优点的结构。NAD 针对应用程序更新提供自动通知，并将更新后的应用程序自动分发到视图节点。

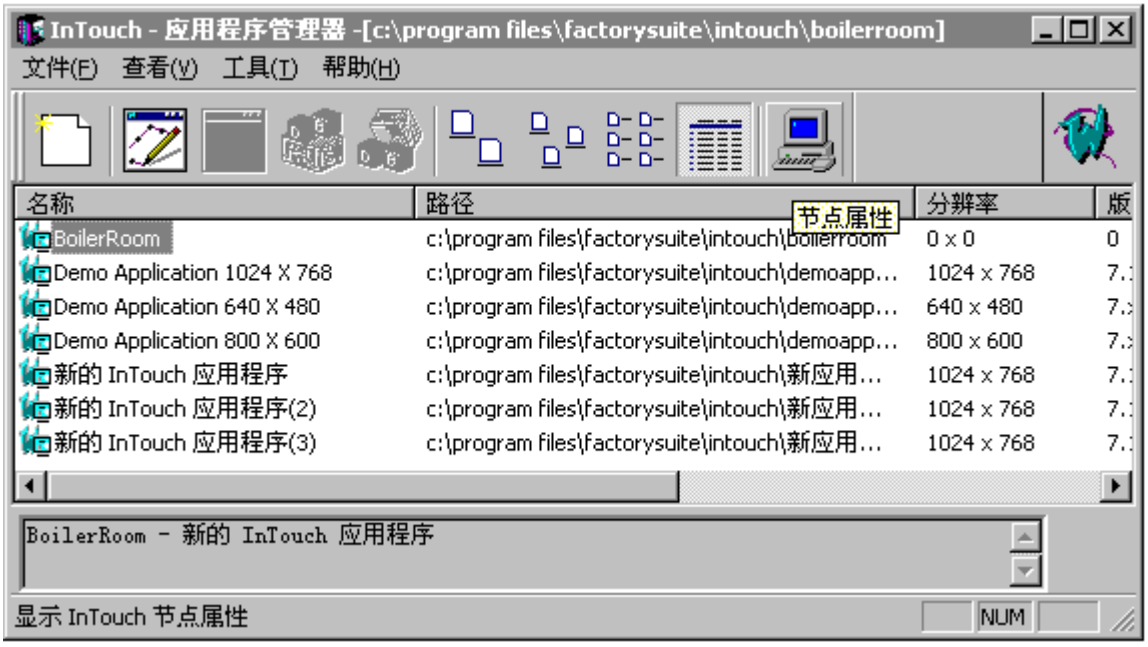
注意：如果您将 WindowViewer 用作 NT 服务，则不能使用 NAD 特性。

如果配置为将历史数据写入主应用程序节点的“应用程序目录”，则所有 NAD 节点会尝试将其历史数据写入主应用程序。为避免出现此问题，需在每个 NAD 节点上配置将历史数据写入本地目录，而**不是**主应用程序节点。



➢ **要对应用程序进行 NAD 配置：**

1. 启动 InTouch 程序 (intouch.exe)。“**InTouch 应用程序管理器**”对话框出现：



2. 单击“**节点属性**”工具，或者在“**工具**”菜单上，单击“**节点属性**”。“**节点配置**”对话框出现并显示“**应用程序开发**”属性页：

   要快速访问“**节点属性**”对话框，右击显示窗口的空白区域，然后单击“**节点属性**”。



注意：“**应用程序开发**”属性页提供了多个选项来指定 NAD 的工作方式。这些设置在各个视图节点，而不是开发节点上配置，这使得可以对每个视图节点进行唯一的配置。

此外，您可以将 WindowViewer 作为 NT 服务运行，此设置保证了操作系统登录和注销时 WindowViewer 的连续操作，比如操作员轮班的情况。通过选择此选项，您还可以在停电或关机并重新开机后自动启动 InTouch。这样，就可以实现无人工作站下的 WindowViewer 自动启动，同时又不影响 NT 操作系统的安全性。但是，如果您将 WindowViewer 作为 NT 服务运行，则不能使用 NAD 特性。

有关运行 Windows NT 服务的详细信息，请参阅附录 A“InTouch Windows NT 服务概览”。

3. 选择“**允许网络应用程序开发**”以启用 NAD，五个“**改变模式**”选项将被激活。每个节点只能选定一个改变模式。

注意：初次复制主应用程序可能要比后续更新花费更长的时间。

4. 在“**本地工作目录**”框中，输入您要 WindowViewer 将主应用程序复制到的目录。

如果节点为开发节点，您可以输入本地目录路径，如：**\InTouch\NAD**。您也可以输入网络远程 UNC 路径，如 **\\node\share\path**。此方法可以为基于文件服务器的网络带来方便，因为在这样的网络中，大多数文件都保存在一个中央位置。如果节点为客户节点（只能运行），则可以使用一个本地目录路径。如果您不指定目录，WindowViewer 会在从中启动 WindowViewer 的目录中自动创建一个名为“NAD”的子目录。

建议您尽量使用本地目录，以避免因网络延迟和故障而影响 WindowViewer 的操作。

警告：请勿使用根目录或指向根目录的 UNC 路径名。在复制主应用程序目录之前，视图节点将删除指定目标应用程序目录中的所有文件和子目录。因此，切勿使用主应用程序目录路径或指向主应用程序目录的 UNC 路径。

该目录只能用作临时目录，并且除了由 NAD 复制的文件之外，不得用于保存任何其它文件。

有关 UNC 路径的详细信息，请参阅“配置文件的 UNC 路径”。

5. 设置“**轮询周期（秒）：**”时间。此值用于设置视图节点轮询开发节点的时间间隔（以秒计），以检查存在的任何变化。

注意：指定此设置时务必小心。如果您将此值设置得过小，WindowViewer 需要花费大量时间来检查主应用程序的变化。这将干扰 WindowViewer 为运行中的应用程序提供服务。

---

6. 在“**改变模式**”组中，选择当主应用程序变化时 WindowViewer 所采取的动作。

忽略变化	让运行时 (WindowViewer) 节点忽略开发节点上的任何变化。
重新启动 WindowViewer	运行时节点复制更新后的主应用程序（如果已配置），然后重新启动 WindowViewer。
提示用户重新启动 WindowViewer	显示一个交互式消息框，通知操作员应用程序已经变化并询问他是否要重新启动 WindowViewer。
装载变化到 WindowViewer	将开发节点上所做的更改动态地载入到 WindowViewer。对于较大规模的更新，此操作可能会影响性能。
提示用户将变化装入 WindowViewer	显示一个交互式消息框，通知操作员应用程序已经变化并询问他是否要将更改动态地载入 WindowViewer。

有关详细信息，请参阅“自定义 NAD 更新功能”。

7. 单击“**确定**”。

自定义 NAD 更新功能

除上面描述的五个更新选项外，NAD 还提供了下面的工具来自定义应用程序的更新行为：

工具	描述
$ApplicationChanged	当主应用程序改变时提供提示。此标记名可用于触发消息，告诉操作员主应用程序已改变。 您也可以在数据改变脚本中使用 **$ApplicationChanged** 系统标记名来建立节点更新通知脚本。此脚本包括启动您自己的对话框或关闭某些过程。随后，您可以使用 **ReloadWindowViewer()** 来初始化更新过程。
ReloadWindowViewer()	当执行此 QuickScript 函数时，它会用更新后的主应用程序动态地更新 WindowViewer，同时又不中断服务。 例如，您可以在按钮上使用此函数，让操作员选择在适当的时间更新应用程序。此函数也可用在 QuickScript 中，以便在指定时间或换班时自动更新。
RestartWindowViewer()	当执行此 QuickScript 函数时，它会自动关闭 WindowViewer，复制更新后的主应用程序（如果已设置），然后重新启动 WindowViewer。 **注意**：此函数已被上述的 **ReloadWindowViwer()** 函数取代。

注意：要使用这些函数，必须在“**节点属性**”对话框中将“**改变模式**”选项设置为“**忽略变化**”。设置此选项可以防止系统免受自定义 NAD 功能的干扰。

有关这些功能的详细信息，请参阅您的联机《***InTouch** 参考指南*》。

手动通知客户应用程序改变

在应用程序开发期间，您可以通过执行 WindowMaker“**特别**”菜单下的“**通知客户**”命令来自动更新 InTouch 客户应用程序。

当您执行此命令时，系统会设置一个标帜，通知所有远程视图节点主应用程序已经改变。这些客户机于是可以根据每个节点所定义的参数，自动启动更新过程。

您也可以使用 InTouch QuickScript 来通知客户机。

有关详细信息，请参阅第 8 章“实时趋势和历史趋势”。

应用程序复制过程

当 WindowViewer 节点复制应用程序时，它会尽可能地在复制过程中保留主应用程序程序的属性（只读、系统、隐藏等）。WindowViewer 同时会复制主应用程序的所有文件和子目录。复制过程不会复制以下文件：***.WVW**、***.DAT**、***.LGH**、***.IDX**、***.LOG**、***.LOK**、***.FSM**、***.STG**、***.DBK**、***.CBK**、***.HBK**、***.KBK**、***.LBK**、***.NBK**、***.OBK**、***.TBK**、***.WBK**、***.XBK**、***.$$$**、**RETENTIV.X**、**RETENTIV.D**、**RETENTIV.A**、**RETENTIV.S**、**RETENTIV.H**、**RETENTIV.T**、**WM.INI**、**DB.INI**、**LINKDEFS.INI**、**TBOX.INI**、**GROUP.DEF** 以及 **ITOCX.CFG**。

注意：WindowViewer 会循环删除目标应用程序目录中的所有文件和子目录，该目录只能用作临时目录（不能用于存放文件）。

应用程序编辑锁定

InTouch 应用程序一次只能由一个开发者进行编辑。为防止多个开发者同时编辑一个应用程序，WindowMaker 在编辑过程中将锁定应用程序。如果您试图将应用程序加载到已锁定的 WindowMaker，则会收到一则消息，指明该应用程序正被另一计算机编辑，您不能对其进行编辑。消息同时会注明当前编辑该应用程序的节点名。

注意：如果 WindowMaker 在加载应用程序时异常关闭，则 appedit.lok 文件会被自动删除。但是，您可以通过从应用程序目录中删除 appedit.lok 文件来手动删除锁定。

动态分辨率转换 (DRC)

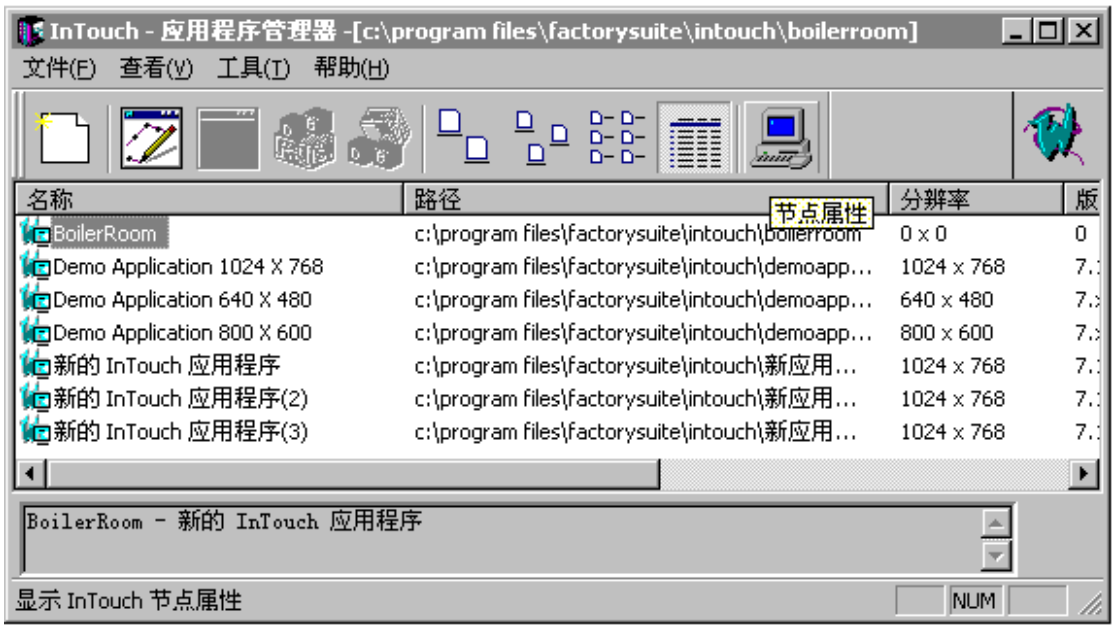
动态分辨率转换 (DRC) 与其它分布式特性一起，可使应用程序免受屏幕分辨率的限制。在 NAD 结构中，InTouch 应用程序在开发节点上创建和维护，然后复制到若干视图节点上。DRC 允许在所有这些节点上查看应用程序，即使它们以不同屏幕分辨率运行。

DRC 使各个视图节点可以根据一系列用户自定义选项，包括自定义分辨率来缩放应用程序。定义分辨率可在 WindowViewer 编译应用程序时进行，而不需要用到 WindowMaker。由于每个视图节点可以使用不同的 DRC 设置，因此每个视图节点必须配置自己的设定值。



➢ **要对应用程序进行 DRC 配置：**

1. 启动 InTouch 程序 (intouch.exe)。“**InTouch 应用程序管理器**”对话框出现：



2. 单击“**节点属性**”工具，或者在“**工具**”菜单上，单击“**节点属性**”。“**节点配置**”对话框出现并显示“**应用程序开发**”属性页：

   要快速访问“**节点属性**”对话框，右击显示窗口的空白区域，然后单击“**节点属性**”。

注意：在应用程序管理器窗口中选定应用程序，然后从“**文件**”菜单上选择“**属性**”命令，将可以显示所选应用程序的“**属性**”对话框。

---

3. 单击“**分辨率**”标签：



4. 如果您希望 WindowViewer 根据您选定的分辨率选项在本机上缩放主应用程序，请选择“**允许 WindowViewer 动态改变分辨率**”（下面将介绍三个分辨率选项）。

注意：如果您不选择此选项，则仅当节点屏幕分辨率与应用程序开发节点的屏幕分辨率完全相同时，WindowViewer 才会运行该应用程序。如果分辨率不同，WindowViewer 会提示操作员运行 WindowMaker，以将应用程序的分辨率转换为节点的分辨率。如果您已经设置指向主应用程序目录的 UNC 路径，则执行此操作时务必小心，因为这会修改原始应用程序。

---

5. 如果您希望 WindowViewer 以开发时的分辨率运行应用程序而忽略节点分辨率，请选择“**使用应用程序分辨率**”。例如，如果应用程序在 640x480 的分辨率下开发，而节点的分辨率为 1024x768，则 WindowViewer 将不会动态地缩放应用程序，而是以 640x480 的分辨率显示该应用程序。

6. 如果您希望 WindowViewer 以节点分辨率运行应用程序而忽略开发应用程序时的分辨率，请选择“**转换为屏幕视频分辨率**”。例如，如果节点以 640x480 的分辨率运行而应用程序是在 1280x1024 的分辨率下开发的，则 WindowViewer 会动态地改变应用程序分辨率（降低），使之适应节点的 640x480 显示（这将是最常用的设置）。

7. 如果您希望 WindowViewer 以“**象素宽度(X)**”和“**象素高度(Y)**”（必须是整型值）框中指定的分辨率运行应用程序，请选择“**自定义分辨率**”。应用程序的分辨率和节点分辨率均将被忽略。例如，如果“**像素宽度 (X)**”和“**像素高度 (Y)**”分别设置为 512 和 384，则应用程序将动态地缩放，以适应节点显示器上 512x384 的象素区域。

8. 单击“**确定**”。

使用多监视器系统

当今市场上供应好几种高级图形适配卡，允许您同时将多个 VGA 监视器与系统相连。这些监视器互相配合，可以创建一个大型的虚拟屏幕。常见的例子是将四个 17 英寸的监视器堆叠成一个立体系统：两个位于下方，两个位于上方。由于每个屏幕的分辨率为 800×600，所以所创建的虚拟屏幕可以达到 1600×1200 象素。

动态分辨率转换 (DRC) 可以轻松地支持这些多监视器系统。您只需选定 DRC 分辨率转换选项，就可以充分利用整个虚拟显示器或仅使用其中的一部分。

如果应用程序设定在偶数个监视器上运行，则当某些对话框跨监视器显示时将会出现问题。比如对话框“**键盘**”，就会因无法访问某些键而产生问题。为解决此问题，InTouch 提供了若干个多监视器配置选项。

- **要在节点上配置多监视器设定：**

1. 使用合适的文本编辑器，例如 Windows 记事本，打开 Windows 目录中的 **WIN.INI** 文件。
2. 定位到 **[InTouch]** 部分并加入下列参数：
3. **[InTouch]**

MultiScreen=1	打开多屏幕模式
MultiScreenWidth=640	单个屏幕的象素宽度
MultiScreenHeight=480	单个屏幕的象素高度

例如，如果您的计算机的分辨率是 2560x1024，分开在两个水平屏幕上显示，您可以输入下列值：

[InTouch]

MultiScreen=1

MultiScreenWidth=1280

MultiScreenHeight=1024

注意：上述输入值会影响到小键盘和 QWERTY 键盘，其它 InTouch 对话框和选项框将不受影响。

分布式应用程序与时区

InTouch 提供的服务可以在跨多个时区的环境下方便地使用应用程序。这些服务由报警和历史系统使用，以允许按发生报警和事件的当地时间来观察数值。例如，如果加州的一名工程师观察堪萨斯一个制造厂上午 10 点发生的一个报警，工程师将看到报警发生时的加州当地时间，即上午 8 点。如果工程师要观察该工厂的历史数据，情形与上面相同。

这些服务的关键是使用 UCT（世界调整时间） - 也称为格林威治平均时间或 GMT - 作为时间参考。每台计算机都配置有当地时间及其所在时区的 UCT 时间偏差。在上面的例子中，位于加州的计算机时区设为 GMT 8 点，而位于堪萨斯的计算机时区设为 GMT 6 点。

InTouch 使用这些 GMT 时间偏差作为检索所有报警和历史数据的基础。在上面的例子中，当位于加州的 InTouch 应用程序从位于堪萨斯的应用程序中收到报警时，它会同时查看两台计算机的 GMT 时间偏差，从而决定报警发生时的加州当地时间。因此，一个在 UCT 6 点时区上午 10:00 发生的报警相当于在 GMT 8 点时区上午 8:00 发生的报警。要使用本功能，每台计算机均必须配置 GMT 时间偏差。

有关详细信息，请参阅第 8 章“历史趋势与夏令时”。

分布式报警

InTouch 支持分布式报警系统，允许显示由本地 InTouch 应用程序及其它网络 InTouch 应用程序的报警系统产生的报警和事件。并可在本地 InTouch 节点或从网络上的远程节点确认报警。

有关设定和配置分布式警报系统的详细信息，请参阅第 7 章“报警/事件”。

分布式历史记录

InTouch 提供的分布式历史记录系统可以从任何 InTouch 5.6（或以上版本）应用程序检索历史数据，甚至是跨网络的历史数据。此系统通过同时从多个历史数据库远程检索数据，扩展了标准 InTouch 历史记录系统的能力。这些数据库称为历史供应器。您最多可以同时显示八个历史供应器，每个供应器对应历史趋势图中的一支笔。

注意：历史供应器可以配置为本地 InTouch 历史或 IndustrialSQL (InSQL) 历史供应器。

有关设定和配置分布式历史系统的详细信息，请参阅第 8 章“实时趋势和历史趋势”。

第 4 章

标记名字典

标记名字典（运行时数据库）是 InTouch 的核心。在运行时，数据库包含数据库中所有项目的当前值。为了创建运行时数据库，InTouch 需要所有已创建变量的信息。每个变量必须指定标记名和变量类型。对某些变量类型 InTouch 需要额外的信息。例如，对于 I/O 类型标记名，InTouch 需要更多的信息以便获得数据并转换它供内部使用。标记名字典就是用来输入这些资料的机制。

本章还将介绍 DBDump 和 DBLoad 这两个数据库实用程序。DBDump 可以将 InTouch 应用程序标记名字典导出为一个文本文件，该文本文件可被其它软件如 Microsoft Excel 访问，以进行修改、保存等。DBLoad 允许将其它软件如 Excel 创建的标记名数据库或另一个 InTouch 应用程序的 DBDump 文件加载到一个现有的 InTouch 应用程序中来。

目录

- 标记名字典特殊功能
- 标记名类型
- 扩展的标记名支持
- 定义新的标记名
- 定义标记名详细资料
- 定义标记名报警条件
- 创建 InTouch SuperTag
- 创建 SuperTag 的其它方法
- 远程标记名引用
- 创建标记名服务器应用程序
- 动态地址引用 (DRA)
- 标记浏览器
- InTouch 交叉引用实用程序
- 打印标记名字典详细资料
- 从字典中删除标记名
- 显示标记名使用计数
- 替换标记名
- 转换占位符标记名
- 定标 I/O 标记名
- 内部系统 $Tagnames
- 标记名点域
- 标记名字典实用程序

标记名字典特殊功能

标记名字典为您提供了下列特殊功能：

功能	描述
标记浏览器	标记浏览器用来选择标记名和标记名**点域**、来自 FactorySuite 应用程序的远程标记名引用和 SuperTag 成员标记名，或任何其它支持标记名字典接口的标记源。
标记名交叉引用	标记名交叉引用允许您交叉引用一个标记名,得到在您的应用程序中使用该标记名的特定位置，包括窗口、脚本、SQL 配置、SPC 触发等。您可以打印交叉引用信息或将它储存到一个文件中。
SuperTag	InTouch 支持 SuperTag 结构，这个结构允许您定义复合标记名类型。您可以定义包含多达 64 个成员标记名和 2 层嵌套的 SuperTag 模板。成员标记名的行为和常规标记名完全一样，它们支持趋势、报警和所有标记名**点域**。
引用	远程标记名引用允许 InTouch 访问 I/O 服务器的数据而不需要在本地标记名字典中创建标记名。远程引用允许您导入和导出窗口或 QuickScript 而不需要从占位符转换标记名。
扩展的标记名支持	InTouch 在其标记名字典中支持多达 61,405 个标记名。（您的系统所支持的标记名数由您的软件许可证所决定）。

标记名类型

在 InTouch 数据库中定义标记名时，您必须根据其用法为每一个标记名指定一种类型。例如，如果一个标记名要读或写来自象 I/O 服务器这样的另一 Windows 应用程序的数值，它就必须是一个 I/O 类型的标记名。下面描述每种 InTouch 标记名类型及其用法。

内存类型标记名

内存类型标记名存在于您的 InTouch 应用程序内。您可以使用它们来生成系统常数和模拟，也可以用它们建立由其它 Windows 程序访问的计算变量。例如：您可以用初始值 3.1416 来定义内存标记名，或者您可以把配方储存在内存标记名组内。在仿真中，您可以用内存标记名控制一个后台 QuickScript 脚本的动作。例如，您可以定义一个在某一动作 QuickScript 中改变的内存标记名“COUNT”，从而为某一过程当前的步骤生成各种动画效果。共有四种内存类型：

内存离散型

内部离散型标记名，值为 0（假，关）或 1（真，开）。

内存整型

值为 -2,147,483,648 到 2,147,483,647 之间的 32 位有符号整数。

内存实型

浮点（小数）内存标记名。浮点数值可以介于 $-3.4e^{38}$ 和 $3.4e^{38}$ 之间。所有浮点计算都按 64 位精度操作，但用 32 位来保存结果。

内存消息型

最长为 131 个字符的文本字符串标记名。

I/O 类型标记名

所有从另一 Windows 程序读取其值或将其值写入另一 Windows 程序的标记名为 I/O 类型标记名。这包括所有来自可编程控制器，过程计算机的输入输出以及来自网络接点的数据。可以通过 Microsoft 动态数据交换 (DDE) 或 Wonderware SuiteLink 通讯协议来访问 I/O 标记名。

当一个读/写 I/O 类型标记名改变时，它将被立即写入远程应用程序。每当链接到远程应用程序的标记名的项改变时，标记名也可以从远程应用程序上更新。缺省时，所有的 I/O 标记名设置为读/写。然而，您可以通过选择“**标记名字典**”对话框中的“只读”选项将其限制为只读。共有四种 I/O 类型：

I/O 离散型

离散型输入/输出标记名，值为 0（假，关）或 1（真，开）。

I/O 整型

值为 -2,147,483,648 到 2,147,483,647 之间的 32 位有符号整数。

I/O 实型

浮点（小数）标记名。浮点值可以介于$\pm 3.4e^{38}$之间。所有浮点计算都按 64 位精度操作，但用 32 位来保存结果。

I/O 消息型

最长为 131 个字符的输入/输出文本字符串标记名。

有关使用 I/O 标记名的详细信息，请参阅第 9 章“I/O 通讯”。

其它类型标记名

您还可以赋予标记名一些特殊标记名类型来执行复合函数，如动态报警显示、历史趋势、监控每一历史趋势笔所绘制的标记名。您还可以用间接标记名类型来将一个标记名赋予多个资源。这些特殊标记名类型描述如下。

组变量

组变量类型用于具有指定报警组的标记名，以生成动态报警显示、磁盘记录和打印记录。您可以用**组变量**类型标记名来生成报警窗口或报警记录，以显示与特定组变量有关的所有报警。您也可以通过赋予**组变量**标记名不同的报警组，来控制所显示或记录的报警。

组变量类型标记名也可以用来创建按钮，通过单击这些按钮，操作员可以在相同的报警窗口中有选择地显示工厂中不同区域的报警。所有与报警组关联的**点域**都可以应用到**组变量**标记名中。

注意：组变量类型在 InTouch 7.11 中仍受支持，但它已失去作用。因为不再使用标准报警系统，所以它不再用于动态地改变报警组。

有关报警的详细信息，请参阅第 7 章“报警/事件”。

历史趋势

当生成一个历史趋势时，InTouch 需要用到**历史趋势**类型的标记名。所有的与历史趋势关联的**点域**都可以运用到**历史趋势**标记名中。

标记 ID

这是与历史趋势对象一起使用的特殊类标记名型。您可以用**标记 ID** 类型的标记名来检索在历史趋势上所绘制的标记名信息。多数情况下，您会用**标记 ID** 标记名来显示赋予指定笔的标记名的名称，或者改变赋予该笔的标记名。

您可以在 QuickScript 中应用一个语句，以将一个新标记名赋给任意历史趋势中的任意笔。例如，您可以在 QuickScript 中应用下面的语句：

```
MyHistTrendTag.Pen1=MyLoggedTag.TagID;
```

当执行此 QuickScript 时，历史趋势中与**历史趋势**标记名 “MyHistTrendTag” 关联的笔 Pen1 将开始为 “MyLoggedTag” 绘制历史记录数据趋势图。

有关运用**标记 ID** 标记名的详细信息，请参阅您的 *《InTouch 参考指南》*。

间接离散、间接模拟、间接消息

间接类型的标记名允许您创建一个窗口，并将该窗口内的标记名重新指定给多个源标记名。例如，您有十五个相同的泵，您想要让操作员监视其报警条件。此时，您不需要创建十五个不同的窗口（每个泵一个），而只需在一个窗口中使用间接标记名，然后用 QuickScript 将间接标记名指向已进入报警状态的那个泵所对应的源标记名，即可调用与各个泵关联的源标记名。您也可以用 Touch 触动按钮 QuickScript，让操作员手动地选择要显示的泵。这种方法可减少开发时间和应用程序大小。

注意：InTouch 也支持间接 SuperTag。所有间接 SuperTag 显示在标记名字典的“**标记名类型**”对话框中。间接 SuperTag 也可用在 InTouch QuickScript 中。

有关详细信息，请参阅“创建间接 SuperTag”。

当您使某个间接标记名等于另一源标记名时，这个间接标记名将与源标记名在所在方面均完全相同，包括点域、脚本等。如果源标记名的值改变，间接标记名会相应地反映这个改变。如果间接标记名的值改变了，源标记也会随之改变。您可以在数据库中将间接标记的值定义为可保持，并在启动时重置为上一次的标记名赋值。

间接标记名通过 **.Name** 点域指定。例如，如果您创建了一个名为 "Setpoint" 的间接模拟标记名，并在 QuickScript 中使用下面的表达式，则 "Setpoint1" 将变成"Setpoint" 值的源标记名：

```
Setpoint.Name = "Setpoint1";  or  Setpoint.Name =
Setpoint1.Name;
```

您也可以连接标记名以便在间接标记名中使用。例如，如果您创建一个如下所示的数据改变 QuickScript，让它在每次标记名 "Number" 值变化时时执行，而间接标记名 "Setpoint" 也相应改变：

```
Number=1;
Setpoint.Name = "Setpoint" + Text(Number, "#" );
```

当此 QuickScript 执行时，模拟标记名 "Number" 的值会转换成文本，并连接到模拟标记名 "Setpoint"，从而使 "Setpoint.Name" 与 "Setpoint1" 相等。间接模拟型标记名可用于整型（整数）和实型（浮点）两种标记名。

当您调用源标记名并且源标记名以引号括起时：

```
Indirect.name= "mytag"
```

或者，源标记名是通过文本字符串与变量连接来定义的：

```
Indirect.Name = "mytag" + Text(Number, "#");
```

在这种情况下，源标记名对于间接标记名而言处于不活动状态。在第二次执行 QuickScript 之前，间接标记名将不能引用源标记名的特性。我们建议您确保所有间接标记名都引用活动状态的标记名。

只有在以下条件下，源标记名才会被激活：

显示在一个打开的应用程序窗口中

用在窗口或动作 QuickScript 中，并且与脚本关联的应用程序窗口已打开

当前正被实时趋势使用

用于报警目的

作为事件来记录并且“事件记录”已启用

记入历史记录并且“历史记录”已启用

用在键、条件、数据改变或应用程序 QuickScript 中

作为 InTouch SPCPro 中的“自动采集”标记名

当前被使用 DDE 的客户端应用程序（如 Microsoft Excel）访问

被使用点存取 (PTACC.DLL) 的本地客户端引用

被指定给间接标记或被其引用（会出现瞬间缓迟，因为脚本必须请求 I/O 点的正确值）

或

为使用 DDE 协议而将 InTouch 访问名配置为“**提示所有项**”

SuperTag

InTouch SuperTag 允许您定义复合标记名类型。您可以定义多达 64 个成员标记名和 2 层嵌套的 SuperTag。成员标记名的行为与常规标记名一样，它们支持趋势、报警和所有标记名**点域**。

有关 SuperTag 的详细信息，请参阅“创建 InTouch SuperTag”。

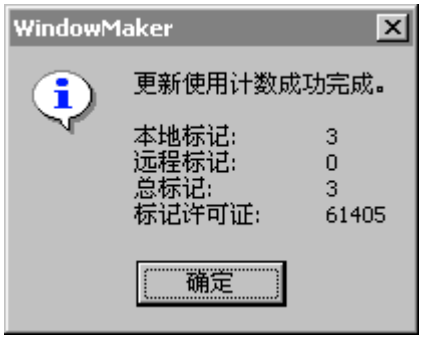
扩展的标记名支持

InTouch 在其标记名字典中支持多达 61,405 个标记名。您的系统所支持的标记名的数量由您的软件许可证决定。

➢ **要决定您的系统所支持的标记名：**

1. 关闭所有的窗口。
2. 在“**特别**”菜单上，单击“**更新使用计数**”。

   一个消息框出现，告诉您更新使用计数需要持续一段时间。此时您可以取消此命令或继续执行。
3. 单击“**是**”继续更新使用计数。
4. 一旦系统完成更新使用计数，将出现下面的对话框：



5. **“标记许可证”**行会显示您的许可证所支持的标记名数。

注意：用在 SPCPro 中的任何“采集标记”、“指示器标记”或“触发器标记”标记名都不计算在内。

6. 单击“**确定**”。

定义新的标记名

标记名最长可达 32 个字符，但必须以一个希腊字符开始（**A-Z** 或 **a-z**）。其余的字符可以是 A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ 和 &。

注意：反斜杠 (\) 只对 SuperTag 有效。

标记名也是自动编号的。例如，如果您输入并保存标记名 R4001，然后单击“**新建**”，则标记名会自动编号为 R4002。如果标记名包含一个字符分隔的数字，它将由 InTouch 找到的第一个整数自动编号。例如，N7-0 将编号为 N7-1。注意只允许正方向的变化。例如，R4002 到 R4003，R4003 到 R4004 等等。

在标记中使用连字符 (-) 时必须小心。连字符在标记名中有效，但它们在表达式或逻辑式中也被用作负号或减号。因而容易引起混淆。

例如，如果您在一个表达式中用 **A=B-C**，您的意思是 **A** 等于 **B** 减 **C**，还是把名为 **B-C** 的标记赋值给一个名为 **A** 的标记呢？InTouch 将假设是后者。为了防止出现这种问题，您应该在标记名与运算符之间插入空格。例如，“**A=B− C**”。

考虑下面的例子：**X-101=FT-101*SP-101**

可以看出，因为这里没有空格，**FT-101** 将乘以 **SP-101** 并将值赋给 **X-101**。

当您第一次访问标记名字典时，会显示内部系统标记名 **$AccessLevel** 的定义。一旦在标记名字典中定义了标记名，当您再次访问它时，将显示最近编辑的标记名定义。

单击“<<”或“>>”可浏览标记名字典中当前存储的标记名定义（当没有要显示的上一个或下一个标记名时，浏览按钮将失效）。

单击“**选择**”以迅速找到一个特殊的标记名定义。“**选择标记**”对话框将在选择模式下出现。

有关标记浏览器的详细信息，请参阅“标记浏览器”。

“**标记名字典**”对话框顶部的选项用于显示各种标记名详细资料对话框，如下所示：

对话框	描述
主要	显示主要标记名字典对话框。在 SuperTag 中，“**主要**”只给出父或根标记名。对父或根标记名所做的任何更改都将改写成员标记名信息。更改后，单击“**保存**”。一个消息框会显示，询问您是否要将根标记名更改应用到成员标记名。
详细	显示所选标记名类型的对应详细资料对话框。
报警	显示所选标记名类型的对应报警配置对话框。
详细和报警	显示所选标记名类型的对应详细和报警配置对话框。
成员	显示 SuperTag 类型标记名的成员详细资料对话框。

- 如果您在标记名字典对话框中右击任意文本输入框，将会出现一个菜单，显示可应用于选定文本的命令。

➢ **要定义一个新标记名：**

1. 在“**特别**”菜单上，单击“**标记名字典**”，或者在应用程序浏览器中，双击“**标记名字典**”。“**标记名字典**”对话框出现。



2. 单击“**新建**”。（“**标记名**”框被清除）。
3. 在“**标记名**”框中，输入新标记名的名称。

   标记名最长可达 32 个字符，但必须以一个希腊字符开始（**A-Z** 或 **a-z**）。其余的字符可以是 A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ 和 &。

   标记名不能使用 **RetVal** 一字。这个字是保留字。如果您试图使用这个字，然后尝试编辑 QuickFunction，将出现一条错误消息 "A variable cannot have that name.Tag exists"（变量不能使用该名称，标记已经存在）。

4. 单击“**类型**”，“**标记类型**”对话框出现：



5. 选择标记名类型，然后单击“**确定**”。所选类型的相应详细资料对话框出现（“详细资料”对话框将在本章稍后介绍）。

   在 TemplateMaker 中创建的任何 SuperTag 名称也将出现在此对话框中，并可用作标记类型。例如，上面的 ColdRoom 和 EvapUnit。如果 SuperTag 不是使用 TemplateMaker 创建的，则会显示名称 "SuperTag"。例如，如果 SuperTag 是在动画链结标记名或表达式输入框、QuickScript 或外部文件中创建，然后再加载到 DBLoad 实用程序。

   有关标记名类型的详细信息，请参阅“标记名类型”。

   有关创建 InTouch SuperTag 的详细信息，请参阅“创建 InTouch SuperTag”。

   注意：如果标记名当前链接到对象或用在 QuickScript 中，则只有在 WindowViewer 没有运行的情况下才能更改其类型。

6. 单击“**组**”将标记名指定给特定报警组。“**报警组**”对话框出现。选择您要指定给标记名的报警组，然后单击“**完成**”。

   注意：如果您没有将标记名指定给特定报警组，则缺省条件下，InTouch 会将它指定给根组 **$System。**

   在创建标记名并将其指定给某个报警组后，如果您不关闭该对话框，则所有之后定义的标记名将被指定给同一报警组（除非您进行更改）。

   有关定义报警组的详细信息，请参阅第 7 章“报警/事件”。

7. 对于 I/O 类型标记名，选择“**只读**”以将标记名限制为在运行时只读。

8. 对于 I/O 类型标记名，选择“**读写**”让标记名在运行时可以读和写。

9. 在“**注释**”框中，输入您要系统储存的有关标记名的任何其它注释（最多 50 个字符）。

   当您第一次访问“**标记名字典**”对话框时，“**注释**”框会显示内部系统标记名 **$AccessLevel** 的缺省注释。您应该删除此注释以防止它与您定义的任何标记名相关联。要删除注释，选择注释然后按 DEL 键。

   有关使用分布式报警系统的详细信息，请参阅第 7 章“报警/事件”。

10. 选择“**记录数据**”，则每当工程单位值变化超过指定的**记录死区**值时，程序会在运行时将标记名写入历史记录文件。否则，程序将缺省每小时记录一次而不管值的变化。

注意：为了实际记录标记名，必须通过“**特别**”菜单上的“**配置历史记录**”命令，来启动历史记录。

如果您决定随后清除此选项以不再记录标记名，则先前记录的标记名数据将无法访问。此外，如果在 WindowViewer 运行时您在 WindowMaker 中更改记录，则所做更改在 WindowViewer 重新启动之前不会生效。

11. 选择“**记录事件**”，记录由操作员、I/O、QuickScript 或系统对标记名所做的所有数据值更改。

- 当您定义标记名来监控事件时，每次标记名值改变，事件消息都会记入报警系统。事件消息会记录数据值如何改变。例如，是由操作员、I/O、脚本还是由系统启动数值改变。

当您选择“**记录事件**”时，“**优先级**”字段将被激活。您所输入的**优先级**值决定了标记名的事件优先级。此字段中的有效输入值是 1 到 999，1 是最高优先级，而 999 是最低优先级。

有关事件和优先级的详细信息，请参阅第 7 章“报警/事件”。

12. 选择“**保留值**”，则每次 WindowViewer 退出时，都会保留标记名的当前值。每次 **WindowViewer** 重新启动时，此值将用作标记名的初始值。

注意：当 WindowViewer 重新启动时，保留值不会被写入 I/O 设备。当 I/O 服务器首次扫描设备时，I/O 值会被更新。

- 当 WindowViewer 正在运行时，将不能选择或清除新的或现有标记名的保留值。当选择此选项时，标记名的初始值会不断更新以反映标记名的当前值。当 WindowViewer 退出时，初始值将设为最后一次保留值。如果随后清除这个选项，标记名的初始值将设定为最后的保留值。

13. 如果您要保留操作员对标记名的报警限字段值所做的任何更改，请选择“**保留参数**”。此值将用作 WindowViewer 重新启动后的报警初始值。

注意：由于所做更改会被立即记录，我们强烈推荐您只对不经常更改的值选择上述两个保留选项。

14. 定义标记名类型的详细资料，然后单击“**关闭**”。

定义标记名详细资料

最初显示的“**标记名字典**”对话框用于输入基本的标记名信息。许多采样点，尤其是输入和输出，需要很多的详细资料以便进行适当的操作。每个指定类型的标记名都有一个专门的详细资料对话框，可用于定义标记名类型的详细资料。

大多标记名类型都有自己的特定详细资料对话框和报警条件对话框。缺省条件下，当您选择标记名类型时，会显示其对应的详细资料对话框。

一旦完成基本标记名的定义，您需要定义标记名和报警状态（如有需要）的详细资料。下面的章节向您介绍定义每种标记名类型的详细资料时必须遵循的步骤。

定义内存离散标记名详细资料

内存离散型标记名存在于您的 InTouch 应用程序之内。当需要一个数值为 0（假，关）或 1（真，开）的内部标记名时，可以定义一个**内存离散**型标记名。

➢ **要定义内存离散型标记名的详细资料：**

1. 选择“**内存离散**”作为标记名类型，出现下面的详细资料对话框。

   如果没有出现，单击“**标记名字典**”对话框顶部的“**详细资料**”。



2. 单击首次加载运行时数据库时，您要储存在标记名中的**初始值**。
3. 如果您要为该标记名定义一个离散报警状态，在该标记名的值等于 1（开，真）时，其状态为“开”，请在“**开消息**”框中输入您要在报警窗口的数值/极限字段中显示的消息。
4. 如果您要为该标记名定义一个离散报警状态，在该标记名的值等于 0（关，假）时，其状态为“开”，请在“**关消息**”框中输入您要在报警窗口的数值/极限域中显示的消息。
5. 如果您要为标记名定义报警条件，单击“**标记名字典**”对话框顶部的“**报警**”或“**详细和报警**”，以显示您所定义的标记名类型的相应报警条件对话框。

   有关报警的详细信息，请参阅第 7 章“报警/事件”。

6. 在定义完标记名后，单击“**关闭**”以保存您的标记名定义并关闭标记名对话框。

定义内存模拟标记名详细资料

内存模拟型标记名存在于您的 InTouch 应用程序内。有两种内存模拟类型：**内存整型**和**内存实型。**当您需要一个介于-2、147、483、648 和 2、147、483、647 之间的 32 位有符号整数的内部的标记名时，就可以定义一个**内存整型**标记名。

当您需要一个浮点值介于$-3.4e^{38}$ 和 $3.4e^{38}$ 之间的内部的标记名时，可以定义一个**内存实型**标记名。所有浮点计算都按 64 位精度操作，但用 32 位来保存结果。

➢ **要定义内存模拟标记名的详细资料：**

1. 选择**内存整型**或**内存实型**作为标记名类型，出现下面的详细资料对话框。

   如果没有出现，单击“**标记名字典**”对话框顶部的“**详细资料**”。



2. 在“**初始值**”框中，输入首次加载运行时数据库时，您要存储在标记名中的值。
3. 在“**最小值**”框中，输入标记名的最小值。此值将用于历史趋势图表、I/O 及 **.Min EU** 标记名点域中。
4. 在“**最大值**”框中，输入标记名的最大值。此值将用于历史趋势图表、I/O 及 **.Max EU** 标记名点域中。
5. 在“**工程单位**”框中，输入您要用于标记名的工程单位的标签。
6. 在“**死区**”框中，输入数据库更新前标记名的工程单位必须变更的数量。
7. 在“**记录死区**”框中，输入标记名记入历史记录文件前标记名的工程单位必须变更的数量。缺省值零表示每次变更都会被记录。

注意：如果您要在标记名的工程单位改变超过**记录死区值**时，将标记名记录到磁盘中，则必须为标记名选择“**记录数据**”。

如果在 WindowViewer 运行时改变**记录死区**值，则在停止并重新启动历史记录前，您所做的更改不会生效。

有关历史记录的详细信息，请参阅第 8 章“实时趋势和历史趋”。

8. 如果您要为标记名定义报警条件，可以单击“**标记名字典**”对话框顶部的“**报警**”或“**详细和报警**”，以显示您所定义的标记名类型的相应报警条件对话框。

   有关报警条件的详细信息，请参阅“定义标记名报警条件”。

9. 在定义完标记名后，单击“**关闭**”以保存您的标记名定义并关闭标记名对话框。

定义内存消息型标记名详细资料

内存消息型标记名存在于您的 InTouch 应用程序内。当需要一个长达 131 个字符的内部文本字符串标记名时，您就可以定义一个**内存消息**型标记名。

➢ **要定义内存消息型标记名的详细资料：**

1. 选择**内存消息**作为您的标记名类型，出现下面的详细资料对话框。

   如果没有出现，单击“**标记名字典**”对话框顶部的“**详细资料**”。



2. 在“**最大长度**”框中，输入消息型标记名所允许的最大字符数（InTouch 允许最大 131 个字符，并将其作为缺省值显示）。
3. 在“**初始值**”框中，输入 WindowViewer 最初启动时标记名所显示的文本字符串。
4. 在定义完标记名后，单击“**关闭**”以保存您的标记名定义并关闭标记名对话框。

定义 I/O 离散型标记名详细资料

所有从另一 Windows 程序读取其值或将其值写入另一 Windows 程序的标记名均为 I/O 类型标记名。这包括所有可编程控制器、过程计算机、其它 Windows 程序以及网络节点数据的输入和输出。

当需要一个值为 0（假，关）或 1（真，开）的 I/O 标记名时，您就可以定义一个 **I/O 离散**型标记名。

➢ **要定义 I/O 离散型标记名的详细资料：**

1. 选择 **I/O 离散型**作为标记名类型，出现下面的详细资料对话框。

   如果没有出现，单击“**标记名字典**”对话框顶部的“**详细资料**”。



2. 单击首次加载运行时数据库时，您要存储在标记名中的**初始值**。（**关**等于 **0**，**开**等于 **1**）。此值不会被写入 I/O 设备。

3. 单击更新运行时数据库时，您要运用到标记名值的**输入转换**：

   直接 I/O 输入值从服务程序直接读入而不加改变。

   取反 当从服务程序读入时，I/O 输入值被反转。例如，假如 I/O 输入值在服务程序中是 0，InTouch 将自动反转、存储并显示为 1。

4. 如果您要为该标记名定义一个离散报警状态，在该标记名的值等于 1（开，真）时，其状态为“开”，请在“**开消息**”框中输入您要在报警窗口的数值/极限域中显示的消息。

5. 如果您要为该标记名定义一个离散报警状态，在该标记名的值等于 0（关，假）时，其状态为“开”，请在“**关消息**”框中输入您要在报警窗口的数值/极限域中显示的消息。

6. 单击“**访问名**”，定义或选择您要赋给标记名的访问名（如果此按钮右边已经显示访问名，并且您没有定义或选择不同的访问名，则该访问名将被赋给标记名）。

   有关访问名的详细信息，请参阅第 9 章“I/O 通信”。

7. 在“**项目**”框中，为标记名将从中读/写其值的服务程序中的数据点输入有效的项目名。例如，如果您要从一个 PLC 寄存器中读值，则可以输入该寄存器的有效标识作为项目名。

   项目名会被自动编号。例如，如果您输入并存储项目名 R4001，然后单击“**新建**”（定义一个新标记名），项目名将自动编号为 R4002。如果项目名包含一个字符分隔的数字，它将自动编号为 InTouch 找到的第一个整数。例如，N7-0 将编号为 N7-1。注意只允许正方向的变化。例如，R4002 到 R4003，R4003 到 R4004 等等。

8. 如果您要为项目名使用该标记名，选择“**使用标记名作为项目名**”。

9. 如果您要为标记名定义报警条件，单击“**标记名字典**”对话框顶部的“**报警**”或“**详细和报警**”，以显示您所定义的标记名类型的相应报警条件对话框。

有关报警条件的详细信息，请参阅“定义标记名报警条件”。

10. 在定义完标记名后，单击“**关闭**”以保存您的标记名定义并关闭标记名对话框。

定义 I/O 模拟标记名详细资料

所有从另一 Windows 程序读取其值或将其值写入另一 Windows 程序的标记名为 I/O 类型标记名。这包括所有可编程控制器、过程计算机、其它 Windows 程序以及网络节点数据的输入和输出。 有两种内存模拟类型:**I/O 整型** 和 **I/O 实型**。

当需要一个包含介于-2、147、483、648 和 2、147、483、647 之间的 32 位有符号整数的 I/O 标记名时，您可以定义一个 **I/O 整型**标记名。

当需要一个浮点值介于-3.4e38 和 3.4e38 之间的 I/O 标记名时，您就可以定义一个 **I/O 实数**型标记名。所有浮点计算都按 64 位精度操作，但用 32 位来保存结果。

➢ **要定义 I/O 模拟标记名的详细资料:**

1. 选择 **I/O 整型** 或 **I/O 实型**作为标记名类型，出现下面的详细资料对话框。

   ☞ 如果没有出现，单击“**标记名字典**”对话框顶部的“**详细资料**”。



2. 在“**初始值**”框中，输入首次加载运行时数据库时，您要储存在标记名中的值。此值不会被写入 I/O 设备。
3. 在“**死区**”框中，输入数据库更新前，标记名的工程单位可以变更的数量。
4. 在“**最小工程单位**”框中，输入当收到最小的原始计数值时，标记名的工程单位值。
5. 在“**最小原始数据**”框中，输入原始 I/O 整型值低嵌位的最小值。
6. 在“**最大工程单位**”框中，输入当收到最大的原始计数值时，标记名的工程单位值。
7. 在“**最大原始数据**”框中，输入原始 I/O 整型值高嵌位的最大值。

   ☞ 您可以用**最小工程单位**、**最小原始数据**、**最大工程单位**和**最大原始数据**值来定标 I/O 标记名。

   有关定标标记名的详细信息，请参阅“定标 I/O 标记名”。

8. 在“**工程单位**”框中，输入您要用于标记名的工程单位的标签。
9. 选择计算工程单位时，数据库用于定标原始计数的**转换**类型，如下所示：

   如果您选择“**线性**”，系统将使用端点之间的线性插值来计算结果。输入的线性换算法则是:

   ```
   EUValue = (RawValue - MinRaw) * ((MaxEU - MinEU) /
   (MaxRaw - MinRaw)) + MinEU
   ```

   输出的线性换算法则是:

   ```
   RawValue = (EUValue - MinEU) * ((MaxRaw - MinRaw) /
   (MaxEU - MinEU)) + MinRaw
   ```

   如果您选择**平方根**，原始数值通常用来插补。这对调整从非线性设备如压力传感器输入的数据非常有用。输入的平方根换算法则是：

   ```
   EUValue = sqrt(RawValue - MinRaw) * ((MaxEU - MinEU) /
   sqrt(MaxRaw - MinRaw)) + MinEU
   ```

   输入的平方根换算法则是：

   ```
   RawValue = square((EUValue - MinEU) * (sqrt(MaxRaw –
   MinRaw) / (MaxEU -MinEU))) + MinRaw
   ```

10. 单击“**访问名**”，定义或选择您要赋给标记名的访问名（如果此按钮右边已经显示访问名，并且您没有定义或选择不同的访问名，则该访问名将被赋给标记名）。

    有关访问名的详细信息，请参阅第 9 章“I/O 通信”。

11. 在“**项目**”框中，为标记名将从中读/写其值的服务程序中的数据点输入有效的项目名。例如，如果您要从一个 PLC 寄存器中读值，则可以输入该寄存器的有效标识作为项目名。

    - 项目名会被自动编号。例如，如果您输入并存储项目名 R4001，然后单击“**新建**”（定义一个新标记名），项目名将自动编号为 R4002。如果项目名包含一个字符分隔的数字，它将自动编号为 InTouch 找到的第一个整数。例如，N7-0 将编号为 N7-1。注意只允许正方向的变化。例如，R4002 到 R4003，R4003 到 R4004 等等。

12. 如果您要为项目名使用该标记名，选择“**使用标记名作为项目名**”。
13. 如果您要为标记名定义报警条件，单击“**标记名字典**”对话框顶部的“**报警**”或“**详细和报警**”，以显示您所定义的标记名类型的相应报警条件对话框。

    有关报警条件的详细信息，请参阅“定义标记名报警条件”。

14. 在“**记录死区**”框中，输入标记名记入历史记录文件前，标记名的工程单位必须变更的数量。

注意：如果您要在标记名的工程单位改变超过**记录死区值**时，将标记名记录到磁盘中，则必须为标记名选择“**记录数据**”。

如果在 WindowViewer 运行时改变**记录死区**值，则在停止并重新启动历史记录前，您所做的更改不会生效。

有关历史记录的详细信息，请参阅第 8 章“实时趋势和历史趋”。

15. 在定义完标记名后，单击“**关闭**”以保存您的标记名定义并关闭标记名对话框。

定义 I/O 消息标记名详细资料

所有从另一 Windows 程序读取其值或将其值写入另一 Windows 程序的标记名为 I/O 类型标记名。这包括所有可编程控制器、过程计算机、其它 Windows 程序以及网络节点数据的输入和输出。 当需要从 I/O 设备采集文本字符串时，您就可以定义一个 **I/O 消息**类型的标记名。I/O 消息型标记名的长度限制在 131 个字符内。

➢ **要定义 I/O 消息型标记名的详细资料：**

1. 选择 **I/O 消息**作为标记名类型，出现下面的详细资料对话框。

   如果没有出现，单击“**标记名字典**”对话框顶部的“**详细资料**”。



2. 在“**最大长度**”框中，输入标记名消息所允许的最大字符数。（InTouch 允许最大 131 个字符，并将其作为缺省值显示）。
3. 在“**初始值**”框中，输入 WindowViewer 初次启动时标记名所显示的文本字符串。此值不会被写入 I/O 设备。
4. 单击“**访问名**”，定义或选择您要赋给标记名的访问名（如果此按钮右边已经显示访问名，并且您没有定义或选择不同的访问名，则该访问名将被赋给标记名）。

   有关访问名的详细信息，请参阅第 9 章“I/O 通信”。
5. 在“**项目**”框中，为标记名从中读/写其值的服务程序中的数据点输入有效的项目名。例如，如果您要从一个 PLC 寄存器中读值，则可以输入该寄存器的有效标识作为项目名。

   项目名会被自动编号。例如，如果您输入并存储项目名 R4001，然后单击“**新建**”（定义一个新标记名），项目名将自动编号为 R4002。如果项目名包含一个字符分隔的数字，它将自动编号为 InTouch 找到的第一个整数。例如，N7-0 将编号为 N7-1。注意只允许正方向的变化。例如，R4002 到 R4003，R4003 到 R4004 等等。
6. 如果您要为项目名使用该标记名，选择“**使用标记名作为项目名**”。
7. 在定义完标记名后，单击“**关闭**”以保存您的标记名定义并关闭标记名对话框。

定义 SuperTag 成员标记名详细资料

成员标记名在 SuperTag 模板中定义。成员标记名的行为和常规标记名一样，并且可以是离散型、整型、实型、消息型或另一 SuperTag。与常规 InTouch 标记名一样，成员标记名支持趋势、报警和所有的标记名**点域**。

有关成员标记名的详细信息，请参阅“创建 InTouch SuperTag”。

在定义标记名并选择 SuperTag 模板作为其标记名类型时，缺省情况下，所有在 SuperTag 模板中定义的成员标记名将设为“内存”数据访问类型。如果“内存”型是您需要的类型，则不需进行特殊的配置。但是，如果您需要将 SuperTag 模板中的任何成员标记名定义为 I/O 类型，则必须进行一些额外的配置。

➢ **要定义 I/O SuperTag 成员标记名：**

1. 选择 SuperTag 模板作为标记名类型，出现下面的详细资料对话框：

   如果没有出现，单击“**标记名字典**”顶部的“**成员**”对话框。



   注意您在“**标记名**”框中输入的新标记名变成了“**成员表**”中所有成员标记名的“父辈”。

2. 单击“**成员表**”箭头，然后选择表中您要定义为 I/O 数据访问类型的成员标记名。

3. 在“**数据访问**”组中，选择“**I/O**”。成员标记名类型 – 离散、模拟（实型或整型）或消息型 – 的相应 I/O 详细资料对话框将出现。

4. 与常规 InTouch I/O 型标记名一样，输入所要求的 I/O 详细资料。

5. 要保存所做的更改，选择表中的另一成员标记名并进行配置，或单击“**关闭**”。

定义标记名报警条件

您可以在定义标记名的同时定义其报警条件。有两种类型的报警详细资料对话框。一种用于离散型标记名，另一种用于模拟（整或实）型标记名。

报警约束标记名

您可以选择对每个报警（或报警子状态）指定一个可以禁止报警的标记名。当标记名为真（非零或非空）时，报警将被有效禁止。同样，当标记名为假（零或空）时，报警不会被禁止。您可以在 WindowMaker 的标记名字典中指定“**报警约束标记**”。**报警约束标记**不能在运行时改变，但可以在运行时改变**约束器标记**的值。

报警可以被有效禁止、停用，或同时被有效禁止和停用。如果报警当前被停用或有效禁止，报警将被“关闭”。这意味着如果您尝试启用当前被有效禁止的报警，报警将不能“打开”，

注意：报警约束标记是常规标记名，包括在使用计数与许可证限制中。

您也可以使用一些只读标记名点域来返回一个包含**报警约束标记**名称的字符串。这些点域是：

AlarmDscInhibitor
AlarmLoLoInhibitor
AlarmLoInhibitor
AlarmHiHiInhibitor
AlarmHiInhibitor
AlarmMajDevInhibitor
AlarmMinDevInhibitor
AlarmRocInhibitor

这些点域会返回标记名的名称。因此，您可以在 InTouch QuickScript 中使用间接标记名引用的名称，来找出**报警约束标记**的当前值，或改变**报警约束标记**的值。这样，您就可以在运行时强制启用或有效禁止报警组。

有关这些报警点域的详细信息，请参阅联机*《InTouch 参考指南》*。

禁止报警包括两个阶段：

1. 指定约束器标记名。
2. 将约束器标记名状态从“假”改为“真”，或从“真”改为“假”。

与停用一样，对于包含子状态的报警，每个子状态可单独禁用。每个子状态可被不同标记名禁用。

与停用一样，被禁止的报警（此时标记名值为“真”）不会等待***确认***。如果报警具有子状态，它只能等待***确认***仍可用的子状态。

每当转换导致报警脱离**有效禁止**状态，程序会执行检查逻辑来决定 InTouch 是否应将项目引入***报警***状态。

如果报警变为**有效禁止**，而项目仍处于***报警***状态，则项目会被强制为不同的（有效）状态。具体的状态则取决于当前何种状态可用，以及该状态是否也被停用或有效禁止。此活动由 InTouch 根据报警类型、极限值等进行处理。

如果报警（或报警子状态）变为**有效禁止**但仍等待***确认***，则项目会被强制为不同的（有效）状态。与项目是否应该报警一样，InTouch 必须决定项目应处于何种状态。

有关报警的详细信息，请参阅第 7 章“报警/事件”。

定义离散标记名报警条件

离散报警对应于离散型标记名。您可以配置***报警***状态是对应于离散型标记名的“真”（开、是、1）状态还是假（关、否、0）状态，以及配置报警的关联***优先级***。

➢ **要定义离散标记名的报警条件：**

1. 在“**特别**”菜单上，单击“**标记名字典**”，或者在应用程序浏览器中，双击“**标记名字典**”。“**标记名字典**”对话框出现。
2. 单击“**标记名字典**”对话框顶部的“**报警**”或者“**详细和报警**”，显示离散报警详细资料对话框：



3. 选择您要使用的“**确认模型**”选项。

   条件：所有进入报警状态或子状态的转换的确认次数取决于确认时间。（这是缺省选项，其作用与先前版本一样）。

   面向事件：确认只针对进入报警状态或子状态的特定转换；只有在针对最近一次此类转换时，确认才会被接受。

   扩展的摘要：确认只是针对特定转换进行的，无论是转入报警状态、子状态还是恢复正常状态。每次从正常状态的转换均标志着新 RTN 组的开始，RTN 组中的所有转换必须单独获得确认，整体 RTN 组才被视为已确认。

注意：在定义**扩展的摘要**确认模型的标记名时，“**报警属性**”对话框中的“**返回意味着确认**”选项不得应用于该标记名。

可以配置任何一个报警来使用这些确认模型之一。

有关详细信息，请参阅第 7 章的“报警确认模型”一节。

1. 在“**报警注释**”框中，输入您要用于 **.AlarmComment** 点域的缺省注释（最多 131 个字符）（这是一个可选点域）。

   有关详细信息，参见“标记名报警注释”。

2. 单击报警时标记名所处的“**报警状态**”。
3. 在“**优先级**”框中，输入一个介于 1 和 999 之间的数值（1 是最高优先级，999 是最低优先级）。您可以用此优先级值来选择您要显示在窗口中、记录到磁盘或打印出来的报警。

4. 要将约束器标记名与报警条件关联，单击报警条件对应的“**报警约束标记**”(...) 按钮。“**选择标记**”对话框出现：

注意：当选定对应的报警条件时，“报警约束标记”（...）按钮即会被激活。您不能在“**报警约束标记**”输入框中输入内容。

5. 在“**选择标记**”对话框中，双击您要用于抑制报警条件的标记名。对话框将关闭，所选标记名将出现在“**报警约束标记**”框中。

有关详细信息，参见“报警约束标记名”。

6. 在完成所有输入后，单击“**关闭**”保存您的标记名定义并关闭所有标记名对话框。

定义模拟型标记名的报警条件

模拟型报警对应于模拟型标记名。您可以配置***报警***状态是否对应于标记名的任意模拟值以及该报警的关联优先级。

➢ **要定义模拟型标记名的报警条件：**

1. 在“**特别**”菜单上，单击“**标记名字典**”，或者在应用程序浏览器中，双击“**标记名字典**”。“**标记名字典**”对话框出现。
2. 单击“**标记名字典**”对话框顶部的“**报警**”或者“**详细和报警**”，显示模拟报警详细资料对话框：



3. 选择您要使用的“**确认模型**”选项。

   条件：所有转换进入报警状态或子状态的确认次数取决于确认时间。（这是缺省选项，其作用与先前版本一样）。

   面向事件：确认只针对进入报警状态或子状态的特定转换；只有在针对最近一次此类转换时，确认才会被接受。

   扩展的摘要：确认只是针对特定转换进行的，无论是转入报警状态、子状态还是恢复正常状态。每次从正常状态的转换均标志着新 RTN 组的开始，RTN 组中的所有转换必须单独获得确认，整体 RTN 组才被视为已确认。

注意：在定义**扩展的摘要**确认模型的标记名时，“**报警属性**”对话框中的“**返回意味着确认**”选项不得应用于标记名。

可以配置任何一个标记名来使用这些确认模型之一。所选确认模型将应用于该标记名的所有报警类型。

有关详细信息，请参阅第 7 章的“报警确认模型”一节。

4. 在“**报警注释**”框中，输入您要用于 **.AlarmComment** 点域的缺省注释（最多 131 个字符）。

   有关详细信息，参见“标记名报警注释”。

5. 选择当模拟型标记名的值超出绝对极限时，用于检测的报警类型 (**LoLo**, **Low**, **High**, **HiHi**)。

6. 在“**报警值**”框中，输入报警的极限值。

   例如，在 **LoLo** 和 **Low** 报警类型下，只要标记名的值小于**报警值**，就会出现报警条件。在 **HiHi** 和 **High** 报警类型下，只要标记名的值大于**报警值**，就会出现报警条件。这些字段可使用实数（例如 100.75）。

7. 在“**值死区**”框中，输入在消除报警之前标记名必须降到报警值以下的工程单位数。

   例如，要从报警状态恢复正常，那么标记名的值不仅必须返回到报警限之内，而且必须返回通过您指定的“**值死区**”。**“值死区”**能够防止由于报警的不断反复引起的“过度”报警（即标记名值在报警限附近上下徘徊、连续的进入和退出报警状态）。

8. 选择当模拟型标记名的值为指定**目标**值的主或副偏差时，您要用于检测的偏差（**副偏差**和**主偏差**）报警类型。

9. 在“**偏差百分比**”框中，输入模拟标记名可以偏离**目标**值以产生副或主偏差报警条件的百分比。此值用标记名范围的百分比表示。在标记名详细资料对话框中输入的**最小工程单位**和**最大工程单位**值将定义该范围。

10. 在“**目标**”框中，输入副偏差和（或）主偏差百分比所基于的标记名所需值或参考值。

例如，假定我们在下面设置一个整型标记名：

最小值 = -1000

最大值 = 1000

副偏差百分比 = 10

主偏差百分比 = 15

目标值= 500

当标记名的总范围为 **-1000 到 +1000 或 2000** 时，要计算主或副偏差的报警值，可以将 **2000** 乘以主或副偏差的百分比：**2000×10%（副）= 200**。如果目标为 **500**，则每当标记名值低于 **300** 或高于 700 时，就会产生副偏差报警。

11. 在“**偏差死区百分比**”框中，输入在消除报警之前标记名值必须降到极限值以下的偏差百分比。

12. 如果您要检测指定时间间隔内报警值的变化是否超出范围，可以选择“**变化率**”。每当标记值改变时，系统都会检测此标记名并发出**变化率**型报警。此时，变化率将使用上次的值、上次更新时间、当前值以及当前时间进行计算。计算所得的值与报警条件允许的变化率进行比较。如果变化率大于报警限，则标记名将达到**变化率**型报警条件。**变化率**报警将持续生效，直至标记名值的下次变化小于指定时间间隔的变化范围。

13. 在“**%/**”框中，输入允许的最大变化百分比。

14. 选择变化的时间间隔单位：“**秒**”、“**分**”或“**小时**”。

15. 在对应的“**优先级**”框中，输入一个介于 1 到 999 之间的数字（1 是最高优先级，999 是最低优先级）。您可以用此优先级值来选择您要显示在窗口中，记录到磁盘或打印出来的报警。

16. 要将**约束器标记名**与报警条件关联，单击报警条件对应的“**报警约束标记**”(...) 按钮。“**选择标记**”对话框出现：

注意：当选定对应的报警条件时，“报警约束标记”（...）按钮即会被激活。您不能在“**报警约束标记**”输入框中输入内容。

---

17. 在“**选择标记**”对话框中，双击您要用于抑制报警条件的标记名。对话框将关闭，所选标记名将出现在“**报警约束标记**”框中。

有关详细信息，参见“报警约束标记名”。

18. 在完成所有输入后，单击“**关闭**”保存您的标记名定义并关闭所有标记名对话框。

标记名报警注释

在过去，InTouch 允许您进行配置，以便在产生报警时将标记名注释附加到报警中。这个标记名注释可以显示在报警向导或报警记录中。InTouch 也提供了允许操作员在确认报警时输入注释的配置设定。该确认注释将更新标记名数据字典中的标记名注释。

为配合分布式报警系统的其它配置选项，新版本扩充了标记名数据字典中每个标记名的信息存储量。现在，您可以创建专门用于报警的全新注释域、输入更长的内容，并将其与标记名注释完全分开。

报警注释是一个字符串型的新点域，可通过 InTouch QuickScript 设置或读取。点域名为 **.AlarmComment**。作为一个 InTouch 字符串，它可以容纳多达 131 个字符。在“报警注释”对话框中，用户可以输入此点域的缺省设定。

作为新版本的一项功能改进，报警不再使用标记名注释作为报警注释，而是使用点域 **.AlarmComment**。此外，如果操作员在确认报警时输入新注释，并且应用程序设定为将此注释复制到标记名数据字典，则 **.AlarmComment** 将被更新 – 而不是标记名注释。

由于这些改进都涉及对标记名数据字典的更改，因此文件标记名中的版本号 .x 已作修订。因此，WindowMaker 在载入包含先前版本标记名数据字典的应用程序时，会自动将旧的标记名数据字典转换为当前版本。作为转换的一部分，标记名注释会被复制到报警注释（缺省设定）。

要将运行时改变写入标记名数据库中的**报警注释**域，请为当前的应用程序在 INTOUCH.INI 文件中添加下面一行（与当前应用程序位于相同的存储目录）：

```
CommentRetentive=1
```

创建 InTouch SuperTag

InTouch 支持 SuperTag 模板结构，可以定义一个名叫 SuperTag 的复合标记名类型。SuperTag 模板最多可容纳 64 个标记名成员和两个嵌套级别。这就意味着，一个 SuperTag 父项可以包含最多 64 个内嵌的子项成员，并且每个子项成员又可以包含 64 个子成员标记名，因而总共可以有 4095 个成员标记名（当一个 SuperTag 模板父项被内嵌到另一个 SuperTag 模板时，它就变成了一个“子成员”）。所有 SuperTag 模板子成员标记名的行为与常规标记名一样，它们支持趋势、报警和所有标记名**点域**。

为方便起见，InTouch 提供了一个 "TemplateMaker" 来让您创建 SuperTag。TemplateMaker 允许您创建、编辑和删除 SuperTag 模板及成员标记名。InTouch 将所有的 SuperTag 模板都保存在 InTouch 安装目录（不是应用程序目录）下的 supertag.dat 文件中。这使得您可以将所创建的模板应用到任何应用程序中。

InTouch 还提供了其它几种方法来创建 SuperTag。例如，您可以从标记名字典、动画链接标记名或表达式输入框、InTouch QuickScript、或者您用 InTouch DBLoad 实用程序加载到应用程序的外部文件等，直接创建 SuperTag。

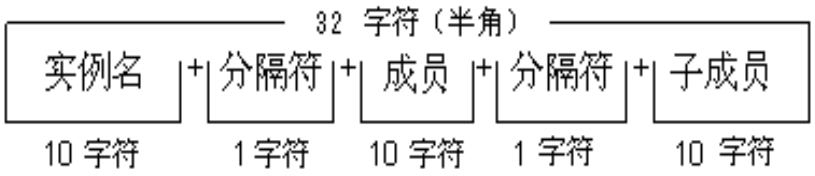
有关详细信息，参见“创建 SuperTag 的其它方法”。

在创建 SuperTag 父模板时，其名称会自动添加到标记名字典的标记名“**标记类型**”对话框中，并可立即供创建新标记名选用。您不必重新启动 WindowMaker，就可以定义一个标记名来使用新建的 SuperTag 类型。

警告：如果您修改一个现有的 SuperTag 模板，该 SuperTag 的所有现有实例不会受影响（实例指的是在标记名字典中使用 SuperTag 类型定义的标记名）。换句话说，您对 SuperTag 所做的修改不可回溯。然而，所有使用修改后的 SuperTag 定义的新实例将使用新的结构。相似地，如果您通过其它方法添加一个成员标记名到 SuperTag 实例，其模板也不会被更新。

InTouch SuperTag 语法

由于 InTouch 标记名限制在 32 字符内，每个 SuperTag 的父辈实例\子辈成员\子成员均限制为最多 32 个字符。SuperTag 引用的层数最多只能为两个模板（父辈实例\子辈成员）和一个成员，如下所示：



SuperTag 模板中的每个成员都可以使用您现在用于访问常规 InTouch 标记名类型的**点域**的标准格式去访问。InTouch 中凡是可以使用常规标记名的地方，都支持使用 SuperTag 引用语法。例如，一个有效的 SuperTag 引用可以是：

```
ColdRoom4\EvapUnit1\FanMotor2.MaxEU
```

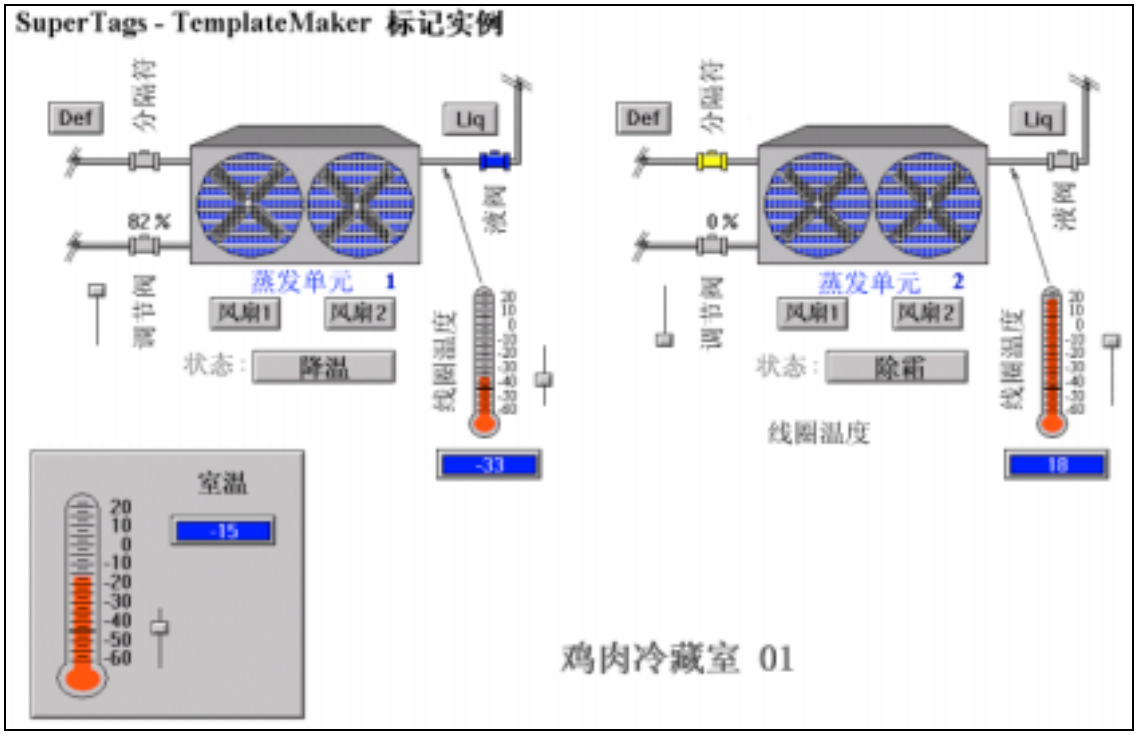
远程标记名引用也支持 SuperTag。语法实例：

```
PLC1:"Turkey\EvapUnit2\PrsRegVlv.EngUnits"
```

有关运用远程标记名引用的详细信息，请参阅“远程标记名引用”。

创建 SuperTag 模板结构

为了形象地描述工厂环境中的 SuperTag 概念，我们假定有四个同样的冷藏室，其中储存着牛肉、猪肉、鸡肉和火鸡。每个冷藏室都有一个室内温度，以及两个脱水器单元。每个脱水器单元有七个我们需要在运行时监控的数据值。例如：



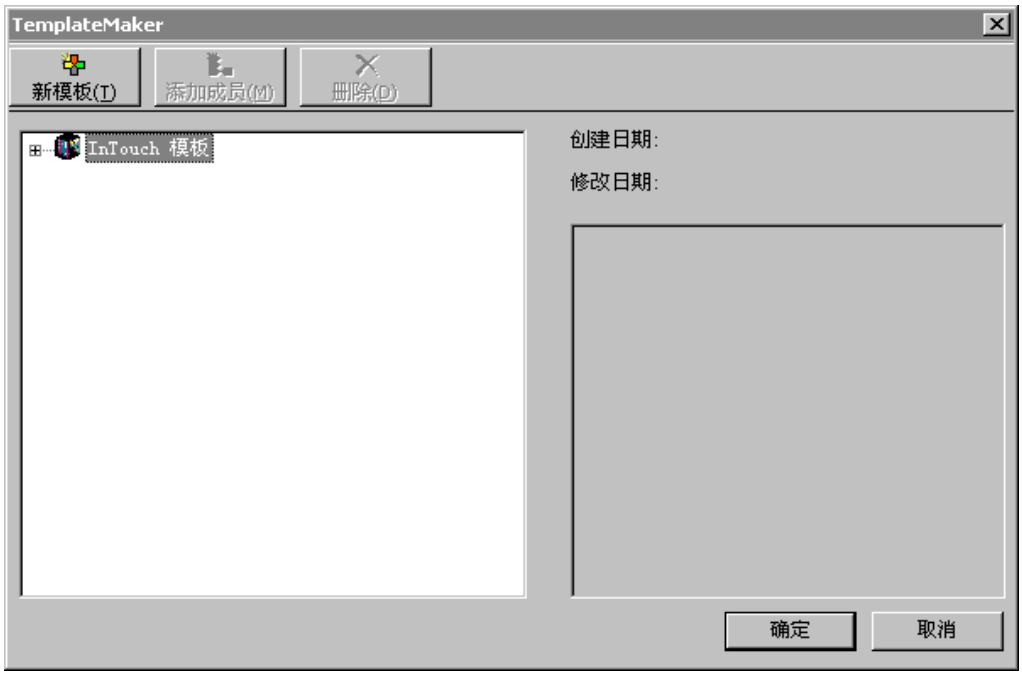
如果我们不创建 SuperTag 模板来实现，就需要为每个冷藏室中的每个数据值人工定义一个单独的标记名，然后乘以冷藏室的总数。换句话说，我们必须在标记名字典中组织和定义数十个标记名！

通过使用 SuperTag，我们就可以节约开发时间并最大程度地降低出错可能。使用上面所述的冷库方案，我们创建一个名为 “EvapUnit” 的 SuperTag 父模板（这个 EvapUnit 稍后将变成 ColdRoom 父模板的子成员，这是一种 “逐渐细化” 的设计概念）。EvapUnit 可以用七个子成员标记名来定义。

成员标记	类型	描述
FanMotor1	离散型	风扇 1 的马达启动器
FanMotor2	离散型	风扇 2 的马达启动器
DefrostVlv	离散型	除霜气体阀状态
LiquidVlv	离散	液体冷却阀状态
CoilTemp	实型	致冷剂的温度
PrsRegVlv	整型	压力调节阀(0-100%)
EvapStatus	消息型	蒸发单元状态字符串

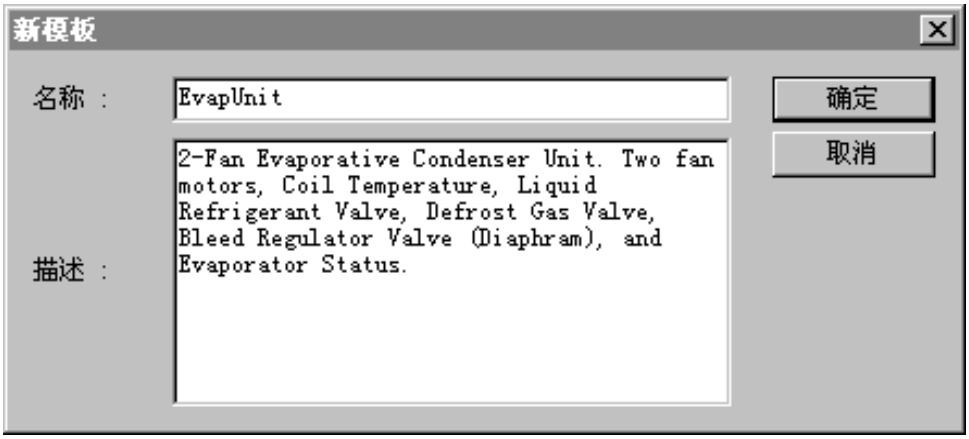
➢ **要创建一个 SuperTag 父模板：**

1. 在“**特殊**“菜单上，单击“**TemplateMaker**”，或者在应用程序浏览器中，双击“**TemplateMaker**”。**TemplateMaker** 对话框出现：



2. 在 TemplateMaker 窗口中，选择“**InTouch 模板**”，然后单击“**新模板**”，或者右击“**InTouch 模板**”，然后选择“**新模板**”。出现“**新模板**”对话框：

   - 您也可以选择“**InTouch 模板**”，然后右击窗口的空白区域。



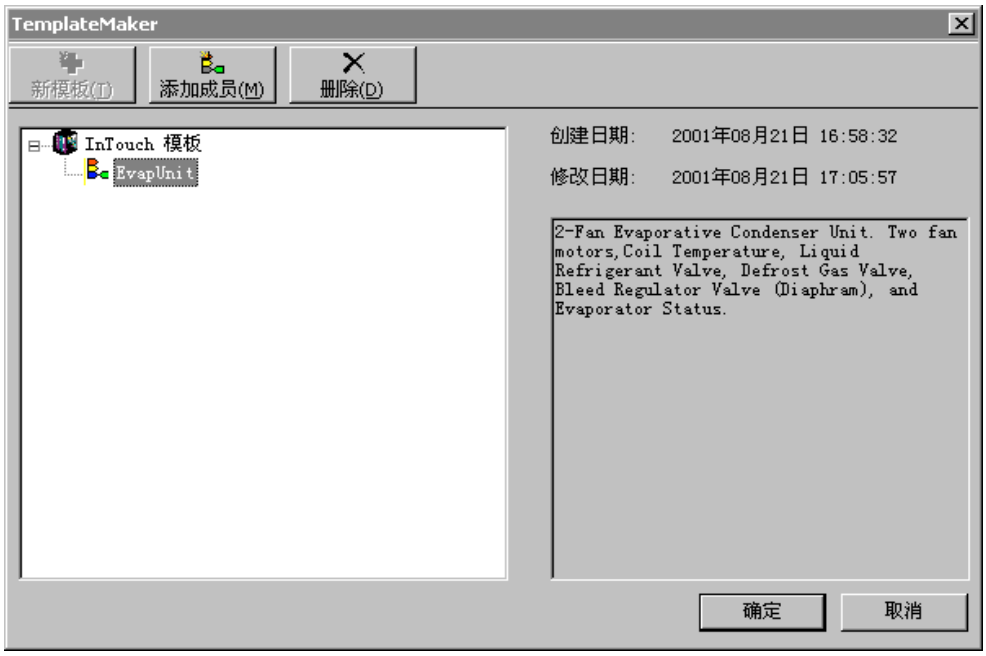
   - 如果您右击任一 TemplateMaker 对话框中的文本输入框，会出现一个菜单，显示可应用于选定文本的命令。

3. 在“**名称**”字段中，为新模板输入一个唯一的名称（最多 10 个字符）。

   当添加新的父模板时，其名称将作为标记名类型之一立即显示在标记名字典的“**标记名类型**”对话框中，并可立即供您选用。您不必重新启动 WindowMaker，就可以定义新的标记名并将其赋给 SuperTag 类型。

4. 在“**描述**”字段中，输入用于描述模板的任何信息。

5. 单击“**确定**”。**TemplateMaker** 对话框重新出现，在窗口中显示新的模板名。

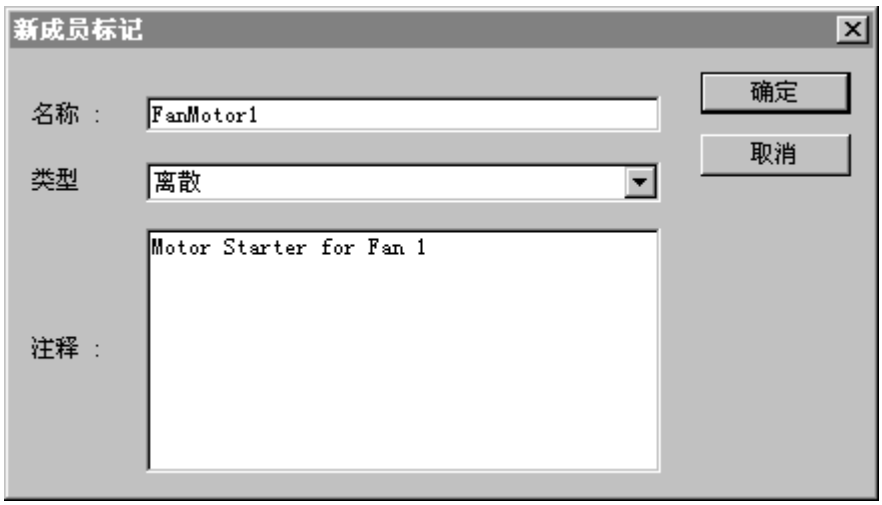


一旦创建模板，“**新成员**”和“**删除**”按钮即被激活。当选择模板名时，模板创建日期和时间和（或）最近一次修改的日期和时间、以及模板的描述也会显示出来。

注意：TemplateMaker 窗口以分层表的方式显示所有当前定义的 SuperTag 父模板及其子成员。要展开模板视图，单击模板名旁边的⊞。所有为父模板名定义的成员标记名都将显示出来。要折叠视图，单击⊟。

➢ **要创建 SuperTag 成员标记名：**

1. 在 TemplateMaker 窗口中，选择 SuperTag 模板（本例中为 EvapUnit），然后单击“**添加成员**”，或者右击 SuperTag 模板名称，然后单击“**添加成员**”。“**新成员标记**”对话框出现：



2. 在“**名称**”框中，输入新成员标记名的名称。
3. 在“**类型**”框中，输入成员的标记名类型，然后单击“**类型**”箭头，从列表中选择类型。类型可以是离散型、整型、实型、消息型或另一 SuperTag 模板。

   如果您输入类型的第一个字母，则列表框中以该字母开头的第一个类型会自动显示在框中。如果有多个类型以该字母开头，您可以继续输入名称的下一个字母，以此类推。

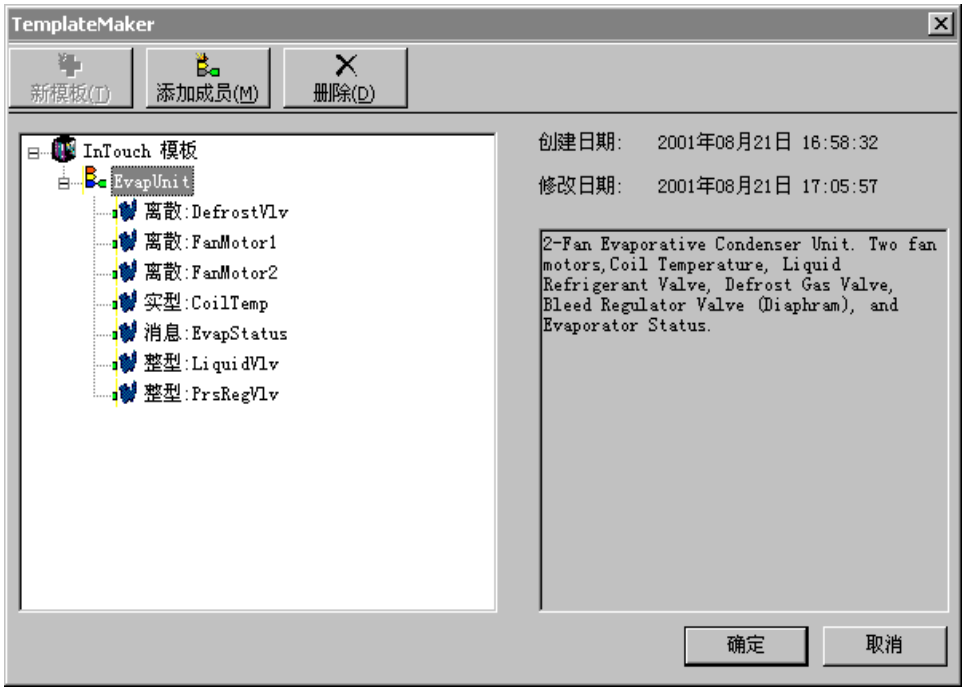
注意：此处指定的类型仅仅是 SuperTag 模板的占位符。缺省条件下，当您在 TemplateMaker 中定义时，所有成员标记名均设置为“内存”类型，但在标记名称字典中定义模板实例时，您必须指定这些标记名是“内存”还是“I/O”类型的标记名。

有关“**成员**”对话框的详细信息，请参阅“定义 SuperTag 成员标记名详细资料”。

4. 在“**注释**”字段中，输入用于描述成员标记名的任何信息。
5. 单击“**确定**”。

   重复此过程，以添加其它的成员标记名到 SuperTag 模板中。

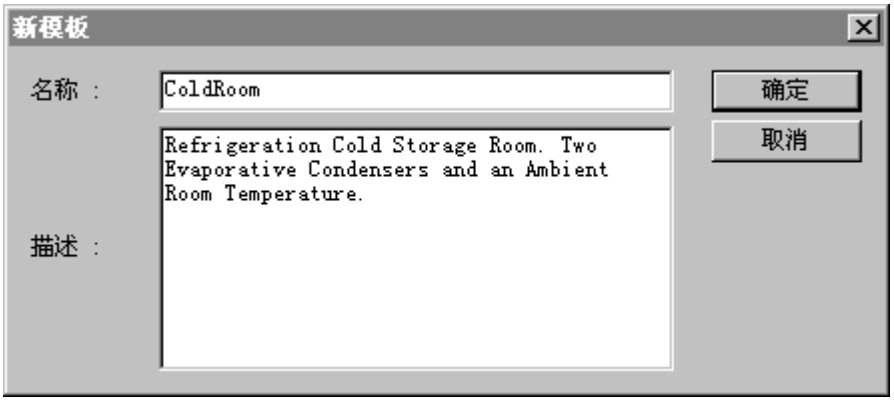
新的成员标记名会添加到 TemplateMaker 窗口中 SuperTag 父模板的下方。



注意：一旦选定成员标记名，“**新成员**”按钮将失效，这是因为成员只能在现有的 SuperTag 父模板下创建。当选择成员时，成员标记名的创建日期和时间、最近一次修改的日期和时间，以及模板描述都会显示出来。

6. 现在，创建另一个父模板 ColdRoom。ColdRoom 将包含一个名为 RoomTemp 的成员标记名和两个 EvapUnit 子成员模板（EvapUnit1 和 EvapUnit2）。这两个子成员模板将使用父 SuperTag 模板 “EvapUnit” 作为其类型。

7. 在 TemplateMaker 窗口中，选择“**InTouch 模板**”，然后单击“**新模板**”，或者右击“**InTouch 模板**”，然后单击“**新模板**”。“**新模板**”对话框出现：



8. 在“**名称**”字段中，为新的父模板输入一个唯一的名称（最多 10 个字符）

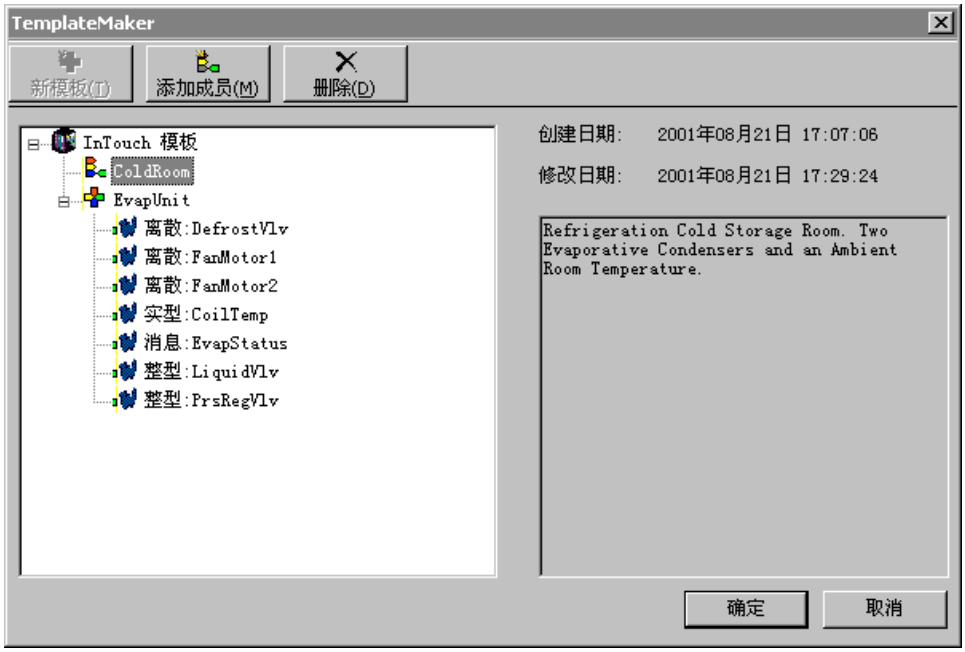
   当添加新的父模板时，其名称将作为标记名类型之一立即显示在标记名字典的“**标记名类型**”对话框中，并且可立即供您选用。您不必重新启动 WindowMaker，就可以定义新的标记名并将其赋给 SuperTag 类型。

9. 在“**描述**”字段中，输入用于描述模板的任何信息。

10. 单击“**确定**”。**TemplateMaker** 对话框重新出现，在窗口中显示新的模板名。

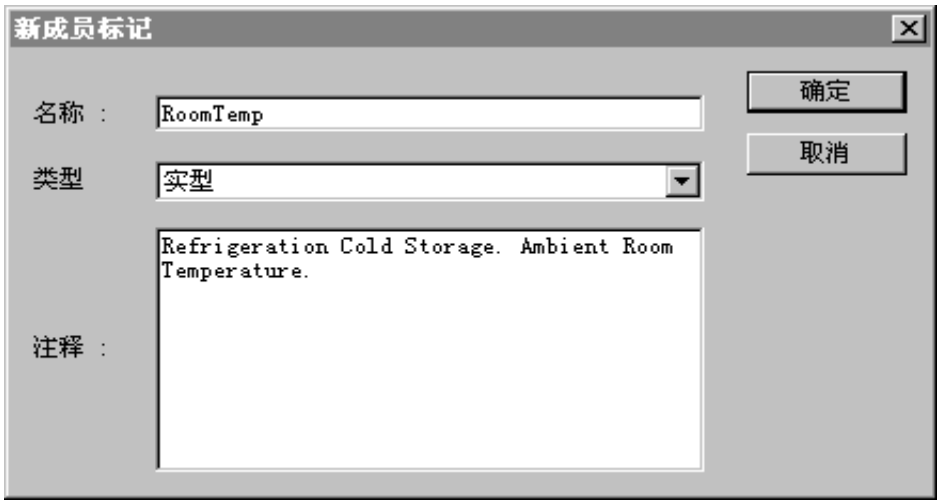
11. 单击“**确定**”。

父模板将添加到 TemplateMaker 窗口的 **InTouch 模板**列表中。



12. 我们现在需要为 ColdRoom 父模板创建三个成员、两个 EvapUnit 子成员和一个名为 RoomTemp 的成员标记名。

13. 在 TemplateMaker 窗口中，选择 SuperTag 父模板（本例中为 ColdRoom），然后单击“”**添加成员**”，或者右击 SuperTag 父模板名，然后单击“**添加成员**”。“**新成员标记**”对话框出现：



14. 在“**名称**”框中，输入新成员标记名的名称。

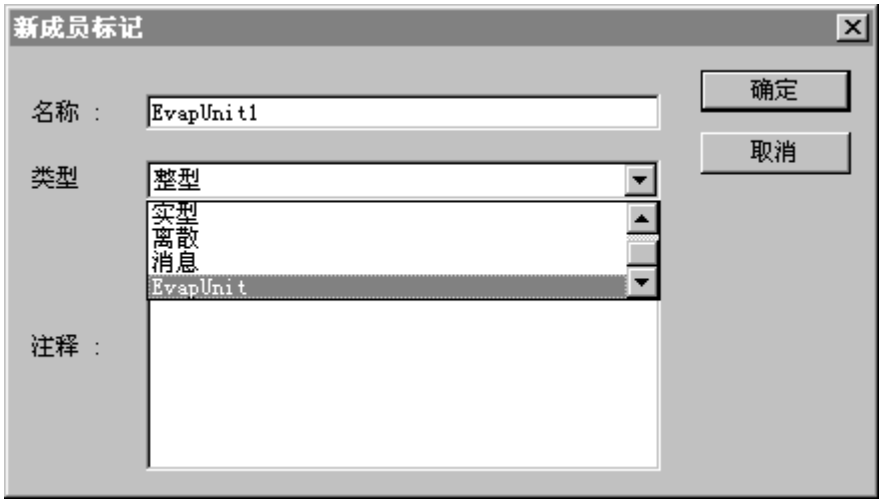
15. 在“**类型**”框中，输入成员的标记名类型，或者单击“**类型**”箭头，从列表中选择类型。类型可以是离散型、整型、实型、消息型或另一 SuperTag 模板。

16. 在“**注释**”字段中，输入用于描述成员标记名的任何信息。

17. 单击“**确定**”。

下一步，我们创建两个子成员模板：EvapUnit1 和 EvapUnit2，它们使用 EvapUnit 模板类型。

1. 在 TemplateMaker 窗口中，选择 SuperTag 父模板（本例中为 ColdRoom），然后单击“”**添加成员**”，或者右击 SuperTag 父模板名，然后单击“**添加成员**”。“**新成员标记**”对话框出现：



2. 在“**名称**”框中，输入新成员标记名的名称。

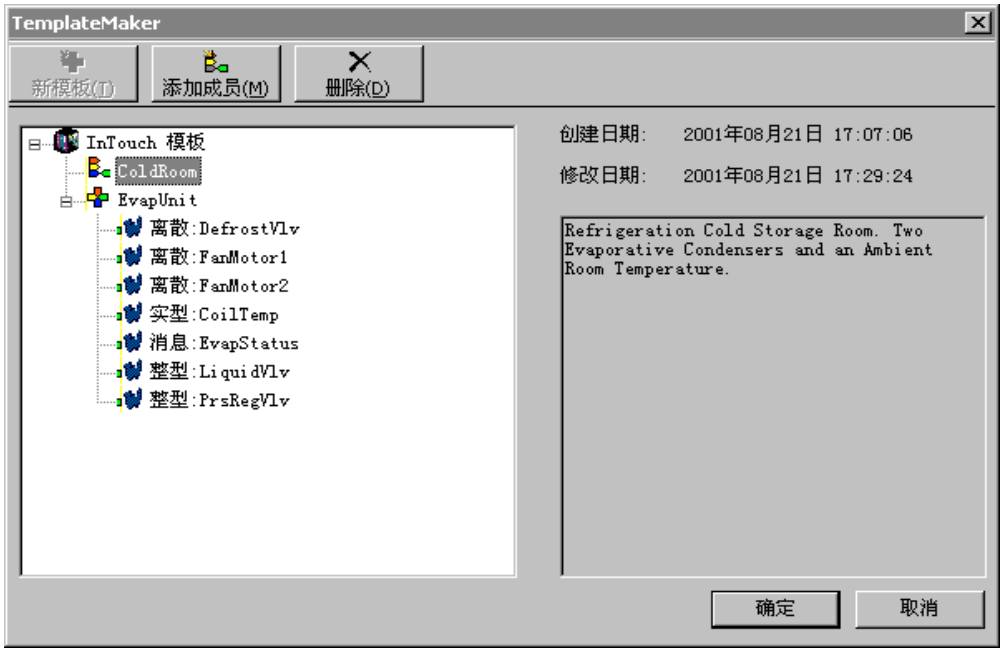
3. 在“**类型**”框中，输入成员的标记名类型，或者单击“**类型**”箭头，从列表中选择类型。此时，我们可以使用特殊模板类型 EvapUnit。

4. 在“**注释**”字段中，输入用于描述成员标记名的任何信息。

5. 单击“**确定**”。

   对 EvapUnit2 重复上述过程。

一旦完成 ColdRoom 父模板，TemplateMaker 窗口将显示下列的模板层次结构：



单击“**确定**”。

至此我们完成了 ColdRoomSuperTag 模板，可以着手创建使用该模板作为其标记名类型的标记名实例。

有关详细信息，请参阅“定义 SuperTag 模板实例”。

创建间接 SuperTag

在定义新的 SuperTag 标记类型时，InTouch 会自动定义对应的间接 SuperTag 标记类型。这个对应的间接 SuperTag 是一个完全拷贝，它包括原件的所有成员标记名。



复件中所有成员的标记类型会自动设置为间接。例如，假定您定义了下面的 SuperTag 标记名类型：

MySuperTagType

离散型	MyDiscrete
整型	MyInteger
实数	MyReal
消息型	MyMessage

InTouch 使用下面的结构来定义间接 SuperTag 标记名类型：

Indirect MySuperTagType

间接	离散型	MyDiscrete
间接	模拟型	MyInteger
间接	模拟型	MyReal
间接	消息型	MyMessage

间接 SuperTag 标记类型的脚本行为类似于间接离散型、间接模拟型或间接消息型的标记类型。参看下面的语句实例：

```
MyIndirectSuperTag.Name = "MySuperTag";
```

如果等号右边的名称指 SuperTag 父项，InTouch 会遍历其子项，并尝试指定一个值给左边间接 SuperTag 的对等成员。如果 InTouch 不能找到对等成员，InTouch 将不会特定子项赋值。

假定上述 SuperTag 具有下列结构：

MyIndirectSuperTag	MySuperTag
Value1	Value1
Value2	Value2
Value3	Value3
ValueX	Value4
	Value5

上面的语句具有下面的结果：

```
MyIndirectSuperTag.Value1 = MySuperTag.Value1
MyIndirectSuperTag.Value2 = MySuperTag.Value2
MyIndirectSuperTag.Value3 = MySuperTag.Value3
```

此外，间接 SuperTag 标记类型的作用与动画链接中的间接离散型、间接模拟型或间接消息型标记名类型一样。例如，触动按钮动作脚本可以执行下面的语句：

```
MyIndirectSuperTag.Name = "MySuperTag";
```

随后，模拟值显示链接可以使用下面的表达式来显示 MySuperTag.Value1 的十倍值。

```
MyIndirectSuperTag.Value1 * 10
```

编辑 SuperTag 模板和成员标记名

您可以随时修改 SuperTag 模板或成员标记名。然而，如果您修改一个现有的 SuperTag 模板或其成员，该模板的所有现有实例将不会受影响（实例指的是在标记名字典中使用 SuperTag 类型定义的标记名）。换句话说，您对 SuperTag 所做的修改不可回溯。然而，所有使用修改后的 SuperTag 定义的新实例将使用新的结构。

- **要编辑现有的 SuperTag 模板或成员标记名：**

1. 在 TemplateMaker 窗口中，双击 SuperTag 模板名（或成员名），或右击它，然后单击“**编辑**”。“**编辑模板**”或“**编辑成员标记**”对话框出现，显示 SuperTag 模板（或成员）的定义。
2. 进行所需的编辑，然后单击“**确定**”

- **要删除 SuperTag 模板或成员：**

1. 在 TemplateMaker 窗口中，选择您要删除的 SuperTag 模板名（或成员名），或右击它，然后单击“**删除**”。屏幕出现一个要求您确认删除的对话框。
2. 单击“**是**”删除所选名称，或单击“**否**”取消删除。

注意：如果您按下 ESC 键而不是“**确定**”来关闭 TemplateMaker，则模板不会被删除。当您删除模板时，也将删除所有相关联的成员标记名。

定义 SuperTag 模板实例

在 TemplateMaker 中一个很重要的概念是区分 SuperTag 模板和模板实例。模板实例是 SuperTag 模板的一个特定的具体示例。模板与实例的最重要区别是父模板名将被实例标记名替代，而子模板名和次成员标记名不会改变。

这相当于一个用来设计蜡纸之类的文字模板，该蜡纸用来制作实际的图画。图画自身，打个比方，就是“模板实例”，它们按照模板和蜡纸的样子进行创建。

再次回到 ColdRoom 模板方案，在创建了该模板后，从这个模板我们可以创建 SuperTag 实例如“牛肉”、“猪肉”、鸡肉”和“火鸡”。为此，我们只要简单地创建四个使用 ColdRoom 作为其类型的标记名。这样我们一次努力，就可以快速地在标记名字典中创建 60 个标记名。从而节约了大量的时间。

在创建"ColdRoom" SuperTag 模板和实例后，我们可以通过使用有效的 SuperTag 引用，在动画链接表达式或 QuickScript 中引用其任何成员。例如：

```
Beef\RoomTemp
Chicken\RoomTemp.RawValue
Chicken\EvapUnit1\FanMotor1.OnMsg
Pork\EvapUnit2\EvapStatus
Turkey\EvapUnit2\PrsRegVlv.EngUnits
```

有关定义模板实例的详细信息，请参阅“定义 SuperTag 成员标记名详细资料”。

➢ **要使用新建的模板来创建 SuperTag：**

1. 在“**特别**”菜单上，单击“**标记名字典**”，或者在应用程序浏览器中，双击“**标记名字典**”。“**标记名字典**”对话框出现：



2. 单击“**新建**”（“**标记名**”框将被清除）。
3. 在“**标记名**”框中，输入新标记名的名称。

   标记名最长可为 32 个字符，但必须以一个希腊字符开头（**A-Z** 或 **a-z**）。其余的字符可以是 A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ 和 &。

   标记名不能使用 **RetVal** 一字。这个字是保留字。如果您试图使用这个字，然后尝试编辑 QuickFunction，将出现一条错误消息 "A variable cannot have that name.Tag exists"（变量不能使用该名称，标记已经存在）。

4. 单击“**类型**”，“**标记名类型**”对话框出现：



5. 从标记名中选择 **ColdRoom**，然后单击“**确定**”。

6. 选择 SuperTag 模板作为标记名类型，出现下面的详细资料对话框：

   ☞ 如果没有出现，单击“**标记名字典**”顶部的“**成员**”对话框。



   注意您在“**标记名**”框中输入的新标记名变成了“**成员表**”中所有成员标记名的“父辈”。

7. 单击“**成员表**”箭头，从表中选择您要定义为 I/O 数据访问类型的成员标记名。

8. 在“**数据访问**”组中，选择“**I/O**”。成员标记名类型 – 离散、模拟（实型或整型）或消息型 – 的相应 I/O 详细资料对话框出现。

9. 与常规 InTouch I/O 型标记名一样，输入所要求的 I/O 详细资料。

10. 要保存您所做的更改，选择表中的另一成员标记名并进行配置，或单击“**关闭**”。

    ☞ 有关标记名类型的详细信息，请参阅“标记名类型”。

有关创建 InTouch SuperTag 的详细信息，请参阅“创建 InTouch SuperTag”。

➢ **要复制一个从模板创建的 SuperTag 实例：**

1. 在标记名字典中选择 SuperTag，然后单击“**新建**”。

2. 出现一个对话框，询问您是否要复制。如果您单击“**是**”，一个对话框会提示您输入新 SuperTag 的名称。输入 10 个字符长的名称，然后单击“**确定**”。

注意：此功能不适用于通过动画链接表达式、InTouch QuickScript，以及通过 DBLoad 实用程序加载到标记名字典中的外部 .csv（逗号分隔的变量）文件等方式创建的 SuperTag。

至此您已按照原件模板创建了一个新的 SuperTag 实例，其唯一的区别是根或父模板名称。

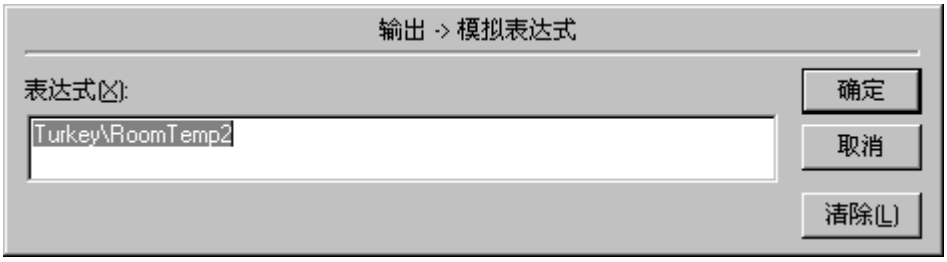
创建 SuperTag 的其它方法

除了 TemplateMaker，InTouch 支持通过动画链接表达式、InTouch QuickScript，以及通过 DBLoad 实用程序加载到标记名字典中的外部 csv（逗号分隔的变量）文件等，来创建 SuperTag。然而，您也可以通过标记名字典将成员或子成员添加到现有的 SuperTag 中，这是最简便的一种方法。

有关使用 DBLoad 创建 SuperTag 的详细信息，请参阅“创建 SuperTag 实例”

注意：在使用其它方法创建成员时，该成员不能反映在 TemplateMaker 中的 SuperTag 模板定义里。

在通过动画链接表达式或 InTouch QuickScript 创建 SuperTag 时，必须使用有效的 SuperTag 格式。例如：



下面的语法实例是有效的：

```
ParentInstance\ChildMember
ParentInstance\ChildMember\Submember
```

而下面的语法例子是无效的：

```
ParentInstance\
```

```
ParentInstance\ChildMember\
```

如果使用了无效的格式，将出现一个错误消息框，通知您语法是无效的。

- 如果在动画表达式或 QuickScript 中指定的 SuperTag 实例和成员标记名当前尚未定义，将出现一个消息框，询问您是否要现在进行定义。单击“**确定**”。标记名字典出现，显示您指定的 SuperTag 实例和成员标记名。

使用标记名字典创建 SuperTag

标记名字典是创建 SuperTag 实例或成员标记名的最简单的替代方法。

➢ **要在标记名字典中创建 SuperTag：**

1. 在“**特别**”菜单上，单击“**标记名字典**”，或者在应用程序浏览器中，双击“**标记名字典**”。“**标记名字典**”对话框出现。单击“**新建**”。



2. 在“**标记名**”框中，输入 SuperTag 实例的准确名称，后跟反斜线 (\) 分隔符和新成员标记名。此例中，我们输入 **Turkey\RoomTemp2**。

注意：在对现有的 SuperTag 实例添加新成员时，实例名的拼写必须完全相符。否则，会添加一个全新的 SuperTag 实例和成员。

标记名最长可为 32 个字符，但必须以一个希腊字符开头（**A-Z** 或 **a-z**）。其余的字符可以是 A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ 和 &。

标记名不能使用 **RetVal** 一字。这个字是保留字。如果您试图使用这个字，然后尝试编辑 QuickFunction，将出现一条错误消息 "A variable cannot have that name.Tag exists"（变量不能使用该名称，标记已经存在）。

3. 单击“**类型**”，选择 SuperTag 成员的类型（在这里不需选择其它选项）。此例中，我们选择了 I/O 实型。

4. 单击“**保存**”或“**关闭**”以添加该成员。

5. 要在不退出标记名字典的情况下查看 Turkey SuperTag 中的成员标记名，单击左或右双箭头按钮。“**成员**”详细对话框出现：



6. 单击“**关闭**”，关闭标记名字典。

7. 当 SuperTag 显示在标记名字典中时，单击“**新建**”，将出现下面的对话框，询问您是否需要制作所显示的 SuperTag 实例的相同副本

注意：在对现有的 SuperTag 实例添加新成员时，实例名的拼写必须完全相符。否则，会添加一个全新的 SuperTag 实例和成员。



8. 单击“**是**”，创建另一个 SuperTag 实例，它是所显示的 SuperTag 实例的完全相同的副本。“**输入名称**”对话框出现：



9. 输入新 SuperTag 实例的名称。

10. 单击“**确定**”。

    标记名字典会为新的 SuperTag 实例自动创建所有成员标记名和子成员标记名，这些标记名将可立即用于动画链接和 InTouch QuickScript。

远程标记名引用

InTouch 为工厂自动化应用提供了真正的客户端/服务器结构。用户不必使用本地标记名字典中的任何标记名，就可以设计出客户端应用程序。这可以通过使用 InTouch 的“远程标记名引用”功能来实现。例如：



在这个实例中，您可以用两种方法获取节点 Node2 上的 "TempTag" 标记名值。

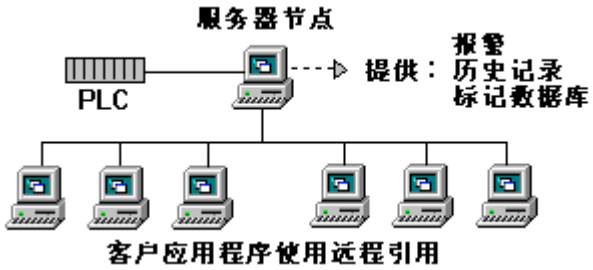
1. 在 Node1 的标记名字典中创建一个 I/O 型标记名，在与 I/O 标记名关联的访问名中使用 Node2 作为“**节点名**”。
2. 使用一个到 "TempTag" 的直接远程引用。例如，**PLC1:"TempTag"**。

换句话说，在窗口或 QuickScript 中，您既可以引用本地标记名，也可以使用 *AccessName:"item"*，来引用一个远程标记名。

有关远程标记名引用语法的详细信息，请参阅“远程标记名引用语法”。"

如果您要直接引用任何其它 FactorySuite 应用程序中的远程标记名，只需要 *AccessName:"item"* 就可以了。您不必在本地标记名字典中定义远程标记名。远程引用也可以访问来自任何 I/O 数据源，如 WonderwareI/O 服务器或 Microsoft Excel 的数据；远程引用也支持 SuperTag。远程标记名引用 SuperTag 的有效语法是：
Accessname:"ParentInstance\ChildMember\SubMember"。

此外，在使用远程标记名引用并导入窗口或 QuickScript 时，所有要做的事情就是将占位符标记名转换成远程标记名引用。您不必在本地标记名字典中定义标记名，而可以从网络上的任一 FactorySuite 应用程序访问远程标记名引用，如下图所示。



远程标记名引用语法

远程标记名引用的有效语法是 *AccessName:"item"*。远程引用中使用的字符与标记名的有效字符相同，即有效的字符可以是：A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ 和 &。如果您确定远程标记名引用中没有使用无效字符，就不必用引号将*项目*部分括起。"

- 远程标记名引用中的“项目”部分也可以使用标记名**点域**。例如，"MyAlarm.HiHi"。

为了使用任何其它字符，您必须将*项目*部分用引号括起。例如，如果您使用; ~， *， /， +， =， ^， |， **， <， >， <=， =>， ==， 和 <>，则必须将*项目*部分放在引号中。

然而，由于某些通用 ASCII I/O 服务器接受任何字符作为有效项目名，我们极力推荐您总将*“项目”*部分用引号括起。

例如，假定您要获取 Alen-Bradley® PLC 整型寄存器中的一个位，并使用“**N10:7/3**”（整数 10 的第三位) 作为项目名。系统会将 “**N10:**”视为访问名，因为正斜杠 (/) 不是一个有效字符。然而，如果您将项目名放在引号中，即 **"N10:7/3"**，系统就会将整个字符串作为项目名。

您也不能连接项目名或远程标记名引用项目名。例如，假定您使用下面的表达式创建了一个字符串输出链接：



当系统执行上面的表达式时，它将使用访问名 “**PLC2**”，并通过 Allen-Bradley I/O 服务器去获取存储在字符串文件 **ST10:1** 中的字符串。然后它会将字符串 “**37**” 附加到从 **ST10:1** 取回的字符串的尾部。如果 **ST10:1** 中存储了 "Green Paint"，则链接到该表达式的字符串输出对象将显示 “**Green Paint37**”。因此，操作员看不到他们本想要的 **ST10:137** 的内容。

每次您使用远程引用 (*accessname:"item"*) 时，InTouch 会验证您指定的访问名是否有效。如果确定访问名没有定义，它将提示您进行定义。如果您在提示下选择“**是**”，“**访问名**”对话框将出现，您可以添加新的访问名。

当远程标记名被激活时，系统也会验证访问名是否有效。如果检测到错误，会将其写入 Wonderware Logger。

您可以删除由远程引用使用的访问名，前提是本地标记名不再使用它。

有关于定义访问名的详细信息，请参阅第 3 章“创建分布式应用程序”。

有关 Wonderware Logger 的详细信息，，请参阅您的*《FactorySuite 系统管理员指南》*。

记录远程引用标记名

缺省条件下，远程引用的标记名会被记入历史记录文件中。要记录远程引用标记名，您必须启用“历史记录”，然后在应用程序目录的 INTOUCH.INI 文件中添加下面一行：

RemoteTagsLogEvents=1

如果不想记录 I/O 标记名，请在应用程序目录的 INTOUCH.INI 文件中添加下面一行：

RemoteTagsNoIOEvents=1

注意：仅当 RemoteTagsLogEvents 设置为 1 时，RemoteTagsNoIOEvents 设置才适用。

远程标记名许可证强制

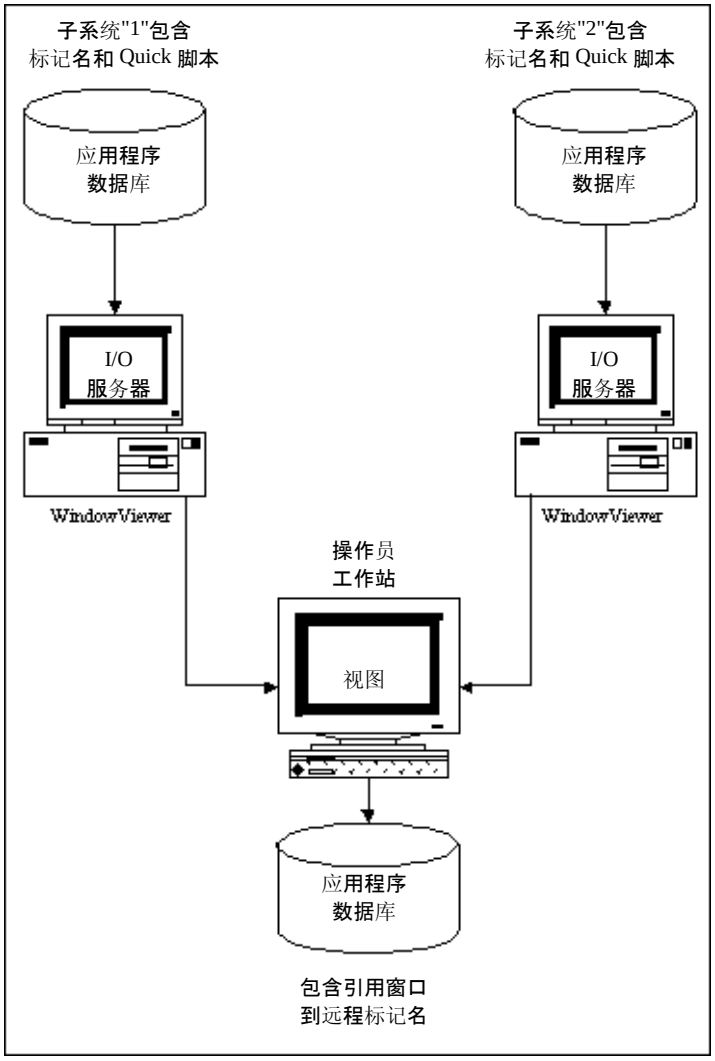
InTouch 标记名数据字典可支持多达 61,405 个标记。InTouch 60K 标记名许可证允许最大数目的标记名引用。如果安装了 60K 标记名许可证，则每次在窗口或 QuickScript 中激活标记名时，许可证强制逻辑所使用的标记名引用计数会增大；每次引用窗口或 QuickScript 关闭和（或）终止时，引用计数会减小。

在 InTouch 7.11 之前的版本中，在一个应用程序进程内，设计时声明的标记名总数加上有关的远程标记名引用数目必须始终小于或等于 60K。InTouch 7.11 60K 标记名许可证允许应用程序设计人员声明超过 60K 的设计时加远程标记名引用数目。当远程标记名分别进入或退出应用程序进程范围时，系统会动态地增大或减小总的标记名引用计数。

注意：在许可证允许少于最大标记名引用数的情况下，一旦标记名在许可限额中被引用和计数，即使标记名不再使用，限额中的计数也不会减少。

创建标记名服务器应用程序

通过创建仅包括 InTouch QuickScript 和标记名的应用程序，您可以建立一个 WindowViewer 实例来充当标记名服务器。您可以创建另一个仅包括窗口（和用于窗口逻辑处理的内存标记名）的应用程序。如果这些窗口仅包含远程标记名引用，则该应用程序可以充当所有过程窗口的“仓库”。在这种情况下，远程标记名引用的标记名位于充当标记名服务器的其它 WindowViewer 实例中。连接到该数据库的一个 WindowViewer 实例，其功能如同操作员工作站。这个 WindowViewer 实例可以打开任何一个窗口，并查看工厂车间任何地方的数据。例如：



远程引用对下列对象有效：

项目	有效对象：
输入链接	离散型用户输入、模拟型用户输入、字符串用户输入、垂直滑动块、水平滑动块和离散值按钮
离散型报警链接	线颜色、填充颜色和文本颜色
模拟型报警	线颜色、填充颜色和文本颜色
表达式	链接和脚本，其中可以指定离散、模拟或字符串型标记名
数据改变脚本	"Tagname[dotfields]"（标记名[点域]）
ActiveX 控件	事件、属性和方法
QuickScript	全部类型

远程引用对下列对象无效：

项目	无效对象：
历史趋势显示	“笔 1”到“笔 8”
确认报警	（因为您无法看见远程标记名进入报警状态，所以不能确认报警）。

注意：对于那些使用不支持远程标记名功能的 InTouch 先前版本创建的应用程序，不必进行转换就可以实施远程标记名引用。但是一经实施，这些应用程序将不能与先前的版本向后兼容。

WindowViewer 支持 32,767 个本地标记名引用和 *x* 个活动远程引用，其中 *x* = 61,405 减去在本地标记名字典中定义的标记名数。

远程引用最长可为 95 个字符。

远程引用也支持函数 **IOSetAccessName**（InTouch 7.0 以前版本中的 **SetDdeTopic**），此时该函数的功能与用于本地标记名时相同。

使用远程标记名引用

在客户端应用程序中，您可以实际使用三种方法来指定远程标记名引用：

1. 在任何动画链接标记名、表达式或 InTouch QuickScript 中使用 *AccessName:"item"* 引用。

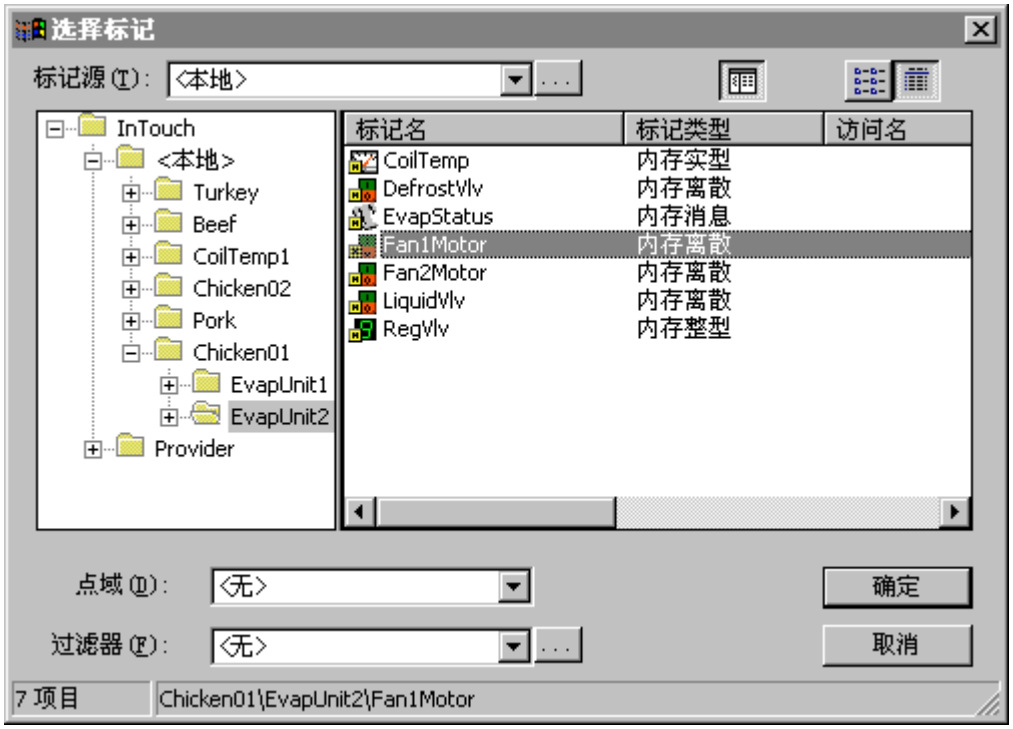
2. 导入窗口或 QuickScript，然后使用 WindowMaker“**特别**”菜单上的“**替代标记**”命令，将占位符标记名转换为远程标记名引用。

   有关转换占位符标记名的详细信息，请参阅“转换标记名为远程引用”。

   InTouch 的一个强大功能是从另一应用程序中导入窗口。当导入窗口时，其所有的脚本和动画链接也被一同导入。但是，动画链接和脚本中使用的所有标记名将自动转换成占位符。您可以将所有的占位符标记名转换为远程标记名引用，并且如果需要，可以设计一个不含本地标记名的应用程序。

   有关导入窗口或脚本的详细信息，请参阅第 2 章“使用 WindowMaker”。

3. 通过在标记浏览器中将远程应用程序配置为标记源，选择您要用于对象或 QuickScript 的远程标记名。例如：



有关从标记浏览器中选择远程标记名的详细信息，请参阅“定义标记源”。

动态地址引用 (DRA)

动态地址引用允许您用单一标记名对多个数据源编址。通过把有效引用赋给 I/O 型标记名的**.Reference** 点域，您就可以动态地改变该标记名的数据源地址。

每个 I/O 型标记名都有一个与其数据源地址关联的引用。**.Refrence** 点域的有效语法包括：

有效语法	描述
`Tagname.Reference="accessname.item"`	改变访问名和项目
`Tagname.Reference="[.]item"`	相同访问名，不同项目
`Tagname.Reference="accessname"`	改变访问名
`Tagname.Reference=""`	使标记名无效。如果没有指定访问名或项目，则使用该域的当前值。

注意：动态地址引用对远程标记名引用无效。

有关 **.Reference** 点域的详细信息，请参阅联机 *《InTouch 参考指南》*。

使用动态引用

动态引用用于查看那些临时需要观察数值的数据点，例如诊断应用中。因为标记名的数据源可以改变，动态引用不能用于那些需要永久保存或连续监视其报警状态的数据。

动态引用在传统应用上的一个很好例子是诊断应用。在这项应用中，使用单个标记名来查看 PLC 上任意一个模拟量点的输入值。这使得维护人员可以立即查看任意采样点的状态，从而实现故障诊断的目的。

- **要创建一个诊断应用程序：**

1. 创建一个 I/O 整型标记名。在本实例中，我们将标记名命名为 "AnalogSpy"。它的初始引用访问名为 **PLC1**，项目名为 **WX001。**
2. 输入 # 号，创建一个文本对象。
3. 双击 # 号，打开动画链接对话框。
4. 单击“**用户输入**”区域中的“**字符串**”。“**输入->字符串标记名**”对话框出现：
5. 在“**标记名**”框中，输入 “**AnalogSpy.Reference**”。
6. 单击“**确定**”。
7. 启动 WindowViewer，编译和运行应用程序。
8. 单击文本对象，为赋给标记名的访问名和项目名输入一个新值。

   例如，要查看访问名为 **PLC6** 的 **WX031** 项目，可以输入 **PLC6.WX031** 作为引用。
9. 如果您要确认新值是否有效，可以使用下一节所介绍的 **.ReferenceComplete** 点域。

   有关 **.ReferenceComplete** 点域的详细信息，请参阅您的联机 *《InTouch 参考指南》*。

使用 IOSetItem 函数改变引用

IOSetItem（InTouch 7.0 以前版本称为 **SetDdeItem**）函数用于设置 I/O 型标记名的 **.Reference** 点域。这个函数的基本格式是：

```
IOSetItem(TagName, AccessName, Item)
```

格式中的标记名、访问名和项目名可指定为文字字符串，或由其它 InTouch 标记名或函数提供的字符串值。例如：标记名 "MyTag1" 的**.Reference** 域可通过如下方式改为指向访问名 "Excel" 和项目 "R1C1"：

```
IOSetItem("MyTag1", "Excel", "R1C1");
```

或者：

```
Number = 1;
TagNameString = "MyTag" + Text(Number, "#");
IOSetItem(TagNameString, "Excel", "R1C1");
```

如果为访问名和项目名指定了一个空字符串 ("")，则标记名将失效。例如，使用下面的语句可使标记名 MyTag2 失效：

```
IOSetItem("MyTag2", "", "");
```

如果只为访问名指定一个空字符串，则标记名的当前项目名会被保留，而其访问名会被更新。例如，下面的语句将标记名 MyTag3 的访问名改为"Excel2"，但不影响当前项目名：

```
IOSetItem("MyTag3", "Excel2", "");
```

同样，如果只为项目名指定了一个空字符串，则标记名的当前访问名会被保留，而其项目名会被更新。例如，下面的语句将标记名 MyTag3 的项目改为 "Excel2"，但不影响当前访问名：

```
IOSetItem("MyTag4", "", "R1C2");
```

有关 **IOSetItem** 的详细信息，请参阅您的联机 *《InTouch 参考指南》*。

使用 .ReferenceComplete 验证引用

每个 I/O 型标记名都有一个**.ReferenceComplete** 域。这个离散域用于确认引用字段中所要求的项目是否反映在 **.Value** 域中。

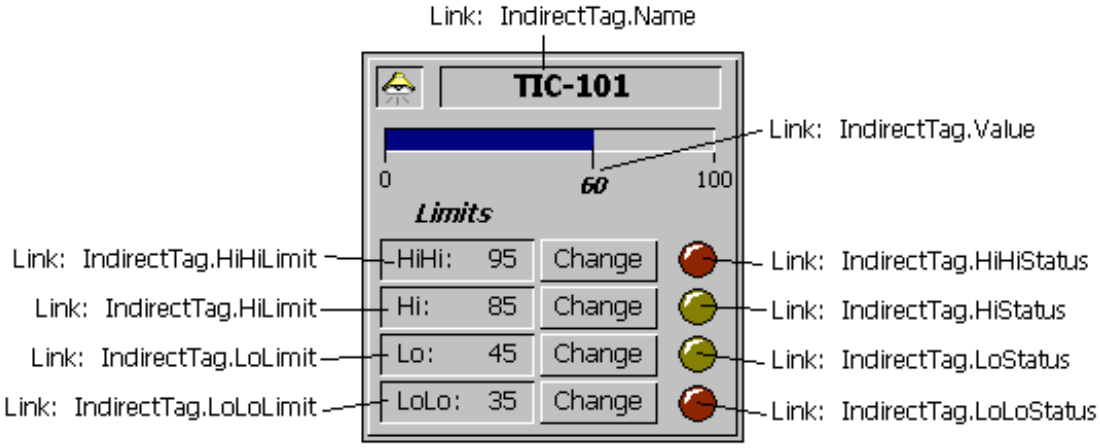
启动 WindowViewer，**.ReferenceComplete** 域初始化为假 (0)。当确认 **.Value** 域由 **.Reference** 域中指定的数据源更新时，**.ReferenceComplete** 值设为真 (1)。如果 **.Reference** 域改变，**.ReferenceComplete** 域自动设为假 (0)，当更新为新值时，再更新为真(1)。

有关 **.ReferenceComplete** 点域的详细信息，请参阅您的联机 *《InTouch 参考指南》*。

在本地标记名中使用间接标记

本节介绍使用间接标记名引用本地标记名字典的常用方法。

间接标记名允许您创建可在多个数据集中使用的“通用”标记名。例如，您可以使用通用面板来修改多个本地标记名的报警限。



您可以让许多不同的标记名使用您的报警限面板。要使面板转至适当的标记名，您可以执行下面的 QuickScript：

```
IndirectTagname.dotfield = "tagname";
```

此处，*tagname* 是在本地标记名字典中定义的实际标记名。

当执行此脚本时，与本地标记名关联的所有**点域**都可以通过间接标记名进行访问。

有关标记名点域的详细信息，请参阅联机《*InTouch 参考指南*》。

在远程引用中使用间接标记名

远程引用标记名在许多方式上与本地标记名存在不同：远程引用的语法是：

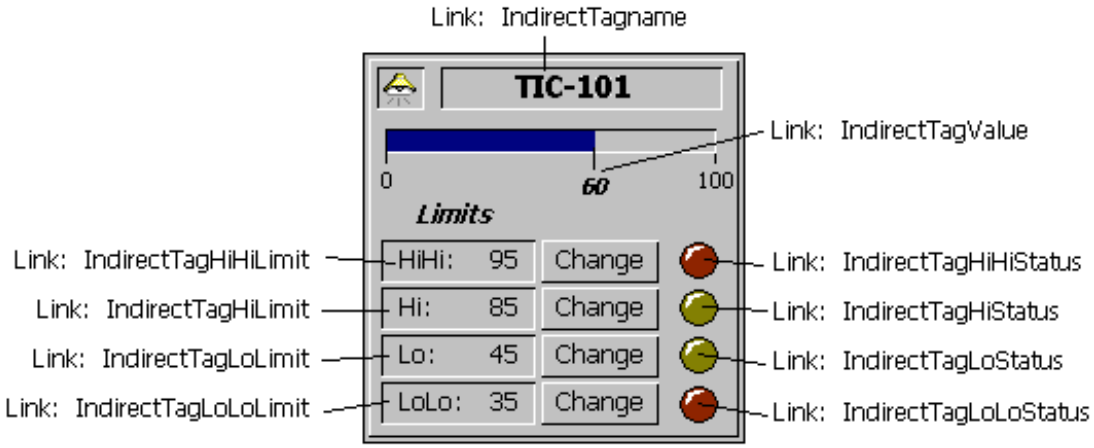
AccessName*:*Item

此处，***AccessName*** 是任何有效的 InTouch 访问名，而 ***Item*** 是受 I/O 服务器（由访问名定义）支持的任何有效项目名。

有关远程引用的详细信息，请参阅“远程标记名引用”。

当使用远程引用时，服务器会向客户端返回一个值，而不是标记名结构。此值包括时间标签和质量标记。因此，指定给远程引用的间接标记名将不能访问任何有关值、时间和质量之外的标记名**点域**。例如，间接标记名不能通过远程引用来访问报警限。

为了解决这个问题，您可以用若干个间接标记名来创建一个面板。例如：



注意此面板使用十个间接标记名，它们都使用隐式 **.Value** 引用。

有关 **.Value** 点域的详细信息，请参阅联机《*InTouch 参考指南*》。

让我们假定此报警面板被转到远程引用标记名 TIC-101，它位于一个名为 TagServer1 的远程 InTouch 节点上。InTouch 访问名配置如下：

访问名： TagServer1

节点名： TagServer1

应用程序名： View

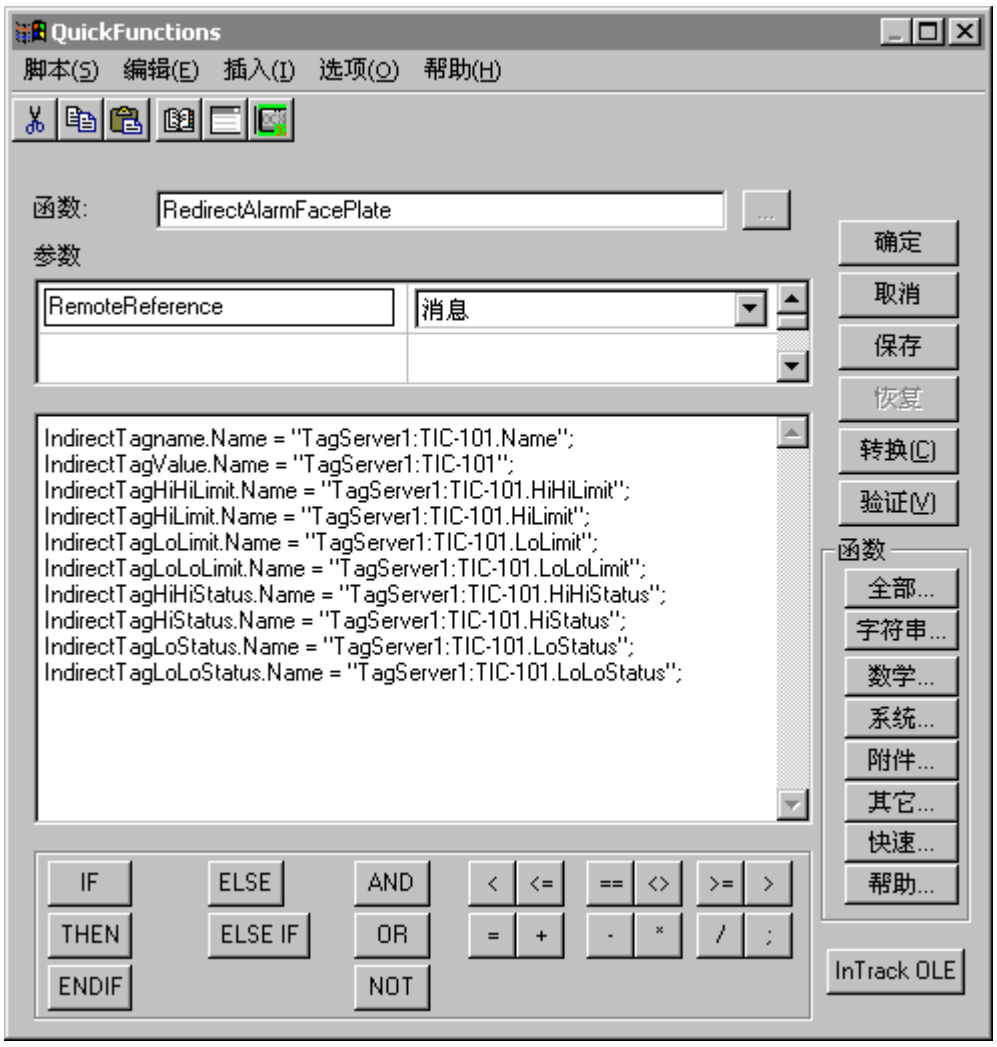
主题名： Tagname

要将面板转到远程引用标记名 TIC-101，可以执行下面的 QuickScript：

```
IndirectTagname.Name = "TagServer1:TIC-101.Name";
IndirectTagValue.Name = "TagServer1:TIC-101";
IndirectTagHiHiLimit.Name = "TagServer1:TIC-101.HiHiLimit";
IndirectTagHiLimit.Name = "TagServer1:TIC-101.HiLimit";
IndirectTagLoLimit.Name = "TagServer1:TIC-101.LoLimit";
IndirectTagLoLoLimit.Name = "TagServer1:TIC-101.LoLoLimit";
IndirectTagHiHiStatus.Name = "TagServer1:TIC-101.HiHiStatus";
IndirectTagHiStatus.Name = "TagServer1:TIC-101.HiStatus";
IndirectTagLoStatus.Name = "TagServer1:TIC-101.LoStatus";
IndirectTagLoLoStatus.Name = "TagServer1:TIC-101.LoLoStatus";
```

由于每次面板改向时都必须执行此脚本，即浪费时间又容易出错。因此，一个更好的方法是创建 InTouch QuickFunction，它允许您只写一次脚本，然后将其传递给远程引用名。

例如，使用上述的脚本命令集，您可以定义一个名为 **RedirectAlarmFacePlate()** 的 QuickFunction 函数：



您现在只需调用 QuickFunction 函数 **RedirectAlarmFacePlate()**，就可以处理整个改向操作。要实现这一点，必须由另一 InTouch QuickScript 来调用 QuickFunction 函数。例如：

```
CALL RedirectAlarmFacePlate ("TagServer1:TIC-101");
```

通常，您应该专门使用远程引用来开发新的应用程序（也可以远程引用 InTouch 中的本地标记名字典）。这可以从图形中提取数据源（本地或远程）。通过在 InTouch 应用程序中使用专有远程引用，当您以后在分布式 FactorySuite 系统中添加新的窗口或图形对象时，就可以显著地减少应用程序的维护工作。

标记浏览器

标记浏览器是一个基本工具，可用于查看和选择本地及远程标记名和标记名**点域**，无论它们是来自 FactorySuite 应用程序，还是任何其它支持 InTouch 标记名字典接口的标记源。标记浏览器允许您选择现有的标记名、添加新的标记名和查看基本标记名字典信息。您也可以使用标记浏览器来访问对话框，以便执行标记名的编辑和复制，以及从远程标记源中选择标记名（远程引用）。

当您第一次访问标记浏览器时，缺省情况下，系统会选择 **<本地>** 作为标记源。这意味着屏幕将显示本地应用程序标记名字典中的标记名。此后，屏幕将显示最近一次访问的标记源的标记名。

标记浏览器的操作模式有两种：“过滤选择模式”和“无限选择模式”。具体模式取决于您使用的访问方法。下面列出了各种模式下访问标记浏览器的基本方法：

无限选择模式

双击动画链接标记名或表达式输入框

双击 ActiveX 或向导标记名或表达式输入框

双击任意 InTouch QuickScript 窗口中的空白区域

在 InTouch QuickScript 编辑器中，选择“**插入**”菜单上的“**标记名**”命令

在 InTouch QuickScript 编辑器中，按下 **Alt+N** 键。

在“**替换标记**”对话框中，双击空白的“**新名称**”框。

在 SQL 访问“**活页表配置**”中，双击“**标记名点域名**”输入框。

过滤选择模式

在标记名字典中，单击“**选择**”按钮

当 WindowMaker 运行时，在配方管理器**单元**定义中，双击“**单元#**”列中的单元。

在运行时，单击“历史趋势设置”对话框中的任意“笔#”按钮。此时，标记浏览器将只显示用标记名字典中选定的“记录数据”选项定义的标记名。

仅当在开发时为历史趋势选定了“**允许运行时改变**”选项时，此功能才被支持。

运行时，单击任一链接到 **HTSelectTag()** 函数的对象。

有关标记浏览器模式的详细信息，请参阅“标记浏览器选择模式”。

标记浏览器的状态栏提供当前显示的标记源的以下各项的状态：

应用程序中的项目总数

当前所选项目名称

所选标记名**点域**（如果有）

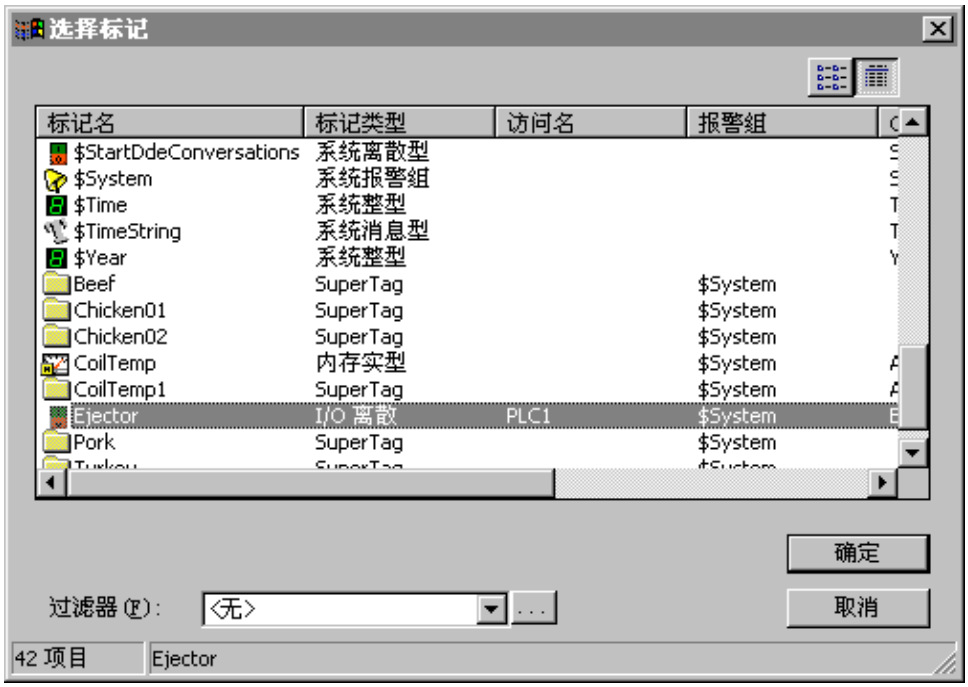
与标记源关联的访问名

标记浏览器选择模式

标记浏览器在两种选择模式下操作：“过滤选择模式”和“无限选择模式。”

过滤选择模式

当您在“**标记名字典**”对话框中单击“**选择**”，或在运行时（操作员允许在运行时改变历史趋势）为历史趋势笔选择一个新的标记名时，此模式将被激活。显示的标记名（可供选择）限于当前的 InTouch 应用程序。例如：



当您从标记名字典中访问标记浏览器，并在该视图中选择标记名时，单击“**确定**”，将出现其标记名字典定义。

注意：此模式下不能选择标记名**点域**。

无限选择模式

要访问无限选择模式，可以双击任意 InTouch QuickScript 窗口中的空白区域、动画链接标记名或表达式框、“**替换标记名**”对话框中的空白“**新名称**”框。此模式下可以显示和选定在本地或远程标记源中定义的标记名。

此种模式下也可以为标记名选定标记名**点域**。当您在此模式下选择标记名和（或）标记名**点域**时，它将自动输入到 InTouch QuickScript、动画链接标记名或表达式框，或您从中访问标记浏览器的其它位置。例如：



➢ **要选择一个点域：**

1. 单击“**点域**”箭头，打开与当前所选标记名类型相关联的**点域**列表。

   缺省条件下，所有类型的标记名最初将显示**<无>**。

   注意：以下情况下不能使用点域：从标记名字典访问标记浏览器；在运行时从“历史趋势设置”对话框中为历史趋势笔选择标记名（配置历史趋势时必须选择“允许运行时改变”选项）。

2. 单击列表中您要附加到所选标记名的“**点域**”。

   不是每种标记名类型都具有相同的**点域**。例如，**离散**类型的标记名有 **.OnMessage**，而模拟类型则没有。如果您选择**离散**类型的标记名并为其指定 **.OnMessage**，然后选择另一个**离散**型标记名，显示的**点域**列表将不会改变。但是，如果您在控制视图列表中选择另外一种类型的标记名，例如模拟类型，则显示的**点域**将恢复为**<无>**。

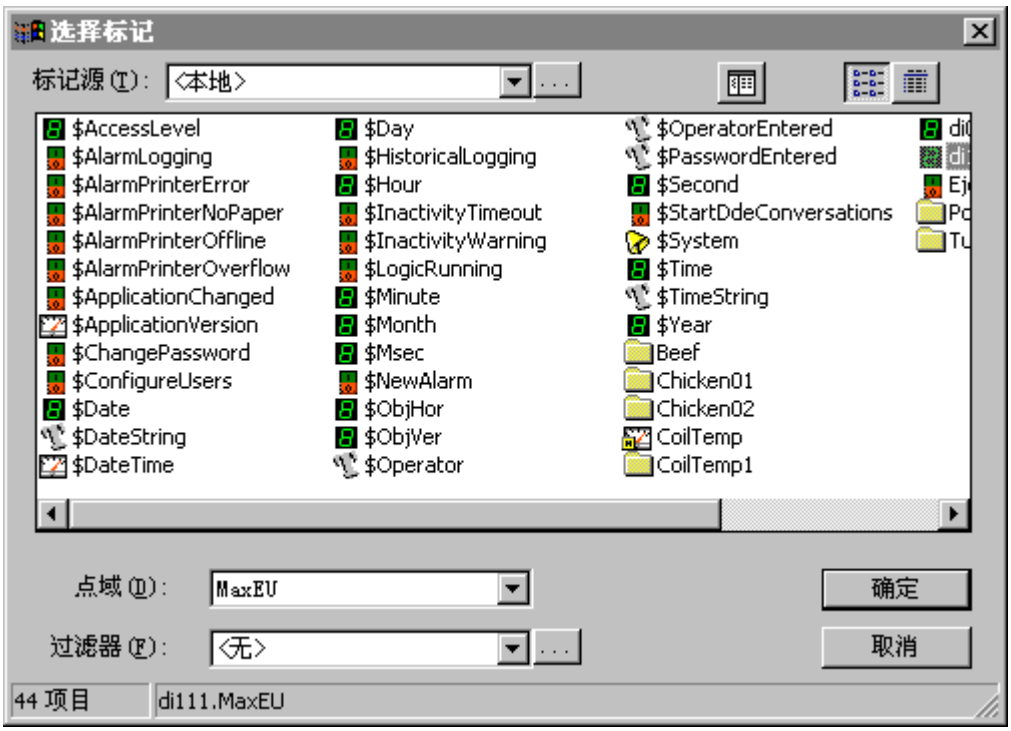
标记浏览器视图

标记浏览器支持三种控制视图：标记名列表控制、标记名详细控制和标记名树形视图控制。

列表视图

列表视图用于在当前选择模式下显示和选择标记名（如上所述）。标记名列表控制视图根据“**列表视图**”和“**详细视图**”按钮的状态，以两种视图显示标记名：

当您选择列表视图时，根据每个标记名的类型，标记名的旁边会显示不同的小图标。列表视图中不显示其它字段。例如：



要刷新“选择标记”对话框中的显示内容，按 F5。

详细视图

当选择详细视图时，屏幕将按多个列显示标记名及其详细信息。显示的详细信息包括标记名名称、标记名类型、访问名、报警组和注释。您可以通过单击各个列的标题名，按每种详细信息类型对列表排序。要选择某个项目，可以单击该项目显示内容的任何部分（而不只是标记名），则整行将辉亮显示。例如：

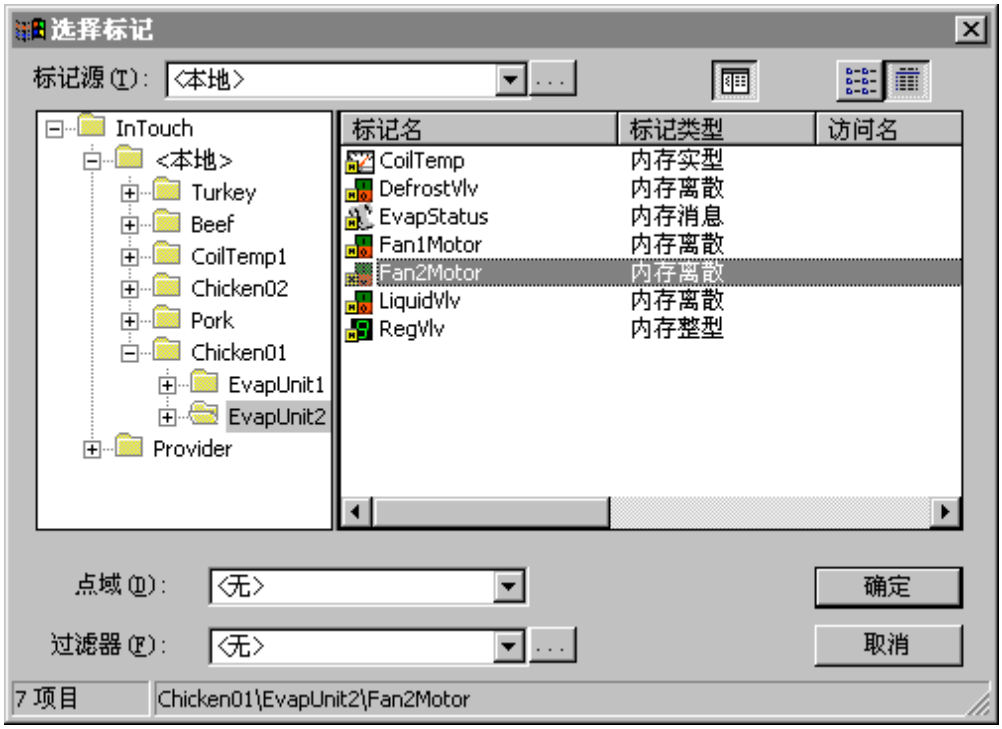


当您切换视图时，所选标记名在新视图中仍可看见并辉亮显示。

树形视图

树形视图根据“**列表视图**”和“**详细视图**”按钮的状态，以两种视图显示标记名。当您选择树形视图时，对话框的左侧会出现一个窗格。通过使用树形视图，您也可以访问任何 SuperTag 模板中的成员标记名。

当您选择树形视图时，如果**详细视图**模式处于活动状态，则标记浏览器显示如下：



- 要在树形视图中展开列表，可双击应用程序名，或单击⊞。要折叠列表，可再次双击应用程序名，或单击⊟。在树形视图窗格中双击应用程序，与在“**标记源**”列表中选择应用程序效果相同。

注意：当您在标记浏览器中向下展开不同的层次时，可以使用 BACKSPACE 键“返回”到上一层。

定义标记源

您必须在标记浏览器中定义要查看的标记源。本节向您介绍有关添加、删除或编辑标记源的步骤。当您添加或编辑标记源定义时，您可以输入以下信息，如您要与标记源的标记名关联的本地网络访问名、用户定义的应用程序名、标记源的数据源等。

注意：您也可以使用这些步骤将占位符标记名转换为远程标记名引用。

有关远程标记名引用的详细信息，请参阅“转换标记名为远程引用”。

➢  **要定义标记源：**

1. 打开标记浏览器，然后单击“定义标记源”按钮。“**定义标记源**”对话框出现：



注意：如果已经定义标记源，则它会列出在所出现的对话框中。列表包括标记源的用户定义名称、标记源的位置（路径）以及与应用程序关联的本地网络访问名。

要选择多个标记源，在单击各个名称的同时按下 SHIFT 键。要选择列表中不连续的多个标记源，在单击各个名称的同时按下 CTRL 键。

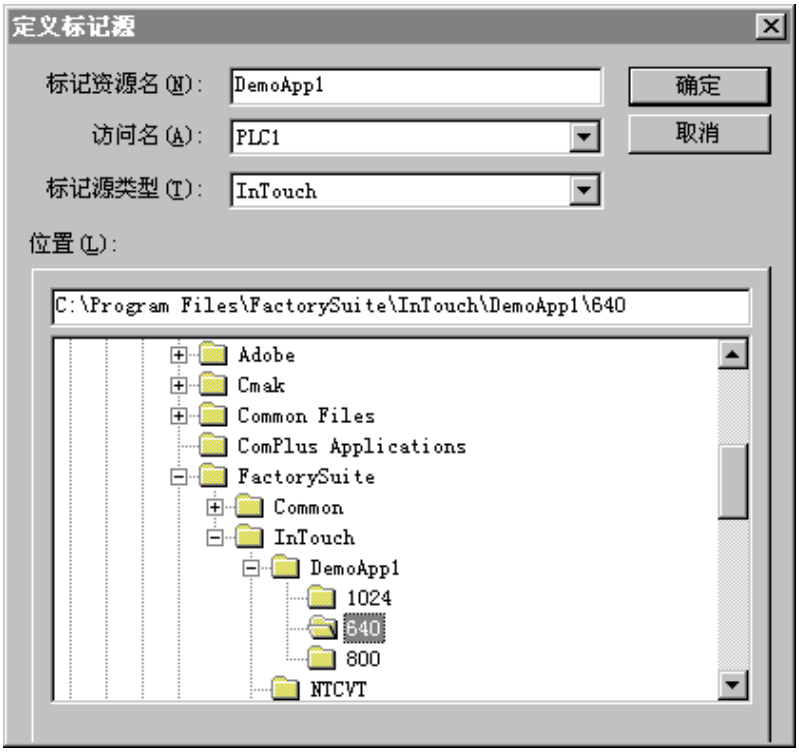
注意：当“**定义标记源**”对话框关闭时，您必须单击标记浏览器中的“**标记源**”箭头，从列表中选择新的标记源。此时标记浏览器将被刷新并显示所选标记源的标记名。

2. 要在标记浏览器中删除“**标记源**”列表中的标记源，单击“定义标记源”按钮。“**定义标记源**”对话框出现。从列表中选择标记源，然后单击“**删除**”。

3. 要编辑已定义的标记源，从列表中选定该标记源，然后单击“**编辑**”。“**定义标记源**”对话框出现，显示所选标记源的配置。

4. 要定义新的标记源，单击“**新建**”。“**定义标记源**”对话框出现。

注意：单击“**新建**”时，如果本地应用程序中没有定义访问名，则会出现一个消息框，告诉您没有定义访问名，不允许定义新的标记源（标记源必须与本地网络访问名相关联）。



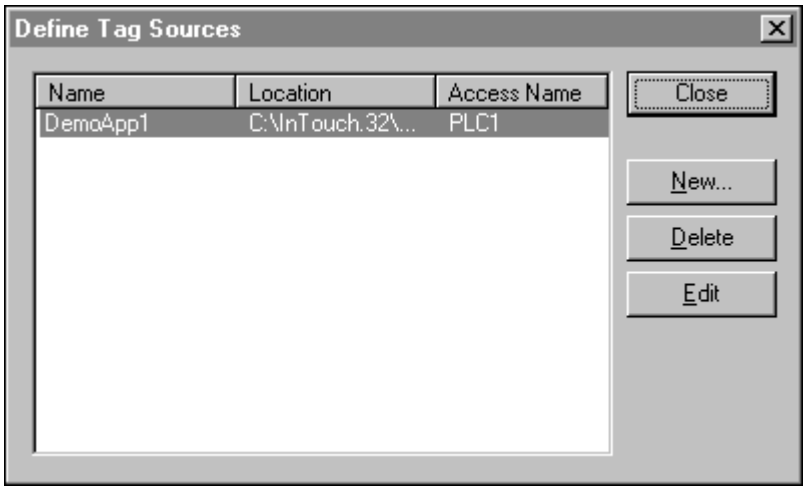
5. 在“**标记资源名**”框中，输入用于标识标记源的名称。

6. 单击“**访问名**”箭头，选择本地应用程序中您要与标记源中的标记名相关联的访问名。

7. 单击“**标记源类型**”箭头，为标记源的标记名数据库选择来源（缺省情况下，显示 **InTouch**）。

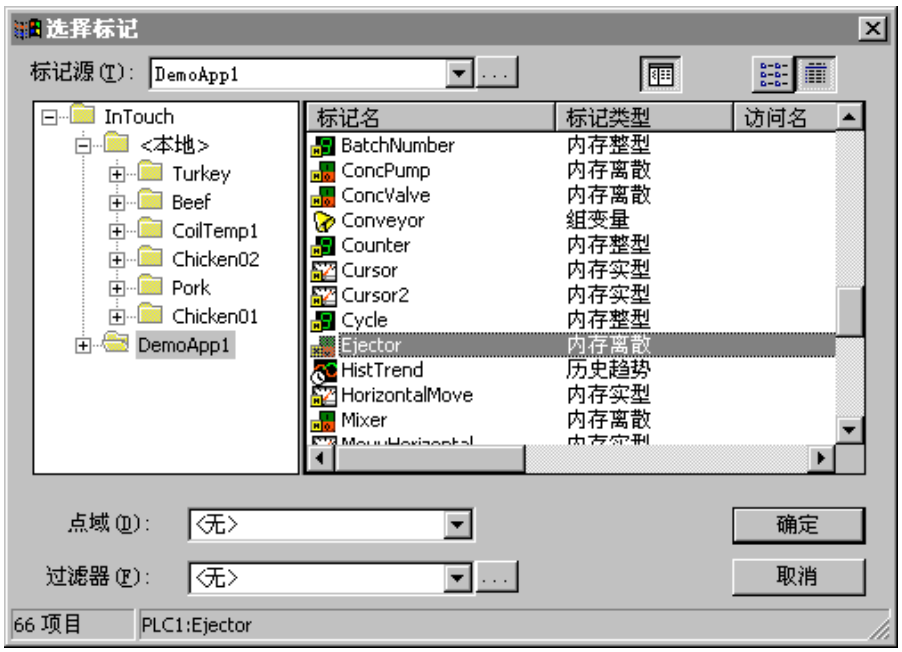
8. “**位置**”框会显示标记源的完整路径。

9. 在目录树窗格中，找到标记源，然后单击“**确定**”。“**定义标记源**”对话框重现出现，显示选定的标记源：



10. 单击“**关闭**”，标记浏览器重新出现。

11. 单击  工具，出现树形视图窗格，显示所有已定义的标记源。



如果您没用使用树形视图模式，单击“**标记源**”箭头，选择您要显示在列表中的标记源名称。标记浏览器会刷新并显示标记源的标记名。

注意：当您第一次访问标记浏览器时，缺省情况下，**标记源**会显示为**<本地>**。此后，屏幕将显示上次访问的标记源。

12. 单击“**确定**”。

定义标记浏览器过滤器

您可以使用本节中介绍的步骤，来定义用于标记浏览器内容显示的过滤器（搜索标准）。通过创建过滤器，您可以对任何标记名列表排序，并只显示符合指定标准的标记名；您可以根据**标记名**、**标记名类型**、**访问名**、**报警组**和标记名**注释**来对标记名排序；您可以使用这些项目的任意组合来设置显示标准；也可以随时保存每个过滤器实例，以便随时重新使用。

例如，如果您在标记名字典中定义了 40,000 个标记名，但只需处理其中 20 个左右赋给某个访问名或报警组的标记名，此时，您就可以创建一个过滤器，指定访问名和（或）报警组作为搜索标准，标记名必须满足这个标准才能显示在标记浏览器中。

➢  **要定义搜索过滤器：**

1. 单击“定义过滤器”按钮。**“定义标记过滤器”**对话框出现：



右击任意文本输入框，将出现一个菜单，显示您可应用到选定文本的命令。

2. 在**“过滤器名”**框中，为当前定义的过滤器输入一个唯一的名称，或单击**“过滤器名”**箭头，从列表中选择一个先前定义的过滤器（在定义过滤器时，您输入的**过滤器名**将添加到列表中）。

- 所有**过滤器选项**控件（**标记名**、**标记类型**、**访问名**、**报警组**和**注释**）允许您输入一个通配符表达式，以限制您的搜索范围。如果不使用过滤器，则屏幕会显示当前所显示的标记源的所有标记名。

  多字符通配符是星号 (*****)。例如，"Asyn*" 将搜索所有以"Asyn*"开头的标记名。

  单字符通配符是问号 (?)。例如，过滤器 "Tag?" 将搜索所有以 "Tag" 开头的四个字符的标记名。过滤器 "Tag??" 将搜索所有以 "Tag??" 开头的五个字符的标记名，等等。

  过滤器可以使用有效的 InTouch 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, !, @, -, ?, #, $, %, _, \ 和 &。

3. 在“**标记名**”框中，输入您要用作过滤器的标记名表达式。如果保留空白，系统将在过滤器定义中忽略此字段。
4. 在“**访问名**”框中，输入您要用作过滤器的本地访问名表达式。如果保留空白，系统将在过滤器定义中忽略此字段。
5. 在“**报警组**”框中，输入您要用作过滤器的报警组名称表达式。如果保留空白，系统将在过滤器定义中忽略此字段。
6. 在“**注释**”框中，输入您要用作过滤器的注释表达式。如果保留空白，系统将在过滤器定义中忽略此字段。
7. 单击“**确定**”，关闭对话框。

   - **过滤器名称**现在将出现在标记浏览器的**过滤器**列表中，选择此过滤器将只显示与过滤器中指定的标准相符的标记名。

➢ **要删除搜索过滤器：**

1. 单击“**过滤器**”箭头，从列表中选择您要删除的过滤器名称。
2. 单击“**删除**”，过滤器被被立刻删除。

InTouch 交叉引用实用程序

标记名交叉引用实用程序允许您在以下各项中决定您的标记名、远程标记名和 SuperTag 的用法：动画链接、向导、InTouch QuickScript、QuickFunction、ActiveX 控制器、脚本以及下列 InTouch 附加程序：SPC Pro、SQL Access Manager 和配方管理器。对于所有对象如向导、ActiveX 控件和动画链接等，它将显示窗口名称和链接到标记名的所有对象的坐标。它还允许您在能找到标记名的地方查看任何 QuickScript 或 QuickFunction。

为方便起见，在您执行其它任务时，标记名交叉引用实用程序可以在 WindowMaker 中保持打开。

➢ **要使用 InTouch 交叉引用实用程序：**

1. 在“**特殊**”菜单上，单击“**交叉引用**”，或在应用程序浏览器中，双击“**交叉引用**”。“**InTouch 交叉引用搜索标准**”对话框出现：



2. “**搜索标准组**”允许您限制搜索范围。要确定搜索范围，只需选择所需的选项。

搜索所有事件	搜索标记名或 SuperTag 在动画链接、InTouch QuickScript 以及所有的附件如 SPC、SQL 访问管理器、配方管理器等程序中的使用。
搜索指定的事件	只在指定的选项中搜索标记名称或 SuperTag。例如，如果您只想搜索在窗口脚本中的用法，可以只选择“在窗口脚本中的使用”。

3. 在“**过滤器**”框中，输入标识您所定义的过滤器的唯一名称，或单击“**过滤器**”箭头，从列表中选择先前定义的过滤器（在定义过滤器时，您输入的名称将添加到**过滤器**列表中）。

   过滤器编辑器控件允许您输入通配符表达式来限制标记名的搜索范围。如果不使用过滤器，将检索当前应用程序中所有标记名的信息。

   多字符通配符是星号(*****)。例如，"Asyn*" 将搜索所有以"Asyn*"开头的标记名。

   单字符通配符是问号(**?**)。例如，过滤器 "Tag?" 将搜索所有以 "Tag" 开头的四个字符的标记名。过滤器 "Tag??" 将搜索所有以 "Tag??" 开头的五个字符的标记名，等等。

   过滤器可以使用有效的 InTouch 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, !, @, -, ?, #, $, %, _, \ 和 &。

   如果在“**过滤器**”框上右击鼠标，一个菜单会出现，显示可应用于选定文本的命令。

4. 单击“**搜索**”，根据指定的查看标准开始交叉引用搜索。

查看交叉引用搜索结果

当执行交叉引用搜索时，“**InTouch 交叉引用实用程序**”对话框会出现，列出根据您指定的**过滤器**找到的所有使用实例。

如果没有使用过滤器，则会显示在当前应用程序的标记名字典中定义的所有标记名。例如：



交叉引用实用程序图标

下面简要介绍可能出现在 InTouch 交叉引用实用程序中的各种图标：

图标	描述
	标记名或 SuperTag 被赋给 InTouch 对象，或用于存储 InTouch QuickScript、向导或附加程序中的值。单击以展开该层的视图。
	单击以折叠被展开层的视图。
	所显示的标记名或 SuperTag 在应用程序的标记名字典中定义，但它不能赋给对象。
	所显示的标记名或 SuperTag 用于动画链接或 InTouch QuickScript。双击或单击⊞以展开视图。
	所显示的标记名或 SuperTag 被赋给动画链接。双击或单击⊞显示窗口名称和赋给动画链接的对象的坐标。
	所显示的标记名或 SuperTag 用于应用程序脚本。双击或单击⊞以展开视图并显示使用该标记名和 SuperTag 的应用程序的脚本类型。
	表示下列各项：所有应用程序**启动时**、**运行期间**和**关闭时**脚本；窗口**显示时**、**显示期间**和**隐藏时**脚本；按键的**按下时**、**按下期间**和**放开时**脚本。双击脚本以进行查看。
	所显示的标记名或 SuperTag 用于窗口脚本。双击或单击⊞展开视图并显示该脚本的窗口名。双击任一窗口名以查看其脚本。
	所显示的标记名或 SuperTag 用于数据改变脚本。双击或单击⊞展开视图，然后双击任何所列脚本以进行查看。
	所显示的标记名或 SuperTag 用于条件脚本。双击或单击⊞展开视图并显示该脚本的条件和类型。例如“**$Hour==12 为真时**”。双击任何所列脚本以进行查看。
	所显示的标记名或 SuperTag 用于键脚本。双击或单击⊞展开视图并显示赋给脚本和脚本类型的键。例如，“**F2 按下时**”。双击任何所列脚本以进行查看。
	所显示的标记名或 SuperTag 用于 QuickFunction。双击或单击⊞展开视图并显示使用该标记名和 SuperTag 的 QuickFunction。单击以展开视图并显示使用该标记名和 SuperTag 的 QuickFunction 的名称。双击任何所列脚本以进行查看。

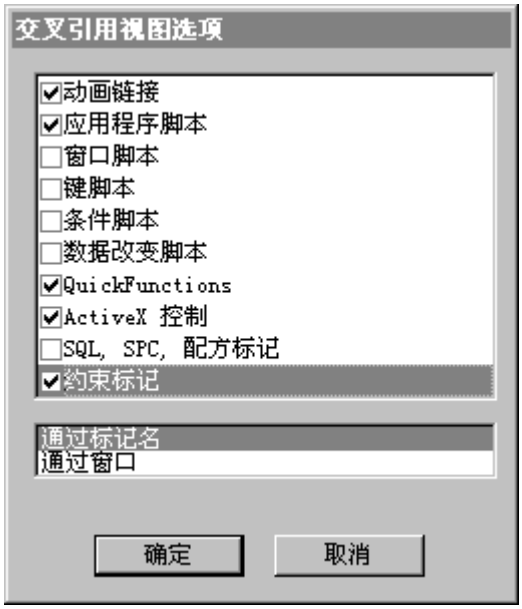
图标	描述
	所显示的标记名或 SuperTag 用于 ActiveX 事件脚本。双击或单击⊞以展开视图并显示 ActiveX 事件脚本。
	当由**窗口**交叉引用时，该图标显示在使用了所显示的标记名和 SuperTag 的窗口名之前。双击或单击⊞以查看窗口中的所有标记名。
	所显示的标记名或 SuperTag 用于 SPC Pro 应用程序。双击或单击⊞以查看其中使用了该标记名和 SuperTag 的 SPC Pro 数据集 。
	所显示的标记名或 SuperTag 用于 SQL 应用程序中。双击或单击⊞以查看其中使用了该标记名或 SuperTag 的 SQL 活页表。
	所显示的标记名或 SuperTag 用于 Recipe 管理器应用程序。
	所显示的标记被用作报警约束器。双击或单击⊞以查看作为约束器的所有标记名。

改变交叉引用搜索标准

如有需要，在完成初始交叉引用搜索后，您可以通过修改原始搜索选项来缩小您的搜索范围。

➢ **要改变搜索选项：**

1. 在“**InTouch 交叉引用实用程序**”对话框中（执行初始搜索后显示），单击“**选项**”。“**InTouch 交叉引用视图选项**”对话框出现：



2. 选择您要修改以用于新搜索的搜索标准选项。

   此处的可用选项取决于您原先在“**InTouch 交叉引用搜索标准**”对话框中选定的**搜索标准**。如果您选择了“**搜索所有事件**”，则所有搜索标准选项都将可用。如果您选择了“**搜索指定的事件**”，则只有您指定的那些选项可用。单击“**取消**”以改变**搜索标准**选择。“**InTouch 交叉引用实用程序**”对话框重新出现。单击“**搜索**”，选择新的“**搜索标准**”选项。

3. 在对话框底部的列表中，选择您要在树形视图中显示标记名还是窗口名，然后单击“**确定**”。

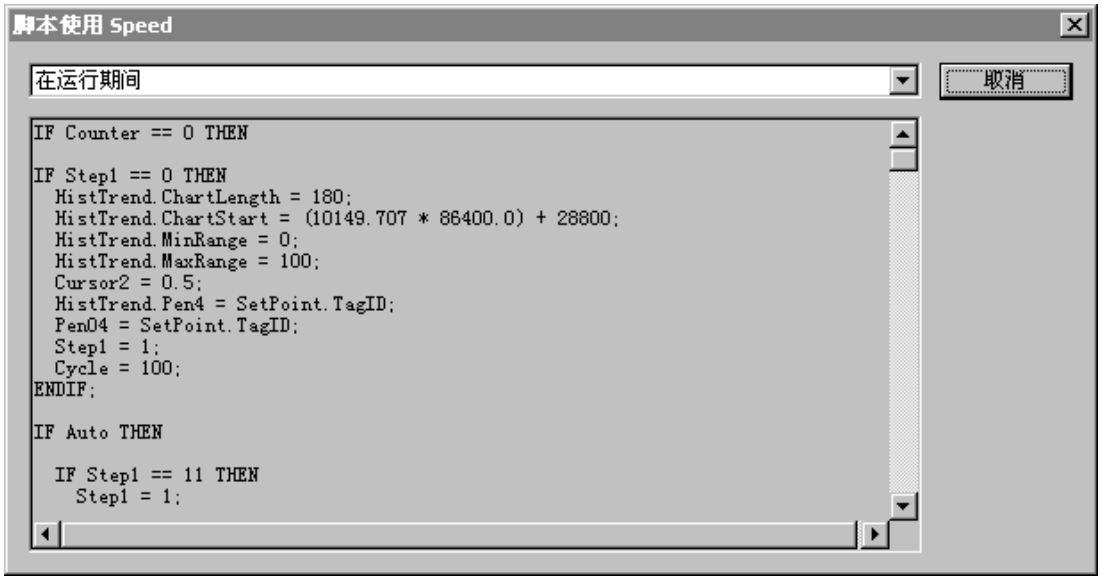
通过标记名交叉引用

按字母顺序列出根据指定的搜索标准找到的所有标记名（缺省视图）。根据您指定的搜索标准，该视图允许您查看在窗口、动画链接、脚本和附件应用程序中找到的所有标记名的使用情况。

您可以双击所显示的标记名，然后双击“**动画链接使用**”来展开视图。当您展开该视图时，将显示窗口名和链接到该标记名的对象位置（坐标）。例如：



此外，您可以双击标记名，然后再双击其关联的任一脚本，以在“**脚本使用*标记名***”对话框中打开它。



屏幕顶部的列表框显示了与所选标记名关联的所有脚本。单击箭头打开列表以选择要查看的其它脚本。对于应用程序、窗口、键和条件脚本，列表包含所有使用该标记名的脚本名；对于数据改变脚本，则只列出标记名。对于 QuickFunction，列表将包含所有 QuickFunction 名。

注意：这是 QuickScript 的只读视图。您不能在该对话框中编辑 QuickScript 文本。但是，您可以复制 QuickScript 任一部分或全部，然后将它粘贴到任一 InTouch QuickScript 编辑器窗口中。

要将 QuickScript 复制到窗口剪贴板，单击“脚本”，然后单击“**全部选择**”，再次右击并单击“**复制**”。您也可以执行 Windows 复制命令 (**Ctrl+C**)。

要将复制的脚本粘贴到另一 InTouch QuickScript 中，在应用程序浏览器的“**脚本**”下，双击您要创建的脚本类型。QuickScript 编辑器出现。在“**编辑**”菜单上，单击“**粘贴**”，或右击脚本窗口，然后单击“**粘贴**”。您也可以运行 Windows 的粘贴命令 (**Ctrl+V**)。

单击“**取消**”关闭对话框。

窗口名的交叉引用

先按窗口名，然后按窗口中的标记名对显示对象进行排序。例如：



注意：该视图只显示窗口中使用的标记名，不包含在动画链接、脚本等中的使用情况。

单击“**展开视图**”，显示当前所显示标记名或窗口的所有可用视图级别。例如：



单击“**收敛视图**”使对话框返回缺省模式。

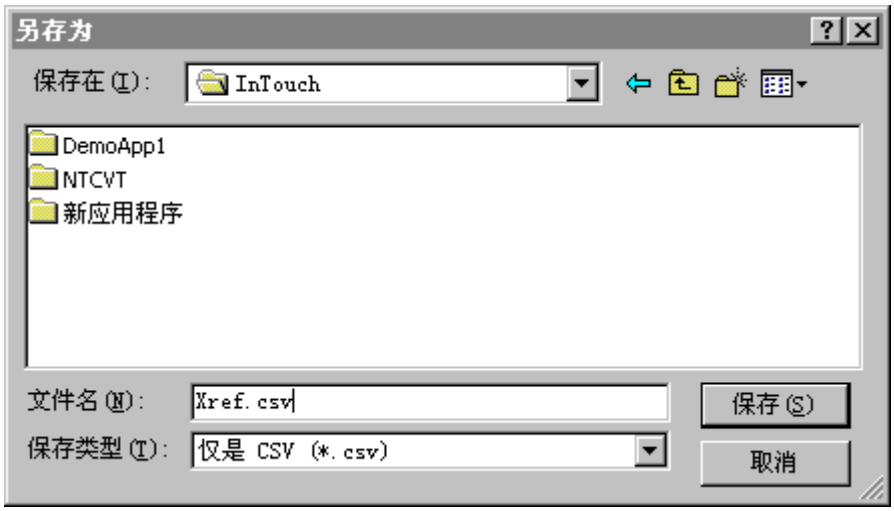
单击“**关闭**”退出交叉引用实用程序。

保存交叉引用文件

交叉引用文件可以保存并稍后在支持逗号分隔变量 (.CSV) 文件的任意文本编辑器程序中进行查看。储存在交叉引用文件中的信息对应于当前显示在“**InTouch 交叉引用实用程序**”对话框中的信息。

➢ **要保存交叉引用文件：**

1. 在“**InTouch 交叉引用实用程序**”对话框中，单击“**另存为**”。“**另存为**”对话框出现：



2. 在“**文件名**”框中，输入您要保存的交叉引用文件的文件名。

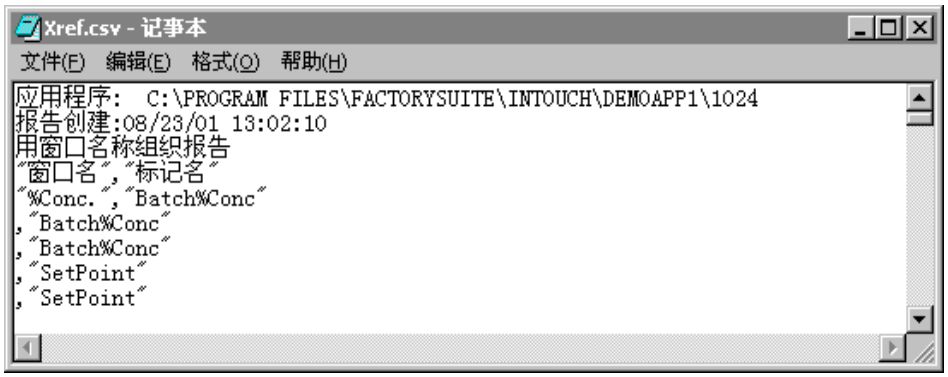
注意：文件必须以 .csv 格式保存。

3. 单击“**保存**”。

打印交叉引用文件

您可以在任何一个支持.CSV 文件格式的文本编辑器程序中打开交叉引用.CSV 文件，并将其作为报告打印。

例如，如果您在记事本中打开此文件，它将显示如下：



要在记事本中打印该文件，在“**文件**”菜单上单击“**打印**”。

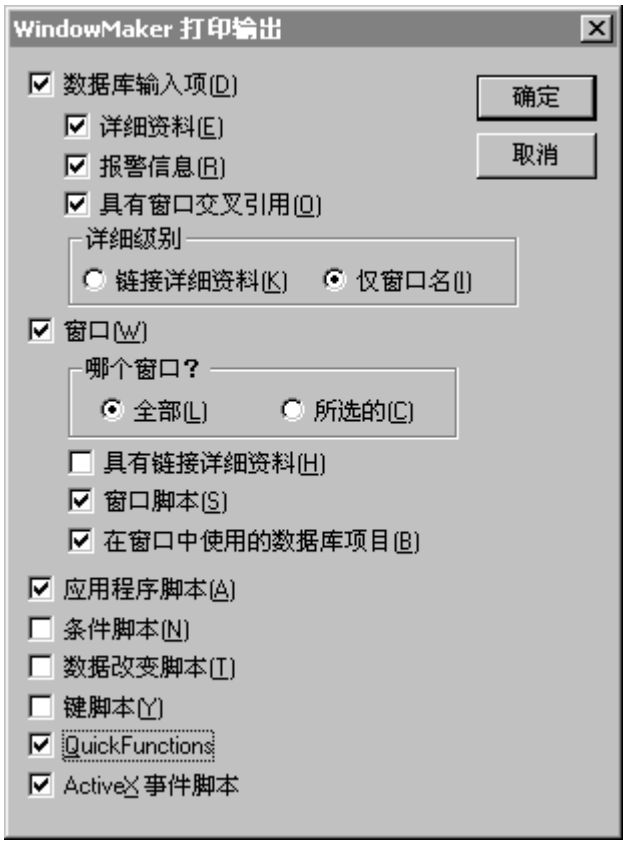
打印标记名字典详细资料

除了打印已保存的交叉引用.CSV 文件外，您还可以打印标记名字典详细资料、报警信息、链接详细资料以及脚本的列表。打印标记名字典详细资料有助于您决定标记名的使用。

注意：Windows 缺省打印机将用于产生 80 列宽的打印输出。缺省打印机的选择和设置通过 Windows 控制面板来进行。

➢ **要打印标记名字典详细资料：**

1. 在“**文件**”菜单上，单击“**打印**”。“**WindowMaker 打印输出**”对话框出现：



2. 如果您要打印所有数据库信息，选择“**数据库输入项**”。如果您选择“**数据库输入项**”，下面三个选项将被激活：

   选择“**详细资料**”以在报告中包括数据库详细资料。

   选择“**报警信息**”以在报告中包括数据库报警信息。

   选择“**具有窗口交叉引用**”以打印所有具有窗口交叉引用的数据库项目。选择此选项将激活“**详细级别**”选项：

选择“**链接详细资料**”以打印使用标记名的位置和动画链接详细资料。

选择“**仅窗口名**”以只打印交叉引用窗口名。

3. 选择“**窗口**”以打印在应用程序窗口中使用的数据库输入项列表。如果您选择“**窗口**”，下面三个选项将被激活：

选择“**全部**”以打印应用程序中所有窗口的数据库输入项。

选择“**所选的**”以只打印指定窗口的数据库输入项。“**打印窗口**”对话框将出现：



4. 选择您要打印的窗口，然后单击“**确定**”（缺省条件下，当对话框出现时，所有的窗口名均被选定）。

选择“**具有链接详细资料**”以打印窗口的链接详细资料。

选择“**窗口脚本**”以打印与窗口关联的脚本。

选择“**在窗口中使用的数据库项目**”以打印窗口中使用的标记名。

选择“**应用程序脚本**”以打印应用程序脚本。

选择“**条件脚本**”以打印与窗口关联的条件脚本。

选择“**数据改变脚本**”以打印与窗口关联的数据改变脚本。

选择“**键脚本**”以打印与窗口关联的键脚本。

选择“**Quick Function**”以打印 QuickFunction。

5. 单击“**确定**”开始打印您的报告。

从字典中删除标记名

InTouch 为数据库中的每个项目保留使用计数。该计数对某些操作来讲不会自动更新，如删除窗口、在链接或脚本中改变标记名等等。在这些情况下，InTouch 认为该标记名继续由应用程序使用，并且不允许您将其删除。所以，要删除该标记名，您需要更新使用计数。

- **要删除不再使用的标记名：**

1. 如果 WindowViewer 正在运行，先将其关闭。
2. 在“**特别**”菜单上，单击“**标记名字典**”。“**标记名字典**”对话框出现。
3. 单击“**选择**”。“**选择标记名**”（标记浏览器）出现。
4. 选择您要删除的标记名，然后单击“**确定**”。“**标记名字典**”对话框出现，显示所选标记名的定义。
5. 单击“**删除**”。

注意：在 WindowViewer 运行期间或 InTouch 认为标记名正被用于应用程序时，“**删除**”按钮不可用。

通过 InTouch 交叉引用实用程序，您可以确定标记名的使用位置（在**“特别”**菜单上，单击“**交叉引用**”）；或者您可以打印窗口中使用的所有标记名链接的报告（在“**文件**”菜单上，单击“**打印**”）。

有关打印报告的详细信息，请参阅“打印标记名字典详细资料”。

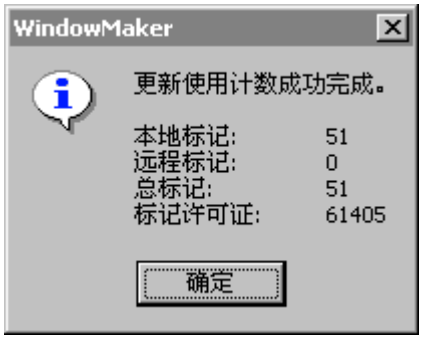
更新使用计数

因为 InTouch 会为数据库中的每个项目保留使用计数，所以您需要更新使用计数，将不再使用的所有标记名设置为零，InTouch 才允许您将其删除。

➢ **要更新标记名使用计数：**

1. 关闭所有的窗口。
2. 在“**特别**”菜单上，单击“**更新使用计数**”。

   一个消息框出现，告诉您更新使用计数需要持续一段时间。此时您可以取消此命令或继续执行。
3. 单击“**是**”继续更新使用计数。一旦系统完成更新使用计数，将出现下面的对话框：



4. 单击“**确定**”。

删除不再使用的标记名

在完成更新使用计数后，InTouch 将允许您删除所有不再使用的标记名。您可以通过在“**标记名字典**”对话框中打开每个标记名然后单击“**删除**”来将其删除，也可以通过使用“**删除未使用标记**”命令一次删除一个或多个标记名。

➢ **要删除多个不再使用的标记名：**

1. 在“**特别**”菜单上，单击“**删除未使用标记**”。“**选择名称以删除**”对话框出现：



2. 选择您要删除的标记名，然后单击“**删除**”。
3. 单击“**全部**”，删除所有显示的标记名。

警告：报警状态的标记名没有使用计数，可能会被意外删除。要确保报警状态的标记名包括在使用计数中，您需要将其用于窗口或 QuickScript 中。

显示标记名使用计数

您可以在 WindowMaker 的菜单栏中显示标记名字典中当前定义的本地标记名数目（标记名计数不包括内部系统标记名或远程标记名引用）。

➢ **要显示标记名计数：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**WindowMaker**”。“**WindowMaker 属性 - 通用**”属性页出现：

   要快速访问对话框，在应用程序浏览器的“**配置**”下，双击“**WindowMaker**”。



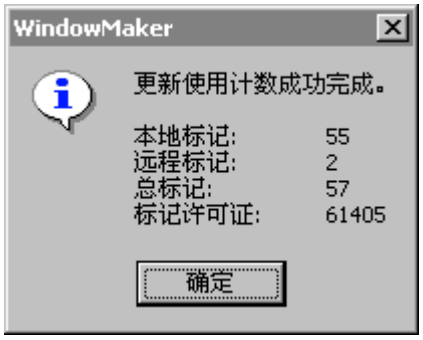
2. 选择“**显示标记计数**”。
3. 单击“**确定**”。
4. 标记名字典中定义的本地标记名总数将显示在 WindowMaker 菜单栏的一端。

   要更新所显示的标记名计数，必须读入整个标记名字典。因此，当启用此选项时，如果对标记名字典进行更改，系统性能可能会下降。如果标记名字典较大，您不应该选择此选项。

➢ **要确定远程标记名的使用情况：**

1. 在“**特别**”菜单上，单击“**更新使用计数**”。
2. 系统将更新标记名的使用信息，然后显示下面的对话框：

   更新使用计数可能需要持续一段时间。



3. “**远程标记**”行将显示应用程序中使用的远程标记名的数目。
4. 单击“**确定**”。

替换标记名

当您复制对象时，所得复件将是原件的完全拷贝，包括链接、动画、脚本等等。然而，如果您要在所复制的对象上使用不同的标记名，则必须更改该标记名。在 WindowMaker 中，这称为“替换标记名”。您可以随时选择和更改任何对象的标记名，也可以选择多个对象并同时更改其标记名。

- 如果您在 WindowViewer 运行时更改对象的标记名，则只有在重新启动 WindowViewer 后，所作更改才会生效。

如果您的系统许可证支持有限的标记名数目，您也可以把本地标记名转换成远程引用标记名，来减少本地标记名字典中定义的标记名数目。

➢ **要将对象标记名变成另一本地标记名：**

1. 选择您要更改其标记名的对象，然后在“**特别**”菜单上，单击“**替代标记**”。“**替换标记名**”对话框出现：

   - 要快速访问该对话框，右击所选对象之一，指向“**替换**”，然后单击“**替代标记**”。



2. 在“**新名**”对话框中，输入新的标记名，然后单击“**确定**”。与所选对象关联的标记名将自动更改。

   - 如果右击“**新名**”框，一个菜单会出现，显示可应用于文本的命令。

   如果在“**新名**”框中双击标记名，则会出现其在标记名字典中的定义。

   如果您清除标记名，然后双击空白的“**新名**”框，则标记浏览器将出现。

转换占位符标记名

当您索引标记名（以移除其服务）或从当前应用程序导入/导出一个窗口或 QuickScript 时，所有与该窗口或 QuickScript 关联的标记名将随窗口一起移动，但是它们不会添加到新应用程序的数据库中，而是自动标记为“占位符”（索引）标记名。您必须转换这些占位符，并且如有必要，在新应用程序的标记名字典中对其进行定义。例如：



在本例中，要将占位符标记名转换为本地标记名，单击“**转换**”。

当您导入一个窗口时，如果任何一个标记名（远程标记名除外）尚未在本地标记名字典中定义，系统会提示您在转换之前先行定义。此时，单击“**确定**”。“**标记名字典**”对话框将出现，您现在可以定义这些标记名。

注意标记名前的占位符 **?d:**、**?i:** 、**?m:**和 **?r:**。它们指明了标记名原先定义的类型：

占位符符号	标记名类型
D	离散型
I	整型
M	消息型
R	实型

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

将标记名转换为远程引用

您可以在“**替换标记名**”对话框中使用多种方法，来将占位符（或本地）标记名转换为远程标记名引用。您可以直接输入远程标记名引用；转换与所导入的窗口关联的占位符标记名；或者，打开标记浏览器并显示标记源的标记名字典以选择远程标记名引用。

➢ **要手动转换标记名为远程标记名引用：**

1. 选择与您要转换成远程标记名引用的本地标记名相关联的对象，然后在“**特别**”菜单上，单击“**替代标记**”。“**替换标记名**”对话框出现：

   要快速访问该对话框，选择所有对象，然后右击所选对象之一，指向“**替换**”，然后单击“**替换标记**”。



2. 在“**新名**”框中，选择您要更改的每个标记名，然后输入远程标记名引用：



3. 单击“**确定**”。

如果您使用这种方法并且不再需要在本地标记名字典里定义这些原始标记名，您可以更新标记名使用计数，然后删除不再使用的标记名。

有关详细信息，请参阅“从字典中删除标记名”。

➢ **要将所导入窗口的标记名转换为远程引用：**

1. 打开所导入的窗口，选择所有对象，然后在“**特别**”菜单上，单击“**替代标记**”。“**替换标记名**”对话框出现。

   有关导入窗口和脚本的详细信息，请参阅第 2 章“使用 WindowMaker”。

   要快速访问该对话框，单击 F2 键（选择窗口中的所有对象），右击所选对象之一，然后指向“**替换**”，再单击“**替代标记**”。



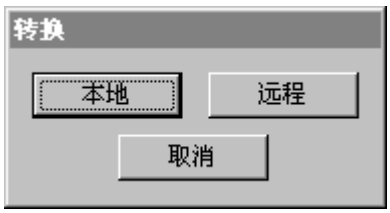
注意：“**索引**”按钮可将链接转换成占位符，占位符使动画链接不能指向本地标记名，从而使之不被意外删除。

您也可以使用“**替代标记**”命令来将本地标记名转换成远程标记名引用。为此，选择与本地标记名关联的对象，然后选择“**替代标记**”命令以在“**替换标记名**”对话框中显示。单击“**索引**”将其变为定位符标记名，然后单击“**转换**”并按下述步骤操作。

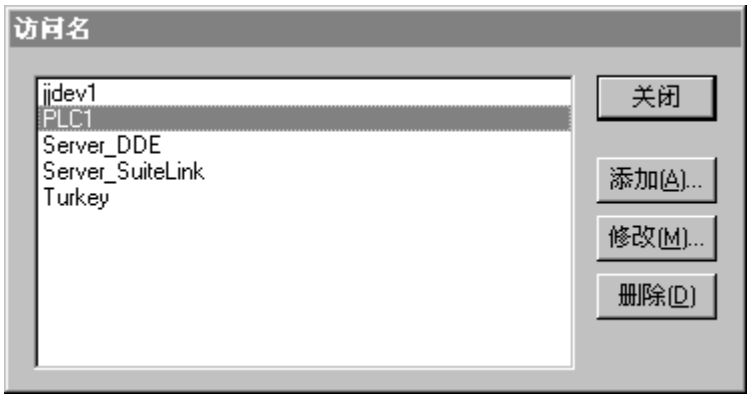
如果您在上述任一选项中使用“**索引**”，而且不再需要在本地标记名字典中定义这些原始标记名，则可以更新标记名使用计数，然后删除不再使用的标记名。

有关详细信息，请参阅“从字典中删除标记名”。

2. 单击“**转换**”。“**转换**”对话框出现：



3. 单击“**远程**”。“**访问名**”对话框出现，列出您在本地应用程序中定义的所有访问名。



4. 双击列表中的访问名，或选择访问名，然后单击“**关闭**”。

   要验证访问名是否正确，单击“**修改**”进行查看。

   如果当前尚未定义指向标记源的访问名，单击“**添加**”以现在进行定义。访问名**必须**包含应用程序所在的远程节点。

5. “**替代标记**”对话框中显示的**所有**标记名将自动转换成远程标记名引用（字首为您选定的访问名）。例如：



6. 单击“**确定**”。

   该步骤对导入的 QuickScript 作用相同，不同的是您要在 InTouch QuickScript 编辑器中打开 QuickScript，然后单击“**转换**”。

   通过从另一应用程序导入窗口或 QuickScript，并把所有与动画链接或 QuickScript 关联的标记名转换成远程标记名引用，您不必在本地标记名字典定义单个标记名，就可以立刻接收来自数以百计的远程标记名的数据。

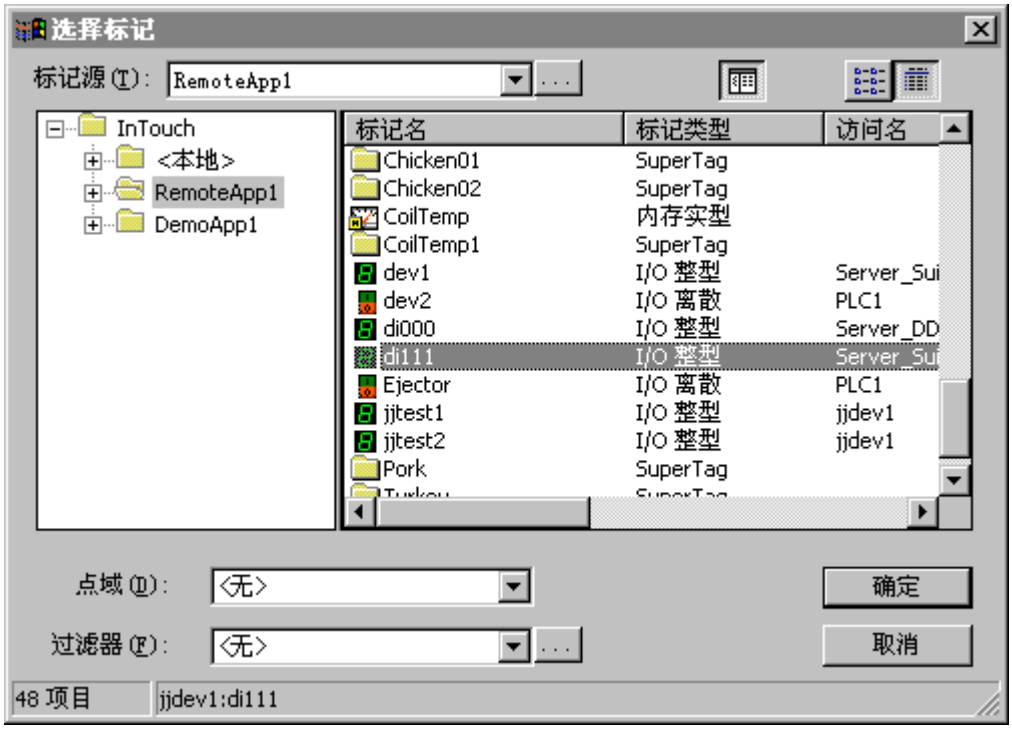
➢ **要在标记浏览器中选择远程标记名引用：**

1. 选择与您要转换成远程标记名引用的本地标记名相关联的对象，然后在“**特别**”菜单上，单击“**替代标记**”。“**标记名字典**”对话框出现。

   要快速访问该对话框，选择所有对象，然后右击所选对象之一，指向“**替换**”，再单击“**替代标记**”，。

2. 清除“**新名**”框中您要用远程标记名引用来代替的标记名，然后双击“**新名**”框。“**选择标记**”对话框出现：

3. 单击  工具以显示树形视图窗格。



4. 如果已在标记浏览器中定义标记源，请在树形视图里选择该标记源以显示其标记名。

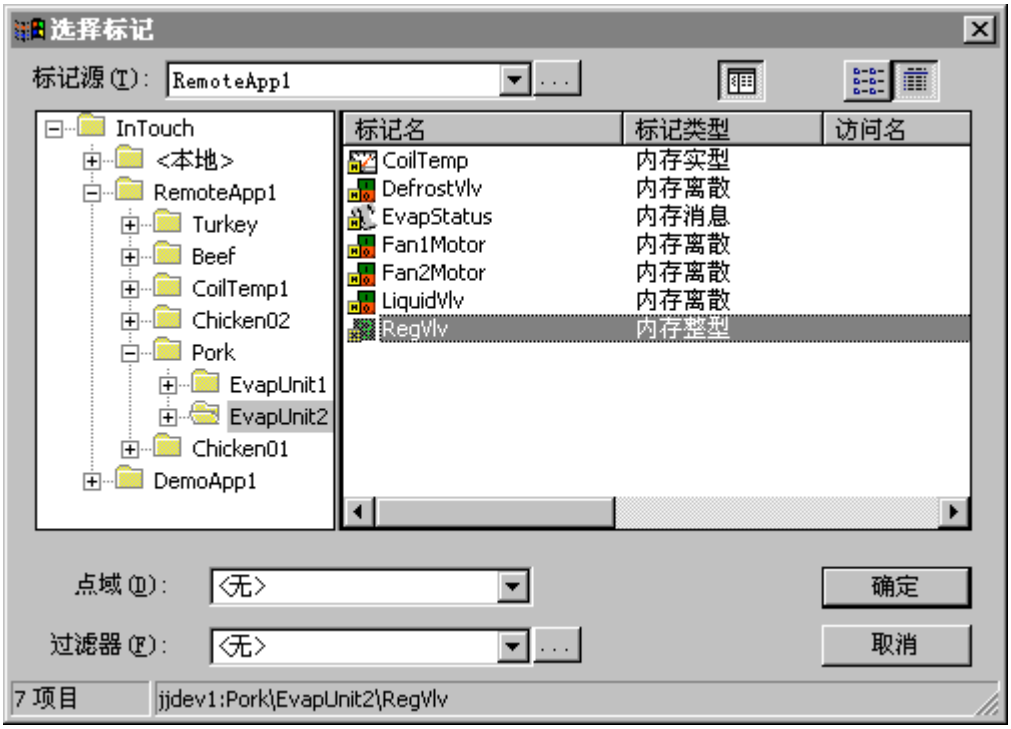
   双击应用程序名或单击⊞，展开树形视图。

注意：如果当前尚未定义标记源，您可以现在进行定义。

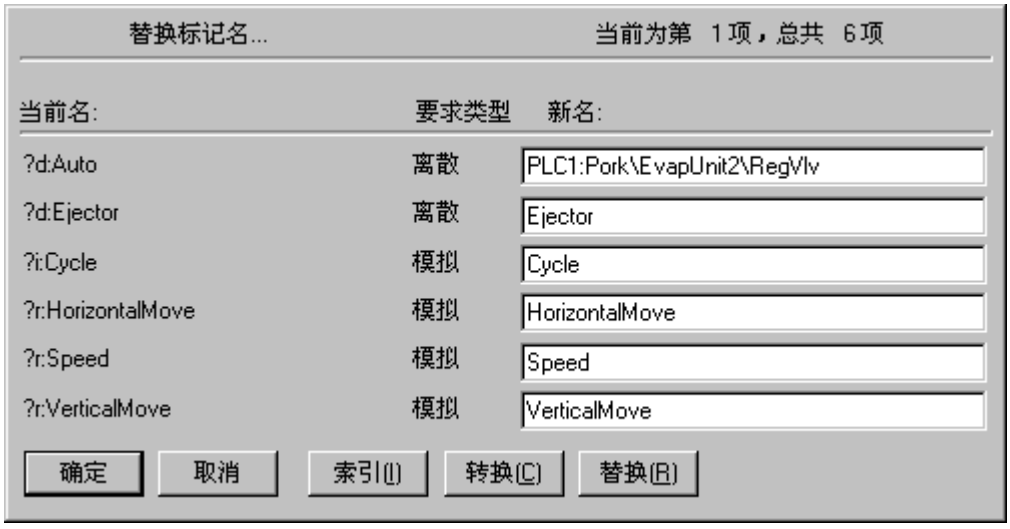
有关定义标记源的详细信息，请参阅“定义标记源。

5. 双击您要使用的远程标记名，或选择该标记名，然后单击“**确定**”。

您也可以选择 SuperTag 成员标记名。例如：



“**替换标记名**”对话框出现，显示远程标记名引用：



6. 单击“**确定**”关闭对话框并将远程标记名与所选对象相关联。

对每个您要在“**替换标记名**”对话框中改变的标记名重复此步骤。

➢ **要替换标记名：**

1. 选择您要更改其标记名的所有对象。
2. 在“**特别**”菜单上，单击“**替代标记**”。“**替换标记名**”对话框出现：

   要快速访问该对话框，右击对象，指向“**替换**”，然后单击“**替代标记**”。



3. 单击“**替换**”。“**替换文本**”对话框出现：



4. 在“**旧文本**”框中，输入您要替换的标记名部分。
5. 在“**新文本**”框中，输入替换文本。
6. 单击“**确定**”。“**替换标记名**”对话框重新出现，显示对该标记名所做的修改：



7. 单击“**确定**”。所有包含已替换文本的选定对象的标记名将自动更改。

定标 I/O 标记名

所有 I/O 型标记名都从另一 Windows 应用程序如 Excel 和 I/O 服务器中接收标记名值。这个值称为“原始”值。当您在标记名字典中定义标记名时，必须输入数值的“最大原始数据”和“最小原始数据”。数据库使用这些值来限制从 I/O 设备取得的实际原始值。例如，如果您将“最小原始数据”值设置为 50，但从 I/O 服务器收到的实际值是 0，那么数据库将强制原始值为 50。

InTouch 不显示原始值，而是显示工程单位（EU）。当您在标记名字典中定义 I/O 型标记名时，必须指定“最小工程单位”和“最大工程单位”值。这些值用于将原始值定标为所显示的值。如果您不需要定标或 I/O 设备已做了定标，请将最大/最小工程单位值设置为等于最大/最小原始数据值。

例如，我们假定一个连接到 PLC 寄存器的流量发送器，在无流量时其值为零，当流量为 100% 时其值为 9999。输入下面的值：

最小工程单位 = 0　　最大工程单位 = 100
最小原始值 = 0　　最大原始值 = 9999

原始值 5000 将显示为 50。

我们再假定一个连接到 PLC 寄存器的流量发送器，当无流量时值为 6400，当流量为 300GPM 时值为 32000。

最小工程单位 = 0　　最大工程单位 = 300
最小原始值 = 6400　　最大原始值 = 32000

在这种情况下，原始值 12800 将显示为 150；原始值 6400 将显示为 0，原始值 0 将显示为 0（所有超出最大和最小值边界的值均受到限制）。

当 I/O 标记名数据从 InTouch 标记名字典写入其它 Windows 应用程序时，上述定标操作相反。

例如操作员可能在数据输入窗口中输入设定点值 0-300 GPM，而系统则将 6400-32000 的值传递给 PLC 寄存器。

这对作为 DDE 客户机的 InTouch 程序总是有效的（包括向另一 Windows 程序发送或取得数据）。然而，当 InTouch 作为数据源时（另一 Windows 程序从 InTouch 请求数据），它将返回工程单位值给发出请求的程序。

例如，如果在 Excel 电子表格中的某个单元格包含远程引用公式：

=view|tagname!speed

则当前标记名 speed 的工程单位值将显示在单元格中。

仪器故障监视

从 7.0 版本开始，InTouch 支持三个标记名**点域**（**.RawValue**、**.MinRaw** 和 **.MaxRaw**），您可以在 InTouch QuickScript 中使用这些点域来监视仪器的数值，以确定是否超出范围、需要校准或出现故障。

内部系统 $Tagnames

InTouch 为您提供了许多预定义的内部系统标记名，您可以使用它们来执行各种操作。例如，如果您要显示当前的时间，就可以将系统标记名 **$Timestring** 链接到一个数值显示字符串。所有内部标记名以美元符号 ($) 开头。内部系统标记名可通过标记浏览器访问。

有关详细信息，请参阅“标记浏览器”。

下面的章节简要介绍内部系统标记名：

系统标记名	描述
$AccessLevel	只读整型安全标记名，用于表达式或脚本中，控制操作员执行特定功能的能力。
$AlarmLogging	在 InTouch 中不再受支持。
$AlarmPrinterError	在 InTouch 中不再受支持。
$AlarmPrinterNoPaper	在 InTouch 中不再受支持。
$AlarmPrinterOffline	在 InTouch 中不再受支持。
$AlarmPrinterOverflow	在 InTouch 中不再受支持。
$ApplicationChanged	只读离散型标记名，反映分布式系统中远程应用程序是否改变。每次在服务器节点的 WindowViewer“**特别**”菜单上选定“**通知客户**”命令时，此数值都会递增。
$ApplicationVersion	只读实型标记名，反映应用程序的当前版本号。每当标记名或 QuickScript 改变、添加或删除时，此数值都会改变。
$ChangePassword	只写离散型安全标记名，允许操作人员将 **$ChangePassword** 标记置为 1，从而向操作员显示一个“改变口令”常规对话框。
$ConfigureUsers	只写离散型安全标记名，可用在离散型按钮上，以允许操作员将 **$ConfigureUsers** 标记名值置为 1，从而显示一个常规“**设置用户**”对话框，让操作员可以编辑安全用户名列表。

系统标记名	描述
$Date	只读整型标记名，显示从 1/1/70 至今的天数。
$DateString	只读内存消息型标记名，以 WIN.INI 文件中设置的相同格式显示日期，例如 4/18/1992（此日期格式通过 Windows 控制面板设置）。
$DateTime	只读实型标记名，显示从 1/1/70 至今的天数（用小数来表示具体的时间）。
$Day	只读整型标记名，显示当前的日期（取值从 1 到 31）。
$HistoricalLogging	读/写离散型标记名，监视或控制历史记录的启动和停止。对于整个应用程序而言，这是一个全局命令。
$Hour	只读整型标记名，显示当前的小时值（取值从 0 到 23）。
$InactivityTimeout	只读离散型安全标记名，当设置的操作人员自动注销时间已到时，其值为 1。
$InactivityWarning	只读离散型安全标记名，当设置的警告操作人员注销即将发生的时间已到时，其值为 1。
$LogicRunning	读/写离散型标记名，用于监视或控制脚本的运行。 当前正在执行的异步用户定义函数脚本不能停止。但是，您可以防止执行新的脚本。
$Minute	只读整型标记名，显示当前的分钟（取值从 0 到 59）。
$Month	只读整型标记名，显示当前的月份（取值从 1 到 12）。
$Msec	只读整型标记，显示当前的毫秒数（取值从 0 到 999）。
$NewAlarm	可读写离散型标记名，每当发生新的报警时，其值为 1。
$ObjHor	只读整型标记名，用于显示所选对象的中心水平像素位置值。
$ObjVer	只读整型标记名，用于显示所选对象的中心垂直像素位置值。

系统标记名	描述
$Operator	只读安全消息型标记名，用于表达式中或脚本中，控制操作人员执行特定功能的能力。
$OperatorEntered	可读写安全消息型标记名，用于设置操作员的“用户名”。
$PasswordEntered	只读安全消息型标记名，用于设置操作员“口令”。
$Second	只读整型标记名，显示当前的秒数（取值从 0 到 59）。
$StartDdeConversations	读/写离散型标记名，用于当“**特别**”菜单被禁用时，启动未初始化的会话。
$System	作为报警根组的只读报警组型标记名。如果标记名未指定给特定报警组名，缺省条件下它将自动指定给此根报警组。所有定义的报警组都是 **$System** 的后代。
$Time	只读整型标记名，用毫秒显示自午夜以后的时间。
$TimeString	只读内存消息型标记名，按 Windows 注册表中设置的格式显示时间，如 12:01:59 PM（此格式可以通过 Windows 的控制面板进行设置）。
$Year	只读整型标记名，用 4 位数字显示年份，如 2008。

有关系统标记名的详细信息，请参阅*《InTouch 参考指南》*。

标记名点域

大多数关于 InTouch 的讨论都涉及对象的概念。对象的概念很广泛也很复杂。在这里，我们将对象的定义限制为屏幕上的一个图形对象的信息集合，或者标记名字典中的一个标记名的信息集合。

例如，如果在屏幕上画出一个矩形，这个矩形会有一些“属性”如线宽、线颜色、填充色、屏幕上的像素位置、指定链接等等。标记与此非常相似。例如创建一个名为 "Analog_Tagname" 的模拟型报警标记名，则它具有与其关联的一些属性，如标记名的名称，标记名的 **HiHi** 报警设置点等等。其中一些属性可以在 InTouch 中通过脚本、表达式和用户输入进行访问，这些属性被称为“**点域**”。访问与标记名关联的点域需要使用语法：**Tagname dotfields**。

例如，要允许在运行时改变标记名 "Analog_Tagname" 的 **HiHi** 报警限，您可以创建一个“**模拟 - 用户输入**”触动链接并将其应用到一个按钮，该按钮用表达式 **Analog_Tagname.HiHiLimit** 进行定义。在运行时，操作员可以简单地单击按钮并且为用于 "Analog_Tagname" 的 **HiHi** 报警限输入一个新值。

您可以使用**点域**来允许输入和输出与标记名关联的数据，您还可以使用历史**点域**来允许用户动态地修改所显示的历史趋势。例如，您可以允许用户控制滚动、锁定或重新定位趋势上的指示器，或重新把笔赋给新标记名。

有关**点域**的详细信息，请参阅联机*《InTouch 参考指南》*。

➢ **要访问标记名点域：**

在任意 InTouch QuickScript 或动画链接标记名或表达式输入框中，输入标记名加一个句点 (**tagname.**)，然后双击句点 (**.**) 右侧。“选择标记”浏览器将出现，显示为当前标记源定义的标记名。



➢ **要选择点域：**

1. 单击“**点域**”箭头来打开**点域**列表，您可以将此列表与当前选定的标记名类型相关联。

   缺省条件，所有类型的标记名最初均显示 **<无>**。

注意：以下情况下不能使用**点域**：从标记名字典访问标记浏览器；在运行时从“历史趋势设置”对话框中为历史趋势笔选择标记名（配置历史趋势时必须选择“允许运行时改变”选项）。

2. 在您要加入选定标记名的列表中，单击“**点域**”。

注意：不是每个标记名类型都有相同的**点域**。例如，**离散**类型的标记名有 **.OnMessage**，而模拟类型则没有。如果您选择**离散**类型的标记名并为其指定 **.OnMessage**，然后选择另一个**离散**类型的标记名，则显示的**点域**列表将不会改变。但是，如果您在控制视图列表中选择另一种类型的标记名，例如模拟类型，显示的**点域**将恢复为 **<无>**。

有关标记浏览器的详细信息，请参阅“标记浏览器”。

以下章节简要介绍标记名**点域**：

点域	描述
.Ack	读/写用于控制/监视标记名和报警组的报警确认状态的离散型标记名点域。 **.Ack** 有一个逆向标记名点域叫 **.Unack**。当出现未确认报警时，**.Unack** 将置为 1。随后 **.Unack** 可用于动画链接或条件脚本对任何未确认的报警触发电铃。
.AckDev	监视/控制标记名上活动偏差类型报警的报警确认状态。
.AckDsc	监视/控制报警或离散型标记的报警确认状态。
.AckROC	监视/控制标记名上活动变化率类型报警的报警确认状态。
.AckValue	监视/控制标记名上活动值类型报警的报警确认状态。
.Alarm	只读离散型点域，当指定的标记名、报警组名存在报警条件时，此值将置为 1。
.AlarmAccess	返回与所选报警关联的标记名的访问名。必须通过单击摘要分布式报警显示来选定报警。
.AlarmAckModel	监视如下与标记名关联的**确认模型**： 0=条件（缺省） 1=面向事件 2=扩展的摘要 应用于含报警的离散或模拟型标记名；只读属性，但可在 WindowMaker 中配置。
.AlarmClass	返回当前报警的类。
.AlarmComment	读/写用于描述报警内容而非标记名内容的文本字符串。在新的应用程序中缺省条件下为空白。 但是，当 InTouch 7.1 应用程序转换为 7.11 应用程序时，为保证反向兼容性，标记名注释将复制到“**报警注释**”中。
.AlarmDate	返回当前报警的日期。
.AlarmDev	发出存在偏差报警的信号。

点域	描述
.AlarmDevCount	记录给定标记名或报警组上活动偏差报警的总数。
.AlarmDevDeadband	读/写模拟（只对整型或实型标记名有效）标记名点域，监视/控制主/副偏差报警的偏差百分比死区。
.AlarmDevUnAckCount	记录给定标记名或报警组上活动的偏差报警总数。
.AlarmDisabled	禁用/启用事件和报警。应用于含报警的离散或模拟型标记名，或应用于报警组。 **注意：**与 **.AlarmEnabled** 相同，但极性相反。
.AlarmDsc	指示离散报警条件当前处于活动状态。
.AlarmDscCount	记录给定标记名或报警组上活动离散报警的总数。
.AlarmDscDisabled	指明标记名是否可产生离散报警。 **注意：**对于离散型标记名，此点域与 **.AlarmDisabled** 相同。
.AlarmDscEnabled	指明标记名是否可产生离散报警。 **注意：**对于离散型标记名，此点域与 .AlarmEnabled 相同。
.AlarmDscInhibitor	将指定的约束器标记名返回给此标记名的离散报警（如果有）。
.AlarmDscUnAckCount	记录给定标记名或报警组上活动的未确认离散报警总数。
.AlarmEnabled	读/写离散型标记名点域，停用/启用标记名和报警组的事件和报警。
.AlarmGroup	包含用于填入分布式报警显示对象的当前查询。
.AlarmGroupSel	返回所选报警的报警组。
.AlarmHiDisabled	禁用/启用含报警的模拟标记名的 **Hi** 报警限。
.AlarmHiEnabled	禁用/启用含报警的模拟标记名的 **Hi** 报警限。 **注意：**与对应的禁用点域相同，但极性相反。
.AlarmHiInhibitor	返回 **Hi** 报警限的**报警约束标记**引用。应用于含报警的离散型标记名。 只读属性，但可在 WindowMaker 中配置。

点域	描述
.AlarmHiHiDisabled	禁用/启用含报警的模拟标记名的 **HiHi** 报警限。
.AlarmHiHiEnabled	禁用/启用含报警的模拟标记名的 **HiHi** 报警限。 **注意：**与对应的禁用点域相同，但极性相反。
.AlarmHiHiInhibitor	返回 **HiHi** 报警限的**报警约束标记**引用。应用于含报警的离散型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmLimit	返回当前报警的极限。
.AlarmLoDisabled	禁用/启用含报警的模拟标记名的 **Lo** 报警限。
.AlarmLoEnabled	禁用/启用含报警的模拟标记名的 **Lo** 报警限。 **注意：**与对应的禁用点域相同，但极性相反。
.AlarmLoInhibitor	返回 **Lo** 报警限的**报警约束标记**引用。应用于含报警的离散型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmLoLoDisabled	禁用/启用含报警的模拟标记名的 **LoLo** 报警限。
.AlarmLoLoEnabled	禁用/启用含报警的模拟标记名的 **LoLo** 报警限。 **注意：**与对应的禁用点域相同，但极性相反。
.AlarmLoLoInhibitor	返回 **LoLo** 报警限的**报警约束标记**引用。应用于含报警的离散型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmMajDevDisabled	禁用/启用含报警的模拟标记名的**主偏差**报警限。
.AlarmMajDevEnabled	禁用/启用含报警的模拟标记名的**主偏差**报警限。 **注意：**与对应的禁用点域相同，但极性相反。
.AlarmMajDevInhibitor	返回**主偏差**报警限的**报警约束标记**引用。应用于含报警的离散型标记名。 只读属性，但可在 WindowMaker 中配置。

点域	描述
.AlarmMinDevDisabled	禁用/启用含报警的模拟标记名的**副偏差**报警限。
.AlarmMinDevEnabled	禁用/启用含报警的模拟标记名的**副偏差**报警限。 **注意：**与对应的禁用点域相同，但极性相反。
.AlarmMinDevInhibitor	返回**副偏差**报警限的**报警约束标记**引用。应用于含报警的离散型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmName	返回当前报警的名称。
.AlarmOprName	返回与报警关联的操作员姓名。
.AlarmOprNode	返回与报警关联的操作员节点。
.AlarmPri	返回报警的优先值 (1-999)。
.AlarmProv	返回报警的供应器。
.AlarmROC	发出存在变化率类型报警的信号。
.AlarmROCCount	记录给定标记名或报警组上活动变化率报警的总数。
.AlarmROCDisabled	禁用/启用含报警的模拟标记名的**变化率**报警限。
.AlarmROCEnabled	禁用/启用含报警的模拟标记名的**变化率**报警限。 **注意：**与对应的禁用点域相同，但极性相反。
.AlarmROCInhibitor	返回**变化率**报警限的约束器标记名引用。应用于含报警的离散型标记名。 只读属性，但可在 WindowMaker 中配置。
.Alarm ROCUnAckCount	记录给定标记名或报警组上的未确认变化率报警的总数。
.AlarmState	返回报警状态。
.AlarmTime	返回报警时间。
.AlarmTotalCount	记录给定标记名或报警组上活动报警的总数。
.AlarmType	返回报警类型。
.AlarmUnAckCount	记录给定标记名或报警组上活动的未确认报警总数。

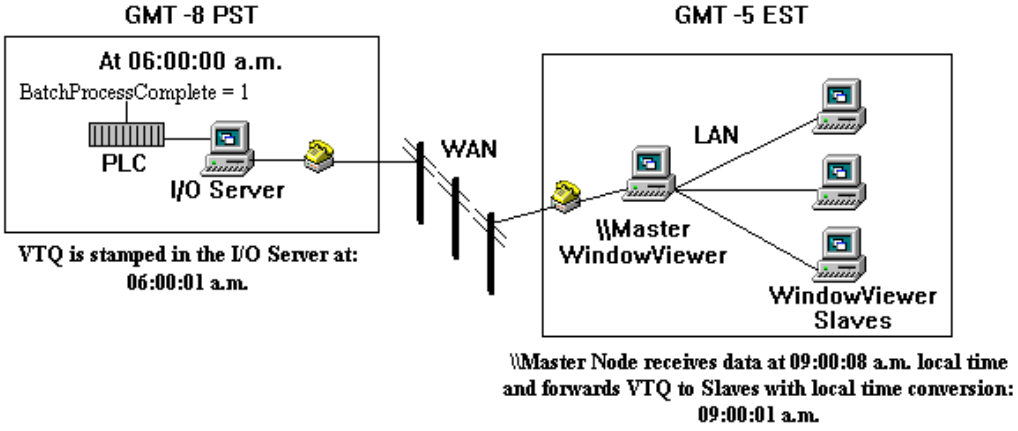
点域	描述
.AlarmUserDefNum1	读/写实型（浮点）、缺省 0 和未设定的值。应用于含报警的离散型标记名、含报警的模拟标记名或报警组。
	注意：此点域的值会附加在报警上，但只限于已设置值的情况下（例如通过脚本或插入来设置）。
.AlarmUserDefNum2	读/写实型（浮点）、缺省 0 和未设定的值。应用于含报警的离散型标记名、含报警的模拟标记名或报警组。
	注意：此点域的值会附加在报警上，但只限于已设置值的情况下（例如通过脚本或插入来设置）。
.AlarmUserDefStr	读/写文本字符串、缺省 "" 和未设置的值。应用于含报警的离散型标记名、含报警的模拟标记名或报警组。
	注意：此点域的值会附加在报警上，但只限于已设置值的情况下（例如通过脚本或插入来设置）。
注意：.AlarmUserDefNum1、**.AlarmUserDefNum2** 和 **AlarmUserDefStr** 点域允许您在报告时，指定一个或多个要追加到报警记录上的值。这些值由 Alarm DB Logger 写入数据库中。您可以在报警上附加三个项目：两个数字和一个字符串。缺省条件下，它们是空的（零和 ""）。 要简化用户值的设置，您可以在报警组及特定的标记名上对其进行设置。例如，InBatch 可以在 **$System** 报警组上设置 **.AlarmUserDefNum1** 中的批号，从而让所有报警都附上批号。 如果在某个报警组上设置 **.AlarmUserDefNum1**，它将应用于该组及其子组中的所有报警。您也可以在标记名上专门设置 **.AlarmUserDefNum1** 的值，在这种情况下，它将只应用于该标记名并改写标记名报警组中 **.AlarmUserDefNum1** 的任何设置。	
.AlarmValDeadband	可读/写模拟型（只对整型或实型标记名有效）点域，监视和/或控制报警死区的值。该点域对报警组和常规标记名有效。
.AlarmValue	返回报警值。
.AlarmValueCount	记录给定标记名或报警组上活动值报警的总数。
.AlarmValueUnAckCount	记录给定标记名或报警组上活动的未确认值报警总数。
.ChartLength	可读/写整型标记名点域，用于控制显示在历史趋势图中的时间长度。**.ChartLength** 以秒为单位显示图表长度。

点域	描述
.ChartStart	可读/写整型标记名点域，用于控制开始时间和/或滚动相应的历史趋势图表。**.ChartStart** 显示自 70 年 1 月 1 日午夜 12 点以来已过的时间（以秒为单位）。
.Comment	只读消息型标记名点域，用于显示在标记名字典中的标记名的注释域。
.DevTarget	可读/写模拟型（只对整型或实型标记名有效）点域，监视和/或控制副偏差和主偏差报警目标。
.DisplayMode	可读/写模拟标记名点域，用于决定在趋势中显示值的方式。
.EngUnits	读写模拟标记名点域，用于访问在标记名字典中指定的模拟标记名的工程单位。 **注意：**写入的数值不被保留。
.Freeze	读/写分布式报警显示对象的冻结状态。
.HiLimit, .HiHiLimit, .LoLimit, .LoLoLimit	读/写模拟标记名点域，监视/控制值报警检查极限。这些点域只对整型和实型标记名有效。
.HiStatus, .HiHiStatus, .LoStatus, .LoLoStatus	只读离散型标记名点域，确定指定类型的报警是否存在。
.ListChanged	指明是否存在任何新报警或分布式报警对象的更新。
.MajorDevPct	读/写实型标记名点域，监视或控制报警检查的主偏差百分比。
.MajorDevStatus	只读离散型标记名点域，决定指定标记名的主偏差报警是否存在。
.MaxEU, .MinEU	只读整性标记名点域，显示标记名的最大值和最小值。
.MaxRange, .MinRange	读/写实型标记名点域，用于表示当前绘制趋势的每个标记名所显示的标记名工程单位范围的百分比。**.MaxRange** 和 **.MinRange** 范围限于从 0 到 100，并且 **.MinRange** 恒小于 **.MaxRange**。如果某个域被赋于小于 **0** 或大于 **100** 的值，这些值将定为 **0** 或 **100**。如果 **.MinRange** 大小或等于 **.MaxRange**，趋势将不显示任何数据。
.MaxRaw, .MinRaw	WindowViewer 作为客户程序从从 I/O 服务器收到的实际原始数据的高/低嵌位设置。标记名点域 **.MaxRaw/.MinRaw** 的值来自标记名数据库中指定 I/O 标记名的最大/最小原始数据值。任何超过或低于该设置的原始值将被嵌位为该值。

点域	描述
.MinorDevPct	读/写实型标记名点域，用于监视/控制报警检查的副偏差百分比。
.MinorDevStatus	只读离散型标记名点域，用于确定指定标记名的副偏差报警是否存在。
.Name	读/写消息型标记名点域，用于显示标记名的实际名称。例如，**标记 ID** 标记名的名称。
.Normal	只读离散型点域，当指定名称没有报警时值为 1。此点域对报警组和常规标记名有效。
.OffMsg, .OnMsg	读/写消息型标记名点域，用于显示标记名字典中指定的离散型标记名的开消息和关消息。 **注意：**写入的数值不被保留。
.Pen1 - .Pen8	读/写**标记 ID** 类型的标记名点域，用于控制每支笔当前正在绘制历史趋势的标记名。 **注意：**将标记名动态赋值给笔的一个更简单方法是使用历史函数 **HTSetPenName** 和 **HTGetPenName**。 有关详细信息，请阅见第 8 章“实时趋势和历史趋”。 有关标记 ID 标记名类型的详细信息，请参阅“标记名类型”。
.Quality	只读整型标记名，允许用户访问作为 I/O 服务器的 I/O 标记名质量。
.QualityLimit	只读整型标记名点域，用于显示当 I/O 连接有效时，由数据源提供的 I/O 值的质量限制。
.QualityLimitString	只读消息型标记名点域，用于显示当 I/O 连接有效时，由 I/O 服务器提供的 I/O 值的质量限制字符串。
.QualityStatus	只读整型标记名点域，用于显示当 I/O 连接有效时，由 I/O 服务器提供的 I/O 值的质量状态。
.QualityStatusString	只读消息型标记名点域，用于显示当 I/O 连接有效时，由 I/O 服务器提供的 I/O 值的质量状态字符串。
.QualitySubstatus	只读整型标记名点域，用于显示当 I/O 连接有效时，由 I/O 服务器提供的 I/O 值的质量状态。

点域	描述
.QualitySubstatusString	只读消息型标记名点域，用于显示当 I/O 连接有效时，由 I/O 服务器提供的 I/O 值的质量状态字符串。 **注意：**如果 I/O 连接有效，质量点域自动复位为初始值零。**.ReferenceComplete** 标志也将同时设置为零。
.RawValue	任何类型（实型/离散型）的标记名点域，用于显示在 InTouch 应用定标前，实际的离散或模拟 I/O 值。
.Reference	读/写标记名点域，与 I/O 类型标记名一起使用，以动态地改变标记名源的地址。
.ReferenceComplete	离散型标记名点域，确认所请求的项目同样反映在 **.Value** 域中。 有关 .Reference 点域的详细信息，请参阅“动态引用编址”。
.ROCPct	读/写标记名点域，用于监视和/或控制报警检查的变化率。
.ROCStatus	只读离散型标记名点域，用于确定指定的标记名是否存在变化率报警。
.ScooterLockLeft	读/写离散型标记名点域。当此点域值为“真”时，则右指示器不能移向左指示器位置的左边。（0=假、1=真）。
.ScooterLockRight	可读/写离散标记名点域，当该字段值为“真”时，左控制器不能移动到右控制器的右边。（0=假、1=真）。
.ScooterPosLeft	读/写实型标记名点域，代表左指示器位置（范围从 0.0 到 1.0）。
.ScooterPosRight	读/写实型标记名点域，代表右指示器位置（范围从 0.0 到 1.0）。
.SuppressRetain	读/写分布式报警显示对象的抑制保持状态。
.TagID	只读**标记 ID** 标记名点域，与历史趋势 **.Pen1-.Pen8 标记 ID** 标记名一起使用，监视和/或控制某支笔当前绘制趋势的标记名（参见前面的 "Pen" 点域描述）。

下面显示 **.Time** 域如何接收其数据：



Therefore, we know from a global time standpoint that,
BatchProcessComplete was set to 1, seven seconds earlier.

点域	描述
.TimeDate	实整型标记名点域，用于显示当 I/O 连接有效时，自 I/O 服务器提供 I/O 值以来已过的整数天数。
.TimeDateString	只读消息型标记名点域，用 Windows 注册表中设置的相同格式来显示日期。例如，当 I/O 连接有效时，I/O 服务器提供 I/O 值的日期 10/31/1997。
.TimeDateTime	只读实型标记名点域，用于显示当 I/O 连接有效时，自 I/O 服务器提供 I/O 值以来已过的天数（用小数表示具体时间）。
.TimeDay	只读整型标记名点域，用于显示当 I/O 连接有效时，I/O 服务器提供 I/O 值的日期。（此值可能是 1-31）。
.TimeHour	只读整型标记名点域，用于显示当 I/O 连接有效时，I/O 服务器提供 I/O 值的小时值（取值为 0-23）。
.TimeMinute	只读整型标记名点域，用于显示当 I/O 连接有效时，I/O 服务器提供 I/O 值的分钟值（取值为 0-59）。
.TimeMonth	只读整型标记名点域，用于显示当 I/O 连接有效时，I/O 服务器提供 I/O 值的月份值（取值为 1-12）。
.TimeMsec	只读整型标记名点域，用于显示当 I/O 连接有效时，I/O 服务器提供 I/O 值的时间（以毫秒为单位）。
.TimeSecond	只读整型标记名点域，用于显示当 I/O 连接有效时，I/O 服务器提供 I/O 值的时间（以秒为单位）（取值为 0-59）。
.TimeTime	只读实型标记名点型，用于显示当 I/O 连接有效时，I/O 服务器提供 I/O 值的时间（自午夜起，以毫秒为单位）。
.TimeTimeString	只读消息型标记名点域，用于显示当 I/O 连接有效时，I/O 服务器提供 I/O 值的日期和时间。

点域	描述
.TimeYear	只读整型标记名点域，用 Windows 注册表中设置的相同格式来显示时间例如，当 I/O 连接有效时，I/O 服务器提供 I/O 值的时间 10/31/1997（此时间格式可以通过 Windows 的控制面板进行设置）。
.Unack	读/写离散型标记名点域，用于控制本地报警的报警未确认状态。
.UpdateCount	只读整型标记名点域，当此趋势的检索完成时，此标记名点域递增。
.UpdateInProgress	只读离散型标记名点域，显示历史数据检索状态（0=未检索，1=正在检索）。
.UpdateTrend	只写离散型标记名点域，当设置为 1 时，将用所有当前值来更新历史趋势图表。历史趋势不会自动自我更新。要更新图表并显示指定标记名的当前值，必须更改图表开始值或图表长度等。在按钮 QuickScript 中使用此点域，操作员将可以在运行时随时更新图表。
.Value	读或读/写模拟标记名点域，显示指定标记名的值。

下列报警**点域**与报警对象关联，而与标记名无关。

.AlarmClass	**.AlarmAccess**	**.AlarmDate**	**.AlarmGroupSel**
.AlarmLimit	**.AlarmName**	**.AlarmOprName**	**.AlarmOprNode**
.AlarmPri	**.AlarmProv**	**.AlarmState**	**.AlarmTime**
.AlarmType	**.AlarmValue**	**.Freeze**	**.ListChanged**
.NumAlarm	**.PendingUpdates**	**.SuppressRetain**	

这些报警**点域**通过下面的函数进行访问：

GetPropertyM(ControlName.Property, MsgTag).

当在运行环境下执行时，**MsgTag** 会取得属性以在选定的行中显示标记名。如果选定多个行，**MsgTag** 的属性为其中第一行的标记名。

有关报警**点域**的详细信息，请参阅联机*《InTouch 参考指南》*。

模拟型标记名的地址位域

整型标记名中的单个位可以用位域来编址。这些位被视为离散量 (0/1)，如果写入，模拟标记名将立即更新。您可以在任何使用离散型标记名的地方使用位域。例如，I/O、脚本、表达式等中。

.00 第零位。

.01 第一位。

.02 以此类推

.

.

.

.31 32 位整型中的最高位

下面是如何在表达式中使用位域的实例：

```
Temperature.08 == 1;
```

下面是如何在 QuickScript 中使用位域的实例：

```
IF Temperature.29 THEN
   Temperature.29 =0;
ENDIF;
```

标记名字典实用程序

InTouch 提供了两个标记名字典实用程序：DBDump 和 DBLoad。DBDump 用于将 InTouch 应用程序标记名字典导出为文本文件，以便在另一程序（例如 Microsoft Excel）中查看或编辑。DBLoad 允许将适当格式化的标记名字典文件（在另一程序如 Excel 中创建，或从另一 InTouch 应用程序导入的 DBDump 文件）加载到一个现有的 InTouch 应用程序中。这两个程序允许分别复制、修改或开发数据库（标记名字典）的各个部分，然后合并到一个应用程序中。

DBLoad 实用程序也可作为 InTouch TemplateMaker 的替代方式，用于创建 SuperTag 实例。

有关详细信息，请参阅“创建 SuperTag 实例”。

注意：DBDump 和 DBLoad 实用程序都是从 InTouch 应用程序管理器 (INTOUCH.EXE) 启动的。此外，您必须对使用较早版本的 InTouch 创建的应用程序进行转换，然后才能提取其标记名字典。

有关创建数据库文件的详细信息，请参阅“创建数据库输入文件”。

DBDump 实用程序

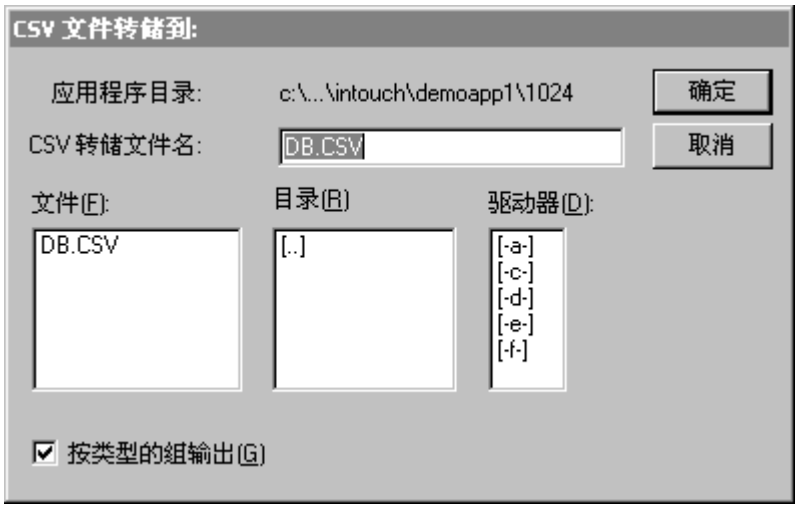


➢ **要提取现有 InTouch 应用程序的标记名字典：**

1. 如果 WindowMaker 和 WindowViewer 正在运行，先将其关闭。
2. 启动 InTouch。**“InTouch 应用程序管理器”**对话框出现：



3. 在“**文件**”菜单上，单击“**DBDump**”，或者单击 DBDump 工具。“**CSV 文件转储到：**”对话框出现：



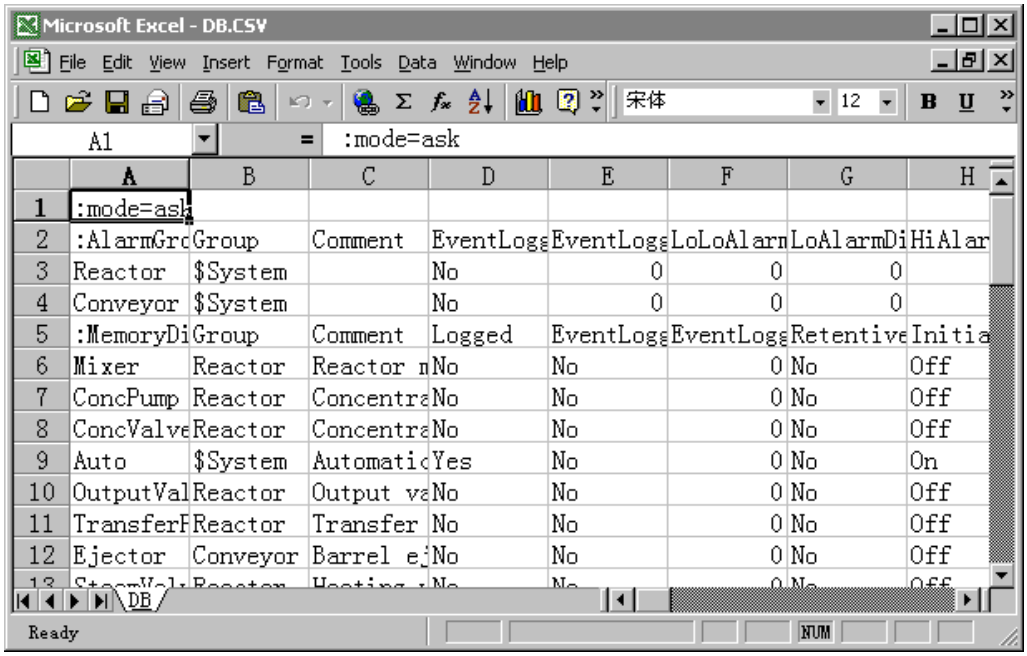
4. 在“CSV 文件转储文件名”框中，输入以.CSV（逗号分隔变量）扩展名结尾的文件名（如果同名文件已经存在，则会出现一个消息框）。

5. 选择“**按类型的组输出**”以按标记名类型而不是按标记名的字母顺序（缺省），对所提取的标记名分组。

注意：要按组提取数据库条目，系统必须针对每个标记名类型读取整个文件。所以，需要更长时间来提取数据。但是，当在支持 .CSV 的应用程序（例如 Microsoft Excel）中打开输出文件时，分组可以使读取或编辑更加容易。转储文件中包括每个标记名类型的占位符，而不管该类型的标记名是否存在。

6. 单击“**确定**”。所选应用程序目录的数据库信息将下载到指定的文件名中。

如果您在 Microsoft Excel 中打开 .CSV 文件，它会将逗号视为分隔符，并将数据记录自动分隔成列。例如：



	A	B	C	D	E	F	G	H
1	:mode=ask							
2	:AlarmGrc	Group	Comment	EventLogg	EventLogg	LoLoAlarn	LoAlarmDi	HiAlar
3	Reactor	$System		No	0	0	0	
4	Conveyor	$System		No	0	0	0	
5	:MemoryDi	Group	Comment	Logged	EventLogg	EventLogg	Retentive	Initia
6	Mixer	Reactor	Reactor n	No	No	0	No	Off
7	ConcPump	Reactor	Concentra	No	No	0	No	Off
8	ConcValve	Reactor	Concentra	No	No	0	No	Off
9	Auto	$System	Automatic	Yes	No	0	No	On
10	OutputVal	Reactor	Output va	No	No	0	No	Off
11	TransferF	Reactor	Transfer	No	No	0	No	Off
12	Ejector	Conveyor	Barrel ej	No	No	0	No	Off

如果您在记事本中打开 .CSV 文件，每条数据记录将由逗号隔开。例如：



DBLoad 实用程序

➢ **要加载/合并数据库输入文件到现有的 InTouch 中：**

1. 如果 WindowMaker 和 WindowViewer 正在运行，先将其关闭。
2. 启动 InTouch。“**InTouch 应用程序管理器**”对话框出现：



3. 在“**文件**”菜上单，单击“**DBLoad**”，或者单击 DBLoad 工具。出现一个消息框，询问您是否已经对应用程序进行备份。单击“**是**”继续。“**要加载 CSV 文件自：**”对话框出现：



4. 在“**CSV 加载文件名**”框中，输入您要加载的 .CSV 文件的路径，或者用“**目录**”和“**驱动器**”列表框找到此文件（一旦正确选定文件，其名称将出现在框中）。
5. 单击“**确定**”。所选文件中包含的数据库信息将开始加载到所选应用程序的标记名字典中。

创建数据库输入文件

DBDump 和 DBLoad 数据库实用程序是用于在标记名字典中执行批处理类型操作的工具。数据库输入文件可以在任何支持逗号分隔变量文件格式 (**.CSV**) 的程序中创建（数据库输入文件必须以逗号分隔变量的格式创建）。例如，写字板、记事本和 Microsoft Excel。一旦创建了输入文件，就可以使用 DBLoad 程序将文件中包含的数据加载/合并到现有的 InTouch 应用程序数据库中了。

您可以通过创建一个新的 InTouch 应用程序来创建数据库输入文件“模板”，然后运行 DBDump 程序来转储其数据库，以创建一个正确格式的.CSV 文件。这使得输入修改内容比从头创建一个输入文件要来得容易。

有关详细信息，请参阅“创建数据库记录模板”。

数据库输入文件格式

数据库输入文件的第一行应该指定当通过 **DBLoad** 加载/合并文件到应用程序时，文件的操作模式“**:mode**”。

- 如果您不指定模式“**:mode=test”**，则缺省条件下将使用“**Ask”**。

有关有效模式关键字的详细信息，请参阅“数据库输入文件操作模式”。

所有数据记录均必须以标记名 **:type** 的有效关键字开始，后跟每条数据记录的有效关键字（以逗号分隔）：

```
:mode=test
:IOMsg,Group,Comment,Logged,Event Logged,Event Logging Priority, . . .
```

每个标记名类型和数据记录都有一个有效的**关键字**。

有关有效标记名类型和数据记录项的详细信息，请参阅“类型和关键字条目”。

随后输入实际的标记名，后跟每条数据记录的值（用冒号分隔）。

```
:mode=test
:IOMsg,Group,Comment,Logged,EventLogged,EventLoggingPriority, . . .
Ingredient_1,$System,"",No,No,999, . . .
```

在上面的例子中，**IOMsg** = Ingredient_1，**Group** = $System，**Logged** = No，**EventLogged** = No，**EventLoggingPriority** = 999。因为输入了 ""，数据记录 **Comment** 将为空白。

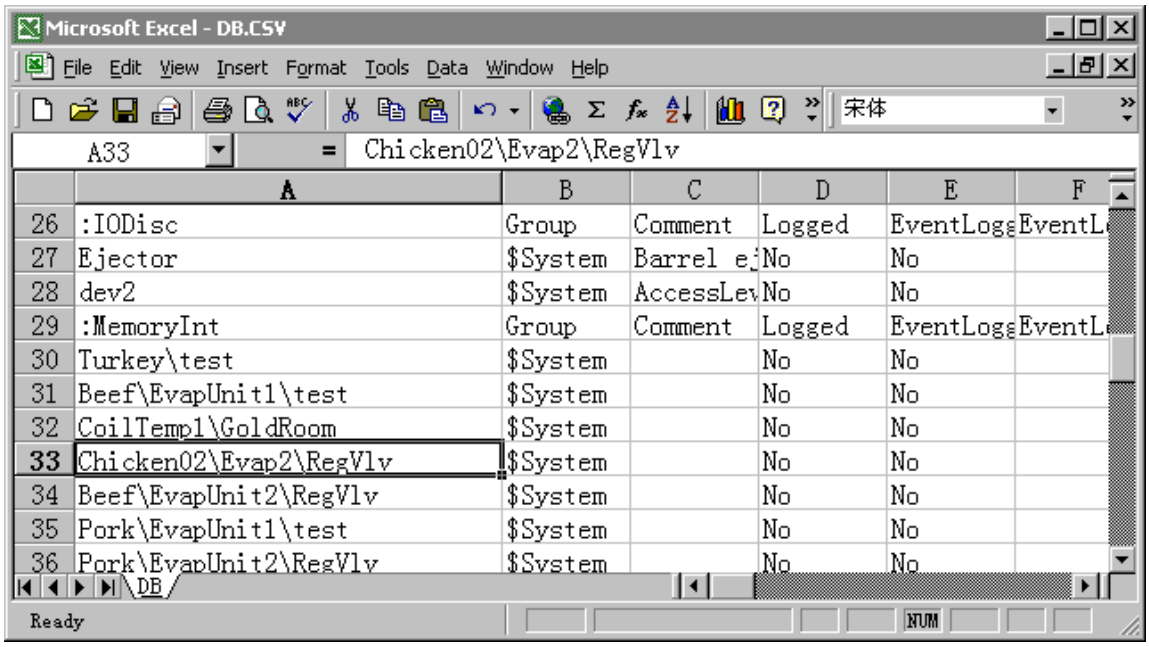
注意：模式和类型关键字之前必须有一个冒号（:）。如果要换行，可在行尾输入一个反斜杠（\）。如果要输入注释，可以 在它前面加上一个分号（;）。

创建 SuperTag 实例

除了 TemplateMaker、动画链接、InTouch QuickScript 和标记名字典外，InTouch 也支持通过 DBLoad 实用程序来创建 SuperTag。

注意：当您用 DBLoad 来创建 SuperTag 实例时，它们不会反映在 TemplateMaker 的 SuperTag 模板定义中。

当您通过 DBLoad 来创建 SuperTag 时，必须使用有效的 SuperTag 格式，而且此 SuperTag 实例的数据记录必须以标记名 **:type** 的有效关键字开始。例如：



下面的语法实例是有效的：

ParentInstance\ChildMember

ParentInstance\ChildMember\Submember

而下面的语法例子是无效的：

ParentInstance

ParentInstance\ChildMember

如果使用了无效的格式，将出现一个错误消息框，通知您语法是无效的。

当您加载包含 SuperTag 实例的 .csv 文件时，它们会自动加入到标记名字典中，并且可立即用于 InTouch QuickScripts 的动画链接中。

空字符串与无输入值

空字符串字段与无输入值字段不同。例如：

```
:Comment="HI"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,"",$System
```

此处：

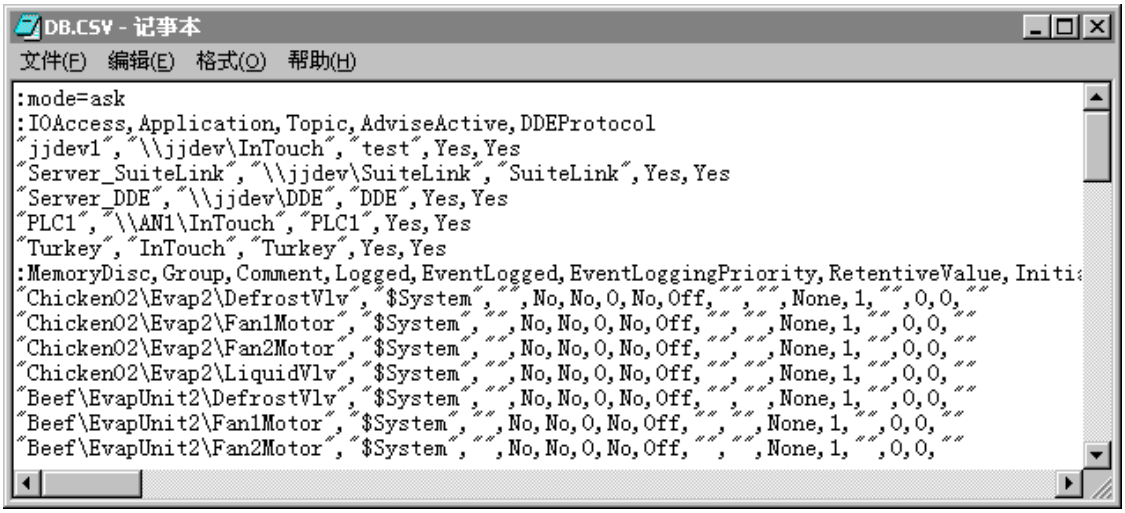
Tagname1 的**注释**字段为**"Hi"**，而 Tagname2 的注释字段为空白。Excel 会将上述内容读入 .CSV 文件，但当保存文件时，则按照下面的内容保存：

```
:Comment="HI"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,,$System
```

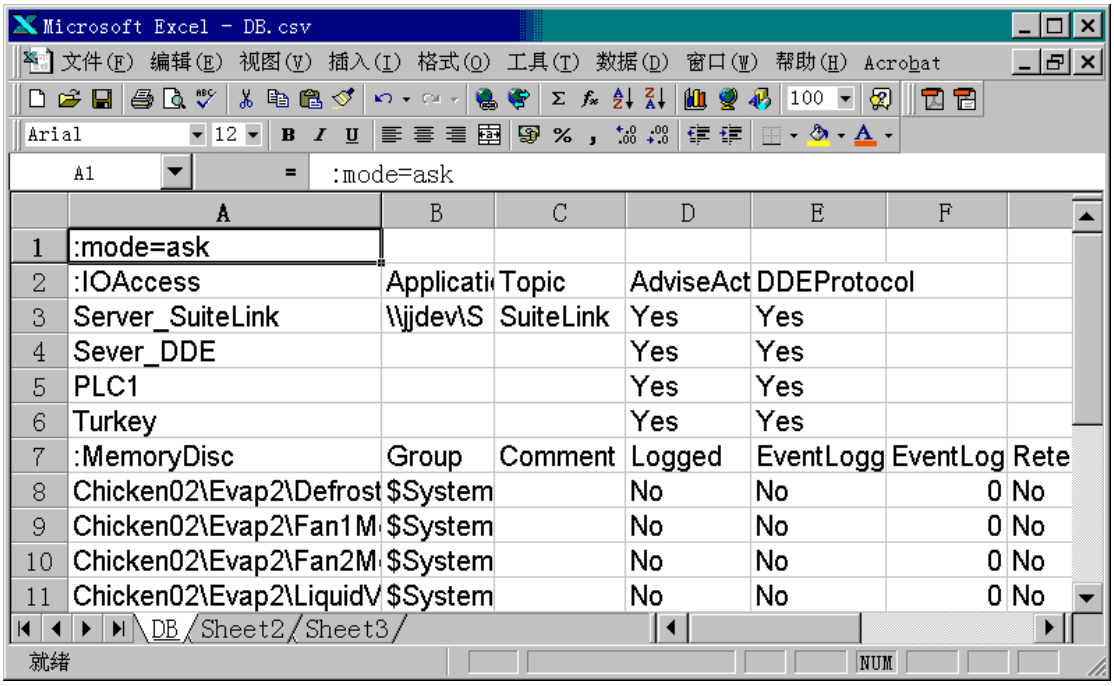
所以，为确保一个**空白**字符串，必须在要保留**空白**的单元格中输入一个空格（使用空格键）。这样，下面的关键字字段将受影响。

```
Comment  Eng Units
OffMsg   Initial Message
OnMsg    Application
ItemName Topic
```

下面说明用 Windows 记事本创建的一个输入文件：



当您用某个程序（例如 Excel）创建输入文件时，除了在不同的列输入每一项之外，其余格式要求相同。这使得读取数据更容易，出错的机会也更小。例如：



	A	B	C	D	E	F	
1	:mode=ask						
2	:IOAccess	Applicati	Topic	AdviseAct	DDEProtocol		
3	Server_SuiteLink	\\jjdev\S	SuiteLink	Yes	Yes		
4	Sever_DDE			Yes	Yes		
5	PLC1			Yes	Yes		
6	Turkey			Yes	Yes		
7	:MemoryDisc	Group	Comment	Logged	EventLogg	EventLog	Rete
8	Chicken02\Evap2\Defrost	$System		No	No	0	No
9	Chicken02\Evap2\Fan1M	$System		No	No	0	No
10	Chicken02\Evap2\Fan2M	$System		No	No	0	No
11	Chicken02\Evap2\LiquidV	$System		No	No	0	No

数据库输入文件操作模式

下面列出有效的操作模式**关键字**，以及在加载时遇到重复的标记名时每种模式发生的动作：

:MODE=REPLACE

:MODE=UPDATE

:MODE=ASK

:MODE=IGNORE

:MODE=TERMINATE

:MODE=TEST

:MODE=REPLACE

删除现有条目，代之以新条目。

:MODE=UPDATE

只用输入文件中明确定义的字段改写现有的定义。

如果字段位于记录中，并且是由您输入或通过 "**:KEYWORD=value**" 机制进行设置，则此字段将被视为已明确定义。如果字段没有在记录中指定并且已经通过“**:KEYWORD=**”命令重置关键字，则当前字段值将不会被更新。

下面的实例描述当更新模式下的输入文件通过 **DBLoad** 加载/合并到应用程序的数据库时将发生的情况：

```
:Mode=update
:Group=Group1
:IODisc,Group,DConversion
Tagname1,Group2,
```

Tagname1 所在的组仅更新为 Group2

```
Tagname2,,
```

Tagname2 所在的组更新为 Group1，而 DConversion 保持不变

```
Tagname3,,Reverse
```

Tagname3 所在的组更新为 Group1，而 DConversion“取反”

下面的语句行将组域“重置”为其缺省值

```
:Group=
```

数据域“Group”重置为其缺省值

```
Tagname4,,
; Tagname4 will be left alone
```

在 .csv 文件中允许使用注释，但必须用一个分号 (**;**) 注明。

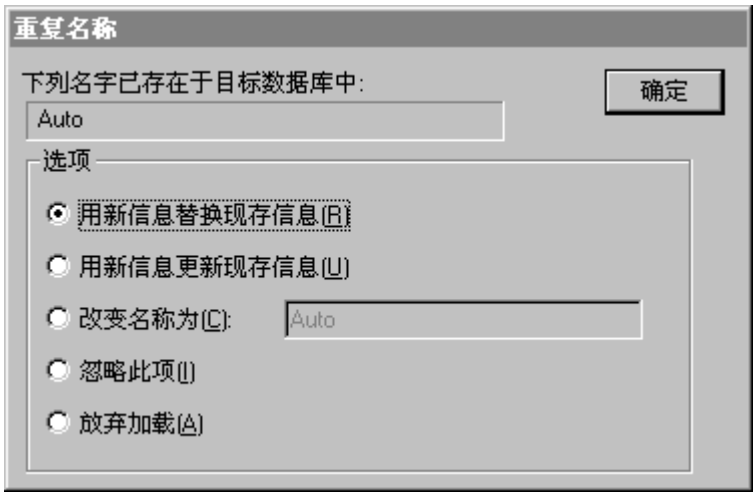
注意：如果要更改标记名类型并且标记名正被使用，则标记名类型必须兼容。例如，如果一个现有的历史趋势标记名正被应用程序使用，则它不能改为 I/O 整型。此外，如果某个标记名用于应用程序的输入链接中，则该标记名不能改为 ReadOnly=yes。由于这些限制，在运行 DBLoad 之前，您必须更新目标应用程序的标记名使用计数。

:MODE=ASK

将输入的或现有条目的标记名改为指定的标记名，然后在标记名字典中插入新定义。

注意：如果没有指定模式，则此模式为缺省输入模式。每当 DBLoad 遇到重复的标记名时，都会出现一个对话框，询问是否替换现有标记名。

当操作模式设为“**ASK**”并且输入文件中的标记名与目标应用程序标记名字典中的标记名重复时，“**重复名称**”对话框将会出现：



选择某个选项，然后单击“**确定**”。当加载完成时，将出现一个确认消息框。

选项	描述
改变名称为	用框中输入的名称替换指定标记名的名称。
忽略此项	忽略显示的标记名，处理过程继续。
用新信息替换现存信息	用新记录替换现有的标记名记录。
用新信息更新现存信息	只用输入文件中明确定义的字段来改写现有的标记名记录。
放弃加载	取消加载功能。

注意：如果因某个问题导致加载失败，将出现一个消息框（错误消息将写入 Wonderware Logger 程序中）。

有关 Wonderware Logger 的详细信息，请参阅您的联机*《FactorySuite 系统管理员指南》*。

:MODE=IGNORE

忽略新记录并继续处理。

:MODE=TERMINATE

终止处理，不更新目标文件。

:MODE=TEST

在这种模式下，DBLoad 如同在替换模式下运行，但是不会修改数据库。它会在 Wonderware Logger 程序中报告发现的错误，并继续加载。这对于在实际处理之前校验输入文件的语法非常有用。

有关 Wonderware Logger 程序的详细信息，请参阅*《FactorySuite 系统管理员指南》*。

注意：一旦输入文件设为测试模式，系统将忽略所有其它模式。您可以输入 **:mode=test** 作为文件的第一行，而不考虑文件中的其它模式改变。

有关操作模式的详细信息，请参阅“创建数据库输入文件”。

创建数据库记录模板

“**关键字**”可用于创建模板记录，以便为后续数据记录中的各个数据字段提供“全局”输入项（除 **TAGNAME** 和 **TYPE** 字段以外，每个可设定用于标记名的字段值都有一个关键字）。

注意：模板记录必须遵照先前所述的有关创建逗号分隔输入文件的格式要求。例如，文件必须以 **:Type** 关键字条目开始。

设置字段缺省值

关键字也可用于设置特定记录字段的缺省值。例如：

:KEYWORD=value

此语句可以为所有后续数据记录设置缺省的引用字段值。此特性可用于为许多记录中保持不变的字段值设置缺省值。如果某个字段定义了缺省值，则当值记录中没有数据时将使用缺省值。例如，“**:GROUP=Reator_Site**”的结果是将所有具有空白 **GROUP** 项的标记名指定给 **Reator_Site** 报警组，而如果为 **GROUP** 输入了标记名如 **$System**，它将仍指定给报警组 **$System**。

重置字段缺省值

如果不指定任何值，则单个关键字可重置为其原始缺省值。例如，“**:GROUP=**”。

重置所有字段缺省值

要重置所有关键字，可使用 **:RESET** 命令。此命令不带参数，只要照原样输入就可以了，它可以改变命令后面的所有项目。

注意：缺省值就是该标记名类型的原始 InTouch 值。例如：内存离散量使用 **Group=$System**、**EventLogging=Off**、**InitialValue=Off** 等作为其缺省值。

:Type 和关键字条目

下表列出了每个标记名 **:Type** 的有效关键字，以及该类型的每个字段值的关键字。

:Type 和关键字	**容许值**	**缺省**
:MemoryDisc		
ACKModel	整型 [0=条件，1=事件，或 2=扩展]	0
AlarmComment	任意文本字符串，读/写，[适用于所有标记类型]	空白 注：在 InTouch 7.1 或之前版本中，与标记名说明注释相同。
AlarmPri	1 到 999	1
AlarmState	无，开或关。	无
Comment	任意文本字符串	""
DscAlarmDisable	1, 0	0
DscAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
InitialDisc	1、0、开、关、真、假、是或否	0
Logged	开、关、是或否	否
OffMsg	任意文本字符串	无
OnMsg	任意文本字符串	无
RetentiveValue	1、0、开、关、真、假、是或否	否

:Type 和关键字	容许值	缺省
:IODisc		
ACKModel	整型 [0=条件，1=事件，或 2=扩展]	0
AlarmComment	任意文本字符串，读/写，[适用于所有标记类型]	空白 注：在 InTouch 7.1 或之前版本中，与标记名说明注释相同。
AlarmPri	1 到 999	1
AlarmState	无，开或关。	无
Comment	任意文本字符串	无
DConversion	直接或取反	直接
AccessName	有效访问名字	无
DscAlarmDisable	1, 0	0
DscAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
InitialDisc	1、0、开、关、真、假、是或否	0
ItemName	有效的项目名	无
ItemUseTagname	真、假、是或否	是
Logged	开、关、是或否	否
OffMsg	任意文本字符串	无
OnMsg	任意文本字符串	无
ReadOnly	是或否	否
RetentiveValue	1、0、开、关、真、假、是或否	否

:Type 和关键字	容许值	缺省
:MemoryInt & :Memory Real		
ACKModel	整型 [0=条件，1=事件，或 2=扩展]	0
AlarmComment	任意文本字符串，读/写，[适用于所有标记类型]	空白 注：在 InTouch 7.1 或之前版本中，与标记名说明注释相同。
AlarmDevDeadband	有效偏差百分比	0
AlarmValueDeadband	有效整型或实型值	0
Comment	任意文本字符串	无
Deadband	有效整型或实型值	0
DevTarget	有效整型或实型值	0
EngUnits	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
HiAlarmPri	1 到 999	1
HiAlarmState	是，否，开，或关	否
HiAlarmValue	有效整型或实型值	0
HiHiAlarmPri	1 到 999	1
HiHiAlarmState	是，否，开，或关	否
HiHiAlarmValue	有效整型或实型值	0
InitialValue	有效整型或实型值	0
LoAlarmPri	1 到 999	1
LoAlarmState	是，否，开，或关	否
LoAlarmValue	有效整型或实型值	0
LogDeadband	有效整型或实型值	0
Logged	开、关、是或否	否

:Type 和关键字	容许值	缺省
LoLoAlarmPri	1 到 999	1
LoLoAlarmState	是，否，开，或关	否
LoLoAlarmValue	有效整型或实型值	0
MajorDevAlarmPri	1 到 999	1
MajorDevAlarmState	是，否，开，或关	否
MajorDevAlarmValue	有效整型或实型值	0
MaxValue	有效整型或实型值	9999
MinorDevAlarmPri	1 到 999	1
MinorDevAlarmState	是，否，开，或关	否
MinorDevAlarmValue	有效整型或实型值	0
MinValue	有效整型或实型值	0
RetentiveAlarmParameters	开、关、是或否	否
RetentiveValue	开、关、是或否	否
ROCAlarmPri	1 到 999	1
ROCAlarmState	是，否，开，或关	否
ROCAlarmValue	任何有效的百分比	0
ROCTimeBase	秒，分，或小时	分

报警类型的报警约束标记名	容许值	缺省
HiAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
HiHiAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
LoAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
LoLoAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
MajDevAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
MinDevAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
ROCAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无

禁用标志	容许值	缺省
HiAlarmDisable	1, 0	0
HiHiAlarmDisable	1, 0	0
LoAlarmDisable	1, 0	0
LoLoAlarmDisable	1, 0	0
MajDevAlarmDisable	1, 0	0
MinDevAlarmDisable	1, 0	0
ROCAlarmDisable	1, 0	0

:Type 和关键字	容许值	缺省
:IOInt & :IOReal		
ACKModel	整型 [0=条件，1=事件，或 2=扩展]	0
AlarmComment	任意文本字符串，读/写，[适用于所有标记类型]	空白 注：在 InTouch 7.1 或之前版本中，与标记名说明注释相同。
AlarmDevDeadband	有效偏差百分比	0
AlarmValueDeadband	有效整型或实型值	0
Comment	任意文本字符串	无
Conversion	线性或平方根	线性
AccessName	有效访问名字	无
Deadband	有效整型或实型值	0
DevTarget	有效整型或实型值	0
EngUnits	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
HiAlarmPri	1 到 999	1
HiAlarmState	是，否，开，或关	否
HiAlarmValue	有效整型或实型值	0
HiHiAlarmPri	1 到 999	1
HiHiAlarmState	是，否，开，或关	否

HiHiAlarmValue	有效整型或实型值	0
InitialValue	有效整型或实型值	0
ItemName	有效的项目名	无
ItemUseTagname	真、假、是或否	否

:Type 和关键字	**容许值**	**缺省**
LoAlarmPri	1 到 999	1
LoAlarmState	是，否，开，或关	否
LoAlarmValue	有效整型或实型值	0
LogDeadband	有效整型或实型值	0
Logged	开、关、是或否	否
LoLoAlarmPri	1 到 999	1
LoLoAlarmState	是，否，开，或关	否
LoLoAlarmValue	是，否，开，或关	0
MajorDevAlarmPri	1 到 999	1
MajorDevAlarmState	是，否，开，或关	否
MajorDevAlarmValue	有效整型或实型值	0
MaxEU	有效整型或实型值	9999
MaxRaw	有效整型或实型值	9999
MinEU	有效整型或实型值	0
MinorDevAlarmPri	1 到 999	1
MinorDevAlarmState	是，否，开，或关	否
MinorDevAlarmValue	有效整型或实型值	0
MinRaw	有效整型或实型值	0
ReadOnly	是或否	否
RetentiveAlarmParameters	开、关、是或否	否
RetentiveValue	开、关、是或否	否
ROCAlarmPri	1 到 999	1
ROCAlarmState	是，否，开，或关	否
ROCAlarmValue	任何有效的百分比	0
ROCTimeBase	秒，分，或小时	分

报警类型的报警约束标记名	容许值	缺省
HiAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
HiHiAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
LoAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
LoLoAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
MajDevAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
MinDevAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
ROCAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无

禁用标志	容许值	缺省
HiAlarmDisable	1, 0	0
HiHiAlarmDisable	1, 0	0
LoAlarmDisable	1, 0	0
LoLoAlarmDisable	1, 0	0
MajDevAlarmDisable	1, 0	0
MinDevAlarmDisable	1, 0	0
ROCAlarmDisable	1, 0	0

:Type 和关键字	容许值	缺省
:MemoryMsg		
Comment	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
InitialMessage	任意文本字符串	无
Logged	开、关、是或否	否
MaxLength	1-131 字符	131
RetentiveValue	开、关、是或否	否
Comment	任意文本字符串	无

AccessName	有效访问名字	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
InitialMessage	任意文本字符串	无
ItemName	有效的项目名	无
ItemUseTagname	真、假、是或否	否
Logged	开、关、是或否	否
MaxLength	1-131 字符	131
ReadOnly	是或否	否
RetentiveValue	开、关、是或否	否

:Type 和关键字	**容许值**	**缺省**
:GroupVar		
Comment	任意文本字符串	无
Group	有效的组名	$System
EventLogged	开、关、是或否	否

:Type 和关键字	**容许值**	**缺省**
:HistoryTrend		
Comment	任意文本字符串	无
Group	有效的组名	$System

:Type 和关键字	**容许值**	**缺省**
:TagID		
Comment	任意文本字符串	无
Group	有效的组名	$System

:Type 和关键字	**容许值**	**缺省**
:IndirectDisc		
Comment	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
RetentiveValue	开、关、是或否	否

:Type 和关键字	容许值	缺省
:IndirectAnalog		
Comment	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
RetentiveValue	开、关、是或否	否

:Type 和关键字	容许值	缺省
:IndirectMsg		
Comment	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
RetentiveValue	开、关、是或否	否

:Type 和关键字	容许值	缺省
:Access		
AdviseActive	是或否	是
NodeName	有效网络节点名	无
Application	有效 I/O 应用程序名	无
Topic	有效 I/O 主题名	无
DDEProtocol	“是”指定 DDE 协议， “否”指定 SuiteLink 协议	是

:Type 和关键字	容许值	缺省
:AlarmGroup		
Comment	任意文本字符串	无
EventLogged	开、关、是或否	否

EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
RetentiveValue	开、关、是或否	否

:Type 和关键字	**容许值**	**缺省**
:AlarmDotfields		
AlarmAckModel	0, 1, 2	0
DscAlarmDisable	1, 0	0
DscAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
HiAlarmDisable	1, 0	0
HiHiAlarmDisable	1, 0	0
LoAlarmDisable	1, 0	0
LoLoAlarmDisable	1, 0	0
MajDevAlarmDisable	1, 0	0
MinDevAlarmDisable	1, 0	0
ROCAlarmDisable	1, 0	0
HiAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
HiHiAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
LoAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
LoLoAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
MajDevAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
MinDevAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无
ROCAlarmInhibitor	标记名引用：任意离散或模拟型标记名	无

第 5 章

创建动画链接

一旦您创建了图形对象或符号，就可以通过动画来赋予其“生命”。通过附加动画链接，您可以改变对象或符号的外观，以反映标记名或表达式值的变化。例如，您可以创建一个泵符号，当关闭时显示为红色，当打开时显示为绿色。您也可以将此泵符号做成一个触控按钮，操作员通过点击鼠标或触摸（当用触摸屏时）可打开或关闭此泵。通过为您的对象或符号定义动画链接，您就能使用这些和许多其它的特殊效果。

InTouch 支持两种基本的链接：触动链接和显示链接。触动链接允许操作者向系统输入。显示链接允许向操作员输出。数值游标或按钮是触动链接的例子。填色、位置或闪烁链接是显示链接的例子。

本章将向您介绍创建各种类型的动画链接的步骤。

目录

- 通用动画链接功能
- 创建触动链接
- 创建用户输入触动链接
- 创建显示链接
- 创建值显示链接

通用动画链接功能

许多动画链接共享下面的通用控件：

对象类型对话框

通用调色板

快速访问标记浏览器

快速访问标记名点域

支持等价键

支持在“**标记名**”或“**表达式**”输入框中右击鼠标（显示一个包含可用于选定文本的命令菜单）。

对象类型对话框

“**对象类型**”对话框显示在屏幕的顶部，“链接选择”对话框的上方。标题对话框对创建的所有链接是通用的。它显示您为动画链接附件选定的对象类型的描述。例如，**按钮**。

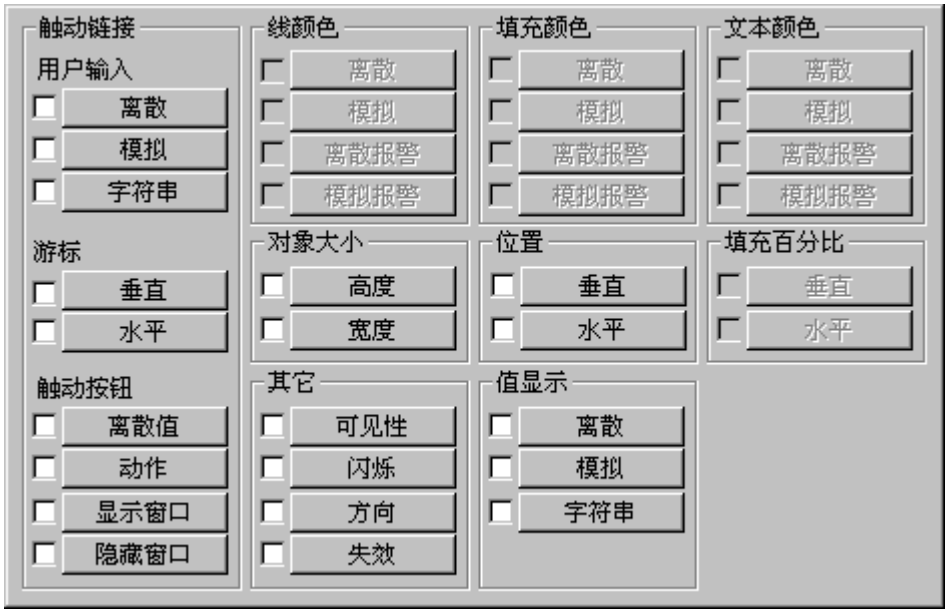


如果多个链接附加到一个对象，您可以单击“**前一链接**”和“**后一链接**”来快速向前或向后翻页，浏览附加到对象的每个链接的链接对话框。

链接按其最初附加到对象的顺序保存。

动画链接选择对话框

您可以为对象或符号定义多个链接。通过组合各种链接，您可以创建可想象的几乎任何屏幕动画效果。您可以改变对象的颜色、大小、位置、可见性、填充级别等等。



指定等价键

您可以指定键盘上的某个键来激活某些动画链接。仅当此链接的对象可见或被选定时，等价键才可操作。

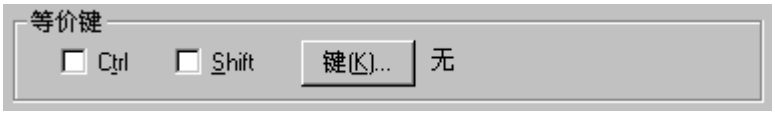
如果对象具有可见性或禁用链接，则当对象不可见或被禁用时，等价键将不可用。

您可以在多个窗口中定义相同的键。但是，只有最近打开的窗口中定义的键会被激活。在窗口重叠的情况下，顶部窗口的键会被激活。

注意：如果活动窗口中的任何对象或动作按钮被指定给用于**键动作脚本**的相同键，此键在活动窗口中的等价键链接将优先于**键动作脚本**执行。

有关键动作脚本的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

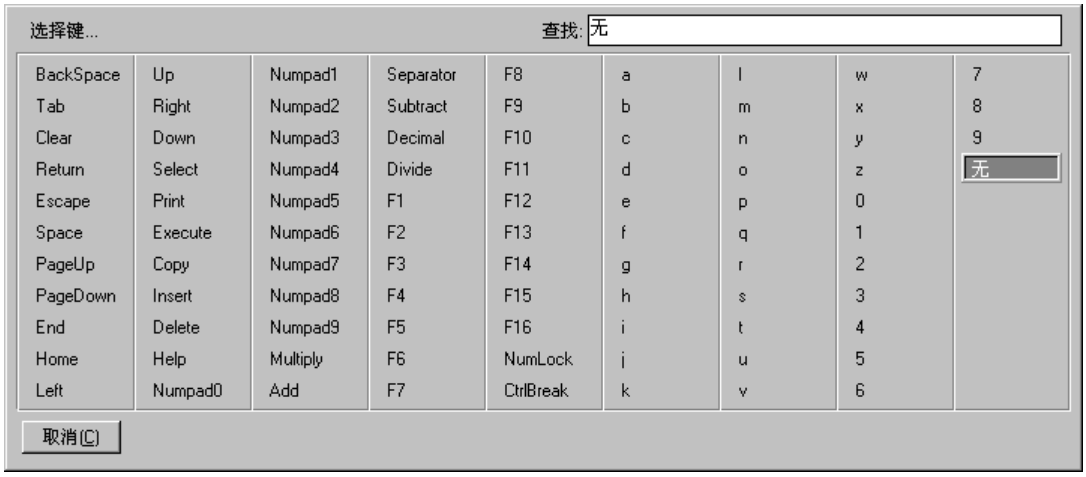
支持等价键的动画链接将在其链接对话框中显示**等价键**组。例如：



注意：直接选择键链接将只显示功能键 1-16。如果使用超过 16 个功能键的定制键盘，您需要从厂家获得设备驱动程序，以允许访问系统上的扩展功能键。

➢ **要对链接指定键：**

1. 如果您希望操作员在按下等价键的同时按住 **Ctrl** 与 **Shift** 键中的一个或两个键，请选择 **Ctrl** 和（或） **Shift** 键。
2. 单击“**键**”。“**选择键**”对话框出现：

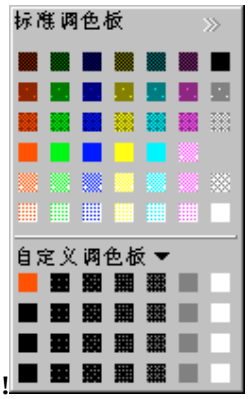


3. 单击您要指定给链接的键。对话框将关闭，链接对话框重新出现，显示“**键**”按钮旁边的选定键名称。

应用颜色链接

您可以对线条、矩形、圆角矩形、椭圆、多边线、多边形以及文本的动态属性应用颜色。当您创建线条、填充或文本对象的颜色时，您需要使用调色板来选择您希望链接到标记名值、标记名报警状态等的颜色。

线颜色和文本颜色链接必须使用纯色。如果您选择抖动（混合）色，缺省条件下，WindowMaker 会选择最接近的纯色。您可以创建自定义的调色板并将其加载到标准的 WindowMaker 调色板中。



当您将颜色链接附加到对象或符号，并单击链接对话框的颜色方框时，调色板将出现。

单击您要用于链接的颜色。调色板会自动关闭，所选颜色将显示在链接详细对话框的颜色方框中。

有关自定义调色板的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

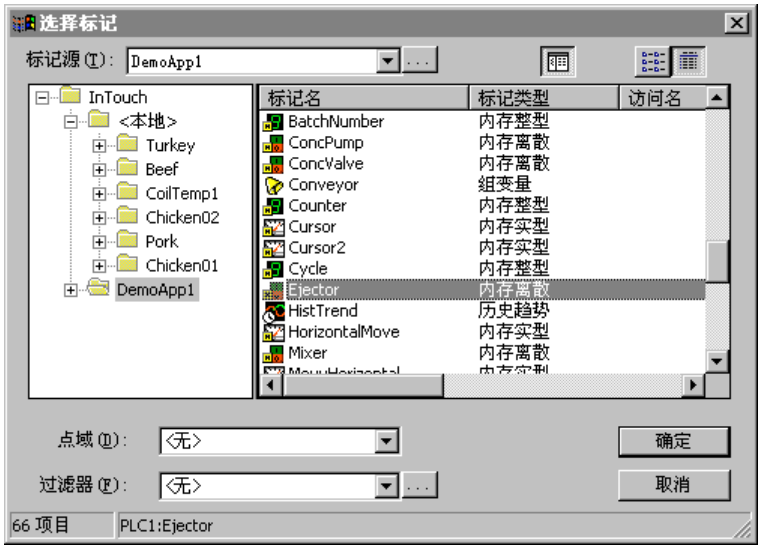
访问标记浏览器

当您通过访问标记浏览器来创建动画链接时，您可以快速查看应用程序中定义的所有标记名。如果您从标记浏览器选择需要指定给链接的标记名，所选标记名会自动插入到“**标记名**”或“**表达式**”框中。

➢ **要访问标记浏览器：**

1. 双击任意空白动画链接的“**标记名**”或“**表达式**”输入框。标记浏览器出现。

2. 单击  工具，在树形视图窗格中显示所有已定义的标记源：



   如果您不使用树形视图模式，单击“**标记源**”箭头，选择您要显示在列表中的标记源名称。标记浏览器将刷新，并显示所选标记源的标记名。

3. 选择您要用于链接的标记名，然后单击“**确定**”，或者双击标记名以同时选定，关闭标记浏览器并将其插入“**标记名**”或“**表达式**”框。

   要使用所选标记名的**点域**，单击“**点域**”箭头并从列表中选择您要使用的**点域**，然后单击“**确定**”。

   要显示标记名的数据库定义，在“**标记名**”或“**表达式**”框中输入标记名，然后双击它。“**标记名字典**”对话框出现，并显示标记名的定义。

有关标记浏览器的详细信息，请参阅第 4 章“标记名字典”。

访问标记名点域

您可以使用两种方法从动画链接“**标记名**”或“**表达式**”输入框访问标记名**点域**。这两种方法如下所述。

➢ **要通过标记浏览器访问标记名点域。**

1. 双击空白“**标记名**”或“**表达式**”输入框。标记浏览器出现，显示为当前标记源定义的标记名：



2. 单击“**点域**”箭头以打开**点域**列表，您可以将此列表与当前选定的标记名类型相关联。缺省条件下，所有类型的标记名最初均显示 **<无>**。

3. 在您要加入选定标记名的列表中，单击“**点域**”。

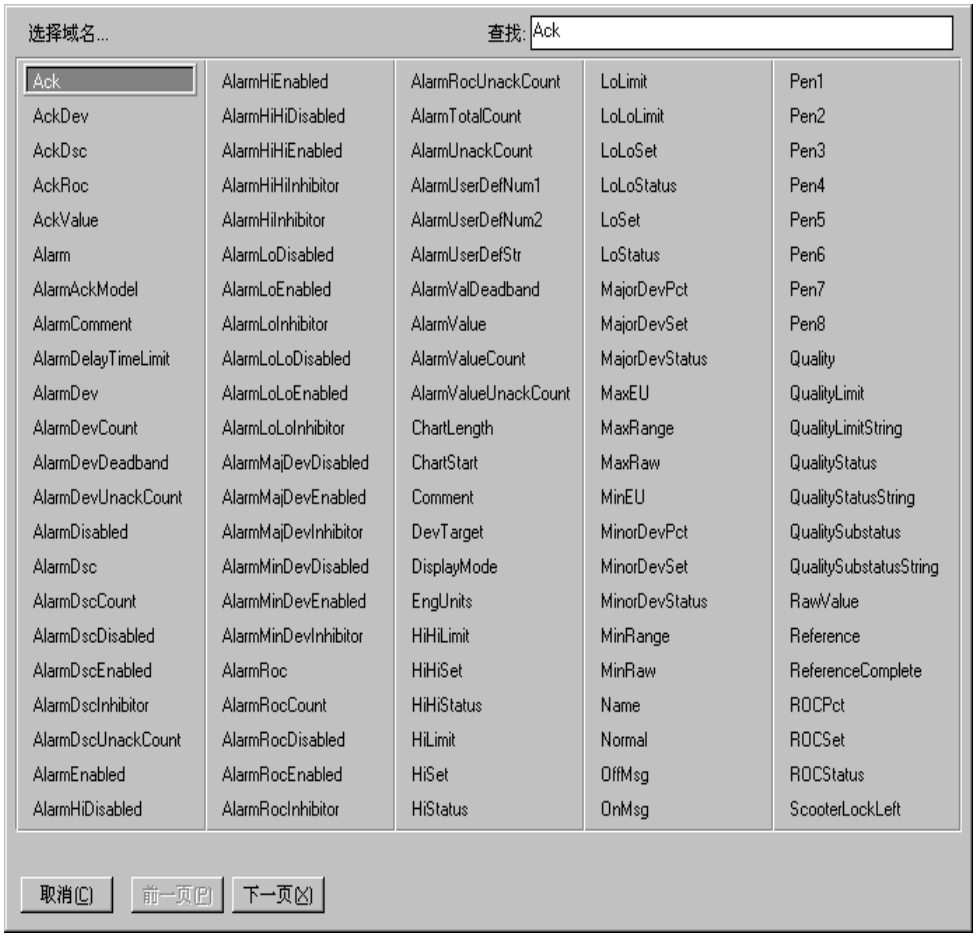
注意：不是每个标记名类型都有相同的**点域**。例如，**离散**类型的标记名有 **.OnMessage**，而模拟类型则没有。如果您选择**离散**类型的标记名并为其指定 **.OnMessage**，然后选择另一个**离散**类型的标记名，则显示的**点域**列表将不会改变。但是，如果您在控制视图列表中选择另一种类型的标记名，例如模拟类型，显示的**点域**将恢复为 **<无>**。

有关标记浏览器的详细信息，请参阅第 4 章“标记名字典”。

有关标记名**点域**的详细信息，请参阅您的联机 *《InTouch 参考指南》*。

➢ **要通过“选择域名”对话框来访问标记名点域：**

1. 在“**标记名**”或“**表达式**”输入框中，输入标记名并加上一个句点 **(tagname.)**，然后双击其右侧；或者只输入一个句点，然后双击其右侧。“**选择域名**”对话框出现，显示所有的标记名点域。



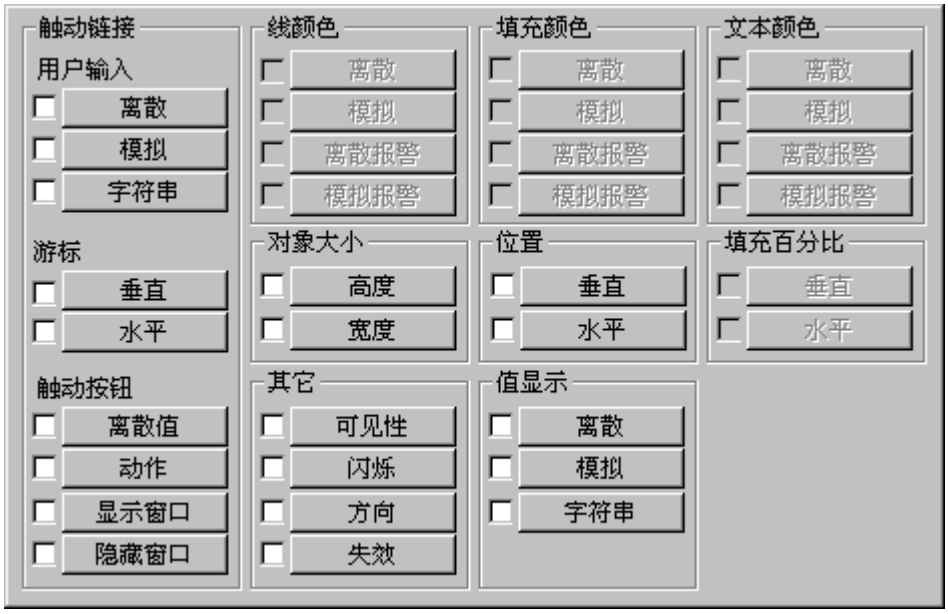
2. 选择您要使用的**点域**。对话框将关闭，所选**点域**将自动插入到“**标记名**”或“**表达式**”输入框中。

动画对象

➢ **要设置对象或符号的动画效果：**

1. 创建并选择您要设置动画的对象（线条、填充图案、文本、按钮或符号）。
2. 在“**特别**”菜单上，单击“**动画链接**”，或者双击对象。包含所有动画链接的对话框将出现：

   您也可以右击对象，然后单击“**动画链接**”。



3. 单击对应于您要附加到所选对象的链接的按钮。

   如果链接不适用于所选对象，则对应按钮将不会激活。

   单击复选框只能选择链接。单击链接名按钮以选择链接并打开其详细定义对话框。如果您单击链接名按钮并接受输入，复选框将自动选定。但是，如果您清除某个链接的复选框，该动画链接将从所选对象被移除。

4. 输入链接定义的详细资料，然后单击“**确定**”。“**链接选择**”对话框重新出现，您可以视需要创建对象的其它链接。
5. 单击“**确定**”以接受对象的所有链接并关闭对话框。

   当创建动画链接时，您在动画链接的标记名或表达式框中输入的标记名必须已在标记名字典中定义，才能将链接指定给标记名。否则会出现一个消息框，询问您是否要现在定义该标记名。单击“**是**”，标记名字典将出现，您即可以定义标记名。

创建触动链接

您可以对希望在运行时“触控”的对象或符号使用“**触动链接**”。此链接允许操作员将数据输入系统。例如，操作员可以打开或关闭阀、输入新的报警值、运行复杂的逻辑脚本，或者使用文本字符串登录，等等。

触动链接在运行时易于识别，因为当您在触控对象上移动光标，或者按 TAB 键从一个对象移动到另一对象时，触控对象上总是被一个“框”包围。如果触动链接对象或符号包含重叠的文本对象，则最顶部的文本对象将用于显示数据值。

通过单击、触摸对象（使用触摸屏时）、按下指定的等价键，或者当对象被“框”起时按下 ENTER 键，操作员可以激活触控按钮。

您可以创建九种触动链接：

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

注意：如果这些链接（三维按钮除外）使用的对象或符号包含文本字段，则当文本对象显示在 WindowViewer 中时，系统会应用当前为文本设置的所有属性（对齐方式、样式、字体等）。当使用文本字段来输入值时，屏幕也会显示输出值，除非启用了相应标记名详细资料对话框中的“**仅输入**”选项。

下面的章节介绍如何创建各种触动链接。

创建用户输入触动链接

您可以使用“**用户输入触动链接**”来创建允许操作员输入系统的触控对象。例如，使用按钮可改变离散状态、模拟值或者安全性登录。“**用户输入**”触动链接有三种类型：

用户输入	描述
离散型	用于控制离散型标记名的值。当此链接被激活时，会出现一个对话框，提示操作员作出选择。
模拟型	用于输入模拟（整型或实型）标记名的值。当此链接被激活时，会出现一个输入框，您可以从标准键盘或可选的屏幕键盘输入值。
字符串	用于创建可输入字符串消息的对象。

注意：在运行时，当链接被激活时，会出现一个用于输入消息值的输入框，或出现一个可选屏幕键盘。

➢ **要创建离散输入链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**用户输入**”区域，单击“**离散**”。“**输入->离散标记名**”对话框出现：



3. 在“**标记名**”框中，输入离散型标记名。

   右击“**标记名**”框，以访问可应用于选定文本的命令。

4. 如果您要为链接指定一个等价键，单击“**键**”。

   有关指定键的详细信息，请参阅“指定等价键”。

5. 在“**用户消息**”框中，键入当输入链接激活时，您希望出现在输入对话框中的消息。

6. 在“**置位提示**”和“**复位提示**”框中，输入您希望在按钮上显示的消息，操作员可在输入对话框中单击此按钮来打开或关闭离散值。

7. 在“**打开消息**”和“**关闭消息**”框中，输入当对象开或关时，您希望显示在对象关联文本字段（如果有的话）中的消息。

8. 如果您不想让输入内容显示在与对象关联的文本字段中，请选择“**仅输入**”（此选项仅适用于具有关联文本字段的对象）。

9. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

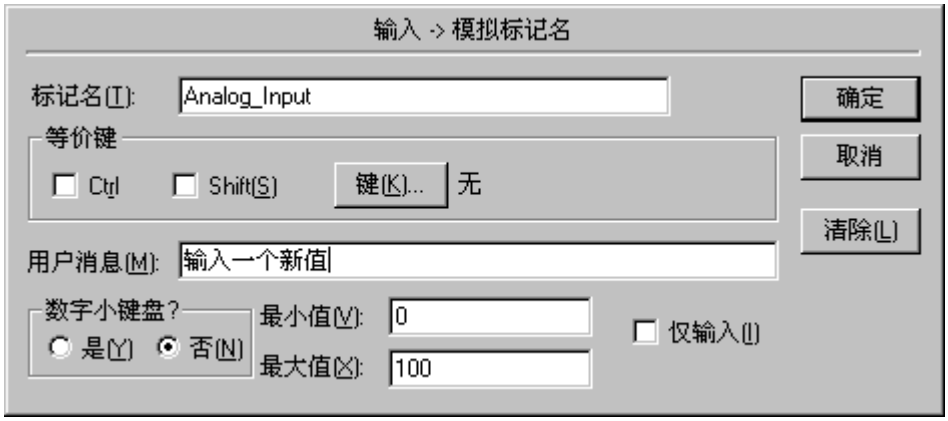
注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

➢ **要创建模拟输入链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**用户输入**”区域，单击“**模拟**”。“**输入-> 模拟标记名**”对话框出现：



注意：如果此链接使用文本字段，则必须适当格式化以正确显示模拟（整数或实数）值的输出。

有关格式化文本字段的详细信息，请参阅第 2 章“使用 WindowMaker”。

3. 在“**标记名**”框中，输入模拟（整型或实型）标记名。

   右击“**标记名**”框，以访问可应用于选定文本的命令。

4. 要指定链接的等价键，单击“**键**”。

   有关指定键的详细信息，请参阅“指定等价键”。

5. 如果您要在激活该链接时显示可选小键盘，请在“**用户消息**”框中，输入您希望出现在小键盘上的提示消息。

6. 如果您希望在屏幕上显示一个数字小键盘来输入字符串的新值，请选择“**是**”。

7. 在“**最小值**”框中，输入标记名的最小输入值。

8. 在“**最大值**”框中，输入标记名的最大输入值。

9. 如果您不想让输入内容显示在与对象关联的文本字段中，请选择“**仅输入**”（此选项仅适用于具有关联文本字段的对象，例如三维按钮）。

10. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

➢ **要创建字符串输入链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**用户输入**”区域中，单击“**字符串**”。“**输入->字符串标记名**”对话框出现：



3. 在“**标记名**”框中，输入消息型标记名。

   右击“**标记名**”框，以访问可应用于选定文本的命令。

4. 如果您要为链接指定一个等价键，单击“**键**”。

   有关指定键的详细信息，请参阅“指定等价键”。

5. 如果您要在激活该链接时显示可选小键盘，请在“**用户消息**”框中，输入您希望显示在小键盘上的提示消息。

6. 如果您希望输入的字符串照原样显示在屏幕上，请对“**回显字符？**”选项选择“**是**”。如果因数据保密（例如口令）而不能显示在屏幕上，请选择“**否**”。

7. 如果您希望在屏幕上显示小键盘来输入字符串的新值，请对“**小键盘？**”选项选择“**是**”。

8. 如果您不想让输入内容显示在与对象关联的文本字段中，请选择“**仅输入**”（此选项仅适用于具有关联文本字段的对象，例如三维按钮）。

   注意：当初次启动 WindowViewer 时，字符串会显示您定义与该对象链接的标记名时在“**初始值**”框中输入的文本。

9. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

创建游标触动链接

您可以使用“**游标触动链接**”来创建对象或符号，用鼠标或其它指针设备（例如触模屏上的手指）可以在窗口中移动这些对象或符号。当移动对象或符号时，与之链接的标记名值会改变。这就使您可以创建用于设置系统值的装置。

一个对象可以具有水平或垂直游标触动链接，或同时具备两种链接。通过在单个对象上使用两种链接，您可以同时改变两个模拟标记名的值。

注意：水平和垂直游标链接使用相同的方法创建。下面将介绍“**水平游标**”链接。

➢ **要创建水平（或垂直）游标链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**游标**”区域中，单击“**水平**”。“**水平游标**”对话框出现：



3. 在“**标记名**”框中，输入模拟（整型或实型）标记名。

   右击“**标记名**”框，以访问可应用于选定文本的命令。

4. 在“**在左端**”框中，输入游标到达最左端位置时的标记名值。

5. 在“**在右端**”框中，输入游标到达最右端位置时的标记名值。

6. 在“**向左**”框中，输入游标可向左移动的象素数。

   在最左端，标记名的值等于“**在左端**”字段中输入的值。

7. 在“**向右**”框中，输入游标可向右移动的象素数。

   在最右端，标记名的值等于在“**在右端**”字段中输入的值。

8. 选择对象上的“**引用位置**”，光标将锁定在该位置来移动对象。

9. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

创建触动按钮触动链接

如果需要创建用鼠标单击或触摸（当使用触摸屏时）时可立即执行操作的对象链接，您可以使用“**触动按钮触动链接**”。这些操作可以是改变**离散值**、执行**动作脚本**，**显示窗口**或**隐藏窗口**命令。下面是四种**触动按钮**链接类型：

触动按钮	描述
离散值	用于将任何对象或符号设置成用于控制离散标记名状态的按钮。按钮动作可以是设置、重置、切换、瞬间打开（直接）和瞬间关闭（取反）类型。
动作	允许任何对象、符号或按钮链接最多三种不同的动作脚本：按下时、按下期间和释放时。 动作脚本可用于将标记名设置为特定的值、显示和（或）隐藏窗口、启动和控制其它应用程序、执行函数等。
显示窗口	用于将对象或符号设置成单击或触摸时可打开一个或多个窗口的按钮。
隐藏窗口	用于将对象或符号设置成单击或触摸时可关闭一个或多个窗口的按钮。

➢ **要创建离散值触动按钮链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**触动按钮**”区域中，单击“**离散值**”。“**按钮->离散值**”对话框出现：



3. 在“**标记名**”框中，输入离散型标记名。

   右击“**标记名**”框，以访问可应用于选定文本的命令。

4. 如果您要为链接指定一个等价键，单击“**键**”。

5. 有关指定键的详细信息，请参阅“对动画链接指定键”。

6. 选择您要用于按钮的“**动作**”选项，如下所示：

动作	描述
直接	每当按下或按住按钮时，将值设置为 1（真、开、是）；释放按钮时，值自动复位为 0（假、关、否）。
取反	按下或按住按钮时，将值设置为 0（假、关、否）；释放按钮时，值自动复位为 0（真、开、是）。
切换	当按下按钮时，反转离散标记名的状态。例如，如果标记名为“1”时按下按钮，值将重置为“0”，反之亦然。
复位	当按下按钮时，将值设置为 0（假、关、否）。
置位	当按下按钮时，将值设置为 1（真、开、是）。

7. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

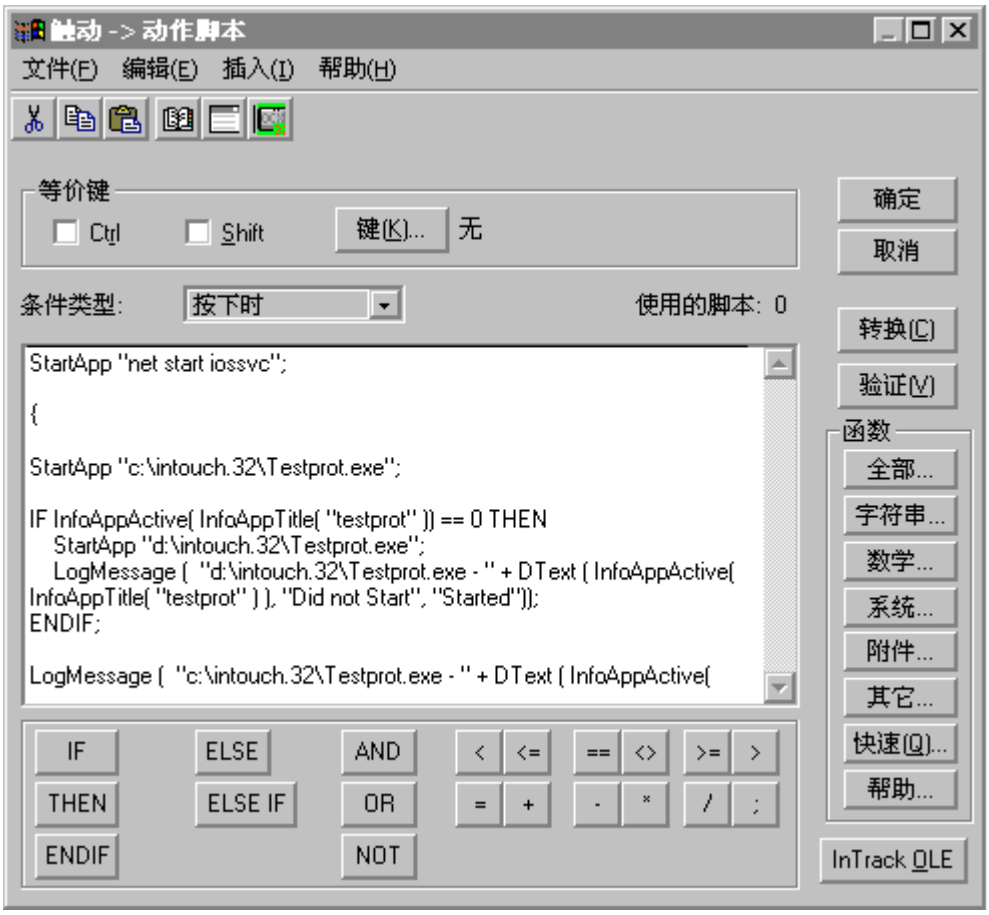
注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

➢ **要创建动作触动按钮链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**触动按钮**”区域中，单击“**动作**”。“**触动 -> 动作脚本**”编辑器出现：



有关编写 QuickScript 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

3. 单击“**条件类型**”箭头，选择您要应用于对象的脚本类型。您可以对到相同的键应用以下所有三种脚本类型：

   按下时 刚按下键时，执行一次脚本。

   按下期间 在按住键时，按一定的时间间隔连续执行脚本。

   释放时 释放键时，执行一次脚本。

   “**按下期间**”脚本在指定的毫秒数之后开始执行。要立即执行脚本，您需要创建两份“**按下时**”脚本。

注意：如果活动窗口中的任何对象或动作按钮被指定给用于**键脚本**的相同键，此键在活动窗口中的等价键链接将优先于**键脚本**执行。

有关脚本的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 在脚本编辑器窗口中单击，输入当对象激活时您希望执行的脚本。

5. 单击“**确定**”，将脚本附加到对象并返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

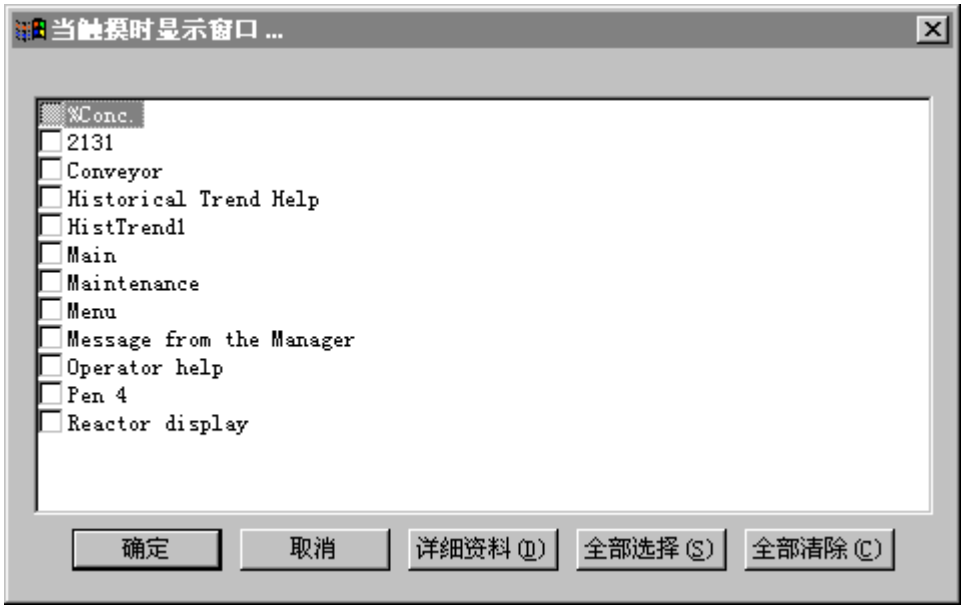
➢ **要创建显示（或隐藏）窗口触动按钮链接：**

注意：显示窗口和**隐藏窗口**链接使用相同的方法创建。下面向您介绍**显示窗口**链接。

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现：

   - 要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**触动按钮**”区域中，单击“**显示窗口**”。“**当触摸时显示窗口**”对话框出现：



3. 选择当单击或触摸对象时，您要打开的窗口。

4. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

   - 如果屏幕显示一个以上的窗口，而其中的一个窗口为**替换**类型，则与之相交的任何其它窗口将会在显示前关闭（如同**显示窗口**动画链接失效一样）。

     要改变窗口类型，右击打开窗口的空白区域，然后单击“**窗口属性**”。“**窗口属性**”对话框出现，您即可以改变类型（如果 WindowViewer 正在运行，则您不能改变窗口类型）。

   有关窗口属性的详细信息，请参阅第 2 章“使用 WindowMaker”。

创建显示链接

您可以使用各种**显示链接**来为操作员提供输出。显示链接有下列八种类型：

显示链接	类型
线颜色、填充颜色和文本颜色	离散、模拟、离散报警、模拟报警。
对象大小	高度、宽度
位置	水平、垂直
填充百分比	水平、垂直
其它	可见性、方向、闪烁、失效
值显示	离散、模拟、字符串

下面的章节将向您介绍如何创建各种显示链接。

创建颜色链接

您可以使用颜色链接对对象的“**线颜色**”、“**填充颜色**”和“**文本颜色**”属性设置动画效果。

注意：线颜色和文本颜色链接必须使用纯色。如果您选择抖动（混合）色，缺省条件下，WindowMaker 会选择最接近的纯色。为避免混合颜色，显卡的最低配置必须为 2MB，颜色深度限制必须高于 256 色，例如 32K 或 65K（有时也叫“高彩色”）。

有关调色板的详细信息，请参阅“应用颜色链接”。

上述每种颜色属性都可以通过定义属性的颜色链接动态地进行设置。颜色属性可以与离散量表达式、模拟量表达式、离散量报警状态或模拟量报警状态的数值相链接。线颜色、填充颜色和文本颜色有四种类型：

颜色链接	描述
离散型	用于控制与离散表达式值相链接的对象或符号的填充颜色、线颜色和文本颜色属性。
模拟型	对象或符号的线颜色、填充颜色和文本颜色可以与模拟标记名（整型或实型）或模拟表达式的值相链接。系统通过指定四个断点来确定五个值范围。您可以选定随值范围改变而显示的五种不同颜色。
离散报警	对象的文本、线和填充颜色均可以与标记名、报警组或组变量的报警状态相链接。此颜色链接允许您选择两种颜色，一种用于标记名的标准状态，另一种用于报警状态。此链接可用于模拟型和离散型两种标记名。如果用于模拟型标记名，则它可以响应标记名的任何报警条件。
模拟报警	对象的文本、线和填充颜色均可以与标记的报警状态、报警组或组变量相链接。允许为常规状态设置特定的颜色，也可以为标记名定义的各种已定义报警条件设置单独的颜色。 **注意：**如果模拟报警动画链接所使用的远程标记名是访问 InTouch 第 7.0 版以前创建的未转换应用程序中的标记名信息，则对象不会进入报警状态。

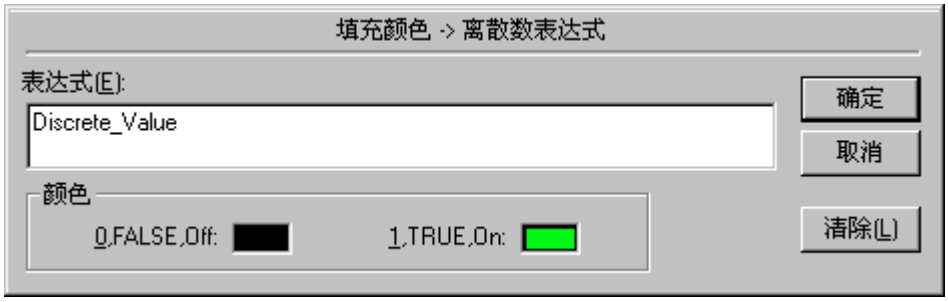
➢ **要创建离散填充颜色链接：**

注意：所有的**线颜色**和**文本颜色**链接均使用与**填充颜色**相同的方法创建。下面将向您介绍如何创建**填充颜色**链接。

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**填充颜色**”（或“**线颜色**”或“**文本颜色**”）区域中，单击“**离散**”。“**填充颜色->离散数表达式**”对话框出现：



3. 在“**表达式**”框中，输入离散型标记名或值为真或假的表达式。

   离散表达式也可以包含模拟型标记名。例如 TankLevel >= 75。在此例中，当变量“TankLevel”值大于或等于“75”时，对象的填充颜色将改变。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 在“**颜色**”组中，单击各个颜色方框以打开调色板。单击调色板中您要用于每种标记名状态的颜色。

注意：线颜色和文本颜色链接必须使用纯色。如果您选择抖动（混合）色，缺省条件下，WindowMaker 会选择最接近的纯色。为避免混合颜色，显卡的最低配置必须为 2MB，颜色深度限制必须高于 256 色，例如 32K 或 65K（有时也叫“高彩色”）。

有关调色板的详细信息，请参阅“应用颜色链接”。

5. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

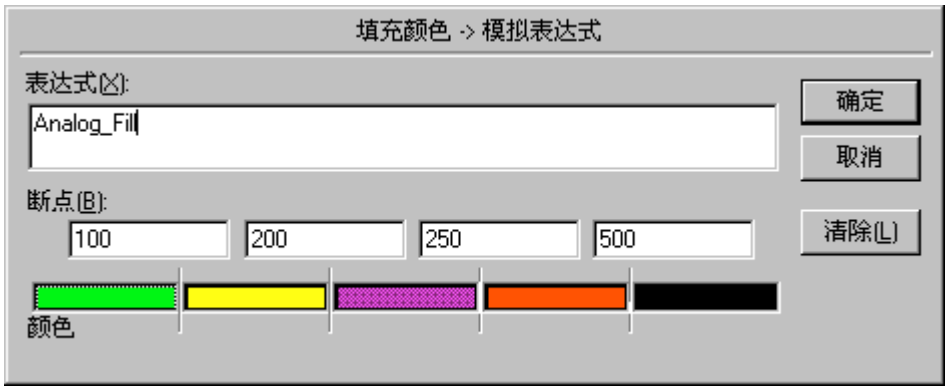
注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

➢ **要创建模拟表达式颜色链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**填充颜色**”（或“**线颜色**”或“**文本颜色**”）区域中，单击“**模拟**”。“**填充颜色->模拟表达式**”对话框出现：



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。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 在每个“**断点**”框中，您可以指定对象将改变颜色的断点值（小数对于实型标记名有效）。

   您不需要使用 4 个不同的值。例如，如果您只希望对象改变 3 次颜色，可输入 3 个值，而对第 3 和第 4 个值用相同的颜色。

5. 在“**颜色**”组中，单击各个颜色方框以打开调色板。单击调色板中您要用于每个断点的颜色。

注意：线颜色和文本颜色链接必须使用纯色。如果您选择抖动（混合）色，缺省条件下，WindowMaker 会选择最接近的纯色。为避免混合颜色，显卡的最低配置必须为 2MB，颜色深度限制必须高于 256 色，例如 32K 或 65K（有时也叫“高彩色”）。

有关调色板的详细信息，请参阅“应用颜色链接”。

6. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

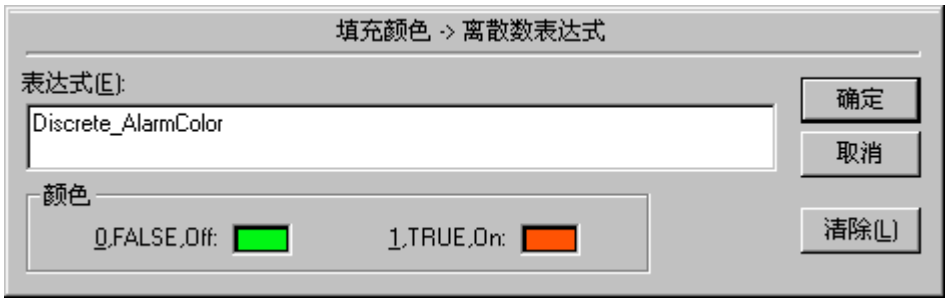
注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

➢ **要创建离散报警状态颜色链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**填充颜色**”（或“**线颜色**”或“**文本颜色**”）区域中，单击“**离散报警**”。“**填充颜色->离散标记名报警状态**”对话框出现：



3. 在“**标记名**”框中，输入您要将其报警状态与对象相关联的离散标记名。

   右击“**标记名**”框，以访问可应用于选定文本的命令。

4. 在“**颜色**”组中，单击各个颜色方框以打开调色板。单击调色板中您要用于每种颜色状态的颜色。

注意：线颜色和文本颜色链接必须使用纯色。如果您选择抖动（混合）色，缺省条件下，WindowMaker 会选择最接近的纯色。为避免混合颜色，显卡的最低配置必须为 2MB，颜色深度限制必须高于 256 色，例如 32K 或 65K（有时也叫“高彩色”）。

有关调色板的详细信息，请参阅“应用颜色链接”。

5. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

➢ **要创建模拟报警状态颜色链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**填充颜色**”（或“**线颜色**”或“**文本颜色**”）区域中，单击“**模拟报警**”。“**填充颜色->模拟标记名报警状态**”对话框出现：



3. 在“**标记名**”框中，输入您要将其报警状态与对象相关联的模拟（整型或实型）标记名。

   右击“**标记名**”框，以访问可应用于选定文本的命令。

4. 在“**报警类型**”组中，选择您要与对象相关联的报警类型。您可以使用三种互相独立的模拟颜色链接类型：

报警类型	描述
值报警	取决于为标记名（见上例）定义的值报警状态，您最多可以选择 5 种不同的颜色。
偏差	取决于为标记名（见上例）定义的偏差报警状态，您最多可以选择 3 种不同的颜色。
ROC（变化率）	取决于为标记名（见上例）定义的变化率报警状态，您最多可以选择 2 种不同的颜色。

5. 在“**颜色**”组中，单击各个颜色方框以打开调色板。单击调色板中您要用于每种颜色状态的颜色。

注意：线颜色和文本颜色链接必须使用纯色。如果您选择抖动（混合）色，缺省条件下，WindowMaker 会选择最接近的纯色。为避免混合颜色，显卡的最低配置必须为 2MB，颜色深度限制必须高于 256 色，例如 32K 或 65K（有时也叫“高彩色”）。

有关调色板的详细信息，请参阅“应用颜色链接”。

6. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

创建对象大小链接

如果您需要根据模拟（整型或实型）标记名或模拟表达式的值来改变对象的高度和（或）宽度，则可以使用“**对象大小**”链接。“对象大小”链接通过设置链接的“锚点”，控制对象高度和（或）宽度及伸缩方向。同一对象可以同时添加高度和宽度链接。

注意：高度和宽度链接的创建方式相同。下面向您介绍**高度**链接。

➢ **要创建高度（或宽度）链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**对象大小**”区域中，单击“**高度**”。“**对象高度-> 模拟量**”对话框出现：



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。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 在“**最大高度值**”框中，输入可使对象达到最大高度的标记名或表达式的值。

5. 在“**最小高度值**”框中，输入可使对象达到最小高度的标记名或表达式的值。

6. 在“**最大 % 高度**”框中，输入当标记名或表达式到达“**最大高度值**”字段中设置的值时，对象的高度百分比 (0-100)。

7. 在“**最小 % 高度**”框中，输入当标记名或表达式到达“**最小高度值**”字段中设置的值时，对象的高度百分比 (0-100)。

   - 高度百分比值以对象实际绘制大小（设为 100%）的百分比来表示。

8. 选择“**锚**”点，对象高度将从此点开始增长。

   - 选择“**顶部**”使对象从顶端往下增长。选择“**中间**”使对象从中间点向两个方向往外增长。选择“**底部**”使对象从底端往上增长。

9. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

创建位置链接

您可以使用“**位置链接**”，使对象根据模拟标记名或表达式值的改变，自动地水平、垂直或同时朝水平和垂直两个方向相应移动。

注意：水平位置和**垂直位置**链接的创建方式相同。下面向您介绍**水平位置**链接。

➢ **要创建水平位置链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**位置**”区域中，单击“**水平**”。“**水平位置**”对话框出现：



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。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 在“**在左端**”框中，输入对象位于最左端位置时的标记名值。
5. 在“**在右端**”框中，输入对象位于最右边位置时的标记名值。
6. 在“**到左边**”框中，输入对象可向绘制位置左侧移动的象素数。

   在最左端，标记名的值等于“**在左端**”字段中输入的值。

7. 在“**到右边**”框，输入对象可向绘制位置右侧移动的象素数。

   在最右端，标记名的值等于在“**在右端**”字段中输入的值。

8. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

创建填充百分比链接

您可以使用“**填充百分比链接**”，以根据模拟标记名的值或用于计算模拟值的表达式，来改变填充图案（或包含填充图案的符号）的填充级别。例如，您可以使用此链接来显示容器中的液面高度。对象或符号可以具有水平填充链接、垂直填充链接，或同时具备二者。

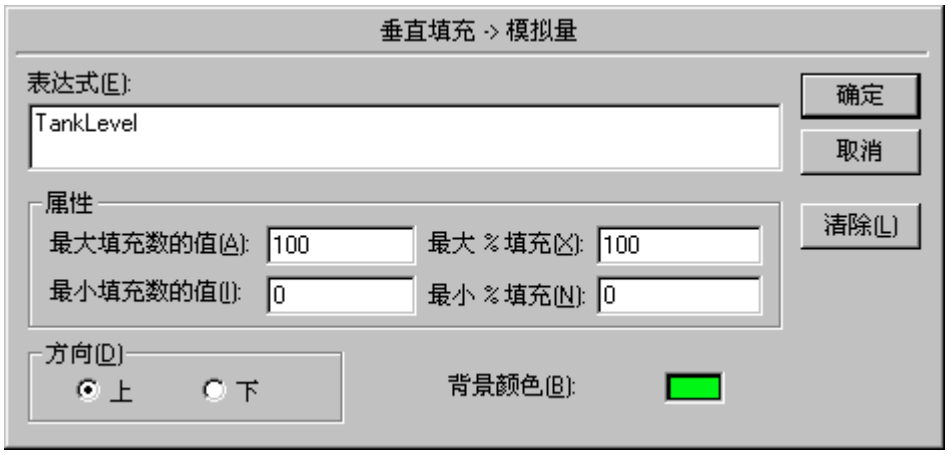
注意：水平和**垂直填充百分比**链接的创建方式相同。下面向您介绍**垂直填充百分比**链接。

➢ **要创建垂直填充百分比链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**填充百分比**”区域中，单击“**垂直**”。“**垂直填充 -> 模拟量**”对话框出现：



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。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 在“**最大填充数的值**”框中，输入可使对象填充到最大级别的表达式的值。

5. 在“**最小填充数的值**”框中，输入可使对象填充到最小级别的表达式的值。

6. 在“**最大 % 填充**”框中，输入当表达式到达“**最大填充数的值**”框中设置的级别时，对象填充的百分比 (0-100)。

   如果表达式的值大于此值，则它将被忽略。

7. 在“**最小 % 填充**”框中，输入当表达式到达“**最小填充数的值**”框中设置的级别时，对象填充的百分比 (0-100)。

   如果表达式的值大于此值，则它将被忽略。

8. 选择填充对象的**方向**。

   如果选择“**上**”，则对象将从下向上填充。如果选择“**下**”，则对象将从上向下填充。

9. 在“**背景颜色**”框中，打开调色板。单击所需的颜色，调色板将从屏幕上消失。

   所选“**背景颜色**”将用作对象“**未填充**”部分的颜色。实际填充的颜色是当您绘制对象时选择的颜色。如果您同时将“**垂直填充百分比**”和“**水平填充百分比**”链接添加到相同的对象，则在其中任一链接对话框中最后选择的颜色将用作背景颜色。

10. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

创建其它链接

系统还提供四种其它链接类型。

其它链接	描述
可见性	用于根据离散标记名或表达式的值来控制对象的可见性。
闪烁	用于根据离散标记名或表达式的值使对象闪烁。
方向	用来根据标记名或表达式的值使对象旋转。
失效	用于根据标记名或表达式的值使对象的触动功能失效。

- 常用作安全策略的一部分。

有关在应用程序中应用安全性的详细信息，请参阅第 3 章“创建分布式应用程序”。

➢ **要创建可见性链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现：
   - 要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。
2. 在“**其它**”区域中，单击“**可见性**”。“**对象可见性->离散值**”对话框出现：



3. 在“**表达式**”框中，输入一个等于离散值的离散标记名或表达式。
   - 离散表达式也可以包含模拟标记名，例如 TankLevel >= 75。在本例中，当标记名 TankLevel 的值大小或等于 75 时，对象将可在窗口中看见。

     右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 选择对象的“**可见状态**”。如果选择“**打开**”，则当表达式的值为真时，对象将不可见；如果选择“**关闭**”，则当表达式的值为真时，对象将可见。

5. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

➢ **要创建闪烁链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现：

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**其它**”区域中，单击“**闪烁**”。“**对象闪烁信号->离散值**”对话框出现：



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。

   离散表达式也可以包含模拟型标记名。例如 TankLevel >= 75。在本例中，当变量“TankLevel”的值大于或等于“75”时，对象将闪烁。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 选择对象的“**闪烁属性**”。

   如果您选择“**不可见闪烁**”，则对象/符号的闪烁方式是在窗口中消失然后重新出现。如果您选择“**具有这些属性的可见闪烁**”，则对象/符号将在窗口中保持可见，改变选定的颜色属性会产生闪烁效果。

   单击“**文本颜色**”、“**线颜色**”和“**填充颜色**”框以打开调色板。单击所需的颜色。调色板将从屏幕上消失。

   - 选择与对象的填充颜色相同的填充闪烁颜色可使对象不“闪烁”。

5. 选择对象的“**闪烁速度**”。

   - 要将闪烁速度配置为“**慢**”、“**中等**”和“**快**”，在“**特别**”菜单上，指向“**配置**”，然后单击“**WindowViewer**”。“**WindowViewer 通用**”属性页出现。在“**闪烁频率**”组中，输入您要用于速度的毫秒数。

     以这些设置所做的任何更改都是全局性的，将影响整个应用程序中所有闪烁链接的闪烁速度。

6. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

➢ **要创建方向链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现：

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**其它**”区域中，单击“**方向**”。“**方向->模拟量**”对话框出现：



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。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 在“**最大逆时针旋转时的值**”框中，输入要使对象旋转到最大逆时针位置的表达式值。

   如果表达式的值大于此值，则它将被忽略。

5. 在“**最大顺时针旋转时的值**”框中，输入要使对象旋转到最大顺时针位置的表达式值。

   如果表达式的值大于此值，则它将被忽略。

6. 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：

7. 在“**顺时针旋转**”框，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：

   - 对象将根据在 WindowMaker 中绘制时的原始位置顺时针或逆时针旋转。

     要强迫对象（如文本）旋转特定的角度，您只需设置“**最大逆时针旋转时的值**”为 360、“**最大顺时针旋转时的值**”为 0、“**逆时针旋转**”为 360、“**顺时针旋转**”为 0，然后在“**表达式**”框中输入角度值，例如 90（对应 90 度角）。请记住，如果未定义标记名，此表达式将永远不会改变，而且对象总是保持在 90 度位置。

     文本可以在 WindowMaker 中设置，但是不能在 WindowMaker 中按标记名的值旋转。

8. 在“**X 轴坐标**”框中，输入旋转中心点从对象中心点水平移动的象素数（正数值位于中心点的右侧）。

   - 方向链接以对象或符号的中心作为旋转中心。

9. 在“**Y 轴坐标**”框中，输入旋转中心点从对象中心点垂直移动的象素数（正数值位于中心点的左侧）。

10. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

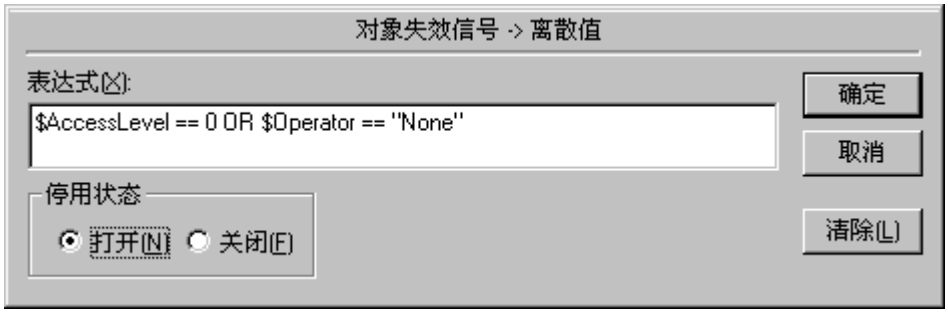
➢ **要创建失效链接：**

如果您要在应用程序中应用安全机制，失效链接将很有用处。例如，您可以根据登录操作员的访问级别或用户名来使某些对象失效。

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**其它**”区域中，单击“**失效**”。“**对象失效信号->离散值**”对话框出现：



3. 在“**表达式**”框中，输入离散标记名或一个等于离散值的表达式。

   通过使用上面的表达式，则在无人登录的情况下，对象或按钮将被禁用，从而防止被擅自篡改。

   离散表达式也可以包含模拟型标记名。例如 TankLevel >= 75。在本例中，当变量“TankLevel”的值大于或等于“75”时，对象将失效。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 选择“**失效状态**”，当离散标记名或表达式为真时，它将关闭或开启对象的功能。

   失效状态“打开”意味着只要表达式为真，那么对象或按钮的触动功能将被关闭，而且不能单击。

5. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

创建值显示链接

值显示链接提供了使用文本对象来显示离散、模拟或字符串标记名值的能力。值显示链接有三种类型：

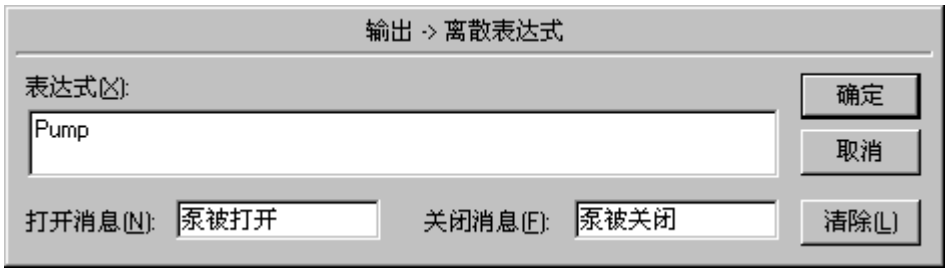
值显示类型	描述
离散型	使用离散表达式的值在文本对象中显示用户定义的开或关消息。
模拟型	在文本对象中显示模拟表达式的值。
字符串	在文本对象中显示字符串表达式的值。

➢ **要创建离散值显示链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现：

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**值显示**”区域中，单击“**离散**”。“**输出量->离散表达式**”对话框出现：



3. 在“**表达式**”框中，输入离散标记名或一个等于离散值的表达式。

   离散表达式也可以包含模拟型标记名。例如，TankLevel >= 75。在本例中，当变量“TankLevel”的值大于或等于“75”时，将显示相应的消息。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 在“**打开消息**”框中，输入当离散表达式的值等于 1（真、开、是）时，您希望显示的消息。
5. 在“**关闭消息**”框中，输入当离散表达式的值等于 0（假、关、否）时，您希望显示的消息。

   - 程序将使用为该对象设置的字体、大小、颜色、对齐方式和链接属性，在原始文本对象的位置显示这些消息。字段的初始内容不会影响运行时显示的消息。

6. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

- 您也可以使用“**值显示输出->字符串表达式**”链接来显示离散标记名的打开或关闭消息。对于此链接，您需要输入下面的表达式：

  DText (Pump, Pump.OnMsg, Pump.OffMsg);

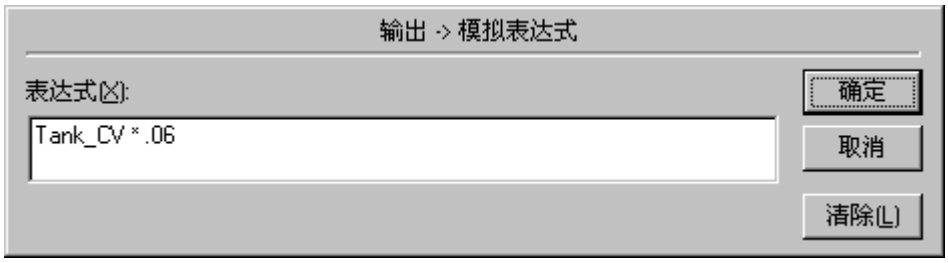
  在此表达式中，**.OnMsg** 和 **.OffMsg** 字符串将从 InTouch 标记名字典中离散型标记名 Pump 的定义中提取。

➢ **要创建模拟值显示链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现。

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**值显示**”区域中，单击“**模拟**”。“**输出量->模拟表达式**”对话框出现：



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。（您也可以在此表达式中使用离散型标记名，此时表达式将只显示 1 或 0）。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

有关格式化模拟显示对象的详细信息，请参阅第 2 章“使用 WindowMaker”。

➢ **要创建字符串值显示链接：**

1. 双击对象，或选定对象然后在“**特别**”菜单上，单击“**动画链接**”。链接选择对话框出现：

   要快速访问动画链接选择对话框，右击对象，然后单击“**动画链接**”。

2. 在“**值显示**”区域中，单击“**字符串**”。“**输出量->字符串表达式**”对话框出现：



3. 在“**表达式**”框中，输入消息型标记名或者等于消息型标记名的表达式。

   在上面的表达式中，函数 **Text()** 用于将整型标记名 TankLevel 的值转换成一个字符串。

   右击“**表达式**”框，以访问可应用于选定文本的命令。

注意：您最多可以为表达式输入 256 个字符。如果需要使用更长的表达式，您可以创建一个 QuickFunction，然后在表达式中调用该函数。

您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

有关 QuickFunction 的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

4. 单击“**确定**”以将链接附加到对象，或返回动画链接对话框。如果需要，您可以现在为对象添加其它链接。

注意：如果您输入的标记名未在标记名字典中定义（远程标记名除外），系统会提示您现在定义。

第 6 章

在 InTouch 中创建 QuickScript

InTouch 脚本是 InTouch 应用程序最强大的功能之一。InTouch QuickScript 功能允许在指定标准满足时执行特定的命令和逻辑操作。例如，按键、打开窗口和改变值等。

QuickFunction 是您可以从其它脚本和动画链接表达式调用的脚本。系统将可重复使用的代码存储在单个脚本和单个位置中，从而支持在一个编辑进程中更新所有脚本实例。

通过使用脚本，您可以创建各种自定义和自动系统函数。

目录

- InTouch QuickScript
- 使用 InTouch QuickScript 编辑器
- 应用程序脚本
- 窗口脚本
- 键脚本
- 触动按钮动作脚本
- 条件脚本
- 数据改变脚本
- ActiveX 事件脚本
- QuickFunction
- 使用局部变量
- 创建 FOR-NEXT 循环脚本
- 脚本编辑样式和语法
- 导入 QuickScript
- 打印脚本
- 脚本函数
- 脚本编辑器错误消息

InTouch QuickScript

所有的 InTouch QuickScript 都是事件驱动的。事件可以是数据改变、条件、单击鼠标和定时器等。处理的顺序取决于具体的应用程序。虽然由相同事件触发的多个脚本在计划方式上可能存在一定的固有顺序，但系统并不能提供任何特定顺序的担保。因而，您不应该建立任何依赖于处理顺序的脚本。

下面简要介绍您可以创建的脚本类型：

脚本类型	描述
应用程序	链接到整个应用程序。
窗口	链接到特定窗口。
键	链接到键盘上的特定键或键组合。
条件	链接到离散型标记名或表达式。
数据改变	仅链接到标记名和（或）**标记名点域**。
QuickFunction	您创建的可从其它 InTouch QuickScript 或动态链接表达调用的脚本。QuickFunction 可以是同步也可以是异步的，而所有其它脚本类型只能是同步的。
动作按钮	与您链接到“**触动链接-动作按钮**”的对象相关联。
ActiveX 事件	在运行时执行 ActiveX 控件事件。
向导	运行时访问增强功能的向导属性。

使用 InTouch QuickScript 编辑器

InTouch QuickScript 编辑器对于所有的 QuickScript 类型基本相同。因此，为避免累述，本章将介绍其公用函数和特性。特定于 QuickScript 脚本类型的项目将在本章稍后该 QuickScript 脚本类型的相应章节中介绍。



QuickScript 编辑器工具栏



您可以使用 QuickScript 编辑器工具来快速访问编辑器功能：

工具	描述
	单击以从 QuickScript 剪切选定的文本。
	单击以从 QuickScript 复制选定的文本。
	单击以将文本从剪贴板粘贴到 QuickScript。
	单击以打开标记名字典。
	单击以从当前应用程序插入窗口名。
	单击以打开 ActiveX 控件浏览器。

QuickScript 编辑器一般步骤

本章节介绍在各种 InTouch QuickScript 编辑器对话框中编写脚本时使用的一般步骤。特定于脚本类型的步骤将在本章稍后该脚本类型的相应章节中向您介绍。

QuickScript 编辑器的底部包含文本、逻辑和数学运算符按钮，您可以单击以将显示的关键字、函数或符号快速插入脚本的光标位置。

- **要缩进/取消缩进脚本中的文本：**

  将光标置于需缩进行的起始位置，然后按 TAB 键。要取消缩进，在按住 SHIFT 键的同时按下 TAB 键。

- **要创建新脚本：**

  在“**脚本**”菜单上，单击“**新建**”。

注意：应用程序、窗口脚本或触动按钮动作脚本没有“脚本”菜单。

此外，如果您通过在应用程序浏览器中双击来打开脚本进行编辑，则不允许使用新建脚本功能。要创建新脚本，关闭当前打开的脚本，然后在“特别”菜单上，指向“脚本”，选择您要创建的脚本类型。

QuickFunction、条件脚本和数据改变脚本将使用缺省名称创建。因此，在将另一脚本加载到编辑器之前，您必须保存（或按〈Esc〉取消）当前显示的脚本。

在创建 QuickScript 时，不能声明与先前定义的标记名同名的局部变量。

➢ **要从应用程序中删除脚本：**

选择您要删除的文本，然后在“**脚本**”菜单上，单击“**删除**”。该脚本即从您的应用程序被完全删除。

注意：被删除的文本不会写入 Windows 剪贴板中。

➢ **要撤消上次动作：**

在“**编辑**”菜单上，单击“**撤消**”。您上次进行的编辑操作（如粘贴），将被撤消。

- 要迅速执行此命令，右击脚本窗口，然后单击“**撤消**”。除非您执行可逆转的动作，否则“**撤消**”命令将不会被激活。

➢ **要选择整个脚本：**

在“**编辑**”菜单上，单击“**全部选定**”。整个脚本将被选定。

- 要迅速执行此命令，右击脚本窗口，然后单击“**全部选定**”。您现在即可以复制、剪切或删除整个脚本。

➢ **要从脚本中剪切选定的文本：**

选择您要删除的文本，然后在“**编辑**”菜单上，单击“**剪切**”。剪切文本将从脚本中被删除并复制到 Windows 剪贴板。您现在就可以将剪切文本粘贴到脚本中的另一位置，或者将其粘贴到另一脚本中。

- 要快速执行此命令，右击脚本窗口，然后单击“**剪切**”。除非您选定了要剪切的文本，否则“**剪切**”命令将不会被激活。

➢ **要从脚本中复制选定的文本：**

选择您要删除的文本，然后在“**编辑**”菜单上，单击“**复制**”。所复制的文本将写入到 Windows 剪贴板中。您现在就可以将复制文本粘贴到脚本中的另一位置，或者将其粘贴到另一脚本中。

- 要快速执行作此命令，右击脚本窗口，然后单击“**复制**”。除非您选定了要复制的文本，否则“**复制**”命令将不会被激活。

注意：当您剪切或复制文本时，它会自动写入 Windows 剪贴板中。在您执行剪切或复制命令之前，此信息将一直保留在剪贴板中。

➢ **要将文本粘贴到脚本中：**

在“**编辑**”菜单上，单击“**粘贴**”。Windows 剪贴板的内容将粘贴到脚本的光标位置。

- 要快速执行“**粘贴**”命令，右击脚本窗口，然后单击“**粘贴**”（如果 Windows 剪贴板中没有内容，“**粘贴**”命令将不会被激活）。

➢ **要清除脚本中的文本：**

在“**编辑**”菜单上，单击“**清除**”。脚本中的所有文本将被删除。但是，脚本并不会从应用程序中被删除。如果您在选择此命令后关闭脚本编辑器并重新打开，脚本将重新出现。如果您只想删除脚本的部分内容，可以辉亮显示要删除的文本，右击脚本窗口的空白区域，然后单击“**删除**”。

- 要完全删除脚本，您必须使用“**脚本**”菜单上的“**删除**”命令，或者选定整个脚本，然后右击脚本窗口的空白区域，单击“**删除**”。

➢ **要将函数插入脚本：**

1. 在“**插入**”菜单上，指向“**函数**”，单击函数类别名。出现相应的“**选择函数**”对话框。
2. 单击您要使用的函数。对话框关闭，函数将自动插入到脚本中的光标位置。

   您可以使用下列函数类型：

函数	描述
全部	“选择函数”对话框出现，显示所有可用函数，包括所安装的各个附加程序（配方管理器、SPC Pro 和 SQL 访问管理器或从 KBCD 安装的其它功能）的函数。 - 您也可以单击“**函数**”组中的“**全部**”按钮来访问这些函数。
字符串	“**选择函数**”对话框出现，显示所有可用字符串函数。 - 您也可以单击“**函数**”组中的“**字符串**”按钮来访问这些函数。
数学	“**选择函数**”对话框出现，显示所有可用数学函数。 - 您也可以单击“**函数**”组中的“**数学**”按钮来访问这些函数。
系统	“**选择函数**”对话框出现，显示所有可用系统函数。例如，启动和（或）激活另一应用程序、读取和（或）写入文件与磁盘信息的函数等。 - 您也可以单击“**函数**”组中的“**系统**”按钮来访问这些函数。

附件

“**选择函数**”对话框出现，显示已安装的各个附加程序（配方管理器、SPC Pro 和 SQL 访问管理器）的所有可用函数。

- 您也可以单击“**函数**”组中的“**附件**”按钮来访问这些函数。

其它

“**选择函数**”对话框出现，显示所有可用的其它函数。例如，报警、历史趋势、窗口控件、ActiveX 控件等的函数。

- 您也可以单击“**函数**”组中的“**其它**”按钮来访问这些函数。

帮助

“**选择功能来为... 获得帮助**”对话框出现，列出所有可用函数。单击任意函数以打开其相应的帮助主题。

- 您也可以单击“**函数**”组中的“**帮助**”按钮来访问这些函数。

Quick Function

“**选择函数**”对话框出现，列出所有可从当前脚本调用的 QuickFunction 名称。

- 您也可以单击“**函数**”组中的“**快速**”按钮来访问所有 QuickFunction。

有关各个脚本函数的详细信息，请参阅“脚本函数”。

➢ **要将标记名插入脚本：**

1. 在“**插入**”菜单上，单击“**标记名**”。标记浏览器将以无限选择模式显示。

   屏幕会显示通过标记浏览器访问的最后一个标记源中定义的标记名。要更改标记源，单击“**标记源**”箭头，选择表中的不同标记源。

   单击“**定义标记源**”，以在“**标记源**”列表中添加或删除标记源。

2. 双击您要使用的标记名，或选择标记名，然后单击“**确定**”。标记浏览器关闭，标记名自动插入到 QuickScript 中的光标位置。

   要快速访问标记浏览器，双击 QuickScript 窗口中的空白区域。

   要访问标记名字典中的特定标记名定义，在 QuickScript 窗口中输入标记名，然后双击它。

有关标记浏览器的详细信息，请参阅第 4 章“标记名字典”。

➢ **要将标记名点域插入脚本：**

1. 在“**插入**”菜单上，单击“**标记名**”。标记浏览器将以无限选择模式显示。

   屏幕会显示通过标记浏览器访问的最后一个标记源中定义的标记名。要更改标记源，单击“**标记源**”箭头，选择表中的不同标记源。

   单击“**定义标记源**”，以在“**标记源**”列表中添加或删除标记源。

2. 选择您要使用的标记名，然后单击“**点域**”箭头。选择列表中要使用的标记名**点域**。

3. 单击“**确定**”。所选标记名**点域**将插入到 QuickScript 中的光标位置。

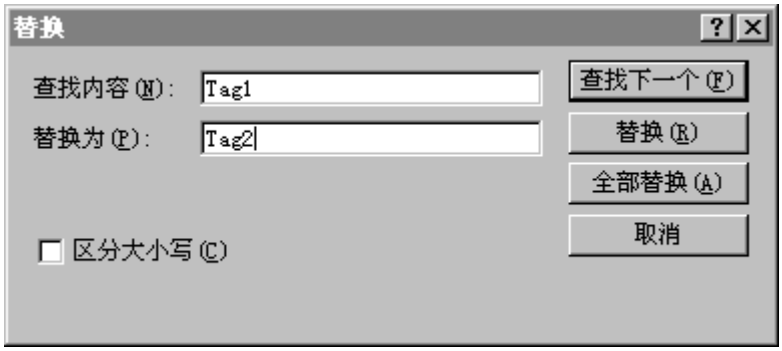
   要快速插入标记名**点域**，输入标记名，后跟句点 (.)，然后双击句点右侧。“**选择域名**”对话框出现。单击您要使用的**点域**。对话框关闭，所选**点域**将自动插入到 QuickScript 中的光标位置。

有关标记浏览器的详细信息，请参阅第 4 章“标记名字典”。

有关标记名**点域**的详细信息，请参阅您的联机《*InTouch 参考指南*》。

➢ **要查找或替换脚本中的标记名：**

1. 在“**编辑**”菜单上，单击“**查找**”。“**替换**”对话框出现：



2. 在“**查找内容**”框中，输入您要查找（或替换）的标记名，然后单击“**查找下一个**”。
3. 在“**替换为**”框中，输入用于替换旧标记名的新标记名，然后单击“**替换**”或“**全部替换**”。

   如果您只想替换旧标记名的某些实例，单击“**查找下一个**”。**InTouch** 开始搜索脚本中的旧标记名。找到的旧标记名将辉亮显示。单击“**替换**”以将其替换为新标记名，或单击“**查找下一个**”以跳过该标记名并继续搜索。如果您要替换特定标记名的所有实例，可在搜索过程中随时单击“**全部替换**”。
4. 如果您要查找标记名的特定大小写实例，请选择“**区分大小写**”选项。
5. 单击“**取消**”以关闭对话框。

➢ **要将窗口名插入脚本：**

1. 在“**插入**”菜单上，单击“**窗口**”。“**窗口名插入**”对话框出现，显示应用程序中的所有窗口名。
2. 单击您要使用的窗口名。对话框关闭，窗口名称自动插入到脚本中的光标位置。

➢ **要验证脚本：**

在编写脚本时，随时单击“**验证**”来校验脚本语法是否准确。

当您单击“**确定**”或“**保存**”时，系统将自动执行验证。如果系统在验证脚本时遇到错误，则会出现一个相应的错误消息框。

有关脚本错误的详细信息，请参阅“脚本编辑器错误消息”。

➢ **要保存脚本：**

如果您要编写多个脚本，在完成其中一个脚本后，您可以单击“**保存**”进行保存，然后在“**脚本**”菜单上，单击“**新建**”来编写另一新脚本。

注意：“应用程序”和“窗口”脚本不支持此功能。否则，当单击“确定”按钮时，会自动执行保存功能。

- 如果系统在保存脚本时遇到错误，则会出现一个相应的错误消息框。

➢ **要恢复脚本：**

如果您在更改脚本后决定清除所做的更改并恢复原始脚本，则可以单击“**恢复**”。

注意：一旦保存，您将不能再恢复脚本。“应用程序”和“窗口”脚本不支持此功能。

➢ **要退出脚本编辑器：**

在“**脚本**”菜单上，单击“**退出**”。脚本编辑器将关闭，除非遇到错误，否则将保存脚本。在“**应用程序脚本编辑器**”中，指向“**文件**”，然后单击“**退出**”。

- 在写完脚本后，您也可以单击“**确定**”来关闭脚本编辑器。

注意：当您选择“退出”、“确定”或单击对话框右上角的“X”按钮时，系统会自动验证脚本的正确性。

有关脚本错误的详细信息，请参阅“脚本编辑器错误消息”。

➢ **要指定脚本的执行频率：**

在“**当运行期间/在显示期间/在向下期间每 0 毫秒**”框中，输入在脚本执行前等待的毫秒数。

- 当您创建应用程序的“**在运行期间**”脚本、窗口的“**在显示期间**”脚本、条件的“**为真/假期间**”脚本，或者键和触动按钮动作的“**按下期间**”脚本时，您必须指定各自执行的频率（以毫秒为单位）。

注意：WindowViewer 会按您指定的时间，尽快运行这些脚本类型。但是，系统并不提供性能保证。此外，脚本的运行速度不会快于您在配置 WindowViewer 属性时指定的“计时间隔”设置。

在 Windows NT 操作系统中，脚本的执行频率不能快于 10 毫秒；在 Windows 2000（或以上版本）操作系统中不能快于 50 毫秒。

有关计时间隔设置的详细信息，请参阅第 2 章“使用 WindowMaker”。

应用程序脚本

应用程序脚本与整个应用程序相链接。您可以使用应用程序脚本来启动另外的应用程序、创建过程模拟以及计算变量等。您可以在应用程序中使用下面三种应用程序脚本类型：

启动时 在应用程序最初启动时执行一次。

在运行期间 在应用程序正在运行时，按指定的频率连续执行。

关闭时 在应用程序退出时执行一次。

注意：应用程序“启动时”QuickScript 在打开任何窗口或执行任何运行初始化之前执行。因此，您不能在应用程序“启动时”QuickScript 中引用 ActiveX 方法、属性或事件。

同样，应用程序“启动时”QuickScript 在 I/O 通讯初始化之前执行。因此，您不能在应用程序“启动时”QuickScript 中使用 I/O 型标记名或远程标记名引用。此外，I/O 型标记名和远程标记名引用不会在应用程序“关闭时”QuickScript 中更新。

数据改变或脚本 QuickScript 不可以从应用程序“关闭时”QuickScript 中执行。

此外，您不能使用应用程序“关闭时”QuickScript 来启动其它应用程序。

➢ **要访问应用程序脚本编辑器：**

在“**特别**”菜单上，指向“**脚本**”，然后单击“**应用程序脚本**”，或者在应用程序浏览器的“**脚本**”下，双击“**应用程序**”。“**应用程序脚本**”编辑器出现：

在应用程序浏览器的“**脚本**”下，您也可以右击“**应用程序**”，然后单击“**打开**”。



当您选择“**在运行期间**”脚本时，“**每 0 毫秒**”框将被激活。在框中，输入脚本执行前等待的毫秒数。如果您要立即执行脚本，则可以创建相同的“**启动时**”脚本。但是，只要“**在运行期间**”脚本的条件或事件相符，脚本就会以指定的频率重复执行。

窗口脚本

窗口脚本与特定窗口相链接。您可以在窗口中使用下列三种脚本类型：

显示时 在窗口最初显示时执行一次。

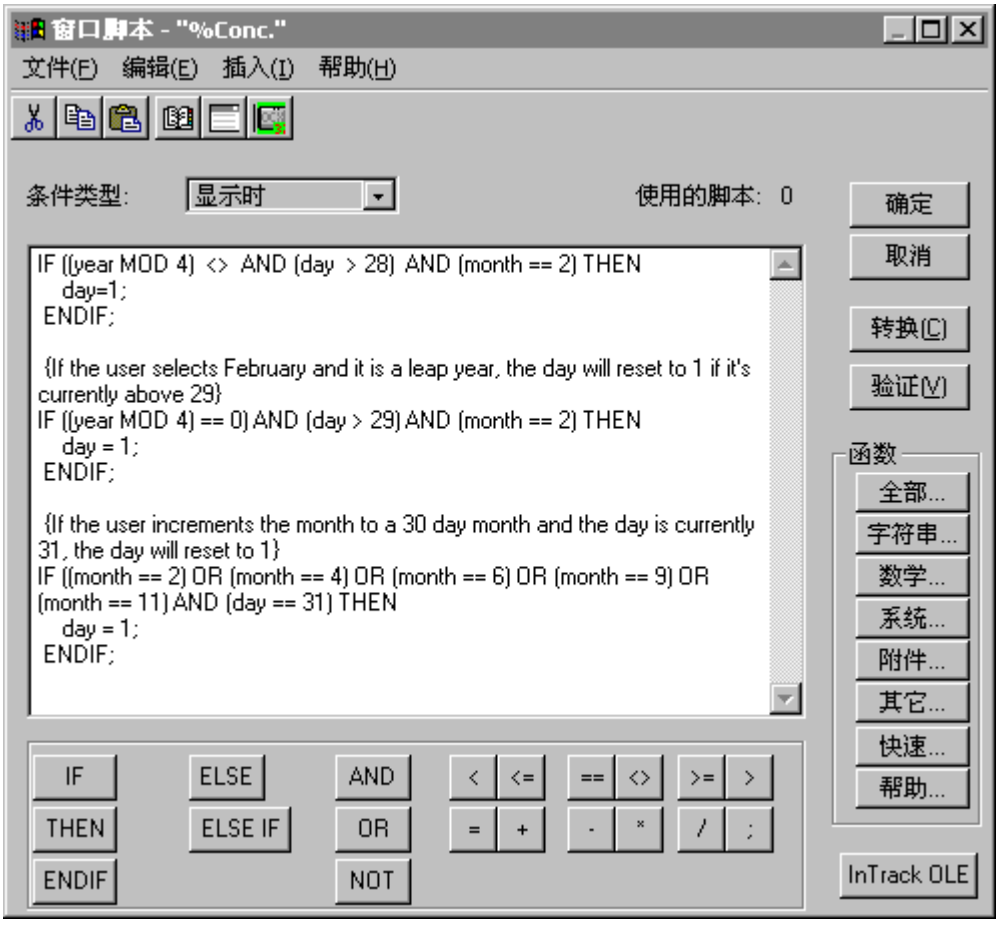
在显示期间 在窗口显示期间，按指定的频率连续执行。

隐藏时 在窗口隐藏时执行一次。

➢ **要创建窗口脚本：**

在“**特别**”菜单上，指向“**脚本**”，然后单击“**窗口脚本**”。“**窗口脚本**”编辑器出现：

- 要快速访问特定窗口的脚本编辑器，在应用程序浏览器的“**窗口**”下，右击窗口名，然后单击“**窗口脚本**”。您也可以右击打开窗口的空白区域，然后单击“**窗口脚本**”。如果选定的窗口存在脚本，则会显示该脚本。



当您选择“**在显示期间**”时，“**每 0 毫秒**”框将被激活。在框中，输入脚本执行前等待的毫秒数。如果您要立即执行脚本，请创建一个相同的“**显示时**”脚本。但是，只要“**在显示期间**”脚本的条件或事件相符，脚本就会以指定的频率重复执行。

注意：如果您对活动窗口应用窗口脚本，然后创建一个新窗口，则活动窗口的脚本可以复制到新窗口。一个消息对话框会出现，询问您是否要复制脚本。

键脚本

键脚本与键盘上的特定键或键组合相链接。您可以使用键脚本来为应用程序创建全局键。例如，返回到主菜单窗口、注销操作员等。您可以对键应用三种键脚本类型：

按下时 键刚按下时执行一次。

按下期间 在按住键的过程中，按指定的频率连续执行。

放开时 键释放时执行一次。

➢ **要访问键脚本编辑器：**

在“**特别**”菜单上，指向“**脚本**”，然后单击“**键脚本**”，或者在应用程序浏览器的“**脚本**”下，双击“**键**”。“**键脚本**”编辑器出现：

- 在应用程序浏览器的“**脚本**”下，您也可以右击“**键**”，然后单击“**打开**”。



当您选择“**按下期间**”时，“**每 0 毫秒**”框将被激活。在框中，输入脚本执行前等待的毫秒数。如果您要立即执行脚本，请创建一个相同的“**按下时**”脚本。但是，只要“**在按下期间**”脚本的条件或事件相符， 脚本就会以指定的频率重复执行。

有关对脚本指定键的详细信息，请参阅“指定脚本的等价键”。

注意：触动按钮动作脚本的本地活动窗口中使用的等价键优先于具有相同等价键的全局键脚本。

在 Windows 2000 中，“放开时”脚本中必须写入包含 ALT 键的所有 SendKey QuickFunction。“放开时”脚本不会处理 ALT 键。

触动按钮动作脚本

触动按钮动作脚本类似于键脚本，不同之处在于它与一个链接到**触动链接 - 动作按钮**的对象相关联（脚本编辑器通过动画链接选择对话框访问）。当操作员单击或者按下指定给链接的对象或按钮时，将执行触动按钮动作脚本。您可以对一个对象使用三种触动按钮动作脚本类型：

按下时 键刚按下时执行一次。

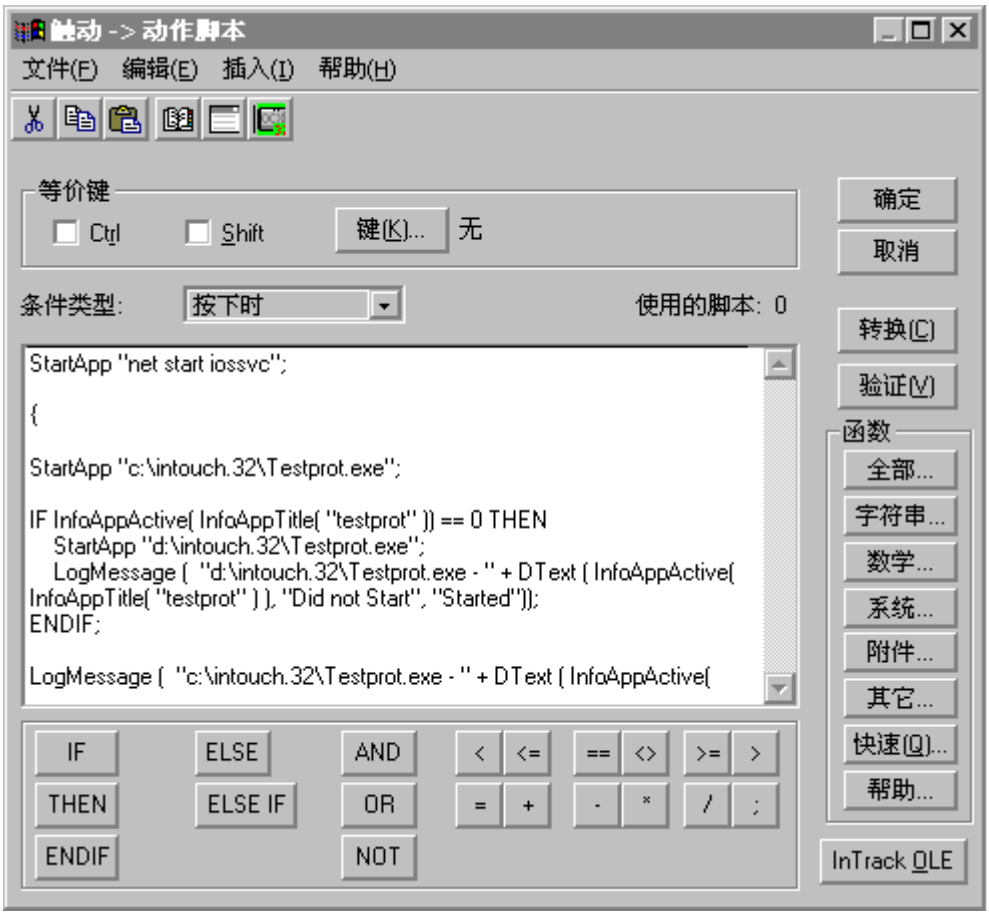
按下期间 按住键的过程中，按指定的频率连续执行。

放开时 键释放时执行一次。

➢ **要创建动作按钮脚本：**

1. 画一个您要链接到脚本的对象或按钮。
2. 双击对象，或者选定对象，然后在“**特别**”菜单上，单击“**动画链接**”。动画链接选择对话框出现。

   要快速访问对话框，右击对象，然后单击“**动态链接**”。
3. 在“**触动按钮**”区域中，单击“**动作**”。“**触动 -> 动作脚本**”编辑器出现：



当您选择“**按下期间**”时，“**每 0 毫秒**”框会被激活。在框中，输入脚本执行前等待的毫秒数。如果您要立即执行脚本，请创建一个相同的“**按下时**”脚本。但是，只要“**按下期间**”脚本的条件或事件相符，脚本将以指定的频率重复执行。

有关对脚本指定键的详细信息，请参阅“指定脚本的等价键”。

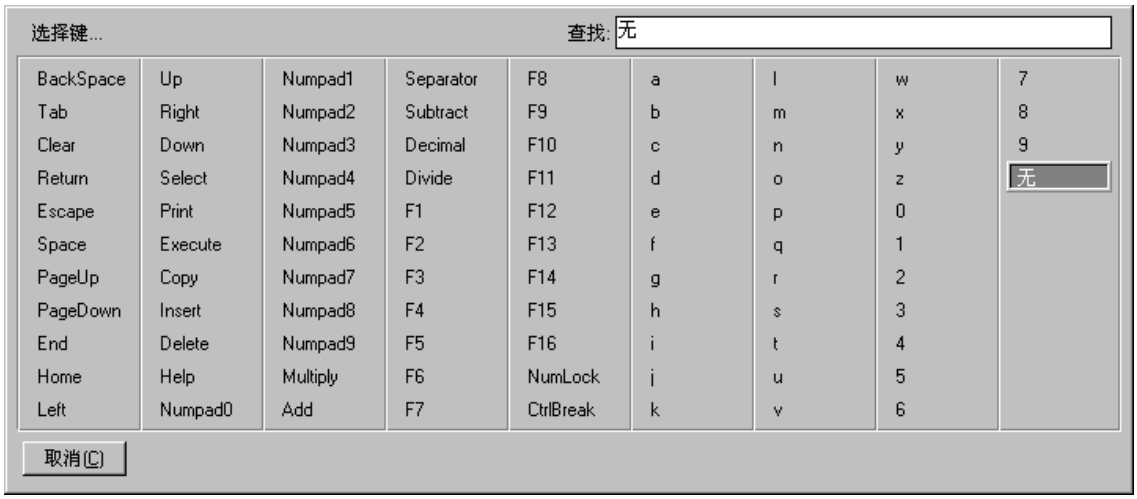
注意：触动按钮动作脚本的本地活动窗口中使用的等价键优先于具有相同等价键的全局键脚本。此外，仅当对象窗口处于活动状态时，等价键才会被激活。

指定脚本的等价键

键脚本和触动动作脚本编辑器与其它 QuickScript 编辑器略有不同。因为所创建的脚本将应用于键，所以您还必须指定操作员需按下以执行脚本的键。

➢ **要对键脚本指定键：**

1. 如果您需要操作员在按下键来执行脚本的同时还必须按住 CTRL 和（或）SHIFT 键，请选择“**Ctrl**”和（或）“**Shift**”。
2. 单击“**键**”以选择您要指定给脚本的键。“**选择键**”对话框出现：



3. 单击所需的键。对话框自动关闭，您选择的键会自动显示在“**键**”框中。

条件脚本

条件脚本与离散标记名或一个求值为真或假的表达式相链接。您也可以使用包含模拟标记名（见下例）的离散表达式。您可以对条件应用四种脚本类型：

为真时 当条件变为真时执行一次。

为假时 当条件变为假时执行一次。

为真期间 当条件为真时连续执行。

为假期间 当条件为假时连续执行。

➢ **要访问条件脚本编辑器：**

1. 在“**特别**”菜单上，指向“**脚本**”，然后单击“**条件脚本**”，或者在应用程序浏览器的“**脚本**”下，双击“**条件**”。“**条件脚本**”编辑器出现：

   ☜ 在应用程序浏览器的“**脚本**”下，您也可以右击“**条件**”，然后单击“**打开**”。



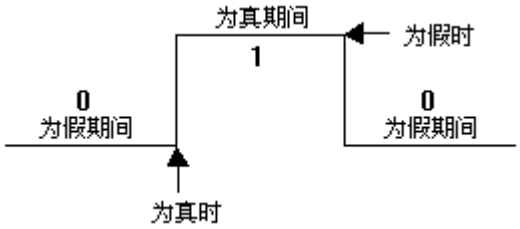
2. 由于条件脚本根据所满足的条件来执行，因此您必须在“**条件**”框中指定条件（离散标记名或表达式）。

注意：每个条件只能应用一个脚本。

- 您也可以使用对模拟标记名求值为真或假的离散表达式。例如 TankLevel >= 75。在本例中，当标记名“TankLevel”的值大于或等于“75”时，脚本将执行。

  仅当条件**变为**真或假时，脚本才会执行。例如，如果 WindowViewer 启动时的初始值为真，则只有在该值变为假后再变为真，才会执行“**为真时**”脚本。

  您可以对相同的条件应用所有四种脚本类型。在经过指定毫秒数之后，“**为真期间**”和“**为假期间**”脚本将开始执行。要立即执行脚本，请创建一个相同的“**为真时**”或“**为假时**”脚本。例如：



3. 在“注释”框中，输入有关脚本文件的任何其它注释。

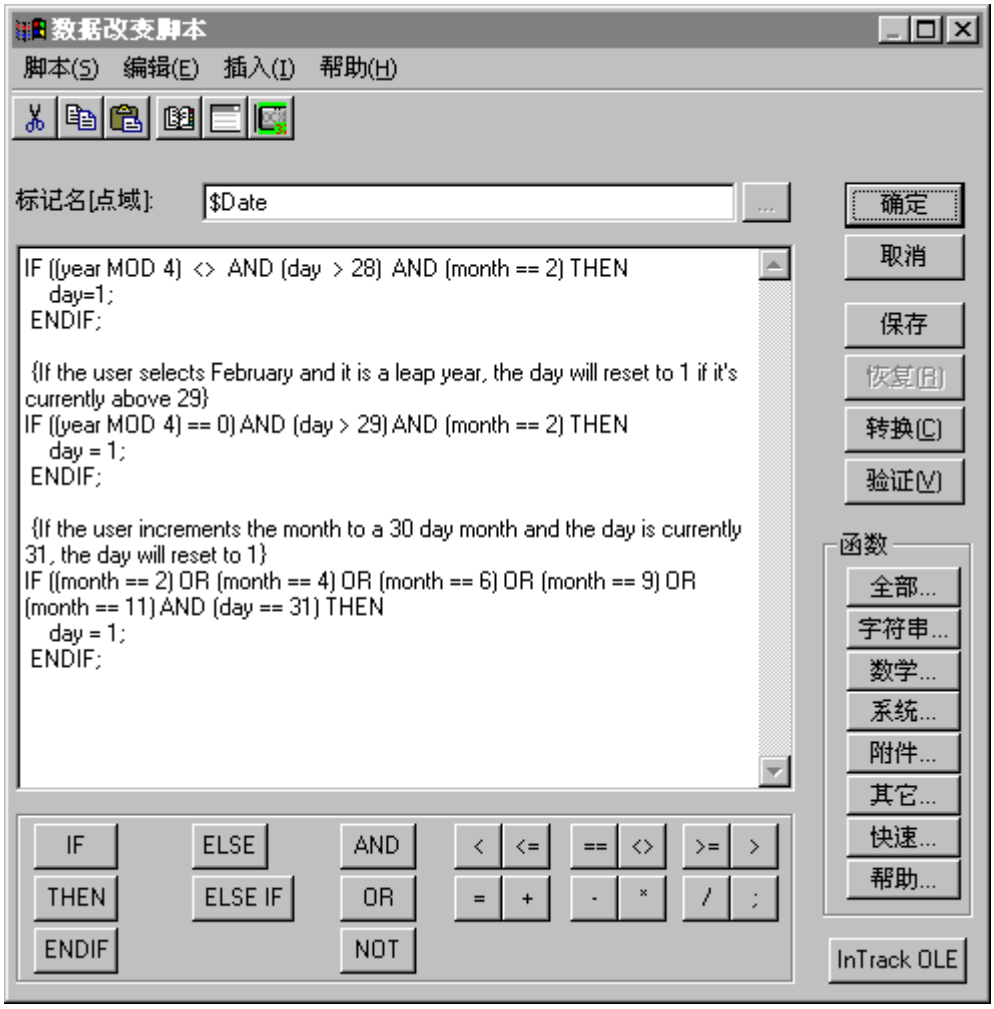
数据改变脚本

数据改变脚本仅与标记名或**标记名点域**相链接。当标记名或**标记名点域**的值变化大于您为标记名字典中的标记名定义的死区值时，将执行数据改变脚本一次。

➢ **要访问数据改变脚本编辑器：**

1. 在“**特别**”菜单上，指向“**脚本**”，然后单击“**数据改变脚本**”，或者在应用程序浏览器的“**脚本**”下，双击“**数据改变**”。“**数据改变脚本**”编辑器出现：

   在应用程序浏览器的“**脚本**”下，您也可以右击 4“**数据**”，然后单击“**打开**”。



2. 因为数据改变脚本是根据数据值的变化来执行的，所以您必须在“**标记名[点域]**”框中指定一个标记名或**标记名点域**。

3. 在“**插入**”菜单上，单击“**标记名**”，或者双击脚本窗口。“**选择标记**”对话框出现。

   要选择一个不含**点域**的标记名，双击该标记名，或者选择标记名，然后单击“**确定**”。所选标记名将插入到脚本中的光标位置。

   - 要快速访问“**选择标记**”对话框，双击脚本编辑窗口内的空白区域。要访问标记名字典中的特定标记名定义，输入标记名，然后双击它。

   要选择**点域**，首先选择您要使用的标记名，然后单击“**点域**”箭头，选择您要与所选标记名关联的**点域**。单击“**确定**”。所选标记名**点域**将插入到脚本中的光标位置。

   - 要快速插入标记名**点域**，输入标记名，后跟句点 (.)，然后双击句点右侧。“**选择域名**”对话框出现。单击您要使用的**点域**。对话框关闭，所选**点域**将自动插入到脚本中的光标位置。

     有关标记名**点域**的详细信息，请参阅您的联机《*InTouch 参考指南*》。

   有关标记浏览器的详细信息，请参阅第 4 章“标记名字典”。

警告：“条件脚本”或“数据改变脚本”中修改（或写入）的标记名不得用作“数据改变脚本”或“条件脚本”表达式的标记名。

例如：Script1 调用 Script2，Script2 调用 Script3，而 Script3 试图调用 Script1（仍处于活动状态）。此时，最后的调用将不会执行。为了避免出现这种情况，请勿使用条件脚本或数据改变脚本中修改过的标记名作为数据改变脚本或条件脚本表达式的标记名。

ActiveX 事件脚本

大多数的 ActiveX 控制都有与之关联的事件。例如单击、双击、按下鼠标和按键都是许多 ActiveX 控件中使用的典型事件。InTouch 提供 ActiveX 事件脚本来支持事件动作。您可以将 ActiveX 事件脚本与每个事件相关联。ActiveX 控件事件在运行时 (WindowViewer) 执行 。

➢ **要访问 ActiveX 事件脚本编辑器：**

1. 单击 ActiveX 控件“**属性**”对话框中的“**事件**”标签，激活“**事件**”属性页。例如：



2. 双击“**脚本**”列中的空白单元格，或者输入 ActiveX 事件脚本名并单击“**确定**”。

3. 如果当前不存在所输入的 ActiveX 事件脚本名，会出现一个消息框，询问您是否要现在创建。单击“**确定**”。“**ActiveX 事件脚本**”编辑器出现：



4. 在“**名称**”框中，输入用于识别 ActiveX 事件脚本的名称。

5. ActiveX 控件方法类似于 ActiveX 控件属性。属性是与对象关联的数据成员，而方法是控制对象的数据函数。要访问 ActiveX 控件方法，在“**插入**”菜单上，单击“**ActiveX**”。“**ActiveX 控件浏览器**”出现：



“**ActiveX 控件浏览器**”会显示应用程序中使用的所有 ActiveX 控件的名称。当您选择控件名时，将显示其相应的方法。选择您要插入脚本的方法，然后单击“**确定**”。

QuickFunction

QuickFunction 指可以从其它脚本或表达式编写及调用的脚本。它们存储于创建它的应用程序中。从其它脚本或表达式调用 QuickFunction 允许您一次性建立脚本，然后再多次重复使用它。通过重复使用这些脚本，可以减少脚本中来回复制的重复代码数量，从而减少应用程序的维护工作。系统将可重复使用的代码存储在一个脚本和一个位置中，从而支持一次编辑更新所有脚本。

QuickFunction 可定义为异步函数，因此当执行时在它们将在主要 WindowViewer（运行时）过程的后台中运行。此功能使得 WindowViewer 可以主程序流中将 SQL 数据库调用等耗时性操作分开。当需要执行这些耗时性操作时，您可以通过创建一个异步 QuickFunction，使得在执行这些操作的同时，仍可以使所有动态链接和其它 InTouch 功能保持活动状态。

例如，假定您需要执行 FOR NEXT 循环计算。通过使用异步 QuickScript，您就可以在计算的同时而又不停止 InTouch 系统的执行。.您可以使用 **IsAnyAsyncFunctionBusy()** 函数来确定计算何时完成并向操作员显示计算结果。

```
If IsAnyAsyncFunctionBusy(120) ==1 then
Show "Calculation running Window";
Endif;
```

注意：我们不建议您在异步 QuickScript 中使用 SQL 命令，因为任何 SQL 命令的失败都可能导致系统不稳定。

有关异步脚本的详细信息，请参阅“异步 QuickFunction 脚本”。

- 动态链接表达式框限制为 255 个字符。但是，您可以创建一个更复杂的 QuickFunction，然后从动画链接表达式框中调用它。这允许您使用 **CALL** 语句来调用一个包含 **RETURN** 语句的复杂脚本，然后将结果返回给表
  达式。

  例如，当加入各为 30 字符长的多个标记名（使用 "**&**" 和 "**:**"）时，您只能在表达式中使用 8 个标记名（加上空格）。但是，通过在表达式中使用语句 **CALL MYSCRIPT()**，您就可以执行包含数百个 30 字符长的标记名的 QuickFunction。此 QuickFunction 使用 RETURN 语句来提供一个返回表达式的值。

注意：您必须使用“触发器”标记名作为 QuickFunction 的参数，以强制更新动画链接。例如，使用标记名 $Second 作为 QuickFunction 的参数可以在每次 $Second 改变值时对动画链接表达式求值，从而每隔一秒调用 QuickFunction 一次。

QuickFunction RETURN 语句不能用于返回消息或字符串型数值。

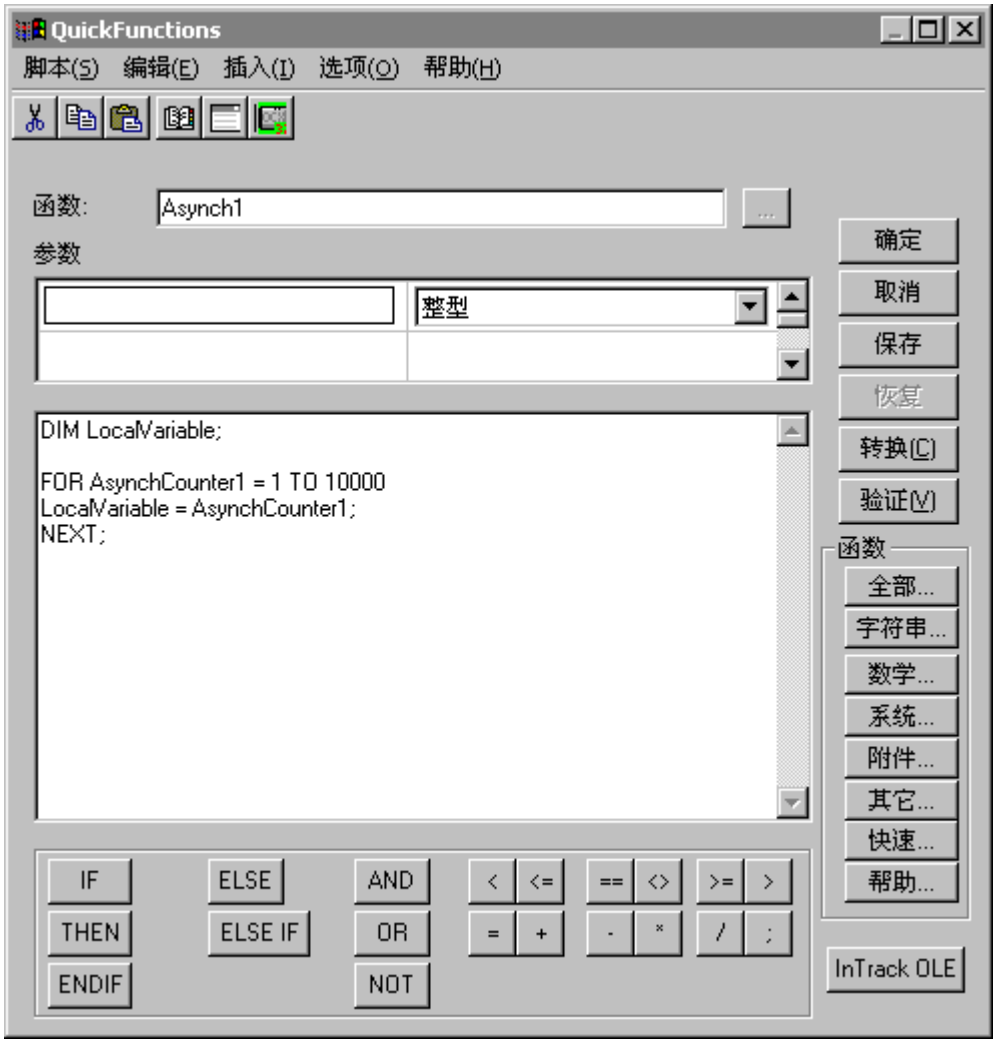
一旦创建并保存 QuickFunction，您就可以立即从任何其它脚本或表达式调用其名称。

有关 QuickFunction 语法的详细信息，请参阅您的联机 *《InTouch 参考指南》*。

➢ **要创建 QuickFunction：**

1. 在“**特别**”菜单上，指向“**脚本**”，然后单击“**QuickFunctions**”，或在应用程序浏览器的“**脚本**”下，双击“**QuickFunctions**”。
“**QuickFunctions**”脚本编辑器出现：

   - 在应用程序浏览器的“**脚本**”下，您也可以右击“**QuickFunctions**”，然后单击“**打开**”。



2. 在“**函数**”框中，输入 QuickFunction 的名称。

   - QuickFunction 名称最长可为 31 个字符（不能使用空格和重复名称）。这是其它 QuickScript 或表达式用于调用 QuickFunction 的名称。如果您在 QuickScript 编辑器中单击“**全部**”或“**快速**”按钮，或者在“**插入**”菜单的“**函数**”下，单击“**全部**”或“**QuickFunctions**”命令，此名称也将出现在“**选择函数**”对话框中。

3. 在“**参数**”框中，输入 QuickScript 的每个参数名，然后单击参数箭头，从列表中选择其数据类型。

有效数据类型如下所示：

数据类型	描述
整型	用于传递整数变量、标记名或常数值。
实型	用于传递实数变量、标记名或常数值。
离散型	用于传递离散变量、标记名或常数值。
消息型	用于传递字符串变量、标记名或常数值。

下面是保留的关键字，不得用作参数名：

Return Call Dim As RetVal

Integer Real Discrete Message

参数名是局部变量，只能存在于其所定义的 QuickFunction 内。每个 QuickFunction 最多可使用 16 个参数。参数名最长可为 31 个字符，并且不能使用空格。此外，参数名还必须以希腊字符 (A-Z) 开头。不能使用重复的名称。

不能使用标记名作为参数名。标记名优先于同名的参数名，并且 QuickScript 将不能正确地执行。参数名被视为局部变量，因而它不占用标记名计数。

5. 一旦您输入参数名并选定其类型，就可以开始编写 QuickFunction 了。

QuickFunction 参数表达式

脚本参数通过数值传递。参数表达式可以是返回整型、实型、离散型、消息型数据值的任何脚本表达式。所有的参数表达式值在执行 QuickFunction 之前，通过调用脚本进行处理。例如：

```
CALL Stuff (5.6, 237, "PI");
```

在此表达式中，实常数 5.6 作为参数 1，整常数 237 作为参数 2，消息常数“PI”作为参数 3。

```
CALL Temp (IntegerTag);
```

IntegerTag 作为参数表达式值传递。

```
CALL ValveOpen (Tag.MaxEU -5);
```

计算所得值 (Tag.MaxEU -5) 作为参数表达式值传递。

参数数据类型匹配

调用语句参数表中的数据类型与被调用的已保存 QuickFunction 数据类型之间必须保持严格的左右对应关系。参数的数目也必须与已保存的 QuickFunction 参数表中的参数数目完全相同。系统在将实型值转换成整型以及将整型值转换成实型时使用强制方法。这就使得您可以修改值类型，将任意模拟参数传递给任何其它模拟类型。

例如，如果您将实型值 1.23 传递给一个整型参数，它将只使用 1 而丢失 .23。同样，如果您将整型值 1 传递给一个实型参数，它会将 1 改为 1.0。但是，尽管使用了此强制方法，我们仍建议您使调用参数类型与相应的 QuickFunction 参数表严格匹配。

有效的 QuickFunction 语法

QuickFunction 可返回值。QuickFunction 语句的语法和形式如下所示：

CALL*QuickFunctionName ([arg1, ... arg16]);*

此处：

CALL	是所有 QuickScript 和表达式调用 QuickFunction 所需的关键字。
QuickFunctionName	1 到 31 个字符的字符串，对应于指定给已保存的 QuickFunction 的名称 。
([arg1, ...arg16])	以括号括起的 0 到 16 逗点分隔参数表达式。

有关 QuickFunction 语法的详细信息，请参阅您的联机 *《InTouch 参考指南》*。

使用 Return 语句

QuickFunction 是一个可从另一脚本（调用脚本）调用的脚本。下面是一个调用 QuickFunction 的脚本实例：

```
RETURNRESULT = CALL MyFunction( StartHour, EndHour);
```

QuickFunction 使用 RETURN 语句来强制将值“返回”给调用脚本。当遇到 **RETURN** 时，QuickFunction 便立即结束执行。在此时，QuickFunction 会返回一个值给调用脚本。返回的数据类型可以是离散型、整型或实型。

在上面的例子中，标记名 **RETURNRESULT** 必须是一个离散型、整型或实型标记名来接收传递值。当遇到 **RETURN** 语句时，QuickFunction 将不会继续执行。

在下面的例子中，如果标记名 ConvertType 等于 0，则程序将计算表达式 abs((Max / Maxprogress) * 100) 并将此实型数据结果返回给调用脚本。

```
IF ConvertType == 0 THEN
   RETURN abs( (Max / Maxprogress) * 100 );
ELSE
   RETURN 0;
ENDIF;
```

返回值的数据类型由上下文决定，例如：

```
RETURN AnalogTag;
```

如果标记名 **AnalogTag** 定义为内存整型，**RETURN** 语句会返回一个模拟值到调用位置，但只可以返回一个值。

异步 QuickFunction 脚本

您可以将 QuickFunction 定义为异步（QuickFunctions 是唯一支持异步功能的 InTouch 脚本类型）。当 WindowViewer 遇到异步 QuickFunction 调用时，便会产生一个独立的线程。一旦产生新线程，WindowViewer 便可以自由地继续调用其它脚本（包括其它异步 QuickFunction）、等待异步脚本结束，或进行图形窗口更新。异步 QuickFunction 将从该新产生的线程实际启动。一旦异步脚本完成执行，新产生的线程便告结束。线程机制对操作员是透明的。

注意：异步 QuickFunction 不能返回值。因此，异步 QuickFunctions 不能用于动画链接表达式中。此外，您可以同时执行的异步 QuickFunction 数目没有限制。但是，为确保足够的系统性能，我们强烈推荐您不要同时执行三个以上的异步 QuickFunction。此外，您不能同时运行同一个 QuickFunction 的一个以上的实例。

➢ **要创建异步 QuickFunction：**

1. 创建一个 QuickFunction。
2. 在“**选项**”菜单上，单击“**异步的**”。例如：



注意：在运行时，一旦异步 QuickFunction 开始执行，便不能停止。但是，如果操作员停止运行所有脚本（通过“逻辑”菜单上的“停止逻辑”命令，或者按下链接到 $LogicRunning 系统标记名的按钮），则不会开始执行新的异步 QuickFunction。

控制异步脚本

您可以使用 **IsAnyAsyncFunctionBusy()** 函数来确认是否有异步 QuickFunction 正在运行。通过使用此函数，您可以使调用异步 QuickFunction 的 QuickScript 等待所有其它异步 QuickFunction 完成处理。这就允许 QuickScript 进行自我重新同步。

此函数的有效语法是：

```
DiscreteTag = IsAnyAsyncFunctionBusy(timeout);
```

此处：

DiscreteTag	是一个离散类型标记名，其返回的值如下所示： 如果函数超时，则等候所有执行中的 QuickFunction 完成，并返回值 1（真）给 ***DiscreteTag***。 如果没有异步 QuickFunction 运行，则立即返回值 0（假），否则 QuickFunction 将等待超时值到达。如果超时后没有异步 QuickFunction 运行，则也会返回值 0。
Timeout	是一个表示等待秒数的整型值，用于确定是否有任何异步 QuickFunctions 正在运行。

例如，假定您要使用异步 QuickFunction 来连接若干个 SQL 数据库，并且您知道需要持续两分钟才能建立连接。首先，您需要执行一个异步 QuickFunction 来连接到 SQL 数据库。下一步，您需要启动函数 **IsAnyAsynchFunctionBusy(120)**，以便在完成 QuickFunction 之前 SQL 有足够的时间来建立连接。

但是，如果在 2 分钟以后仍没有建立连接，并且异步 QuickFunctions 仍在忙于尝度连接，则 **IsAnyAsynchFunctionBusy()** 函数会返回一个值 1（真）。您现在就可以显示一则错误信息，告诉操作员 SQL 连接失败。

例如，您可以使用下面的窗口“**显示时**”QuickScript：

```
IF IsAnyAsyncFunctionBusy(120) == 1 THEN
   SHOW "SQL Connection Error Dialog";
ENDIF;
```

使用局部变量

您可以在脚本内声明局部变量来存储临时结果，并用临时脚本值来创建复杂的计算，如此既不影响或减少授权标记名计数，又可提高系统性能。

局部变量或标记名可在同一脚本内交替使用。但是，一旦脚本结束，局部变量便会释放其值及含义，但标记名因为是全局的，因而仍可以保留其值。与标记名不同，局部变量只能在脚本正文内声明。指定脚本正文内可以声明的局部脚本变量数目仅受可用内存的限制。一旦声明了局部变量，您就可以在相同脚本正文内的一个或多个表达式中使用它。在脚本正文内添加局部变量名的表达式和语法规则与标记名相同，唯一的例外是局部变量不支持**点域**引用。

与标记名一样，局部变量既可以用在语句和表达式的左边，也可以用在语句和表达式的右边，这些语句和表达式可以包括不同数据类型的其它局部变量和标记名。

注意：不能定义与现有标记名同名的局部变量；不能创建与已定义的局部变量同名的标记名。如果发生同名现象，局部变量将优先于标记名，而标记名值会反映对局部变量的任何更改。

有效的局部变量语法

每个局部变量均必须在脚本内用一个单独的 **DIM** 语句声明（一行一个变量，并且不允许层迭）。**DIM** 语句的语法和格式如下所示：

```
DIM LocalVarName [ AS data-type ];
```

此处：

DIM	是必不可少的关键字。
LocalVarName	是符合标记名格式和限制的变量名。变量名最长可为 32 个字符，并且必须必须以 A-Z 或 a-z 开头，其余字符可以是：A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ 和 &。 如果脚本内的已声明变量名和标记名之间发生抵触（二者同名），变量名将优先于标记名。 例如，假定数据库中有一个标记名定义为“Temp”，并且您声明“DIM Temp AS Integer；”。在声明的脚本内，语句中使用“Temp”的表达式将引用与局部变量“Temp”，而非标记名“Temp”相关联的值。

AS 是一个可选关键字。

如果您在 DIM 语句中省略 AS 子句，则缺省条件下，变量声明为整型数据类型。例如：

```
DIM LocVar1;
```

等同于：

```
DIM LocVar1 AS Integer;
```

data-type 可以是下面的四个关键字之一：

Integer
```
DIM LocVar1 AS Integer;
```

Real
```
DIM LocVar2 AS Real;
```

Discrete
```
DIM LocVar3 AS Discrete;
```

Message
```
DIM LocVar4 AS Message;
```

InTouch **DIM** 语句不能层迭。例如，下面的例子是无效的，您不能使用：

```
DIM LocVar1 AS Integer, LocVar2 AS Real;
DIM LocVar3, LocVar4, LocVar5, AS Message;
```

要在 InTouch 中声明多个变量，您必须为每个变量输入一个单独的 DIM 语句。例如，下面的例子是有效的：

```
DIM LocVar1 AS Integer;
DIM LocVar2 AS Real;
```

注意：数据类型关键字不区分大小写。

DIM 语句行必须用分号（;）结束。

层迭 DIM 语句不受支持。

DIM 语句可以位于脚本正文内的任何地方，但是必须在第一个引用它的脚本语句或表达式之前。

如果局部变量在 DIM 语句之前被引用，则在验证脚本时脚本编辑器会将其读作标记名，并要求您进行定义。

有关局部变量语法的详细信息，请参阅您的联机*《InTouch 参考指南》*。

创建 FOR-NEXT 循环脚本

FOR-NEXT 循环用于在脚本的单一执行过程中多次执行脚本内的函数（或函数集）。FOR-NEXT 循环的一般格式如下所示：

```
FOR AnalogTag = start_expression TO end_expression [STEP
change_expression]
  ...statements...
IF (condition) THEN
   [EXIT FOR;]
   ENDIF;
   ...statements...
NEXT;
```

此处：

[ ]	括号内的项目为可选参数。
BOLDCASE	**大写粗体**项目表示脚本语言的保留关键字。
italics	小写*斜体*项目表示变量数据。
AnalogTag	InTouch 模拟型标记名。
start_expression	有效的 InTouch 表达式，通过初始化 *AnalogTag* 值来执行循环。
end_expression	有效的 InTouch 表达式。如果 *AnalogTag* 大于 *end_expression*，则脚本跳至 **NEXT** 语句之后的第一个语句执行（这仅适用于递增循环的情形，如果是递减循环，则当 *IntegerTag* 小于 *end_expression* 时，循环将中止）。
change_expression	有效的 InTouch 表达式，用于定义当执行 **NEXT** 语句后 *AnalogTag* 的增量或减量值。

注意：*Change_expression* 可以是正数也可以是负数。如果 *change_expression* 为正，则 *start_expression* 必须小于或等于 *end_expression*，否则将不会执行循环中的语句；如果 c*hange_expression* 为负，则 *start_expression* 必须大于或等于 *end_expression*，以执行循环正文。如果没有设置 **STEP**，则 *change_expression* 缺省为 1。

...statements...	一个或多个有效 InTouch 脚本语言语句，可以是嵌套 **FOR** 循 环。嵌套循环要求使用不同于外部循环的 *change_expressions*。
FOR	标志“For”循环的开始。
TO	标志 *end_expression* 的开始。
STEP	标志 *change_expression* 的开始
EXIT FOR	立刻终止循环，脚本同时跳到 **NEXT** 语句后的第一个语句执行。
NEXT	标志循环语句的结束。

当您执行 FOR...LOOP 函数时，InTouch 会：

1. 设置 *Analog* 等于 *start_expression*。
2. 检验 *Analog* 是否大于 *end_expression*。如果大于，InTouch 将退出循环（如果 c*hange_expression* 为负，InTouch 将检验 *Analog* 是否小于 *end_expression*）。
3. 执行语句。
4. 增大 *AnalogTag*，增量为 1 或指定的 *change_expression*。
5. 重复步骤 2 到 4。

嵌套 FOR-NEXT 循环

FOR-NEXT 循环可以嵌套。可能的嵌套层数由系统内存和可用资源决定。

屏幕更新和性能影响

在执行 FOR-NEXT 循环的过程中，屏幕更新子系统会暂停，直到循环结束。

- 在执行 FOR-NEXT 循环的过程中，屏幕上的所有动画均会暂停。因此，您不能使用 FOR-NEXT 循环来设置对象的屏幕运动效果，因为所有的移动只有在循环完成后才会发生。
- 实时趋势会暂停。
- 会暂停“正在更新”的历史趋势。
- 在循环执行过程中，屏幕上的显示值不会更新。虽然这些变量取得了新值，但只有在循环完成后才会显示这些新值。
- FOR-NEXT 循环正文内修改过的 I/O 型标记名值只有在循环执行结束后才会传递给 I/O 服务器。因此，如果您在每次 FOR-NEXT 循环相互作用期间修改 I/O 型标记名的值，则只有该 I/O 型标记名的最终值会传递给 PLC。

注意：FOR-NEXT 循环会暂停 InTouch 中的所有操作。在执行期间，WindowViewer 中没有数据移动、没有动画链接更新，也没有其它脚本执行，包括异步 QuickFunction。但是，异步 QuickFunction 中使用的 FOR-NEXT 循环不能暂停其它运算。

循环执行时间限制

缺省条件下，FOR-NEXT 循环必须在 5 秒钟内完成执行。这是一个内置于 FOR-NEXT 子系统的安全限制。所有 FOR-NEXT 循环均强制使用时间限制。例如：

```
FOR X = 1 TO 1000000
   FileWriteMessage("C:\LOG.TXT","Hello");
NEXT;
```

注意：您可以通过在应用程序目录的 INTOUCH.INI 文件中添加下列开关来扩展此限制。

```
LoopTimeout=20
```

此处，20 是提前终止循环前的秒数。

此循环很可能超过 5 秒的限制。在 Wonderware Logger 中，会出现一则如下所示的消息：

```
95/03/07 07:34:40.550/VIEW    /
Exceeded loop time limit of 5 seconds.

95/03/07 07:34:40.550/VIEW    /
FOR-NEXT Timeout at X = 65464
```

有关 Wonderware Logger 程序的详细信息，请参阅您的联机 *《FactorySuite 系统管理员指南》*。

此错误消息指出在满足 *end_condition* 之前 FOR-NEXT 循环已经终止，它还提供了循环终止时的循环变量值。此信息允许您跟踪哪一条 FOR-NEXT 语句存在问题。

注意：5 秒时间限制仅当 FOR-NEXT 循环中每次到达 NEXT 语句时计算。例如，假定您执行下面的脚本：

```
FOR Index = 1 to 10

SQLInsert(ConnectionID,"ORG","list1");
SQLInsert(ConnectionID,"ORG","list2");
SQLInsert(ConnectionID,"ORG","list3");
SQLInsert(ConnectionID,"ORG","list4");

NEXT;
```

如果每个 SQLInsert() 函数需要 12 秒来完成，则所有四个插入动作会在循环退出之前完成执行，这是因为五秒时间限制只有在到达 NEXT 语句时才会计算。

循环执行后的循环变量值

在 Visual Basic（和大多数的其它流行编程语言）中，循环结束时的循环变量值定义如下：

从初始值 *Start_Condition* 开始，索引值按增量 *Step_Expression* 持续增大，直至到达最后一次循环，此时索引值超过 *End_Expression* 值。

因此，如果某个循环如下所示：

```
FOR Index = 2 TO 25 STEP 7
   { Some script statements }
NEXT;
```

则索引值将如下变化：

迭代	值	计算
1	2	
2	9	2 + 7
3	16	2 + 7 + 7
4	23	2 + 7 + 7 + 7
5	30	2 + 7 + 7 + 7 + 7

当值到达 30 时，因为超过 *End_Expression*，循环将停止。索引的最终值为 30。

嵌套控制结构

一个控制结构可以置于其它控制结构中（如 FOR...NEXT 循环内的 IF...THEN 语句）。一个控制结构置于另一控制结构中称为*嵌套*。

实例：

```
FOR TagX = 1 TO 5
FOR TagY = 1 TO 10
   ...statements...
   IF (condition) THEN
   [EXIT FOR;]
   ENDIF;
   ...statements...
   NEXT;
NEXT;
```

此处：

第一个 NEXT 结束内层 FOR 循环，最后一个 NEXT 结束外层 FOR 循环。类似地，在嵌套 IF 语句中，ENDIF 语句自动对于前面最近的 IF 语句。

退出控制结构

EXIT FOR 语句允许您从 FOR 循环中直接退出。EXIT FOR 在语法上很简单。EXIT FOR 可以根据需要在 FOR 循环内出现多次。例如：

```
FOR TagX = 1 TO 10;
...statements...
IF (condition) THEN
    EXIT FOR;
ENDIF;
...statements...
NEXT;
```

下面是各种 FOR-NEXT 循环脚本的一些例子：

例 1： “简单的数学运算 2”

此循环通过设置标记名 NumberToRaise 和 Power 的值输入链接，允许使您设定乘方数和幂：

```
Product = 1;

NumberToRaise = 4;

Power = 12;

FOR Index = 1 TO Power
    Product = Product * NumberToRaise;

NEXT;
```

一旦上述脚本执行完毕，Product 的值将为 16,777,216。

例 2：“使用间接标记名的复杂 FOR-NEXT”

此循环利用 FOR-NEXT 结构的“EXIT FOR”和“STEP”部分对 100 个标记名进行搜索，以找到 NumberEntered 的等量标记名。

注意：本实例假定已经存在 100 个内存整型标记名 (TAG1 - TAG100)。操作员将数字输入到标记名 NumberEntered 中，循环结构将搜索 TAG1 - TAG100 以找出匹配值。此外，还创建一个间接模拟标记名：*IndirectTag*

```
Found = 0;

FOR Index = 1 TO 100

   IndirectTag.NAME = "TAGNAME" + TEXT( Index, "#" );

   IF (IndirectTag.NAME == ("TAGNAME"+
      Text(NumberEntered,"#"))) THEN

      Found = 1;

      EXIT FOR;

   ENDIF;

NEXT;

IF (Found==1) THEN

   Show "NumberFound"; {window notifying search was successful}


ELSE

Show "NumberNotFound";

ENDIF;
```

一旦脚本执行完毕，一个窗口将显示，指明是否找到该数值。

注意：注意此脚本中两个附加函数的用法：Show() 和 TEXT()。

例 3：

此循环执行奇数运算，但演示了嵌套 FOR-NEXT 循环的用法，以及 FOR-NEXT 结构中“STEP”部分的使用：

```
MyTag = -1;

FOR Index = 1000 TO -1000 STEP -5

   IF (MyTag > Index) THEN

      FOR Index2 = 1 TO 10 STEP 2

         MyTag = MyTag * (Index + 11);

      NEXT;

   ENDIF;
NEXT;
```

脚本完成处理后的值为：

MyTag = -7776、Index = -1005 以及 Index2 = 11。

脚本编辑样式和语法

InTouch 脚本编辑器支持两种脚本“样式”：“简单”和“复杂“。简单脚本允许赋值、比较、执行简单数学函数等；复杂脚本能以 IF-THEN-ELSE 型语句的形式执行逻辑操作。此外，InTouch 还支持使用内置复杂函数，以及本地 QuickFunction。

复杂函数的一个例子是 **StartApp(*ApplicationName*)**，此函数实际启动在参数 "(*ApplicationName*)" 中确定的 Windows 应用程序。函数可用于编写简单和复杂脚本。下面的章节将完整介绍每种样式。

表达式和脚本的基本语法

QuickScript 和动画链接表达式对话框中所用的语法与计算器的代数语法很相似。大多数的表达式都是以下形式的赋值语句：

```
a = (b - c) / (2 + x) * xyz;
```

此语句将等号 (=) 右边表达式的值赋给变量“a”。每个表达式必须以分号 (;) 结尾。表达式中的运算数可以是常量也可以是变量。单个标记名必须出现在赋值运算符 (=) 的左边。

内存或 **I/O 消息**型标记名可以通过使用加号 (+) 运算符连接。例如，标记名可以在连接后用于**间接**型标记名中。如果创建了下面的数据改变脚本，则每当 "Number" 值改变时，间接标记名 "Setpoint" 都会相应改变：

```
Number=1;
Setpoint.Name = "Setpoint" + Text(Number, "#" );
```

此处：结果为“Setpoint1”。

简单脚本

简单脚本提供赋值、数学运算和函数等逻辑功能。下面是简单脚本的一个例子：

```
React_temp = 150;

ResultTag = (Sample1 + Sample2)/2;

{this is a comment}

Show "Main Menu";
```

在本例中，脚本对标记名 React_temp 赋值 150。"Sample1" 与 "Sample2" 相加，所得结果除以“2”，并在运行脚本时在屏幕上显示 "Main Menu" 窗口。

注意：注意每个逻辑语句必须以分号（;）结尾，一个脚本可以包括多个逻辑语句。另外，在脚本编辑器中允许加上注释。注释用一对花括号 {}标明。参数 ″Main Menu″（窗口名）也使用函数 Show，以打开指定的窗口。

除了简单的赋值、数学运算符和函数外，InTouch 还支持使用“运算数”的其它运算，如标记名、常数等等。下面列出的所有运算均支持**离散**、**整型**和**实型**标记名。**消息**型标记名只能用于比较和赋值运算。下面是 InTouch 支持的运算列表：

需要一个运算数的运算

~	取补
-	取反
NOT	逻辑 NOT

需要两个运算数的运算

*	乘
/	除
+	加和连接
-	减
=	赋值
MOD	取余数
SHL	左移位
SHR	右移位
&	位与
^	异或
|	同或
**	乘方
<	小于
>	大于
<=	小于或等于
>=	大于或等于
==	等于（“等价于”）
<>	不等于
AND	逻辑与
OR	逻辑或

运算符的优先级

下面列出了运算符运算的优先级次序。列表中的第一个运算符将首先求值，第二个运算符其次求值，以此类推。同一行中的运算符具有相同的优先级。运算符按最高优先级到最低优先级的次序列出。

( ) 最高优先级

- , NOT, ~

**

* , /, MOD

+, -

SHL, SHR

<, >, <=, > =

==, <>

&

^

|

=

AND

OR 最低优先级

优先级实例

因为 * 优先级高于 +，

所以 B+C*D 等价于 B+(C*D)；

因为 * 和 / 具有相同优先级，

所以 B / C*D 等价于 (B / C)*D；

再看其它几个例子：

B* - D 等价于 B* (- D)；

B or C and D 等价于 B or (C and D)。

运算符的描述

上述运算符的参数可以是数字也可以是标记名。在参数外加括号是可选的，而且运算符名称不区分大小写。

括号 ( )

括号用于保证运算求值的正确顺序。它们也可使复杂的表达式更易于阅读。括号内的运算最先求值（优先于没有括号时的其它优先级规则）。如果您不清楚或需要改写运算的优先顺序，请使用括号。在下面的例子中，使用括号强制 B 和 C 相加后再乘以 D：

```
( B + C ) * D;
```

取反(-)

取反是个一元运算符，它将正整数或正实数转换成负数。

取补 (~)

此运算符产生 32 位整数的补数。换句话说，它将每一个 0 位变成 1 位，每个 1 位变成 0 位。补数运算符是一个接受整型运算数的一元运算符。

乘方 (**)

此二元运算符返回一个乘方值。底和幂可以是任意实数或整数，但需遵守以下限定：

零底和负数幂无效。

例如：**"0 ** -2"** 和 **"0 ** -2.5"**

负数底和分数幂无效。

例如：**"-2 ** 2.5"** 和 **"-2 ** -2.5"**

无效的运算数得出的结果为 0。此外，运算结果不能过大或过小，以致无法用实数表示。例如：

```
1 ** 1 = 1.0
3 ** 2 = 9.0
10 ** 5 = 100,000.0
```

乘 (*)、除 (/)、加 (+)、减 (-)

这些二元运算符执行基本的数学运算。加号 (+) 也可用于连接**内存**或 **I/O 消息**型标记名。例如，标记名可在连接后用于**间接**标记名中。如果创建了下面的数据改变脚本，则每当“Number”值改变时，间接标记名“Setpoint”会相应改变：

```
Number=1;
Setpoint.Name = "Setpoint" + Text(Number, "#" );
```

此处：结果为“Setpoint1”。

取余数 (MOD)

MOD 是一个二元运算符，它用右边的整数除左边的整数，所得余数即是 MOD 运算的结果。例如：

97 MOD 8 等于 1
63 MOD 5 等于 3

左移位 (SHL)、右移位 (SHR)

SHL 和 **SHR** 是只使用整型运算数的二元运算符。运算符左边的 32 位字的二进位内容将移动（右或左）运算符右边指定的位数。移出的位将丢失。移走后空的位将填入零（此移位是无符号移位）。

位与 (&)

这是一个二元位与运算符，它比较两个 32 位整型字的每一位。常用此运算符来屏蔽一系列位。下例中的运算将屏蔽掉（设为 0）32 位字的高 24 位：例如：

```
result = name & 0xff;
```

异或 (^) 和同或 (|)

异或和**同或**是位逻辑运算符，它们比较两个 32 位整型字的每一位。**异或**寻找对应位置的相反位状态。如果对应位相同，则得出结果 0；如果对应值相反，则得出 1。例如：

0 ^ 0 得 0

0 ^ 1 得 1

1 ^ 0 得 1

1 ^ 1 得 0

同或检查对应位是否为 1。如果两个对应位有一个为 1，结果即为 1。只有在两个对应位均为 0 时才得 0。例如：

0 | 0 得 0

0 | 1 得 1

1 | 0 得 1

1 | 1 得 1

赋值 (=)

赋值是接受整型或实型运算数的二元运算符。每个语句只能包含一个赋值运算符。赋值运算符左边只能有一个名称。赋值运算符的等号 (=) 读作“赋值为”或“设为”。

注意：请勿将等号与 IF-THEN-ELSE 子句及相对关系中使用的等价符号（“==”）相混淆。

比较运算符 (<, >, <=, >=, ==, <>)

这些运算符用于 IF-THEN-ELSE 语句中，用以执行各种基于表达式状态的指令。

AND（与）、OR（或）和 NOT（非）

这些运算符只适用于离散型标记名。如果这些运算符用于整数或实数，它们将如下*转换*：

实型到离散型：如果实数为 0.0，则离散量为 0，否则为 1；

整型到离散型：如果整数是为 0，则离散量为 0，否则为 1。

因此，如果表达式是：“Disc1 = Real1 AND Real2”；并且 Real1 为 23.7，Real2 为 0.0，则 Disc1 将赋为 0，因为 Real1 会转换成 1，Real2 会转换成 0。

复杂脚本

复杂脚本使您能以 IF-THEN-ELSE 脚本的形式执行逻辑运算，使用 FOR-NEXT 脚本结构来处理循环。下面是 IF-THEN-ELSE 脚本的一个例子:

```
IF React_temp > 200 THEN
    React_temp_sp = 150;
    PRValve = 1;
    PlaySound("c:\alert.wav",1);
ELSE
    PRValve = 0;
    PlaySound("c:\All_Ok.wav",1);
ENDIF;
```

在此例中，脚本检查反应炉温度是否高于 200 度。如果超过，将“150”的值赋给“React_temp_sp”标记名，“PRValve”将打开，并通过调用 **Playsound()** 函数播放“alert.wav”文件。反之，如果炉温低于“200”，则关闭“PRValve”并播放“All_Ok.wav”文件。

注意：每个 IF 语句需要有对应的 ENDIF 语句。如果脚本的功能不需要，则不必包括 ELSE 语句。请注意此复杂脚本中函数 PlaySound(*path_text,number*) 的用法。

简单数学运算

此循环执行简单的迭代数学计算。Product 等于 NumberToRaise 的 10 次方，即 **Product=NumberToRaise^{10}**。

```
Product = 1;
NumberToRaise = 4;
FOR Index = 1 TO 10
    Product = Product * NumberToRaise;
NEXT;
```

一旦执行此脚本，Product 值将为 1048576。

注意：FOR-NEXT 循环会暂停 InTouch 中的所有运算。在执行期间，WindowViewer 中没有数据移动、没有动画链接更新，也没有其它脚本执行，包括异步 QuickFunction。但是，异步 QuickFunction 中使用的 FOR-NEXT 循环不能暂停其它运算。

脚本中的 IF-THEN-ELSE 和比较运算

IF-THEN-ELSE 语句用于根据表达式的状态有条件地执行不同的指令。下面的比较运算符用于设置 IF-THEN-ELSE 语句的条件：

< 小于

> 大于

<= 小于或等于

>= 大于或等于

== 等于（“等价于”）

<> 不等于

以下是各种复杂脚本的一些例子：

IF-THEN 语句，无 ELSE 子句：

```
IF a <> 0 THEN
   a = a + 100;
ENDIF;
```

IF-THEN-ELSE 语句，包含一个 ELSE 子句：

```
IF temp > 500 THEN
   Disc = 1;
   Real = 43.7;
ELSE
   Disc = 0;
   Real = 93.4;
ENDIF;
```

IF-THEN-ELSE 语句，包含一个 ELSE IF 子句，无 ELSE 子句：

```
IF temp > 500 THEN
   Disc = Disc * 10;
ELSE
   IF temp > 250 THEN
      x = y / z;
      a = abc + def;
   ENDIF;
ENDIF;
```

IF-THEN-ELSE 语句，包含一个 END IF 子句和一个 ELSE 子句：

```
IF temp > 500 THEN
   Disc = Disc - 10;
ELSE
   IF temp < 250 THEN
      Disc = Disc + 10;
   ELSE
      Disc = Disc + 50;
      Real = 100;
   ENDIF;
ENDIF;
```

注意：每个 IF 语句必须有对应的 ENDIF，并且每个语句行必须用分号结束。

IF-THEN-ELSE 语句，包含多个 ELSE IF 子句和一个 ELSE 子句：

```
IF temp > 100 THEN
    temphihi = 1
    Disc = 50;
ELSE
    IF temp > 80 THEN
        temphi = 1;
ELSE
    IF temp < 10 THEN
        templo = 1;
ELSE
    IF temp < 30 THEN
        templolo = 1;
ELSE
        tempok = 1;
    ENDIF;
ENDIF;
ENDIF;
ENDIF;
```

IF-THEN-ELSE 语句，检测条件 1 *或*条件 2：

```
IF (pump1 < 50.0) OR (pump2 < 50.0) THEN
    alarm-1 = 1;
ELSE
    alarm-1 = 0;
ENDIF;
```

IF-THEN-ELSE 语句，检测条件 1 *和*条件 2：

```
IF (pump1 < 50.0) AND (pump2 < 50.0) THEN
    alarm-2 = 1;
ELSE
    alarm-2 = 0;
ENDIF;
```

IF-THEN-ELSE 语句检测是否相等：

```
IF a > 50 THEN
    IF b == 100 THEN
        c = 0;
    ENDIF;
ENDIF;
```

导入 QuickScript

将 QuickScript 从一个 InTouch 应用程序导入当前应用程序可以节省大量时间。同时还为您提供一种建立远程标记名引用的简便快捷方法。它允许您重复使用先前建立的 QuickScript。在将 QuickScript 从一个 InTouch 应用程序移动到另一应用程序时，您必须使用“**文件**”菜单上的“**导入**”命令。

有关远程标记名引用的详细信息，请参阅第 4 章“标记名字典”。

➢ **要导入 QuickScript：**

1. 关闭当前应用程序中的所有窗口。
2. 在“**文件**”菜单上，单击“**导入**”。“**浏览文件夹**”对话框出现：



3. 找到并且选择包含您要导入的 QuickScript 的应用程序目录（文件夹）。
4. 单击“**确定**”。出现下面的对话框：



5. 选择您要导入的 QuickScript 类型。

6. 单击“**选择**”。“**选择 *ScriptType* 脚本**”对话框出现：



7. 选择您要导入的 QuickScript，然后单击“**确定**”关闭对话框。

注意：当您将 ActiveX 事件脚本从一个应用程序导入另一应用程序时，所有 ActiveX 事件脚本均将导入。此外，为使导入的 ActiveX 事件脚本能够在新应用程序中正常运行，原先为之创建脚本的相同 ActiveX 控制和相同事件也必须用于新应用程序中，并且必须载入到内存中。如果关闭包含 ActiveX 控件的窗口，则其 ActiveX 事件脚本，或包含与该 ActiveX 控件相关联的脚本函数的任何其它 InTouch QuickScript 将不能正常执行。

---

8. 单击“**导入**”，系统会自动开始将选定的 QuickScript 导入到当前应用程序中。

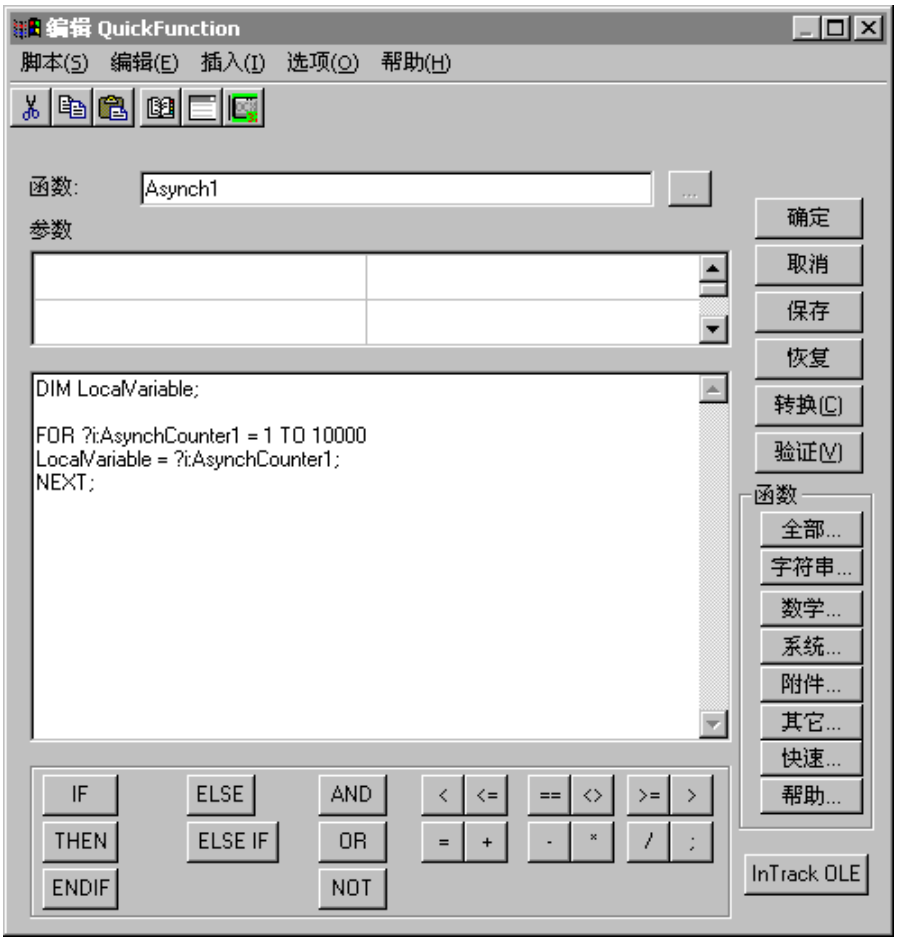
注意：在将 QuickScript 导入新应用程序时，QuickScript 中的所有标记名将随 QuickScript 一起导入，但是它们不会添加到您的标记名字典中，而是自动转换成“占位符”标记名。您必须先转换这些占位符标记名，然后才能使用它们，如果当前还没有在新应用程序的标记名字典中定义，则系统会要求您对其分别定义。

当所导入 QuickScript 中的标记名转换成占位符标记名时，每个标记名开头会加上三个索引字符。例如，当导入一个离散型标记名时，该标记名的开头会有三个字符“**?d:**”。当导入一个长度为 30、31 或 32 字符的标记名时，每个标记名开头仍会添加这三个索引字符。但是，增加这三个字符不会截短现有标记名的长度。例如，占位符标记名的 32 字符标记名会增加到 35 字符。这三个增加的空间仅分配用于占位符标记名。标准标记名不支持这种标记名长度的增加。

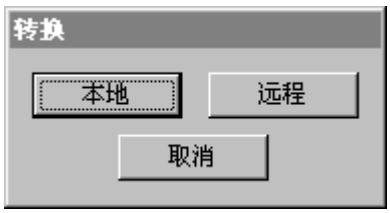
➢ **要转换所导入脚本中的占位符标记名：**

1. 在“**特别**”菜单上，指向“**脚本**”，然后单击导入的 QuickScript 类型，或者在应用程序浏览器的“**脚本**”下，双击导入的 QuickScript 类型。QuickScript 编辑器出现，显示所选脚本类型的第一个 QuickScript。例如，如果您导入了 **QuickFunction** 脚本，则 **QuickFunction** 脚本编辑器会出现：

   要快速打开导入的 QuickScript，在应用程序浏览器中双击“**脚本**”，然后双击 QuickScript 类型，。



2. 单击“**转换**”，“**转换**”对话框出现：



3. 单击“**本地**”以将 QuickScript 中的标记名转换成本地标记名。
4. 在转换标记名后，单击 QuickScript 编辑器中的“**确定**”。

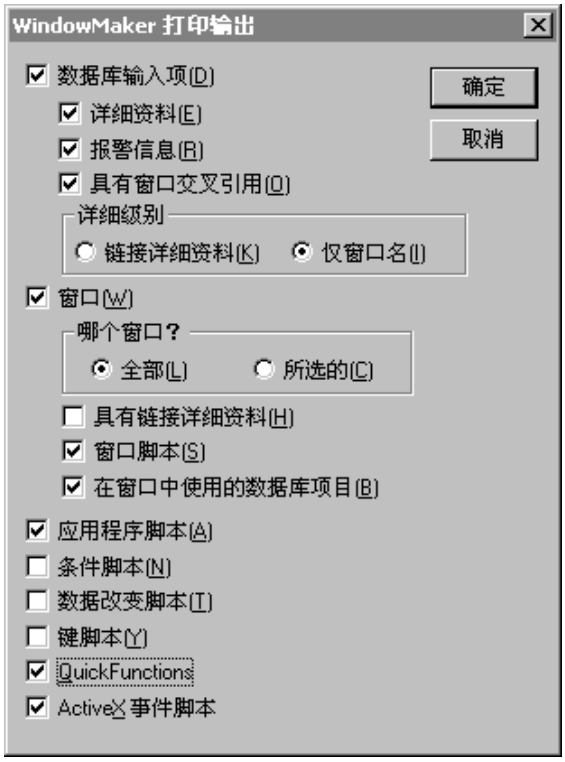
有关转换为远程标记名引用的详细信息，请参阅第 4 章“标记名字典”。

打印脚本

您可以打印各种 InTouch QuickScript 类别中的所有脚本。

➢ **要打印脚本：**

1. 在“**文件**”菜单上，单击“**打印**”。“**WindowMaker 打印输出**”对话框出现：



2. 要打印窗口脚本，选择“**窗口**”，然后选择“**窗口脚本**”。在“**哪个窗口？**”组中，选择“**全部**”以打印应用程序中所有窗口的脚本。要打印指定窗口的脚本，选择“**所选的**”。“**打印窗口**”对话框出现。选择您要打印其脚本的窗口，然后单击“**确定**”。

   如果您选择的窗口没有与之链接的脚本，则报表中会打印下面的消息：“*窗口名称*的窗口脚本：无”。

3. 要打印 QuickScript 类型的所有脚本，选择 QuickScript 类型，然后单击“**确定**”。

脚本函数

InTouch 提供了大量的内置函数，利用这些函数，您可以链接到对象或按钮，或者在脚本中使用它们来执行各种任务。例如，确认警报、隐藏窗口、改变当前绘制趋势图的标记名等等。

这些函数可通过“**插入**”菜单或单击脚本编辑器“**函数**”区域中的相应按钮进行访问。一旦在相应的对话框中选定了函数，该函数及其必要参数将自动粘贴到脚本中的光标位置。在将函数粘贴到脚本后，您就可以辉亮显示要修改的参数，然后输入新值。

字符串函数

字符串函数在字符串变量中使用。下面简要介绍各种字符串脚本函数。

函数	描述
Dtext	根据离散型标记名的值来更改消息型标记名。
StringASCII	返回指定消息标记名中第一个字符的 ASCII 值。
StringChar	返回指定 ASCII 码的对应字符。
StringFromIntg	将整型值转换成其它形式的字符串。
StringFromReal	将实型值转换成浮点数或指数形式的字符串。
StringFromTime	将时间值（自 1970 年 1 月 1 日起以秒为单位）转换成指定的字符串表示。
StringInString	返回*搜索内容*首先找到的*文本*位置。
StringLeft	从文本最左端的字符开始，返回由 Chars 指定的字符数。
StringLen	返回文本长度为一整型值。
StringLower	将文本中的所有大写字符转换为小写，并将结果字符串返回 MessageResult。
StringMid	从位置 *StartChar* 开始，从文本中提取由 Chars 指定的字符数。此函数与其对应函数 StringLeft() 和 StringRight() 稍有不同，它允许用户指定从消息标记中提取的字符串的首尾位置。

函数	描述
StringReplace	替换或更改指定字符串的某一部分。
StringRight	从文本最右端的字符开始，返回由 Chars 指定的字符数。
StringSpace	产生消息型标记名或表达式中的空格字符串。
StringTest	检验文本的第一个字符以决定其类型。
StringToIntg	将消息型标记名中的数值转换成可应用数学运算的整型值。
StringToReal	将消息型标记名的数值转换成可应用数学运算的实型（浮点）值。.
StringTrim	从文本中移除不需要的空格。
StringUpper	将文本中的所有小写字符转换成大写。
Text	使消息型标记名根据指定的 ***Format_Text*** 显示模拟（整型或实型）标记名的值。

有关每个函数的有效语法及其使用实例的详细信息，请参阅您的联机*《InTouch 参考指南》*。

数学函数

数学函数在整型或实型标记名中使用。在下面的数学函数中，**ResultNumericTags** 和 **InputNumericTags** 既可以是实型也可以是整型，并且可以混合使用。但要记住，将函数的非整型结果返回为**整型**标记名会导致返回值被截短（小数点右边的部分将丢失）。下面的例子假定 ***ResultNumericTag*** 定义为**内存实型**或 **I/O 实型**标记名。

下面简要介绍各种数学脚本函数。

函数	描述
Abs	返回指定数字的绝对值（无符号的等价值）。
ArcCos	给定一个介于 -1 和 1（包括）之间的数字，返回其*余弦*等于该数字的介于 0 和 180 度之间的一个角。
ArcSin	给定一个介于 -1 和 1（包括）之间的数字，返回其*正弦*等于该数字的介于 -90 和 90 度之间的一个角。
ArcTan	给定一个数值，返回其*正切*等于该值的介于 -90 和 90 度之间的一个角。
Cos	返回给定角（以度表示）的*余弦*值。
Exp	返回 ***e*** 的某次幂。
Int	返回小于或等于指定数字的下一个整数。
Log	返回某个数字的自然对数值。
LogN	返回以 ***n*** 为底的 ***x*** 的*对数*值。
Pi	返回 *Pi* 的值。
Round	将实数舍入为指定的精度。
Sgn	确定值的符号（正、零或负）。
Sin	返回给定角（以度表示）的*正弦*值。
Sqrt	返回某个数字的*平方根*。
Tan	返回给定角（以度表示）的*正切*值。
Trunc	通过简单地删除小数点右边的部分来截短一个实型（浮点）数字。

有关每个函数的有效语法及其使用实例的详细信息，请参阅您的联机*《InTouch 参考指南》*。

系统函数

系统函数用于执行系统动作，例如激活另一个 Windows 应用程序，复制、删除或移动文件，以及检索有关应用程序的信息等。系统函数有两种类型：文件和信息。系统文件函数用于读写文件中的数据。它们有两个公共参数：***Filename*** 和 ***FileOffset***。

Filename 指从中读写的文件名，此文件名必须包括完整的路径。***FileOffset*** 指开始读写操作的文件位置（从文件头开始，以字节计算）。文件的第一个字节为 ***FileOffset*** 0。当执行完毕后，每个函数会返回文件刚刚读取或写入数据后的第一个字节位置。例如，如果函数从字节位置 10 开始读入 5 个字节的数据，则会返回值 15。

FileOffset 标记名既可用作函数参数，也可用作返回标记名。这样便可以实现连续操作。

例如：

***FileOffset*=FileReadMessage(*Filename,FileOffset,Message_Tagname,0*);**

在上面的实例中，从 ***Filename*** 读取一行文本。起始位置由 ***FileOffset*** 的初始值指定（例如 0，即是指文件的开始位置）。下一次读取的开始位置返回给 ***FileOffset***，以准备下一次调用 **FileReadMessage( )**。每次调用函数时，随着 **FileReadMessage( )** 读取文件，***FileOffset*** 会变得越来越大。

下面简单介绍每个系统脚本函数。

函数	描述
ActivateApp	激活另一个当前运行的 Windows 应用程序。
FileCopy	复制 *SourceFile* 到 *DestFile*，与 DOS 复制命令或 Windows 文件中的复制功能相似。
FileDelete	删除不必要或不需要的文件。
FileMove	类似于 FileCopy()，不同的是它将文件从一处移动到另一处，而不是复制一份。
FileReadFields	从指定的文件读取逗点分离变量 (CSV) 记录。
FileReadMessage	从指定的文件读取指定的字节数（或整行）。
FileWriteFields	将逗号分离变量 (CSV) 记录写入指定文件。
FileWriteMessage	将指定的字节数（或整行）写入指定的文件。

函数	描述
InfoAppActive	测试应用程序是否处于活动状态。
InfoAppTitle	返回应用程序标题或当前运行的指定程序的 Windows 任务列表名。
InfoDisk	返回指定的本地（或网络）磁盘驱动器信息。
InfoFile	返回指定文件或子目录的信息。
InfoInTouchAppDir	返回当前 InTouch 应用程序目录。
InfoResources	返回下列各种系统资源值： **第 1 和第 2 种情形：** 在 Windows NT 下，GDI 和 USER 是硬编码的，返回 50%。 **第 3 种情形：** 在 Windows NT 下返回“分页文件的可用字节”。 **第 4 种情形：** 在 Windows NT 下返回搜索所有顶层窗口的结果。它仅计算可见的独立窗口。这不是系统中实际的“当前运行的任务数”。其最接近的数字是当您在 Windows NT 中运行任务管理器时显示在“应用程序”标签上的项目计数。
IsAnyAsyncFunctionBusy	用于找出是否有任何异步 QuickFunction 正在运行。使用此函数，您可以使调用异步 QuickFunction 的 QuickScript 等待所有其它异步 QuickFunction 完成处理。这允许 QuickScript 自我重新同步。
StartApp	自动启动另一 Windows 应用程序。

有关每个函数的有效语法及其使用实例的详细信息，请参阅您的联机*《InTouch 参考指南》*。

其它函数

您可以使用其它一些函数来执行其它一些动作，如隐藏窗口、监控历史趋势图表、打印窗口、向其它应用程序发送键等。

下面简单介绍各种其它脚本函数。

有关新报警函数的列表，请参阅第 7 章“报警/事件”。

有关新报警函数的详细信息，请参阅 *《InTouch 参考指南》*。

- 以“alm”开头的函数名仅用于分布式报警系统。以“wc”开头的函数名用于窗口控件对象（列表框、文本框、组合框等）。以“HT”开头的函数名仅用于历史趋势对象。

函数	描述
Ack	确认任何尚未确认的报警。此函数可应用于标记名、报警组或组变量。
AlmAckAll	确认当前请求的所有报警，包括当前未显示在报警显示对象中的报警。
AlmAckDisplay	仅确认报警显示对象中当前可见的那些报警。
almAckRecent	确认最近发生的报警。
almAckSelect	仅确认报警显示对象中选定的那些报警。
almDefQuery	使用缺省属性执行查询以更新报警显示对象。
almMoveWindow	滚动报警显示对象窗口。
almQuery	执行查询以更新报警显示对象。
almSelectAll	切换选择报警显示对象中的所有报警。
almSelectItem	切换选择报警显示对象中辉亮显示的项目。
almShowStats	显示报警显示对象的统计屏幕。
ChangePassword	显示“改变口令”对话框以允许登录操作员改变其口令。
DialogStringEntry	在屏幕上显示字母数字键盘，允许操作员改变标记名字典中消息型标记名的当前字符串值。
DialogValueEntry	在屏幕上显示数字小键盘，允许操作员更改标记名字典中离散型、整型或实型标记名的当前值。

函数	描述
GetNodeName	返回 NetDDE 节点名给字符串变量。
GetPropertyD	在运行时检索指定属性的离散值。
GetPropertyI	在运行时检索指定属性的离散值。
GetPropertyM	在运行时检索指定属性的消息值。
Hide	在脚本中隐藏不同的窗口。Hide 函数必须位于要隐藏的每个窗口名之前。
HideSelf	隐藏当前的活动窗口。
HTGetLastError	确定在最后一次检索指定笔时是否出现错误。
HTGetPenName	返回指定趋势的指定笔号当前所用的标记名。
HTGetTimeAtScooter	返回由 *ScootName* 和 *ScootLoc* 指定的指示器位置处的样本自 1970 年 1 月 1 日 GMT 00:00:00 以来的时间（以秒为单位）。
HTGetTimeStringAtScooter	返回由 *ScootNum* 和 *ScootLoc* 指定的指示器位置处的样本包含时间/日期的字符串。
HTGetValue	返回整个趋势的指定笔的请求类型值。
HTGetValueAtScooter	返回指定的指示器位置，趋势和笔号处的样本的请求类型值。
HTGetValueAtZone	针对包含某一趋势指定笔的左右指示器位置之间的数据，返回请求类型值。
HTScrollLeft	将趋势的起始时间值设置为比当前起始时间早趋势宽度的一定百分比。其效果是将图表的日期/时间向左滚动给定的百分比。
HTScrollRight	将趋势的起始时间值设置为比当前起始时间晚趋势宽度的一定百分比。其效果是将图表的日期/时间向右滚动给定的百分比。
HTSelectTag	显示“选择标记”对话框，使操作员可以为指定的笔选择不同的标记名此对话框仅列出标记名字典中的历史记录（选定“记录数据”选项）的已定义标记名。

函数	描述
HTSetPenName	对趋势笔指定不同的标记名。
HTUpdateToCurrentTime	通过使结束时间等于当前时间来检索和显示数据。开始时间将等于结束时间减去图表的宽度。
HTZoomIn	计算新图表宽度和开始时间。如果趋势的 **.ScooterPosLeft** 为 0.0 而 **.ScooterPosRight** 为 1.0，那么新图表的宽度将等于旧图表宽度除以 2。
HTZoomOut	计算新图表的宽度和开始时间。新图表的宽度等于旧图表的宽度乘以 2。
IOReinitialize	关闭所有的现有 I/O 对话并重新启动设置 I/O 对话的全过程。当执行此函数时，所有 I/O 点均会受影响。
IOGetApplication	返回为特定访问名定义的应用程序名给指定的标记名。
IOGetNode	返回为特定访问名定义的节点信息（地址）给指定的标记名。
IOGetTopic	返回为特定访问名定义的主题名给指定的标记名。
IOReinitialize	关闭所有的现有 I/O 对话并重新启动设置 I/O 对话的全过程。所有的 I/O 点均将受影响。
IOSetAccessName	在运行时修改访问名的*节点*、*应用程序*或*主题*名称部分，以允许实施 InTouch 的热备份策略。
IOSetItem	更改 I/O 型标记名 ***.Reference*** 点域中的访问名和（或）项目。
LogMessage	将用户自定义消息写入 Wonderware Logger。
PlaySound	通过 Windows 声音设备（如已安装），播放由 .wav 文件名或 WIN.INI 文件的[声音]部分中的项目指定的 wave 格式声音。
PrintHT	可用于按钮上打印与指定历史趋势类型标记名相关联的历史趋势图。使用此函数时，历史趋势必须在屏幕上可见。

函数	描述
PrintScreen	打印指定的屏幕。
PrintWindow	打印指定的窗口。
ReloadWindowViewer	允许用户控制重新载入 WindowViewer 的过程。
RestartWindowViewer	允许用户控制关闭和重新启动 WindowViewer 的过程。
SendKeys	向应用程序发送键。
SetPropertyD	指定要在运行时写入的属性的离散值。
SetPropertyI	指定要在运行时写入的属性的整型值。
SetPropertyM	指定要在运行时写入的属性的消息型值。
Show	显示指定的窗口（窗口名必须用引号括起）。
ShowAt	指定窗口显示时的水平和垂直象素位置。当窗口打开时，它将位于水平和垂直坐标的中心。
ShowHome	显示“起始”窗口。起始窗口是选定在 WindowViewer 启动时自动打开的窗口（起始窗口在“WindowViewer 属性- 起始窗口”属性页中选定）。 有关起始窗口的详细信息，请参阅第 2 章“使用 WindowMaker”。
ShowTopLeftAt	指定窗口显示时其左上角的水平和垂直象素位置。当窗口打开时，其左上角将位于水平和垂直坐标的相交处。
wcAddItem	在列表框或组合框中加入指定的字符串。
wcClear	从列表框或组合框中删除所有项目。
wcDeleteItem	删除与列表框或组合框中的项目索引参数关联的项目。
wcDeleteSelection	从列表中删除当前选定的项目。适用于列表框和组合框。

函数	描述
wcErrorMessage	对于给定的错误号，**wcErrorMessage()** 返回描述该错误的一个字符串消息。适用于列表框、文本框、组合框、单选钮和复选框。
wcFindItem	决定列表框或组合框中与指定字符串匹配的第一个项目的对应索引。
wcGetItem	返回与列表框或组合框中的相应索引相关联的项目字符串的值属性。
WcGetItemData	检索与列表框或组合框中的列表项目关联的整型值。
WcInsertItem	在列表框或组合框中插入字符串。
wcLoadlist	用新项目替代列表框或组合框的内容。
wcLoadText	用新字符串替代文本框的内容。
wcSavelist	用列表对象中的项目替代文件名的内容。
wcSaveText	将文本框中包含的文本保存到文件名。
wcSetItemData	对列表框中的项目指定一个整型值。

有关每个函数的有效语法及其使用实例的详细信息，请参阅您的联机*《InTouch 参考指南》*。

WW DDE 函数

您不应该使用 WW DDE 函数作为常规 InTouch DDE 通讯的替代。只要有可能，您就应该创建 DDE 型标记名来从外部应用程序获取数据或向外部应用程序发送数据。WW DDE 函数用于支持那些不能使用 InTouch 支持的 DDE 进行通讯的应用程序。例如，某些应用程序仅支持 DDE 执行或赋值。

WWExecute()、**WWPoke()** 和 **WWRequest()** 函数使用与 Visual Basic (DDEML) 相同的 Windows 函数。一个简单的函数实际上可以执行好几件事情。例如，**WWPoke()** 可以在一个函数中执行 DDE 初始化、DDE 赋值和 DDE 终止。这使得 WW DDE 函数不易出错，但同时也降低了处理大量 DDE 消息的效率。作为使用这些函数的一般准则，您切勿：

循环使用这些函数（多次重复调用）。

在一行和相同的脚本中调用多个 DDE 函数。

在另一 DDE 应用程序中使用它们来调用冗长的任务。

如果 DDE 命令执行另一应用程序中的冗长任务，则它可能会用完所有可用的处理器时间。但是，即使在通讯发生困难时，也不会丢失数据；即使 I/O 服务器不能向 InTouch 发送消息，它也会继续尝试。

下面简要介绍每个脚本函数。有关每个函数的有效语法及其使用实例的详细信息，请参阅您的联机*《InTouch 参考指南》*。

函数	描述
WWControl	允许您从 InTouch 恢复、最小化、最大化或关闭应用程序。
WWExecute	向指定的*应用程序*和*主题*发送命令（使用 DDE 执行）。
WWPoke	向指定的*应用程序*、*主题*和*项目*插入数值（使用 DDE 赋值）。
WWRequest	一次性请求特定*应用程序*、*主题*和*项目*的数值（使用 DDE 请求）。

脚本编辑器错误消息

如果脚本编辑器在验证脚本时检测到任何错误，它会显示一个相应的消息框：例如：



在大多数情况下，当脚本遇到错误时，InTouch 会将光标置于出错位置。但是，在某些情况下（例如缺少 ENDIF 语句），光标会出现在脚本末尾。只有在纠正所有错误后，系统才会接受脚本。

错误消息	定义
仅可比较报警组的相等性	不能比较报警组的 <, >, <=, >= 关系。
对此字符串不能进行加减乘除	不支持字符串的这些运算。
不要把报警组与另一类型混合	尝试与另一类型（如整型）的报警组进行比较，或尝试在需要报警组的地方使用其它类型。
不要把字符串与另一类型混合	尝试与另一类型（如整型）的字符串进行比较，或尝试在需要字符串的地方使用其它类型。
无法取反报警组	已使用负号 (-)。
不能否定访问名	“-”或“~”不允许在 DDE 访问名称之前。
无法取反字符串	已使用负号 (-)。
无法取反 TagID	已使用负号 (-)。
无法取反窗口	“-”或“~”不允许在窗口名之前。
不能以此方式使用访问名	在此背景下不能使用 DDE 访问名。
在此种方式下无法使用 HistTrendTag	尝试与另一类型（如整型）的字符串进行比较，或尝试在需要字符串的地方使用其它类型。

错误消息	定义
在此种方式下不能使用 TagID	标记 ID 型变量不能在这种背景下使用。
在此种方式下不能使用窗口名称	窗口名不能在这种背景下使用。
E 格式必须在-38 和+38 之间	最大的“e”格式数在 e-38 和 e+38 之间。
E 格式必须在 E 之后有数字	有效的 E 格式是 n.nnen，1.e 是非法的。
函数参数之后，要求有")"	函数名后需要一个匹配的右括号以对应左括号。
函数名之后，要求有"("	函数名后需要一个左括号。
在 0x 后要求一个数	在 InTouch 中可输入十六进制（以 16 为基数）数。要输入十六进制数，请先写 0x 然后再跟数字。
要求 IF 之后是表达式	缺少离散型表达式。
对函数要求模拟参数	该函数的参数需要一个模拟值。
要求另一个参数	函数需要比当前更多的参数。
要求另一个操作数	如果输入“a +”，InTouch 将显示此错误消息。
要求赋值	在动作脚本中输入标记名，并对下一个逻辑运算赋值。
对函数要求逗号和另外的参数	此函数需要更多的参数。
要求 DLL 名称	在这种背景下必须使用 DLL 名称。
要求 ENDIF	每个 IF 必须有一个 ENDIF。
要求 ENDIF 或 ELSE	每个 IF/THEN 必须同 ENDIF 或 ELSE 相匹配。
等号 (=) 之后期望表达式	在动作脚本中，输入了标记名和赋值符，没有给定具体赋值。如果输入 => 而不是 >=，也会导致此错误。
要求函数名	在这种背景下必须使用函数名。
要求右括号	未找到匹配的“)”。

错误消息	定义
要求名	此参数要求一个标记名。
在语句表中要求名	语句中缺少名称。
要求分号	行的结尾缺少分号。
要求字符串	给定的参数必须是字符串表达式，例如字符串标记名或常量字符串（用双引号 " 括起的文本）
要求 THEN	IF 语句后缺少 THEN。
要求窗口名 - 必须是字符串表达式	给定的参数必须是字符串表达式，例如字符串标记名或常量字符串（用双引号 " 括起的文本）
额外表达式	例如，表达式“a b”不合法，“a + b”合法。
函数仅在动作脚本或逻辑中合法	某些函数仅在脚本中合法，在表达式中不合法。
IF 表达式必须为离散型 (使用 == 勿使用 =)	因为使用赋值符 (=) 而不是比较符 (==) 导致出现此错误。例如，“IF a=b THEN …”应为“IF a==b THEN…”。如果使用了“IF x THEN”而 x 不是离散型标记名，也会导致此错误。
无效或缺少操作数	运算符所需的运算数无效或缺少。
无效占位符名 - 在 ?x 后必须有字符	在占位符名中描述字符后必须跟“?x:”。
无效占位符名 - 第二个字符必须是 d,i,a,r,m,v,g,h,t	占位符名中的第二个描述字符无效。
无效占位符名 - 第三个字符必须是":"	占位符名中的第三个描述字符无效。
逻辑与/或必须使用离散	离散表达式必须使用 AND/OR 运算符。因此如果 x 和 y 是离散型标记名，则“x AND y”正确；如 果是任何其它类型，则会收到此错误消息。
逻辑非必须使用离散	离散表达式必须使用 NOT 运算符。因此如果 x 是离散型标记，则“NOT x 正确；如果 x 是任何其它类型，则会出现此错误消息。

错误消息	定义
最大字符串有 131 个字符	字符串超过允许的最大长度。
必须指定函数的返回值	某些函数需要对返回值进行函数求值。
在小数点后面必须有数字	语法“1.”是非法的。
此参数必须具有历史趋势变量	在这种背景下必须使用历史趋势型变量。
此参数必须具有可写的模拟变量或名称点域。	参数必须是整型或实数变量，或变量的整型或实型值。
此参数必须具有可写的离散变量	此函数的参数必须是一个离散型并且未选定只读的标记名。
此参数必须具有可写的整型变量	此函数的参数必须是一个整型并且未选定只读的标记名。
此参数必须具有可写的消息变量	此函数的参数必须是一个消息型并且没有选定只读的标记名。
此参数必须具有可写的实型变量	此函数的参数必须是一个实型并且没有选定只读的标记名。
此参数必须具有可写的变量	此函数的参数必须是一个未选定只读的标记名。
函数的返回值不能被分配	某些函数不能返回值，并且不能求算其返回值。
名太长	标记名长度必须小于等于 32。
没有右注释符	左注释分隔符 ({) 必须具有匹配的右注释分隔符 (})。
没有右引号	缺少右引号 (")。
显示缓冲区中没有足够的空间	没有足够的内存用于此操作。释放内存将可完成此操作。
表达式缓冲区中没有足够的空间	此时没有足够的内存用于此操作。释放内存将可完成此操作。
0x 后只允许 8 位数	当输入一个十六进制数值时，只允许 8 位数。

错误消息	定义
数字太大	数字的绝对值必须小于 2e38。
数字太小	数字的绝对值必须大于 2e-38。
参数太多	所提供的参数多于此函数所需的数目。
尝试赋值到一个只读名	对只读标记名赋值是非法的。
未定义域名	没有定义点域名，很可能是拼写错误。
不识别字符	辉亮显示的字符对于表达式或动作脚本是非法字符。
为标准标记名使用保留的域名（例如:SP）	标记名不能使用点域名。

窗口控件和分布式报警的错误消息

窗口控件和分布式报警 QuickScript 函数会根据 QuickScript 函数的处理结果来返回值。用于错误诊断的返回值可赋给整型标记名。例如：

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

此处：

ErrorNumber 是保存返回的错误值的**整型**标记名。函数的返回值可以传递给 **wcErrorMessage()**，**wcErrorMessage()** 会返回有关错误的字符串说明。
例如：

```
ErrorMsg = wcErrorMessge(ErrorNumber);
```

此处：

ErrorMsg 是保存返回错误的文本说明的**消息**型标记名。下表列出了这些数值及其说明：

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	指定项目已经存在
-5	未知的对象名
-6	未知的属性名
-x*	未知的错误

* *-x* 表示任何其它数字。

第 7 章

报警/事件

InTouch 的通知系统能通知操作人员有关过程和系统的情况。本系统支持“过程报警”和“系统事件”的显示、记录和打印。过程报警是指过程情况的警告，而系统事件则指系统正常状态下的消息。

InTouch 支持分布式报警系统，分布式报警系统允许显示由本地 InTouch 应用程序及其它网络 InTouch 程序报警系统产生的报警和事件，并可在本地 InTouch 节点或从网络上的远程节点确认报警。

本章描述了报警系统、各种不同类型的报警状态和分组层次结构。各部分分别讲述了如何添加、修改和删除报警组，指定报警组的标记名，定义标记名的报警状态，显示、记录和打印报警，以及如何配置报警系统。

目录

- 简介
- 报警背景知识
- 报警和事件
- 报警摘要与报警历史
- 标准报警子系统、主/从报警与分布式报警子系统
- 分布式报警与 InTouch 旧有标准式报警的比较
- 终端服务报警支持
- 支持其它报警源
- 报警类型
- 报警优先级
- 报警组
- 分布式报警组列表
- 报警确认模型
- 扩展的报警摘要
- 发布/预订机制
- 报警数据存储
- 标记名报警配置
- 报警可见性控制
- 配置报警系统

- 分布式报警显示
- 选择和配置报警查询收藏夹
- 确认本地报警
- 分布式报警属性和函数
- 报警点域
- 从旧 InTouch 标准报警系统移植到分布式报警系统
- 从旧的主/从报警系统移植到分布式报警系统
- Alarm Printer 公用程序
- Alarm DB Logger 公用程序
- 报警数据库视图 ActiveX 控件
- 安装
- 属性
- 方法和事件
- 热备份与同步
- 热备份使用实例
- Alarm DB Purge/Archive 公用程序
- Alarm DB Restore 公用程序
- AlarmSuite 历史迁移实用程序
- 分布式报警数据库视图
- 视图与存储过程列的定义
- 报警历史数据库视图
- 事件历史数据库视图
- 报警与事件历史数据库视图
- 报警计数器数据库存储过程
- 在 Enterprise Manager 中查看存储过程定义
- 事件计数器数据库存储过程
- AlarmSuite 报警记录数据库视图

简介

InTouch 分布式报警系统是独立于 WindowViewer 的一套软件组件。这些组件允许 InTouch 及其它 FactorySuite 程序在多节点系统中互相传递报警信息。充当*报警供应器*的 FactorySuite 程序执行报警状态的实际检测工作，通过 API 调用 将通知传递给分布式报警系统。反之，充当报警接收器的 FactorySuite 程序则通过 API 调用从分布式报警系统获取通知和状态消息，然后执行报警信息的显示。

在 InTouch 之前的版本中，分布式报警系统的重点放在通讯和一组报警接收器身上，这些报警接收器支持基本但却相当灵活的显示和存储功能：在 InTouch 视图窗口中显示报警、报警历史存储、报警记录和报警打印等。增强的分布式报警系统在提高性能的同时，还支持下列功能：

报警停用与禁止

报警显示抑制

报警 SQL 数据库存储

增强的分布式报警系统将继续提供显示、记录、打印和确认过程报警及系统事件的服务；处理由本地 InTouch 应用程序及其它节点上运行的 InTouch 应用程序所产生的报警和事件，而无需识别不同节点上的应用程序。实际上，一旦配置为使用分布式报警系统，甚至可处理非 InTouch 报警供应器的报警。

分布式报警系统的特点还包括：

能显示和确认来自网络上任一 InTouch 节点上的报警。

新的报警显示带有内置滚动条、可调大小的显示列、多项报警选择、更新的状态栏、动态显示类型以及报警优先级显示颜色。

QuickScript 函数能对报警显示和报警确认提供动态控制。

分组机制允许多个报警组跨越不同的应用程序，用一个名称调用它们。

当接到确认后，能对报警添加注释。

支持非 InTouch 报警供应器

在过去，InTouch 一直是 FactorySuite 环境下的主要报警供应器（报警发生器）。从 InTouch 第 7.11 版开始，非 InTouch 报警供应器也将受分布式报警系统的支持 — 象 InBatch 和 InControl、I/O 服务器和报警打印服务器等 FactorySuite 组件，以及其它第三方组件等。

注意：要创建可产生和跟踪报警的第三方软件，必须用到 Wonderware *报警套件*。有关报警套件的详细信息，请联系您当地的 Wonderware 分销商。

分发您的应用程序

应用程序可手动或使用 NAD 系统进行分布。当分发应用程序时，报警组列表文件也会自动分发，如同它是应用程序的一部分。

有关 NAD 的详细信息，请参阅第 3 章“创建分布式应用程序”。

报警背景知识

下面是有关报警的一些通用术语和概念，无论用于何种系统、采用何种应用方式，这些术语和概念都能够适用。

报警：一般而言，**报警**是概念更加广泛的**状态**的一种特定类型 – 具体来说，报警是一种***异常状态***。通常，报警用于在发生某种错误或到达某个特定处理阶段时发出信号。例如，报警可能指示锅炉超出安全温度极限，也可能只是简单地通知换班时间到了。

优先级：**严重程度**或**优先级**与报警关联，用于指明情况有多“糟”，或者条件有多“重要”。当锅炉超出温度极限时，严重程度可能“非常糟”或“非常重要”，需立即引起注意以挽救生命和财产安全。与之相反，当换班时间到达时，严重程度通常“不太坏”或非常轻微。报警的严重程度通常取决于具体环境 – 工厂应用、设备种类、后备系统的可用性、潜在损坏或停机成本等。

注意：InTouch 使用**优先级**标准，其中 1 最重要，999 最不重要。

有关优先级的详细信息，请参阅“报警优先级”。

子状态：报警状态可能还包含**子状态**，在这种情况下称为**多状态报警**。例如，模拟报警通常具有多种极限，例如 "High" 和 "Low" 用于规定正常操作范围，"HiHi" 和 "LoLo" 表示远离正常条件的操作范围。上述锅炉温度水平对于其任一*子状态*，可能处于*报警*状态之下。当继续处于整体*报警*状态时，也可在两个子状态之间转换。

事件：**事件**是***可检测的事务***，它不一定与报警关联。报警状态间的来回*转换*肯定包括一种事件。事件也可能表示操作人员的操作、更改系统配置或某些系统错误。辨别**状态**与**事件**这两个概念非常重要。**状态**可能持续几分钟、几小时、几天或几周，但**事件**是暂时的，一旦发生将立即结束。*报警*是一种状态，而*报警通知*则是一个事件。

确认：报警的一个主要用途是向人员或系统的另一部分发出有关状态的警告。人员或系统应随之***确认***报警，指明已经获悉该警告。这与采取校正措施不同，校正措施可能不会立即执行。确认也与报警状态是否恢复正常无关 – 即使没有任何外部干预，也可能会恢复正常。确认只是表示有人已注意到报警的存在。高或中优先级的报警通常要求予以确认，而优先级非常低的报警则可能不要求确认。虽然产生报警的条件可能消失（例如，过高的温度又降了下来），但在确认之前，报警本身并不视为“已经处理”。

有关确认的详细信息，请参阅“报警确认模型”。

报警组：报警可分成**报警组**，以便跟踪管理。这些报警组可能表示工厂的不同区域、设备部件、操作人员责任或特定功能。报警组也可按父组和子组**层次结构**进行编排。

有关报警的详细信息，请参阅“报警组”。

区域：通常，工厂可划分成不同的**区域**，用于表示实际地点、操作人员责任、流程阶段和设备类型等等。与报警组一样，区域也具有层次结构，可划分成主区域和子区域。然而，考虑到一些工厂会将这两个概念区分对待，所以**区域**应与**报警组**分开使用。分布式报警系统不提供对区域的明显支持，但可使用报警组，将报警划分成与区域相对应的集合。

报警和事件

InTouch 使用两种通知方式来通知操作员有关过程活动的情况：报警与事件。报警是过程状态出现问题时发出的警告，要求操作人员做出响应。当一过程值超出用户自定义的极限时，譬如模拟量超出了上限值，报警即被触发。触发后进入*未确认*报警状态，可以通知操作人员发生了什么问题。一旦操作人员确认报警，系统会返回一*确认*状态。InTouch 可加以配置，要求即使引起报警的条件消失，仍需对报警进行确认。这可以确保操作员获悉那些引起临时报警状态但已恢复正常的事件。

事件表示系统正常状态时的消息，不要求操作人员响应。当某一系统条件满足时，事件即被触发，比如一个操作者登录到 InTouch 中。如果配置了 InTouch，它可以把事件记录到报警数据库中或进行打印输出。

当在标记名字典中定义标记名时，您可以配置任意标记名以监控事件的发生。_当您定义标记名来监控事件时，每次标记名值改变时，事件消息都会记录到报警系统中。事件消息记录数据如何改变：是由操作员、I/O、脚本还是由系统启动数值改变。

有关设置标记名来监控事件的详细信息，请参阅第 4 章“标记名字典”。

报警摘要与报警历史

InTouch 使用***报警摘要***和***报警历史***这两个词，来分别指“当前活动”状态下的报警和“已经结束”的报警。作此划分是考虑到操作员可能需要查看等待确认的所有当前报警屏幕的“摘要”，而所有其它报警信息都是更不紧急的历史记录。在分布式报警系统中，这两种**报警记录**保存在不同的存储区中。

标准报警子系统、主/从报警与分布式报警子系统

InTouch 最早的报警系统称为标准报警子系统，侧重于单个计算机节点上运行的单个应用程序。后来，一些客户希望能够在多个节点上运行应用程序，并使用某些节点作为“控制台”来跟踪其它节点上发出的一些报警。于是 InTouch 增加了主/从报警处理来支持这项应用。然而，该系统结果发现存在一些重大缺陷：第一，不能直接支持远程报警的确认；第二，要求主/从系统中的所有节点运行相同的应用程序。实际上，所有的节点还必须运行相同应用程序的相同版本。这是因为报警记录使用标记名字典中的实际索引来识别哪个标记名与报警相对应，因些这些索引必须与 InTouch 应用程序的所有运行实例完全匹配。分布式报警子系统的设计正是为了克服这些缺陷。实际上，分布式报警子系统与 InTouch 一起运行，因此不需要与标记名字典具备如此紧密的联结。当报警记录报告给分布式报警子系统时，所有相关信息均纳入了报警记录，包括标记名、注释、报警类型、报警限等。因此，不需要回头查阅标记名字典来取得此信息。这使得不同节点可以运行不同版本的 InTouch 应用程序，甚至不同的应用程序。此外，它还支持非 InTouch 的报警供应器和报警接收器。

由于标准报警子系统（主/从报警）实际上是内嵌在 InTouch 中，因些它总是报告那些配置为 ***可报警*** 的 InTouch 标记名的报警信息。对于那些使用该系统来实施报警报告的客户，增加分布式报警子系统后，单个报警会报告两次，并且报告方法也不同。由于这造成了一定程度的混乱，客户可能选择禁用分布式报警系统，而 InTouch 的缺省配置更是将其设为关闭。此外，分布式报警子系统中的许多增强功能也被耽搁，因为它们不能在标准报警和主/从子系统中实施。

因此，与其在两套系统中保留这些增强功能，或者为保证两个系统间的兼容性而限制增强功能的开发，Wonderware 最终还是决定摒弃旧有的标准报警系统。

分布式报警与 InTouch 旧有标准式报警的比较

分布式报警系统是旧有 InTouch 标准式报警系统（第 7.1 或更早的版本）的替代产品。下表从应用程序开发者的角度概括了两个系统之间的主要区别:

功能	标准式报警	分布式报警
确认报警	使用.*Ack* 和 Ack 函数	使用 *ackAlmX* QuickScript 函数或鼠标右键弹出式菜单
报警显示颜色	两种颜色（一种用于已确认的报警，另一种用于未确认的报警）	八种颜色（基于优先级范围和确认状态）
在摘要和历史类型显示之间交替显示报警	需要两种显示，一种作为摘要显示，一种作为历史显示	仅需要一种显示，可以执行多项查询并在运行时进行动态切换
分类报警	使用报警组	使用报警组和报警组列表
改变显示	使用组变量来改变报警组	使用 *almQuery* QuickScript 函数或鼠标右键弹出式菜单
翻页报警显示	PgUp、PgDn	滚动条和 *almMoveWindow* QuickScript
调整列大小	只在开发时	运行时或开发时
选择报警进行确认	不能使用	可通过鼠标操作或 *almSelectX* QuickScript 函数选择报警
报警抑制	不能使用	使用 QuickScript 函数或鼠标右键弹出式菜单

终端服务报警支持

通过配合 Terminal Services for InTouch 使用分布式报警系统，运行在不同终端进程上的报警客户可以选择要显示的报警数据及其显示方式。

报警供应器通过一个唯一标识其应用程序及其应用程序实例的名称加以识别。当报警供应器或报警接收器向分布式报警系统注册后，这个信息将可为分布式报警系统所用。

报警供应器所在的节点则通过一个唯一标识系统中的计算机节点的名称加以识别。当在计算机节点上启动其某个实例后，这个信息将提供给分布式报警系统。

在记录报警事件时，通过节点和完整的报警供应器名称可识别报警来源。

注意：连接到终端服务节点的打印机目前尚不支持报警事件的逐一打印。

此外，报警供应器在终端进程上尚不受支持，它们只在终端操作台受支持。

有关终端服务的详细信息，请参阅您的联机*《Terminal Services for InTouch 布署指南》*。

支持其它报警源

除 InTouch 外，分布式报警系统还支持来自各种来源（也称为“报警供应器”）的报警。特别是在结合 Wonderware I/O 服务器使用时，还能支持来自 RTU 的报警。在以批次模式接收时，分布式报警系统能够按顺序插入“批次”RTU 报警记录。

这些报警记录的日期/时间标签由报警供应器提供，而不是由分布式报警系统自动生成。

客户发出的报警请求，譬如确认，由分布式报警系统发送消息给报警供应器来完成。报警供应器随后对消息进行内部处理，并回复最新的报警记录，指明出现的新状态或配置（如果有的话）。

报警供应器会将其所有特定信息写入报警记录中，报警供应器（而非分布式报警系统）会在报警记录中写入报警名称、报警类别、极限、启用项目、状态及任何其它配置。

有关终端服务的详细信息，请参阅您的联机*《Terminal Services for InTouch 布署指南》*。

报警类型

InTouch 按照特征将报警分为几种通用类别，这些类别称为“*类*”和“*类型*”。分布式报警系统将所有报警分成五种基本*条件*:离散、偏差、变化率、值和 SPC。下表概括了两种系统的报警条件：

报警条件	分布式类	分布式类型
离散	DSC	DSC
偏差 - 主	DEV	MAJDEV
偏差 - 副	DEV	MINDEV
变化率	ROC	ROC
SPC	SPC	SPC
值 - LoLo	VALUE	LOLO
值 - Low	VALUE	LO
值 - High	VALUE	HI
值 - HiHi	VALUE	HIHI

您可以将每种报警与 InTouch 标记名相关联。取决于标记名的类型，您可以为它定义一个或多个报警类或类型。这些配置在标记名字典中进行。

有关定义报警条件的信息，请参阅第 4 章“标记名字典”。

您还可以配置报警打印和（或）分布式报警显示对象，来显示报警*类*字段和（或）报警*类型*字段。

事件类型

InTouch 同样按照特征将事件分为几种通用类别，这些类别称为*事件类型*。下表概括了这些事件类型：

事件	条件
ACK	确认报警
ALM	产生报警
EVT	产生报警事件
RTN	标记名从报警状态返回到正常状态
SYS	产生事件
USER	$Operator 改变
DDE	从 DDE 客户端插入标记名值
LGC	QuickScript 修改标记名值
OPR	操作者使用输入值来修改标记名值

上面的前六个事件会在事件记录启用时自动配置。其余三个必须由用户针对标记名字典中的每个标记名进行定义。

有关事件的详细信息，请参阅“报警和事件”。

报警优先级

InTouch 中配置的每个报警都有一个与之关联的优先值。此值表示报警的严重程度，范围从 “1” 到 “999”，其中 “1” 表示最严重。使用这些优先级创建报警范围，分派报警给每个优先级，用户可以很容易地从非关键性报警中过滤出关键性报警。用户也可以根据优先级范围创建动画链接、确认脚本、过滤查看和打印。

例如，一个加工厂决定采用有四个严重程度层次的报警，他们可以建立如下所示的范围：

报警严重程度	优先级范围
关键	1 - 249
主	250 - 499
副	500 - 749
提示	750 - 999

工厂工程师在创建 InTouch 标记名和报警条件时，通过选择上述范围内的优先级，为每个报警指定一个严重程度。通过这些设置好的范围，工厂的操作人员很容易显示和打印某一特定严重程度的报警。

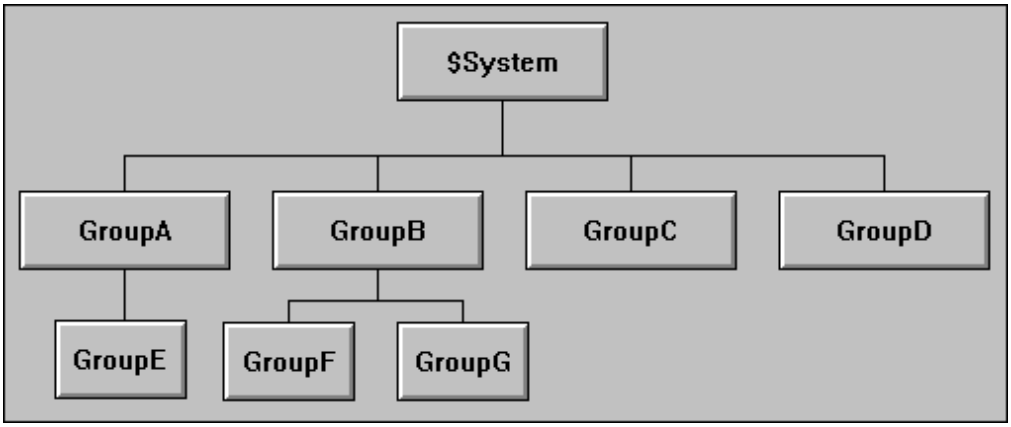
注意：可以而且通常只需使用四种优先级：1 表示关键、2 表示主、3 表示副、4 表示提示。

报警组

每个 InTouch 报警可指定给一个逻辑报警组。这些报警组可由用户自定义，并排列成多达 32 层的层次结构。报警组在组织、工厂设计或其它公共系统的基础上提供一种报警分类方法，对于过滤报警显示器、Alarm Printer 和确认脚本都很有用。

每个标记名都与一个报警组相关联。如果您没有把报警组名与标记名相关联，按缺省 InTouch 会自动地将其与根组 **$System** 相关联。任何报警组都可有关联的标记名和其它报警组名。报警组被组织成分层的树结构，根组 **$System** 位于树的顶端。所有定义的报警组都自动地变成根组的后代。

结构树最多可有 32 层，每一个报警组最多可以有 32 个子组，每一个子组最多可有 32 个亚组等，直至层数达到最大值 32 层为止。



上图仅显示报警组，并不显示每个组内的标记名。树的概念类似于 Windows 的文件夹目录结构，一个目录可以包含其它子目录（类似于组）和文件名（类似于标记名）。

分布式报警系统还使用这些组作为其报警组列表的基础。

有关详细信息，请参阅“分布式报警组列表”。

注意：虽然报警组在购买 InTouch 许可时并不算作标记名，但它们在数据库中却被视为标记名。因此，报警组总数加实际标记名数不能超过 61,405。

警告：报警组数过大 (100/1000) 会影响系统性能。有关详细信息，请参阅联机*《报警布署指南》*。

➢ **创建报警组：**

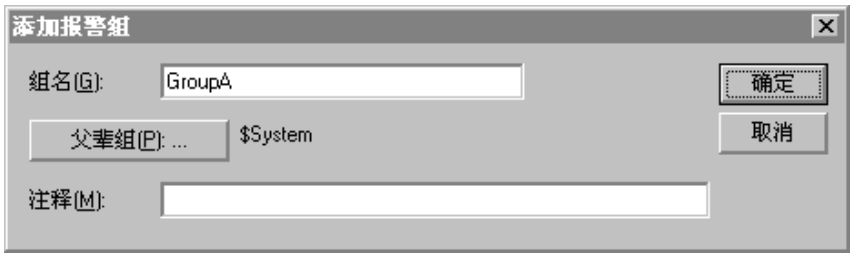
1. 在**“特别”**菜单上，选择**“报警组”**。**“报警组”**对话框出现：

   - 您也可以创建报警组，并在标记名字典中定义标记名时将其与标记名相关联。



2. 单击“**添加**”，“**添加报警组**”对话框出现：

   - “**修改**”和“**删除**”按钮在定义报警组之后才可用。不能修改或删除 **$System** 报警组。

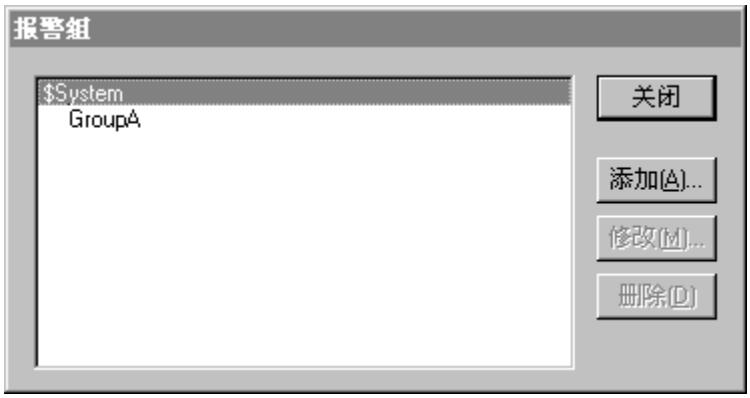


   - 如果右击任何一个报警配置对话框中的文本框，将出现一个菜单，显示可应用到选定文本的命令。

3. 在“**组名**”对话框中，输入新报警组名。因为这是您创建的第一个报警组，它会自动分配到 **$System** 父辈组。

   - 在创建一个报警组之后，它将可用作父辈组。

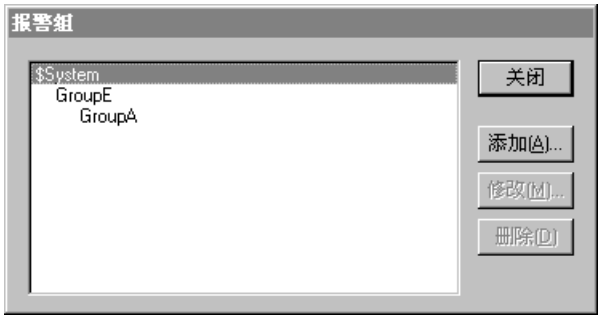
4. 单击**“父辈组”**将报警组分配到一个不同的父辈组。“**报警组选择**”对话框出现：



5. 在“**选择报警组**”列表中，双击您想用作新报警组的父辈组的报警组名称（或选中它，然后单击“**确定**”）。“**添加报警组**”对话框重新出现，显示选中的父辈组。例如：



6. 在“**注释**”框中，给新报警组输入任意注释。
7. 单击“**确定**”。“**报警组**”对话框出现，显示您的报警组分层结构：

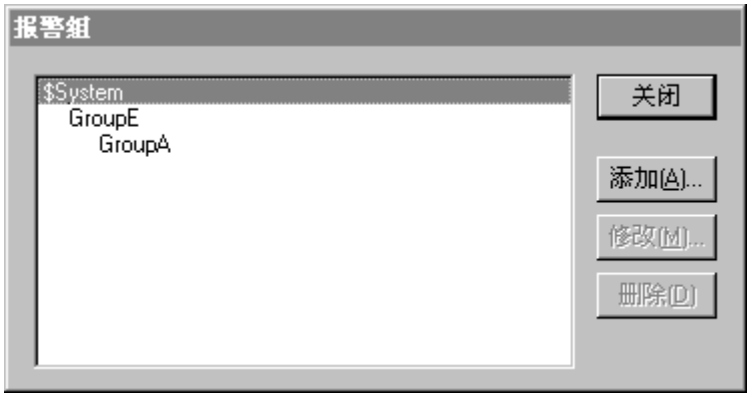


8. 单击“**关闭**”。

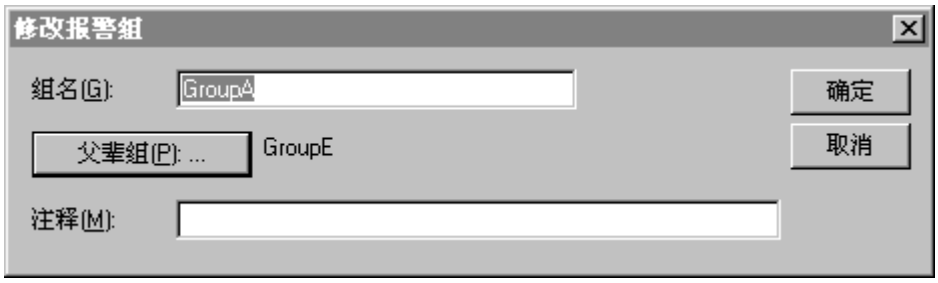
➢ **修改报警组：**

1. 在“**特别**”菜单上，选择“**报警组**”。“**报警组**”对话框出现：

   您也可以在标记名字典中定义标记名时修改报警组。

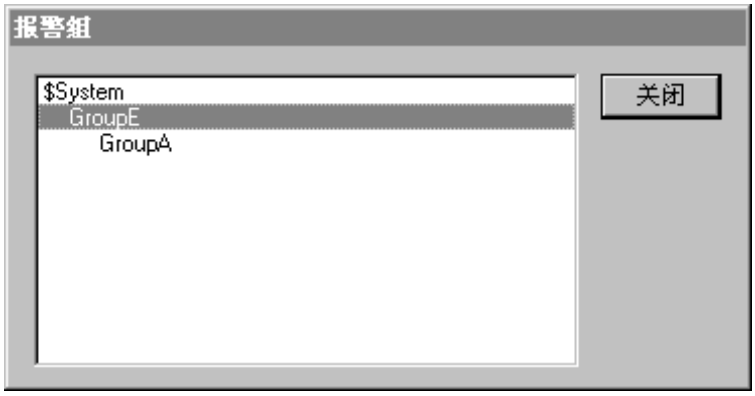


2. 在列表中选中要修改的报警组，然后单击“**修改**”。“**修改报警组**”对话框出现：

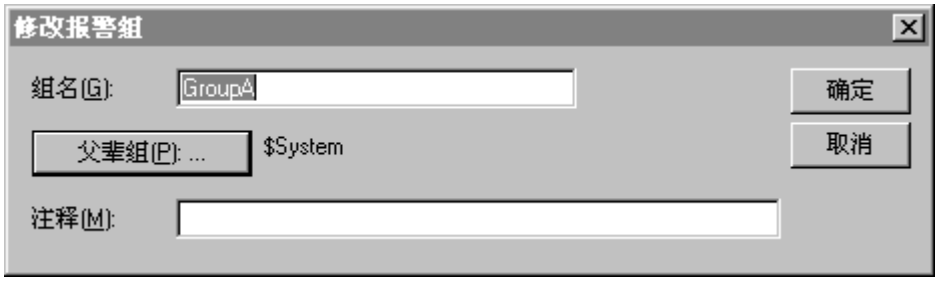


.

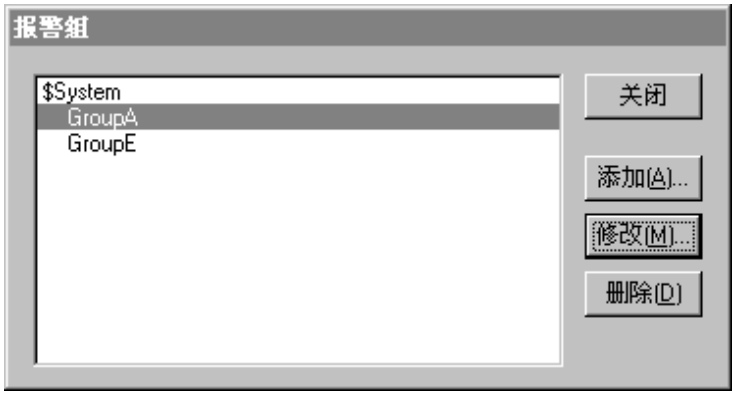
对报警组进行所需的更改。如果您要更改报警组的父辈组，单击“**父辈组**”。“**报警组**”对话框出现：



选择新父辈组，然后单击“关闭”。“修改报警组”对话框重新出现，显示新父辈组。



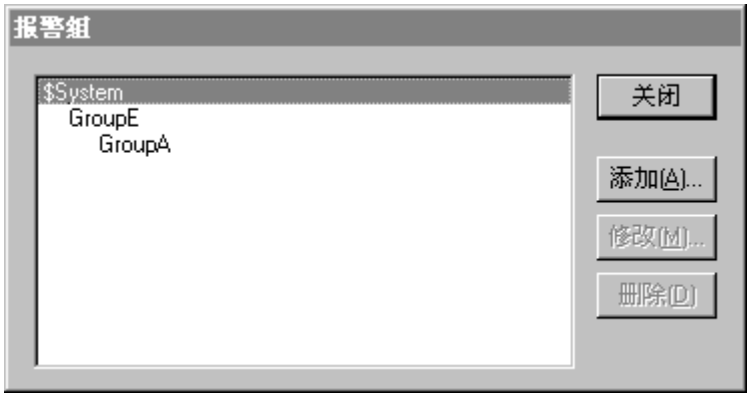
3. 单击“**确定**”。



➢ **删除报警组：**

1. 在“**特别**”菜单上，选择“**报警组**”。“**报警组**”对话框出现：

   您也可以在标记名字典中定义标记名时删除报警组。



2. 在列表中选中要删除的报警组，然后单击“**删除**”。屏幕出现一个要求您确认删除的对话框。单击“**是**”以删除报警组，或单击“**否**”取消删除。

3. 单击“**关闭**”。

分布式报警组列表

分布式报警系统使用报警组机制将报警分成本地分层树结构，分布式报警显示可使用该结构来过滤要显示的报警。分布式报警系统还允许从网络上的多个节点查看报警组。此外，分布式报警系统允许通过创建**报警组列表**来组织报警视图。

报警组列表中包括 InTouch 节点名和每个节点上定义的报警组名，它还可以包含其它报警组列表名和本地报警组。报警接收器，如分布式报警显示器，使用这个列表来查询报警。

例如，如果对跨越几个 InTouch 节点的所有锅炉报警感兴趣，您可以建立一个名为"BoilerAlarms"的查询。附在该查询上的列表将包含与锅炉报警相对应的所有节点上的全部报警组。

➢ **创建报警组列表：**

1. 在“**特别**”菜单上，指向“**配置**”，然后选择“**分布式名称管理器**”。“**分布式名称管理器**”对话框出现，并显示“**分布式报警**”属性页：

   - 要快速访问此对话框，在应用程序浏览器的“**配置**”下，双击“**分布式名称管理器**”。



   - 如果右击任何一个报警配置对话框中的文本框，将出现一个菜单，显示可应用到选定文本的命令。

2. 在“**组属性**”区域，在“**名称**”框内输入查询名。
3. 在“**成员**”框内，输入您要纳入查询的 InTouch 节点和报警组列表。这些列表的有效语法包括：

标准组输入：

\\Node\InTouch!Group	在远程节点上通向报警组的完全合格路径
\InTouch!Group	同上，但节点假设为本地
GroupList	另一个组列表

快捷组输入：

Node.Group	此快捷方式等于 **\\Node\InTouch!Group**
.Group	此快捷方式等于 **\\InTouch!Group**

这里的 **Node** 表示 InTouch 远程节点的名称，而 **.Group** 表示该节点上的报警组。如果报警组为本地，您仅可输入带句点的报警组名。
例如 **.AlarmGroup**。

“**快捷组输入**”为您提供一个简便方法，可以容易地把节点和报警组信息输入对话框。当保存报警组列表时，切记将此信息转换成**标准组输入**格式。

注意：Node.Group 和**.Group** 语法仅可用于此配置对话框，在报警显示配置或任何报警 QuickScript 函数中，它是无效的。

---

4. 单击“**添加**”将此列表加入报警组文件。“**成员**”的语法将自动转换。例如，**.$System** 将转换为 **\InTouch!$System**，如下所示：



如果右击任何一个报警配置对话框中的文本框，将出现一个菜单，显示可应用到选定文本的命令。

5. 单击“**确定**”。

报警确认模型

本章节介绍报警的报告、跟踪和确认方式。

当一个项目从**正常**状态转到**报警**状态时，会产生其报警的一个新*实例*。如果报警属多状态报警，则在仍处于报警状态时改变到新子状态都会被视为相同报警实例的转换。报警实例的*持续时间*在报警恢复**正常**时停止 – 以后再转换到报警状态将产生新的报警实例。

InTouch 分布式报警系统可跟踪报警的每一个*实例*，无论是首次进入报警状态、进行子状态转换、恢复到**正常**状态、等待**确认**，还是**获得确认**时。

面向条件报警：确认是针对报警的*实例*进行的。报警实例在首次进入**报警**状态时，开始等待确认 **(ACK)**。 如果获得确认 **(ACKed)** 并随后转入新的报警子状态（例如从 “Hi” 转入 HiHi”），它开始等待下一次确认 **(ACK)**。每次获得确认 **(ACK)** 时，报警实例都会被接受并应用于迄今出现的报警的所有转换。当最新的实例获得确认 **(ACKed)** 时，报警即被视为已获得确认 **(ACKed)**。

面向事件报警：确认是针对报警的*实例*进行的，并且必须针对进入报警状态或子状态的最近一次转换。报警*实例*在首次进入**报警**状态时，开始等待确认 **(ACK)**。如果获得确认 **(ACKed)** 并随后转入新的报警子状态，它开始等待下一次确认 **(ACK)**。每次后续转换都会被指定一个序号，确认 **(ACK)** 必须在其后附上与其相对应的转换的序号。确认 **(ACK)** 只有在针对最近一次转换做出时才被接受。被拒绝的 **ACK** 可能会被记录用作诊断用途，但不会记入系统中。如果被接受，它将应用于迄今出现的报警的所有转换。当最新的实例获得确认 **(ACKed)** 时，报警即被视为已获得确认 **(ACKed)**。面向事件的模型可确保报警在不同状态间变化时，**ACK** 仍可对应最新的信息。在具有较短等待时间的系统中，它可能看起来象是面向条件的报警 – 但在如互联网等其它环境中，此模型的特性会显得很重要。

扩展的摘要报警：确认是针对报警的每次转换进行的。首次进入报警状态必须获得确认 **(ACKed)**，恢复为正常状态也必须单独进行确认 **(ACKed)**。到新报警状态的任何转换都被视为必须 **ACKed** 的新事件，其 **RTN** 也必须 **ACKed**。子状态转换被视为属于“**RTN 组**”，自项目从先前的***正常***状态首次进入***报警***状态开始。如果项目***恢复正常***，随后再次进入***报警***状态，则会产生新的 **RTN 组**。每次转换均必须单独而明确地进行确认，只有在项目***恢复正常***并且所有待定 **RTN 组** 中的所有转换均***获得确认***时，报警才被视为已确认。

注意：此处使用“摘要”一词来表示“等待确认”的意思。此报警模型在工业中也称为“回铃报警”。

概括地说，在**面向条件**的报警中，确认取决于进行确认时进入***已确认***状态的所有项目。在**面向事件**的报警（如在 OPC 中），确认只有在其针对最近一次“激活”事件时进行时才被接受。在**扩展的摘要**报警中，所有事件 –“活动”、“不活动”以及不同的子状态 – 均必须获得确认，整体报警才被视为已确认。

扩展的报警摘要

当报警出现时，它会在报警显示对象中生成一条记录，表明所出现的报警条件及报警的日期和时间标签。这个记录将保留在显示对象中，直到操作员确认报警并恢复正常状态 (RTN)。如果恢复正常状态在报警确认之前发生，报警对象中会显示两条记录。

例如，如果锅炉温度超出上限状态并触发报警，但在操作员确认报警状态之前已恢复为正常温度范围，则会产生一条记录来显示报警状态，同时产生一条附加记录，显示报警状态未得到确认。

确认只是针对特定转换进行的，无论是转入报警状态、子状态还是恢复正常状态。每次从正常状态的转换均标志着新 RTN 组的开始，RTN 组中的所有转换必须单独获得确认，整体 RTN 组才被视为已确认。

使用扩展的摘要报警

在定义标记名并选择**扩展的摘要**作为其确认模型时，要求操作员确认出现的报警，即使触发报警的状态已恢复正常。确认报警状态会改变报警项目的颜色，但不会改变所显示的时间标签。只有在报警获得确认并且报警已恢复到正常状态时，报警才会从显示对象中被清除。

注意：在定义**扩展的摘要**确认模型的标记名时，“**报警属性**”对话框中的“**返回意味着确认**”选项不得应用于标记名。

有关详细信息，请参阅第 4 章“标记名字典”。

发布/预订机制

在任何给定的节点上，都有**报警供应器**（发布者）与**报警接收器**（预订者）的集合。InTouch 分布式报警系统提供**发布/预订机制**在节点之间和软件组件之间传递报警信息。

报警供应器

报警供应器会跟踪***应报警***项目 – 即可能进入***报警***状态的项目 – 并向分布式报警系统提供这些项目的列表，包括有关项目任何层次结构分组的信息。它也会在项目状态改变时通知分布式报警系统。这些改变包括项目是否处于或脱离***报警***状态，以及最近的报警是否***已被确认***。

此外，报警供应器还会跟踪项目是否被禁用。

注意：在 Windows 2000 中，如果报警供应器与报警接收器处于不同的域中，报警接收器将不能查看报警，除非所用查询均具有供应器设备的完全合格名称或供应器名称的 IP 地址。例如，处于不同域中的报警供应器，可在查询中指定为：\\provider1.b3.wonderware.com\intouch!$system，此处的 "provider1" 是机器名，"b3.wonderware.com" 是域的主 DNS 后缀。

报警接收器

报警接收器向分布式报警系统提供一套**查询**，以寻找其想要收到通知的应报警项目。在被报警接收器更改或移除之前，**查询**将保持活动状态，并指定报警供应器或报警组 – 与使用“通用符”的 SQL 查询非常相似。每当报警供应器发出变化通知时，分布式报警系统会检查报警是否与已注册的查询匹配，然后将更新信息传递到相应的报警接收器。一旦收到更新通知，报警接收器即显示或记录与项目状态或其转换有关的信息。报警接收器也可***确认***报警，即发送确认通知给分布式报警系统，确定报警和报警供应器。此通知将传递给报警供应器，后者随之将项目状态更新为***已确认***（如果适当），并反过来通知分布式报警系统，从而确保更新信息发布到所有相关的报警接收器。

注意：分布式报警系统中的大部分通讯是将**报警查询**和**报警记录**从一个节点发送到另一个节点。在节点内，报警查询和报警记录由**报警缓冲区**记录并缓存，以尽可能地减少网络流量。

报警数据存储

分布式报警系统使用以下几种数据存储形式：

报警高速缓存：有关当前和最新报警的大部份信息都缓存在各个计算机节点的内存中。InTouch 最早的报警系统使用两个高速缓冲存储器：一个用于存储“摘要”（当前）报警，一个用于存储“历史”报警。这种模型也在分布式报警系统中使用。

摘要（当前）报警缓存可根据需要扩展，直至达到可用内存的极限，以容纳所有的当前报警。历史报警缓存只可扩展到 .INI 文件中配置的预定极限。一旦历史缓存达到其极限，则在添加新的报警记录时，旧的报警记录会被丢弃。在多节点环境下，各个节点上的报警缓存共同组成分布式内存数据库。

报警记录：Alarm DB Logger 会创建一个数据库来记录报警发生、子状态转换、确认及恢复正常的时间。实际上，这些数据组成系统中报警的永久或准永久性历史记录。

分布式报警系统支持记录到如 Microsoft SQL Server 或 MSDE 等数据库。这些数据库支持通过开放的非专有接口进行访问，因而可以轻易地检查或分析数据库内容。

注意：最早的 InTouch 标准报警系统使用可由用户配置的格式，记录到一个**平面文本文件**中。后来的标准报警子系统中摒弃了这种记录形式。

分布式报警系统基于**查询**方式，因而支持使用一个计算机节点来记录多个其它节点的报警。

标记名报警配置

InTouch 允许您为标记名字典中定义的每个标记名定义报警配置（缺省条件下，所有标记名都已禁用报警）。其基本思想是通过指定报警类型和用户自选极限，任何标记名都可以配置为***应报警***。对于这些标记名，每当标记名值改变时，**报警逻辑**都会被激活。此报警逻辑是 InTouch 的内部子程序，它通过检查报警类型，将新值与给定极限比较，来决定标记名是否达到***报警***条件。任何状态转换将随后报告给分布式报警系统。

InTouch 中定义了三种基本报警类型（或类），这些类型可细分为七种附加报警子类型：

离散：离散报警对应于离散型标记名。您可以配置*报警*状态是对应于离散型标记名的 TRUE 状态还是 FALSE 状态，以及配置报警的关联*优先级*。离散型标记名的配置对话框如下所示：



模拟：模拟报警对应于整型或实型（浮点）标记名。模拟类型有七种报警类：

值：当前值与一个或多个极限比较。如果值超过极限，则声明报警状态。您可以分别配置 "LoLo"、"Lo"、"Hi"和 "HiHi"' 等极限的值和优先级，并指明是否使用每一个极限。

偏差：当前值与目标值比较，所得差的绝对值与一个或多个极限比较，这些极限以标记名值***范围***的百分比表示 – 即已配置的最大允许值与最小允许值之间的总差值。您可以分别配置“副偏差”极限和“主偏差”极限的值和优先级，并指明是否使用每一个极限。您也可以配置“偏差死区”的值，该值也以标记名范围的百分比表示。此值控制标记名值的百分比必须位于何种范围内才不致引发报警。

变化率：变化率计算使用当前值和先前值，以及当前时间和上次更新时间。如果变化率的绝对值超出极限，则会声明报警状态。您可以配置“变化率”极限的值和优先级，以及是否使用该极限。极限以标记名值范围的百分比表示 – 可以是秒、分钟或小时。模拟型标记名的配置对话框如下所示：



SPC：SPCPro 程序可以产生统计过程控制报警。此类型在 InTouch 之外定义，并在与其它报警不同的软件执行路径下操作。

有关 SPC 报警的详细信息，请参阅联机 *《SPCPro 用户指南》*。

有关定义报警条件的详细信息，请参阅第 4 章“定义标记名报警条件”。

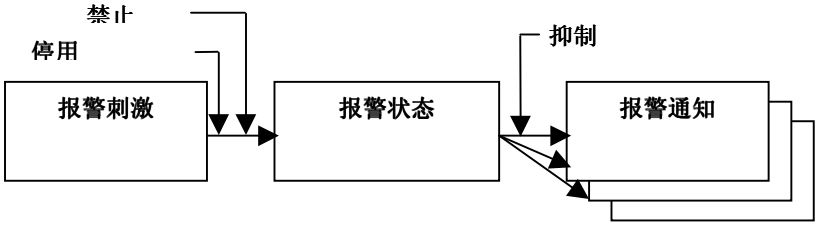
报警可见性控制

出于各种原因，您有时可能需要“关闭”某些报警，但并不实际移除项目的报警配置。InTouch 支持三种基本类型的报警可见性控制：**停用**、**禁止**和**抑制**，它们在下面的三阶段报警模型中介绍。

报警刺激：达到***应报警***项目的值或状态。

报警状态：应报警项目与报警限和条件比较，决定是否处于***报警***状态。状态转换将被记录和报告。

报警通知：报警状态的报告和更新信息通过一个或多个报警客户机***显示***和（或）***记录***。



在这个模型的基础上，报警供应器可处理下列可见性控制类型：

报警停用：报警供应器可**停用**报警，方法是设置**停用**报警的标记（例如状态位），这种方法不涉及报警配置更改。当报警被停用时，可视情形决定是否需要继续检查导致项目进入***报警***状态的条件，但当报警停用生效时，报警条件不能将项目引入***报警***状态。实际上，项目处于“强制正常”状态。

停用报警将中断**报警刺激**与**报警状态**之间的联系。

注意停用报警完全在报警供应器端进行。

所有报警接收器会看到同样的结果。

因为报警不能进入***报警***状态，当报警被禁止时，报警历史中不会记录该报警项目。

报警禁止：**禁止**报警的方法是将标记名标示为禁止，此标记名叫做***约束器标记***。这种方法不涉及报警配置更改。当约束器标记名为 FALSE（零或 NULL）时，将按正常方式处理报警，但当约束器标记名为 TRUE（非零或非 NULL）时，项目将不能报警。其作用实际上与报警**停用**相同。此时，我们称报警被**有效禁止**了。很明显，报警禁止包括两个段：

1. 指定约束器标记名。
2. 将约束器标记名状态从 FALSE 改为 TRUE，或从 TRUE 改为 FALSE。

与停用一样，**禁止**报警将中断**报警刺激**与**报警状态**之间的联系。

注意：将标记名指定为报警的约束器标记名会增加其交叉引用使用次数。

在三阶段报警模型的基础上，报警接收器可处理下列可见性控制类型：

报警抑制：报警接收器可通过设立一套排斥标准来**抑制**一个或多个报警。如果报警与排斥标准匹配，它将不能在报警接收器端看见。这就是说，它不会显示在该特定报警接收器的显示器上，也不能在其上打印或记录（取决于其功能）。报警的实际产生完成不受抑制的影响。实际上，抑制会导致报警接收器忽略某些报警。

抑制报警会中断**报警状态**与**报警通知**之间的联系。

注意抑制报警完全在报警接收器端进行。

每个报警接收器会看到不同的结果。这就是说，报警接收器可以有自己的一套排斥标准，因此可将不同的报警组设为隐藏。实际上，报警抑制是报警查询的扩展。

因为报警的实际产生并不会受到影响，所以即使一个或多个报警接收器抑制报警，该报警的项目仍可记入报警历史（假定 Alarm DB Logger 自身也不抑制报警）。

每次报警接收器更改其抑制标准时，WWLogger 都会进行相应的记录。

注意：如果在 alarmbuf.ini 中添加一行 "SuppressionLog=1"，则 “SuppressionLog=0” 会关闭抑制标准改变时写入 WWLogger 的功能。缺省条件下，此值设为关闭。

运行时的停用和禁止状态转换

InTouch 允许一次禁用或启用标记名的所有报警。此外，对于包含子状态的报警，每个子状态可单独禁用。因此，可以禁用所有子状态，或禁用某些子状态而启用其它子状态。例如，模拟“值”报警可以启用 "Hi" 而禁用"HiHi"。

在停用模式下，对于包含子状态的报警，您也可以为个别子状态单独指定约束器标记名。每个子状态可被不同的标记名禁止，您可以保留不对某些子状态指定约束器标记名。例如，模拟“值”报警的 "Hi" 值可被一个标记名禁止、"HiHi" 值被另一个标记名禁止，而 "Lo" 和 "LoLo" 完全不被禁止。

InTouch 只允许在 WindowMaker 中更改约束器标记名。

在 WindowViewer 中，每当转换导致报警脱离**有效禁止**状态，程序会执行检查逻辑来决定报警供应器是否应将项目引入***报警***状态。

在运行时，报警供应器不会为**停用**的报警或子状态，或**有效禁止**的报警或子状态产生报警。您可以在运行时更改报警的禁用或启用状态。更改约束器标记名的值也可以在运行时进行。然而，指定报警的约束器标记名只能在 WindowMaker 中进行设置或更改。

报警或子状态可单独停用、禁止，或同时被停用和禁止。如果报警当前被停用或有效禁止，报警将被“关闭”。只有在既未被有效禁止又同时被启用时，报警才能够变为活动状态。因此，即使报警被启用，只要它被另一标记有效禁止，则仍不能从报警供应器进入报警接收器。

如果报警或子状态没有指定约束器标记名，则其效果就如同约束器标记名始终设为 FALSE 一样，也就是说，项目永远不能被有效禁止。

每次报警从**禁用**转到**启用**，程序都会执行检查逻辑来决定是否应由报警供应器将项目转入***报警***状态。

如果报警变为**停用**或**有效禁止**，而项目处于***报警***状态，则项目会被强制为不同的（有效）状态。具体的状态则取决于当前何种状态可用，以及该状态是否也被停用。此活动由报警供应器根据报警类型、极限值等进行处理。

被停用或有效禁止的报警不会等待***确认***。如果报警具有子状态，它只能等待***确认***仍可用的子状态。

实例：

作为简单**停用**的一个实例，InTouch 应用工程师可以通过设置控件或脚本来设置标记名的适当点域：.AlarmDisabled 用于禁用报警，.AlarmEnabled 用于启用报警。此点域也可在外部应用程序中进行控制，方法是通过 NetDDE 或 SuiteLink 使用 InTouch 作为 I/O 服务器。

有关报警点域的详细信息，请参阅联机*《Intouch 参考指南》*。

作为**禁止**的一个实例，InTouch 应用工程师可以设置一个名为 "Phase3" 的标记名，该标记名在特定执行阶段会变为 True 值。对于应用程序中的特定报警，工程师可以将 Phase3 指定为约束器标记名。这意味着当 Phase3 为 True 时，报警会被停用，当 Phase3 为 False 时，报警正常工作。

配置报警系统

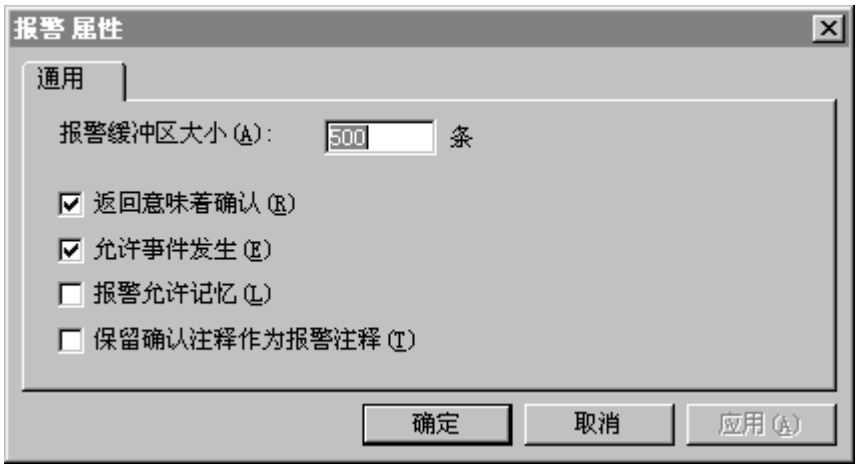
您可以配置报警系统的各种参数，如是否启用事件、当报警恢复正常时是否需要确认等等。

注意：配置对话框与所有标准 Windows 属性页一样，即在单击“**确定**”之前不会记录任何设置。如果单击“**取消**”，将忽略所有输入并关闭对话框。

报警/事件通用属性

➢ **配置报警/事件通用属性：**

1. 在“**特别**”菜单上，指向“**配置**”，然后选择“**报警**”；或者在应用程序浏览器的“**配置**”下，双击“**报警**”。“**报警属性**”对话框出现：



如果右击任何一个报警配置对话框中的文本框，将出现一个弹出式菜单，显示可应用到选定文本的命令。

2. 在“**报警缓冲区大小**”框中，输入您要让 WindowViewer 保存的“内存”报警事件数（节点可为摘要或历史查询存储的最大报警数）。

注意：只有“内存”报警事件才能显示在报警显示对象中。如果未使用报警，此值应设为 1 以节约内存。

如果将此值设置得太高，将会降低系统性能。对于分布式报警系统，建议使用值 500。

---

3. 如果您想要自动确认恢复正常状态 (RTN) 的报警标记名，选择“**返回意味着确认**”。如果要让操作员在恢复正常状态之后确认报警，请不要选择这个选项。

注意：在定义**扩展的摘要**确认模型的标记名时，请不要选取“**返回意味着**
确认”。

---

4. 如果您想打开由操作员、I/O、QuickScripts 或系统初始化的整个数据变化的记录，选择“**允许事件发生**”（只对已选取“**记录事件**”的标记名有效）。

   有关事件的详细信息，请参阅“报警和事件”。

5. 如果您想在 WindowViewer 关闭时保留 **AlarmEnable** 变量的状态，请选择“**报警允许记忆**”。

6. 如果要让自定义注释改写报警注释，选择“**保留确认注释作为报警注释**”。否则，将使用标记名字典中的报警注释。

   有关详细信息，请参阅“在报警确认函数后添加注释”。

7. 单击“**确定**”，保存设置并关闭对话框。

分布式报警显示

分布式报警系统使用单个显示对象来显示本地和远程产生的报警。该显示对象的特点包括：内置滚动条、可调大小的显示列、多项报警选择、更新状态栏、上下文相关右击菜单和基于报警优先级的报警显示颜色。

InTouch 允许您修改报警显示的外观（包括显示的信息）、各种报警状态的颜色、报警组以及显示的报警优先级。

日期	时间	分类	类型	优先级	名称	组	供应器	值	报警限
16 Aug	17:55	Value	HIHI	1	Alarm1	GroupName	Provider	15	14
16 Aug	17:55	Value	HI	250	Alarm2	GroupName	Provider	15	14
16 Aug	17:55	Value	LO	500	Alarm3	GroupName	Provider	15	14
16 Aug	17:55	Value	LOLO	750	Alarm4	GroupName	Provider	15	14
16 Aug	17:55	Dev	Minor	1	Alarm5	GroupName	Provider	15	14
16 Aug	17:55	Dev	Major	250	Alarm6	GroupName	Provider	15	14
16 Aug	17:55	ROC	1	500	Alarm7	GroupName	Provider	15	14
16 Aug	17:55	Custom	1	750	Alarm8	GroupName	Provider	15	14

滚动条

分布式报警显示具有内置的水平和垂直滚动条，可以让您滚动查看列出的报警。您可以配置这些滚动条的显示方式。

下一页/上一页控制

.NextPage 和 **.PrevPage** 报警显示控制属性也同样支持分布式报警显示。

有关详细信息，请参阅“报警显示控制属性”。

可调大小的显示列

分布式报警显示使用网格来放置报警消息。此网格允许动态地改变列宽，只需选择列并将它拖动到需要的列宽就可以了。此功能只在运行时可用。您可以配置能否使用网格来改变列的大小。

网格列的改变不会被保存，因此，如果您在改变网格列后，关闭包含报警显示的窗口，当再打开窗口时，网格列又将以缺省宽度显示。

双击垂直网格线可自动调整列大小。

多项选择

网格允许您在一个列表框中选择一个或多个报警。通过使用本章后面所述的 QuickScript 函数 **almAckSelect()**，能确认所选定的报警。在配置分布式报警显示时，您可以定义选择动作，以允许切换选择（逐项）或多项选择（在按下 Ctrl 或 Shift 的同时单击鼠标来选择多项报警）。您可以关闭运行时选择。

报警消息颜色

根据报警优先级以及是否被确认，您还可以为每个显示报警消息配置最多八种不同的颜色。

更新状态栏

分布式报警显示的状态栏包含三个指示器：状态消息、当前报警查询和进度条。这些指示器提供了显示查询的当前状态概览，以及分布式报警对象中的可用抑制的详细信息。当冻结生效时，状态栏的右窗格呈红色；当抑制生效时，状态栏的左窗格呈红色。当抑制生效时，左窗格中显示“抑制”字样。您可以在运行时关闭状态栏的显示。



功能	描述
状态消息	状态栏左端的状态消息提供当前查询状态的更详尽说明。
进度条	状态栏右端的更新进度条提供当前查询进度的形象化指示。
报警查询	“报警查询”可提供当前报警查询的形象化指示。 **注意**：如果在冻结显示时收到新报警，状态栏中会显示“新报警”字样。

状态消息	状态/指示器	进度条
无	无查询	无
更新未完成	查询未完成	蓝/绿
更新成功	查询结束	实心蓝色
抑制	查询名	实心蓝色
冻结	查询名	红色
新报警	查询名	红色

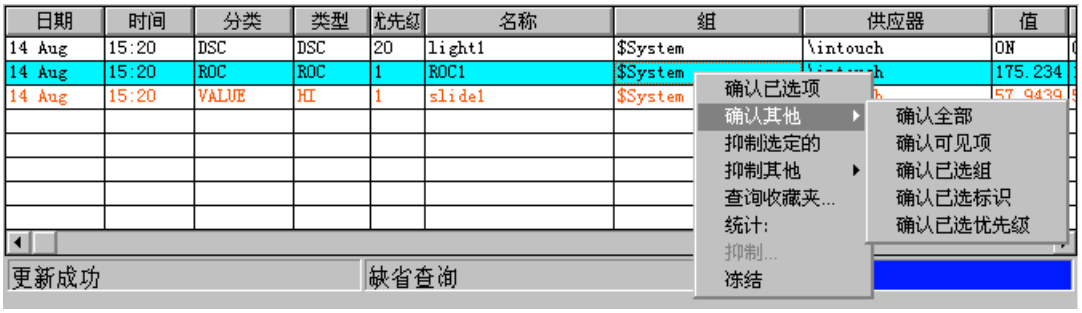
右击菜单

分布式报警显示对象提供了一个右击菜单来快速访问命令，以应用于运行时屏幕中包含的显示对象和（或）一个或多个选定的报警、报警组、标记名和属性。

注意：缺省条件下，右击菜单是被禁用的，但可以通过在属性页中进行相应的设置来启用它。此外，使用右击菜单还可调用脚本函数来执行任何需要的操作。

确认已选项 – 操作员在运行时右击屏幕并指向“**确认已选项**”时，可快速确认所选报警。

确认其他 – 操作员右击屏幕并指向“**确认其他**”，可打开一个包含确认命令的子菜单。在运行时，操作员使用此右击菜单功能，可快速确认屏幕中的所有报警，或仅确认可见报警、已选报警组、已选标记名和已选优先级。



抑制选定的 – 操作员在运行时右击屏幕并指向“**抑制选定的**”，可快速抑制所选报警。

抑制其他 – 操作员右击屏幕并指向“**抑制其他**”，可打开一个包含抑制命令的子菜单。在运行时，操作员使用此右击菜单功能，可快速抑制屏幕中的所有报警，或仅抑制可见报警、已选报警组、已选标记名和已选优级。



查询收藏夹 – 右击屏幕并指向“**查询收藏夹**”，出现“**报警查询**”对话框。

有关详细信息，请参阅“选择和配置报警查询收藏夹”。

统计 – 打开“报警统计”对话框。

抑制 – 打开“报警抑制”对话框。

冻结 – 冻结当前显示内容。

选择和配置报警查询收藏夹

使用分布式报警显示对象右击菜单中的“**查询收藏夹**”命令，您可以从先前定义的报警查询列表中快速选择要显示的报警查询。此外，使用“**查询收藏夹**”命令，您可以创建新的指定查询、编辑现有查询或删除现有查询。

➢ **选择要显示的报警查询：**

1. 在**运行时**右击分布式报警显示对象。
2. 单击“**查询收藏夹**”，“**报警查询**”对话框出现：



3. 选择您要在当前已定义查询列表中显示的指定查询。
4. 单击“**确定**”。分布式报警显示对象现在将显示所选查询名称的报警信息。

有关动态控制分布式报警显示对象的详细信息，请参阅“分布式报警显示属性和函数”。

另请参阅“报警查询 QuickScript 函数”。

- **添加新的指定查询：**

1. 在**运行时**右击分布式报警显示对象。
2. 单击“**查询收藏夹**”，“**报警查询**”对话框出现：



3. 单击“**添加**”，“**添加查询**”对话框出现：



4. 在“**名称**”对话框中，输入您要用于查询的名称。

5. 在“**查询**”输入框中，输入您要执行的 InTouch 报警查询组。您可以指定一个或多个报警供应器和报警组，方法与 Alarm Printer 公用程序的报警查询相同。

   有关详细信息，请参阅“分布式报警显示通用属性”。

6. 在“**从优先级**”字段中，输入最高优先级报警值（1 到 999）。

7. 在“**到优先级**”字段中，输入最低优先级报警值（1 到 999）。

注意：InTouch 中配置的每个报警都有一个与之关联的优先值。此值表示报警的严重程度，范围从“1”到“999”，其中“1”表示最严重。

有关报警优先级的详细信息，请参阅“报警优先级”。

8. 单击“**报警状态**”箭头，选择您要在报警查询中使用的报警状态。

9. 选择您要使用的“**显示类型**”选项。

   有关详细信息，请参阅“报警摘要与报警历史”。

   另请参阅“动态控制显示类型”。

10. 单击“**确定**”，关闭“**添加查询**”对话框。

11. 在“**报警查询**”对话框上单击“**确定**”，确认添加。

- **修改现有的指定查询：**

1. 右击分布式报警显示对象。

2. 单击“**查询收藏夹**”，“**报警查询**”对话框出现。

3. 选择您要在当前已定义查询列表中修改的指定查询。

4. 单击“**修改**”，“**修改查询**”对话框出现：

5. 进行必要的修改，然后单击“**确定**”，关闭“**修改查询**”对话框。

6. 在“**报警查询**”对话框上单击“**确定**”，确认修改。

注意：所做修改不会自动应用于那些使用所修改报警查询的其它分布式报警对象中。

- **删除现有的指定查询：**

1. 右击分布式报警显示对象。

2. 单击“**查询收藏夹**”，“**报警查询**”对话框出现。

3. 选择您要在当前已定义查询列表中删除的指定查询。

4. 单击“**删除**”，出现一个消息框，询问您是否确认要删除所选的指定查询。

5. 单击“**是**”。

6. 在“**报警查询**”对话框上单击“**确定**”，确认删除。

注意：删除不会自动应用于那些使用被删除的报警查询的其它分布式报警对象中。

分布式报警显示准则

分布式报警显示是个复杂的向导。虽然它可能象“标准显示向导”一样出现，但它使用的机制与“窗口控制向导”相同。此机制要求在使用如分布式报警显示等对象时，需要遵循一些准则。这些准则如下：

每项显示都必须有一个标识符以便关联的 QuickScript 函数知道需要修改的显示。在“**报警配置**”对话框中作为“**显示名称**”输入的标识符对于每一个显示必须是唯一的。

显示不应该覆盖在诸如窗口控件或图形对象等其它 InTouch 对象上。通过在 WindowMaker 中单击分布式报警显示，并检查显示“柄”，可以很容易地证实这一点。这些显示柄不应该接触屏幕上的其它图形对象。

应该有节制地使用显示，在一个屏幕上放置大量显示对象会导致系统性能的降低。如有可能，限制屏幕上的显示数量，必要时调用有附加显示（对话框）的额外屏幕。

创建分布式报警显示

➢ **创建分布式报警显示：**

1. 单击“**向导/ActiveX 工具栏**”中的“**向导**”工具，“**向导选择**”对话框出现：



2. 在向导列表中选择“**报警显示**”。
3. 双击“**分布式报警显示**”向导，或选中它然后单击“**确定**”。对话框关闭，窗口重新出现，光标处于“粘贴”模式。
4. 单击窗口，贴入报警显示向导：

日期	时间	分类	类型	优先级	名称	組	供应器	值	报警限
16 Aug	17:55	Value	HIHI	1	Alarm1	GroupName	Provider	15	14
16 Aug	17:55	Value	HI	250	Alarm2	GroupName	Provider	15	14
16 Aug	17:55	Value	LO	500	Alarm3	GroupName	Provider	15	14
16 Aug	17:55	Value	LOLO	750	Alarm4	GroupName	Provider	15	14
16 Aug	17:55	Dev	Minor	1	Alarm5	GroupName	Provider	15	14
16 Aug	17:55	Dev	Major	250	Alarm6	GroupName	Provider	15	14
16 Aug	17:55	ROC	1	500	Alarm7	GroupName	Provider	15	14
16 Aug	17:55	Custom	1	750	Alarm8	GroupName	Provider	15	14

要设置向导大小，指向一个选择柄然后拖拽它直到达到要求的大小。

5. 您现在可以开始配置显示了。

配置分布式报警显示

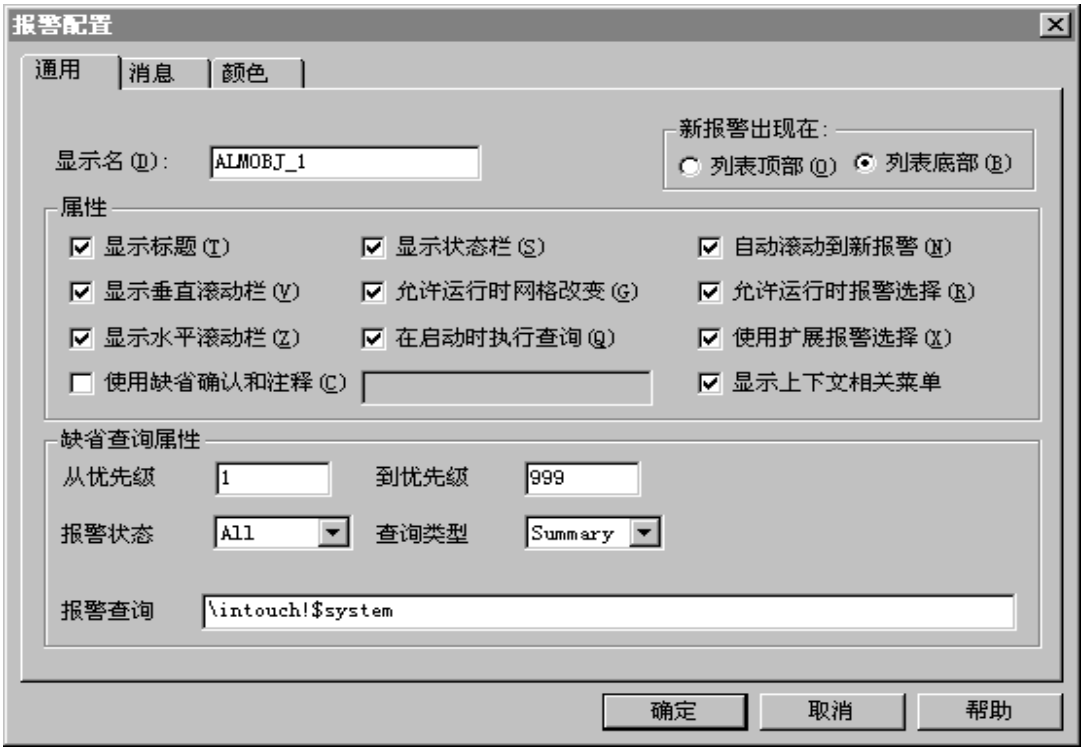
“报警配置”对话框有三个属性页，包括**“通用”**、**“消息”**和**“颜色”**设置选项。

注意：配置对话框与所有标准 Windows 属性页一样，即在单击**“确定”**之前不会记录任何设置。系统将核实选项输入内容是否正确，当您从一个属性页（标签）切换到另一个属性页时，如果输入内容验证失败，包含失败输入内容的属性页将返回到屏幕最上端，并且出现消息框来指示错误。如果单击**“取消”**，将忽略所有输入并关闭对话框。

分布式报警显示通用属性

➢ **配置分布式报警显示：**

1. 双击分布式报警显示，或右击它然后单击**“属性”**。**“报警配置”**对话框出现，并显示**“通用”**属性页：



如果右击任何一个报警配置对话框中的文本框，将出现一个菜单，显示可应用到选定文本的命令。

2. 在“**显示名**”框内，键入报警显示的名称。对于使用的每个报警，此名称均必须是唯一的。

   此处键入的名称将作为此对象的引用，用于整个系统中，用于执行诸如报警确认查询等任务。

3. 选择“**新报警出现在**”选项，设置新报警在对象中的显示位置：

   列表顶部 在列表顶部显示最新的报警。

   列表底部 在列表底部显示最新的报警。

4. 如下所述选择**属性**：

属性	描述
显示标题	显示报警消息标题栏。
显示垂直滚动条	显示垂直滚动条。
显示水平滚动条	显示水平滚动条。
使用缺省确认注释	如果选定此框并输入了字符串，则当用户确认报警而未输入注释时，此字符串将用作缺省注释。
显示状态栏	显示状态栏。
允许运行时网格改变	允许用户在运行时改变列设置。
在启动时执行查询	如果选定此框，将使用缺省查询属性来自动开始更新显示；如果未选定，您需要在显示更新以前，执行 **almDefQuery** 或 **almQuery**。
自动滚动到新报警	如果用户从开始处滚动列表，将自动跳到新报警（这里的新报警是指那些显示对象中当前没有显示的报警）。
允许运行时报警选择	允许用户在运行时选择报警。
使用扩展报警选择	按住 Ctrl 或 Shift 键并配合鼠标可选择多项报警。缺省情况是通过单击来切换选择报警（只有在选定“**允许运行时报警选择**”复选框时才可以看到）。
显示上下文相关菜单	激活右击弹出式菜单。

5. 如下所述选择“**缺省查询属性**”选项：

   当您选中“**启动时执行查询**”选项，或者执行 **almDefQuery** QuickScript 函数时，将使用“**缺省查询属性**”。

属性	描述
从优先级	缺省最小报警优先级。
到优先级	缺省最大报警优先级。有关报警优先级的详细信息，参见本章开始部分的“报警优先级”。
报警状态	缺省报警状态查询（全部、未确认、确认）。
查询类型	设置“摘要”或“历史”显示类型。
报警查询	设置初始报警查询。此字段只接受文本，不接受标记。这些列表的有效语法包括： **\\Node\InTouch!Group** 报警组的完整路径 **\InTouch!Group** 本地报警组的完整路径 **GroupList** 另一个组列表

注意：要执行多项查询，用空格将每项查询分开。

例如：**\\Master\InTouch!MyGroup LocalGroupList**

分布式报警消息格式

分布式报警显示对象中显示的信息是可配置的。这包括您要显示的信息，以及在某些情况下，显示项目包含的字符数等等。

注意：消息中项目的出现顺序是固定的，不能改变。

➢ **配置报警显示信息格式：**

1. 在“**特殊**”菜单上，指向“**配置**”，选择“**报警**”，然后选择“**属性**”。“**报警配置**”对话框出现。
2. 单击“**消息**”标签，激活“**消息**”属性页：



☜ 如果右击任何一个报警配置对话框中的文本框，将出现一个菜单，显示可应用到选定文本的命令。

预览区（对话框底部）会显示当前配置的报警消息的一个实例。这个实例使用所选字体来显示信息，但不包括颜色。

3. 如果您要在报警信息中显示日期，请选择“**日期**”，然后单击箭头选择日期格式。可选格式为：

选择	显示	选择	显示
DD MMM	28 Feb	**MM/DD**	02/28
DD MM YYYY	28 Feb 1997	**MM/DD/YY**	02/28/97
DD/MM	28/02	**MMM DD**	Feb 28
DD/MM/YY	28/02/97	**MMM DD YYYY**	Feb 28 1997

4. 如果您要在报警信息中显示时间，请选择“**时间**”，然后单击箭头选择时间格式。此字段中的值将用作规定时间格式的模板。例如，要规定时间为 **10:24:30 AM**，用 **HH:MM:SS AP**。模板字符如下：

   AP 选择 AM/PM 格式。例如下午三点钟显示为 3:00 PM。非此格式的时间缺省使用 24 小时军用时间格式。例如下午三点钟显示为 15:00。

   HH 显示发生报警/事件的小时。

   MM 显示发生报警/事件的分钟。

   SS 显示发生报警/事件的秒。

   SSS 显示发生报警/事件的毫秒。

5. 在“**时间**”下面的排序框内，选择用于对报警对象中的报警排序的顺序。共有三个选项：

   OAT 原始报警时间 – 即报警发生的日期/时间标签。

   LCT 上次更改时间 – 即报警实例最近一次状态改变的日期/时间标签，包括报警发生、子状态改变、恢复正常或确认。

   LCT 但确认为 OAT 上次更改时间，但确认时使用原始报警时间 – 即在报警未确认时使用 LCT，但在报警获得确认时使用 OAT。

6. 如果您要显示报警状态，选择“**报警状态**”**(UnAck,Ack)**。
7. 如果您要显示报警类别，选择“**报警分类**” **(VALUE.DEV, ROC..)**。
8. 如果您要显示报警类型，选择“**报警类型**” **(HIHI,LO,MAJDEV,…)**。

   有关可用报警类型的详细信息，请参阅“报警类型”。

9. 如果您要显示与报警条件关联的操作员名字，选择“**操作员**”。在“长度”字段中输入一个值，以控制显示字符的数目（最多 16 个字符）。
10. 如果您要显示报警优先级，选择“**优先级**”。
11. 如果您要显示标记名的注释，选择“注释”。这些注释是在数据库中定义标记名时输入“注释”框的。在“长度”框中，输入您要显示的字符数（最多 50 个字符）。
12. 单击“**选择显示字体**”以访问“**字体**”对话框，改变报警显示中使用的字体、样式和大小。
13. 如果您要显示报警/标记名，选择“**报警名**”。在“**长度**”框中，输入您

要显示的字符数（最多 64 个字符）。

14. 如果您要显示报警组名，选择**“组名”**。在**“长度”**框中，输入您要显示的字符数（最多 64 个字符）。

15. 如果您要显示报警供应器名称，选择**“报警供应器”**。在**“长度”**框中，输入您要显示的字符数（最多 64 个字符）。

16. 如果您要在报警发生时显示标记名值，选择**“报警值”**。在**“长度”**框中，输入您要显示的字符数。

    输入的值应该足够大以满足精度要求（最多 32 个字符）。

17. 如果您要显示标记名的报警极限值，选择**“极限”**。在**“长度”**框中，输入您要显示的字符数。

    此字段值应该足够大以满足精度要求（最多 32 个字符）。

18. 单击**“确定”**。

分布式报警显示颜色属性

➢ **配置报警显示颜色：**

1. 双击分布式报警显示对象，或右击它然后单击“**属性**”。“**报警配置**”对话框将出现。
2. 单击“**颜色**”标签，激活“**颜色**”属性页：



如果右击任何一个报警配置对话框中的文本框，将出现一个菜单，显示可应用到选定文本的命令。

3. 在“**显示**”组的顶部，单击每个颜色框以打开 InTouch 调色板。在调色板上单击您要用于下列各种情况的颜色：

选项	描述
窗口	设置显示背景颜色。
网格	设置显示网格颜色。
选择背景	设置辉亮文本背景颜色。
选择文本	设置辉亮文本颜色。
标题栏背景	设置标题栏背景颜色（仅当“显示标题”选项打开时才可见）。
标题栏文本	设置标题栏文本颜色（仅当“显示标题”选项打开时才可见）。
报警返回	设置返回报警颜色（未经确认返回到正常状态的报警）。
事件	设置事件报警颜色。

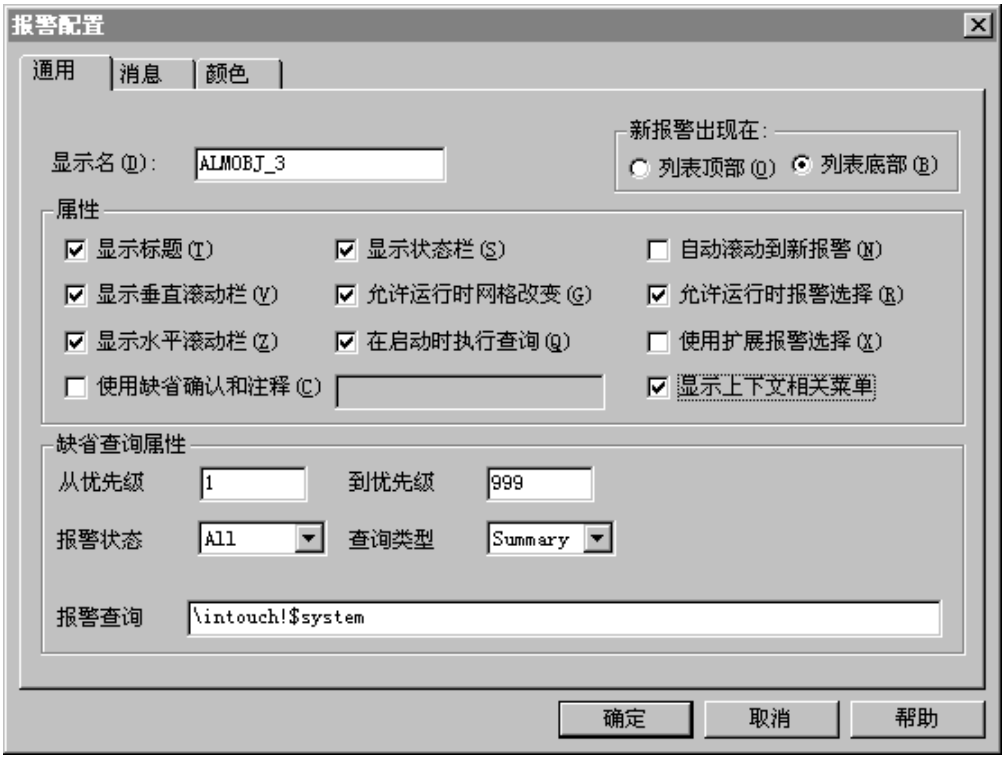
4. 在“**报警优先级**”框内，键入用于报警显示的断点值。
5. 单击“**未确认报警**”和“**确认报警**”颜色框以打开 InTouch 调色板。单击调色板中您要使用的颜色。
6. 单击“**确定**”。

动态控制显示类型

分布式报警显示可以显示活动报警的摘要或历史报警的列表。您可以动态地改变是显示报警摘要，还是显示报警历史。

➢ **改变分布式报警对象查询缺省值：**

1. 双击分布式报警对象，或右击它然后单击“**属性**”。“**报警配置**”对话框出现，并显示“**通用**”属性页：



☞ 如果右击任何一个报警配置对话框中的文本框，将出现一个菜单，显示可应用到选定文本的命令。

2. 单击“**查询类型**”箭头并选择您要用于运行时缺省的报警显示类型。

   例如，在运行时，显示类型由您使用的查询 QuickScript 函数决定。如果针对 *AlarmObj_2* 显示对象，您运行 *almQuery( )* QuickScript，*类型*参数设置为“摘要”，则显示对象将显示当前报警的摘要信息。相反，针对同样的显示对象，如果在运行 *almQuery()* 时使用*类型*参数设为“历史”，那么将显示历史报警。属性 *QueryType* 反映了报警显示的当前状态。

3. 单击“**确定**”。

在报警确认函数后添加注释

您可以在每个报警确认函数后面添加注释。确认报警的操作员可以使用此注释来添加有关报警的信息。InTouch 可以将这些报警存储在报警数据库中并打印出来。分布式报警系统在数据库中使用标记名报警定义的“**报警注释**”框，将这些注释传递到记录和打印子系统。

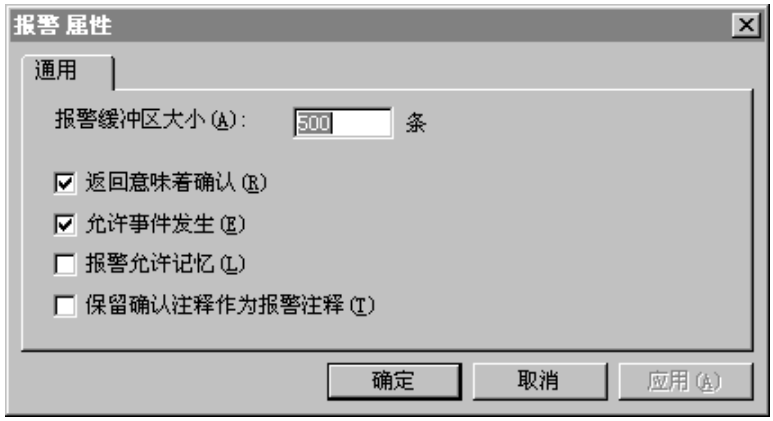
您可以选择是否允许报警系统使用标记名“**报警注释**”框来添加确认注释。如果使用标记名“**报警注释**”框来添加确认注释，则该字段中的任何注释都会在运行时被改写（这不会影响标记名数据字典）。

注意：要允许操作员在运行时动态地输入注释，在 INTOUCH.INI 文件中添加 **`CommentRetentive=1`**。此注释属于“视图”进程间的标记名。

警告： 结合 **CommentRententive=1** 使用 "**almAck**" 会导致传递给函数的任何注释都写入标记名字典。

➢ **使用报警确认注释：**

1. 在“**特别**”菜单上，指向“**配置**”，然后选择“**报警**”。“**报警属性**”对话框出现，并显示“**通用**”属性页：



   - 如果右击任何一个报警配置对话框中的文本框，将出现一个菜单，显示可应用到选定文本的命令。

2. 如果您要使用各种 **almAck()** 函数用于自定义注释，选择“**保留确认注释作为报警注释**”。否则，将使用标记名字典中的报警注释。

   有关报警注释的详细信息，请参阅第 4 章中的“标记名报警注释”。

3. 单击“**确定**”。

使用分布式显示监视本地报警

分布式报警显示可用于显示和确认本地和远程报警。

➢ **建立仅用于监视本地报警的显示：**

1. 将分布式报警显示对象粘贴到窗口（单击向导工具，选择“**报警显示**”类别，双击“**分布式报警显示**”向导，然后单击窗口将其粘贴到窗口中）。
2. 双击显示对象，或右击它然后单击“**属性**”。“**报警配置**”对话框将出现，并显示“**通用**”属性页。
3. 在“**报警查询**”对话框中，输入 **\InTouch!$System**。

   您可以将 **$System** 替换成任何有效的报警组，也可以定义一个仅包含 **\InTouch!$System** 的报警组列表，然后在第三步中使用该组列表，而不使用直接引用。
4. 针对显示类型和程序需要的过滤设置，配置**“缺省查询属性”**的其它参数。
5. 切换到 WindowViewer 来运行应用程序。

显示报警统计

分布式报警系统提供一个内置的报警统计对话框。应用程序开发者可设计应用程序来调用“**报警统计**”对话框，以列出特定报警显示的当前查询状态。此对话框也可以从右击菜单中选择“**统计**”来调用。

有关详细信息，请参阅“选择和配置报警查询收藏夹”。

“**报警统计**”对话框为您提供特定报警显示的当前报警查询的概述，它列出了实际报警供应器的请求和每个请求的结果。需要特别注意的是，即使您可能只请求一个报警组列表名，该列表名仍可能等于几个单一报警供应器的查询。例如：



- 对话框中的每一行都列有数字和查询。数字代表已返回的查询百分数，对话框提供查询结果的静态显示。

➢ **更新查询列表中检索到的报警百分比：**

1. 右击分布式报警对象。
2. 从子菜单中选择“**统计**”。
3. 选择您要更新的报警查询，然后单击“**更新**”。
4. 单击“**确定**”关闭对话框。

确认本地报警

要确认本地报警，您可以在动作或键 QuickScript 中使用 **.Ack**（**点域**），或者在分布式报警对象上使用上下文相关右击菜单。

注意：扩展的摘要报警不建议使用。

➢ **创建本地报警确认按钮：**

1. 创建一个三维按钮或任何其它对象，使动作、键 QuickScript 脚本可以与其链接。
2. 双击对象，或选择它然后在“**特别**”菜单上，选择“**动画链接**”。
3. 在“动画链接”选择对话框的“**触动按钮**”区域，单击“**动作**”。QuickScript 编辑器出现。

   输入下列任一 QuickScript 语句：

语句	描述
Ack $System;	确认系统中的所有本地报警。
Ack *Group Name*;	确认特定报警组中的所有本地报警。
Ack *Tagname*;	确认特定标记名的报警。
$System.Ack=l;	确认系统中的所有本地报警。
***Group Name*.Ack=l;**	确认特定报警组中的所有本地报警。
***Tagname*.Ack=l;**	确认特定标记名的报警。
***Tagname*.AckDev=1;**	确认特定标记名的偏差报警。
***Tagname*.AckROC=1;**	确认特定标记名的变化率报警。
***Tagname*.AckValue=1;**	确认特定标记名的值报警。

4. 单击“**确定**”。

有关 QuickScript 编辑器及其特性的详细信息，请参阅第 6 章“在 InTouch 中创建 QuickScript”。

分布式报警属性和函数

分布式报警系统包括多个标记名**点域**和 QuickScript 函数。下面的部分将简单介绍如何使用 QuickScript 函数。

有关如何使用 QuickScript 函数的详细信息和实例，请参阅您的*《InTouch 参考指南》*。

报警显示监视属性

报警显示具有几个陈述 QuickScript 的属性，可用于在运行时监视显示状态。这些属性可通过 *GetPropertyX( )* 函数进行访问，这里的 X 指数据类型（D 表示离散型，I 表示整型，M 表示消息型）。

查询属性	描述
.AlarmGroup	包含当前查询列表的消息型属性。
.PriFrom	包含当前查询优先级低过滤值的整型属性。
.PriTo	包含当前查询优先级高过滤值的整型属性。
.QueryType	包含当前查询类型的整型属性： **1 = 历史** **2 = 摘要**
.QueryState	包含当类查询过滤的整型属性： **0 = 全部** **1 = 未确认** **2 = 确认**

有关详细信息，请参阅“动态控制显示类型”。

查询状态优先级	描述
.Successful	包含当前查询状态的离散型属性： **0 = 错误** **1 = 确认**
.ProvidersReq	包含当前查询中的报警供应器数的整型属性。
.ProvidersRet	包含成功返回查询结果的报警供应器数的整型属性。
.NumAlarms	包含当前查询中的报警数的整型属性。
.PageNum	包含报警显示中显示的当前页码的整型属性。
.TotalPages	包含报警显示中的总页数的整型属性。

报警显示控制属性

报警显示也有两个陈述 QuickScript 的属性，可用于控制**运行时**显示屏幕的移动。这些属性通过 *SetPropertyD* 函数来控制。

控制属性	描述
.NextPage	当此属性从 0 转变为 1 时，报警显示向下滚动一页。
.PrevPage	当此属性从 0 转变为 1 时，报警显示向上滚动一页。

每当此离散型变量值从“关”（0 或 False）变为“开”（1 或 Ture）时，报警显示对象都会显示与该 QuickScript 相对应的页（下一页或上一页）。一旦显示完该页，离散型变量将自动复位为“关”（0 或 False）。

注意：这些函数的提供简化了从先前 InTouch 标准显示到当前分布式显示的转换过程，其功能已被滚动条和 *almMoveWindow* QuickScript 函数所代替。

报警确认 QuickScript 函数

分布式报警系统能够确认任一可查询的报警（仅限于摘要显示）。为提供此能力，分布式报警系统加入了报警确认 QuickScript 函数，这些函数补充了 InTouch 报警系统用于确认本地报警和报警组的 **.Ack** 点域。有关这些函数的特定语法在 *《InTouch 参考指南》* 中介绍。

函数	描述
almAckAll	确认当前报警查询中的所有报警。由于报警显示区域有限，所以 **almAckAll** 函数可能确认显示中不可见的报警。
almAckDisplay	仅确认报警显示中当前可见的那些报警。
almAckGroup	确认来自相同报警供应器的具有给定组名的所有报警。
almAckPriority	确认具有相同报警供应器名和组名的指定优先级范围内的所有报警。
almAckRecent	仅确认当前报警查询中最近发生的报警。
almAckSelect	分布式报警显示允许在运行时单击鼠标来选择报警。**AlmAckSelect** 函数可用于确认这些报警。
almAckSelectedGroup	确认来自相同报警供应器的与一个或多个所选报警具有相同组名的所有报警。
almAckSelectedPriority	确认来自相同报警供应器和组名的与一个或多个所选报警具有相同优先级的所有报警。
almAckSelectedTag	确认具有来自相同报警供应器和组名的相同标记名，并且与一个或多个所选报警具有相同优先级的所有报警。
almAckTag	确认给定优先级范围内具有相同标记名、报警供应器名和组名的所有报警

报警抑制 QuickScript 函数

分布式报警系统能够通过确立一套排斥标准，在报警接收器端抑制一个或多个报警。如果报警与排斥标准匹配，它将不能在报警接收器端看见，不会出现在显示器中、不能打印，也不能在特定报警接收器端记录。为提供此能力，分布式报警系统加入了报警抑制 QuickScript 函数，有关这些函数的特定语法在*《InTouch 参考指南》*中详细介绍。

函数	描述
almSuppressAll	抑制所有当前活动报警的所有当前和未来显示。
almSuppressSelected	抑制所选报警的当前和未来显示。
almSuppressDisplay	抑制任何可见报警的当前和未来显示。
almSuppressSelectedGroup	抑制与同名报警供应器内的一个或多个所选报警处于相同报警组的任何报警的当前和未来显示。
almSuppressSelectedPriority	抑制与同名报警供应器内的一个或多个所选报警具有相同优先级的任何报警的当前和未来显示。
almSuppressSelectedTag	抑制与同名报警供应器内的一个或多个所选报警具有相同标记名的任何报警的当前和未来显示。
almSuppressGroup	抑制具有相同报警供应器名的、给定组名内的任何报警的当前和未来显示。
almSuppressPriority	抑制具有相同报警供应器名和组名的、给定优先级范围内的任何报警的当前和未来显示。
almSuppressTag	抑制具有相同报警供应器名、组名和优先级范围的、由给定标记名发出的任何报警的当前和未来显示。
almUnSuppressAll	清除报警抑制。
almSuppressRetain	保持报警查询间的报警抑制。

报警显示控制 QuickScript 函数

分布式报警系统提供了若干个 QuickScript 函数来处理显示对象。这些函数允许移动显示窗口、在显示屏幕内选择报警、显示已选报警数，以及显示统计窗口。

有关这些函数的特定语法在联机*《InTouch 参考指南》*中介绍。

函数	描述
almMoveWindow	提供用于控制显示窗口的命令，这些命令包括：向上翻页 (Page Up)、向下翻页 (Page Down)、向右滚动 (Scroll Right)、向左滚动 (Scroll Left)、向上一行 (Line Up)、向下一行 (Line Down)、顶端 (Top)、末端 (End) 等。
AlmSelectAll	切换选择显示中的所有报警。由于报警显示区域有限，所以 **almSelectAll** 函数可能选择显示中不可见的报警。
almSelectionCount	返回包含分布式报警对象中操作员所选报警数的整型值。
AlmSelectGroup	切换选择给定报警供应器内的给定报警组的所有报警。
AlmSelectItem	切换选择报警显示中辉亮显示的项目。
almSelectPriority	切换选择给定报警供应器、给定报警组、给定优先级范围内的所有报警。
AlmShowStats	显示报警统计对话框。
AlmSelectTag	切换选择特定报警供应器/组/标记内的所有报警。您也可以指定优先级范围，或使用 1-999。
AlmUnSelectAll	取消选择所有选定的记录。

报警查询 QuickScript 函数

分布式显示通过提交报警查询来检索报警信息。此查询参数和查询类型由三个 QuickScript 函数之一指定。有关这些函数的特定语法在*《InTouch 参考指南》*中详细介绍。

函数	描述
almDefQuery	用配置对话缺省属性执行查询。这些属性包括：从优先级、到优先级、报警列表和显示类型。缺省属性只能在开发时改变，并且不能被其它报警查询改写。
almQuery	执行摘要或历史报警信息查询。本函数提供了所有查询属性。
almSetQueryByName	使用与用户自定义查询名关联的查询参数启动新报警查询。

注意：在 Windows 2000 中，如果报警供应器与报警接收器处于不同的域中，报警接收器将不能查看报警，除非所用查询均具有供应器设备的完全合格名称或供应器名称的 IP 地址。例如，处于不同域中的报警供应器，可在查询中指定为：\\provider1.b3.wonderware.com\intouch!$system，此处的 "provider1" 是机器名，"b3.wonderware.com" 是域的主 DNS 后缀。

报警点域

InTouch 提供了各种报警“**点域**”来动态控制和（或）监视各种报警状态。许多这样的**点域**可通过 I/O 表达式和（或）脚本进行访问。I/O 访问提供了利用其它 Windows 应用程序来监视和（或）控制特定标记名报警信息的能力，例如，使用 Excel 或远程查看应用程序（本章稍后介绍）。

例如，如果您创建一个名为 "**Analog_Tagname**" 的模拟报警标记名，它将具有与其相关的“属性”，如名称、**HiHi** 设置点等等。其中一些属性可以通过逻辑脚本、表达式和用户输入来访问，它们被称为**点域**。

访问与标记名关联的点域需要使用语法：**Tagname.field**。例如，如果您要在运行时改变 **Analog_Tagname** 上的 **HiHi** 报警限，您可以创建一个应用于按钮的“**模拟 – 用户输入**”触动链接，**Analog_Tagname.HiHiLimit** 将作为链接对话框中的表达式输入。在运行时，操作员只需单击按钮，即可为 **Analog_Tagname** 的 **HiHi** 报警限输入一个新值。

下面简单介绍了有关如何使用报警点域的实例。

点域	描述
.Ack	读/写用于控制/监视标记名和报警组的报警确认状态的离散型标记名点域。 **.Ack** 有一个相反的标记名点域叫 **.Unack**。当出现未确认报警时，**.Unack** 将置为 1，**.Unack** 随后可用于动画链接或条件脚本中，针对未确认报警触发电铃。
.AckDev	监视/控制标记名上活动偏差类型报警的报警确认状态。
.AckDsc	监视/控制离散型标记名的当前确认状态。
.AckROC	监视/控制标记名上活动变化率类型报警的报警确认状态。
.AckValue	监视标记名上活动值类型报警的报警确认状态。
.Alarm	发出存在报警的信号。
.AlarmAccess	返回与所选报警关联的标记名。必须通过单击分布式报警摘要显示来选定报警。

点域	描述
.AlarmAckModel	如下监视与标记名关联的**确认模型**： 0=条件（缺省） 1=事件 2=扩充 应用于含报警的离散或模拟型标记名；只读属性，但可在 WindowMaker 中配置。
.AlarmClass	返回当前报警的类。
.AlarmComment	读/写用于描述报警内容而非标记名内容的文本字符串。在新的应用程序中缺省条件下为空白。 但是，当 InTouch 7.1 应用程序转换为 7.11 应用程序时，为保证反向兼容性，标记名注释将复制到“**报警注释**”中。
.AlarmDate	返回当前报警的日期。
.AlarmDev	发出存在偏差报警的信号。
.AlarmDevCount	记录给定标记名或报警组上活动偏差报警的总数。
.AlarmDevDeadband	监视/控制副偏差和主偏差报警偏离死区的百分比。
.AlarmDevUnAckCount	记录给定标记名或报警组上活动的未确认偏差报警总数。
.AlarmDisabled	禁用/启用事件和报警。应用于含报警的离散或模拟型标记名，或应用于报警组。 **注意**：与 **.AlarmEnabled** 相同，但极性相反。
.AlarmDsc	指示离散报警条件当前处于活动状态。
.AlarmDSCCount	记录给定标记名或报警组上活动离散报警的总数。
.AlarmDscDisabled	表明标记名是否可产生离散报警。 **注意**：对于离散型标记名，此点域与 **.AlarmDisabled** 相同。
.AlarmDscEnabled	表明标记名是否可产生离散报警。 **注意**：对于离散型标记名，此点域与 **.AlarmEnabled** 相同。

点域	描述
.AlarmDscInhibitor	返回指定给此标记名的离散报警的约束器标记名（如果有）。
.AlarmDscUnAckCount	记录给定标记名或报警组上活动的未确认离散报警总数。
.AlarmEnabled	禁用/启用事件和报警。
.AlarmGroup	包含用于填入分布式报警显示对象的当前查询。
.AlarmGroupSel	返回报警的报警组。
.AlarmHiDisabled	禁用/启用含报警的模拟标记名的 **Hi** 报警。
.AlarmHiEnabled	禁用/启用含报警的模拟标记名的 **Hi** 报警。 **注意**：与对应的禁用点域相同，但极性相反。
.AlarmHiInhibitor	返回 **Hi** 报警限的约束器标记名引用。应用于含报警的模拟型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmHiHiDisabled	禁用/启用含报警的模拟标记名的 **HiHi** 报警。
.AlarmHiHiEnabled	禁用/启用含报警的模拟标记名的 **HiHi** 报警。 **注意**：与对应的禁用点域相同，但极性相反。
.AlarmHiHiInhibitor	返回 **HiHi** 报警限的约束器标记名引用。应用于含报警的模拟型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmLimit	返回当前报警的报警限。
.AlarmLoDisabled	禁用/启用含报警的模拟标记名的 **Lo** 报警。
.AlarmLoEnabled	禁用/启用含报警的模拟标记名的 **Lo** 报警。 **注意**：与对应的禁用点域相同，但极性相反。
.AlarmLoInhibitor	返回 **Lo** 报警限的约束器标记名引用。应用于含报警的模拟型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmLoLoDisabled	禁用/启用含报警的模拟标记名的 **LoLo** 报警。

点域	描述
.AlarmLoLoEnabled	禁用/启用含报警的模拟标记名的 **LoLo** 报警。 **注意**：与对应的禁用点域相同，但极性相反。
.AlarmLoLoInhibitor	返回 **LoLo** 报警限的约束器标记名引用。应用于含报警的模拟型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmMajDevDisabled	禁用/启用含报警的模拟标记名的**主偏差**报警。
.AlarmMajDevEnabled	禁用/启用含报警的模拟标记名的**主偏差**报警。 **注意**：与对应的禁用点域相同，但极性相反。
.AlarmMajDevInhibitor	返回**主偏差**报警限的约束器标记名引用。应用于含报警的模拟型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmMinDevDisabled	禁用/启用含报警的模拟标记名的**副偏差**报警。
.AlarmMinDevEnabled	禁用/启用含报警的模拟标记名的**副偏差**报警。 **注意**：与对应的禁用点域相同，但极性相反。
.AlarmMinDevInhibitor	返回**副偏差**报警限的约束器标记名引用。应用于含报警的模拟型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmName	返回当前报警的名称。
.AlarmOprName	返回当前报警的操作员。
.AlarmOprNode	返回当前报警的操作员节点。
.AlarmPri	返回当前报警的优先值 (1-999)。
.AlarmProv	返回当前报警的供应器。
.AlarmROC	发出存在变化率类型报警的信号。
.AlarmROCCount	记录给定标记名或报警组上活动变化率报警的总数。
.AlarmROCDisabled	禁用/启用含报警的模拟标记名的**变化率**报警。

点域	描述
.AlarmROCEnabled	禁用/启用含报警的模拟标记名的**变化率**报警。
	注意：与对应的禁用点域相同，但极性相反。
.AlarmROCInhibitor	返回**变化率**报警限的约束器标记名引用。应用于含报警的模拟型标记名。 只读属性，但可在 WindowMaker 中配置。
.AlarmROCUnAckCount	记录给定标记名或报警组上的未确认变化率报警的总数。
.AlarmState	返回当前报警的状态。
.AlarmTime	返回当前报警的时间。
.AlarmTotalCount	记录给定标记名或报警组上活动报警的总数。
.AlarmType	返回当前报警的类型。
.AlarmUnAckCount	记录给定标记名或报警组上活动的未确认报警总数。
.AlarmUserDefNum1	读/写整型（浮点）、缺省为 0，值未设定。应用于含报警的离散型标记名、含报警的模拟标记名或报警组。
	注意：此点域的值会附加在报警上，但只限于已设置值的情况下（例如通过脚本或插入来设置）。
.AlarmUserDefNum2	读/写整型（浮点）、缺省为 0，值未设定。应用于含报警的离散型标记名、含报警的模拟标记名或报警组。
	注意：此点域的值会附加在报警上，但只限于已设置值的情况下（例如通过脚本或插入来设置）。
.AlarmUserDefStr	读/写文本字符串、缺省为""，值未设定。应用于含报警的离散型标记名、含报警的模拟标记名或报警组。
	注意：此点域的值会附加在报警上，但只限于已设置值的情况下（例如通过脚本或插入来设置）。

注意：.AlarmUserDefNum1、.AlarmUserDefNum2 和 **AlarmUserDefStr** 点域允许您在报告时，指定一个或多个要追加到报警记录上的值。这些值由 Alarm DB Logger 写入数据库中。您可以在报警上附加三个项目：两个数字和一个字符串。缺省条件下，它们是空的（零和 ""）。

要简化用户值的设置，您可以在报警组及特定的标记名上设置它们。例如，InBatch 可以在 **$System** 报警组上设置 **.AlarmUserDefNum1** 中的批号，从而让所有报警都附上批号。

如果在某个报警组上设置 **.AlarmUserDefNum1**，它将应用于该组及其子组中的所有报警。您也可以在标记名上专门设置 **.AlarmUserDefNum1** 的值，在这种情况下，它将只应用于该标记名并改写标记名报警组中 **.AlarmUserDefNum1** 的任何设置。

点域	描述
.AlarmValDeadband	监视/控制报警死区值。
.AlarmValue	返回当前报警值。
.AlarmValueCount	记录给定标记名或报警组上活动值报警的总数。
.AlarmValueUnAckCount	记录给定标记名或报警组上活动的未确认值报警总数。
.DevTarget	监视/控制副偏差和主偏差报警目标。
.Freeze	读/写分布式报警显示对象的冻结状态。
.HiLimit, .HiHiLimit, .LoLimit, .LoLoLimit	读/写模拟标记名点域，监视/控制值报警检查极限。这些点域只对整型和实型标记名有效。
.HiStatus, .HiHiStatus, .LoStatus, .LoLoStatus	只读离散标记名点域，确定是否存在指定类型的报警。这些点域只对整型和实型标记名有效。
.MajorDevPct	读/写整型点域，监视或控制报警检查的主偏差百分比。
.MajorDevStatus	只读离散型点域，确定是否存在指定标记名的主偏差报警。
.MinorDevPct	读/写整型点域，监视和（或）控制报警检查的副偏差百分比。
.MinorDevStatus	只读离散型点域，确定是否存在指定标记名的副偏差报警。

点域	描述
.Name	读/写消息型点域，用于显示标记名的实际名称。例如，它能确定组变量所指向的报警组的名称或 TagID 标记名的名称，也可以被改写以改变组变量所指向的报警组。
.ListChanged	指明是否存在任何新报警或分布式报警对象的更新。
.Normal	只读离散型点域，当指定标记名没有报警时为 1。此点域对报警组和组变量及常规标记名有效。
.NumAlarm	显示分布式报警对象中的报警总数。
.PendingUpdates	指明报警显示对象有任何等待更新的内容。
.ROCPct	读/写点域，用于监视和（或）控制报警检查的变化率。
.ROCStatus	只读离散型点域，用于确定是否存在指定标记名的变化率报警。
.SuppressRetain	读/写分布式报警显示对象的抑制保持状态。

下列报警点域与报警对象（分布式报警显示）关联，而与标记名无关。

.AlarmClass	**.AlarmAccess**	**.AlarmDate**	**.AlarmGroupSel**
.AlarmLimit	**.AlarmName**	**.AlarmOprName**	**.AlarmOprNode**
.AlarmPri	**.AlarmProv**	**.AlarmState**	**.AlarmTime**
.AlarmType	**.AlarmValue**	**.Freeze**	**.ListChanged**
.NumAlarm	**.PendingUpdates**	**.SuppressRetain**	

这些报警点域通过下面的函数进行访问：

```
GetPropertyM(ControlName.Property, MsgTag)
```

当在运行环境下执行时，**MsgTag** 会取得属性以在选定的行中显示标记名。如果选定多个行，**MsgTag** 的属性为其中第一行里的标记名。

有关报警点域的详细信息，请参阅联机*《Intouch 参考指南》*。

从旧 InTouch 标准报警系统移植到分布式报警系统

InTouch 可自动移植使用标准报警对象的 InTouch 应用程序。当您尝试在 WindowMaker 中打开带有标准报警对象的任何应用程序时，InTouch 会使用缺省值，自动将所有标准报警对象转换为分布式报警对象，但颜色、字体、表达式和报警查询等设置不会移植。

当您尝试在 Window Viewer 中打开使用 7.11 之前版本创建的应用程序时，屏幕会弹出一个对话框，提示您运行 WindowMaker 来转换这个应用程序。如果应用程序中包含标准报警对象，转换过程会将标准报警对象移植到分布式报警对象。

如果您单击“**是**”，应用程序将在 WindowMaker 中打开并转换。如果单击“**否**”，则 Viewer 将关闭。

从旧的主/从报警系统移植到分布式报警系统

正如“从旧标准报警系统移植到分布式报警系统”部分所述，主/从应用程序中的所有标准报警显示会被分布式报警显示所代替。新的分布式报警显示将使用缺省查询：“\\nodename\intouch!$system”，这里的 nodename 指主节点的名称。

您现在应能够使用分布式报警显示来查看和确认远程报警。

通过对 .Ack 点域赋值来确认远程报警的操作与以前相同，并不需要更改应用程序。取决于 I/O 标记名是为 NetDDE 还是 SuiteLink 配置，可能需要启用 NetDDE。但是，由于现有可以通过分布式报警显示来确认报警，因此您可以决定不再需要独立的控件来发出确认。

通过提示 .Alarm 点域来监视报警状态的操作也与原来相同，并不需要更改应用程序。取决于 I/O 标记名是为 NetDDE 还是 SuiteLink 配置，可能需要启用 NetDDE。

Alarm Printer 公用程序

Alarm Printer 公用程序为您提供了一个集中化的 Alarm Printer 公用程序，可以通过单个独立的公用程序打印出横跨多个节点的报警信息。Alarm Printer 可让您访问分布式报警。通过 Alarm Printer 公用程序，您可以使用专用的行式打印机，逐个事件地打印报警状态的变化。您可以定义不同的打印选项，并将这些定义保存在 Alarm Printer 公用程序配置文件 (*.ALC) 中，以供日后检索使用。

报警的每次转换构成**事件**，事件可以为诊断故障提供帮助。分布式报警系统可以配置，当特定事件发生时，在行式打印机上将其打印出来。当发生灾难性系统故障和（或）掉电时，客户通常将打印输出视为最后的手段，这指的是通过串行端口或并行端口打印到点矩阵打印机上。Windows 网络打印机和激光打印机通常不适用于此功能，因为它们会将整页缓存在内存中，再将其实际打印出来 – 这意味着信息可能会在系统故障或掉电时丢失。

注意：Alarm Printer 公用程序只查询历史数据，而不查询摘要数据。

报警打印日期/时间标签

Alarm Printer 的缺省日期格式为：

DD MMM YYYY

此处的 **DD** 指日期，**MMM** 指月份，**YYYY** 指年度。

Alarm Printer 的缺省时间格式为：

HH:MM:SS.SSS

这里的 **HH** 指小时，**MM** 指分钟，**SS** 指秒，**SSS** 指毫秒。

注意：无论计算机的时区如何设置，报警记录的时间字段始终为 **GMT** 格式。

您可以在“**配置设定**”对话框的“**消息**”属性页中更改日期和（或）时间的缺省格式。

有关详细信息，请参阅“配置 Alarm Printer 公用程序”。

使用一个或多个 Alarm Printer

您可以同时运行 Alarm Printer 的多个实例 (AlmPrt.exe)。Alarm Printer 的每个实例必须配置为打印到不同的打印机，并且必须用自己的独立报警查询进行配置。

- 您可以配置 Alarm Printer 的不同实例，以打印特定优先级范围内的报警。例如，某个 Alarm Printer 实例可能只打印高优先级报警，而另一个实例则只打印低优先级报警。同样，您可以使用 Alarm Printer 的一个实例来打印来自工厂某个区域的报警，而使用另一个实例来打印另一区域的报警。

使用 Alarm Printer

在首次启动 Alarm Printer 时，“**Alarm Printer**”窗口会出现，显示 Alarm Printer 的缺省配置设定。

注意：如果运行时在命令提示符下打开，或双击其 *.ALC 文件名，则可能会显示一个特定的报警配置文件。



➢ **要创建新的 Alarm Printer 配置文件：**

1. 在“**文件**”菜单上，单击“**新建**”。Alarm Printer 缺省**报警状态**为“全部”，**优先级**为“1-999”，**打印端口**为“<无>”。
2. 在“**输出**”菜单上，选择“**配置**”。“**配置设定**”对话框出现。输入您需要的设置。
3. 在“**文件**”菜单上，选择“**保存**”。

➢ **要编辑现有的 Alarm Printer 配置文件：**

1. 在“**文件**”菜单上，单击“**打开**”。
2. 选择您要编辑的 Alarm Printer 配置文件。
3. 要保存更改，在“**文件**”菜单上，选择“**保存**”。
4. 或者，在“**文件**”菜单上，选择“**另存为**”，将更改保存到一个新文件，而不改变现有文件。

有关 Alarm Printer 配置的详细信息，请参阅“配置 Alarm Printer”。

Alarm Printer 工具栏



Alarm Printer 工具栏上的工具允许您快速地访问所有 Alarm Printer 命令。

下图图示介绍了每个工具：

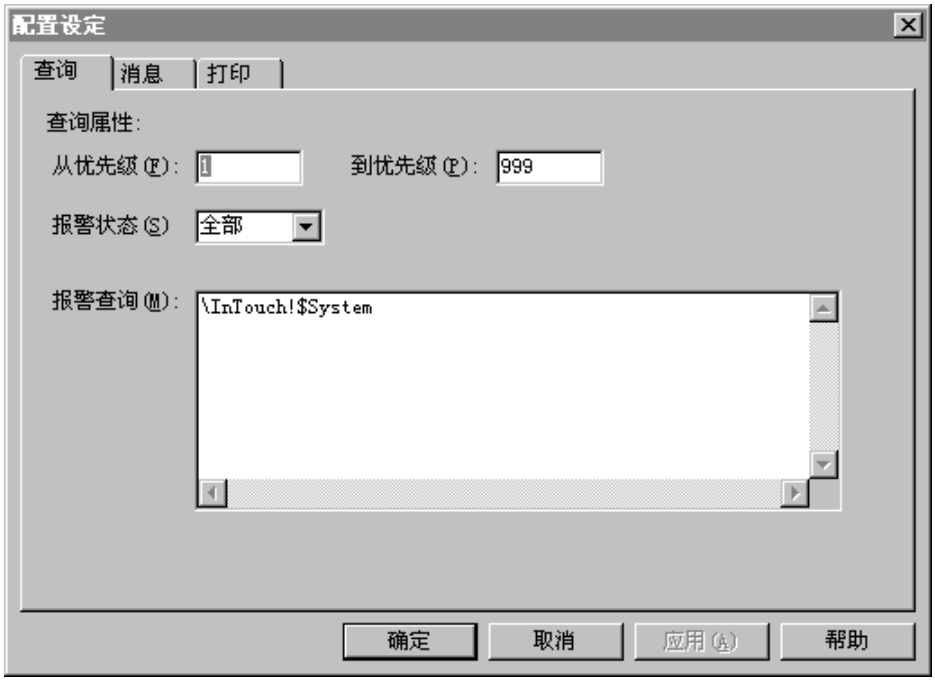
工具	描述
	创建新的 Alarm Printer 配置文件。
	打开现有的 Alarm Printer 配置文件。
	将当前 Alarm Printer 配置文件保存到磁盘。
	配置 Alarm Printer。
	启动/停止查询。
	启用/禁用报警打印。

配置 Alarm Printer

Alarm Printer 配置保存在 Alarm Printer 配置文件 (*.alc) 中，可保存的配置文件数没有限制。Alarm Printer 针对运行的 Alarm Printer 的每个实例，可只使用一个配置文件。

➢ **要配置 Alarm Printer 查询属性：**

1. 选择“**配置**”，或单击 Alarm Printer 工具栏上的“配置报警查询”工具。
2. “**配置设定**”对话框出现：



3. 在“**从优先级**”字段中，输入最高优先级报警值（1 到 999）。
4. 在“**到优先级**”字段中，输入最低优先级报警值（1 到 999）。

注意：InTouch 中配置的每个报警都有一个与之关联的优先值。此值表示报警的严重程度，范围从“1”到“999”，其中“1”表示最严重。

有关报警优先级的详细信息，请参阅“报警优先级”。

5. 单击“**报警状态**”箭头，在列表中选择您要使用的报警状态。
6. 在“**报警查询**”输入框中，输入您要执行的 InTouch 报警查询组。您可以指定一个或多个报警供应器和报警组，方法与分布式报警显示的报警查询一样。有关详细信息，请参阅“配置分布式报警显示”。

➢ **要配置报警记录属性：**

1. 在“**输出**”菜单上，选择“**配置**”，或单击 Alarm Printer 工具栏上的“配置报警查询”工具。
2. “**配置设定**”对话框出现。
3. 选择“**消息**”标签，激活“**消息**”属性页：



您选择的每个选项将显示为所生成的打印输出中的不同字段。设置为特定字符长度的字段将按该长度限制截断。

如果右击任何一个报警配置对话框中的文本框，将出现一个菜单，显示可应用到选定文本的命令。

注意：大多数打印机每行可打印 80 或 132 个字符。

4. 要打印报警日期，选择“**日期**”选项，然后单击箭头选择日期格式。可选格式为：

选择	**格式**	**选择**	**格式**
DD MMM	28 Feb	**MM/DD**	02/28
DD MM YYYY	28 Feb 1997	**MM/DD/YY**	02/28/97
DD/MM	28/02	**MMM DD**	Feb 28
DD/MM/YY	28/02/97	**MMM DD YYYY**	Feb 28 1997

5. 要打印报警时间，选择“**时间**”选项，然后单击箭头选择时间格式。可选格式为：

选择	描述
AP	选择 AM/PM 格式。例如下午三点钟显示为 3:00 PM。非此格式的时间缺省使用 24 小时军用时间格式。例如下午三点钟显示为 15:00。
HH	打印发生报警/事件的小时。
MM	打印发生报警/事件的分钟。
SS	打印发生报警/事件的秒。
SSS	打印发生报警/事件的毫秒。

6. 在“**时间**”下面的排序框内，选择报警记录中的报警排序顺序。共有三个选项：

选择	描述
OAT	原始报警时间 – 即报警发生的日期/时间戳。
LCT	上次更改时间 – 即报警实例最近一次状态改变的日期/时间戳，包括报警发生、子状态改变、恢复正常或确认。
LCT 但确认为 OAT	上次更改时间，但确认时使用原始报警时间 – 即在报警未确认时使用 LCT，但在报警获得确认时使用 OAT。

7. 要打印报警状态，选择“**报警状态 (UnAck, Ack)**”选项。
8. 要打印报警类，选择“**报警分类 (VALUE, DEV, ROC..)**”选项。
9. 要打印报警类型，选择“**报警类型 (HIHI, LO, MAJDEV,…)**”选项。

   有关可用报警类型的详细信息，请参阅“报警类型”。

10. 要打印报警优先级，选择“**优先级**”选项。
11. 要打印报警名称（标记名），选择“**报警名**”选项。在“**长度**”框中，输入报警名允许的字符数（最多 64 个字符）。
12. 要打印报警组名，选择“**组名**”选项。在“**长度**”框中，输入报警组名允许的字符数（最多 64 个字符）。
13. 要打印报警供应器名称，选择“**报警供应器**”选项。在“**长度**”框中，输入报警供应器名允许的字符数（最多 64 个字符）。
14. 要打印标记名值，选择“**报警值**”选项。在“**长度**”框中，输入报警值允许的字符数（最多 32 个字符）。输入的数值应该足够大以满足精度要求。

15. 要打印标记名的报警限，选择“**极限**”选项。在“**长度**”框中，输入报警限允许的字符数（最多 32 个字符）。输入的数值应该足够大以满足精度要求。
16. 要打印与报警条件关联的操作员 ID，选择“**操作员**”选项。在“**长度**”框中，输入操作员 ID 允许的字符数（最多 16 个字符）。
17. 要打印与标记名关联的报警注释，选择“**注释**”选项。在“**长度**”框中，输入注释允许的字符数（最多 131 个字符）。
18. 选择“**删除尾部空格**”选项，以在实际字段值长度小于该字段的配置长度时删除所打印字段的附加尾部空格。

➢ **要配置报警打印：**

1. 选择“**Alarm Printer**”对话框的“**配置**”菜单，或单击 Alarm Printer 工具栏上的“配置报警查询”工具。
2. “**配置设定**”对话框出现。
3. 选择“**打印**”标签。



4. 选择您要 Alarm Printer 用于打印报表的打印机端口选项。

注意：用于打印报警状态改变的打印机不得用于其它打印任务，因为这会干扰 Alarm Printer 的打印作业。

5. 如果选定 COM 端口，单击“**端口配置**”以配置串行端口。“**COM# 属性**”对话框出现。

要使用网络打印机上的打印机端口，您可以在启动 InTouch 之前使用命令环境下的 NET USE 命令。

下面的实例将网络流量从 COM1 转移到服务器 "PRTSRV" 上的网络打印机 "LINEP1"：

```
net use com1: \\PRTSRV\LINEP1 /persistent:yes.
```



6. 通过单击各个字段右边的箭头，并选择与当前使用的 COM 端口相应的设置，来配置 COM 端口。

7. 单击“**确定**”。

运行 Alarm Printer 的报警查询

每项查询都会记录当前打开的 Alarm Printer 配置文件 (*.ALC) 中指定的所有报警，如果未指定文件，则记录由 Alarm Printer 配置期间的当前选定设置所指定的所有报警。

有关配置选项的详细信息，请参阅“配置 Alarm Printer”。

您可以在 Alarm Printer 上运行多项查询，每项查询指定不同的参数。各项查询由 Alarm Printer 的不同实例执行。如果 Alarm Printer 的实例运行相同的查询，则查询结果将会重复。

当 Alarm Printer 运行时，您可以手动地启动或停止查询，如下所示：

-  **要启动或停止报警查询：**

  在“**查询**”菜单上，选择“**启动/停止**”。

  要快速启动或停止查询，单击工具栏上的“启动/停止查询”工具。

- **要自动启动 Alarm Printer 并打开特定文件：**

  您可以在系统启动时自动启动 Alarm Printer 并打开特定的文件，方法是在 .BAT 批文件中使用下面的命令：

  ALMPRT.EXE MYQUERY.ALC

  此处的 **MYQUERY.ALC** 是打开的 Alarm Printer 配置文件名（指定 **.EXE** 扩展名是一可选项）。

- **要自动启动 Alarm Printer 并运行查询：**

  要防止因系统意外关机或重新启动而导致的任何查询数据丢失，您可以通过在 .BAT 批文件中使用下列命令来自动启动 Alarm Printer 并自动运行特定的查询。

  ALMPRT.EXE -q MYQUERY.ALC

  在命令中使用 **–q**，可以在系统启动时自动运行查询（指定 **.EXE** 扩展名是一可选项）。

Alarm DB Logger 公用程序

InTouch 分布式报警系统包括 Alarm DB Logger 公用程序，该程序将报警和事件记入 Microsoft SQL Server 和（或） Microsoft Data Engine (MSDE)。

注意：MSDE 是 SQL Server 的变型，但它具有自己的独到之处，尤其是其易于使用性，您可以将原先为 MSDE 构建的数据库轻易地加入到完全版的 SQL Server 服务中，而不需要升级数据库或将数据库中的单个表从 MSDE 复制到完全版的 SQL Server 上。这使之更适用于要求经济地布署大量计算机资源的环境中。MSDE 数据库的极限为 2 GB。

Alarm DB Logger 是一个报警接收器。您可以配置其“**报警查询**”来定义需要记录的报警。您可以使用 Alarm DB Logger 来指定报警查询并记录报警结果。这些报警查询通过分布式报警系统的报警接收器接口发送出去。

Alarm DB Logger 还具有自动重新连接的能力。当与数据库的连接中断时，Alarm DB Logger 会按一定的间隔继续检查数据库连接。在重新建立连接后，记录工作继续进行。

无论是作为服务还是常规应用程序运行，Alarm DB Logger 都会将所有错误报告给 Wonderware Logger。

Alarm DB Logger 包括以下两个组件：

1. **Alarm DB Logger Manager 公用程序** – 此公用程序单独运行时只负责启动和停止记录操作。启动后既可作为服务又可作为常规应用程序开始工作（取决于在 Alarm DB Logger Manager 中选择的运行模式）。记录公用程序从注册表检索设置信息并执行记录。
2. **Alarm DB Logger 配置公用程序** – 此公用程序负责用户输入和数据库配置。Alarm DB Logger Manager 允许您选择 Alarm DB Logger 的运行方式（作为 Windows 服务还是常规应用程序）。

   注意：Alarm DB Logger Manager 只将设置值保存到注册表中。此公用程序负责启动和停止 Alarm DB Logger，还负责显示智能缓存的状态。如在运行 wwalmlogger.exe 时关闭 Alarm DB Logger Manager (almlogwiz.exe)（按 Esc 键或单击对话框右上角的 “X” 按钮），记录操作不会停止。

   进度控制状态表明装入内存缓冲区的报警记录百分比。当 SQL Server 连接中断和（或）报警的传输速率高于 Alarm DB Logger 的记录速率时，报警将被缓存。

Alarm DB Logger 配置公用程序使您能够：

将应用程序作为 Windows 服务或常规应用程序来运行。

选择数据库连接类型 – SQL Server 或 MSDE

在数据库中创建必要的 SQL 表

指定作为记录实例一部分的报警查询

选择记录模式 – 详细或合并。

检查/撤消检查事件记录

设置性能调整能参数 – 自动重新连接速率与性能调整参数不同，它取决于与 SQL Server 尝试建立连接的超时值。

将设置保存到注册表中

有关详细信息，请参阅“Alarm DB Logger 配置”。

记入 SQL 数据库

Alarm DB Logger 将报警数据记入数据库。如果 OLEDB 提供者是 SQL Server，您需要在 Alarm DB Logger Manager 中指定 SQL Server 机器名。Alarm DB Logger 会自动创建必要的数据结构，如果数据库中尚不存在该数据结构。

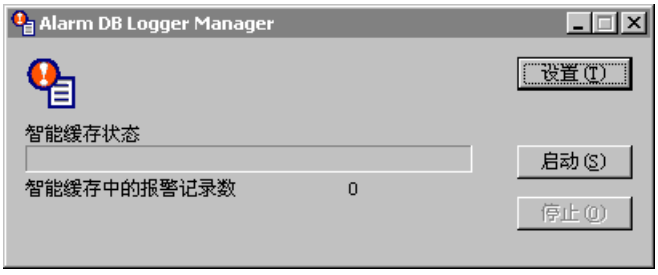
有关详细信息，请参阅“分布式报警数据库视图”。

使用 Alarm DB Logger 公用程序

Alarm DB Logger 公用程序单独运行时只负责记录工作，启动后既可作为服务又可作为常规应用程序开始工作（取决于在配置 Alarm DB Logger 时选择的运行模式）。记录公用程序从注册表检索设置信息并执行记录。Alarm DB Logger 是一个报警接收器。

➢ **要使用 Alarm DB Logger 公用程序：**

1. 启动 Alarm DB Logger Manager，**“Alarm DB Logger Manager”** 对话框出现。



当最小化时，Alarm DB Logger Manager 显示为系统托盘中的一个图标。当右击图标时，会出现一个弹出式菜单，显示下列命令：

启动 – 开始报警记录过程。
停止 – 结束报警记录过程。
设置 – 打开“**Alarm DB Logger Manager – 配置**”对话框。
隐藏窗口 – Alarm DB Logger Manager 对话框最小化为系统托盘中的一个图标。
显示窗口 – 打开并最大化“Alarm DB Logger Manager”对话框。
关闭 – 退出 Alarm DB Logger Manager 公用程序。

2. **智能缓存状态**表明装入内存缓冲区的报警记录百分比。
3. 单击“**设置**”来配置 Alarm DB Logger，**“Alarm DB Logger Manager – 配置”**对话框出现：

   有关配置 Alarm DB Logger 的详细信息，请参阅“Alarm DB Logger 配置”。
4. 单击“**启动**”开始报警记录过程。
5. 单击“**停止**”结束报警记录过程。

Alarm DB Logger 配置

在开始使用 Alarm DB Logger 之前，必须配置若干项目，如数据库连接、查询列表及记录模式等等。此信息通过 Alarm DB Logger Manager 配置。

第一个 Alarm DB Logger Manager 配置对话框处理数据库连接。您可以选择 SQL Server 或 Microsoft Data Engine (MSDE) 来存储数据。Alarm DB Logger 配置公用程序也允许创建必要的数据结构（表、视图和存储过程）来测试数据库连接。

➢ **要配置 Alarm DB Logger：**

1. 启动 Alarm DB Logger Manager，**“Alarm DB Logger Manager”** 对话框出现。
2. 单击“**设置**”，**“Alarm DB Logger Manager – 配置”**对话框出现：



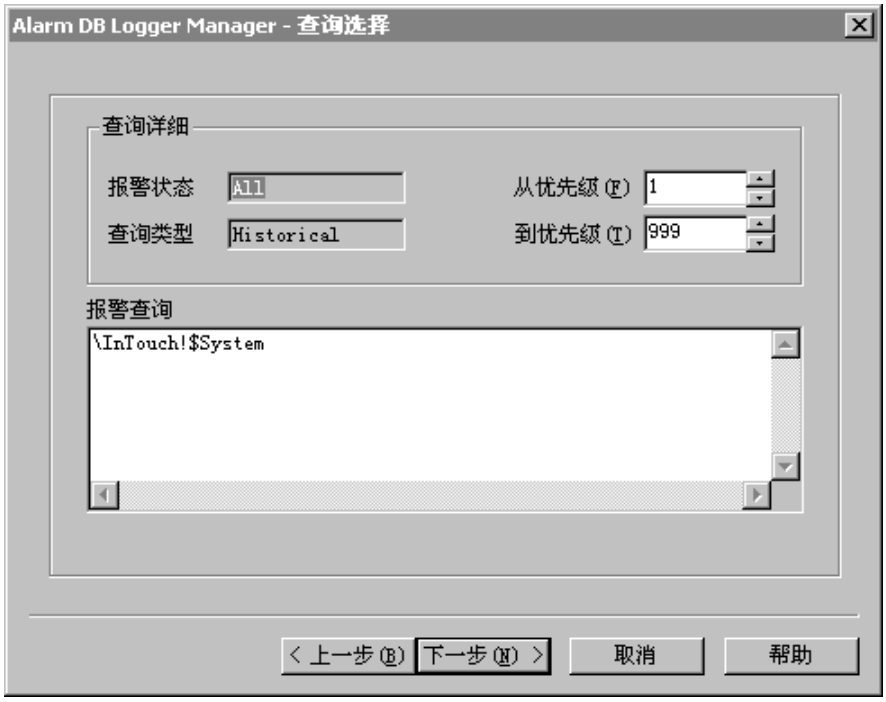
3. 在“**SQL Server/MSDE**”组中，单击“**服务器名**”箭头打开可用 SQL/MSDE 服务器的列表，然后选择您要使用的数据库服务器名。
4. 在“**用户信息**”组的相应输入框中，输入您的“**用户名**”和“**口令**”（基于安全考虑，所有**口令**字符均显示为 **）。
5. 单击“**测试连接**”来测试与目标数据库的连接。
6. 单击“**创建**”在选定的服务器上创建新的数据库/表。
7. 单击“**下一步**”访问“**Alarm DB Logger Manager – 查询选择**”对话框。

Alarm DB Logger 查询配置

第二个 Alarm DB Logger Manager 对话框指需选取用以记录报警的查询详细信息。它提供一个多行文本输入框来输入查询，还允许您选择“**从优先级**”和“**到优先级**”值。

➢ **要配置查询详细信息：**

1. 启动 Alarm DB Logger Manager 公用程序（如果尚未运行），**“Alarm DB Logger Manager 配置”**对话框出现。
2. 单击“**下一步**”，**“Alarm DB Logger Manager – 查询选择”**对话框出现：



注意：“**报警状态**”字段显示要记录的报警状态；“**查询类型**”字段显示查询类型（二者都是只读字段）。

3. 单击“**从优先级**”箭头，选择报警优先级范围的起始值。
4. 单击“**到优先级**”箭头，选择报警优先级范围的结束值。
5. 在从主“**报警查询**”输入框中，输入您要执行的 InTouch 报警查询组。
6. 单击“**下一步**”，**“**Alarm DB Logger Manager – **高级设定”**对话框出现。

Alarm DB Logger 高级设定配置

第三个 Alarm DB Logger Manager 对话框指高级查询设置，如运行模式（作为 Windows 服务还是常规应用程序）。它也提供用于记录事件的选项。您可以通过设置从内存缓存区到数据库的报警刷新频率，来调整报警记录的性能。

➢ **要配置高级设定详细信息：**

1. 启动 Alarm DB Logger Manager（如果尚未运行），**"Alarm DB Logger Manager 配置"**对话框出现。
2. 单击**"下一步"**，**"Alarm DB Logger Manager – 查询选择"**对话框出现。
3. 单击**"下一步"**，**"Alarm DB Logger Manager – 高级设定"**对话框出现：



4. 在**"记录器运行方式为"**组中，选择您要用于记录器的选项。

   Windows 服务 – 配置记录器作为 Windows 服务运行。

   正常应用程序 – 配置记录器作为常规应用程序运行。

5. 选择**"记录事件"**选项，以将所有事件记入事件表。

6. 在“**报警记录间隔为 # 毫秒**”输入框中，输入以毫秒计的报警记录间隔。

注意：重新连接速率与此毫秒率设定不同，它取决于与 SQL Server 尝试建立连接的超时值。

将性能调整参数设置得太高会影响系统性能。

7. 单击“**完成**”关闭 Alarm DB Logger Manager 公用程序，并将所有记录器配置设定保存到注册表中。

报警数据库视图 ActiveX 控件

InTouch 提供了一个报警数据库视图 ActiveX 控件，允许您形象地查看新 Alarm DB Logger 数据库中的报警数据。您可以使用此控件来查看所有报警和事件信息。通过指定下列属性，您可以设计 ActiveX 控件和所显示数据的外观。

上下文相关菜单功能

显示模式

列表控制选项

不同属性的颜色

字体类型、样式和大小

数据库规格（服务器名、用户 ID 和口令）

报警选择（过滤器）

查询过滤器

列管理

排序

一旦设计控件格式，用户将能够进行以下调整来操纵其所查看的数据：

对列内的信息排序

更新显示

执行查询

调整列宽

报警数据库视图 ActiveX 控件可拖入 WindowMaker 中并调整大小和位置，您可以使用属性页来配置控件。来自数据库的数据可以在报警数据库视图 ActiveX 控件的视图窗口中进行查看。所查看数据的类型和格式取决于属性页中设置的属性。

安装

报警数据库视图 ActiveX 控件随 InTouch 7.11 一起安装。

- **要将此控件粘贴到 WindowMaker 窗口中：**

1. 从“**特别**”菜单上，选择“配置”。
2. 选择“**向导/ActiveX 安装**”，打开“向导/ActiveX 安装”对话框。
3. 选择 **ActiveX 控件**安装属性页。
4. 从可用 ActiveX 控件列表中选择“Wonderware 报警数据库查看控制”，然后单击“**安装**”。报警数据库视图控件名将显示在“已安装的 ActiveX 控件”文本区域中。
5. 单击“**关闭**”。
6. 打开“向导选择”对话框。
7. 选择 ActiveX 控件下的 **AlmDbViewCtrl**，然后单击“**确定**”。
8. 将控件粘贴到窗口上并调整到所需的大小。

卸载

1. 删除粘贴到窗口上的所有报警数据库视图控件。
2. 从“**特别**”菜单上，选择“配置”。
3. 选择“**向导/ActiveX 安装**”，打开“向导/ActiveX 安装”对话框。
4. 选择 **ActiveX 控件**安装属性页。Wonderware 报警数据库视图控件名将显示在“已安装的 ActiveX 控件”文本区域中。
5. 单击“Wonderware 报警数据库查看控制”，然后单击“**删除**”。
6. 在显示警告消息时单击“**是**”。
7. 单击“**关闭**”。

属性

本节通过将“属性”对话框中显示的那些属性与脚本编辑器中可配置的那些属性相关联，来介绍此控件的各种属性。

注意：在下面的章节中，用诸如 Object.RefreshMenu 之类的符号来描述语法。在所有的实例中，“Object” 指相当于对象或控件的对象表达式。
例如：
AlmDBView1.RefreshMenu = False
此表达式指设置对象 AlmDBView1 的属性值。

访问属性对话框的方式有两种：

双击控件，或

右击控件，然后从弹出式菜单中选择“属性”菜单。

控件名称页

属性对话框的“控件名”标签提供与应用程序有关的控件的详细信息。

注意：InTouch 将粘贴到应用程序上的第一个控件命名为 AlmDBViewCtrl1。

“扩展属性”区域提供有关控件的窗口配置（左、顶部、宽度、高度）的详细信息，您可以改变这些属性。

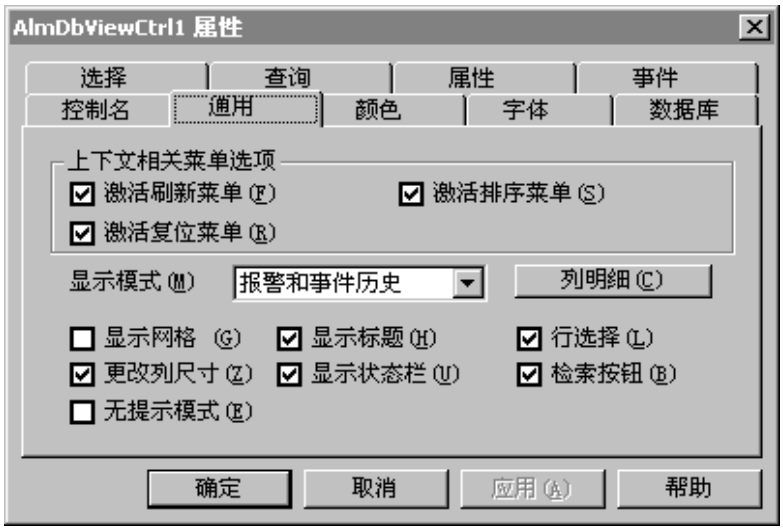
撤消选择**“可见的”**复选框可使控件在运行时不可见。

禁用的 **GUID** 文本框显示此 ActiveX 控件的唯一 ID。



通用属性页

“通用”标签提供此控件的用户界面中包含的特征选项。



上下文相关菜单选项

“上下文相关菜单选项”区域提供的选项可启用和禁用此控件中可用的上下文相关菜单功能。

注意：以下关于 AlmDbViewCtrl 属性和方法的描述基于使用 InTouch 编程时的下列假定：

"True" 并且类型 = 整型

设置：指定一个不等于 0 的值或指定 "True"

读取：获得整型值 -1

"False"并且类型 = 整型

设置：指定 0 或 "False"

读取：获得整型值 0

设置实例：

```
#AlmDbViewCtrl6.SortMenu = "True"; {enables the Sort Menu
   in the context Menu of the Control}

#AlmDbViewCtrl6.SortMenu = 1; {enables the Sort Menu in the
   context Menu of the Control}

#AlmDbViewCtrl6.SortMenu = "False"; {disables the Sort Menu
   in the context Menu of the Control}

#AlmDbViewCtrl6.SortMenu = 0; {disables the Sort Menu in
   the context Menu of the Control}
```

读取实例：

```
IF #AlmDbViewCtrl6.ShowFetch == 0 THEN {checks for the
   actual state of the property}

#AlmDbViewCtrl6.ShowFetch = 1; {toggles the property into
   the "True" state if it was previously in the "False"
   state}

ELSE

#AlmDbViewCtrl6.ShowFetch = 0; {toggles the property into
   the "False" state if it was previously in the "True"
   state}

ENDIF;

Int_Tag = #AlmDbViewCtrl6.ShowGrid {will assign -1 to Int_Tag
   when Grid is shown, will assign 0 when Grid is not shown}
```

激活刷新菜单

选择此复选框以启用控件右击菜单的“刷新菜单”选项。“刷新”菜单会刷新控件到数据库中，如果连接成功，它会显示范围 1 到 MaxRecords 中的记录集。

属性：	**RefreshMenu**		
用途：	返回或设置一个决定是否在上下文相关菜单中显示“刷新”菜单项的值。		
类型：	整型		
缺省：	True		
语法：	*Object*.RefreshMenu [= *Integer*]		
	Integer	整型表达式，指定是否按“设置”中的设置显示“刷新”菜单项。	
		True	（缺省）显示“刷新”菜单项
		False	不显示“刷新”菜单项
实例：			
	设置：	**#AlmDbView1.RefreshMenu = “False”;**	
	读取：	**Tagname = #AlmDbView1.RefreshMenu;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

激活排序菜单

选择此复选框以启用控件右击菜单的“**排序菜单**”选项。此菜单显示用于设置用户自定义列排序的辅助排序菜单。

属性：	SortMenu		
用途：	返回或设置一个决定是否在上下文相关菜单中显示“排序”菜单项的值。		
类型：	整型		
缺省：	True		
语法：	*Object*.SortMenu *[= Integer]*		
	Integer	整型表达式，指定是否按“设置”中的设置显示“排序”菜单项。	
		True	（缺省）显示“排序”菜单项
		False	不显示“排序”菜单项
实例：			
	设置：	**#AlmDbView1.SortMenu = “False”;**	
	读取：	**tagname = #AlmDbView1.SortMenu;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

第二位排序

当单击此弹出式菜单时，显示“**第二位排序**”对话框。



此对话框显示控件中当前显示的列清单。您可以使用此对话框按升序或降序对单个列或多个列进行排序。

要指定需排序的列，选择列名旁边的复选框。使用“排序顺序”箭头键对列重新排列。

例如，如果您要先按报警状态，然后按日期进行降序排序：

1. 选择这两个复选框（“日期”和“状态”）。
2. 选择“**状态**”行。
3. 单击向上的**排序顺序**箭头键。
4. 从“**排序类型**”区域选择“**上升**”单选钮。

5. 单击“**确定**”。

激活复位菜单

此复选框用于启用控件右击菜单的“**复位菜单**”选项。复位菜单将所有列还原为设计时保存的设置。

属性：	ResetMenu		
用途：	返回或设置一个决定是否在上下文相关菜单中显示“复位”菜单项的值。		
类型：	整型		
缺省：	True		
语法：	*Object*.ResetMenu *[= Integer]*		
	Integer	整型表达式，指定是否按“设置”中的设置显示“复位”菜单项。	
		True	（缺省）显示“复位”菜单项
		False	不显示“复位”菜单项
实例：			
	设置：	**#AlmDbView1.ResetMenu = “False”;**	
	读取：	**tagname = #AlmDbView1.ResetMenu;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

显示模式

此下拉式列表框显示可用视图选项的列表。列表内容包括：

排警和事件历史

报警历史

事件历史

属性：	DisplayMode
用途：	返回控件的显示模式，此属性为**只读**属性。
类型：	字符串
缺省：	排警和事件历史
语法：	*Object*.DisplayMode
实例：	
	tagname = #AlmDbView1.DisplayMode;
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。

运行时特征复选框

“显示网格”复选框

“**显示网格**”复选框用于启用或禁用控件网格的显示。

属性：	ShowGrid		
用途：	返回或设置一个决定是否在控件中显示网格线的值。		
类型：	整型		
缺省：	False		
语法：	*Object*.ShowGrid *[= Integer]*		
	Integer	整型表达式，指定是否按“设置”中的设置显示网格线。	
		True	在控件中显示网格线
		False	（缺省）不显示网格线
实例：			
	设置：	**#AlmDbView1.ShowGrid = “True”;**	
	读取：	**tagname = #AlmDbView1.ShowGrid;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

“显示标题”复选框

“**显示标题**”复选框用于启用或禁用控件标题的显示。

属性：	ShowHeading		
用途：	返回或设置一个决定是否在控件中显示列标题的值。		
类型：	整型		
缺省：	True		
语法：	*Object*.ShowHeading *[= Integer]*		
	Integer	整型表达式，指定是否按“设置”中的设置显示列标题。	
		True	（缺省）在控件中显示列标题。
		False	不显示列标题。
实例：			
	设置：	**#AlmDbView1.Showheading = “False”;**	
	读取：	**tagname = #AlmDbView1.Showheading;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

主排序

标题为用户提供了主排序选项。单击标题可对符合最后一次查询条件的所有行进行排序，并向用户显示排序所得的第一条记录。

移动列

单击并拖动列标题，可在控件中移动列并将其与其它列并排放置。

无提示模式

选择此复选框可启用“无提示模式”选项。此属性决定控件是否处于无提示模式。当选定此复选框时，不会显示错误消息。要查看错误，可调用方法 “GetLastError” 来返回错误消息。

属性：	SilentMode		
用途：	返回或设置一个决定控件是否处于无提示模式的值。		
类型：	整型		
缺省：	False		
语法：	*Object*.SilentMode *[= Integer]*		
	Integer	整型表达式，指定控件是否处于无提示模式。	
		True	处于无提示模式。
		False	（缺省）关闭无提示模式。
实例：			
	设置：	**#AlmDbView1.SilentMode = “True”;**	
	读取：	**tagname = #AlmDbView1.SilentMode;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

“行选择”复选框

“**行选择**”复选框用于启用或禁用控件的“行选择”选项。

属性：	RowSelection		
用途：	返回或设置一个决定是否允许在运行时选择行的值。		
类型：	整型		
缺省：	True		
语法：	*Object*.RowSelection *[= Integer]*		
	Integer	整型表达式，指定是否按“设置”中的设置，允许在运行时选择行。	
		True	（缺省）允许在控件中选择行。
		False	不允许选择行。
实例：			
	设置：	**#AlmDbView1.RowSelection = “False”;**	
	读取：	**tagname = #AlmDbView1.RowSelection;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

注意：如果不允许选择行 – 即属性设置为 “False” – 则不会产生“单击”或“双击”事件。

“更改列尺寸”复选框

“**更改列尺寸**”复选框用于启用或禁用控件的列尺寸调整选项。

用来设置以下属性：

属性：	ColumnResize		
用途：	返回或设置一个用于决定是否可在运行时调整列大小的值。		
类型：	整型		
缺省：	True		
语法：	*Object*.ColumnResize *[= Integer]*		
	Integer	整型表达式，指定是否可按“设置”中的设置，在运行时调整列大小。	
		True	（缺省）可在运行时调整列大小。
		False	不能调整列大小。
实例：			
	设置：	**#AlmDbView1.ColumnResize = “False”;**	
	读取：	**tagname = #AlmDbView1.ColumnResize;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

“显示状态栏”复选框

“**显示状态栏**”复选框用于启用或禁用控件底部状态栏的显示。

用来设置以下属性：

属性：	ShowStatusBar		
用途：	返回或设置一个决定是否显示状态栏的值。		
类型：	整型		
缺省：	True		
语法：	*Object*.ShowStatusBar *[= Integer]*		
	Integer	整型表达式，指定是否按“设置”中的设置显示状态栏。	
		True	（缺省）显示状态栏。
		False	不显示状态栏。
实例：			
	设置：	**#AlmDbView1.ShowStatusBar = “False”;**	
	读取：	**tagname = #AlmDbView1.ShowStatusBar;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

状态栏

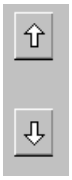
状态栏显示控件的当前状态。



左窗格显示服务器名（属性 ServerName），窗格右侧显示服务器的连接状态（属性 ConnectStatus）。中间窗格显示符合查询条件的总记录数（属性 TotalRowcount）中当前显示的记录数（属性 RowCount）。

“检索按钮”复选框

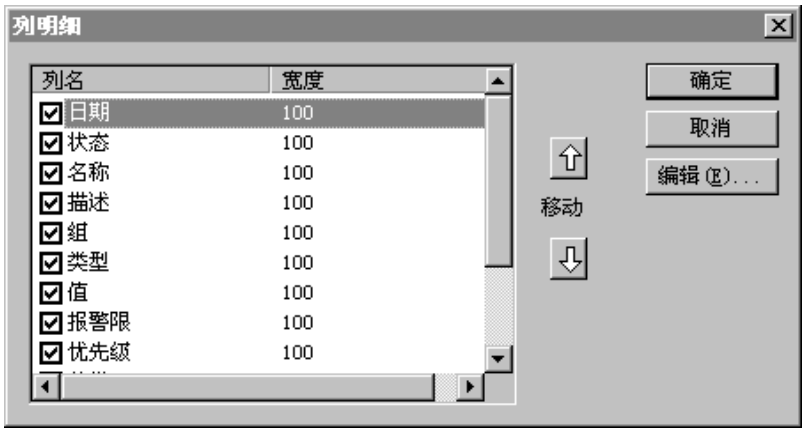
“**检索按钮**”复选框用于启用或禁用控件右侧检索按钮的显示。



属性：	ShowFetch		
用途：	返回或设置一个决定是否显示检索按钮的值。		
类型：	整型		
缺省：	True		
语法：	*Object*.ShowFetch *[= Integer]*		
	Integer	整型表达式，指定是否按“设置”中的设置显示检索按钮。	
		True	（缺省）显示检索按钮。
		False	不显示检索按钮。
实例：			
	设置：	**#AlmDbView1.ShowFetch = “False”;**	
	读取：	**tagname = #AlmDbView1.ShowFetch;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

“列明细”按钮

“列明细”对话框通过单击“属性”对话框上的“**列明细**”按钮来调用。
“列详细”对话框用于选择要显示的列，指定显示顺序并设置列名和列宽。



单击列名旁边的复选框来选择列。只有选定的列才会显示在控件中。

使用向上和向下**移动**箭头键对列重新排列。

选择列，然后单击“编辑”来调用该列的“编辑”对话框。



“编辑”对话框用于设置列标题名和列宽。

颜色页

“**颜色**”标签用于设置各种报警和事件记录的颜色。



特性名

“特性名”下拉式列表框显示可设置颜色的属性列表。列表包含下列属性：

属性：	**AckAlmBackColor**	
用途：	返回或设置已确认报警的背景颜色，此颜色应用于控件中显示的状态为 ACK_ALM 的记录。	
类型：	整型	
缺省：	Window Background Color	
语法：	*Object*.AckAlmBackColor *[= color]*	
	Color	决定背景颜色的一个值或常量。
实例：		
	设置颜色：	**#AlmDbView1.AckAlmBackColor = 0X80000005;**
	读取颜色：	**clr = #AlmDbView1.AckAlmBackColor;**
	此处的控件名是 AlmDbView1，clr 定义为整型标记名。	

属性：	AckAlmForeColor	
用途：	返回或设置已确认报警的前景颜色，此颜色应用于控件中显示的状态为 ACK_ALM 的记录。	
类型：	整型	
缺省：	蓝色	
语法：	*Object*.AckAlmForeColor *[= color]*	
	Color	决定指定对象颜色的一个值或常量。
实例：		
	设置颜色：	`#AlmDbView1.AckAlmForeColor = 0X00880000;`
	读取颜色：	`clr = #AlmDbView1.AckAlmForeColor;`
	此处的控件名是 AlmDbView1，clr 定义为整型标记名。	

属性：	AckRtnForeColor	
用途：	返回或设置已确认和已返回报警的前景颜色，此颜色应用于控件中显示的状态为 ACK_RTN 的记录。	
类型：	整型	
缺省：	绿色	
语法：	*Object*.AckRtnForeColor *[= color]*	
	Color	决定指定对象颜色的一个值或常量。
实例：		
	设置颜色：	`#AlmDbView1.AckRtnForeColor = 0X00008000;`
	读取颜色：	`clr = #AlmDbView1.AckRtnForeColor;`
	此处的控件名是 AlmDbView1，clr 定义为整型标记名。	

属性：	AckRtnBackColor	
用途：	返回或设置已确认和已返回报警的背景颜色，此颜色应用于控件中显示的状态为 ACK_RTN 的记录。	
类型：	整型	
缺省：	窗口背景色	
语法：	*Object*.AckRtnBackColor *[= color]*	
	Color	决定指定对象颜色的一个值或常量。
实例：		
	设置颜色：	`#AlmDbView1.AckRtnBackColor = 0X80000005;`
	读取颜色：	`clr = #AlmDbView1.AckRtnBackColor;`
	此处的控件名是 AlmDbView1，clr 定义为整型标记名。	

属性：	EventForeColor	
用途：	返回或设置事件报警的前景颜色，此颜色应用于控件中显示的状态为 EVT_EVT 的记录。	
类型：	整型	
缺省：	品红色	
语法：	*Object*.EventForeColor *[= color]*	
	Color	决定指定对象颜色的一个值或常量。
实例：		
	设置颜色：	`#AlmDbView1.EventForeColor = 0X00800080;`
	读取颜色：	`clr = #AlmDbView1.EventForeColor;`
	此处的控件名是 AlmDbView1，clr 定义为整型标记名。	

属性：	EventBackColor	
用途：	返回或设置事件报警的背景颜色，此颜色应用于控件中显示的状态为 EVT_EVT 的记录。	
类型：	整型	
缺省：	窗口背景色	
语法：	*Object*.EventBackColor *[= color]*	
	Color	决定指定对象颜色的一个值或常量。
实例：		
	设置颜色：	`#AlmDbView1.EventBackColor = 0X80000005;`
	读取颜色：	`clr = #AlmDbView1.EventBackColor;`
	此处的控件名是 AlmDbView1，clr 定义为整型标记名。	

属性：	UnAckAlmForeColor	
用途：	返回或设置未确认报警的前景颜色，此颜色应用于控件中显示的状态为 UNACK_ALM 的记录。	
类型：	整型	
缺省：	红色	
语法：	*Object*.UnAckAlmForeColor *[= color]*	
	Color	决定指定对象颜色的一个值或常量。
实例：		
	设置颜色：	`#AlmDbView1.UnAckAlmForeColor = 0X00800080;`
	读取颜色：	`clr = #AlmDbView1.UnAckAlmForeColor;`
	此处的控件名是 AlmDbView1，clr 定义为整型标记名。	

属性：	UnAckAlmBackColor	
用途：	返回或设置未确认报警的背景颜色，此颜色应用于控件中显示的状态为 UNACK_ALM 的记录。	
类型：	整型	
缺省：	窗口背景色	
语法：	*Object*.UnAckAlmBackColor *[= color]*	
	Color	决定指定对象颜色的一个值或常量。
实例：		
	设置颜色：	`#AlmDbView1.UnAckAlmBackColor = 0X80000005;`
	读取颜色：	`clr = #AlmDbView1.UnAckAlmBackColor;`
	此处的控件名是 AlmDbView1，clr 定义为整型标记名。	

属性：	UnAckRtnForeColor	
用途：	返回或设置未确认和已返回报警的前景颜色，此颜色应用于控件中显示的状态为 UNACK_RTN 的记录。	
类型：	整型	
缺省：	窗口文本	
语法：	*Object*.UnAckRtnForeColor *[= color]*	
	Color	决定指定对象颜色的一个值或常量。
实例：		
	设置颜色：	`#AlmDbView1.UnAckRtnForeColor = 0X80000005;`
	读取颜色：	`clr = #AlmDbView1.UnAckRtnForeColor;`
	此处的控件名是 AlmDbView1，clr 定义为整型标记名。	

属性：	UnAckRtnBackColor	
用途：	返回或设置未确认和已返回报警的背景颜色，此颜色应用于控件中显示的状态为 UNACK_RTN 的记录。	
类型：	整型	
缺省：	窗口背景色	
语法：	*Object.*UnAckRtnBackColor *[= color]*	
	Color	决定指定对象颜色的一个值或常量。
实例：		
	设置颜色：	**#AlmDbView1.UnAckRtnBackColor = 0X80000005;**
	读取颜色：	**clr = #AlmDbView1.UnAckRtnBackColor;**
	此处的控件名是 AlmDbView1，clr 定义为整型标记名。	

系统颜色

基于颜色的所有属性显示在“**颜色**”属性页中。您可以使用调色板来选择每种属性的颜色，也可以通过从“**系统颜色**”下拉式列表框中选择某种系统颜色，来选择系统中已定义的系统颜色。

字体页

“**字体**”标签用于设置控件中记录和标题的显示字体。



数据库页

“数据库”标签用于设置服务器名及其相关属性，以使控件可在需要时连接到数据库。



服务器名

“**服务器名**”下拉式列表框包括网络中可用的 SQL 服务器和 MSDE 服务器名列表。用户可以编辑此列表框。此列表框用来设置以下属性：

属性：	ServerName	
用途：	返回或设置控件所连接到的用于获取数据的服务器名。	
类型：	消息型	
语法：	*Object.*ServerName *[= text]*	
实例：		
	设置：	**#AlmDbView1.ServerName = “Prod1”;**
	读取：	**tagname = #AlmDbView1.ServerName;**
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。“Prod1” 是 Alarm DB Logger 用以记录数据的机器名。	

用户 ID

“**用户 ID**”字段用于输入数据库服务器的用户名，以便建立所需的连接。此处用来设置以下属性：

属性：	UserID	
用途：	返回或设置当控件连接到 SQL 服务器以获取数据时所使用的用户 ID。	
类型：	消息型	
语法：	*Object.UserID [= text]*	
	Text	对用户 ID 求值的字符串表达式。
实例：		
	设置：	**#AlmDbView1.UserID = “sa”;**
	读取：	**Tagname = #AlmDbView1.UserID;**
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。“sa” 是用于连接到 SQL 服务器的用户 ID。	

口令

“**口令**”字段用于输入数据库服务器用户名的口令，以便建立所需的连接。此处用来设置以下属性：

属性：	Password	
用途：	返回或设置当控件连接到 SQL 服务器以获取数据时所使用的口令。	
类型：	消息型	
语法：	*Object.*Password *[= text]*	
	text	对口令求值的字符串表达式。
实例：		
	设置：	**AlmDbView1.Password = “pswd”;**
	读取：	**tagname = AlmDbView1.Password;**
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。“pswd” 是用于连接到 SQL 服务器的口令。	

测试连接

“**测试连接**”按钮用于在设计时使用给定的服务器名、用户名和口令来测试与 WWALMDB 数据库的连接。

“自动连接”复选框

缺省条件下“自动连接”复选框是被禁用的。一旦填入服务器名和用户 ID 字段，此复选框将变为启用状态。您可以使用此复选框来启用或禁用运行时控件到服务器的数据库自动连接。此复选框用来设置以下属性：

属性：	AutoConnect		
用途：	返回或设置一个决定当控件一进入运行模式时是否连接到数据库的值。		
类型：	整型		
缺省：	False		
语法：	*Object*.AutoConnect *[= Integer]*		
	Integer	整型表达式，指定控件是否按“设置”中的设置，在一进入运行模式时立即连接到数据库。	
		True	连接到数据库。
		False	（缺省）不连接到数据库。
实例：			
	设置：	**#AlmDbView1.AutoConnect = “False”;**	
	读取：	**tagname = #AlmDbView1.AutoConnect;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

注意：如果此属性在设计时设置为 “False”，控件在运行模式下将不会连接到数据库。必须执行对 “Connect” 方法的显式调用来连接到数据库。

“选择”标签

“**选择**”标签提供了有关需要根据时间来检索的记录组的详细信息，还给出了显示的最大记录数、是否显示日期和时间以及日期和时间的显示格式。



“日期”复选框

“**日期**”复选框用于启用或禁用在控件的日期列中显示记录的日期详细信息。此复选框用来设置以下属性：

属性：	ShowDate		
用途：	返回或设置一个决定控件是否显示日期的值。		
类型：	整型		
缺省：	True		
语法：	*Object*.ShowDate *[= Integer]*		
	Integer	整型表达式，指定是否按“设置”中的设置显示日期。	
		True	（缺省）以 “Date” 属性中指定的格式显示日期。
		False	不显示日期。
实例：			
	设置：	**#AlmDbView1.ShowDate = “False”;**	
	读取：	**tagname = #AlmDbView1.ShowDate;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

“日期”列表框

一旦选定“**日期**”复选框，将启用“日期”列表框。该列表框包括各种日期显示模式。用来设置以下属性：

属性：	**Date**	
用途：	返回或设置控件用以显示日期的日期格式。	
类型：	消息型	
缺省：	“DD MMM”	
语法：	*Object*.Date *[= text]*	
	Text	对日期格式求值的字符串表达式。
实例：		
	设置：	**#AlmDbView1.Date = “DD/MM/YY”;**
	读取：	**tagname = #AlmDbView1.Date;**
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。	

备注

注意：此属性必须具有以下字符串之一：

DD MMM

DD MMM YYYY

DD/MM

DD/MM/YY

MM/DD

MM/DD/YYYY

MMM DD

MMM DD YYYY

YY/MM/DD

YYYY/MM/DD

“时间”复选框

“**时间**”复选框用于启用或禁用在控件的日期列中显示记录的时间详细信息。用来设置以下属性：

<table>
<tr><td>属性：</td><td colspan="3">ShowTime</td></tr>
<tr><td>用途：</td><td colspan="3">返回或设置一个决定控件是否显示时间的值。</td></tr>
<tr><td>类型：</td><td colspan="3">整型</td></tr>
<tr><td>缺省：</td><td colspan="3">True</td></tr>
<tr><td>语法：</td><td colspan="3">Object.ShowTime [= Integer]</td></tr>
<tr><td></td><td>Integer</td><td colspan="2">整型表达式，指定是否按“设置”中的设置显示日期。</td></tr>
<tr><td></td><td></td><td>True</td><td>（缺省）以 “Time” 属性中指定的格式显示时间。</td></tr>
<tr><td></td><td></td><td>False</td><td>不显示时间。</td></tr>
<tr><td>实例：</td><td colspan="3"></td></tr>
<tr><td></td><td>设置：</td><td colspan="2">#AlmDbView1.ShowTime = “False”;</td></tr>
<tr><td></td><td>读取：</td><td colspan="2">tagname = #AlmDbView1.ShowTime;</td></tr>
<tr><td></td><td colspan="3">此处的控件名是 AlmDbView1，标记名定义为整型标记名。</td></tr>
</table>

注意：“ShowDate” 或 “ShowTime” 之一必须设置为 1，但不能设置为 “False”。如果二者均已设置，则时间将跟在日期之后。

“时间”列表框

一旦选定“**时间**”复选框，将启用“时间”列表框。该列表框包括各种时间显示模式。用来设置以下属性：

属性：	**Time**	
用途：	返回或设置用于显示控件中的时间的格式。	
类型：	消息型	
缺省：	“HH:MM”	
语法：	*Object.Time [= text]*	
	Text	对时间格式求值的字符串表达式。
实例：		
	设置：	**#AlmDbView1.Time = “HH:MM:SS”;**
	读取：	**tagname = #AlmDbView1.Time;**
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。	

备注

注意：此属性必须具有以下字符串之一：

HH:MM

HH:MM AP

HH:MM:SS

HH:MM:SS AP

HH:MM:SS.SSS

HH:MM:SS.SSS AP

“使用指定时间”复选框

此复选框用于设置用户是为要检索的记录输入指定的开始和结束时间，还是根据“**Duration**”属性来计算开始和结束时间。此处用来设置以下属性：

<table>
<tr><td>属性：</td><td colspan="3">SpecificTime</td></tr>
<tr><td>用途：</td><td colspan="3">返回或设置一个决定控件是使用 “start time” 和 “end time” 属性，还是根据 “Duration” 属性值来计算开始时间和结束时间的值。</td></tr>
<tr><td>类型：</td><td colspan="3">整型</td></tr>
<tr><td>缺省：</td><td colspan="3">False</td></tr>
<tr><td>语法：</td><td colspan="3">Object.SpecificTime [= Integer]</td></tr>
<tr><td></td><td>整型</td><td colspan="2">整型表达式，指定控件是否按“设置”中的设置，使用 “start time” 和 “end time” 属性。</td></tr>
<tr><td></td><td></td><td>True</td><td>使用 “Start time” 和 “End time” 属性。</td></tr>
<tr><td></td><td></td><td>False</td><td>（缺省）根据 “Duration” 属性来计算开始时间和结束时间。</td></tr>
<tr><td>实例：</td><td colspan="3"></td></tr>
<tr><td></td><td>设置：</td><td colspan="2">#AlmDbView1.SpecificTime = “True”;</td></tr>
<tr><td></td><td>读取：</td><td colspan="2">tagname = #AlmDbView1.SpecificTime;</td></tr>
<tr><td></td><td colspan="3">此处的控件名是 AlmDbView1，标记名定义为整型标记名。</td></tr>
</table>

持续时间

一旦撤消选定“**使用指定时间**”复选框，将启用“**持续时间**”下拉式列表框。该列表框提供了一个相对当前时间的预定义时间间隔列表。此处用来设置以下属性：

属性：	Duration	
用途：	返回或设置控件用于设置开始时间和结束时间的持续时间。	
类型：	消息型	
缺省：	“最后一小时”	
语法：	*Object*.Duration *[= text]*	
	text	对持续时间求值的字符串表达式。
实例：		
	设置：	**#AlmDbView1.Duration = “Last Month”;**
	读取：	**Tagname = #AlmDbView1.Duration;**
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。	

备注

注意：此属性必须具有以下字符串之一：

最后一分钟

最后 5 分钟

最后 15 分钟

最后半小时

最后一小时

最后 2 小时

最后 4 小时

最后 8 小时

最后 12 小时

最后一天

最后 2 天

最后 3 天

最后一星期

最后 2 星期

最后一个月

最后 3 个月

开始时间

仅当选定“**使用指定时间**”复选框时，此日期-时间控件才被启用。此控件用于设置检索报警记录的时间间隔的开始时间。此处用来设置以下属性：

属性：	**StartTime**	
用途：	返回或设置开始时间。	
类型：	消息型	
语法：	*Object*.StartTime *[= text]*	
	text	对开始时间求值的字符串表达式。
实例：		
	设置：	**#AlmDbView1.StartTime= “3/21/2001 19:14:06”;**
	读取：	**tagname = #AlmDbView1.StartTime;**
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。	

注意：返回的字符串为 **Date** 和 **Time** 属性中设置的格式。要设置开始时间值，字符串必须为 MM/DD/YYYY HH:MM:SS 格式。此属性可处理从 1970 年 1 月 1 日午夜到 2038 年 1 月 18 日 19:14:07 的任意时区中的日期。

结束时间

仅当选定“**使用指定时间**”复选框时，此日期-时间控件才被启用。此控件用于设置检索报警记录的时间间隔的结束时间。此处用来设置以下属性：

属性：	**EndTime**	
用途：	返回或设置结束时间。	
类型：	消息型	
语法：	*Object*.EndTime *[= text]*	
	Text	对结束时间求值的字符串表达式。
实例：		
	设置：	**#AlmDbView1.EndTime = “3/21/2001 21:14:06”;**
	读取：	**tagname = #AlmDbView1.EndTime;**
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。	

注意：返回的字符串为 **Date** 和 **Time** 属性中设置的格式。要设置结束时间值，字符串必须为 MM/DD/YYYY HH:MM:SS 格式。此属性可处理从 1970 年 1 月 1 日午夜到 2038 年 1 月 18 日 19:14:07 的任意时区中的日期。

最大记录数

“**最大记录数**”字段用于编辑控件一个实例中可显示的记录数。用来设置以下属性：

属性：	MaxRecords	
用途：	返回或设置一个用于指定给定时间需获取的最大记录数的值。	
类型：	整型	
缺省：	100	
语法：	*Object*.MaxRecords *[=integer]*	
	Integer	整型表达式，指定给定时间需获取的记录数。
实例：		
	To Set:	**#AlmDbView1.MaxRecords = 500;**
	To Get:	**tagname = #AlmDbView1.MaxRecords;**
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。	

注意：最大记录数的范围从 1 到 1000。为保持最佳性能，将此值设置得越小越好。

“查询”标签

“**查询**”标签用于提供有关供应器名、组名、从优先级、到优先级的详细信息，以便可以根据用户的首选项来检索记录。



供应器名

“**供应器名**”字段用于全部或部分地输入供应器名。用来设置以下属性：

属性：	ProviderName	
用途：	返回或设置供应器名。检索给定供应器数据库中的数据。	
类型：	消息型	
语法：	*Object*.ProviderName *[= text]*	
	text	对供应器名求值的字符串表达式。
实例：		
	设置：	**#AlmDbView1.ProviderName = “Prov1/InTouch";**
	读取：	**tagname = #AlmDbView1.ProviderName;**
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。	

“精确匹配”复选框

仅当在“供应器名”文本框中输入内容时，“**精确匹配**”复选框才被启用。此复选框用于启用或禁用用户所给供应器名与数据库中的供应器名的比较。用来设置以下属性：

属性：	ProviderExactMatch		
用途：	返回或设置的值将决定控件是检索给定供应器，还是检索供应器名子字符串下的所有供应器的数据。		
类型：	整型		
缺省：	False		
语法：	*Object*.ProviderExactMatch *[= Integer]*		
	Integer	指定数据方法的整型表达式。按“设置”所述检索标记名。	
		True	只检索由 “Provider Name” 属性指定的供应器的数据。
		False	（缺省）检索其名称是 “Provider Name” 属性的子字符串的所有供应器的数据。
实例：			
	设置：	**#AlmDbView1.ProviderExactMatch = “True”;**	
	读取：	**tagname = #AlmDbView1.ProviderExactMatch;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

组名

“**组名**”字段用于全部或部分地输入组名。用来设置以下属性：

属性：	GroupName	
用途：	返回或设置组名。检索给定报警组数据库中的数据。	
类型：	消息型	
语法：	*Object*.GroupName *[= text]*	
	text	对组名求值的字符串表达式。
实例：		
	设置：	**#AlmDbView1.GroupName = “$System”;**
	读取：	**tagname = #AlmDbView1.GroupName;**
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。	

“精确匹配”复选框

仅当在“组名”文本框中输入内容时，此“**精确匹配**”复选框才被启用。此复选框用于启用或禁用用户所给组名与数据库中的组名的比较。用来设置以下属性：

属性：	GroupExactMatch		
用途：	返回或设置的值将决定控件是检索给定组，还是检索组名子字符串下的所有组的数据。		
类型：	整型		
缺省：	False		
语法：	*Object*.ProviderExactMatch *[= Integer]*		
	Integer	指定数据方法的整型表达式，按“设置”所述检索标记名。	
		True	只检索由 “Group Name” 属性指定的报警组的数据。
		False	（缺省）检索其名称是“**Group Name**” 属性的子字符串的所有报警组的数据。
实例：			
	设置：	**#AlmDbView1.GroupExactMatch = “True”;**	
	读取：	**tagname = #AlmDbView1.GroupExactMatch;**	
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。		

从优先级

“**从优先级**”字段用于指定最高优先级，以便可以检索查看该优先级以上的记录。用来设置以下属性：

属性：	**FromPriority**	
用途：	返回或设置的值决定需要检索的报警记录的优先级下限。	
类型：	整型	
缺省：	1	
语法：	*Object*.FromPriority *[= integer]*	
	Integer	整型表达式，指定需要检索的报警的优先级下限。
实例：		
	设置：	**#AlmDbView1.FromPriority = 500;**
	读取：	**tagname = #AlmDbView1.FromPriority;**
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。	

注意：此优先级范围从 1 到 999，并且必须小于或等于 “**ToPriority**” 属性。InTouch 的优先级范围从 1（最高）到 999（最低）。

到优先级

“**到优先级**”字段用于指定最低优先级，以便可以检索查看该优先级以下的记录。用来设置以下属性：

属性：	**ToPriority**	
用途：	返回或设置的值决定需要检索的报警记录的优先级上限。	
类型：	整型	
缺省：	999	
语法：	*Object.*ToPriority *[= integer]*	
	Integer	整型表达式，指定需要检索的报警的优先级上限。
实例：		
	设置：	**#AlmDbView1.ToPriority = 889;**
	读取：	**tagname = #AlmDbView1.ToPriority;**
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。	

注意：此优先级范围从 1 到 999，并且必须大于或等于 “**FromPriority**” 属性。InTouch 的优先级范围从 1（最高）到 999（最低）。

属性页

“属性”标签用于将标记名指定给此控件的各种属性。



通用属性

属性：	RowCount
用途：	返回控件中当前显示的记录数。此属性为**只读**属性。
类型：	数字型
语法：	*Object*.RowCount
实例：	
	***tagname* = #AlmDbView1.RowCount;**
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。

属性：	TotalRowCount
用途：	返回当前查询的总记录数。此属性为**只读**属性。
类型：	整型
语法：	*Object*.TotalRowCount
实例：	
	tagname = #AlmDbView1.TotalRowCount;
	此处的控件名是 AlmDbView1，标记名定义为整型标记名。

注意：RowCount 是当前查询所返回的行数，一般与 MaxRecords 属性相同，但当检索到的的记录数小于 MaxRecords 属性时，二者将不相同。例如，如果符合特定查询标准的记录数为 950，而最大记录数为 100，则最后一页将显示 50 条记录，并且行数为 50。在相同的实例中，TotalRowCount 始终为 950。

属性：	ConnectStatus
用途：	返回连接的状态。此属性为**只读**属性。
类型：	消息型
语法：	*Object*.ConnectStatus
实例：	
	Tagname **= #AlmDbView1.ConnectStatus;**
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。

返回值：

返回值	**描述**
Connected	控件已连接到数据库。
Not Connected	控件未连接到数据库。
In Progress	控件正在连接到数据库。

方法和事件

方法

Method:	Connect
用途：	将控件连接到数据库，如果连接成功，则显示范围 1 到 MaxRecords 的记录集。
语法：	*Object*.Connect
实例：	
	`#AlmDbView1.Connect();`
	此处的控件名是 AlmDbView1。

方法：	Disconnect
用途：	断开控件与数据库的连接。
语法：	*Object*.Disconnect
实例：	
	`#AlmDbView1.Disconnect();`
	此处的控件名是 AlmDbView1。

方法：	Refresh
用途：	从数据库刷新控件，如果连接成功，则显示范围 1 到 MaxRecords 的记录集。
语法：	*Object*.Refresh
实例：	
	`#AlmDbView1.Refresh();`
	此处的控件名是 AlmDbView1。

.

方法：	Reset
用途：	将所有列重置为设计时保存的设置。
语法：	*Object*.Reset
实例：	
	`#AlmDbView1.Reset();`
	此处的控件名是 AlmDbView1。

方法：	ShowSort
用途：	在启用 “SortMenu” 属性的情况下，显示 “第二位排序” 对话框。
语法：	*Object*.ShowSort
实例：	
	`#AlmDbView1.ShowSort();`
	此处的控件名是 AlmDbView1。

方法：	ShowContext

用途：	在启用 “RefreshMenu” 或 “ResetMenu” 或 “SortMenu” 等任一属性的情况下，显示上下文相关菜单。
语法：	*Object*.ShowContext
实例：	
	#AlmDbView1.ShowContext();
	此处的控件名是 AlmDbView1。

方法：	**GetItem**	
用途：	返回指定行和列的数据为一字符串。	
语法：	*Object*.GetItem*(Integer, message)*	
	Integer	对控件中的指定行求值的整型表达式。
	Message	对控件中的列名求值的字符串表达式。
实例：		
	***tagname* = #AlmDbView1.GetItem(1, “Group”);**	
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。	

注意：行号从零开始。

方法：	**GetSelectedItem**	
用途：	返回指定行和列的数据为一字符串	
语法：	*Object*.GetSelectedItem*(message)*	
	Message	对控件中的列名求值的字符串表达式。
实例：		
	***tagname* = #AlmDbView1.GetSelectedItem(“State”);**	
	此处的控件名是 AlmDbView1，标记名定义为消息型标记名。	

方法：	GetNext
用途：	检索数据库中的下一组记录（如果有的话）。
语法：	*Object.*GetNext
实例：	
	#AlmDbView1.GetNext();
	此处的控件名是 AlmDbView1。

方法：	GetPrevious
用途：	检索数据库中的上一组记录（如果有的话）。
语法：	*Object.*GetPrevious
实例：	
	#AlmDbView1.GetPrevious();
	此处的控件名是 AlmDbView1。

GetLastError

选择此复选框以返回无提示模式下的最后一条错误消息。

方法：	GetLastError	
用途：	返回无提示模式下的最后一条错误消息。	
语法：	*Object. GetLastError()*	
实例：		
	读取：	**Tagname = #AlmDbView1.GetLastError();**
	此处的控件名是 AlmDbView1，标记名定义为变量或字符串。	

方法：	AboutBox
用途：	显示“关于”对话框。
语法：	*Object.*AboutBox
实例：	
	#AlmDbView1.AboutBox();
	此处的控件名是 AlmDbView1。

事件页

“**事件**”标签用于将某些脚本函数指定给此控件上发生的各种事件。



事件：	Click	
用途：	当用户在控件上单击并释放鼠标左键后出现。	
语法：	*tagname* = *#Thisevent*.clicknRow	
	nRow	整型表达式，反映发生事件所在的行。
	#Thisevent	关键字

事件：	DoubleClick	
用途：	当用户在控件上双击并释放鼠标左键后出现。	
语法：	*Tagname* = *#ThisEvent*.DoubleClick*nRow*	
	NRow	整型表达式，反映发生事件所在的行。
	#Thisevent	关键字

热备份与同步

InTouch 分布式报警系统为报警供应器应用程序如 InTouch 等，提供了在网络上发出报警通知及接收报警确认的能力。它也为远程节点上的报警接收器应用程序（客户端）提供了查询、显示和确认这些报警的能力。在分布式报警环境下，可能有许多报警供应器和报警接收器通过网络连接在一起。

此外，有可能将某些报警供应器配置为系统中其它报警供应器的***备份***。使用备份报警供应器的目的是建立一个安全保障系统 – 当***主***报警供应器出现故障时，***辅助***报警供应器能够接管其工作来发出报警通知，从而确保能够在某些关键环境下产生报警。例如，您可以设置两个独立的计算机节点，这两个节点运行同样的 InTouch 应用程序，与相同的设备连接，甚至还可能使用冗余的一组 I/O 连接。或者，主报警供应器可能是一个“全功能”的应用程序，而辅助供应器则是一个“最小化”的系统，只处理最关键的部分来保证设备正常工作和免受损坏。通常，只有一些关键的报警供应器才会采用辅助报警供应器来作为备用。

要设置备份配置，您可以定义一个***热备份对***，指定其***名称***并分别标识为***主***报警供应器和***辅助***报警供应器。

注意：只有当主供应器处于不活动状态时，才会显示来自辅助供应器的报警。

备份配置的基础是备份接收器（客户端）使用一个引用***热备份对***名的报警查询。对于该***备份对***，客户端只看到一组报警，即来自主供应器的报警，如果主供应器出现故障，则获得来自辅助供应器的报警。此外，主供应器和辅助供应器上的报警确认是完全***同步***的。这就意味着，如果报警在主供应器上获得确认，则它也在辅助供应器上得到了确认。

InTouch 分布式报警系统的热备份与同步特性包括：

1. 提供一个配置公用程序，简化建立报警备份对、以及指定主供应器和辅助供应器的过程。
2. 提供一个用于执行***报警映射***的配置公用程序。
3. 内置的报警确认同步。
4. 报警供应器开机/关机过程中的同步交换。

有关热备份对的注意事项

1. 热备份仅支持 InTouch 7.11，不支持其它客户端。
2. 扩展的摘要报警或或面向事件的报警不支持热备份。
3. 如果您让报警客户端（分布式报警显示对象）查询热备份，随后又再次单独查询备份对的主供应器，则分布式报警显示对象会显示重复的记录。
4. 供应器不得配置为一个以上热备份对的主或辅助供应器。
5. 如果在确认主供应器的记录后，再次出现辅助供应器（当确认发生时刚好停机）的记录，则辅助供应器端的已确认记录的时间标签将是辅助供应器的时间标签，而与主供应器不同。
6. 查询热备份对的报警客户端（分布式报警显示对象）会在相应的列中显示配对名作为供应器，而不显示供应器的节点名。
7. 您可以选择“**设计时**”和“**运行时**”报警记录字段的任意组合来进行映射，但切记要确保映射不会导致多重引用。
8. 当映射“**值**”和“**极限**”关键字段时，数值会被舍入为四位小数位后再行映射。
9. 不具有特定设计时和运行时映射组合的报警记录将使用缺省的运行时映射。
10. 充当热备份客户端的本地节点不能配置为一个主备份供应器。
11. 报警供应器热备份对不得使用从中进行配置的客户端节点作为一个报警供应器。使用热备份公用程序的备份对，其主和辅助报警供应器必须不同于从中进行配置的系统。

热备份配置

热备份同步公用程序负责所有必要的热备份配置。当安装 InTouch 时，热备份公用程序会自动添加到 WindowMaker 的应用程序浏览器中。

➢ **要配置热备份：**

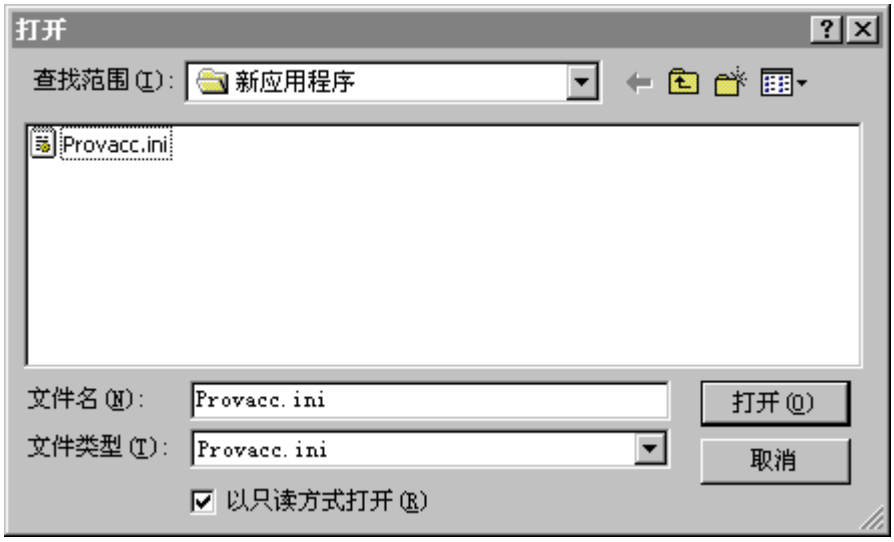
1. 启动 WindowMaker。
2. 双击应用程序浏览器中的 Alarm Hot Backup Manager。“**Alarm Hot Backup Manager**”对话框出现：



注意：缺省条件下，公用程序会自动检查上次打开的 InTouch 应用程序文件夹中的 provacc.ini 文件。如果文件可用，则公用程序会用该文件来进行自我配置。否则，公用程序会在要求的文件夹中创建一个新文件。

➢ **要更改缺省 INI 文件：**

1. 启动 WindowMaker。
2. 双击应用程序浏览器中的 Alarm Hot Backup Manager 图标。“**Alarm Hot Backup Manager**”对话框出现。
3. 在“**文件**”菜单上，选择“**打开**”命令。“**打开**”对话框出现，列出所有可用的 provacc.ini 文件（此对话框只允许您选择一个 provacc.ini 文件）。



4. 如有必要，改变路径，然后单击“**打开**”（建议将 provacc.ini 文件放在应用程序目录中）。

➢ **要添加新的热备份对：**

1. 启动 WindowMaker。
2. 双击应用程序浏览器中的 **Alarm Hot Backup Manager**。“**Alarm Hot Backup Manager**”对话框出现。
3. 单击“**新建对**”。“**添加新对**”对话框出现：



4. 在“**热备份对名**”框中，输入新备份对的唯一名称。备份对名最长可为 32 个字母数字字符。

注意：备份对名**不**允许重复。下划线 (_) 是唯一允许位于热备份对名开头、中间和结尾的特殊字符。

---

5. 在“**主节点**”组的“**名称**”框中，输入运行主供应器应用程序的计算机名称。节点名最长可为 32 个字母数字字符。

注意：特定应用程序的节点名在整个热备份列表中必须是唯一的。如果您输入了错误或重复的名称，会出现相应的错误消息框。

---

6. 在“**主节点**”组的“**供应器**”框中，输入报警供应器的名称，例如 InTouch。供应器名最长可为 32 个字母数字字符。

7. 在“**主节点**”组的“**组**”框中，输入报警组名以查询来自主供应器的报警，例如输入 $System。报警组名最长可为 32 个字母数字字符。
8. 在“**备份节点**”组的“**名称**”框中，输入运行备份供应器应用程序的计算机名称。节点名最长可为 32 个字母数字字符。

注意：特定应用程序的节点名在整个热备份列表中必须是唯一的。如果您输入了错误或重复的名称，会出现相应的错误消息框。

9. 在“**备份节点**”组的“**供应器**”框中，输入报警的备份供应器名称，例如 InTouch。供应器名最长可为 32 个字母数字字符。
10. 在“**备份节点**”组的“**组**”框中，输入报警组名以查询来自备份供应器的报警，例如输入 $System。报警组名最长可为 32 个字母数字字符。

注意：辅助报警供应器和热备份客户机不能位于同一节点中。

11. 单击“**确定**”。

注意：您必须执行“**文件**”菜单上的“**保存**”命令，以保存对 INI 文件所做的所有更改。在退出公用程序之前，系统会提示您保存更改。

➢ **要编辑现有的热备份对：**

1. 启动 WindowMaker。

   双击应用程序浏览器中的 **Alarm Hot Backup Manager**。“**Alarm Hot Backup Manager**”对话框出现。

2. 选择您要修改的热备份对名,然后单击“**修改对**”。“**配置热备份对**”对话框出现：



注意：特定应用程序的节点名在整个热备份列表中必须是唯一的。如果您输入了错误或重复的名称，会出现相应的错误消息框。

有关此对话框中的字段的详细信息，请参阅“添加新的热备份对”章节。

3. 编辑必要的字段，然后单击“**确定**”。

注意：您必须执行“**文件**”菜单上的“**保存**”命令，以保存对 INI 文件所做的所有更改。在退出公用程序之前，系统会提示您保存更改。

➢ **要删除现有的热备份对：**

1. 启动 WindowMaker。

   ☞ 双击应用程序浏览器中的 **Alarm Hot Backup Manager**，“**Alarm Hot Backup Manager**”对话框出现。

2. 选择您要删除的热备份对，然后单击“**删除对**”。屏幕出现一个要求您确认删除的对话框。

3. 单击“**是**”。

注意：您必须执行“**文件**”菜单上的“**保存**”命令，以保存对 INI 文件所做的所有更改。在退出公用程序之前，系统会提示您保存更改。

设置报警记录的关键码域

要实现同步，您必须确定报警记录域的组合，以使这些域的组合值可唯一产生一个映射关键码。组合映射关键码域可以是设计时和运行时的结合。

➢ **要选择报警记录关键码域：**

1. 启动 Alarm Hot Backup Manager 公用程序。

   您可以通过在 WindowMaker 的应用程序浏览器中双击它来启动。

2. 选择热备份对，然后单击“**设置关键码域**”。“**选定关键码域**”对话框出现：



3. 在“**报警记录域**”组中，选择您要纳入映射关键码列表的域（所选域将添加到“**已选定的域**”列表框中）。

4. 如果某个报警记录域的可能值在应用程序设计时不同并且已知，请为这个域选择“**设计时**”选项。例如，在 InTouch 供应器的情况下，“名称”域值在设计时已知，因为它会采用主和备份节点应用程序中定义的标记名值。缺省条件下，“**报警名**”和“**运行时**”是预先选定的。

注意：“**报警记录域**”可在运行时或设计时配置。所有设计时域会转入任意标记名映射的“**映射报警记录**”对话框。

有关详细信息，请参阅“映射报警记录”。

5. 如果某个报警记录域的可能值在主和备份供应器中均相同，请为这个域选择“**运行时**”选项。缺省条件下，“**报警名**”和“**运行时**”是预先选定的。

注意：在运行时，如果*设计时域*是组合关键码的一部分，并且主供应器不能在映射表中找到该关键码的映射项，则热备份会在备份供应器中自动查找具有相同标记名的报警，并发送对相应报警的确认，反之亦然。

6. 单击“**确定**”。

注意：您必须执行“**文件**”菜单上的“**保存**”命令，以保存对 INI 文件所做的所有更改。在退出公用程序之前，系统会提示您保存更改。

映射报警记录

每次主和辅助报警供应器为不同的应用程序时，您都需要映射报警供应器。例如，在主供应器是“全功能”的应用程序而辅助供应器是“最小化”的应用程序时，报警名称可能不同。报警供应器映射可建立主供应器与辅助供应器之间的通信，以便报警在某个供应器上获得确认时，分布式报警系统知道其它供应器上要确认的报警。

注意：映射从一个 .csv（逗号分隔的变量）文件导入，该文件需要使用简单的文本文件编辑器或 Microsoft Excel 创建。报警域名必须与“**映射报警记录**”对话框中包含的名称匹配。否则，导入过程会中止并显示一个警告消息框。

每个热备份对由应用程序层的唯一名称确定，例如：

热备份对名	主节点!供应器	备份节点!供应器
BoilerRoom1	NodeA!InTouch	NodeX!InTouch

注意：为了简化工作，如果未提供映射，分布式报警系统假定同一报警同时来自主和辅助供应器上。

您可以合并报警记录的域值，如报警组、报警名和优先级，以产生一个唯一确定报警记录的“组合映射关键码”。

InTouch 报警供应器会为“名称”域赋予产生报警的标记名的名称。因此，对于给定的热备份对，会使用组和“名称”域的组合来产生一个映射关键码，

例如：

供应器节点	备份节点
$System!TagA	$System!TagB

如果供应器结合名称域和注释域作为一个唯一域，则映射关键码可以是名称和注释的组合。

供应器节点	备份节点
tagA!CommentA	tagB!CommentB

对于第三方供应器的任何其它域组合，也可以使用此方法。

导入文件所需要的详细信息

下列情况会阻止文件导入。

所需的列数应填入导入文件的所有记录值，任何记录都不得少出或多出。

导入文件的标题应与“映射报警记录”对话框的标题相同，并且按照相同的顺序。

在下列情况下，如果导入的记录有误，系统会提示用户忽略该特定的记录号，或者中止导入过程。

“组”列值不能包含空格。

“名称”列值不能包含空格。

“类”列值不得是 VALUE、DEV、ROC 和 DSC 之外的任何值。

“类型”列值不得是 LOLO、LO、HI、HIHI、MinDev、MajDev、ROC 和 DSC 之外的任何值。

“优先级”列值应为从 1 到 999 的数字。

当特定节点中的特定记录的“类”或“类型”值未知时，“值”和“极限”列值可以是 Null 之外的任何值。

当特定节点的特定记录的“类”值为 Value、Dev 或 ROC 时，“值”和“极限”列值只能接受字符 1234567890.-+eE。

当特定节点的特定记录的“类型”值为 LOLO、LO、HI、HIHI、MinDev、MajDev 或 ROC 时，“值”和“极限”列值只能接受字符 1234567890.-+eE。

当特定节点中的特定记录的任一“类”或“类型”值为 DSC 时，“值”和“极限”列值可以是 Null 之外的任何值。

“注释”列值没有限制。

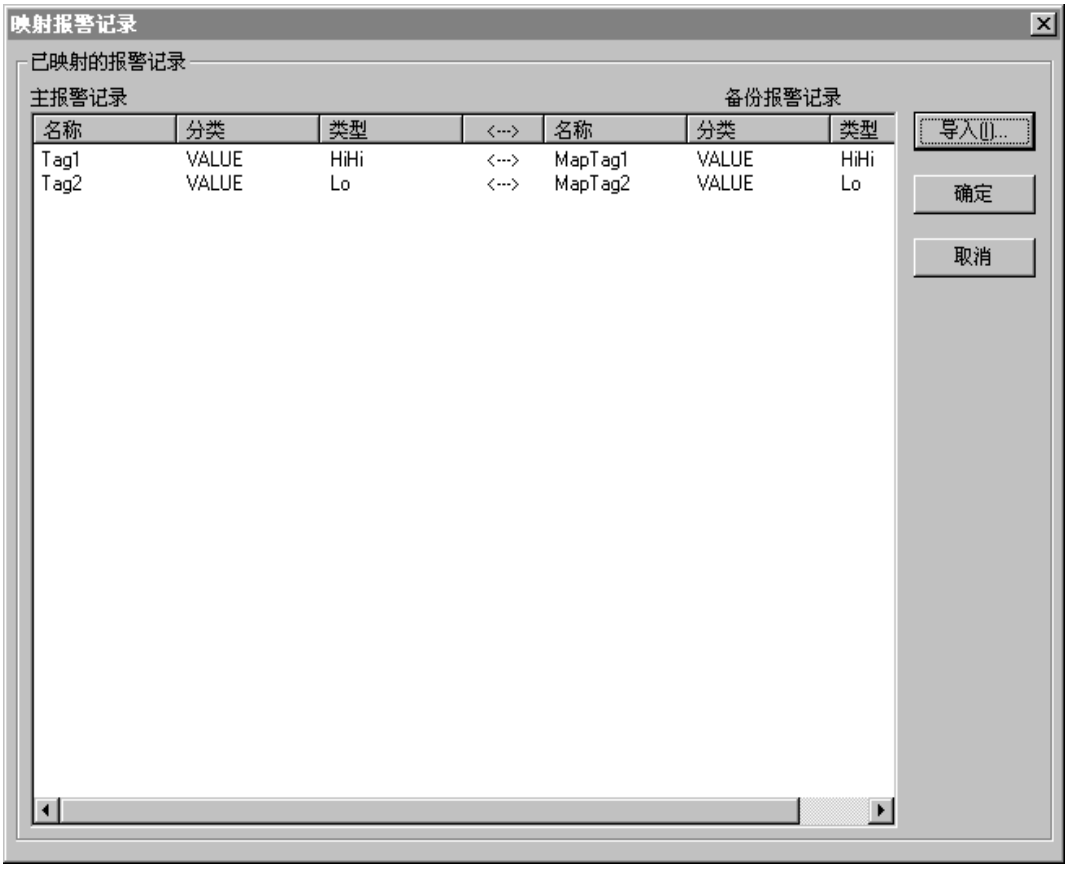
导入文件中不得有重复的记录。如果记录重复，系统会忽略这些记录并在导入过程结束后向用户提供详细信息。

- **要映射报警记录：**

1. 启动 **Alarm Hot Backup Manager** 公用程序。

   您可以通过在 WindowMaker 的应用程序浏览器中双击它来启动。

2. 选择热备份对，然后单击“**映射报警**”。“**映射报警记录**”出现并列出映射信息。



3. 单击“**导入**”，“**打开**”对话框出现。
4. 找到并选择您要导入的 .csv 文件。
5. 单击“**确定**”。

注意：您必须执行“**文件**”菜单上的“**保存**”命令，以保存对 INI 文件所做的所有更改。在退出公用程序之前，系统会提示您保存更改。
当重新导入时，如果某个记录已在此对话框中导入并显示，系统会提示用户该记录将被覆盖。

注意：在“映射报警记录”对话框中导入 CSV 后，并没有完成“报警类”与“报警类型”字段之间的交叉验证。

热备份使用实例

本章节向您介绍一个设置和实施报警热备份对的使用实例。

假定您在三台机器上安装了 InTouch，其中两台（**MachineA** 和 **MachineB**）运行“相同”的 InTouch 报警供应器应用程序。**MachineA** 的应用程序可以产生两个名为 **Tag1** 和 **Tag2** 的摘要报警；**MachineB** 的应用程序可以产生两个逻辑上相同的报警，分别为 **MapTag1** 和 **MapTag2**。

第三台机器是一台经过热备份配置的 InTouch 报警接收器。

➢ **要配置热备份对：**

1. 在客户端节点上创建一个新的 InTouch 应用程序。
2. 启动 WindowMaker。
3. 双击应用程序浏览器中的 **Alarm Hot Backup Manager**，“**Alarm Hot Backup Manager**”对话框出现。



4. 单击“**新建对**”。“**添加新对**”对话框出现：



5. 在“**热备份对名**”框中输入 **Pair1**。
6. 因为主报警供应器 InTouch 应用程序运行在 **MachineA** 上并使用报警组 **$System**，我们对“**主节点**”组字段作如下配置：

   主节点名称 = **MachineA**

   主节点供应器 = **InTouch**

   主节点组 = **$System**。
7. 因为辅助报警供应器 InTouch 应用程序运行在 **MachineB** 上并使用报警组 **$System**，我们对“**备份节点**”组字段作如下配置：

   备份节点名称 = **MachineB**

   备份节点供应器 = **InTouch**

   备份节点组 = **$System**。
8. 配置完成后，单击“**确定**”保存此新备份对。

下一步，我们需要映射关键码域。

- **要映射报警记录关键码域：**

1. 双击应用程序浏览器中的 **Alarm Hot Backup Manager**，“**Alarm Hot Backup Manager**”对话框出现。
2. 在列表中选择热备份对 **Pair1**，然后单击“**设置关键码域**”。“**选定关键码域**”对话框出现：

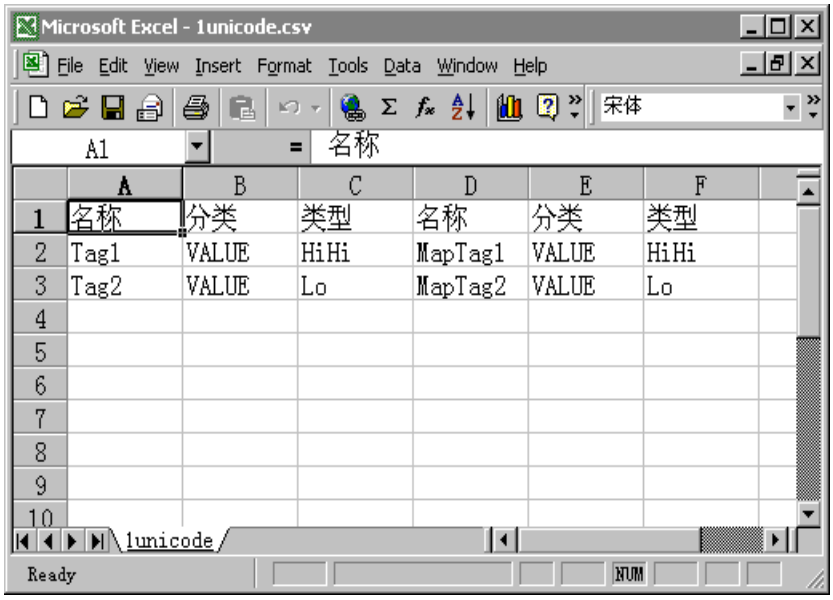


3. 因为我们通过报警名来映射报警并且两个供应器的值不同，所以选择“**报警名**”和“**设计时**”选项。
4. 下一步，单击“确定”来保存配置。“**热备份配置**”消息框出现，提示保存所做的配置。
5. 单击“**是**”。现在我们可以来创建一个包含所有报警记录映射的 .csv 文件。

➢ **要创建报警映射 .CSV 文件：**

在本使用实例中我们在三台机器上安装了 InTouch，其中两台（**MachineA** 和 **MachineB**）运行“相同”的 InTouch 报警供应器应用程序。**MachineA** 的应用程序可以产生两个名为 **Tag1** 和 **Tag2** 的摘要报警；**MachineB** 的应用程序可以产生两个逻辑上相同的报警，分别为 **MapTag1** 和 **MapTag2**。

因此，我们使用 Microsoft Excel 来创建一个包含报警记录映射的名为 **pair1.csv** 的 .csv 文件（映射将从此 .csv 文件导入）。报警记录映射确保当报警在一个报警供应器上获得确认时，分布式报警系统知道其它报警供应器上要确认的报警。



	A	B	C	D	E	F
1	名称	分类	类型	名称	分类	类型
2	Tag1	VALUE	HiHi	MapTag1	VALUE	HiHi
3	Tag2	VALUE	Lo	MapTag2	VALUE	Lo
4						
5						
6						
7						
8						
9						
10						

注意：报警域名必须与“**映射报警记录**”对话框中包含的名称匹配。否则，导入过程会中止并显示一个警告消息框。

有关详细信息，请参阅“映射报警记录”。

现在我们准备导入这个包含所有报警记录映射的 .csv 文件。

- **要导入报警记录映射 .csv 文件：**

1. 启动 WindowMaker。
2. 双击应用程序浏览器中的 **Alarm Hot Backup Manager**，“**Alarm Hot Backup Manager**”对话框出现。
3. 在列表中选择 **Pair1**，单击“**映射报警**”，出现一个空白“映射报警记录”对话框。
4. 下一步，单击“**导入**”，“**打开**”对话框出现。找到并选择 .csv 文件 **(pair1.csv)**，然后单击“**打开**”。分布式报警系统开始导入过程。



5. 一旦导入过程完成，单击“**确定**”。

现在可以准备运行热备份应用程序了。

- **要运行热备份：**

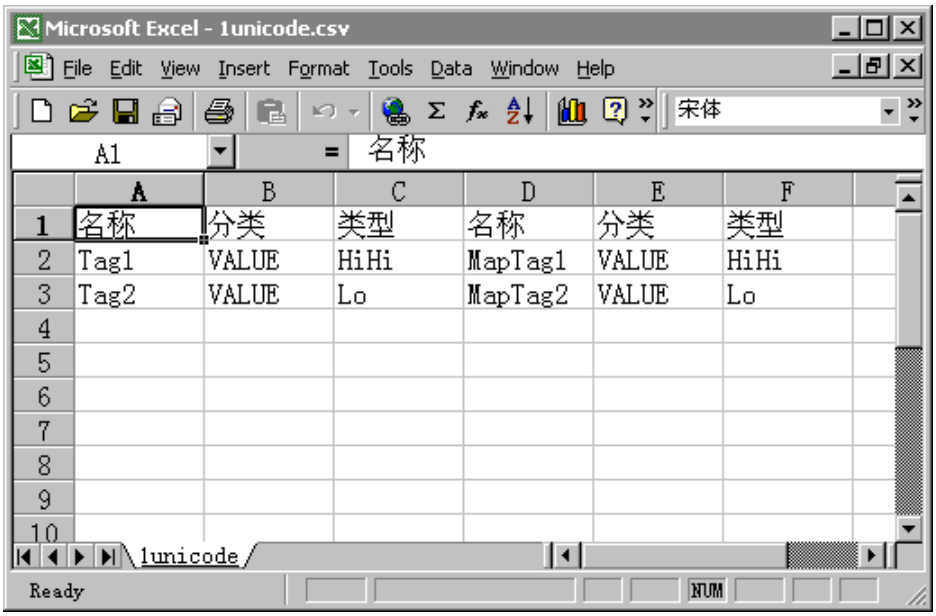
1. 启动两个报警供应器应用程序。
2. 在运行时，产生报警 **Tag1**、**Tag2**、**MapTag1**、**MappTag2**。
3. 在客户端应用程序中，创建一个含 **Pair1** 查询（热备份对名）的分布式报警显示对象。
4. 运行客户端应用程序，显示包含此分布式报警显示对象的窗口。
5. 分布式报警对象现在将显示 Machine A 的报警。
6. 在主报警供应器 **MachineA** 上，确认报警如 **Tag2**。
7. 辅助报警供应器 **MachineB** 上对应的已映射报警 **MapTag2** 将自动获得确认。

同步实例

假定在“**选定关键码域**”对话框中，选定“**报警名**”、“**报警类**”和“**报警类型**”作为**设计时**关键码域，选定“**报警组**”作为**运行时**域。



并已使用 Microsoft Excel 创建了一个对应的报警记录映射 .csv 文件 (**Mapfile.csv**)。



	A	B	C	D	E	F
1	名称	分类	类型	名称	分类	类型
2	Tag1	VALUE	HiHi	MapTag1	VALUE	HiHi
3	Tag2	VALUE	Lo	MapTag2	VALUE	Lo
4						
5						
6						
7						
8						
9						
10						

一旦通过“**映射报警记录**”对话框导入 **Mapfile.csv** 文件，包含 **VALUE** 类和 **HIHI** 类型的名为 **tag1** 的报警将与包含 **VALUE** 类和 **HIHI** 类型的名为 **maptag1** 的报警建立映射。因此，每次在主报警节点上确认 **HiHi** 报警 **Tag1** 时，辅助报警供应器节点和客户端节点上也会确认 **HiHi** 报警 **MapTag1**，条件是主节点上的 **Tag1** 和辅助节点上的 **MapTag1** 的**报警组**名（运行时已映射字段）保持相同。

反之，如果 **MapTag2** 的 **Lo** 报警在辅助报警供应器节点上获得确认，**Tag2** 的 **Lo** 报警会自动在客户端及主报警节点上得到确认 – 假定 **MapTag2** 和 **Tag2** 的报警组名（运行时已映射字段）保持相同。

仅当**设计时**和**运行时**映射匹配时，才会发生确认同步。

注意：您可以选择“**设计时**”和“**运行时**”报警记录域的任意组合来进行映射，但切记要确保映射不会导致多重引用。

例如，如果在“**选定关键码域**”对话框中选定了两个**报警记录域**，比如**类**和**优先级**，则非常可能有一个以上的报警符合标准。在这种情况下，将不能保证热备份同步起作用。在传递确认的过程中，也可能会出现符合标准的随机报警，而其它匹配的报警可能将不能获得确认。

Alarm DB Purge/Archive 公用程序

Alarm DB Purge/Archive 公用程序是随 InTouch 一起安装的一个独立应用程序。Alarm DB Purge/Archive 公用程序负责配置报警数据库的清除/归档操作，并显示任何清除活动的状态。此公用程序还可根据时间来安排清除和归档作业。

您可以使用 Alarm DB Purge/Archive 公用程序来：

指定需要清除和（或）归档的记录类型（合并报警记录或报警详细记录）。

根据配置来指定自动定时清除/归档作业。

在必要时清除整个数据库。

设置执行清除/归档活动的时间。

指定执行清除/归档的时间间隔，可以是每天、每星期或每月。

显示清除/归档的状态。

保存状态信息到日志文件（您在将来可以使用此日志文件来检查清除/归档活动的状态）。

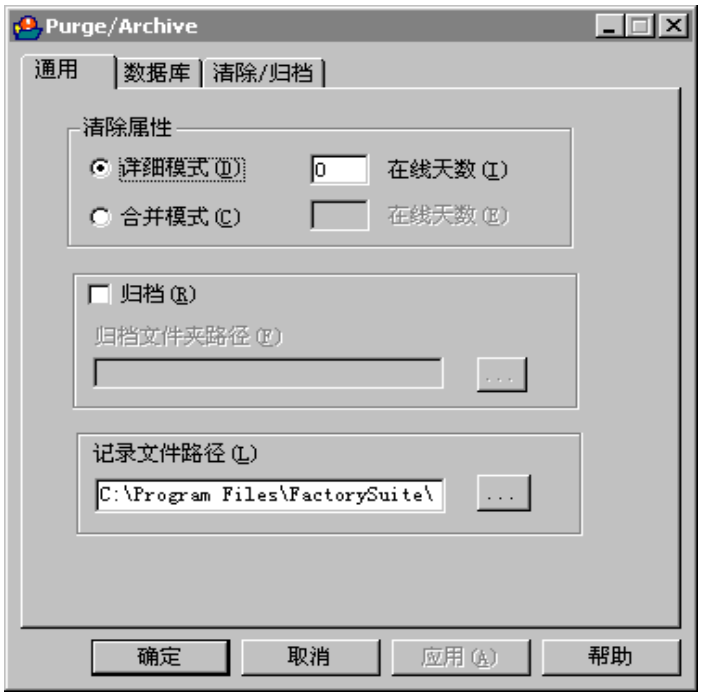
清除/归档通用属性配置

Alarm DB Purge/Archive 公用程序的“**通用**”属性页用于设置清除/归档属性。您可以选择需要清除的表类型 – “**详细报警**”或“**合并报警**”表；选择归档您要清除的数据；选择归档数据的文件夹路径。为清除/归档的每个表建立归档文件；通过表名、日期和时间的唯一组合来自动创建归档文件名。例如，在 2000 年 11 月 10 日下午 5:30 归档的 **AlarmMaster** 的归档文件名为 **AlarmMaster_11102000_1730.txt**。

有关详细信息，请参阅“分布式报警数据库视图”。

➢ **要清除/归档报警数据库：**

1. 启动 Alarm DB Purge/Archive 公用程序。“**Purge/Archive**”（清除/归档）对话框出现。



当最小化时，Alarm DB Purge/Archive 公用程序显示为系统托盘中的一个图标。当右击图标时，会出现一个弹出式菜单，显示下列命令：

现在清除全部 – 清除整个数据库。
现在清除 – 用现有设置立即执行清除。
取消清除 – 停止当前的清除操作（此命令仅当清除正在进行时才会被激活）。当停止清除时，数据库将恢复为其原始状态。
激活 – 激活定时清除。
去激活 – 取消激活清除定时器。
现在测试 – 执行清除测试来检查与数据库及目标位置的连接。
清除状态 – 清除状态窗口。
隐藏窗口 – 将 Alarm DB Purge/Archive 公用程序最小化为系统托盘中的一个图标。
显示窗口 – 打开并最大化 Alarm DB Purge/Archive 公用程序。
退出 – 关闭 Alarm DB Purge/Archive 公用程序。

右击 Alarm DB Purge/Archive 公用程序属性页的任意位置，会出现相同的弹出式菜单。

2. 在“**清除属性**”组中，选择您要用于清除的选项：

详细模式 – 清除在详细模式下记录的报警。

合并模式 – 清除在合并模式下记录的报警。

3. 当您选择“**清除属性**”模式后，相应的“**在线天数**”输入框将被激活。输入您要保留数据的天数。所有指定天数之前的数据都会被清除（有效的输入值是 0 – 9999），如果天数为 0，则数据不会被清除。

4. 选择“**归档**”选项，然后通过在“**归档文件夹路径**”输入框中输入完整路径，或通过单击相应的“浏览 (**...**) ”按钮寻找并选择文件，来指定您要归档到的数据库文件的完整路径（最长 225 个字母数字字符）。一旦选定，该路径将自动显示在输入框中。

5. 在“**记录文件路径**”输入框中，指定用于存储状态文件的文件夹的完整路径（最长 225 个字母数字字符），或单击相应的“浏览 (**...**) ”按钮来寻找并选择文件（一旦选定，该路径将自动显示在输入框中）。

注意：缺省条件下，记录文件 (wwalmpurge.log) 存储在 FactorySuite\Common 文件夹之下，但用户可以改写。每次执行清除/归档活动时，记录将被追加到此文件中。

---

6. 单击“**应用**”将设置保存到注册表中，或单击“**确定**”保存设置并关闭 Alarm DB Purge/Archive 公用程序。

清除/归档数据库配置

Alarm DB Purge/Archive 的“数据库”属性页允许您选择清除/归档活动将访问的服务器。

➢ **要配置 Alarm DB Purge/Archive 公用程序数据库：**

1. 启动 Alarm DB Purge/Archive 公用程序，“**Purge/Archive**”（清除/归档）对话框出现。
2. 单击“**数据库**”标签来激活“**数据库**”属性。



如果右击属性页，会出现一个弹出式菜单，显示下列命令：

现在清除全部 – 清除整个数据库。
现在清除 – 用现有设置立即执行清除。
取消清除 – 停止当前的清除操作（此命令仅当清除正在进行时才会被激活）。当停止清除时，数据库将恢复为其原始状态。
激活 – 激活定时清除。
去激活 – 取消激活清除定时器。
现在测试 – 执行清除测试来检查与数据库及目标位置的连接。
清除状态 – 清除状态窗口。
隐藏窗口 – 将 Alarm DB Purge/Archive 公用程序最小化为系统托盘中的一个图标。
显示窗口 – 打开并最大化 Alarm DB Purge/Archive 公用程序。
退出 – 关闭 Alarm DB Purge/Archive 公用程序。

3. 在“**SQL Server/MSDE**”组中，单击“**服务器名**”箭头打开可用 SQL/MSDE 服务器的列表，然后选择您要使用的数据库服务器名。
4. “**数据库**”框显示数据库名（这是一个只读字段，缺省为 **WWALMDB**）。
5. 在“**用户信息**”组的相应输入框中，输入您的“**用户名**”和“**口令**”（基于安全考虑，所有**口令**字符均显示为 **）。
6. 单击“**测试连接**”来测试与数据库的连接。
7. 单击“**应用**”将设置保存到注册表中，或单击“**确定**”保存设置并关闭 Alarm DB Purge/Archive 公用程序。

清除/归档配置

Alarm DB Purge/Archive 公用程序的“**清除/归档**”属性页允许您设置触发自动清除和归档活动的时间，它还允许您执行清除测试来检查与数据库和目标位置的连接，并启动和停止清除。

➢ **要配置 Alarm DB Purge/Archive 公用程序时间间隔：**

1. 启动 Alarm DB Purge/Archive 公用程序，“**Purge/Archive**”（清除/归档）对话框出现。
2. 单击“**清除/归档**”标签来激活“**清除/归档**”属性页。



如果右击属性表，会出现一个弹出式菜单，显示下列命令：

- **现在清除全部** – 清除整个数据库。
  现在清除 – 用现有设置立即执行清除。
  取消清除 – 停止当前的清除操作（此命令仅当清除正在进行时才会被激活）。当停止清除时，数据库将恢复为其原始状态。
  激活 – 激活定时清除。
  去激活 – 取消激活清除定时器。
  现在测试 – 执行清除测试来检查与数据库及目标位置的连接。
  清除状态 – 清除状态窗口。
  隐藏窗口 – 将 Alarm DB Purge/Archive 公用程序最小化为系统托盘中的一个图标。
  显示窗口 – 打开并最大化 Alarm DB Purge/Archive 公用程序。
  退出 – 关闭 Alarm DB Purge/Archive 公用程序。

3. 选择您要使用的“**时间间隔**”选项。

   每天 - 每天在“**时间**”字段中选定的时间开始清除。

   每星期 – 每星期在“**日**”字段中选定的日期及“**时间**”字段中选定的时间开始清除。

   每月 – 每月在“**日**”字段中选定的日期及“**时间**”字段中选定的时间开始清除。

注意：“**日**”缺省为空白。如果您选择“**每星期**”或“**每月**”，则必须选择日期。

---

4. 在选定“**时间间隔**”选项后，单击“**时间**”向上/向下箭头来选择开始清除/归档活动的每天时间。
5. 在选定“**时间间隔**”选项后（仅限于“**每星期**”或“**每月**”），单击“**日**”箭头来选择执行清除/归档活动的日期（当“**时间间隔**”为“**每星期**”时指星期几，当“**时间间隔**”为“**每月**”时指日期）。
6. 清除活动的状态显示在“**状态**”区域中。
7. 单击“**现在清除**”，用现有设置立即执行清除。

注意：单击“**现在清除**”会改写激活时间，并立即执行清除和归档操作。“**现在清除**”会检查是否存在归档文件并追加到相同文件中。如果没有归档文件，则会按照命名惯例创建一个文件，然后将此文件用于归档。“**现在清除**”不会删除表中的项目，如 **ProviderSession**、**Query** 和 **Cause** 等，这些项目通过外部关键字约束条件与 **AlarmMaster** 等主表链接。这些表中的相关记录会写入到文件中，以保持数据的一致性，同时也会保存到数据库中。

---

8. 单击“**现在测试**”，执行清除测试来检查与数据库及目标位置的连接（仅当选定“清除/归档”标签中的“**归档**”选项后，此按钮才会被激活。它将在指定的归档路径中创建空白归档文件）。
9. 单击“**取消清除**”来停止清除（此按钮仅当清除正在进行时才会被激活）。当停止清除时，数据库将恢复为其原始状态。
10. 单击“**现在清除全部**”来清除整个数据库。

警告：仅当 Alarm DB Logger 不在运行时，才选择“**现在清除全部**”。当 Alarm DB Logger 正在记入数据库时，如果成功确认“**现在清除全部**”，Alarm DB Logger 会停止将数据记入数据库，并开始缓存记录。

---

11. 单击“**应用**”将设置保存到注册表中，或单击“**确定**”保存设置并关闭 Alarm DB Purge/Archive 公用程序。

Alarm DB Restore 公用程序

Alarm DB Restore 公用程序用于将备份文件中的归档报警记录恢复到数据库中。

Alarm DB Restore 公用程序可协助您将归档数据恢复到数据库中，以便执行进一步的操作。

恢复数据库配置

“**配置**”属性页用于指定数据需要恢复到的数据库。如果指定的数据库不在服务器上，系统会提示您在服务器上创建一个新数据库。一旦确认，程序会用缺省参数在指定服务器上创建新数据库。

- **要配置需恢复的数据库：**

1. 启动 **Alarm DB Restore** 公用程序。“**Restore**”（恢复）对话框出现，并显示“**配置**”标签。



- 当最小化时，Alarm DB Restore 公用程序显示为系统托盘中的一个图标。当右击图标时，会出现一个弹出式菜单，显示下列命令：

  恢复 – 开始恢复过程。
  取消恢复 – 结束恢复过程。
  清除状态 – 清除状态窗口。
  隐藏窗口 – 将 Alarm DB Restore 公用程序最小化为系统托盘中的一个图标。
  显示窗口 – 打开并最大化 Alarm DB Restore 公用程序。
  退出 – 关闭 Alarm DB Restore 公用程序。

  右击 Alarm DB Restore 公用程序属性页的任意位置，会出现相同的弹出式菜单。

2. 在“**SQL Server/MSDE**”组中，单击“**服务器名**”箭头打开可用 SQL/MSDE 服务器的列表，然后选择您要使用的数据库服务器名。
3. “**数据库名**”框显示数据库名（这是一个只读字段，缺省为 **WWALMDB**）。
4. 在“**用户信息**”组的相应输入框中，输入您的“**用户名**”和“**口令**”（基于安全考虑，所有**口令**字符均显示为 **）。
5. 单击“**测试连接**”来测试与数据库的连接。

恢复文件名配置

Alarm DB Restore 公用程序的“**选择**”属性页用于指定需要从中执行恢复的文件名，它还显示一个状态列表框，表明将数据恢复到数据库的进度状态。您可以从此对话框取消正在进行中的恢复操作。如果取消恢复，数据库将恢复为其原始状态。

➢ **要选择需恢复的数据库：**

1. 启动 Alarm DB Restore 公用程序。“**Restore**”（恢复）对话框出现，并显示“**配置**”标签。
2. 单击“**选择**”标签激活“**选择**”属性页。



如果右击属性表，会出现一个弹出式菜单，显示下列命令：

恢复 – 开始恢复过程。
取消恢复 – 结束恢复过程。
清除状态 – 清除状态窗口。
隐藏窗口 – 将 Alarm DB Restore 公用程序最小化为系统托盘中的一个图标。
显示窗口 – 打开并最大化 Alarm DB Restore 公用程序。
退出 – 关闭 Alarm DB Restore 公用程序。

3. 在“**归档文件目录**”输入框中，输入归档文件位置的完整路径（最长为 255 个字母数字字符），或单击相应的“浏览 (...)”按钮来寻找并选择创建及存储归档文件的文件夹（一旦选定，该路径将自动显示在输入框中）。

4. 在“**记录文件目录**”输入框中，指定用于创建和归档记录文件的完整路径（最长 225 个字母数字字符），或单击相应的“浏览 (**...**)”按钮来寻找并选择文件夹（一旦选定，记录文件的完整路径将自动显示在输入框中）。

5. 如果您要在目标数据库中重新创建表，请选择“**重建表**”。如果您不选择此选项，数据会被追加到现有的表中。

警告：重建表会导致现有数据的丢失。

---

6. 当您选择“**重建表**”选项时，“**记录模式**”组会被激活。选择您要使用的模式：

   详细的 – 仅恢复详细表。

   合并的 – 仅恢复合并表。

7. 在“**在此之后恢复（日期/时间）**”字段中，选择自此执行恢复的日期和时间。

注意：日期指定自此执行恢复的日期；时间指定在所指日期开始恢复的时间；两个字段缺省为当前日期和时间。

AlarmSuite History Migration 实用程序

AlarmSuite History Migration 实用程序随 InTouch 一起安装。AlarmSuite History Migration 实用程序负责将 AlarmSuite 数据从 AlarmSuite 数据库移植到新的 InTouch 分布式报警系统的数据库模式中。

有关详细信息，请参阅“分布式报警数据库视图”。

➢ **要移植 AlarmSuite 历史数据：**

1. 启动 AlarmSuite History Migration 实用程序。**“AlarmSuite History Migration 实用程序”**对话框出现：



2. 在**“(AlarmSuite) 源数据库”**组中，单击 **DSN** 箭头打开在本地机器上配置的 ODBC DSN 名称列表。
3. 选择您所要用的 ODBC DSN 名称。
4. 在**“用户名”**输入框中，输入您的 DSN 用户名。
5. 在**“口令”**输入框中，输入您的 DSN 用户口令（基于安全考虑，所有输入字符均显示为 **）。
6. 选择您要使用的**“清除源数据库”**选项。

   是 – 在数据移植结束时清除源数据库（创建表是事务处理的一部分，但创建数据库却不是。如果您选择**“是”**，会在从源数据库移植数据后提交清除事务处理。

   否 – 在数据移植结束时不清除源数据库

7. 在“**目标数据库**”组中，单击“**服务器名**”箭头（在 “**SQL Server / MSDE**” 组中），以打开可用 SQL/MSDE 服务器的列表。选择您要使用的数据库服务器名。
8. “**数据库名**”框显示数据库名（这是一个只读字段，缺省为 **WWALMDB**）。
9. 在“**用户名**”输入框中，输入您的 SQL 服务器用户名。
10. 在“**口令**”输入框中，输入您的 SQL 服务器口令（基于安全考虑，所有**口令**字符均显示为 **）。
11. 如果您要在目标数据库中重新创建表，请选择“**重建表**”。如果您不选择此选项，数据会被追加到现有的表中。

注意：重建表会导致现有数据的丢失。

---

12. 当您选择“**重建表**”选项时，“**记录模式**”组会被激活。选择您要使用的
模式：

    详细的 – 仅恢复详细表。

    合并的 – 仅恢复合并表。
13. 单击“**测试连接**”来测试与源及目标数据库的连接。
14. 单击“**启动**”开始将数据从源数据库移植到目标数据库。
15. 单击“**停止**”来中止进行中的移植过程。这将回滚事务处理（单击“**关闭**”相当于单击“**停止**”）。

注意：创建表是事务处理的一部分，但创建数据库却不是。如果您将“**清除源数据库**”选项设置为“**是**”，会在从源数据库移植数据后提交清除事务处理。

---

16. 一旦移植过程完成，单击“**关闭**”退出程序。

注意：当数据移植正在进行时，不得使用源或目标数据库来记录新数据。这可能会引起数据完整性的局部丢失，从而导致错误的搜索结果。

分布式报警数据库视图

本章节介绍 InTouch 7.11 所提供的数据库视图。这些视图提供易于查询的虚拟表，允许分析过去和当前出现的报警及事件。使用这些视图可通过一种 Wonderware 长期（包括后续产品版本）支持的方式来分析报警和事件历史，而不论这些版本中的实际数据库模式的变化和差异。最后，提供视图可与 AlarmSuite 产品保持向后兼容性。在安装产品并创建报警记录数据库后，就可以使用所有的视图了。

视图与存储过程简介

InTouch 分布式报警系统提供了一套工具来查看和分析由 Alarm DB Logger 记入关系数据库的历史报警和事件信息。这些功能以数据库视图和存储过程的形式提供。

数据库视图可以看成是一个简单的逻辑表，通过一种简单易用的格式，结合来自多个基本数据库表的信息。视图包含在数据库中，数据库视图通常称为虚拟数据库表，因为它们可以象实际的数据库表一样，使用标准 SQL 来进行查询。当查询视图时，会返回一组记录（或行）。每行具有若干列包含记录数据的信息。

分布式报警系统的数据库视图提供了一种查看自动化应用程序中出现的历史报警和事件的方式。通过对视图执行 SQL 命令，使用复杂的过滤标准，可在这些视图上执行复杂的分析。例如，您可以检索工厂特定区域特定时间段内出现的所有 HiHi 报警的报警记录。或者，您可以检索由某个报警供应器节点记录的所有数据更改事件。

存储过程是 SQL 语句的集合，这些语句共同执行特定的函数并从数据库返回数据。因此，它们可以看成是子程序或函数。存储过程可接受那些控制其如何操作的输入参数，然后以结果记录集的形式返回数据。存储过程的用户无需使用 SQL 语法，只要提供存储过程名和输入参数就可以执行存储过程。返回的结果是显示为 SQL 记录集的一组记录，与视图非常相似。由于存储过程通过内嵌的 SQL 语句存在于数据库中，减少了返回客户端的往返行程，因而还能提高系统性能。

InTouch 分布式报警系统提供的存储过程允许用户从数据库中提取信息，而不需要知道 SQL 的细节。存储过程内部承担大量的工作，如制定适当的 SQL 查询、连接和执行复杂的逻辑等。

使用存储过程和视图都可以为用户带来极大的好处，即用户不需要知道数据库的内部细节，就可以非常轻松地检索报警和事件分析及报告所需的信息。

下面的章节将定义随供的视图和存储过程，并通过简单易学的实例来揭示其灵活性和强大功能。

视图与存储过程列的定义

存储过程和视图会返回信息表，每个表包含许多列。本章节将提供由视图和存储过程（稍后介绍）返回的所有列的定义。

列名	描述
AlarmCount	整型值，指明指定时间段内的报警次数。如果报警在进入选定时间范围之前已经存在，则不会计算在内。
AlarmState	统一码字符串，指明报警的状态如 UNACK（未确认-报警中）、UNACK_RTN（未确认-恢复正常）、ACK（已确认-报警中）、ACK_RTN（已确认-恢复正常）。ACK 状态仍具有报警中的项目。
AlarmType	见 Type。
Area	统一码字符串，指明产生报警或事件的标记名或对象所属的报警组或区域名称。
Category	统一码字符串，指明报警或事件的类或类别，如 Value、Dev、ROC 等。
CheckValue	统一码字符串，指明报警的报警限值或事件的上次取值。
Comment	统一码字符串，指明报警注释。
Description	统一码字符串，指明报警或产生报警的对象的描述字符串。如果是确认报警，则为确认注释。
EngUnits	工程单位值（将来），在 7.11 中返回为 “”。
EventCount	指定时间段内发生的事件次数。
EventStamp	日期和时间值，指明报警或事件的日期和时间，采用当地时间格式。
EventStampUTC (GMT)	日期和时间值，指明报警或事件的日期和时间，采用世界调整时间格式。
GroupName	见 Area。
LastEvent (Future)	日期和时间值，指明此报警最后一次转换的日期和时间，采用当地时间格式。
Limit	统一码字符串，指明报警或事件出现时报警变量的报警限值。视图 v_AlarmSuiteAlarmLog 返回此值为浮点类型。
NodeName	统一码字符串，指明提供报警或事件给分布式报警系统的节点名。
Operator	统一码字符串，指明与报警或事件关联的操作员姓名。

Priority	整型值，指明报警优先值 (1-999)。
Provider	统一码字符串，指明提供报警或事件给分布式报警系统的应用程序或组件。
TagName	统一码字符串，指明产生报警或事件的对象名。
TimeInState（将来）	整型值，指明报警进入当前报警状态的时间（以秒计）。
Type	统一码字符串，指明类别内的报警类型，如 HiHi、LoLo 等。
Units	工程单位字符串（将来）。在 7.11 中返回为 “”。
Value	统一码字符串，指明报警或事件出现时报警变量的当前值。视图 v_AlarmSuiteAlarmLog 返回此值为浮点类型。
ValueString	统一码字符串，指明报警或事件出现时报警变量的当前值。

注意：将来：保留供将来使用。

- **要在 Enterprise Manager 中查看视图定义：**

1. 展开服务器组，然后展开服务器。
2. 展开“**数据库**”。
3. 展开视图所属的数据库。
4. 单击“**视图**”。
5. 在详细资料窗格中，右击存储过程。
6. 单击“**属性**”。

- **要调用视图：**

与存储过程不同，视图不带参数。要在视图中查看数据，执行以下操作。

1. 在 Enterprise Manager 中，展开服务器组。
2. 展开服务器。
3. 展开“**数据库**”。
4. 展开视图所属的数据库。
5. 单击“**视图**”。
6. 在详细资料窗格中，右击“视图”，然后选择“**打开行**”。
7. 单击“**全部行**”。

报警历史数据库视图

此视图提供了选定时间范围内出现的所有报警和报警转换事件的历史列表。查询指定开始和结束日期及时间（由列 EventStamp 或 EventStampUTC 指定）。_返回的记录包括报警来源、报警确认、报警启用、报警禁用，以及报警恢复正常事件。注意视图中所有的字符串均为统一码。

v_AlarmHistory

列名	数据类型	描述
EventStamp	Datetime	报警事件的日期和时间（数据库的本地时间）
AlarmState	nChar	报警状态：UNACK、UNACK_RTN、ACK、ACK_RTN、DISABLED（将来）、ENABLED（将来）之一
TagName	nChar	产生报警的对象名，如 TIC101
Description	nVarchar	报警的描述字符串，可以缺省为对象说明（或 InTouch 中的注释），或确认记录的确认注释。
Area	nChar	报警的区域或组名
Type	nChar	报警类型，如 Hi、HiHi、ROC、PV.HiAlarm
Value	nChar	报警时的报警变量值
CheckValue	nChar	报警时的报警限值
Priority	Integer	报警优先级
Category	nChar	报警类或报警类别，如 Value、Dev、ROC、Process、Batch、System 等。
Provider	nChar	报警供应器：节点/InTouch 或 GalaxyName。
Operator	nChar	操作员姓名，如 JoeR（如果有）。
Comment	nVarchar	将来版本增强
EngUnits	nChar	将来版本增强
EventStampUTC	DateTime	报警事件的 UTC 日期/时间

'SQL Query Analyzer' 中的 SQL 语句也可以写入视图中的视图数据，例如，在“查询”窗口中输入：

```
Select top 100 * from v_AlarmHistory where Priority>10
AND((Provider LIKE '%adelphi%')AND (Area LIKE '%$s%'))
```

-- 从 v_AlarmHistory 视图中选择所有记录

```
Select * from v_AlarmHistory
```

-- 从 v_AlarmHistory 视图中选择优先级高于 100 的所有记录

```
Select * from v_AlarmHistory WHERE Priority >=100
```

辉亮显示（选择）以上文本，然后按 **F5**。结果数据将显示在窗口的底部区域。

有关 SQL 语句的详细信息，请参阅 SQL Server 手册。

事件历史数据库视图

此数据库视图提供了选定时间范围内出现的所有非报警事件的历史列表。查询客户端指定开始和结束日期及时间。返回的记录包括所有非报警事件。

v_EventHistory

列名	数据类型	描述
EventStamp	Datetime	事件的日期和时间。
TagName	NChar	产生事件的对象名，如 Pump1。
Description	nVarChar	事件的描述字符串，可以缺省为对象说明或 InTouch 中的注释。
Area	NChar	事件的区域或组名。
Type	NChar	事件类型，如“操作员数据改变”、“启动”等。
Value	NChar	新值（如果有）。
CheckValue	NChar	旧值（如果有）。
Category	NChar	事件类别或类，如 Value、Process、Batch、System 等。
Provider	NChar	事件发生器，如节点 /InTouch 或用户更改的视图引擎名。
Operator	NChar	操作员姓名，如 JoeR（如果有）。
Operator2	NChar	将来版本增强。
Units	NChar	将来版本增强。
MsgText	NChar	将来版本增强。
EventStampUTC	DateTime	事件的 UTC 日期/时间。

报警与事件历史数据库视图

此数据库视图提供了选定时间范围内出现的所有事件和报警的历史列表。查询客户端指定开始和结束日期及时间。返回的记录包括所有报警和事件。此视图将报警视图和事件视图合而为一，即将这两个视图中的记录合并在一起。

v_AlarmEventHistory

列名	数据类型	描述
EventStamp	Datetime	事件的日期和时间。
AlarmState	nChar	报警状态：UNACK、UNACK_RTN、ACK、ACK_RTN、DISABLED（将来）、ENABLED（将来）之一，不适用于事件。
TagName	nChar	产生报警的对象名，如 TIC101。
Description	nVarchar	报警/事件的描述字符串，可以缺省为对象说明（或 InTouch 中的注释），或确认记录的确认注释。
Area	nChar	报警的区域或组名。
Type	nChar	报警或事件的类型，如 Hi、HiHi、ROC、PV.HiAlarm、操作员数据更改等。
Value	nChar	报警时的报警变量值。
CheckValue	nChar	报警时的报警限值，或事件的旧值。
Priority	Integer	报警优先级。
Category	nChar	报警或事件类，或事件类别，如 Value、Process、Batch、System 等。
Provider	nChar	报警供应器，如节点/InTouch 或 GalaxyName。
Operator	nChar	确认操作员或数据更改操作员的姓名。
Comment	Varchar	将来版本增强。
Units	nChar	将来版本增强。
Operator2	nChar	将来版本增强。
EventStampUTC	DateTime	事件的 UTC 日期/时间。

'SQL Query Analyzer' 中的 SQL 语句也可以写入视图中的视图数据，例如：

要：

从 v_AlarmEventHistory 视图中选择标记名为 "MyTag1"、报警状态为 ACK_RTN 或 ACK 的所有记录的 TagName、Area 和 Type 列，并按供应器排序，

在“查询”窗口中输入下面的语句：

```
Select TagName,Area,Type FROM v_AlarmEventHistory WHERE
TagName='MyTag1' AND (AlarmState='ACK_RTN' OR
AlarmState='ACK') ORDER BY Provider
```

辉亮显示（选择）以上文本，然后按 **F5**。结果数据将显示在窗口的底部区域。有关 SQL 语句的详细信息，请参阅 SQL Server 手册。

报警计数器数据库存储过程

此数据库存储过程提供指定时间间隔内出现的每种报警的报警次数。查询客户端必须指定开始和结束日期及时间。它包含五个可选参数：TagName、Class、Type、Provider 和 Comment。计数器仅用于统计源报警数（不用于如确认或恢复正常等转换）。因此，如果发生报警并且随后获得确认并返回正常状态，则该报警的计数仅为 1（而不是 3）。此视图用于显示报警频率。每一次报警由 TagName、Provider、Type 和 Category 唯一确定。此视图可以回答这样的问题：供应器 Node1|InTouch（供应器）上的对象 TIC101（标记名）在这个时间段内共有多少次发出 HiHi（类型）的值报警（类别）？

sp_AlarmCounter

注意：报警计数器仅适用于详细模式的记录，它不支持合并模式记录。

列名	数据类型	描述
TagName	Nchar	产生报警的对象名，如 TIC101。
Area	Nchar	报警的区域或组名。
Type	Nchar	报警类型，如 Hi、HiHi、ROC、PV.HiAlarm。
Category	Nchar	报警类或报警类别，如 Value、Process、Batch 等。
AlarmCount	Integer	时间范围内的报警次数。如果报警在进入选定时间范围之前已经发生，则不会计算在内。
Priority	Integer	报警优先级。
Provider	Nchar	报警供应器，如节点/InTouch 或 GalaxyName。

在 Enterprise Manager 中查看存储过程定义

➢ **要在 Enterprise Manager 中查看存储过程定义：**

1. 展开服务器组，然后展开服务器。
2. 展开“**数据库**”。
3. 展开存储过程所属的数据库。
4. 单击“**存储过程**”。
5. 在详细资料窗格中，右击存储过程，然后单击“**属性**”。

调用存储过程

存储过程可以使用 Transact-SQL 语句 EXECUTE 从 SQL 服务器调用。

下面是调用存储的一个实例：

在 'SQL Query Analyzer' 的“查询”窗口中，输入：

```
EXECUTE sp_AlarmCounter @StartDate='2001-01-01',
@EndDate='2001-03-31', @Tagname = ‘tag1’, @Type = ‘LO’,
@Provider = ‘WW21353\InTouch’, @Comment = ‘SSAADD’
```

辉亮显示（选择）以上文本，然后按 **F5**。结果数据将显示在窗口的底部区域。

StartDate 和 EndDate 是 sp_AlarmCounter 存储过程的两个强制性参数，其余五个参数是可选参数。这意味着您可以忽略除 StartDate 和 EndDate 之外的任何其它五个参数。当某个参数被忽略时，存储过程将不会使用它来过滤，而显示包含所忽略参数的所有匹配值的一般结果。

StartDate 和 EndDate 是 SQL Server 中 ‘datetime’ 类型的变量，可以接受*《SQL Server 用户指南》*中指定的各种格式的数据，您需要使用这两个参数来传递执行存储过程所需的持续时间。

事件计数器数据库存储过程

此数据库存储过程提供某个时间间隔内发生的特定标记名特定类型的事件次数。查询客户端必须指定开始和结束日期及时间。它包含三个可选参数：TagName、Provider 和 Comment。计数器仅适用于非报警事件。此视图用于显示每种事件的发生频率。例如，泵打开多少次？TagnName、Provider、Category 和 Type 用于唯一地确定事件并执行计数。

sp_EventCounter

列名	数据类型	描述
TagName	Nchar	产生报警的对象名，如 TIC101。
Area	NChar	报警的区域或组名。
Type	NChar	事件类型。
Category	NChar	报警类或报警类别，如 Value、Process、Batch 等。
EventCount	Integer	在指定的时间范围内标记名此类型的事件发生的次数。
Provider	NChar	事件供应器：_节点/InTouch 或 GalaxyName。

同样，sp_EventCounter 存储过程也采用两个强制性参数 StartDate 和 EndDate，以及三个可选参数 Tagname、Provider 和 Comment。

```
EXECUTE sp_AlarmCounter @StartDate='2001-01-01 23:23:23',
@EndDate='2001-03-31 23:23:23', @Tagname = '$NewAlarm'
```

AlarmSuite 报警记录数据库视图

此数据库视图提供有关历史报警和事件的视图：其返回的表与名为 AlarmLog 的实际 Alarm Suite 表具有相同的列。支持 AlarmSuite AlarmLog 表的查询也支持此视图，但这些查询中表名必须改为视图名。

v_AlarmSuiteAlarmLog

列名	数据类型	描述
EventStamp	Datetime	事件的日期和时间。
EventType	NChar	由 AlarmSuite 定义的事件类型。
AlarmType	NChar	报警类型。
AlarmState	NChar	报警的确认状态。
NodeName	NChar	报警的节点。
TagName	NChar	报告报警的对象名。
GroupName	NChar	报警组或区域名。
Comment	NChar	确认注释（如果有）。
Value	Float	报警变量值。
Limit	Float	报警时的报警限值。
ValueString	NChar	见 Alarm Suite。
Operator	NChar	操作员姓名。
Priority	Integer	优先级。
Units	NChar	在 7.11 中返回为 “”。

'SQL Query Analyzer' 中的 SQL 语句也可以写入视图中的视图数据，例如，在“查询”窗口中输入：

```
-- select all distinct tagnames from the v_AlarmSuiteAlarmLog
view

Select Distinct TagName From v_AlarmSuiteAlarmLog
```

辉亮显示（选择）以上文本，然后按 **F5**。结果数据将显示在窗口的底部区域。

有关 SQL 语句的详细信息，请参阅 SQL 服务器手册。

第 8 章

实时趋势和历史趋势

InTouch 为您提供了两种趋势显示对象：“实时趋势”和“历史趋势”。您可以对两种趋势对象进行配置，以显示特定时间内多个标记名的图形表象。实时趋势允许您最多用四支笔（数据值）画图，而历史趋势则允许您最多用八支笔画图。两种趋势都使用 WindowMaker 的特殊工具来建立。InTouch 还让您可以完全控制趋势的配置。例如，您可以指定时间间隔、值范围、网格分辨率、时间标签位置、笔数目和颜色属性。

FactorySuite Productivity Pack 包括一个允许您用 16 支笔画图的笔配置网格。有关详细信息，请参阅您的 *《Productivity Pack 用户指南》*。

InTouch 还支持分布式历史系统，允许您从任何 InTouch 历史记录文件中检索历史数据，甚至跨网络进行检索。

除趋势能力之外，InTouch 还包括 HistData 实用程序，这是一个设计用于处理 InTouch 历史记录文件的程序。HistData 实用程序可以将加密历史记录文件 (.LGH) 转换成逗号分隔变量 (.CSV) 文件，以用于 Microsoft Excel 等电子数据表或文本编辑环境中。

目录

- 实时趋势
- 历史趋势
- 在运行时配置历史趋势
- 历史趋势点域
- 历史 QuickScript 函数
- 分布式历史系统
- 创建历史趋势指示器
- 历史趋势和夏令时
- HistData 实用程序
- 在 Excel 中使用 HistData

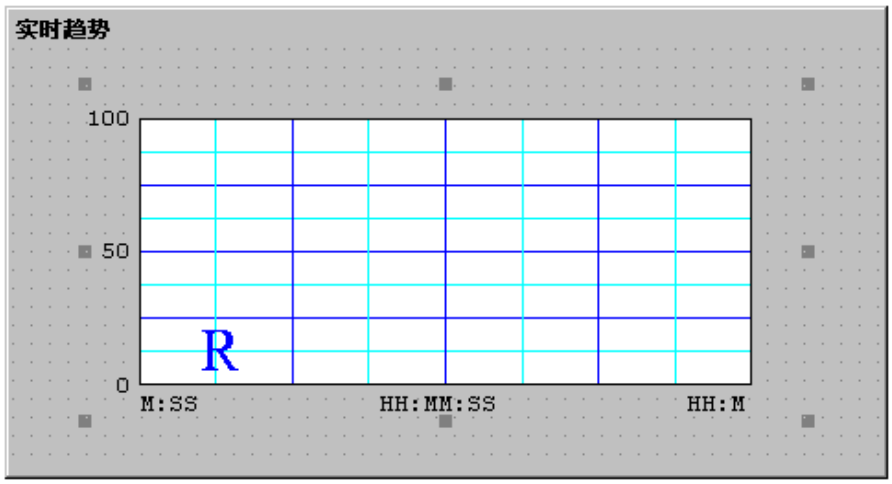
实时趋势

实时趋势是动态的，它们在运行期间不断更新。它们最多可以绘制四个本地标记名或表达式的变化。

创建实时趋势

➢  **要创建实时趋势：**

1. 选择“**绘图对象工具栏**”中的实时趋势工具。
2. 在窗口中单击，然后对角拖动鼠标绘制一个所需趋势大小的矩形框（可以用选定的任意大小绘制趋势图表，而且在屏幕上放置的趋势图表数目不限）
3. 释放鼠标，实时趋势对象即出现在窗口中：



在运行期间数据自右向左写入趋势中。

4. 双击趋势，打开其配置对话框。

趋势对象类似于在 WindowMaker 中绘制的任何其它对象。您可以通过拖动鼠标来移动趋势对象，也可以通过拖动对象的某个“句柄”来调节其大小。您可以在窗口中添加多个趋势。

配置实时趋势

第一次粘贴实时趋势对象时，系统将使用缺省配置。
一旦实时趋势配置完毕，在创建下一个趋势时将使用此相同配置作为缺省配置。

➢ **要配置实时趋势：**

1. 双击趋势，或者选定趋势，然后在“**特别**”菜单上，单击“**动画链结**”。“**实时趋势配置**”对话框出现：



如果您在实时趋势配置对话框中右击文本框，会出现一个菜单，显示可应用于选定文本的命令。

注意：“实时趋势配置”对话框中的所有项目都与趋势大小无关，并且不能在运行时修改。

2. 在“**时间间隔**”框中，输入您要水平（x 轴）显示的趋势时间长度，然后选择时间长度的增量选项。

   例如，如果您在“**时间间隔**”框中输入 30，然后选择“**分钟**”，则图表的水平时间长度将为 30 分钟。

3. 在“**样本间隔**”框中，输入用以求算趋势表达式以及更新图表的频率，然后选择与此数值相关的时间增量选项。

   例如，如果您在“**间隔**”框中输入 10，并选择“**秒**”作为时间增量，则表达式将每隔 10 秒钟求值一次。

4. 在“**颜色**”组中，单击“**图表颜色**”方框以打开 InTouch 调色板。单击调色板中您要用于趋势背景的颜色。

5. 在“**颜色**”组中，单击“**边框颜色**”方框以打开 InTouch 调色板。单击调色板中您要用于趋势边框的颜色。

   重复此过程以选定所有颜色。

6. 在“**时间刻度数**”组的“**主刻度数**”框中，输入趋势中所需的主时间刻度数，然后选择您要用于刻度线的颜色。

   主时间刻度数之间的最大时间为 65536 秒或 18 小时 12 分钟 16 秒。

   主刻度数的数值必须是**副/主刻度数**的偶数倍。

7. 在“**时间刻度数**”组的“**副/主刻度**”框中，输入每个主时间刻度内可见的副时间刻度数，然后选择您要用于刻度线的颜色。

8. 如果您要在趋势的顶部显示时间标签，请在“**时间刻度数**”组中，选择“**顶部标签**”。

9. 如果您要在趋势的底部显示时间标签，请在“**时间刻度数**”组中，选择“**底部标签**”。

   趋势既可包含顶部标签也可包含底部标签，或者没有任何标签。

10. 如果您使用了时间标签，请在“**时间刻度数**”组的“**副/主刻度**”框中，输入趋势的每个主时间刻度线的时间标签数。

11. 在“**时间刻度数**”组中，选择您要用于主时间刻度线的颜色。

12. “**值刻度数**”组的设置方法与“**时间刻度数**”组的设置相同，不同的是副刻度和主刻度会设置趋势的垂直值（y 轴）范围。此范围使用工程单位，并且应用于所有要绘制趋势的标记名。

    要在运行时显示副值和主值刻度的小数点，您必须在此处对数值格式化。例如，0.00 - 100.00。

13. 在“**表达式**”框中，输入您要用每支**笔**绘制其趋势的本地标记名或表达式。

    同一趋势中最多可见到 4 支笔。这些笔可用于显示任何本地标记名或包含一个或多个本地标记名的表达式（消息型标记将无法记录或绘制趋势）。趋势表达式对于创建自定义显示很有用处，它可用于显示各种范围的标记名。

14. 单击“颜色”方框，选择用于在趋势中绘制每个标记名的每支笔的颜色。

15. 在“**宽度**”框中，输入每支笔的象素宽度值。
    选择大于 1 的笔宽会显著降低更新和打印趋势的性能。

16. 单击“**选择显示字体**”以访问“**字体**”对话框，选择打印趋势时的字体、样式和大小。

17. 如果您要使趋势仅在活动窗口中显示时才更新，选择“**只在载入内存时更新**”。

    如果您不选择此选项，则即使趋势不在打开的窗口中，也总是会被更新。这将导致整体系统性能降低。

18. 单击“**确定**”。

- **要增强实时趋势性能：**

1. 将笔宽设置为 '1'。
2. 确保实时趋势上面没有其它对象。
3. 减少样本数。

例如，如果时间间隔设为 30 分钟，采样间隔设为 2 秒，则 30 分钟内的样本数为：

30* 60/2 = 900

如果时间间隔设为 30 分钟，采样间隔设为 5 秒，则 30 分钟内的样本数为：

30* 60/5 = 360

历史趋势

历史趋势可以为您提供一个历史时间和日期的数据“快照”。历史趋势不是动态的。与实时趋势不同，历史趋势只有在接到指示时才会更新，例如通过执行 QuickScript 或者由操作员执行单击按钮等动作时。

一次最多可以为八个标记名（笔）绘制趋势，但对显示的趋势数目没有限制。您可以充分灵活地设计趋势界面。您可以创建一个“指示器”，让操作员在趋势上移动，以根据指示器的当前位置来访问各种数据。例如，当操作员将“指示器”放在具有可见数据的趋势区域中时，将返回所有当前绘制趋势在该位置的时间和数值。

您也可以创建按钮在指示器或数据（例如最大值到最小值）之间放大或缩小。您可以显示整个图表或指示器之间的区域的平均或者标准偏差。历史趋势也可以滚动任意的时间量。您可以创建自定义刻度，并将其链接至“**.MinEU**”和“**.MaxEU**”标记名点域，以显示最小和最大工程单位。

分布式历史系统将历史趋势的检索能力扩展到了远程记录数据库。此系统允许在单个趋势中显示多个历史记录数据库信息。

注意：您必须为标记名字典中的每个标记名启用“**记录数据**”选项，以便绘制其趋势。

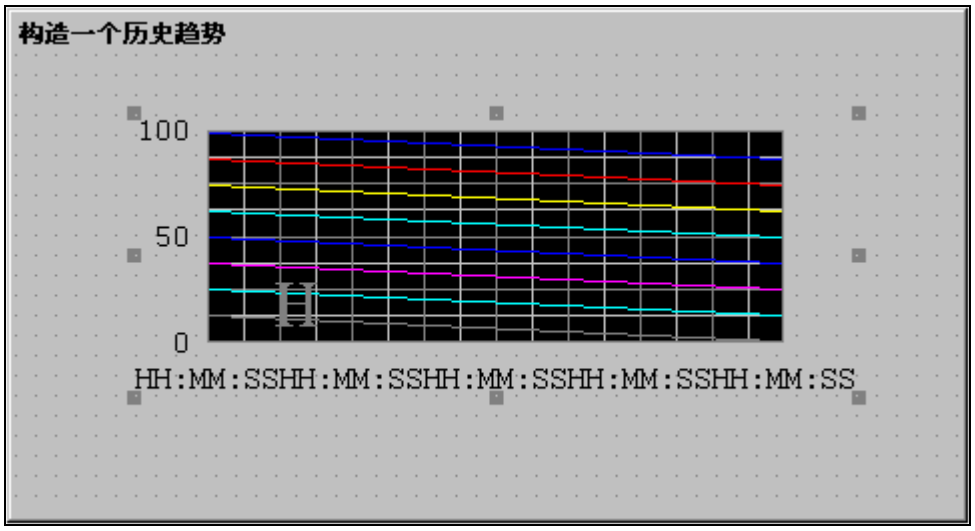
有关记录标记名的详细信息，请参阅“记录标记名”。

另请参阅“配置历史记录属性”。

创建历史趋势

➢  **要创建历史趋势：**

1. 选择“**向导工具拦**”中的历史趋势工具。历史趋势工具用于绘制历史趋势对象。
2. 在窗口中单击，然后对角拖动鼠标绘制一个所需趋势大小的矩形框
   - 您可以绘制所选任意大小的趋势图表，也可以在窗口中添加多个趋势。
3. 释放鼠标，历史趋势即出现在窗口中：



- 趋势对象类似于在 WindowMaker 中绘制的任何其它对象。您可以通过拖动鼠标来移动趋势对象，也可以通过拖动“句柄”来调节其大小。

配置历史趋势

第一次粘贴历史趋势对象时，系统将使用缺省配置。一旦趋势配置完毕，下次创建趋势时将用此相同配置作为缺省配置。

➢ **要配置历史趋势：**

1. 双击趋势，或者选定趋势，然后在“**特别**”菜单上，单击“**动画链结**”。“**历史趋势配置**”对话框出现：



如果您右击任意历史趋势配置对话框中的文本框，一个菜单会出现，显示可应用于选定文本的命令。

2. 在“**历史标记**”框中，输入您要用于趋势的标记名。

   如果输入的标记名当前尚未在标记名字典中定义，则系统会询问您是否要现在进行定义。如果您选择“**是**”以现在定义标记名，InTouch 会自动显示“**标记名字典**”对话框，并缺省将标记名类型设置为“**历史趋势**”（标记名必须定义为**历史趋势**类型）。每个历史趋势必须使用不同的标记名。

3. 在“**初始化时间长度**”框中，输入需要水平（x 轴）显示的趋势时间长度，然后选择时间长度的增量选项。

   例如：如果您在“**初始化时间长度**”框中输入 30，然后选择“**分钟**”，则图表的水平时间长度将为 30 分钟。

4. 如下选择您要用于趋势的“**初始化显示模式**”：

   最小/最大 图表上的每个象素将显示以象素表示的时间点的最小到最大范围。

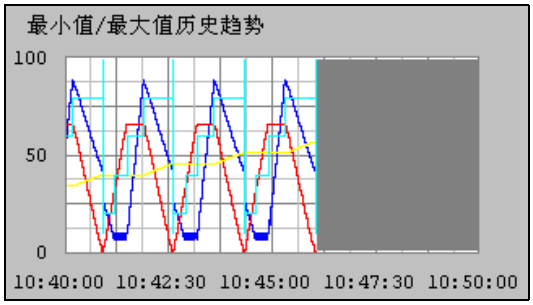
   平均 显示每个象素的平均值，例如时间段。

5. 在“**颜色**”组中，单击“**图表颜色**”方框以打开 InTouch 调色板。单击调色板中您要用于趋势背景的颜色。

6. 在“**颜色**”组中，单击“**边框颜色**”方框以打开 InTouch 调色板。单击调色板中您要用于趋势边框的颜色。

   重复此过程以选定所有颜色。

   右边的空白区域表示这段时间内没有收集数据，原因是 WindowViewer 没有运行或者“历史记录”没有激活。



7. 在“**时间刻度数**”组的“**主刻度数**”框中，输入趋势中所需的主时间刻度数，然后选择您要用于刻度线的颜色。

   主刻度数的数值必须是**副/主刻度数**的偶数倍。

8. 在“**时间刻度数**”组的“**副/主刻度**”框中，输入每个主时间刻度内可见的副时间刻度数，然后选择您要用于刻度线的颜色。

9. 如果您要在趋势的顶部显示时间标签，请在“**时间刻度数**”组中，选择“**顶部标签**”。

10. 如果您要在趋势的底部显示时间标签，请在“**时间刻度数**”组中，选择“**底部标签**”。趋势既可包含顶部标签也可包含底部标签，或者没有任何标签。

11. 如果您使用了时间标签，请在“**时间刻度数**”组的“**副/主刻度**”框中，输入趋势的每个主时间刻度线的时间标签数。

12. 在“**时间刻度数**”组中，选择您要用于主时间刻度线的颜色。

13. “**值刻度数**”组的设置方法与“**时间刻度数**”组的配置相同。副值和主值刻度确定了趋势的垂直坐标值（Y 轴）。此范围使用工程单位，并且应用于要绘制趋势的所有标记名。

    - 要在运行时显示副值和主值刻度的小数点，必须在此处对数值格式化。例如，0.00 – 100.00。



14. 在“**标记名**”框中，双击打开“**供应器列表**”对话框。选择您要用于此**笔**的标记名供应器。单击“**确定**”。一个浏览器会出现，显示该供应器的本地或远程标记名。双击选定所需的标记名。重复此过程，选定您要让每支**笔**绘制其趋势的标记名或表达式。

    同一趋势中最多可看见八支笔（消息型标记无法记录或绘制趋势）。

注意：FactorySuite Productivity Pack 包括一个允许您用 16 支笔画图的笔配置网格。有关详细信息，请参阅您的*《Productivity Pack 用户指南》*。

---

15. 单击颜色框，选择趋势中每支笔用于绘制标记名的颜色。

16. 在“**宽度**”框中，输入每支笔的象素宽度值。

注意：选择的笔宽如果大于 1，会显著降低屏幕更新和打印的性能。

17. 如果您希望操作员能够在运行时改变趋势的配置，选择“**允许运行时更改**”。这些更改包括笔的赋值、起始日期、时间等。

    如果您选择了此选项，则当操作员在运行时单击趋势（或者在使用触摸屏时触摸）时，“**运行时设置**”对话框会出现，操作员即可以更改趋势。

    有关详细信息，请参阅“在运行时更新历史趋势”。

18. 单击“**选择显示字体**”以访问“**字体**”对话框，选择您要用于趋势显示的字体、样式和大小。

19. 单击“**确定**”。

使用历史趋势向导

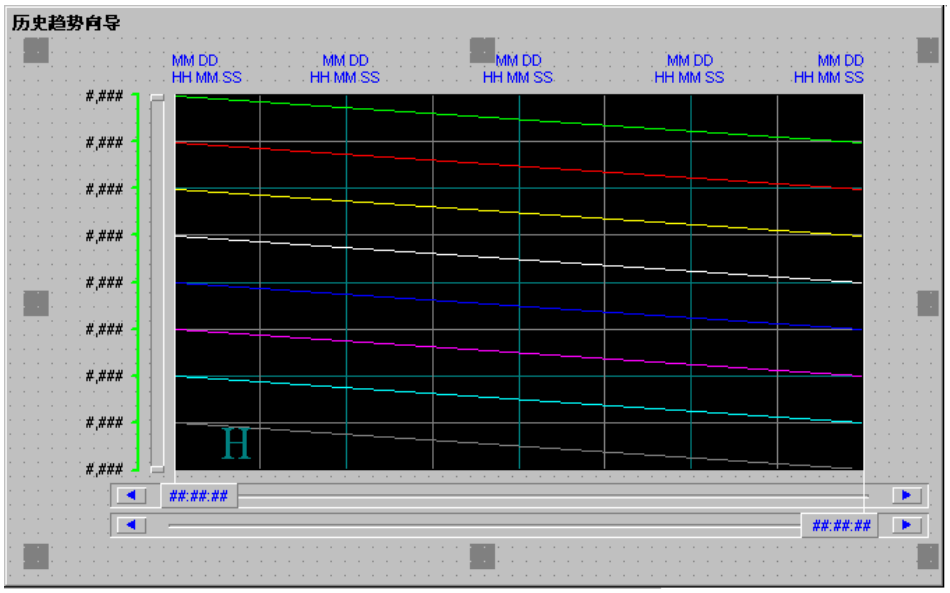
InTouch 为您提供了一种快速和简单的建立历史趋势的方法：趋势向导。趋势向导使得您只需单击几下鼠标，就可以配置一个包含指示器和缩放等工具的全功能历史趋势。

➢ **要使用历史趋势向导：**

1. 单击“**向导工具栏**”中的向导工具。“**向导选择**”对话框出现：



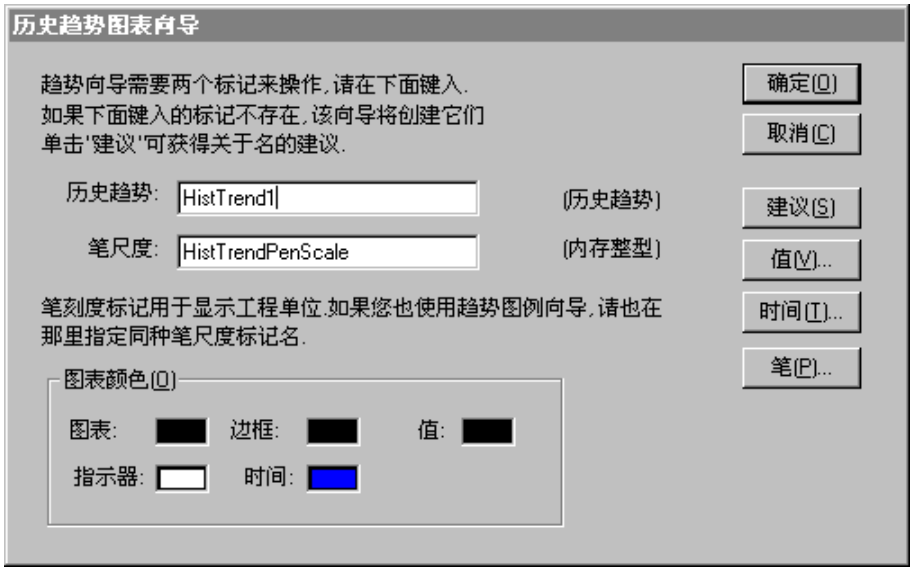
2. 在向导列表中选择“**趋势**”，以显示可用的趋势向导。
3. 选择“**含指示器和刻度的历史趋势**”向导，然后单击“**确定**”。对话框关闭，窗口重新出现，光标处于“粘贴”模式。
4. 单击窗口以粘贴趋势向导。



趋势对象类似于在 WindowMaker 中绘制的任何其它对象，可以通过拖动鼠标来移动，也可以通过拖动对象的“句柄”来调节大小。一个窗口中可以添加多个趋势。

5. 您现在可以开始配置趋势向导。

6. 双击趋势向导以打开“**历史趋势图表向导**”配置对话框。



如果您右击向导配置对话框中的文本框，会出现一个菜单，显示可用于选定文本的命令。

7. 输入配置趋势所需的信息，然后单击“**确定**”。

   如果您要让向导自动填入配置值，单击“**建议**”。历史趋势向导的配置值与那些使用“**绘图对象工具栏**”上的 WindowMaker 趋势工具来建立历史趋势对象图时的设置相同。

8. 要在趋势中添加缩放和移动功能，或者在趋势中添加笔控制，请分别使用“趋势缩放/平移面板”和“趋势笔图例”向导。要使所有这些部件配合工作，必须使用“**历史趋势**”标记名。

   与所有的 InTouch 向导一样，该向导也可以分解为不同的部件。

➢  **要分解向导：**

1. 选择历史趋势向导。
2. 在“**排列**”菜单上，单击“**分解单元**”，或者单击“**排列工具栏**”中的分解单元工具。
3. 您随后就可以根据需要来自定义了。

记录标记名

在 WindowViewer 中，要记录的标记名值的变化一旦超过指定的**记录死区**，即会被写入历史记录文件中。缺省条件下，无论有无变化，系统将每小时写入一次。要使标记名的值写入历史记录文件，必须将其配置为记入标记名字典中。

对于整型和实型（浮点）标记名类型，您可以在其各自的详细对话框中设置“**记录死区**”。“**记录死区**”控制标记名的值改变多少个工程单位数，才会被记入磁盘中。

➢ **要配置需记录的标记名：**

1. 在“**特别**”菜单上，单击“**标记名字典**”，或者在应用程序浏览器中，双击“**标记名字典**”。“**标记名字典**”对话框出现：



2. 打开所需的标记名定义，然后选择“**记录数据**”。

注意：要实际记录标记名，您必须启用记录选项（如下节所述）。

如果您从记录标记名改为不记录标记名，则已记录的标记名数据将不可访问。

在 WindowMaker 运行时对 WindowViewer 记录所做的任何更改，只有在重新启动 WindowViewer 后才会生效。

最小/最大工程单位对于显示历史趋势数据非常重要。历史趋势显示 0 - 100% 的工程单位范围。

配置历史记录属性

要使您在“**记录数据**”选项中配置的标记名写入历史记录文件，您必须选定全局记录功能。

➢ **要配置历史记录：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**历史记录**”。“**历史记录属性**”对话框出现：
2. 要快速访问此对话框，在应用程序浏览器的“**配置**”下，双击“**历史记录**”。



如果您右击任意历史趋势配置对话框中的文本框，一个菜单会出现，显示可应用于选定文本的命令。

3. 选择“**允许历史记录**”以启用全局标记名记录。
4. 在“**保持记录文件时间**”框中，输入您要将记录文件保存到磁盘的天数（今天之前）。

   InTouch 会在每天（24 小时）创建并保存两个历史记录文件。因此，当您设置此选项时必须考虑磁盘空间。如果硬盘没有足够的可用空间来保存历史记录文件，记录将停止，您必须清理磁盘空间才能重新开始记录。通过将 **$HistoricalLogging** 内部标记名链结到按钮或 QuickScript，或者使用 WindowViewer 中的“**重新启动历史日志**”命令，您可以在运行时启动和停止历史记录。

   例如，如果您输入 10，而今天是 12 号，则从 2 号到 12 号（包括今天在内的 10 天）的记录文件将保存到磁盘上。1 号的文件将被自动删除。如果您输入 0，则记录文件将无限期保留。

5. 如果您要将历史记录文件保存到应用程序目录中，请选择“**在应用程序目录中存储记录文件**”。或者，选择“**在指定目录中存储记录文件**”，并输入您要使用的目录的完整路径。

注意：此输入项必须是一个 DOS 路径，例如 C:\HISTLOG ；或者，如果您执行分布式历史，该路径必须是一个通用命名惯例 (UNC) 路径，例如 \\node\share\directory。

如果配置为将历史数据写入主应用程序节点的“应用程序目录”，则所有 NAD 节点会尝试将其历史数据写入主应用程序。为避免出现此问题，需在每个 NAD 节点上配置将历史数据写入本地目录，而**不是**主应用程序节点。

有关分布式历史的详细信息，请参阅“分布式历史系统”。

缺省状态下，历史记录文件命名如下：

YYMMDD00.LGH and YYMMDD00.IDX

此处：

YY 显示创建文件的年度 (99,01)

MM 显示创建文件的月份 (01-12)

DD 显示创建文件的日期 (01-31)

00 始终显示零

例如，如果文件创建于 1997 年 10 月 31 日，则文件将命名如下：

97103100.LGH

和

97103100.IDX

注意：本版本的 InTouch 支持扩展名为 .LGH 及 .IDX 的较新记录文件。较早版本的 InTouch 使用扩展名 .LOG 作为记录文件。

---

6. 在“**登录节点名**”框中，输入将记入历史记录文件的 NetDDE 节点名（不是计算机名）。

7. 在“**缺省%打印页**”框中，输入页面大小占趋势大小的百分比。

   例如：如果您在打印历史趋势时使用 50%，则打印的趋势将占半页（垂直和水平）。该大小尺寸的打印输出时间约为整页打印的 1/4。

   影响打印历史趋势性能的因素有很多。最主要的因素是打印页面上的图表大小。您可以通过减小所用页面的百分比来提高性能。

8. 在“**最长连续打印时间**”框中，输入历史趋势打印模块连续打印所需的毫秒数（处理器时间间隔）。
9. 在“**打印等候时间**”框中，输入历史趋势打印模块在打印输出之间等待的毫秒数。
10. 如果您要使用彩色打印机或绘图仪，请选择“**打印时总是使用彩色**”。
11. 单击“**选定打印字体**”，以访问 Windows“**字体**”对话框。
12. 单击“**确定**”以保存设置并关闭对话框。

控制历史记录频率

当通过“**历史记录**”命令启用历史记录时，如果标记名的值没有发生变化，则指定要记录的所有标记名的值会每隔一小时自动写入历史记录文件中。如果标记名发生变化，并且变化的工程单位值超过其记录死区值（缺省为 0），则会在变化时自动记录。您可以通过在 **INTOUCH.INI** 文件（位于您的应用程序目录）的 **[INTOUCH]** 部分中添加参数 **ForceLogging=#** 来改写此缺省值。此参数值以分钟表示，可以设置为 5 和 120 之间的一个值。

要记录标记名的*当前*值（即使变化小于或等于记录死区值），请添加参数 **ForceLogCurrentValue=1**。例如：

通过添加参数 **ForceLogging=15** 和 **ForceLogCurrentValue=1**，当前标记名值会至少每隔 15 分钟或者是在标记名值变化时写入历史记录文件。

注意：“ForceLogCurrentValue”参数强制记录标记名的当前值，即使变化小于或等于记录死区值。

有关记录死区的详细信息，请参阅“记录标记名”。

在运行时配置历史趋势

如果您在配置历史趋势时选择“**允许运行时更改**”选项，则该趋势在 WindowViewer 中将是“触控”的，操作员将可以更改笔的赋值、更改开始日期和时间等。

➢ **要在运行时配置历史趋势：**

1. 在 WindowViewer 中单击趋势，“**历史趋势设置**”对话框出现：



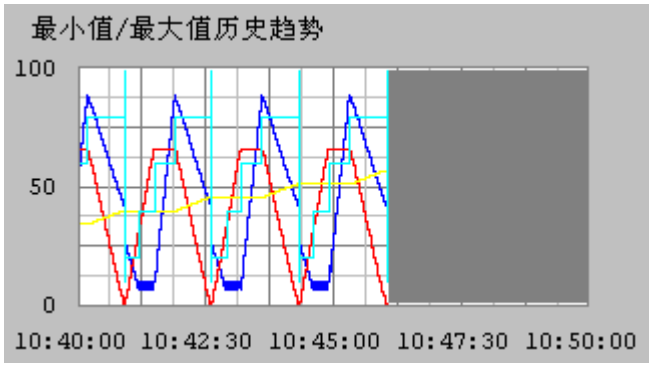
2. 在“**图表开始**”组中，输入图表的开始日期和时间。
3. 选择图表的**显示模式**。

注意：趋势的显示方式会影响性能。首要因素是创建趋势时所画线条的长度。线越长，创建图表所需的时间越长。线宽也是影响性能的因素，画宽线条的时间要长很多。“**最小/最大**”或“**平均/散点**”图表通常比“**均值/棒图**”的速度要快得多。

下面举例说明了三种绘图方式：

“最小/最大”历史趋势

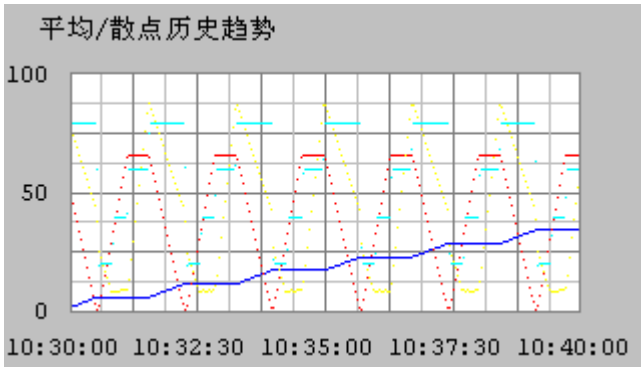
这种模式以工程单位的百分比为纵坐标，以时间长度为横坐标来显示趋势或变化，并且着重时间流和变化率，而非变化量。



注意：右边的空白区域表示这段时间内没有收集数据，原因是 WindowViewer 没有运行或者“历史记录”没有激活。

“平均/散点”历史趋势

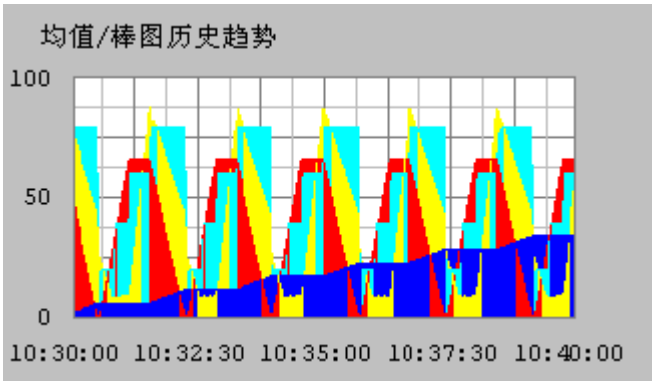
这种模式显示特定时间段内的时间点的平均值。



有关平均/散点历史趋势的详细信息，请参阅“平均/散点历史趋势计算”。

“均值/棒图”历史趋势

这种模式以条形图的方式显示特定时间段内时间点的平均值。



4. 在“**图表长度**”框中，输入显示在趋势中的水平（x 轴）时间长度，然后选择长度的时间增量。例如，如果您输入 1 并选择“**小时**”，则趋势长度将为 1 小时。

5. 在“**图表范围**”框中，输入趋势缩放的工程单位刻度的百分比（显示在趋势中的 y 轴垂直范围）。

注意：范围的单位是工程单位刻度的百分比。取值范围为 0 到 100。例如，如果您要记录选定标记从 40% 到 45% 的变化趋势，请分别在“**最小%**”和“**最大%**”范围框中输入 40 和 45。

---

6. 单击每个“**笔#**”来选择您要用该笔绘制趋势的标记名。标记浏览器以过滤选择模式显示：



注意：对于选定的标记源，只有在定义时选定了“**记录数据**”选项的标记名才会显示。

7. 双击您要在趋势上用选定笔画图的标记名，或选取标记名，然后单击“**确定**”。“**历史趋势设置**”对话框重新出现，显示您原先单击的“**笔#**”按钮旁边的选定标记名。

   您可以单击“**过滤器**”箭头来打开已定义的过滤器列表，以便在标记浏览器中显示过滤项目。此列表的第一项是**<无>**，表示没有使用过滤器。对于选定的标记源，只有在定义时选定了“**记录数据**”选项的标记名才会显示。

   当您使用过滤器，或者单击“**过滤器**”按钮来创建新的过滤器时，标记浏览器会重新显示满足所选标记源的过滤器中指定的标准、并且选定了“**记录数据**”选项的标记名。

   有关标记浏览器和过滤器的详细信息，请参阅第 4 章“标记名字典”。

8. 单击“**打印**”以打印历史趋势。

   打印操作在后台进行，WindowViewer 可以继续处理所有其它输入。WindowViewer 在打印时会在菜单中添加两个菜单项：“**取消打印**”和“***X*% 完成**”。单击“**取消打印**”将取消当前的打印作业。

   选择“**打印**”后，在 WindowViewer 菜单栏中的“**取消打印**”和“**X % 完成**”菜单项消失之前，请勿更改趋势。在此期间，WindowViewer 会将趋势信息保存到内存中以便打印。一旦这两个菜单项从菜单栏消失，您就可以更改趋势更不会影响进行中的打印作业。

   您可以创建一个按钮来打印历史趋势，方法是将其链接至执行 **PrintHT** QuickScript 函数的动作 QuickScript。

   ```
   PrintHT(HistTrendTagname);
   ```

   打印操作使用当前历史趋势作为打印的基础。因此，如果“**历史趋势设置**”对话框中的任何字段发生变化，“**打印**”按钮将失效。您不能打印设置过程中所做的更改，除非您单击“**历史趋势设置**”对话框中的“**确定**”，然后再次访问它并单击“**打印**”。

注意：使用“**打印**”选项或 PrintHT() 函数打印历史趋势时，不会打印 y 轴和 x 轴值。要打印 y 轴和 x 轴值，您可以使用 PrintWindow() 或 PrintScreen()。

有关打印历史趋势的详细信息，请参阅“配置历史趋势打印”。

“平均/散点”历史趋势计算

平均/散点模式是历史趋势对象的三种显示模式之一。当标记名定义为启用“**记录数据**”选项时，标记名的值会被记入历史记录文件。记入的数据可从历史记录文件读取并显示在历史趋势对象上。在平均/散点模式下，显示在对象上的点数取决于标记名值的变化速度。历史趋势对象划分成许多刻度，刻度数限制为 2000，因此，可显示的最大点数为 2000。

例如，考虑某个应用程序，其要绘制趋势的标记名在一小时的时间间隔内，每隔一秒记录一次。显示缓冲区内的点数限制为 2000，这就意味着 3600 秒的时间范围分隔成 3600/2000 或 1.8 秒的时间间隔。因此，所画标记名的值在此时间间隔内的实际平均值为 1.8 秒。

如果标记名的值在此 1.8 秒的时间间隔内发生变化，则系统会计算平均值并在历史趋势对象上显示一点。如果值为常数，则会显示为一条线。

考虑另一方案，其中的标记名图表长度为 1 秒。因为显示缓冲区中的点数限制在 2000，所以 1 秒的时间范围被分割成 1/2000 或 0.0005 秒的时间间隔（即 0.5 毫秒舍入为 1 毫秒）。因此，所画标记名的值在此时间间隔内的实际平均值为 1 毫秒。

当游标与历史趋势的图表长度关联并且快速移动时，相比游标慢速移动时显示的点数，所显示的点数减少了。

在运行时更新历史趋势

在 WindowViewer 中，当历史趋势首次显示时，它会显示指定配置的数据。与实时趋势不同，历史趋势不会连续更新。要在显示初始数据后更新历史趋势，**必须对趋势进行更改**。您可以使用下列任何方法来更新趋势：

- **要在运行时更新历史趋势：**

1. 选择“**历史趋势设置**”对话框中的“**允许运行时更改**”选项（在 WindowMaker 中），以使操作员可以手动更改趋势的时间和（或）日期来强制更新。
2. 在 QuickScript 或按钮中使用下面的语句，以允许操作员更新图表：

   ```
   Hist_TrendTag.UpdateTrend = 1
   ```

3. 在 QuickScript 或按钮中使用下面的语句：

   ```
   HTUpdateToCurrentTime(Hist_Tag);

   HTScrollLeft(Hist_Tag,Percent);

   HTScrollRight(Hist_Tag,Percent);

   HTZoomIn(Hist_Tag,LockString);

   HTZoomOut(Hist_Tag,LockString);

   HTSetPenName(Hist_Tag,PenNum,Tagname);
   ```

4. 更改下列任意趋势标记名点域：

   .ChartStart
   .ChartLength
   .MaxRange
   .MinRange
   .Pen1-.Pen8

有关使用 QuickScript 函数和点域的详细信息，请参阅您的联机 *《InTouch 参考指南》*。

配置历史趋势打印

影响打印历史趋势性能的因素有很多，其中最主要的因素是打印页上的趋势大小。趋势的显示模式也影响打印性能。“**最小/最大**”或“**平均/散点**”打印输出通常要比“**平均/棒图**”趋势输出快得多。趋势图上的线条越长越宽，则打印时间越长。

因为打印操作在“后台”进行，InTouch 会为打印进程分配一定的时间，同时为其它进程分配一定的时间。此进程时间取决于配置历史记录时在“**最长连续打印时间**”和“**打印等候时间**”框中设置的值。

换句话说，InTouch 会用“**最长连续打印时间**”框中指定的毫秒数来处理打印，然后用“**打印等候时间**”字段中指定的毫秒数来处理其它请求。为提高打印优先级，您可以增大“**最长连续打印时间**”的值，同时减小“**打印等候时间**”的值。如要降低打印优先级，则刚好相反。

➢ **要配置历史趋势打印：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**历史记录**”。“**历史记录属性**”对话框出现：

   ☞ 要快速访问此对话框，在应用程序浏览器的“**配置**”下，双击“**历史记录**”。



   ☞ 如果您右击任意历史趋势配置对话框中的文本框，一个菜单会出现，显示可应用于选定文本的命令。

2. 在“**缺省 % 打印页**”文本框中，指定用于打印趋势的页面百分比。

   如果您在此框中输入 50，WindowViewer 会使用半页来打印（垂直和水平）。该尺寸的打印输出时间约为整页打印的 1/4。

   您可以研究使用 **PrintWindow** QuickScript 函数来作为一种打印替代方式。

3. 在“**最长连续打印时间**”框中，输入打印模块连续打印的处理时间（毫秒）。

4. 在“**打印等候时间**”框中，输入打印模块在使用另一处理时间之前等待的时间（毫秒）。

   影响打印性能的另一个因素是您为趋势选定的背景颜色。在大多数情况，白色背景在打印时要快得多。您最好试一下白色和其它颜色的对比打印，看是否有明显差别。

5. 单击“**选定打印字体**”以访问“**字体**”对话框，选择您要用于打印输出的字体、样式和大小。

6. 单击“**确定**”。

历史趋势点域

许多**点域**只适用于历史趋势标记名。下面向您简要介绍这些历史趋势**点域**。

点域	描述
.ChartLength	可读/写整型标记名点域，用于控制显示在历史趋势图中的时间长度。**.ChartLength** 以秒为单位显示图表长度。
.ChartStart	可读/写整型标记名点域，用于控制开始时间和（或）滚动相应的历史趋势。**.ChartStart** 显示自 70 年 1 月 1 日午夜 12 点以来已过的时间（以秒为单位）。
.DisplayMode	可读/写模拟标记名点域，用于决定在趋势中显示值的方式。
.MaxRange, .MinRange	读/写实型标记名点域，用于表示当前绘制趋势的每个标记名所显示的标记名工程单位范围的百分比。**.MaxRange** 和 **.MinRange** 范围限于从 0 到 100，并且 **.MinRange** 恒小于 **.MaxRange**。如果某个域被赋于小于 **0** 或大于 **100** 的值，这些值将定为 **0** 或 **100**。如果 **.MinRange** 大小或等于 **.MaxRange**，趋势将不显示任何数据。
.Pen1 - .Pen8	读/写**标记 ID** 类型的标记名点域，用于控制每支笔当前正在绘制历史趋势的标记名。一个标记 ID 型标记名只能等价于另一标记 ID 型标记名。它不能与其它标记名类型混用，除非其它标记名添加了 **.TagID** 扩展名。**.TagID** 不能用于远程历史供应器标记名。

点域	描述
.ScooterLockLeft	读/写离散型点域。当此点域值为“真”时，则右指示器不能移向左指示器位置的左边。（0=假、1=真）。
.ScooterLockRight	读/写离散型点域。当此点域值为“真”时，则左指示器不能移向右指示器位置的右边。（0=假、1=真）。
.ScooterPosLeft	读/写实型点域，代表左指示器位置（范围从 0.0 到 1.0）。
.ScooterPosRight	读/写实型点域，代表右指示器位置（范围从 0.0 到 1.0）。
.TagID	读/写标记 ID 标记名点域，与历史趋势 .Pen1-.Pen8 标记 ID 标记名一起使用，监视和（或）控制某支笔当前绘制趋势的标记名。
.UpdateCount	只读整型点域，当此趋势的检索完成时，此标记名点域递增。
.UpdateInProgress	只读离散型点域，显示历史数据检索状况（0=未检索，1=正在检索)。
.UpdateTrend	读/写离散型标记名点域，当设置为 1 时，将用所有当前值来更新历史趋势。

有关使用**点域**的详细信息，请参阅您的联机 *《Intouch 参考指南》*。

历史 QuickScript 函数

您可以使用若干个内部函数来指定由每支笔绘制趋势的标记名、显示指示器位置的数值，以及滚动趋势某个百分比等。

有关如何使用这些函数及其有效参数的完整示例，请参阅您的 *《InTouch 参考指南》*。

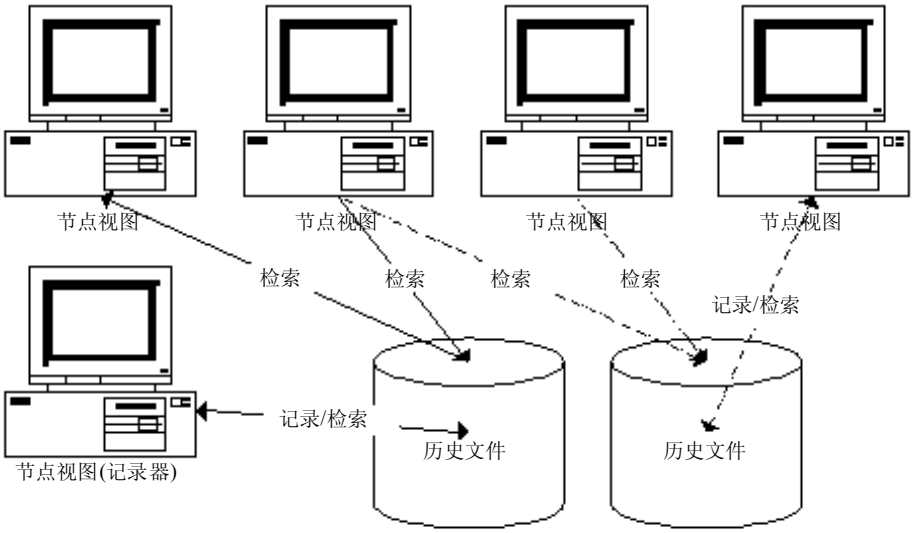
函数	描述
HTGetLastError	确定最后一次检索指定笔时是否出现错误。
HTGetPenName	返回指定趋势的指定笔号当前所用的标记名。
HTGetTimeAtScooter	返回由 *ScootNum* 和 *ScootLoc* 指定的指示器位置处的样本自 1970 年 1 月 1 日 GMT 00:00:00 以来的时间（以秒为单位）。当 *UpdateCount*、*ScootNum* 和 *ScootLoc* 中任一参数发生变化时，会对表达式重新求值。这可以确保在新的检索或指示器移动后对表达式重新求值。
HTGetTimeStringAtScooter	返回由 *ScootNum* 和 *ScootLoc* 指定的指示器位置处的样本包含时间/日期的字符串。当 *UpdateCount*、*ScootNum* 和 *ScootLoc* 中任一参数发生变化时，会对表达式重新求值。这可以确保在新的检索或指示器移动后对表达式重新求值。字符串的格式决定返回值的内容。
HTGetValue	返回整个趋势的指定笔的请求类型值。
HTGetValueAtScooter	返回指定的指示器位置，趋势和笔号处的样本的请求类型值。使用 *UpdateCount* 参数，可在检索完成后对表达式求值。
HTGetValueAtZone	针对包含某一趋势指定笔的左右指示器位置之间的数据，返回请求时间值。
HTScrollLeft	将趋势的起始时间值设置为比当前起始时间早趋势宽度的一定百分比。其效果是将图表的日期/时间向左滚动给定的百分比。

函数	描述
HTScrollRight	将趋势的起始时间值设置为比当前起始时间晚趋势宽度的一定百分比。其效果是将图表的日期/时间向右滚动给定的百分比。
HTSetPenName	对趋势笔指定不同的标记名。
HTUpdateToCurrentTime	通过使结束时间等于当前时间来检索和显示数据。开始时间将等于结束时间减去图表的宽度。
HTZoomIn	计算新图表的宽度和开始时间。如果趋势的 .ScooterPosLeft 为 0.0 而 .ScooterPosRight 为 1.0，那么新图表的宽度将等于旧图表宽度除以 2。新的开始时间将根据 *LockString* 的值进行计算。
HTZoomOut	计算新图表的宽度和开始时间。新图表的宽度等于旧图表宽度乘以 2。新的开始时间将根据 *LockString* 的值进行计算。

分布式历史系统

InTouch 提供的分布式历史系统可从任何 InTouch 应用程序甚至跨网络检索数据。此系统通过同时从多个历史数据库远程检索数据，扩展了标准 InTouch 历史记录系统的能力。这些数据库称为历史供应器。您最多可以同时显示八个历史供应器，每个供应器对应历史趋势图中的一支笔。

使用分布式历史系统提供的功能，您可以轻易地配置一个网络系统来访问多个历史记录供应器：



每个分布式历史文件限于在一个节点写入（记录）文件。但是，可查看该文件的 InTouch 节点数目或类型并没有限制。

只有在 InTouch 第 5.6 版或以上版本开发的应用程序才可以成为历史供应器。要远程查看较早版本的历史文件，您首先必须将该应用程序转换成第 5.6 版或以上的版本。

远程节点从历史文件中检索数据可能不能看到最近一个小时的数据（取决于记录器节点的时间）。远程趋势只可查看已经写入记录节点磁盘的数据。

在采集了标记名的 22 个样本之后，选定了“记录数据”选项的各个标记名会自动写入磁盘。如果执行 **HTUpdateToCurrentTime()** 函数，则无论是否采集了足够的样本，数据都会写入磁盘。缺省条件下，数据每隔一个小时写入磁盘一次。您可以通过在 intouch.ini 文件中添加下面一行来更改此时间间隔：

```
ForceLogging=X;
```

此处 **X** 是分钟值，可设置为介于 5 和 120 之间的任何时间间隔。

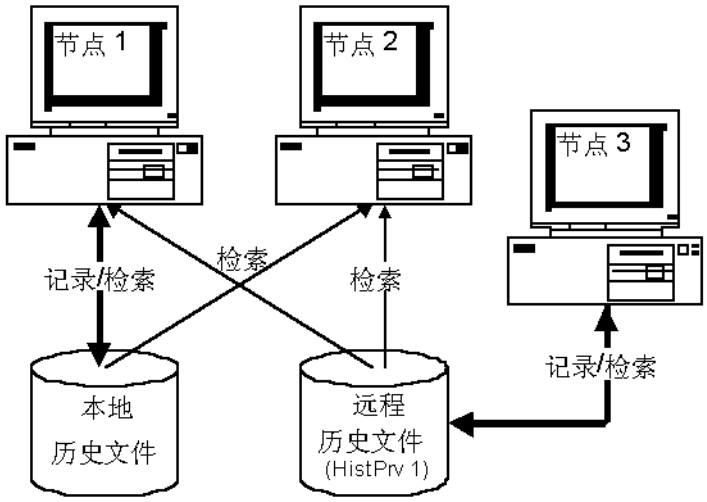
注意：当您使用分布式历史时，必须运行 Wonderware NetDDE Helper 服务。

有关 Windows NT 服务的详细信息，请参阅附录 A“InTouch Windows NT 服务概览”。

使用分布式历史系统

下图说明如何设置分布式历史系统。此系统是使用网络应用程序开发工具 (NAD) 来分发应用程序的典型分布式应用程序。

有关 NAD 的详细信息，请参阅第 3 章“创建分布式应用程序”。



节点 1 和 2 包含相同 InTouch 应用程序的副本，但是应用程序配置为只允许节点 1 记录**到**本地历史文件，而两个节点都可以**从**本地历史文件或远程历史文件中检索数据。节点 3 也可以记录到远程历史文件位置或者从远程历史文件检索数据。此供应器（节点 3）指定的名称为 **HistPrv1**。节点 1 既是开发站也是运行站，而节点 2 只是运行站。

创建此应用程序所需的主要步骤包括：

1. 创建历史供应器列表。
2. 创建并配置历史趋势对象。
3. 配置应用程序的分布式记录。
4. 分发应用程序。

所有这些步骤均将在本章中向您介绍。

分发您的应用程序

您可以手动或使用 NAD 系统来分发您的应用程序。当分发应用程序时，历史供应器列表文件将作为应用程序的一部分进行发发。

有关使用 NAD 的详细信息，请参阅第 3 章“创建分布式应用程序”。

在分发应用程序之后，您可以运行视图节点并检索本地标记名和远程历史供应器的标记名。应用程序可在所有视图节点上运行，只有记录节点可记录历史记录文件，其它节点只能从中读取。

配置分布式历史供应器列表

您想要从中检索历史数据的每个远程历史供应器都必须在 InTouch 历史供应器列表中注册。此列表允许您指定每个历史供应器的名称和网络位置。每次您在 InTouch 中引用历史供应器时，都需要使用这些名称。

➢ **要配置历史供应器列表：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**分布式名称管理器**”。“**分布式名称管理器**”对话框出现：

   - 要快速访问此对话框，在应用程序浏览器的“**配置**”下，双击“**分布式名称管理器**”。



   - 如果您右击任意历史趋势配置对话框中的文本框，一个菜单会出现，显示可应用于选定文本的命令。

2. 单击“**分布式历史**”标签以激活分布式历史供应器属性页。

3. 在“**供应器名称**”框中，输入您要用于新历史供应器的名称。

4. 如果您要访问 InTouch 应用程序的记录文件，请选择“**InTouch 供应器**”并在“**UNC**”框中输入 InTouch 应用程序目录的 UNC（通用命名惯例）路径。有效格式为：

   \\Node\Share\ApplicationPath

   在上面的例子中，**HistPrv1** 的路径包括节点名“HistNode”、共享“C$”，以及应用程序路径“\Apps\HistApp”。

   如果 UNC 位置是口令保护的，您必须首先使用 Windows 资源管理器来建立连接。

   有关 UNC 路径的详细信息，请参阅第 3 章“创建分布式应用程序”。

5. 选择“**InSQL 供应器**”以访问 IndustrialSQL 服务器运行时数据库的数据，然后单击“**配置 InSQL 供应器**”。“**InSql History Provider 属性**”对话框出现：



   缺省条件下，会显示最近一次成功登录的登录参数。如有必要，可修改登录参数以连接到选定的 IndustrialSQL 服务器。

6. 在“**供应器名**”框中，输入您要用于 InSQL 供应器的名称（用户自定义）。

7. 在“**数据源**”框中，输入 Industrial SQL Server 数据库所在的节点名（不超过 35 个字符）。

8. 在“**用户**”框中，输入您的登录帐号名。

9. 在“**口令**”框中，输入登录帐号的口令。

10. 在“**再次输入口令**”框中，再次输入口令以进行确认。

11. 单击“**测试**”来验证与 InSQL 服务器的连接。一个消息框会出现，通知您连接是成功还是失败。单击“**确定**”，关闭此消息框。

12. 单击“**确定**”。

注意：用户帐号包含用户名和口令。用户帐号必须已授权检索数据，否则登录将失败。有关用户帐号的详细信息，与与您的系统管理员联系。

当在 InSQL 数据库中查询 InTouch 趋势对象的数据时，程序会检索指定时间周期（一行计数）的 1000 个等距行并将其画在历史趋势图上。历史趋势上显示的标记名的最小和最大值不一定是标记名的实际最小和最大值。

HistData **不能**从 InSQL 供应器检索历史信息。

有关历史供应器的详细信息，请参阅“配置分布式历史供应器列表”。

13. 单击“**添加**”。

当 WindowViewer 显示历史供应器名时，历史系统会在供应器列表中查找该名称。如果名称在列表中，它会从该供应器中读取历史记录文件。如果名称不存在，则引用将被忽略，并在 Wonderware Logger 中写入一条错误消息。此时，本地 InTouch 应用程序将被视为历史供应器，它不必在此文件中配置。

有关 Wonderware Logger 的详细信息，请参阅您的*《FactorySuite 系统管理员指南》*。

配置远程历史供应器

历史趋势支持本地和远程历史供应器标记名的显示。

- **要显示远程历史供应器的标记名：**

  1. 双击历史趋势，以访问“**历史趋势配置**”对话框。
  2. 在每支笔的“**标记名**”框中，按下面的格式输入标记名：

     `HistPrv1.tagname`

     每支笔可以引用不同的远程历史供应器。例如，当您配置历史趋势时，如果您要 Pen1 在定义为 **HistPrv1** 的远程历史供应器中绘制标记名 **Boiler1**，您就可以在 Pen1 的“**标记名**”框中输入 **HistPrv1.Boiler1**。

     注意：`.TagID` 标记名点域不能用于远程历史供应器标记名引用中。

使用标记浏览器访问远程历史供应器

下面的步骤介绍应用程序开发人员可以如何使用标记浏览器来选择远程标记名引用。

➢ **要将远程历史供应器定义为标记源：**

1. 创建一个访问名，指定历史供应器所在的“**节点名**”。

   访问名中指定的“**节点名**”不一定是标记名所在节点的实际名称。但是，您必须创建访问名，否则不能将远程历史供应器定义为标记源。

2. 双击历史趋势。“**历史趋势配置**”对话框出现。

3. 双击笔的“**标记名**”输入框。标记浏览器出现。

4. 单击“定义标记源”按钮 ，以将远程历史供应器定义为标记源。

5. 单击“**标记源**”箭头，从表中选择新的远程历史供应器，或单击树形视图按钮并在树形视窗格中选择标记源。标记浏览器随后即重新显示选定的远程历史供应器的标记名。

6. 双击您要指定给历史笔的标记名，或者选择该标记名，然后单击“**确定**”。

7. “**历史趋势配置**”对话框重新出现，并在笔的“**标记名**”框中按以下格式显示选定的标记名： **AccessName:Item**.

8. 将“**AccessName:**”部分替换为您在分布式名称管理器中定义的历史供应器名。例如，**HistPrv1.Tagname**。

   此过程看似麻烦，但是，一旦在标记浏览器中将历史供应器定义为标记源，以后您每次双击另一标记名输入框时，只需在标记浏览器中双击标记名，然后将“**AccessName:**”部分替换为历史供应器名就可以了。在指定远程历史供应器的标记名时，此过程可以减少出错机会。

注意：在 WindowViewer 中，如果允许在运行时更改历史趋势，则当用户单击笔按钮来更改标记名时，标记浏览器会出现，但是只能访问本地应用程序的标记名。

有关使用标记浏览器的详细信息，请参阅第 4 章“标记名字典”。

动态配置远程历史供应器

在运行时，您同样可以动态配置历史趋势的远程历史供应器，为此，您要创建一个 QuickScript 来指定 **HTSetPenName** 函数中的远程历史供应器标记名引用。例如：

```
HTSetPenName("HistTrendTag", 1, "HistPrv1.Boiler1");
```

此处，1 是指用于绘制指定的远程历史供应器标记名的笔。

注意：运行时远程历史供应器不支持“历史趋势配置”对话框和 .Pen。

有关 InTouch QuickScript 函数的详细信息，请参阅您的联机 *《InTouch 参考指南》*。

配置分布式历史记录

➢ **要配置分布式历史记录：**

1. 在“**特别**”菜单上，指向“**配置**”，然后单击“**历史记录**”。“**历史记录属性**”对话框出现：

   - 要快速访问此对话框，在应用程序浏览器的“**配置**”下，双击“**历史记录**”。



   - 如果您右击任意历史趋势配置对话框中的文本框，一个菜单会出现，显示可应用于选定文本的命令。

2. 选择“**允许历史记录**”以启用全局标记名记录。
3. 选择“**在指定目录中存储记录文件**”，然后在输入框中，输入存储记录文件的位置路径。您必须输入有效的通用命名惯例 (UNC) 路径。例如“\\Node\Share\Path”。

   如果使用 NAD，确保路径指向与应用程序目录不同的目录。
4. 在“**登录节点名**”框中，输入要记入历史记录文件的节点名。

   此设置只允许此处输入的节点名记入历史文件。
5. 单击“**确定**”。

注意：在将选定了“**允许历史记录**”选项的应用程序分发到 WindowViewer 节点时，该节点会检查此选项以决定是否应该记录。如果选定了“**允许历史记录**”，则可能的设置为：
域等于节点名 – 允许记录
域不等于节点名 – 不允许记录

创建历史趋势指示器

指示器是时间刻度上的位置指示，可改变位置来获得某个精确时间点的特定数据。通过在指示器**点域**中输入游标对象，您可以在历史趋势显示图上滑动并访问所需的数据区段。您可以使用 QuickScript 函数来访问指定的指示器位置的平均、最小和最大值。通过创建左右指示器并使用附加的 InTouch QuickScript 函数，可以根据指示器之间或指示器位置的数据分析返回值。分析类型包括平均、最小、最大、最小/最大值、最大/最大工程单位及标准偏差。您也可以在两个指示器之间执行缩放操作。

有关指示器点域的详细信息，请参阅“历史趋势点域”。

此外，您还可以增加根据图表上的已知位置来显示数据的功能，以及对趋势图进行缩放操作。以下几页介绍如何使用链接和表达式，以在历史趋势中加入这些功能。

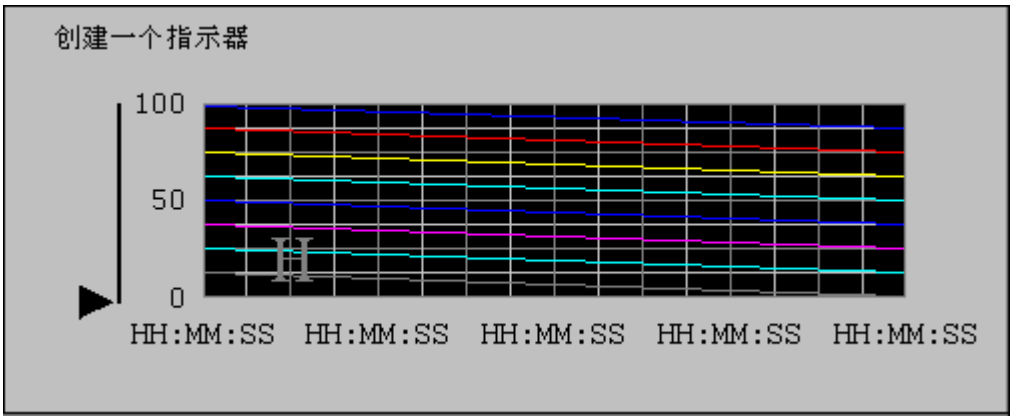
这些功能已经在历史趋势向导中配置。

有关详细信息，请参阅“使用历史趋势向导”。

您可以使用“**水平游标**”链接来创建一个可在历史趋势上移动的指示器（每个趋势支持左和右两个指示器）。

➢ **要创建指示器：**

1. 创建用作指示器的对象。在下面的例子中，我们将使用多边形和垂直线符号。



2. 要正确定义指示器的水平游标链接，您需要知道图表的宽度。要确定图表的宽度，从其一端到另一端画一条水平线，注意 WindowMaker 状态栏中显示的值。观察从左边起的第三个数值，这就是您的图表宽度。记下此值。删除该测量线。

3. 双击指示器对象。动画链接选择对话框出现。

4. 在“**触动链接 - 游标**”区域中，单击“**水平**”。“**水平游标**”对话框出现：



5. 在“**标记名**”框中，输入历史趋势名加上 **.field** 和 **.ScooterPosLeft**。

   例如，如果趋势包含标记名 **Trend1**，则标记名为 **Trend1.ScooterPosLeft**。“**在左端**”的值为 0.0；“**在右端**”的值为 1.0。“**水平移动向左**”的值为 0；“**向右**”的值为从趋势一端画水平线到另一端时的象素值（上例中，图表为 250 个象素宽）。

6. 单击“**确定**”。

将左指示器移至趋势的最左端，这样就完成了左指示器。使用 **.ScooterPosRight** 重复此过程以绘制右指示器。值域保持不变。

显示指示器位置的数值

要根据指示器的当前位置来显示值，您需要创建一个数字文本对象，如 #.00 并使用表达式将其指定给“**值显示-模拟**”链接。例如：

```
HTGetValueAtScooter( "Trend1", Trend1.UpdateCount, 1,
Trend1.ScooterPosLeft, 1, "PenValue" )
```

此实例在 Trend1 左边的指示器位置检索笔 1 的值。如果游标在运行时移动，“**值显示 - 模拟**”链接将跟随新指示器位置值的变化而自动更新。

检索区域间的数值

要取得指示器当前位置之间的数据，无论是最大值、最小值、平均值或标准偏差，您都需要建立一个数字文本对象（如 #.00），并使用表达式将它指定给“**值显示-模拟**”链接。例如：

```
HTGetValueAtZone( "Trend1", Trend1.UpdateCount,
Trend1.ScooterPosLeft, Trend1.ScooterPosRight, 1,
"PenMaxValue" )
```

此实例检索 Trend1 左指示器位置与右指示器位置之间 1 号笔的最大值。

缩放

要在趋势上放大，您需要创建一个按钮并将其指定给“**触动按钮 – 动作**”链接。选择“**按下时**”条件，输入下面的 QuickScript：

```
HTZoomIn( "Trend1", "Center" );
```

此实例以趋势的中央时间为基准，当指示器位于最左和最右位置时，使新图表的宽度等于旧图表宽度除以 2。如果指示器移动，新图表的宽度为左右指示器之间的时间，而不使用 ***LockString***（“中心”）。

有关使用历史 QuickScrip 函数的详细信息，请参阅“历史 QuickScript 函数”。

历史趋势和夏令时

InTouch 的历史记录功能会根据计算机时钟，在每天的午夜创建一个新的历史记录文件。当选定了历史记录选项的标记名值变化超过记录死区值时，该值将写入历史记录文件。历史记录文件中的每条记录都包含标记名的最新值，以及数值的更新时间和日期。InTouch 历史引擎使用世界调整时间 (UTC)，也称为格林威治平均时间 (GMT) 作为记录和检索数据的时间。下面的例子介绍历史引擎如何处理时区和夏令时问题：

假定有一个标记名 (**tag1**) 在 GMT 西部时间上午 10:00:00 时（计算机时间）发生值变化：

Logged_TimeStamp=Computer_Time+(Offset -DST)

此处，**Logged_TimeStamp** 是在记录文件中输入的记录的时间标签。

例 1：
时区设为太平洋标准时间 (GMT - 8) 并启用夏令时：
Logged_TimeStamp=10:00:00 +(8-1)=17:00:00

例 2：
时区设为太平洋标准时间 (GMT -8) 并禁用夏令时：
Logged_TimeStamp=10:00:00 +(8-0)=18:00:00

例 3：
时区设为阿姆斯特丹、柏林、伯尔尼、罗马时间 (GMT +1) 并启用夏令时：
Logged_TimeStamp=10:00:00 +(-1+1)=10:00:00

例 4：
时区设为蒙罗维亚、卡萨布兰卡时间 (GMT0) 并禁用夏令时：
Logged_TimeStamp=10:00:00 +(-0 +0)=10:00:00

在例 4 中，记录文件中的 **TimeStamp** 与计算机时间相同（这是理想设置）。

历史趋势也会考虑进行历史数据检索的计算机的时区和夏令时设置。如果记录计算机与检索计算机之间时区设置不同，在 10:00:00 时（记录节点时间）写下的记录可能会在检索节点上显示不同的时间。在大多数情况下，如果在相同的计算机（或具有相同时区设置的计算机）上执行记录和检索，则所检索的记录是正确的。

➢ **要在 Windows 控制面板中设置时区：**

1. 打开 Windows 控制面板。
2. 双击“**日期/时间**”图标，或者双击 Windows **任务栏**上的时钟。“**日期/时间属性**”对话框出现：



3. 单击“**时区**”标签，然后单击箭头以打开时区列表。
4. 从列表选择您的时区。
5. 取消选定“**根据夏令时自动调整时钟**”。
6. 单击“**确定**”。

注意：Windows 2000 和 Windows NT 操作系统可能设置为根据夏令时自动调整时钟。建议您禁用此功能，因为它会在夏令时切换时导致报告错误的数值。要禁用此功能，使用 Windows 控制面板上的日期/时间工具，或者双击 Windows 任务栏上的时钟。

有关系统时间的详细信息，请参阅第 3 章“建立分布式应用程序”。

自动更改系统时间

Windows 2000 和 Windows NT 操作系统会试图根据夏令时自动调整时钟。您可以通过 Windows 控制面板上的日期/时间工具关闭此功能，然后手动调整时间，也可以使用 InTouch QuickScript 在以下时间自动设置时钟。

- **要在春季向前调拨 Windows 2000 和 Windows NT 的时钟：**

创建下面的条件 QuickScript：

```
$Year == yyyy and $Month == 04 and $Day == dd and
$Hour == 02 and DaylightSavingsTime == 0 ;
```

此处：

yyyy = 年份（如 1993、1994 或 1995 ...）
dd = 时间变化的日期
DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。其初始值为 0。

ON TRUE:

```
DaylightSavingsTime = 1;
StartApp "c:\Winnt\System32\control.exe Date/Time";
{您可能需要调整路径来反映文件在系统中的位置}
SendKeys "%(t)" ;
SendKeys "%(03)" ;
SendKeys "%{TAB}" ;
SendKeys "%{TAB}" ;
SendKeys "%{TAB}" ;
SendKeys "% {ENTER}"
Flag=O;
```

注意：取决于计算机系统的速度，由第二个脚本延迟的时间（为真时）可以增大或减小。此延迟时间允许在发送键之前启动 CONTROL.EXE。

- **要在春季使用命令行向前调拨 Windows 95（或以上版本）的时钟：**

  创建下面的条件 QuickScript：

  ```
  $Year==yyyy and $Month == 04 and $Day == dd and $Hour ==02 and
  DaylightSavingsTime == 0;
  ```

  此处：

  yyyy = 年份（如 1993、1994 或 1995 ...）
  dd = 时间变化的日期
  DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。其初始值为 0。

  ON TRUE:
  ```
  DaylightSavingsTime = 1;
  StartApp "c:\Command.com /c time 03:00";
  ```

- **要在春季使用 WWDosCommand() 函数向前调拨 Windows NT 的时钟：**

  创建下面的条件 QuickScript：

  ```
  $Year==yyyy and $Month == 04 and $Day == dd and $Hour ==02 and
  DaylightSavingsTime == 0;
  ```

  此处：
  yyyy = 年份（如 1993、1994 或 1995 ...）
  dd = 时间变化的日期
  DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。其初始值为 0。

  ON TRUE:
  ```
  DaylightSavingsTime = 1;
  WWDosCommand ("time 03:00", "Invisible";
  ```

- **要在秋季回拨 Windows 2000 或 Windows NT 操作系统的时钟：**

  创建下面的条件 QuickScript：

```
$Year == yyyy and $Month == 10 and $Day == dd and
$Hour == 02 and DaylightSavingsTime == 0 ;
```

  此处：
  yyyy = 年份（如 1993、1994 或 1995 ...）
  dd = 时间变化的日期
  DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。其初始值为 0。

注意：每当时钟回拨时，历史记录引擎可能会改写历史记录文件中的现有数据。为防止由此造成的数据丢失，请在回拨时钟之前备份您的记录文件。

ON TRUE:

```
DaylightSavingsTime = 1;
StartApp "c:\windows\control.exe Date/Time" ;
```

注意：您可能需要更改路径来反映 CONTROL.EXE 在计算机系统中的位置。

Flag 是一个用户自定义的内存整型标记名，其初始值为 0。

```
Flag==1;
```

WHILE TRUE EVERY 1000 MSEC:

```
SendKeys "%(t)" ;
SendKeys "%(01)" ;
SendKeys "01" ;
SendKeys "%{TAB}" ;
SendKeys "%{TAB}" ;
SendKeys "%{TAB}" ;
SendKeys ″% {ENTER}″
Flag=1;

Flag==1;
```

注意：取决于计算机系统的速度，由第二个脚本延迟的时间（为真时）可以增大或减小。此延迟时间允许在发送键之前启动 CONTROL.EXE。

- **要在秋季使用命令行回拨 Windows 95 的时钟：**

  创建下面的条件 QuickScript：

  ```
  $Year==yyyy and $Month == 04 and $Day == dd and $Hour ==02 and
  DaylightSavingsTime == 0;
  ```

  此处：
  yyyy = 年份（如 1993、1994 或 1995 ...）
  dd = 时间变化的日期
  DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。其初始值为 0。

  ON TRUE:
  ```
  DaylightSavingsTime = 1;
  StartApp "c:\Command.com /c time 01:00";
  ```

- **要在秋季使用 WWDosCommand() 函数回拨 Windows NT 的时钟：**

  创建下面的条件 QuickScript：

  ```
  $Year==yyyy and $Month == 04 and $Day == dd and $Hour ==02 and
  DaylightSavingsTime == 0;
  ```

  此处：
  yyyy = 年份（如 1993、1994 或 1995 ...）
  dd = 时间变化的日期
  DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。其初始值为 0。

  ON TRUE:
  ```
  DaylightSavingsTime = 1;
  WWDosCommand ("time 01:00", "Invisible";
  ```

注意：每当系统时钟回拨时，历史记录引擎可能会改写历史记录文件中的现有数据。为防止由此造成的数据丢失，请在回拨时钟之前备份您的记录文件。

- **要在秋季回拨时钟同时防止数据丢失：**

  创建下面的条件 QuickScript：

  ```
  $Year==yyyy and $Month == 04 and $Day == dd and $Hour ==02 and
  DaylightSavingsTime == 0;
  ```

  此处：
  yyyy = 年份（如 1993、1994 或 1995 ...）
  dd = 时间变化的日期
  DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。其初始值为 0。

  ON TRUE:
  ```
  DaylightSavingsTime = 1;
  $Historicallogging=0;
  ```

  此脚本将停止历史记录。

  ```
  StartApp "C:\Winnt\system32\control.exe date/time
  ```

  注意：您可能需要更改路径来反映 CONTROL.EXE 在计算机系统中的位置。

  Flag 是一个用户自定义的内存整型标记名，其初始值为 0。

  ```
  Flag==1;
  ```

  WHILE TRUE EVERY 1000 MSEC:
  ```
  SendKeys "%(t)";
  SendKeys "01";
  SendKeys "%{TAB}" ;
  SendKeys "%{TAB}" ;
  SendKeys "%{TAB}" ;
  SendKeys "% {ENTER}"
  FileMove("c:\history\9910dd00.igh", "c:\backup","monitor");
  Flag=2;
  ```

  ```
  Flag==2
  ```

  WHILE TRUE EVERY 5000 MSEC:
  ```
  FileMove("c:\history\9910dd00.idx", "c:\backup", "monitor");
  Flag=3;
  ```

  ```
  Flag==3
  ```

  WHILE TRUE EVERY 5000 MSEC:
  ```
  $Historicallogging=1;
  Flag=0;
  ```

  此处：
  history 是记录文件所在的目录。
  backup 是记录文件要移动到的目录。
  9910DD.IGH 和 9910DD00.IDX 是要移动的记录文件。
  'DD' 将替换为当前日期。

  取决于记录文件的大小和计算机系统的速度，条件 QuickScript 中的 5000 毫秒延迟时间可以增大或减小。

HistData 实用程序

使有 HistData 实用程序，可以对由 InTouch 创建的历史数据文件进行 DDE（动态数据交换）访问，并将选定的历史数据移动到请求程序，例如 Microsoft Excel。HistData 使您可以即时查看历史数据或创建文件以便日后访问。访问历史数据可以通过请求程序中的宏函数或从 InTouch 中实现。

注意：HistData 程序应先于使用它的任何程序启动（然后缩小为一个图标）。

HistData 程序不支持远程标记名引用。

HistData 数据库

HistData 程序包含自己的内部数据库。内部数据库中的项目用于指定要访问的历史数据的起始周期、持续时间和采样间隔等。下表列出了在 HistData 程序中定义的项目：

项目	类型	描述
DATADIR	消息型	包含历史数据文件的目录的路径名，例如 C:\InTouch\App。
DBDIR	消息型	包含 InTouch 标记名字典的目录的路径名，例如 C:\InTouch\App。
STARTDATE	消息型	MM/DD/YY 格式的数据采样开始日期。
STARTTIME	消息型	使用 24 小时时钟的 HH:MM:SS 格式的数据采样开始时间。
DURATION	消息型	数据返回的时间长度。**DURATION** 可以用星期、天、时、分、秒等来表示。下面是有效的字符： **w**（星期）、**d**（天）、**h**（小时）、**m**（分）、**s**（秒）。允许使用小数值。 例如，扩 .5s 表示 500 毫秒。若只需一个样本，请将 **DURATION** 设为 0（零）。

项目	类型	描述
INTERVAL	消息型	采样间隔的时间长度。**INTERVAL** 可以用星期、日、小时、分和秒来表示，例如 1W 表示一星期。允许使用小数值，例如用 .25d 来表示 6 小时 （有效字符与 **DURATION** 相同）。 **注意：DURATION** 和 **INTERVAL** 允许的最大时间长度均为 6 周。这适用于所有请求类型、日、秒等。例如，如果使用日，则最大值为 42（7 天 x 6 周 = 42）。
TAGS	消息型	返回历史数据的标记名列表。**TAGS** 以“TagA,TagB,TagZ”格式输入。此外，采样日期和（或）时间可以通过使用内部系统标记名 **$Date** 和 **$Time** 来请求。例如： "$Date,TagA,TagB" 或 "$Time,TagA,TagB" 或 "$Date,$Time,TagA,TagB"
TAGS1, TAGS2,....	消息型	**TAGS** 字符串在 WindowViewer 中的长度可以为 131 个字符，在 Excel 中的长度可以为 255 个字符。字符串可以通过迭加标记名项目“Tags1”、“Tags2”等来延长，以适应更长字符串的要求。如果标记名需要附加额外的标记名文本，请在字符串末输入一个加号 (+)。例如： TAGS="$Date,ProdLevel,ProdTemp,+" TAGS1="ReactLevel,Temp,GasLevel,+" TAGS2="MotorStatus" **注意：**不允许使用重复标记名，每个标记字符串的最大长度为 512 字节。

项目	类型	描述
PRINTTAGNAMES	离散型	此项目缺省为 1，使 HistData 在输出文件的第一行及关联值列的上方打印标记名。如果不打印标记名，则此项目值必须须为 0（零）。
DATA	消息型	此项目用于以逗号分隔变量的格式保存 HistData 程序中的请求数据。它由通过 DDE 来“**提示**”或“**请求**”数据的其它应用程序使用。
SENDDATA	整型	当设置为 1 时，HistData 会用请求数据来更新 **DATA** 项目。更新完成后，**SENDDATA** 自动重置为 0（零）。 **注意：**如果您在使用 **SENDDATA** 时收到一则错误消息，告诉您请求的数据太多，则须缩短 **DURATION** 或减少请求的标记名数。
FILENAME	消息型	输入用于写入请求数据的文件的路径名，例如 C:\INTOUCH\HDFILE.CSV。
WRITEFILE	整型	当设置为 1 时，HistData 会将请求数据写入由 **FILENAME** 项目名指定的文件。当文件更新完成后，**WRITEFILE** 会自动重置为 0（零）。
STATUS	离散型	显示最后一次操作的状态。1 表示成功，0（零）表示出错。
ERROR	消息型	包含上次错误说明的字符串。如 **STATUS** 为 1，则显示“无”；如 **STATUS** 为 0（零），则显示错误消息字符串。

在 InTouch 中使用 HistData

在本节，我们创建一个示例窗口，向您演示在 InTouch 中读取和显示历史数据的方法。

为了使 InTouch 能够从 HistData 程序请求数据，我们定义下面的访问名：



访问名可以是长达 32 个字符的任意名称。**主题名**也可以是长达 32 个字符的任意名称。建议访问名和主题名使用相同的名称。**应用程序名**必须是程序名 **HistData**（不带 .exe）。

此外，建议每次使用 HistData 时选定“**提示所有项**”选项。

有关访问名的详细信息，请参阅第 9 章“I/O 通讯”。

在定义访问名后，为每个 HistData 内部数据库项目创建下面的 I/O 型标记名：

标记名	I/O 标记类型	标记名	I/O 标记类型
DATA	消息型	**SENDDATA**	整型
DATADIR	消息型	**STARTDATE**	消息型
DBDIR	消息型	**STARTTIME**	消息型
DURATION	消息型	**STATUS**	离散型
ERROR	消息型	**TAGS**	消息型
FILENAME	消息型	**WRITEFILE**	整型
INTERVAL	消息型	**PRINTTAGNAMES**	离散型

如果您要将数据不仅发送至 .CSV 文件，而且发送至名叫 **Data** 的项目，以便可以从其它应用程序访问它，请创建下面的两个附加标记名：

标记名	I/O 标记类型	访问名	项目
HDWSendData	离散型	Viewstream1	**SendData**
HDWData	消息型	Viewstream1	**Data**

注意：如果您在创建并使用这些标记名后请求大量信息，您会收到一则错误消息：请求的数据太多 – 请缩短持续时间或减少标记数。

当您使用 HistData 向导时，向导会自动创建 HistData 标记名和访问名。要使用 HistData 向导：

1. 单击向导工具。
2. 打开向导的趋势组。
3. 双击 HistData 向导并将其粘贴到窗口中。
4. 双击 HistData 向导并对其指定 Hist Tend 型标记名。
5. 单击“**确定**”以创建 HistData 标记名。

在创建 I/O 型标记名后，创建一个名叫 **HistData** 的新窗口，如下例所示：



号链接至输入链接。例如，# 号与 **HDWDataDir** 建立“用户输入/字符串”链接。用户输入链接允许您在运行时更改标记名的值。

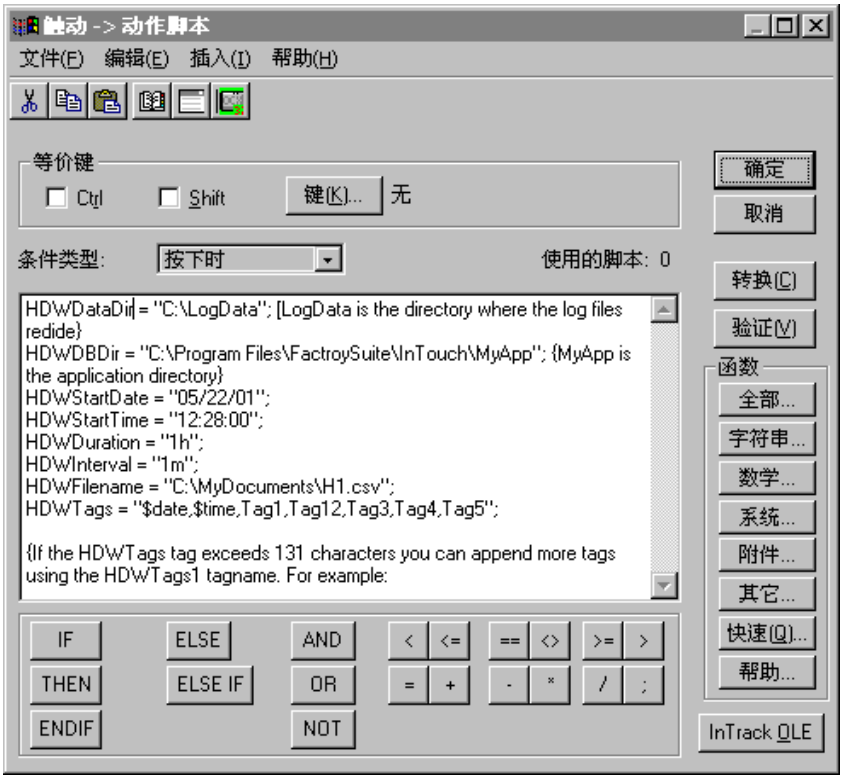
“**Status**”按钮通过标记名 **HDWStatus** 链接至“填充颜色 – 离散数表达式”：



“**Write File**”按钮通过标记名 **HDWWriteFile** 链接至“填充颜色 – 离散数表达式”：



“**Initialize Data**”按钮链接至“触动按钮 – 动作脚本”。



当按下“**Initialize Data**”按钮时，HistData 项目即被初始化为所需的值。如有必要，您也可以通过使用“用户输入链接”在运行时更改这些值。

“**Write File**”按钮链接至“触动按钮 – 动作脚本”。



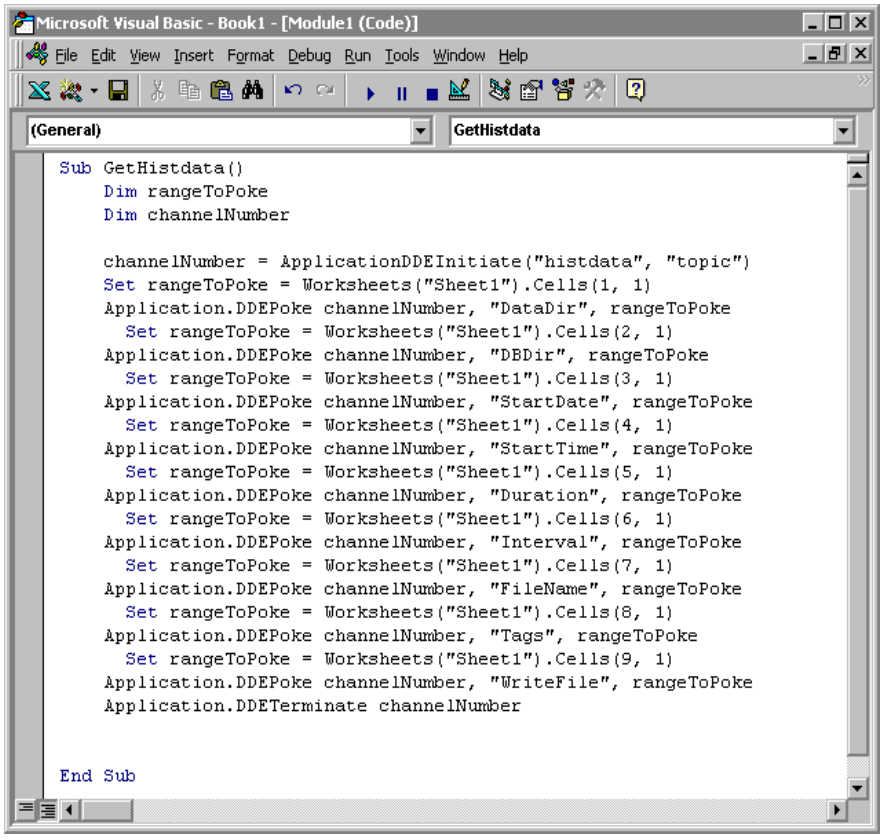
单击“**WriteFile**”按钮可生成 .CSV 文件。

在配置并保存所有上述项目后，启动 **HistData** 并将其最小化。下一步，启动 WindowViewer 并打开 HistData 窗口。单击“**Initialize**”按钮并在必要时修改 HistData 项目，然后单击“**WriteFile**”按钮。如果操作成功，“**Status**”的值将变为“**开**”，并显示与“**开**”状态关联的颜色。如果操作不成功，“**Status**”的值将变为“**关**”，并显示**错误消息**来指明出错原因。

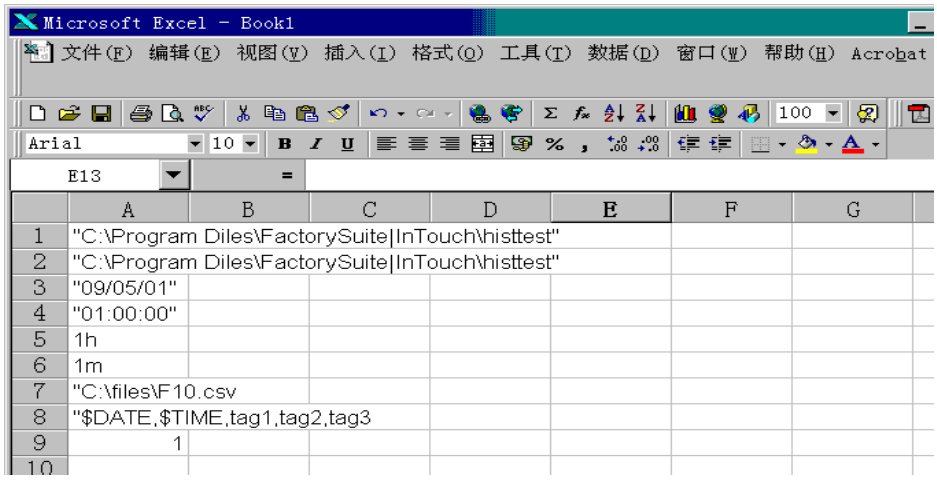
在 Excel 中使用 HistData

HistData 程序会对 Microsoft Excel 等产品的 **INITIATE**、**POKE** 和 **TERMINATE** 函数作出响应。程序使用带 **keyword**（内部数据库项目）和的 **POKE** 函数来设置用于定义查询的参数。一旦正确设置查询，程序会运行宏来请求选定的历史数据。

您可以编写 Excel 宏来与 HistData 互相操作。下面是用 Excel 5.0VBA（适用于 Excel 7.0 或 Excel 97）编写的宏的一个实例：



插入的数据包含在一个名为 **Sheet1** 的 Excel 电子表格的单元格中。下面是包含要插入的数据的 Sheet1 的一个实例：



常见 HistData 错误消息

错误消息	原因/解决方法
请求的数据太多 – 请缩短持续时间或减少标记名数目。	当使用 SendData 项目并且所请求的数据过多时，会出现此错误消息。如果您只想创建一个包含加密记录文件数据的 .CSV 文件，***请勿使用 SendData***。
无法打开文件 C:\FILES1\HISTDATA.CSV	目录 **Files1** 不存在；此外，检查路径拼写是否正确。
无法打开文件 C:\FILES	未定义 CSV 文件。
DATADIR 项目无效	DataDir 项目的指定目录不存在。请检查拼写是否正确。
STARTDATE 项目无效	发送至 StartDate 项目的格式无效。
未找到记录文件	DataDir 项目的指定路径中没有请求日期的记录文件。
数据库中找不到标记名 TAGX	请求的标记名（本例中为 **TAGX**）不存在于标记名字典中。请检查标记名的拼写是否正确。
C:\IT6.0B\HISTEST 中找不到 tagname.x	文件 **tagname.x** 不存在或者损坏。

第 9 章

I/O 通讯

InTouch 利用 Microsoft 动态数据交换 (DDE)、FastDDE、NetDDE 和 Wonderware SuiteLink 协议来与其它 Windows 程序、Wonderware I/O 服务器及第三方 I/O 服务器程序与现实世界进行通讯。

目录

- 支持的通讯协议
- Wonderware SuiteLink
- InTouch I/O 地址命名惯例
- InTouch I/O 地址
- InTouch 访问名
- 在 InTouch 中定义 I/O 项目
- 监视 I/O 对话的状态
- 监视 I/O 服务器通讯状态
- 监视多个输入设备状态

支持的通讯协议

动态数据交换 (DDE) 是一个由 Microsoft 开发的通讯协议，该协议允许 Windows 环境下的应用程序彼此发送/接收数据和指令。它在两个同时运行的应用程序之间实现客户端与服务器的关系。*服务器*应用程序提供数据并接受对这些数据感兴趣的其它应用程序的请求。发出请求的应用程序叫做*客户端*。某些应用程序（例如 InTouch 和 Microsoft Excel）可以同时作为*客户端*和*服务器*。

FastDDE 提供了一种方法，可以将许多 Wonderware DDE 专用消息压缩成单个 Microsoft DDE 消息。这种压缩由于减少了*客户端* 和 *服务器*之间所需的 DDE 交易总数，从而提高了效率和性能。虽然 Wonderware 的 FastDDE 已经将 DDE 应用扩展到各个行业，但这种扩展正在到达其分布式环境下的性能极限。

NetDDE 扩展展了标准 Windows DDE 的功能，以包括局域网之间和通过串行端口的通讯。网络扩展允许那些通过网络或调制解调器连接的运行于不同计算机上的应用程序进行 DDE 链接。例如，NetDDE 支持运行于 IBM PC 上的应用程序（通过 LAN 或调制解调器连接），与 VMS 和 UNIX 等操作环境下的非 PC 平台 DDE 应用程序之间的动态数据交换。

Wonderware SuiteLink 使用基于 TCP/IP 的协议。SuiteLink 专门设计用于满足象数据完整性、高吞吐量以及方便诊断这样的工业需要。Microsoft Windows NT 和 Windows 2000 均支持这种通讯标准。

SuiteLink 不是 DDE、FastDDE 或 NetDDE 的替代。客户端与服务器之间的每次连接取决于您的网络状况。SuiteLink 是专门为高速工业应用而设计的，它具有下列特点：

数值时间质量 (VTQ) 在所有传送给识别 VTQ 的客户机的数据值上均会插入一个时间标签和质量指示器。

对数据吞吐量、服务器加载、计算机资源消耗以及网络传输的广泛诊断可以通过 Microsoft Windows NT 操作系统性能监视器来实现。这个特点对于规划和维护分布式工业网络是至关重要的。

应用程序之间可维持一致的高数据容量，而不管这些应用程序是位于单一节点上还是分布在多个节点上。

网络传输协议是使用 Microsoft 标准 Winsock 接口的 TCP/IP。

Wonderware SuiteLink

Wonderware SuiteLink 使用基于 TCP/IP 的协议。SuiteLink 专门设计用于满足象数据完整性、高吞吐量以及方便诊断这样的工业需要。Microsoft Windows NT 和 Windows 2000 均支持这种通讯标准。

➢ **要使用 SuiteLink 通讯协议：**

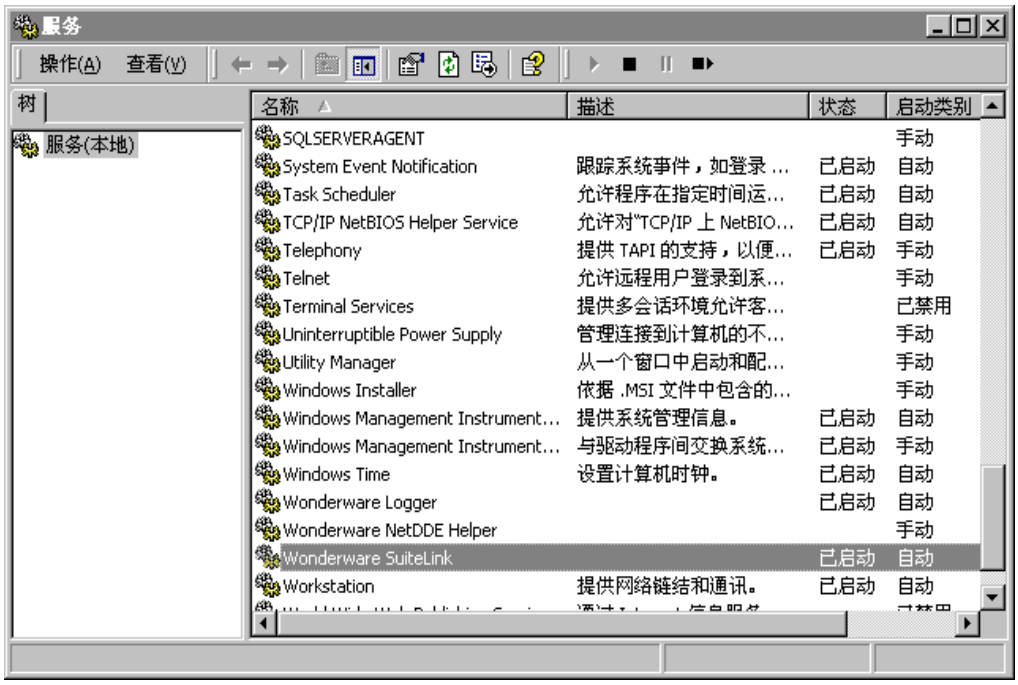
1. Microsoft TCP/IP 必须已配置并能正常工作。
2. 必须使用不超过 15 个字符的计算机名（**节点名**）。

   有关安装和配置 Microsoft TCP/IP 的详细信息，请参阅 Microsoft Windows 操作系统文档。
3. Wonderware SuiteLink 必须作为一个服务程序运行。如果由于某些原因导致 SuiteLink 停止工作，你需要再次将其启动（安装 InTouch 时会自动安装 SuiteLink，并且它在缺省条件下的配置是作为 NT 服务自动启动）。

   有关 Windows NT 服务的详细信息，请参阅附录 A“InTouch Windows NT 服务概览”。

➢ **要将 SuiteLink 作为 NT 服务启动：**

1. 打开 Windows 控制面板。
2. 双击“**服务**”。“**服务**”对话框出现：



3. 选择“**Wonderware SuiteLink**”，然后单击“**启动**”。
4. 单击“**关闭**”。

InTouch I/O 地址命名惯例

InTouch 使用由三部分组成的命名惯例来识别 I/O 服务器程序中的数据元素，这三个部分是*应用程序名*、*主题名*和*项目名*。要从另一应用程序取得数据，*客户端*程序 (InTouch) 需要通过指定上述三项来打开通往*服务器*程序的通道。

要使 InTouch 能够从其它程序取得数据值，它还必须知道提供数据值的*应用程序*名，应用程序中包含数据值的*主题*名以及*主题*中的特定*项目*名。此外，InTouch 需要知道数据类型：离散型、整型、实型（浮点）或消息型（字符串）。在 InTouch 数据库中定义标记名时，这些信息决定标记名的 I/O 类型。现在，当 WindowViewer 运行时，它会自动执行所需的所有动作来取得和保存该*项目*的值。

例如：在 Excel 中，*应用程序名*是 “Excel”，*主题名*是包含数据的指定工作表，*项目名*是工作表中将要读出/写入数据的单元格。

InTouch I/O 地址

当另一 Windows 应用程序从 InTouch 中请求数据值时，它也必须知道这三个 I/O 地址项。下面简要介绍 InTouch 的 I/O 地址命名惯例：

VIEW（*应用程序名*）是包含指定数据元素的 InTouch 运行时程序。

TAGNAME（*主题名*）是每次在 InTouch 数据库中读写标记名时使用的名称。

ActualTagname（*项目名*）是在 InTouch 标记名字典中为项目定义的实际标记名。

例如，要从 Excel 访问 InTouch 中的一个数据值，需要在从中写入数据值的单元格中输入 DDE 远程引用公式：

=VIEW|TAGNAME!'ActualTagname'

注意：如果您使用 Wonderware NetDDE 进行联网，则 I/O 地址的*应用程序名*部分必须以远程节点名开头，并且在前面加两个反斜杠，后面再跟一个反斜杠。例如：

\\NodeName\VIEW|TAGNAME!'*ActualTagname*'

InTouch 访问名

当您创建 I/O 型标记名或远程标记名引用时，必须将其与访问名相关联。访问名包含用于与其它 I/O 数据源通讯的信息，包括节点名、应用程序名和主题名。

➢ **要创建访问名：**

1. 在“**特别**”菜单上，单击“**访问名**”，或者在应用程序浏览器的“**配置**”下，双击“**访问名**”。“**访问名**”对话框出现。
2. 在应用程序浏览器中，您可以右击“**访问名**”，然后单击“**打开**”。您也可以在标记名字典中定义 I/O 型标记名时创建访问名。



3. 单击“**添加**”，“**添加访问名**”对话框出现：



4. 在“**访问名**”框中，输入您要 InTouch 使用的访问名（为简便起见，请使用与*主题名*相同的名称）。

   InTouch 使用访问名来引用实时 I/O 数据。每个访问名相当于一个 I/O 地址，可以包含节点、应用程序和主题。在分布式应用程序中，I/O 引用可以设置为指向网络 I/O 服务器的全局地址，或指向本地 I/O 服务器的本地地址。

5 如果数据位于网络 I/O 服务器中，请在“**节点名**”框中输入远程节点名。

6. 在“**应用程序名**”框中，输入从中获取数据值的 I/O 服务器程序的实际程序名。在本例中，数值来自 Wonderware Modbus I/O 服务器，因此使用 **MODBUS**。**切勿**输入程序名的 **.exe** 扩展名部分。

7. 在“**主题名**”框中，输入你要访问的*主题名*。**主题名**是特定于应用程序的数据元素子组。在数据来自 Wonderware I/O 服务器程序的情况下，*主题名*与 I/O 服务器程序中为*主题*配置的名称**完全**相同。当与 Microsoft Excel 通讯时，*主题名*必须是保存工作表时所指定的名称。例如 Book1.xls。

8. 选择您所使用的协议。

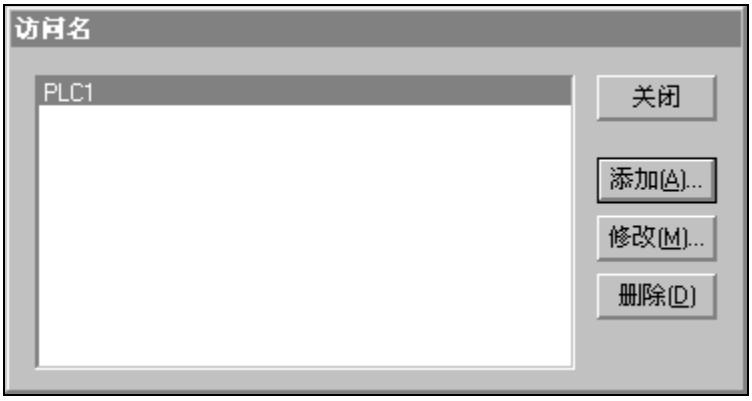
   有关详细信息，请参阅“支持的通讯协议”。

9. 选择您要用于提示服务器的选项：

提示所有项	轮询所有数据，无论其是否在可见窗口中，或者已报警、记录、绘制趋势或用于脚本中。选择此项将影响性能，所以不建议您使用。
只提示激活项	只轮询可见窗口中，以及已报警、记录或用于脚本中的数据点。

注意：系统不会轮询触动按钮动作脚本，除非它显示在可见窗口中。

10. 单击“**确定**”以接受新访问名并关闭对话框。“**访问名**”对话框重新出现，显示列表中选定的新访问名。



11. 单击“**关闭**”，以关闭对话框并返回到标记名定义。

➢ **要修改或删除访问名：**

1. 在“**特别**”菜单上，单击“**访问名**”，或者在应用程序浏览器的“**配置**”下，双击“**访问名**”。“**访问名**”对话框出现。

2. 在应用程序浏览器中，你可以右击“**访问名**”，然后单击“**打开**”。



3. 要更改访问名的定义，从列表中选择访问名，然后单击“**修改**”。“**修改访问名**”对话框出现。进行所需的修改，然后单击“**确定**”。“**访问名**”对话框重新出现。单击“**关闭**”，或者重复此过程以修改其它已定义的访问名。

4. 要删除访问名，从列表中选定后单击“**删除**”。一个消息框会出现，要求您确认删除选定的访问名。单击“**是**”以删除，或者单击“**否**”取消删除。单击“**关闭**”，或者重复此过程以删除其它已定义的访问名，。

注意：不能删除与标记名关联的访问名。

■

在 InTouch 中定义 I/O 项目

在标记名字典中定义 I/O 型标记名时，InTouch 可以接收来自其它本地或远程 Windows 应用程序的数据。每个 I/O 型标记名都引用 I/O 服务器程序中的一个有效*项目*。

有关分布式应用程序的详细信息，请参阅第 3 章“创建分布式应用程序”。

➢ **要定义 I/O 型标记名：**

1. 在“**特别**”菜单上，单击“**标记名字典**”，或者在应用程序浏览器中，双击“**标记名字典**”。“**标记名字典**”对话框出现。
2. 单击“**新建**”，清除“**标记名**”框。

   如果您右击标记名字典对话框中的任意文本输入框，一个菜单会出现，显示可应用于选定文本的命令。



当您第一次访问标记名字典时，会显示内部系统标记名 **$AccessLevel** 的定义。一旦在标记名字典中定义了标记名，当您再次访问它时，将显示最近编辑的标记名定义。

3. 在“**标记名**”框中，输入新标记名的名称。

   标记名最长可为 32 个字符，但必须以一个希腊字符开头（A-Z 或 a-z）。其余的字符可以是 A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ 和 &。

   标记名也是自动编号的。例如，如果您输入并保存标记名 R4001，然后单击“**新建**”，则标记名会自动编号为 R4002。如果标记名包含一个字符分隔的数字，它将由 InTouch 找到的第一个整数自动编号。例如，N7-0 将编号为 N7-1。注意只允许正方向的变化。例如，R4002 到 R4003，R4003 到 R4004 等等。

   标记名不能使用 **RetVal** 一字。这个字是保留字。如果您试图使用这个字，然后尝试编辑 QuickFunction，将出现一条错误消息 "A variable cannot have that name.Tag exists"（变量不能使用该名称，标记已经存在）。

4. 单击“**类型**”，“**选择标记名类型**”对话框出现。



5. 根据下表选择标记名的 I/O 类型：

标记类型	输入/输出值
I/O 离散型	真、开、是 (1) 或假、关、否 (0)
I/O 整型	整数
I/O 实型	浮点小数
I/O 消息型	字符串

6. 在选定 I/O 类型后，单击“**确定**”。选定 I/O 类型的相应“详细资料”对话框会出现。例如，如果您选择 I/O 整型，则会出现下面的对话框：



如果“详细资料”对话框没有出现，单击屏幕顶部的“**详细**”。

7. 指定定义*项目*所需的所有详细资料。

8. 单击“**访问名**”，“**访问名**”对话框出现：



9. 双击列表中您要使用的访问名，或者选定访问名，然后单击“**关闭**”。

   您选定的访问名（现在与该标记名定义相关联）出现在详细资料对话框中的“**访问名**”按钮旁边。例如：



10. 在“**项目**”框中，输入 I/O 服务器程序中数据值的*项目名*。

注意：“标记名”是 InTouch 引用数据值时使用的名称。“**项目**”则是远程 Windows 应用程序引用相同值时使用的名称。这些名称不必相同，但是如有可能，建议您使用相同的名称。此外，如果**项目**是 Excel 中的一个单元格，则必须指定为 Excel 中的定义名，或者是其行/列标识符。例如，R1C1。

---

11. 单击“**关闭**”。

有关定义 I/O 标记名的详细信息，请参阅第 4 章“标记名字典”。

监视 I/O 对话的状态

WindowViewer 支持一个内置的名为 **IOStatus**（InTouch 7.0 以前版本中称为 **DDEStatus**）的*主题名*，它可用于监视特定 I/O 对话的状态。

使用 IOStatus 主题名

我们假定 WindowViewer（视图）正在与一个连接到 PLC 的 Wonderware 仿真 I/O 服务器通讯，该 PLC 在 I/O 服务器的*主题名*定义为 **PLC1**

（“仿真”是一个通用的 Wonderware I/O 服务器，通常用作培训工具。“仿真”包括在 FactorySuite 中）。

➢ **要监视 I/O 通讯的状态：**

1. 在“**特别**”菜单上，单击“**标记名字典**”，或者在应用程序浏览器中，双击“**标记名字典**”。“**标记名字典**”对话框出现。
2. 创建一个 **I/O 离散**型标记名（在本例中，为简便起见，我们将该标记名设定为与要监视的*主题名*相同）。

   在使用 **IOStatus** 来监视 I/O 对话时，必须至少为所监视的访问名定义一个 I/O 型标记名。



3. 单击“**访问名**”，为定义其*主题名*的 **IOStatus** 的访问名定义指定一个标记名。“**访问名**”对话框出现：



注意当前已经存在一个名为 **PLC1**（我们要监视的主题）的访问名定义。要确保访问名正确（其**主题名**为 **PLC1**），单击“**修改**”以查看其定义：



在本例中，因为我们将标记名定义为与*主题名*同名，因此可以轻易地找到包含正确*主题名*的访问名。

4. 单击“**取消**”关闭对话框，并返回到最初的“**访问名定义**”对话框。

5. 单击“**添加**”，“**添加访问名**”对话框出现：



6. 在“**访问名**”框中，输入 **IOStatus**。

   因为我们在 WindowViewer 中监视其状态，所以在“**应用程序名**”框中，输入“View”。

7. 在“**主题名**”框中，输入 InTouch 内部*主题* **IOStatus**。

8. 选择“**只提示激活项**”。

9. 单击“**确定**”关闭对话框。初始“**访问名定义**”对话框重新出现，在列表中显示新的**访问名 IOStatus**：



10. 单击“**关闭**”以关闭对话框，并将此新的**访问名**与 **I/O 离散型**标记名相关联：

■



11. 在“**项目**”框中，输入您要监视的实际**主题名**的**访问名**。在本例中为 **PLC1**。

    因为本例中的标记名与“**主题名**”同名，您可以选择“**使用标记名作为项目名**”，使之自动输入到**项目**框中。

注意：当使用内置的主题 **IOStatus**（InTouch 7.0 以前版本中称为 **DDEStatus**）来监视 I/O 对话时，“**项目**”框总是使用您在“**访问名**”框中输入的名称。

在 Excel 中使用 IOStatus 主题名

通过在工作表单元格的公式中输入相同的信息，您也可以在 Excel 中执行相同类型的监视活动。例如，要监视上述的相同主题，可以输入下面的公式：

=view|IOStatus!'PLC1'

监视 I/O 服务器通讯状态

每个*主题名*都有一个内置的离散*项目* **Status**，可用于监视与 I/O 服务器程序通讯的状态。当与设备的通讯失败时（如电缆断开，PLC 断电等），**Status** 设为“0”；当通讯成功时，**Status** 设为“1”。

注意：当使用“Status”项目来监视主题的状态时，所监视的主题必须至少具有一个活动状态的 I/O 点。

通过定义标记名，并将其与使用“**Status**”作为*项目名*的设备*主题*相关联，您可以从 InTouch 中读取服务器通讯的状态。例如，如果 WindowViewer 使用 Wonderware 仿真 I/O 服务器与 PLC 通讯，则访问名定义将为：



要监视*主题* PLC1 的所有通讯状态，您需要创建下面的标记名定义：



有关 I/O 通讯疑难排解的详细信息，请参阅您的《Wonderware I/O 服务器用户指南》。

在 Excel 中，您可以通过在单元格中输入下面的公式来读取 PLC 通讯的状态：

=SIMULATE|PLC1!'STATUS'

监视多个输入设备状态

本节介绍如何使用多个输入设备来显示对象的状态。

例 1：

在本例中，我们观察使用两个输入端的弹簧复位机动阀的状态。两个输入端代表安装在阀上的一对限位开关。其中一个输入端只有在阀处于打开位置时才接通，而在阀移动或关闭时断开。另一个输入端则是在阀处于关闭位置时接通，而在阀移动或处于打开位置时断开。两个输入端的真值表如下所示：

真值表 1

输入#1 (打开)	输入#2 (关闭)	阀位置	结果
1	0	打开	1 + 0 = **1**
0	1	关闭	0 + 1 = **1**
0	0	InTravel	0 + 0 = **0**
1	1	InValid Position	1 + 1 = **1**

0 = OFF 1 = ON

输入端可以通过将关闭的输入端乘以 2 来进行加权，因此阀位置的结果变为如下：

真值表 2
(输入#2x2)

输入#1 (打开)	输入#2 (关闭)	阀位置	结果
1	0 x 2 = 0	打开	1 + 0 = **1**
0	1 x 2 = 2	关闭	0 + 2 = **2**
0	0 x 2 = 0	InTravel	0 + 0 = **0**
1	1 x 2 = 2	InValid Position	1 + 2 = **3**

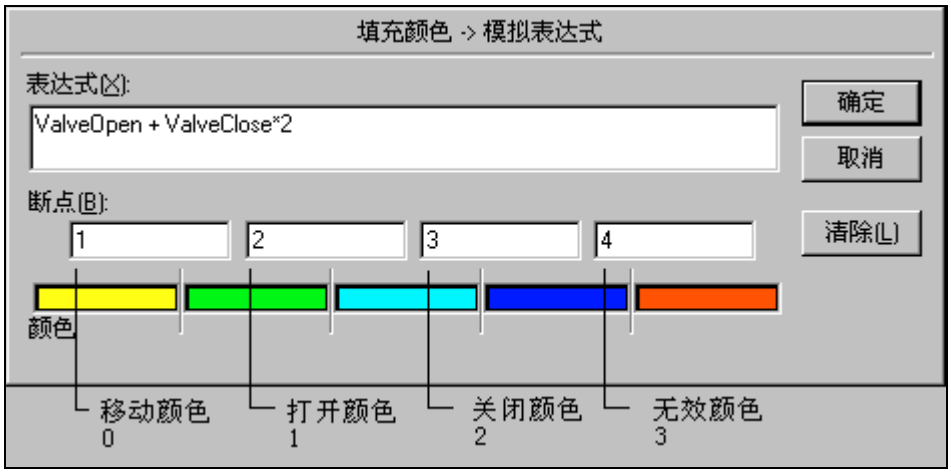
0 = OFF 1 = ON

注意：可以使用无效位置来显示损坏的限位开关。

由于阀位置之间存在显著的数值差别，因此我们可以使用一个“**填充颜色- 模拟**”动画链接来显示阀的状态。

有关创建动画链接的详细信息，请参阅第 5 章“创建动画链接”。

我们现在创建两个 I/O 离散型标记名。一个用于阀打开输入端，一个用于阀关闭输入端。例如，**ValveOpen** 和 **ValveClosed**。然后我们创建一个对象来显示阀的状态。此对象通过下面的属性指定给“**填充颜色- 模拟**”动画链接：



例 2：

在本例中，我们在现有两个输入端的基础上再增加一个输入端。此新的输入端是使阀打开的实际输出端。此新的输入端将在阀打开或正在打开时接通，而在阀关闭或正在关闭时断开。此时，真值表如下所示：

真值表 3

输入#1 (打开)	输入#2 (关闭)	输入#3 (打开)	阀位置	结果
0	0	1	正在打开	0 + 0 + 1 = **1**
1	0	1	打开	1 + 0 + 1 = **2**
0	0	0	正在关闭	0 + 0 + 0 = **0**
0	1	0	关闭	0 + 1 + 0 = **1**
0 = OFF	1 = ON			

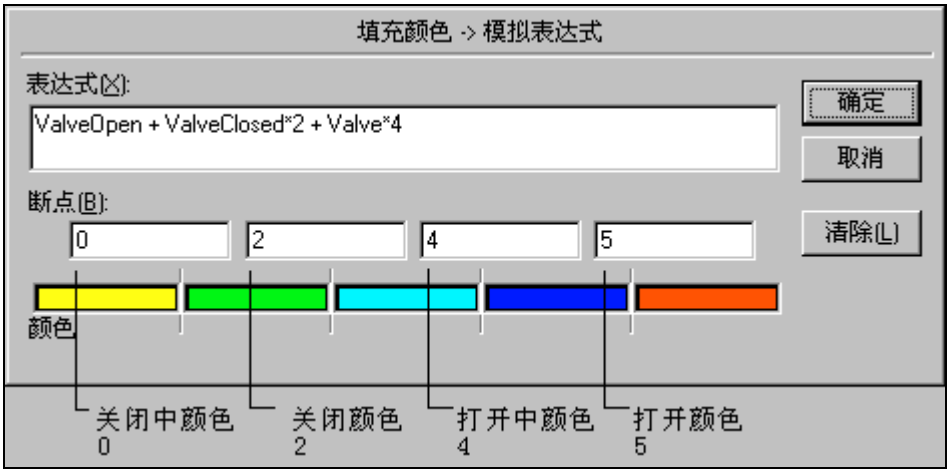
我们再次对输入端加权。如前所述，我们将关闭输入端乘以 2，将新的输入端乘以 4，得出下面的结果：

真值表 4
(输入 #2x2 输入 #3x4)

输入#1 (打开)	输入#2 (关闭)	输入#3 (打开)	阀位置	结果
0	0 x 2 = 0	1 x 4 = 4	正在打开	0 + 0 + 4 = **4**
1	0 x 2 = 0	1 x 4 = 4	打开	1 + 0 + 4 = **5**
0	0 x 2 = 0	0 x 4 = 0	正在关闭	0 + 0 + 0 = **0**
0	1 x 2 = 2	0 x 4 = 0	关闭	0 + 2 + 0 = **2**
0 = OFF	1 = ON			

■

我们再为新的打开输入端创建一个 I/O 离散型标记名 **(Valve)**，并用下面的属性将其指定给“**填充颜色 - 模拟**”动画链接：



通过此方法，我们还可以使用其它的输入端，然后将第 4 个输入端乘以 8，第 5 个输入端乘以 16，以此类推。

第 10 章

Terminal Services for InTouch

本章概述了终端服务的概念及其功能和优点，并假设您已具备安装和使用终端服务的知识。有关详细信息，请参阅《*Terminal Services for InTouch 布署指南*》。

目录

- 简介
- 终端服务器概念
- 终端服务优点
- 终端服务高级客户端
- 已知问题和限制
- 服务器硬件要求
- RDP 客户端的硬件要求
- 规划终端服务安装
- 安装终端服务
- 安装 Terminal Services for InTouch
- Wonderware Logger 控制台和客户端文件
- 建立终端会话

简介

终端服务是包括在 Microsoft Windows 2000 Server 操作系统中的可配置服务，能够从服务器集中运行基于 Windows 的 32 位应用程序。终端服务已完全集成到 Windows 2000 Server 内核中。终端服务仿真程序客户端适用于各种不同的桌面平台（MS-DOS 、Windows、Macintosh、UNIX、Java 等），非 Windows 平台则需要第三方的附加软件，如 Citrix MetaFrame。

与传统的服务器端/客户端环境不同的是，在 Windows 2000 Server 中启用终端服务时，所有应用程序处理均可在服务器上进行。终端服务的客户端不进行任何应用程序的本机处理，而只显示应用程序的输出。终端服务技术只向客户端传送应用程序外观显示 — 图形用户界面 (GUI)。每个用户登录后，仅感觉到自己的会话，这是由服务器操作系统透明管理的，独立于其它的客户端会话。

实施终端服务多用户、以服务器为中心的计算环境的最大优点是系统管理及桌面/系统布署的简化。通过实施以服务器为中心的计算环境，您可以大大缩减总拥有成本以及运行成本。

从应用程序开发的角度来看，终端服务的最大优点之一是那些性能优良的 16 位或 32 位 Windows 应用程序可以照旧运行 — 在终端服务下运行，而不需要进行任何程序更改。但这不是意味着现有的所有应用程序都能在终端服务下完全等效地运行。理解如何设计应用程序，从而充分利用 Windows 2000 终端服务的新功能是很重要的。

FactorySuite 应用程序在后端服务器运行应用程序逻辑，同时使用客户端设备进行数据采集或显示，因而比其它应用程序更适于在以服务器为中心的多用户环境中进行布署。

使用终端服务需要在网络布置的方式和考虑用户的方式上进行一些调整。

有关详细信息，请参阅《Terminal Services for InTouch 布署指南》。

注意：无论是 Microsoft 还是 Citrix，除非用户载入量很小，否则均建议在终端服务器上运行 BackOffice 产品，如 SQL Server。随着载入量增加，后台处理在等待运行时排队，看上去好象处于锁定状态。

采用多用户计算的方法不是做出“非此即彼”的决定 — 多用户计算可极好地并存于分布式系统或基于客户端/服务器的信息构架中。在很多应用程序环境中，需要采用多种环境来使最终用户达到最佳效率和生产力。

Windows 终端服务可安装在任何支持 Windows 2000 Server 的计算机上。另外需要约 14 MB 的磁盘空间来存放客户端安装文件，此外不需要为操作系统准备额外的空间。但是，对于在应用程序服务器模式下使用 Windows 终端服务的机器来说，实际要求可能要高些。由于每个用户都各自用服务器来执行其程序，因而需要准确判断您的用户是如何工作及其实际要求是什么。每套系统都有所不同。

有关详细信息，请参阅“欢迎使用 InTouch”中的“系统要求”部分。

本章旨在向您概括介绍 Windows 2000 Terminal Services 概念。

终端服务器概念

终端服务器又回到了计算系统主机的概念上。所有应用程序处理和逻辑均发生在终端服务器上。终端连接到终端服务器上时，会打开一个会话并利用其部分资源。由于终端服务器提供了全部的资源，客户端的维护和更新工作量大为减少。主要注意力又重新转向集中的服务器上来，因为它归属于大型机。这样就大大简化了维护工作。

服务器端和客户端的通信通过诸如 RPD（远程桌面协议）和 ICA（独立计算结构体系）之类的服务器/客户端软件来实现。每个终端需要一个安装好的 RDP 或 ICA 客户端应用程序，将输入数据（键盘输入、鼠标移动和串行通信）传送到服务器上的通信软件。该信息由服务器处理，而图形显示（屏幕图像）则返回至终端进行显示。

RDP（远程桌面协议）是一种通信协议，其服务器端软件包含在 Windows 2000 Server 的终端服务之中。它包括一个通信公用程序，可以为 16 位和 32 位 Windows 计算机创建 RDP 客户端磁盘，以便通过 TCP/IP 进行通信。

ICA（独立计算机结构体系）是 Citrix 提供的第三方通信协议。它是与 Citrix Device Services 或 Citrix MetaFrame 服务器端软件一起使用的。ICA 具有比 RDP 更大的灵活性，在 Windows 3.11/95/98/NT 和 Windows 2000 的基础上，增加了可用于 Unix、Linux、OS/2、DOS、Windows CE、PDA、Java 和 Macintosh 平台的客户端软件。在 TCP/IP 的基础上，ICA 协议增加了通过 IPX、SPX、NetBIOS 及 RAS 远程访问的功能。ICA 是 Windows 2000 完全支持的协议。

在传统计算机上安装 RDP 或 ICA 协议，就可以使传统计算机变成一台在终端服务器上完成处理并在其屏幕上显示终端服务器桌面的客户机。这是一台“胖客户机”，因为这台计算机具有操作系统和硬盘，但与终端服务器连接并在终端服务器上运行应用程序。胖客户端系统常用于延长老化计算机的寿命。通过载入 RDP 或 ICA 客户端并作为终端使用，Windows 3.11 计算机就可以运行 Windows 2000。胖客户机的缺点是有可能出现磁盘故障并需要不断维护硬盘。

“瘦客户机”"是不带硬盘并与终端服务器一起使用的客户机。这种机器配备操作系统和芯片内置的客户端软件，或者具有从服务器下载的操作系统和客户端，它们与终端服务器连接并在那里运行其应用程序。瘦客户机有其优势，因为其所需维护量极少，而且由于排除了硬盘故障的机会，因而提高了可靠性。使用网络下载操作系统的瘦客户机可以在不更换芯片或组件的情况下进行系统升级。升级服务器上的下载软件就可升级所有瘦客户机。

“总体拥有成本”的思路（TCO）很简单：最大限度地增加公司的技术投资回报 (ROI)，同时最大限度地减少所投入的成本。您打算减少的成本可以分为两类：*硬成本*和*软成本*。硬成本是采购和配置硬件所需的成本。软成本是与终端用户支持、培训以及生产环境维护有关的成本。采用瘦客户机（没有硬盘的终端）、终端、新型家电类设备、小型 PC 机，有希望减少购置计算机硬件，特别是桌面级别的*硬成本*，同时减少与系统管理有关的管理性软成本。换句话说，通过实施以瘦客户端/服务器为中心的计算系统模式，IS 经理可以大大降低计算机设备的总体拥有成本，同时提高为用户提供的服务水平。

用户环境

您可以通过在 Windows“**程序**”菜单上运行终端服务客户端命令，以从客户端访问终端服务。一旦与终端服务器连接，客户机的环境看起来与 Windows 2000 server 和 Windows 2000 Professional 环境一样。由于应用程序不在本机运行，因而用户完全一目了然。

应用程序在客户端桌面的终端服务窗口上打开。所有应用程序处理均在运行终端服务的服务器上进行，服务器将显示结果发送到所有客户机。

终端服务的多用户系统环境由三个部分组成：

1. **终端服务器**。服务器管理用于每个客户端会话的计算资源，并为登录的所有用户提供各自不同的独特环境。服务器接收和处理远程客户端执行的所有键盘输入和鼠标动作，并将操作系统和应用程序的显示输出发送给相应的客户端。
2. **远程桌面协议**（**RDP**）。RDP 支持客户端和服务器之间的通信。RDP 在将图形界面要素传递到客户端方面进行了优化。RDP 是应用层协议，它依赖传输控制协议/Internet 协议 （TCP/IP）在网络间载送信号。RDP 的基础是多通道会议的国际通信联盟（ITU）T.120 标准。
3. **客户端**。终端会话作为本机桌面环境的一个窗口打开。在该窗口内运行的是终端服务器的远程桌面。计算机或设备只需要极少的必备软件，即可建立与服务器的连接并显示用户界面。

终端服务优点

本章介绍终端服务的功能和优点。

更为快速地将 Windows 2000 呈现到桌面。在从旧桌面移植到 Windows 2000 Professional 时，终端服务起到了消除其中差距的桥梁作用，它将 Windows 2000 桌面的体验“虚拟”发送到非计算机桌面及需要硬件升级才能在本机运行完整 Windows 2000 操作系统的计算机。终端服务客户端可以在各种不同的平台上获得，包括 MS-DOS、Windows 终端、Macintosh 及 UNIX（与 MS-DOS、Macintosh 及 UNIX 计算机连接需要其它软件。）

充分利用现有硬件。终端服务允许计算机同时作为瘦客户机和全功能的个人计算机进行操作，是对分布式计算模式的扩展。计算机可以继续发挥其在现有网络中的作用，同时还可充当瘦客户机来模拟 Windows 2000 Professional 桌面。

远程管理和支持。终端服务为 Windows 2000 Server 提供了远程管理的功能，使系统管理员可以从广域网的任何一台客户机或通过拨号连接来远程管理其服务器。管理员可从某个桌面远程管理 Windows 2000 服务器。管理员可访问系统管理工具并能够进行所有管理性工作，包括软件安装，就象在服务器本机上工作一样。

取代文字终端。由于许多 Windows 终端本身就支持同一台设备上的终端仿真，所以可以用 Windows 终端来取代文字终端。Windows 终端提供了利用来自大型机系统的数据来访问更新的 Windows 应用程序，如电子邮件的可能。

更为安全可靠。管理员可利用多层加密技术，根据安全要求以不同的级别（低、中、高）为 Windows 2000 Server 和终端服务客户机之间传输的所有数据或某些数据进行加密。另外，终端服务登录过程包括更改口令、桌面解锁和屏幕保护解锁等功能。登录过程是加密的，从而确保用户名和口令的安全传送。终端服务支持服务器与客户机之间的 40 位和 128 位两种加密（此功能仅适用于美国和加拿大）。

- 因为在客户机上不驻留任何应用程序或用户数据，终端服务提供了更为有力的安全保障。终端服务还支持多级加密功能，可在服务器与客户机之间的传输链路有被窃密的可能性时使用这项功能。加密级别有三种：低、中、高。所有加密级别均采用 RSA RC4 加密技术。
- 采用瘦客户机可防止在出现计算机故障的情况下丢失数据。因为数据是在终端服务器上进行处理和保存的，客户机损坏时不会导致数据的破坏。这样就减少了需要加强数据保护的节点数。

有关详细信息，请参阅 Microsoft Windows 2000 Server 管理员手册。

远程控制。管理员可利用远程控制功能临时控制用户的会话，以及观察用户的活动。管理员还可与用户互动并以用户的名义执行命令。要充分利用使用 RDP 的远程控制功能，两个客户机均必须与 Windows 2000 终端服务器连接。

注意：在启用远程控制会话之前，您可以通过在客户机上显示信息及要求客户机允许查看或参与会话的方法来通知用户。远程控制的设置可设为在无需用户同意的情况下进行远程控制。鼠标和键盘输入可以启用，也可关闭。

统一程序布署。通过在 Windows 2000 Server 上运行终端服务，所有的程序执行、数据处理和数据存储均可在服务器上完成，从而实现程序的统一布署。终端服务保证所有客户端均可访问程序的当前版本。软件只安装在服务器上，而不是整个组织的每台机器上都要安装，从而降低了升级每台计算机所需的相关成本。

集中管理。Windows 终端服务具有集中管理的功能，同时也为特定用户提供了使用 Windows 桌面环境的灵活性。由于 Windows 终端服务会话的所有应用程序只驻留在一处 — 服务器上 — 处理能力和系统软件版本的所有升级均在服务器级上进行，而不是在桌面级上进行，从而免去了耗时的现场升级活动，无需开发、安装和升级客户端应用程序。因此，大大减少了安装和相关的软件费用。用户特征数据存储在终端服务器上，这样可对客户端桌面进行集中管理，从而提高了应用程序管理效率（集中备份和存储、版本升级等等）。

另外，Windows 终端服务还允许管理员检查用户会话的进行情况，甚至可以直接进行控制。支持人员可以在不离开办公桌的情况下，真实准确地观察到用户所看到的内容。如果用户作相应配置，支持人员可以参与会话的控制，与用户一起解决难题。因为花在排除故障上的时间更少，使得维护成本降低，本机感染病毒的危险减小，软件升级更为容易，发生故障的平台更易更换而不会丢失数据。

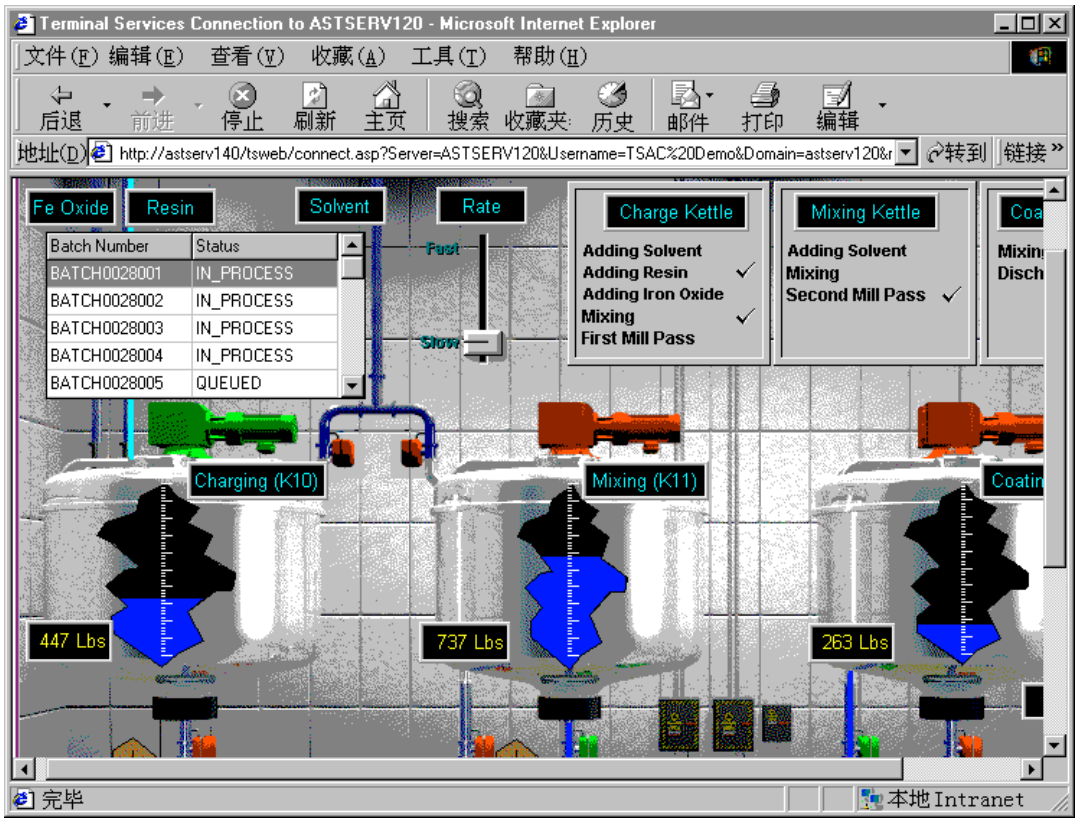
当配置为远程管理模式时，Windows 终端服务还可用作管理工具。管理员可直接在他们的桌面登录到服务器，进行正常系统维护而无需回到服务器控制台上。

通过实施 Windows 终端服务，您可以集中管理传统上驻留在用户桌面的三个项目内容：

1. 应用程序支持和维护
2. 操作系统维护
3. 存储管理

终端服务高级客户端

Microsoft 的终端服务高级客户端 (TSAC) 是一个基于 Win32® 的 ActiveX® 控件，可用于在 Microsoft® Internet Explorer 内运行终端服务会话。通过使用 TSAC，您现在可以在 Internet 上运行全功能的 InTouch 应用程序，同时可以达到与局域网相同的性能和速度。



优点

此可下载的 ActiveX 控件可提供与完整的端终服务用户端相同的功能，其独特之处在于能够在 Web 上实现这些功能。TSAC 可提供下列优点：

在 Internet Explorer 内运行会话。不需要在客户机上安装终端仿真软件，只需使用 Internet Explorer 4 或以上版本，并在必要时将 URL 地址指向**终端服务器**。

快速简便地访问终端服务器。TSAC 特别适合根据需要来访问**终端服务器**。URL 地址可以包含可选字段，例如用户名和屏幕大小，您可以轻松地访问**终端服务器**，就如单击“收藏夹”链接一样。

通用界面。与 Internet Explorer 一样的外观使得它成为查看 MS Office™ 应用程序、浏览工厂信息、使用 ActiveFactory™ 或 SuiteVoyager™ 进行趋势分析的首选 GUI。

安装

TSAC 是 Microsoft 网址上提供的一个免费 ActiveX 控件。它必须安装在运行 Internet Information Services (IIS) 第 4.0 或以上版本的计算机上。但此限制只适用于 Web 服务器。用户可以下载控件，并从任何所支持的 Web 浏览器（Internet Explorer 4.x 或 5.x 的 Windows 32 位版本）上查看会话。

注意：有关最新信息或要下载 TSAC，请访问 http://www.microsoft.com/WINDOWS2000/downloads/recommended/TSAC/default.asp。

如何使用

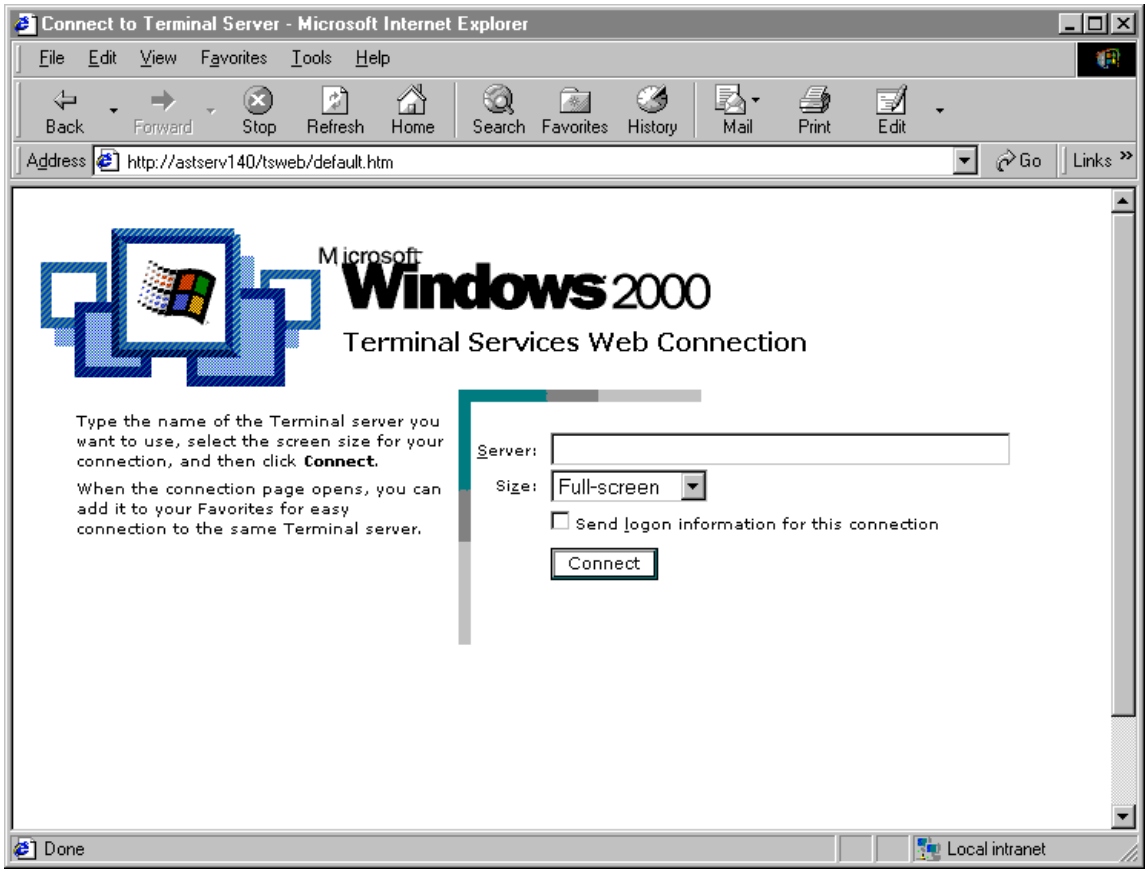
一旦在 Web 服务器上安装 TSAC，用户就可以指向缺省登录页并传递特定的用户信息来初始化**终端服务器**会话。TSWeb 目录中安装了三个示例 Web 页。这些示例页可以照原样运行，也可以加以修改。

注意：有关如何配置和使用示例页的详细信息，请参阅《Microsoft® Terminal Services ActiveX® 客户端控件布署指南》。

Default.htm。Default.htm 是设计从用户端收集**终端服务器**连接信息的示例登录页。您可以通过下面的 URL 来访问该缺省页：

http://MyWebServer/TSWeb/default.htm

此处，`MyWebServer` 是 Web 服务器的计算机名或 IP 地址。



Connect.asp。Connect.asp 是包含用于寄存终端服务器会话的实际 ActiveX 客户端控件的示例页。Connect.asp 的设计不能独立运行，而必须使用下面的参数来调用：

http://MyWebServer/TSWeb/connect.asp?Server=MyTSServer&Username=MyUser&Domain=MyDomain&rW=800&rH=600

此处，`MyWebServer` 是 Web 服务器的计算机名或 IP 地址，`MyTSServer` 是**终端服务器**的计算机名，`MyUser` 是有效的登录名，`MyDomain` 是已定义登录名的计算机名。

注意：要使用此示例页，必须在 Web 服务器上启用 Active Server Pages (ASP)。

如果您通过*防火墙*来访问 Internet，请确保打开 TCP 端口 3389。

保护基于 Web 的应用程序

远程控制流程的后面潜伏着安全和可靠性问题，因为 Internet 面临着日益增长的非法访问的风险。Internet 是一个公共媒体，任何一次连接都可能意外地暴露机密信息，或令系统遭受恶意行为的蓄意破坏。为充分保护您的**终端服务器**及其控件下的流程，您需要规划一套充分有效的信息安全 (INFOSEC) 政策。该 INFOSEC 政策应包括物理安全性、网络安全性、应用安全性以及安全性跟踪。

物理安全性

物理安全性是指服务器及其所连接的客户端系统的操作环境。

请将终端服务器置于一个可免遭物理攻击或远离不利条件的受保护空间内。确保该空间只向授权（可依赖）人士开放。

制订数据备份时间表，并公布有关如何恢复数据的程序。

当终端服务器当机时，立即评估其风险。使用电涌抑制器、不间断电源和备用服务器等硬件保护措施有助于确保系统持续运行。通过网络负载平衡或实施高可用性系统可有效减少因部件故障而导致停工的机会。

网络安全性

网络安全性是指**终端服务器**与客户机之间的数据传输。

提供充分的计算机登录安全性。

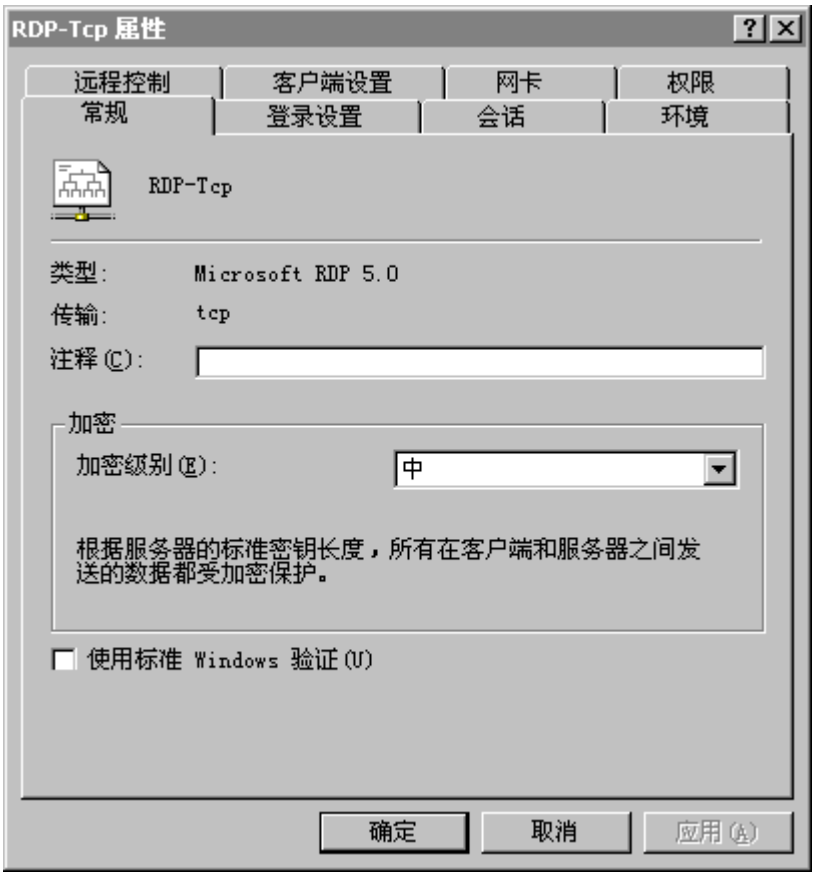
有关详细信息，请参阅《*Terminal Services for InTouch 布署指南*》。

启用中等（或以上）加密。加密可防止期诈行为的发生，期诈是指非法截取数据、用户验证、部分或全部的数据传输。终端服务提供多级加密。所有加密级别均采用 RSA RC4 加密模式。

级别	描述
低	此级别使用 56 位或 40 位密钥对从客户端发往服务器的所有数据加密。在连接 Windows 2000 客户端的情况下，Windows 2000 **终端服务器**使用 56 位密钥，在连接更早版本的客户端时则使用 40 位密钥。这种只加密输入数据的方法可保护用户口令等机密数据。
中 **（缺省）**	此级别使用 56 位或 40 位密钥，对双向传送的数据（从客户端到服务器及从服务器到客户端）进行加密。在连接 Windows 2000 客户端的情况下，Windows 2000 **终端服务器**使用 56 位密钥，在连接更早版本的客户端时则使用 40 位密钥。使用中等加密，可对要在网络上传输以显示在远程客户端上的机密数据进行加密。
高	高加密方法应用于双向传输的所有数据，它使用 128 位的密钥进行加密。

要启用加密：

1. 在 Windows 任务栏上单击“**开始**”，指向“**程序**”，指向“**管理工具**”，然后单击“**终端服务配置**”。
2. 双击“**RDP-Tcp**”，“**RDP-Tcp Properties**”（RDP-Tcp 属性）对话框出现：



3. 单击“**General**”（一般）标签，激活“**一般**”属性页。
4. 选择适当的“**Encryption level**”（加密级别）。

应用程序安全性

应用程序安全性是指 InTouch 应用程序、Industrial *SQL Server* 及其它机密信息系统中内嵌的安全性。

使用 $Operator 标记名以提供 InTouch 应用程序内的安全性。通过在应用程序中应用安全性，您可以控制操作员允许执行的特定功能，这可以通过将这些功能与内部标记名相链接来实现。

有关 $Operator 标记名的详细信息，请参阅您的联机 *《InTouch 用户手册》*第 2 章中的“使用 InTouch 安全性”部分。

使用新的 TseGetClientId() QuickScript 取代 GetNodeName() QuickScript 来识别客户机。在使用终端服务时，GetNodeName() 会返回终端服务器名，而不是客户机名。

对 SQL Server 系统管理员 (SA) 帐号添加口令。当您安装 Industrial SQL Server 时，系统会创建一个全功能的 "sa" 登录 ID 及一个空口令。请不要使用此帐号来访问数据，而使用缺省登录 ID（例如 wwUser）。

有关数据库安全性的详细信息，请参阅 *《Industrial SQL Server 系统管理员指南》*中的“管理安全性”章节。

安全性跟踪

安全性跟踪是指监视非法入侵企图的能力。如果您怀疑系统处在任意类型的攻击之下，则可以启用对一系列可跟踪事件的日志跟踪。缺省条件下，安全性日志/跟踪是被禁用的，因为它通常要求大量的处理资源。因此，我们建议您在开始时只选择监视少数事件。

警告：系统跟踪要求大量的资源。确保在评估您的试用服务器时启用跟踪功能，否则可能会导致硬件配置过低。

要配置跟踪功能，请参阅“Windows 2000 本机安全性政策”部分的“跟踪政策”。

附加信息

有关上述及相关安全性注意事项的进一步说明，请参阅下面的资源：

National Computer Security Center (NCSC) "Rainbow Books" – http://www.radium.ncsc.mil/tpep/library/rainbow/index.html

Common Criteria (CC) for Information Technology Security Evaluation – http://www.commoncriteria.org/cc.html

Microsoft Privacy & Security Fundamentals: Security – http://www.microsoft.com/privacy/safeinternet/security/best_practices/default.htm

Windows 2000 Security Technical Overview – http://www.microsoft.com/technet/win2000/win2ksrv/sectech.asp

Default Access Control Settings in Windows 2000 – http://www.microsoft.com/technet/win2000/win2ksrv/technote/secdefs.asp

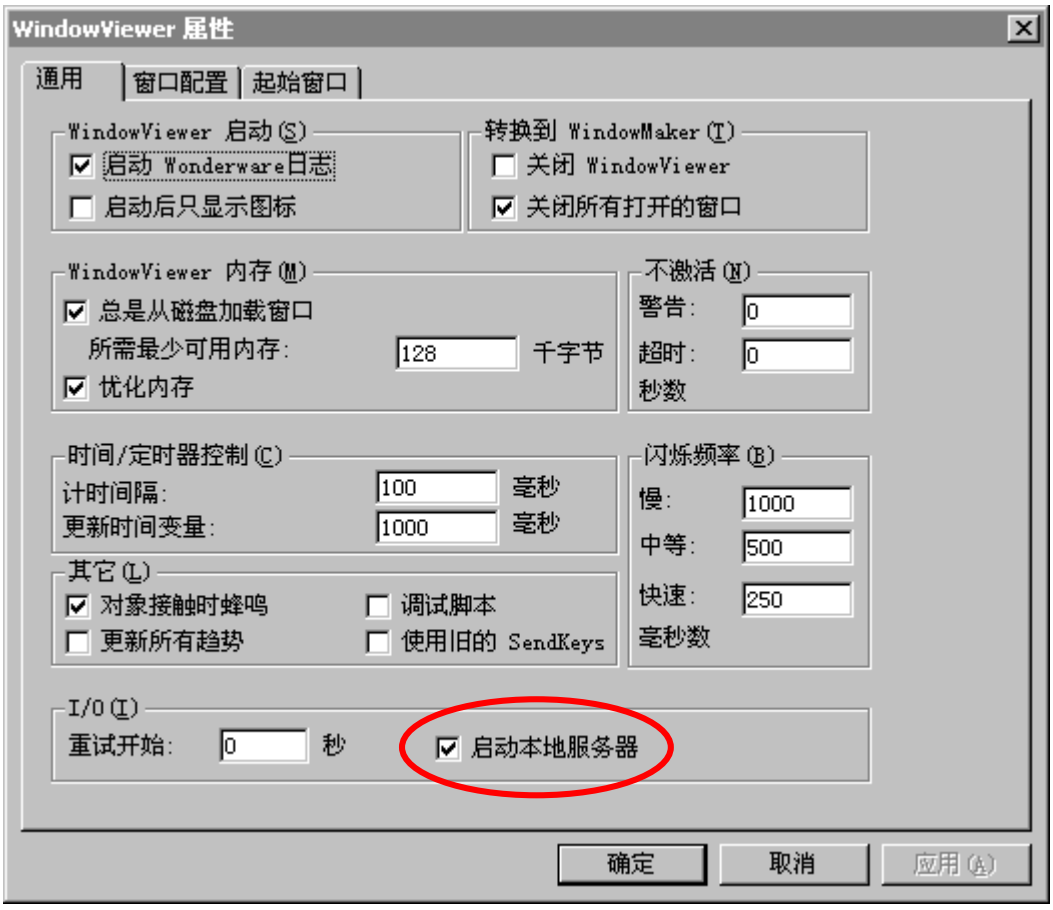
已知问题和限制

下面介绍您在**终端服务器**上运行应用程序时的一些限制及您可能需要实施的建议解决方法。Wonderware 将全力在下一版本中解决当中的大多数问题。

功能	是否支持?	注释
AlarmSuite Logger	否	使用标记名服务器（仅限于单机）
与 I/O 设备或 MS Office 2000（如 Excel）的 DDE	否	使用标记名服务器（控制台或单机），这包括 DDE QuickScripts：**WWExecute()**, **WWpoke()** 和 **WWRequest()**
MS Office 2000 中的 DDE（例如 Excel 中的热链接配置）	是	Excel 和 InTouch 必须在同一会话中运行
历史趋势	是	使用标记名服务器或 NAD 来记录数值。多个会话可能会读取相同的历史文件。
InBatch	否	开发中 – 将在未来版本中提供支持
InControl	是	InControl 必须在控制台上运行，并且要求 v7.1 Patch 03 或以上版本。
InSQL ActiveX 控件 (OLE DB)	是	InSQL 应在单机上
InTouch Alarm Logger	否	开发中 – 将在未来版本中提供支持
InTrack OLE Automation	否	开发中 – 将在未来版本中提供支持
音频 QuickScript	是	需要使用 Citrix MetaFrame
打印报警	否	开发中，即将提供支持。并行打印机可能需要配置为使用 “net use” 命令从客户端打印。详情参阅 Tech Note 149。
保留标记名	是	必须使用 NAD
SPC Pro	否	不支持
SQL Access (ODBC)	是	数据库必须位于单机上
与 I/O 设备或另一 InTouch 应用程序的 SuiteLink。	是	当与另一视图会话进行通讯时，需添加**终端服务器**节点名并在应用程序名中添加所需会话的 IP 地址。例如 view10.103.25.6。

启动本地 I/O

InTouch 不能在**终端服务器**环境下启动 I/O 服务器。为避免在启动 **WindowViewer** 时出现“正在初始化 I/O”错误消息，请在“**WindowViewer 通用**”属性页上关闭“**启动本地服务器**”选项。



注意：取决于视图会话的启动顺序，您可能需要执行 **IOReinitialize** QuickScript。在启动应用程序以从这些服务器读取数值之前，必须运行所有服务器（I/O 设备或视图应用程序）。

执行脚本

由于运行于**终端服务器**上的所有应用程序都使用相同的定时引用（服务器时钟），因此脚本有可能在异常 CPU 加载期间不会执行。异常 CPU 加载可能由视频处理超负荷引起，或者是多个应用程序定义了相同的脚本触发器（例如 End-of-Shift 事件）。因此，如果服务器忙于处理许多客户端的脚本，则在通常需要启动脚本的时间间隔内，它可能不会启动另一客户端的脚本。这可能引起客户机上的脚本不执行。

为确保正确执行脚本，请将具有公用触发器的脚本合并到一个单一应用程序中，例如标记名服务器。这是进行服务器试验布署的主要原因之一。试验布署可以让您有机会执行因果分析方案并决定所选硬件配置是否足够。

服务器硬件要求

Terminal Services for InTouch 必须安装在新的 **Windows 2000 Server** 或 **Advanced Server** 操作系统上，而不能从 Windows NT 系统升级。下表提供了推荐的硬件平台。它们可以为执行典型 InTouch 应用程序提供良好的性能。

推荐硬件

CPU [1]	物理内存 [2, 3]	虚拟内存 [4]	客户机数
Pentium III 450 MHz	384 MB	960 MB	5
Pentium III 500 MHz	1024 MB	2560 MB	15
Pentium III 700 MHz	2048 MB	5120 MB	25

1. 多处理器可以提高系统性能。
2. Windows 2000 Advanced Server 需增加 128 MB RAM。
3. 内存要求取决定于应用程序负荷及所连接的用户数。RDP 需要每位用户 40-60 MB 来运行 InTouch，ICA 的要求要更低些。
4. 虚拟内存（页面文件大小）应为物理内存的 2.5 倍。

注意：要估计一台服务器可支持的用户数，一个较好的方法是让两到五个用户在系统上试运行，以评定系统性能，然后得出结果。

有关分析系统性能的详细信息，请参阅《*Terminal Services for InTouch 布署指南*》。

硬盘空间

一个或多个硬盘，其中包含系统文件的分区必须至少具备 2 GB 空间。

通过使用冗余阵列磁盘机 (RAID)，有助于防止服务器当机时造成的数据损失。

网络连接

10/100Mbps 的网卡；网络使用 TCP/IP 协议。

其它驱动器

光驱

使用高密 3.5 英寸磁盘驱动器作为 A 盘，除非：

计算机支持从光盘启动安装程序。

您通过网络安装 Windows 2000。

附件

键盘、鼠标（或其它指针设备）和显示器（VGA 或更高配置）。

UPS（不间断电源）

注意：在安装 Windows 2000 之前，检查您的硬件是否在 Windows 2000 兼容硬件表（HCL）中列出。(HCL).由于 Microsoft 只对 Windows 2000 HCL 中列出的设备提供经测试的驱动程序，因而使用未在 HCL 中列出的硬件可能会在安装后出现问题。网站 http://www.microsoft.com/hwtest/hcl 提供了最新版本的 HCL。

检查影响系统性能的外围设备

下列外围设备也可能影响运行终端服务的服务器的性能：

硬盘硬盘速度对终端服务器的性能非常重要。小型计算机系统接口 (SCSI) 磁盘驱动器，特别是与 Fast SCSI 和 SCSI-2 兼容的设备，其输入输出能力明显优于其它驱动器类型。

网络适配器。建议使用高性能的网络适配器，特别是当用户需要访问网络服务器上存储的数据或者访问客户端/服务器应用程序（如 Wonderware InTouch）时。使用多个网络适配器可显著提高网络输入输出能力。

RDP 客户端的硬件要求

运行终端服务的客户机不需要很强的处理能力。例如，带 32 MB RAM 和 VGA 显卡的 Pentium 机就足够了。因此，可以很容易地将终端服务集成到包含较旧计算机和设备的网络中。

注意：使用标准 VGA 显卡可能会限制显示大小和颜色深度。

取决于所使用的客户机型号，您需要注意某些性能方面的问题。例如，使用 Microsoft Windows CE 作为操作系统的设备，其速度比不上使用 Linux 的相同设备。

规划终端服务安装

终端服务安装的成功关键是正确规划。在安装终端服务之前，请执行下列任务：

确定您需要安装在服务器上的客户应用程序（例如 Wonderware InTouch）。

确定客户机的硬件要求。

确定支持客户机所需的服务器配置。

确定终端服务以及运行所有其它应用程序所需的许可证。

在安装终端服务之前，确定您要在客户端桌面上布署的应用程序。大多数可在 Windows 2000 上正确运行的应用程序都可以在终端服务器上运行。但某些应用程序可能需要在局部修改之后，才能在终端服务环境下成功运行。在运行环境下布署应用程序之前，请先在试用服务器上安装这些应用程序，以确保与您现有的应用程序兼容。

有关详细信息，请参阅*《Terminal Services for InTouch 布署指南》*。

安装终端服务

您可以在 Windows 2000 Server 安装期间安装终端服务，也可以在安装后通过控制面板上的“**添加/删除程序**”安装终端服务。

重要！！终端服务必须在安装任何软件产品之前安装。软件产品必须在“安装模式”下添加到终端服务器。使用 Windows 控制面板中的“**添加/删除程序**”可以自动激活“安装模式”。它允许您为机器上的所有用户配置软件，而不只限于原来安装程序的那些用户。

终端服务组件

终端服务安装包括下列组件和选项：

终端服务。为访问服务器上基于 Windows 的应用程序提供多会话环境。终端服务包括两个子组件：

1. **客户端生成器文件**。包含为终端服务客户端制作安装磁盘的向导。
2. **启用终端服务**。启用服务器上的终端服务软件。您可以使用此选项来启动和停止终端服务。

终端服务授权。将计算机配置为提供客户许可证的终端服务授权服务器。



要安装终端服务：

1. 在 Windows 任务栏上，单击“**开始**”，指向“**设置**”，然后单击“**控制面板**”。
2. 双击“**添加/删除程序**”图标。“**添加/删除程序**”对话框出现。
3. 单击“**添加/删除 Windows 组件**”。“**Windows 组件向导**”出现。
4. 选择“**终端服务**”和“**终端服务授权**”选项，然后单击“**下一步**”（终端服务授权可能安装在不同的机器上）。
5. 选择“**应用程序服务器模式**”选项，然后单击“**下一步**”。
6. 选择“**跟 Windows 2000 用户兼容的权限**”选项，然后单击“**下一步**”。
7. 选择“**您的域或工作组**”选项，并指定授权服务器数据库的目录位置。

注意：仅当您选定“**终端服务授权**”选项时，此选项才会出现。

8. 单击“**下一步**”开始安装。

一旦安装完成，“**管理工具**”菜单中会增加几个菜单项。下表介绍了这些新增的菜单项：

菜单项	描述
终端服务客户端生成器	为安装终端服务客户端软件制作软盘。
终端服务配置	管理终端服务协议配置和服务器设置。每个网络适配器只能配置一个 RDP 连接。
终端服务授权	管理终端服务的客户端访问许可证。
终端服务管理器	管理和监视运行终端服务的服务器上的会话和进程。

安装 Terminal Services for InTouch

要安装 Terminal Services for InTouch，您必须使用内置的管理员帐号，登录到运行终端服务的服务器上。

注意：如果您已经安装了 InTouch 的非终端服务版，您将无法升级到 InTouch 的终端服务版。但是，您可以从先前安装的 Terminal Services for InTouch 版升级到最新的版本。



要安装 Wonderware Terminal Services for InTouch：

1. 将 Wonderware Terminal Services for InTouch 光盘放入光驱，等待程序自动运行以自动激活 InTouch **SETUP.EXE** 程序。
2. 当“**欢迎**”对话框出现时，单击“**下一步**”。
3. 跟随屏幕说明安装 InTouch。
4. 一旦安装系统文件和公用组件，程序会提示您重新启动系统。单击“**确定**”以重新启动计算机。在系统重新启动后，程序会自动恢复 InTouch 安装。

注意：如果该系统已安装有 Terminal Services for InTouch，程序不会如上面的步骤 4 所述，提示您重新启动系统。

5. 在成功安装 InTouch 后，一个对话框会出现，告诉您安装已完成。
6. 此外，在成功安装 InTouch 后，还可能会出现下面的“**Question**”（问题）对话框：



单击“是”以继续安装 MSDE 2000，或者单击“否”以退出 InTouch 安装。

注意：Microsoft 建议在终端服务器节点之外的节点上安装 SQL Server/MSDE。有关如何设置 SQL Server/MSDE 节点的详细信息，请参阅《Terminal Services for InTouch 布署指南》或《InTouch 报警布署指南》。

在安装 InTouch 后，您可以使用三个选项：**“整个开发系统”**、**“只有运行时”**或**“Factory Focus”**。这三个选项决定**终端服务器**及其相应的客户端运行 InTouch 的方式。您不能混合使用和匹配 InTouch 配置。例如，如果您要让某些客户端运行“运行时”，而让其它一些客户端运行“Factory Focus”，则您需要安装两台**终端服务器**。

7. 安装相应的 Wonderware 许可证。

   您现在就可以开始使用 Terminal Services for InTouch 了。

在终端环境下测试您的应用程序

为确保应用程序能够在终端服务环境下正确安装和运行，在终端环境下进行测试是关键。因而有必要构建一个典型的使用方案，确保适当的会话数运行应用程序一段时间来模拟实际的应用。

Terminal Services for InTouch QuickScript 函数

本节向您介绍终端服务可用的三个 InTouch QuickScript 函数。

TseGetClientId()

如果 View 应用程序当前在终端服务客户端上运行，此函数返回字符串版本的客户端 ID（客户端的 TCP/IP 地址），否则返回一个空的字符串。

语法

```
MessageResult=TseGetClientId();
```

参数	描述
MessageResult	显示客户端 ID 的消息型标记名。

备注

此 ID 供内部使用，用于生成 SuiteLink 服务器名及 logger 文件名。

示例

```
MsgTag=TseGetClientID();
```

客户端 IP 地址，例如 **10.103.202.1** 将返回给 **MsgTag**。

TseQueryRunningOnConsole()

如果 View 应用程序当前在终端服务器控制台运行，此函数将返回一个非零的整型值，否则返回零 (0) 值。

语法

```
Result=TseQueryRunningOnConsole();
```

参数	描述
Result	如果 View 在 TS 控制台上运行，返回 0 。

备注

无

示例

```
IntTag=TseQueryRunningOnConsole();
```

如果 WindowViewer 在终端服务控制台上运行，返回 IntTag = 1。

TseQueryRunningOnClient()

如果 View 应用程序当前在终端服务器服务端运行，此函数将返回一个非零的整型值，否则返回零 (0) 值。

语法

```
Result=TseQueryRunningOnClient();
```

参数	描述
Result	如果 View 在 TS 客户端运行，返回 0 。

备注

无

示例

```
IntTag=TseQueryRunningOnClient();
```

如果 WindowViewer 在终端服务客户端运行，返回 IntTag = 1。

Wonderware Logger 控制台和客户端文件

在 Windows 终端服务器上，系统会根据从中产生日志消息的客户端的 IP 地址，创建日志文件。这意味着，不同客户机的日志消息存储在不同的日志文件中。这些文件名的格式为 "**SClientIPAddress_YYYYMMDD.AEH**"（客户端 IP 地址前接一个 "S"）。

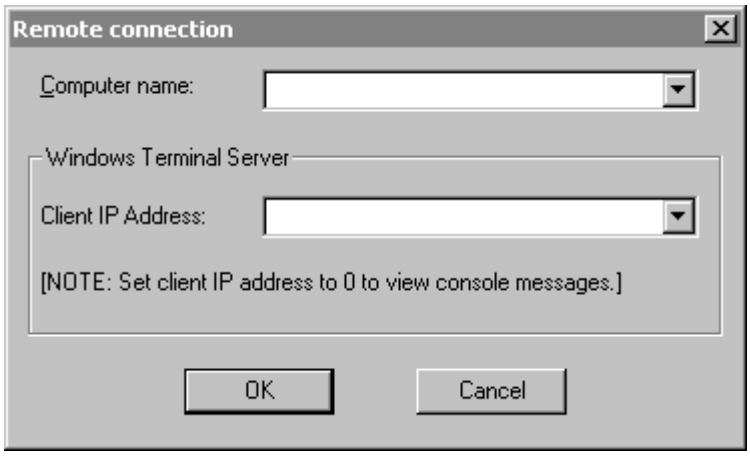
例如，**S10_32_13_103_20001225.AEH**，此处的 **"**10_32_13_103" 从实际 IP 地址 "10.32.13.103" 转换而来，后跟日志创建的年、月和日。

*控制台*日志文件名遵循标准的 Wonderware Logger 命名惯例，如下所示：**YYYYMMDD.AEH**。

例如 **20001225.AEH**

要在 Windows 终端服务器上查看指定客户机的日志文件：

1. 在“**连接**”菜单上，单击“**远程**”。“**Remote connection**”（远程连接）对话模式出现：



2. 在“**Computer name**”（计算机名）输入框中，输入日志文件所在的计算机名。
3. 在“**Client IP Address**”（客户端 IP 地址）输入框中，输入您要查看其日志文件的客户端的 IP 地址。该地址必须是有效的 IP 地址，并且必须属于连接到指定**计算机名**的客户端。

注意：输入 "0" 以显示控制台的日志文件。

---

4. 单击“**OK**”（确定）以查看日志文件。

   该计算机名会出现在 Wonderware Logger Viewer 窗口的标题栏中。如果您要查看客户端而不是控制台上的日志文件，则在计算机名后面还会显示客户端的 IP 地址。

建立终端会话

在您完成服务器和客户端配置后，您就可以建立终端会话。您可以从客户端访问网络和本地资源，包括硬盘和打印机。当会话结束时，您可以断开连接并稍后重新建立会话，也可以注销以完全关闭会话。

连接到终端服务器

当您连接到终端服务器时，您可以根据低速网络来调整选项以提高会话性能。

要连接到终端服务器：

1. 启动终端服务客户端。“终端服务客户端”对话框出现，显示下列选项：

选项	操作
服务器	输入终端服务名或 TCP/IP 地址。
屏幕区域	选择屏幕分辨率。此设置与服务器的屏幕分辨率无关。
可用服务器	从可用服务器列表选择服务器。
低速连接	如果您使用调制解调器或低速网络，请单击“低速连接”。
缓存位图到磁盘	单击“缓存位图到磁盘”以将桌面显示元素保存到本地缓存中。此选项可以从本地缓存刷新屏幕，并提高系统。

2. 单击“**连接**”。
3. 登录到终端服务器。

结束终端会话

终端服务提供了两个选项来结束终端会话。

断开会话连接。断开连接，并让会话继续在服务器上运行。您可以稍后重新连接到服务器并恢复会话。例如，如果您在服务器上执行耗时性任务，例如运行数据库查询，则可以在启动任务后断开会话连接。稍后，您可以再次登录到服务器，恢复会话并恢复任务或检查结果。

注销会话。注销会话可以结束当前在服务器上运行的会话。任何在会话内运行的应用程序都会关闭，所有未保存的数据均将丢失。切记要注销会话，以将服务器资源让给新会话。

其它应用程序问题

某些应用程序所具备的功能可能无法支持终端服务或运行性能低下。例如，需要条形码扫描仪或智能卡阅读器等特殊硬件的应用程序只有在以下条件下才能使用终端服务：

设备已连接到计算机或终端，并且外围设备被识别为键盘型设备。

连接软件和硬件支持与客户端的连接。

配置客户端设置

为确保系统资源用于活动状态的终端会话，您可以为断开连接和闲置的会话设置时间限制。

您可以在服务器终端服务配置“**RDP-Tcp 属性**”对话框的“会话”属性页中，指定会话的时间限制。下表介绍用于限制会话长度的设置。

设置	描述
结束断开连接的会话	指定断开连接的会话可保留的最长持续时间。会话将重置，并且在时间限制过后将不可再恢复。
活动会话限制	指定连接的最长持续时间。当时间限制到达时，系统将断开会话连接，让会话继续在服务器上运行或重置会话。
闲置会话限制	指定在会话断开连接或重置前允许的最长闲置时间（无连接活动的时间）。

第 11 章

InTouch 应用程序发布器

本章概括介绍 InTouch 应用程序发布器，您可以使用它来创建自解压文件，其中包含您需要的所有相关文件和安装步骤，以最小化 Internet 上的下载时间。

目录

- 发布 InTouch 应用程序
- 用多种分辨率发布应用程序

发布 InTouch 应用程序

应用程序发布器可以创建一个自解压文件，该文件包含在另一 InTouch 节点上安装应用程序所需的所有相关文件和安装程序。应用程序发布器通过压缩应用程来最大程度地减少在 Internet 上的下载时间

注意：您可以将这些压缩程序复制到软盘、zip 盘等，以便在其它节点上安装。

应用程序发布器的压缩率约为 19:1，因而可以将 20 MB 的完整开发程序压缩在一张标准的软盘上面。如果文件只是运行时文件，则可以将 30 到 40 MB 的应用程序压缩到一张标准软盘上。

一旦发布应用程序，您就可以在一个或多个其它 InTouch 7.1（或以上版本）节点上重建 InTouch 应用程序。

您可以使用多种方法将要发布的应用程序文件载入另一 InTouch 节点：

复制到软盘

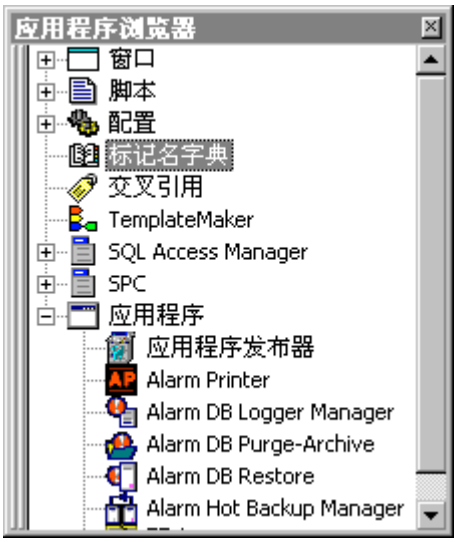
通过公司的 Intranet 复制文件

通过 Internet 复制文件

通过将自解压文件作为程序在目标节点上运行，您就可以使用重建后的 InTouch 应用程序。

➢ **要发布 InTouch 应用程序**

1. 在 WindowMaker 中，从应用程序浏览器中，打开“**应用程序**”。



2. 选择“**应用程序发布器**”。“**InTouch 应用程序发布器 – 第一步/共四步**”对话框出现：



3. 单击“**下一步**”，“**InTouch 应用程序发布器 – 第二步/共四步**”对话框出现：



4. 在“**作者名**”框中，输入有关应用程序的联系人姓名（最多 256 个字符），这是一个可选输入框。例如：

   Ernie Tankhammer

   Ernie Tankhammer Senior Engineer Phone 1-800-555-1234

5. 在“**描述**”框中，输入应用程序的说明（最多 256 个字符），这是一个可选输入框。例如：

   Real Time Production Status

   Plant Number 5-Area 3 that requires access to zone 3 FactorySuite File Server for data point access.

6. 在“**软件包名**”框中，输入软件包的名称（最多 32 个标准 Windows 文件命名字符）。您**必须**输入**软件包名**。

注意：软件包名是用于识别要发布的应用程序的唯一名称。它用于维护同一发布应用程序的可同时下载的版本。

---

7. 单击“**下一步**”，“**InTouch 应用程序发布器 – 第三步/共四步**”对话框出现：



8. 在输入框中，输入 InTouch 应用程序的源目录。缺省为 WindowMaker 应用程序目录。

9. 选择“**只能运行**”选项，以防止在要发布的文件中包括 WindowMaker 开发文件。

10. 单击“**下一步**”，“**InTouch 应用程序发布器 – 第四步/共四步**”对话框出现：



11. 检查第一个框中的可执行文件名是否正确（缺省条件下，此文件名与您指定的“**软件包名**”相同）。

12. 在第二个框中，输入用于保存要发布的文件的目录路径（缺省条件下，文件保存到您当前的临时目录）。或者，单击“**浏览**”以找到不同的目录。

注意：在上面的实例中，文件会发布到公司 intranet 上的 FactorySuite Web 服务器中。

---

13. 单击“**完成**”。

用多种分辨率发布应用程序

您可以用多种分辨率发布应用程序。这可以避免客户端无法以正确的分辨率查看应用程序的问题。

- **要按不同的分辨率分布应用程序：**

1. 在运行 WindowMaker 时，改变个人计算机的屏幕分辨率。
2. WindowMaker 会通知您分辨率已改变，并提示您接受改变，单击**“确定”**。
3. 从应用程序浏览器中，打开**“应用程序”**，然后运行应用程序发布器。
4. 更改应用程序的**“软件包名”**。例如，将 **Dairy** 改为 **Dairy_2**。

注意：如果您要保留**软件包名**不变，而只改变文件名，请在**“InTouch 应用程序发布器 – 第四步/共四步”**对话框中更改文件名。

要以多种分辨率发布相同应用程序，您**必须**更改文件名。如果文件名相同，它会覆盖原来的应用程序。

当发布到 Web 服务器时，应用程序显示如下：

描述	Dairy Processing Application
发布者	Navin Johnson
文件名……	Dairy.exe / Video Resolution…(1024x768)

描述	Dairy Processing Application
发布者	Navin Johnson
文件名……	Dairy_2.exe / Video Resolution…(800x600)

附 录 A

InTouch Windows NT 服务概览

服务指的是 Windows NT 中的一个进程，它无需用户干预就可执行特定的自动后台系统功能。您可以配置服务在每次应用程序所在的计算机启动时自动启动。此外，服务在启动时不会影响 Windows NT 的安全系统。本附录将讨论 InTouch（7.1 或以上版本）的 Windows NT 服务。

InTouch 具有以下四种 Windows NT 服务：

InTouch 服务	**Windows NT 服务缺省路径**
Wonderware Logger	C:\Program files\Factorysuite\Common\Wwlogsvc.exe
Wonderware SuiteLink	C:\Program files\Factorysuite\Common\Slssvc.exe
Wonderware NetDDE Helper	C:\Program files\Factorysuite\Common\Wwnetdde.exe
Wonderware WindowViewer	C:\Program files\Factorysuite\Intouch\View.exe

注意：有关其它 FactorySuite 组件的 Windows NT 服务的详细信息，请参阅*《Wonderware FactorySuite 管理员指南》*。

为什么要使用 Windows NT 服务？

InTouch Windows NT 服务可提供下列功能：

操作员登录和注销。

在正常工厂运行期间，操作员通常要在换班时登录和注销 Windows NT 操作系统。当 WindowViewer 作为 Windows NT 服务运行时，即使在不同操作员登录和注销的情况下，它仍可以继续记录历史数据、收集报警信息、处理脚本、充当 I/O 服务器，以及作为 I/O 客户端来写入数值。

断电损失。

在供电中断的情况下，工厂操作员通常要在工厂恢复运行之前等待相当一段时间。大多数的灾难恢复计划均要求在恢复供电后，立即启动基本计算机系统。Microsoft Windows NT 工作站和 Windows NT 服务器可以在电源恢复后自动重新启动。由于 InTouch 组件软件是作为一组 Windows NT 服务运行的，因此您的关键系统可以在不影响 Windows NT 安全性的前提下立即开始运行。

持续运行。

现代工厂视资源效率为首要考虑因素。为获得最高水平的生产能力，许多工厂都使用持续运行环境。当 InTouch 组件作为 Windows NT 服务运行时，工厂可以保证一周七天、一天 24 小时的连续作业。

在桌面和系统环境下运行 Windows NT 服务

当您启动 Windows NT 操作系统时，配置为自动启动的服务在“后台”启动，而不会在桌面上显示图形用户界面 (GUI)。这种情况下的服务是在*系统环境*下运行的。当操作员或其他用户登录到系统后，任何在系统环境下运行的具有关联 GUI 的服务会在用户登录时自动显示在桌面上。此时，服务是在*桌面环境*下运行的。

因此，如果您配置 WindowViewer 服务为自动启动，则每当 Windows NT 系统启动时，服务会立即在系统环境下运行。当用户登录到系统后，WindowViewer 服务继续运行，但此时是在桌面环境下运行（WindowViewer 的 GUI 会自动显示）。

注意：如果您在定义 InTouch 访问名时启用**“只提示激活项”**选项，并且定义了只在某些 InTouch 应用程序窗口中激活的 I/O 标记名（标记名不用在应用程序的其他地方），则可以撤消激活这些标记名。一旦您注销系统，WindowViewer 虽然仍在运行，但不会显示 GUI，因此无法使用键盘、鼠标或其它指针设备来浏览应用程序。但是，如果您从 QuickScript 关闭应用程序窗口，窗口会自动从内存卸载，从而终止与这些标记名的链接。

创建主用户帐号

当 Windows NT 操作系统启动服务时，它将在一个称为*本地系统帐号*的用户环境下运行。配置为自动启动的每个 Windows NT 服务均在此本地系统帐号下运行。当您使用自己的帐号登录到 Windows NT 操作系统并启动应用程序时，则系统将在您的用户帐号下运行。

本地系统帐号在 Windows NT 安全系统中具有很少的权限，因而不能保证登录到系统的用户有权运行 InTouch 服务。这包括访问文件、与其它程序和服务交换数据以及修改注册表等操作。InTouch 要求用户帐号在 Windows NT 安全系统中具有比本地系统帐号更高的访问级别，从而使服务可以执行这些类型的操作。

要实现这一点，您必须创建一个 InTouch“**主**”**用户帐号**（建议用户帐号名为 "wwservices"）。此主帐号应具有本地计算机的管理权限，以便有足够的权限来启动 InTouch 服务。随后在 InTouch 安装期间，当 FactorySuite“**公用组件**”对话框出现时，您可以输入本地计算机的节点名（不是域名），以及主帐号的用户名和口令。

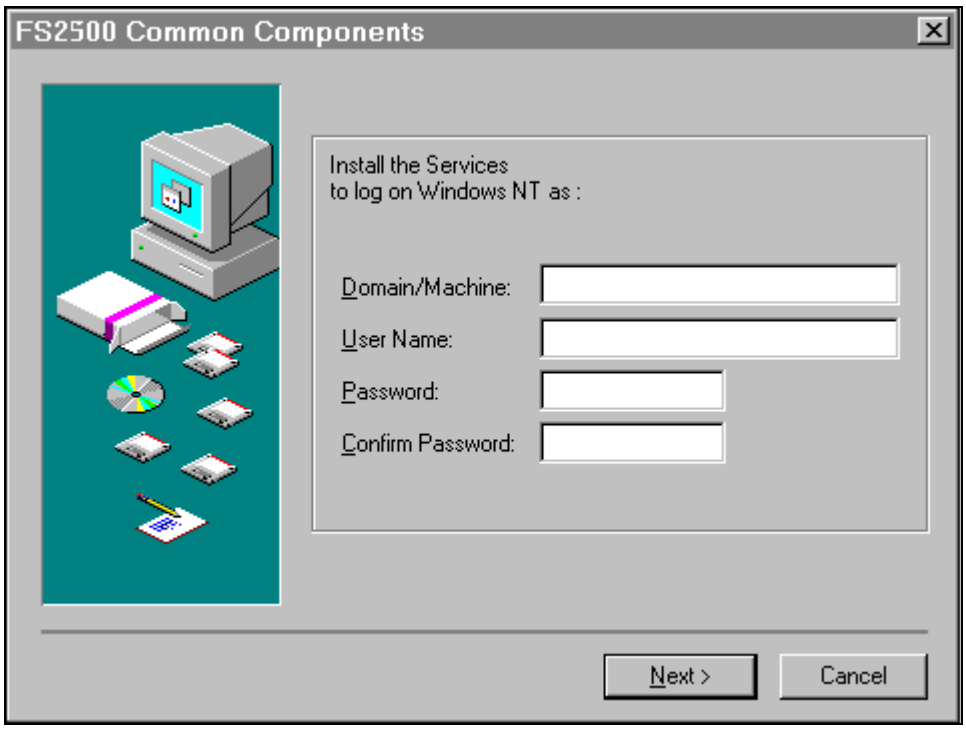
注意：要对主帐号进行任何更改，必须在安装 InTouch 期间进行或通过运行 Wonderware Service User 程序 WWUSER.EXE 来完成。

在设置主帐号、登录到计算机并启动 InTouch 组件（例如 WindowViewer）后，该组件的服务即在主帐号而不是用户帐号下运行。这就确保了您具有必要的权限来启动 InTouch 服务。

注意：在讨论 Windows NT 服务时，可能使用*模拟帐号*一词而不是主帐号。模拟帐号是提供对您的站点或服务器的限制资源“区域”访问的用户/用户组帐号。在 Windows NT 中，它称为“NT 用户/用户组帐号”，必须由系统管理员设置。

配置系统权限

在 InTouch 安装期间，系统会提示您为管理帐号输入用户名和口令。此信息通常用于设置您的 NT 用户模拟帐户。Wonderware 服务，如 Wonderware NetDDE Helper 和 Wonderware WindowViewer，使用此信息来自动登录，并且在系统自动启动期间自动启动相应的服务。



➢ **要配置系统权限：**

1. 在“**Domain/Machine**”（域/机器）框中，输入您的系统域名或节点名。
2. 在“**User Name**”（用户名）框中，输入您的用户标识。
3. 在“**Password**”（口令）框中，输入您的系统口令。

注意：您提供的**用户名**和**口令**必须是通过“NT 用户管理员”配置的有效管理员级别登录帐号。

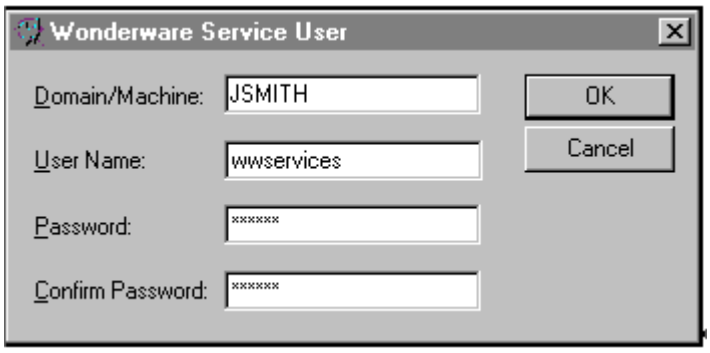
---

4. 在“**Confirm Password**”（确认口令）框中，重新输入系统口令以进行确认。

   安装后，如果您需要修改此信息，请运行安装目录下的 Wonderware Service User 程序 (WWUSER.EXE)。例如，\Program Files\FactorySuite\Common。

如果您已安装 InTouch 但需要进行修改以使用新的“主”用户帐号，请参考下面有关如何设置新主帐号的步骤。

- **要设置 FactorySuite 服务以使用新的主帐号：**运行 Wonderware Service User 程序（缺省条件下，wwuser.exe 位于目录 C:\Program Files\Factorysuite\Common 下）。**“Wonderware Service User”**对话框出现：



注意：您在此处输入的信息必须与当前主帐号相符。

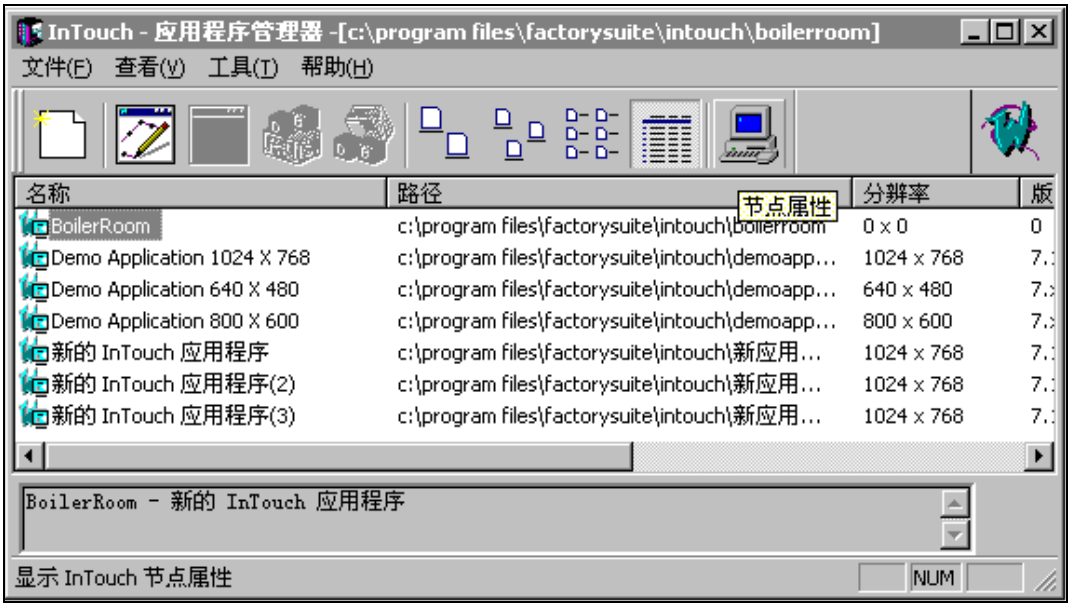
2. 在**“Domain/Machine”**（域/机器）框中，输入机器名。
3. 在**“User Name”**（用户名）框中，输入新的主帐号名。
4. 在**“Password”**（口令）框中，输入口令。
5. 单击**“OK”**（确定）。

配置 InTouch 服务

InTouch 服务，包括 **Wonderware Logger**、**Wonderware SuiteLink** 和 **Wonderware NetDDE Helper** 等，均在 FactorySuite 安装期间进行安装并配置为自动启动。但是，您必须配置 **WindowViewer** 作为 Windows NT 服务自动启动。

➢ **要配置 WindowViewer 作为 NT 服务启动:**

1. 启动 InTouch 程序 (intouch.exe)。“**InTouch 应用程序管理器**”对话框出现：



2. 单击“**节点属性**”工具，或者在“**工具**”菜单上，单击“**节点属性**”。“**节点配置**”对话框出现并显示“**应用程序开发**”属性页：

   要快速访问“**节点属性**”对话框，右击显示窗口的空白区域，然后单击“**节点属性**”。

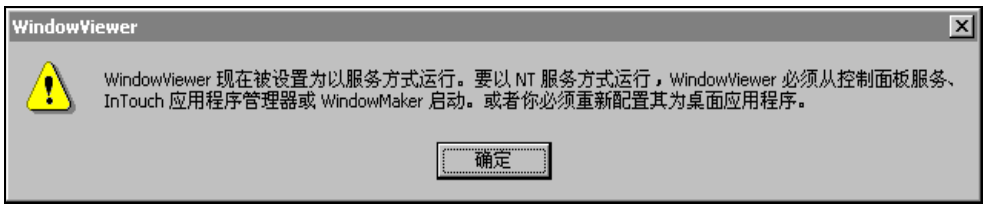


3. 选择“**以 NT 服务方式启动 WindowViewer**”选项。

注意：当 WindowViewer 作为 Windows NT 服务运行时，网络应用程序开发是禁用的。

4. 单击“**确定**”。

注意：如果 WindowViewer 设置为 NT 服务并随后直接启动（从其图标或 Windows 开始菜单等），则在 WindowViewer 显示窗口之前约有 15 秒的延迟。此延迟是因为 WindowViewer 尝试连接到 NT 服务控制管理器而造成的。如果未能连接到服务控制管理器，WindowViewer 会显示下面的消息框：



如果单击“**是**”，WindowViewer 将作为应用程序，而不是 NT 服务来启动。如果单击“**否**”，用于启动 WindowViewer 的命令将被取消。

您可以使用其它一些方法来让 WindowViewer 作为 NT 服务启动：

1. 在 WindowMaker 中，单击菜单栏右上角的“**运行！**”快速切换开关。
2. 从 InTouch 应用程序管理器启动 WindowViewer。
3. 要在命令提示符下启动 WindowViewer 服务，请使用下面的命令：

   Net Start View

4. 要在命令提示符下停止 WindowViewer 服务，请使用下面的命令：

   `Net Stop View`

警告：如果 WindowViewer 配置为以 NT 服务方式自动启动，则当系统重新启动时，上次由 InTouch 应用程序管理器引用的 InTouch 应用程序会自动启动。但是，如果您当前正编辑多个 InTouch 应用程序，则当掉电或重新启动系统后，可能会启动错误的应用程序。

➢ **要停止运行 WindowViewer 服务：**

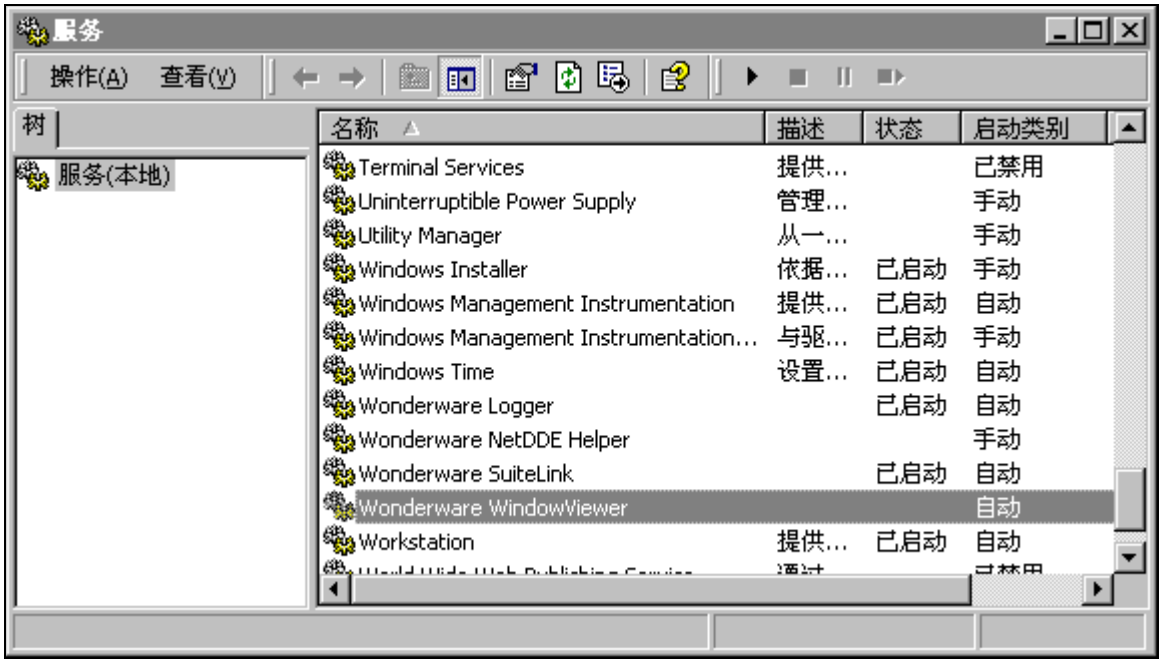
通过在“**节点属性**”对话框中关闭“**以 NT 服务方式启动 WindowViewer**”选项，停止 WindowViewer 作为 NT 服务运行。

您可以通过关闭计算机来停止 WindowViewer 或任何其它服务，也可以通过 Windows 控制面板来停止（或启动）任何服务。

➢ **要通过控制面板停止/启动 NT 服务：**

1. 打开控制面板，双击“**服务**”图标。

2. “**服务**”对话框出现：



3. 在服务列表上辉亮显示您要停止（或启动）的服务，单击“**停止**”或“**启动**”，然后单击“**关闭**”。

手动安装/删除 NT 服务

通常，InTouch 的 Windows NT 服务在 FactorySuite 安装过程中自动安装，而通过双击控制面板中的“**添加/删除程序**”图标自动删除。但是，如果您需要手动安装或删除 InTouch Windows NT 服务（例如 Wonderware WindowViewer），可以执行以下步骤。

- **要安装 Windows NT 服务：**

1. 从“开始”菜单选择“**运行**”，或者打开一个 MS-DOS 命令提示符窗口。
2. 输入下面的命令：

```
"C:\Program Files\FactorySuite\InTouch\view.exe" -
install
```

- **要删除 Windows NT 服务：**

1. 从“开始”菜单选择“**运行**”，或者打开一个 MS-DOS 命令提示符窗口。
2. 输入下面的命令：

```
"C:\Program Files\FactorySuite\InTouch\view.exe" –
remove
```

注意：这两个命令是安装和删除 Wonderware WindowViewer 服务的一个示例。对于其它的 InTouch 服务，路径名将包括目录 **\Common** 而不是 \InTouch。

如果您在“**运行**”对话框中输入命令，请确保将命令用双引号 (" ") 括起。例如 "Program Files"。如果您删除服务，Windows NT 会先停止该服务，然后再删除它。删除服务不会删除应用程序的 .exe 文件。

配置 NT 服务的启动选项

➢ **要配置 InTouch 服务的启动选项：**

1. 打开 Windows 控制面板。
2. 双击“**服务**”图标，“**服务**”对话框出现。
3. 从“服务”列表中选择所需的 InTouch 服务，然后单击“**启动**”。
4. “**服务**”对话框再次出现：



5. 在“**启动类型**”组中，选择您要使用的选项，然后单击“**确定**”。

自动	当 Windows NT 重新启动时，服务自动启动而无需用户干预。
手动	用户或应用程序进程必须明确启动服务。
禁用	禁止启动服务（此选项通常保留供排除故障使用）。

注意：Wonderware 建议您为所有 InTouch 服务使用缺省启动选项。

InTouch 服务之间的相关性

当安装服务时，程序会向 Windows NT 操作系统提供一份相关性列表。如果某项服务要在其它服务启动之后才能启动，Windows NT 会检查以确保其它服务已经运行，然后再启动该项服务。取决于运行 WindowViewer 的要求，您应该了解以下相关性。

如果您计划使用分布式报警或分布式历史，或者需要访问 Network DDE 数据，则必须运行 Wonderware NetDDE Helper 服务。Wonderware NetDDE Helper 服务取决于当前安装的 Network DDE 和 Network DDE DSDM 服务是配置为手动还是自动启动。在安装过程中，Wonderware NetDDE Helper 服务配置为手动启动，这意味着 WindowViewer 会在启动时自动启动此服务。

如果您需要 WindowViewer 充当 SuiteLink 服务器或客户端，则必须运行 Wonderware SuiteLink 服务。Wonderware SuiteLink 服务还要求已安装 Microsoft TCP/IP。

如果您要在运行 WindowViewer 时存储任何消息或错误，您必须确保已安装 Wonderware Logger 服务。在 FactorySuite 公用组件安装期间，必须安装 Wonderware SuiteLink 和 Wonderware Logger 服务并将其配置为**自动**启动运行。

有关详细信息，请参阅“配置 NT 服务的启动选项”。

注销 Windows NT

当您注销 Windows NT 操作系统时，它会发送控制注销消息 **wm_close** 和 **wm_terminate**，以关闭当前在桌面环境下运行的所有应用程序。缺省条件下，Microsoft Network DDE 安装并配置为“监听”这些注销消息。如果收到任何消息，Microsoft Network DDE 会自动终止当前网络连接，并且在您注销系统时，Microsoft Network DDE 不允许建立任何新连接。

但是，**Wonderware NetDDE Helper** 服务也会监听和捕获 Microsoft Network DDE 的注销消息，以保持 Microsoft Network DDE 连接。因此，当您注销时，Microsoft Network DDE 允许建立新连接。

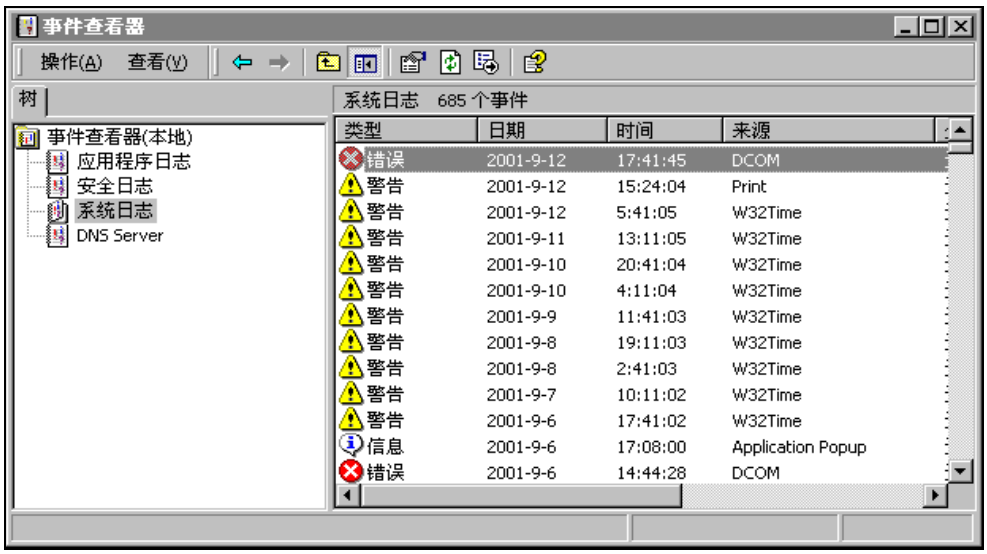
有关详细信息，请参阅“在桌面和系统环境下运行 Windows NT 服务”。

InTouch 服务疑难解答

以下是 Wonderware 技术支持在安装或运行 InTouch 服务当中遇到的一些常见问题和解决方法：

错误“一个或多个服务无法启动...”

如果您遇到此错误，请从“开始”菜单中，选择“**程序/管理工具（公用）/事件查看器**”。“**事件查看器 – 系统日志**”对话框出现，显示在启动任何 Windows NT 服务期间出现的参考消息、警告或错误列表。



您可以查看因 InTouch 服务无法启动而生成的任何警告或错误消息。如果“**事件查看器**”指明 Wonderware WindowViewer 服务无法启动，则大多数是因为该服务所依赖的其它 InTouch 服务尚未运行。

有关详细信息，请参阅“InTouch 服务之间的相关性”。

在配置 Wonderware WindowViewer 服务的运行选项后，必须将其作为服务来启动和停止。

如果您从快捷方式图标或通过从“开始”菜单选择“**程序/Wonderware FactorySuite/InTouch WindowViewer**”来启动 WindowViewer，WindowViewer 将暂停约 15 秒，然后才显示下面的消息：



单击“**是**”以让 Wonderware WindowViewer 服务作为您的用户帐号下的桌面应用程序启动，并显示其 GUI。

警告：如果选择“**是**”以将 WindowViewer 作为桌面应用程序运行，则当您注销系统时，WindowViewer 会关闭。由于它原来不是作为服务启动的，因而不会继续作为服务运行。这意味着在您注销后，InTouch 应用程序不会继续运行并收集历史数据、显示报警或处理脚本。

要使 InTouch 应用程序能够继续作为服务运行，您需要重新将 WindowViewer 配置为以 NT 服务方式启动。

单击“**否**”以取消操作，WindowViewer 应用程序将不会启动。

有关详细信息，请参阅“以 NT 服务方式停止和启动 WindowViewer”。

InTouch 服务无法安装或启动

如果在安装 InTouch 后无法安装或启动 InTouch 服务，请执行以下步骤：

1. 启动 Windows NT 用户管理员窗口，并创建一个新的主用户帐号。

   注意：要以服务方式启动 InTouch 组件，此用户帐号必须具有本地计算机的管理权限。如果您在域名列表中不能看到计算机的节点名，请手动输入节点名。

   有关详细信息，请参阅“创建主用户帐号”。

2. 确认您的计算机节点名不超过 14 个字符。如果节点名中包含下划线字符 (_) 或短划线 (-)，请将节点名改为不含下划线或短划线的等于或少于 14 个字符的名称。
3. 在安装期间如果您收到输入域名的提示，请输入计算机的节点名而不要输入域名。然后输入在步骤 1 中创建的用户名和口令。
4. 如果已经安装 InTouch，您仍可以通过从目录 C:\Program Files\Factorysuite\Common 运行 Wonderware Service User 程序 wwuser.exe，来指定域名、用户名和口令。
5. 重新启动计算机。
6. 使用任何有效帐号登录到网域。即使您的网域已经关闭，它也不会影响到您在本地计算机上运行的 InTouch 应用程序。

InTouch 服务的注册表项

所有四个 InTouch 服务均在 Windows NT 注册表中作为注册表项列出，如下所示：

InTouch 服务注册表项

Wonderware SuiteLink
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SLS

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\slssvc

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SuiteLink

Wonderware Logger
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WWLOGSVC

Wonderware NetDDE Helper
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WWNetDDE

Wonderware WindowViewer
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VIEW

词汇表

快捷键

应用程序在创建键盘接口时使用快捷键，通常作为使用菜单来指示选项的一种替代方法。快捷键是对应用程序具有特殊含义并可用于生成命令消息的击键。

访问

数据采集；寻找所需数据。

活动应用程序

其窗口当前处于键盘中心的应用程序。要接收和处理消息，应用程序可以不必是活动应用程序。当应用程序取得或失去“活动应用程序”状态时，应用程序将接获通知消息。通常由用户决定活动应用程序，但应用程序可以改写此决定。

ActiveX

松散定义的技术集合，允许软件组件彼此间相互作用。

ActiveX 控件/容器

ActiveX 控件的前身为 OLE 控件或 OCX，是以标准方式执行特定功能的独立软件组件，为可重复使用的组件定义标准接口。ActiveX 控件不是独立的应用程序，而是置入控制容器内的服务器。要使用 ActiveX 控件，必须将其置于 ActiveX 容器内。InTouch 即是一种 ActiveX 容器。VisualBasic 和 Internet 浏览器也是 ActiveX 容器。

报警 在出现严重偏离正常条件的情况下，所显示或激活的警告信号。

算法 机械执行以完成特定程序的一系列指令。

模拟 指通过对连续物理量的测量来表示数量。

应用程序 用于执行特定工作的程序或程序组，例如 InTouch。

应用程序发布器 InTouch 应用程序发布器可以创建一个自解压文件，该文件包含在另一 InTouch 节点上安装应用程序所需的所有相关文件和安装程序。应用程序发布器通过压缩应用程来最大程度地减少在 Internet 上的下载时间。

参数 可赋予逻辑或数字值的一个变量。InTouch QuickFunction 最多可指定 16 个参数。参见 **QuickFunction**。

赋值操作符 在赋值语句中，操作符用于将其右端的值赋给其左端的变量。

赋值语句 编程语言的语句，将一个值赋给一个变量，例如 *x = x + 1* *或 y = 6*。

星号 星号 (*) 在许多编程语言中用来表示乘法操作符。

异步 传输数据的一种方法，这种方法对于传输期间字符位置的定时是可变的。参见“**异步传输**”。

b 字节或波特的缩写。存储场合下一般用“位”；通讯场合下一般用波特率。Kb = 1000 字节或波特（技术上而言 1K = 1024 字节）。参见“**波特**”或“**字节**”。

后台/背景

在多任务程序设计中，执行低优先级程序的一种工作环境；显示屏中未被显示文字或图形（前景）覆盖的部分。

备份

建立指定文件的备份，将其从一个软盘或硬盘传送到另一可移动或固定的磁盘。

带宽

网络上通信通路的传输能力，以每秒兆位 (Mbps) 表示。例如，Ethernet 的带宽为 10 Mbps；高速 Ethernet 的带宽为 100 Mbps。

波特率

测量数据传输速率的单位。一波特等于每秒一位。因为单个字符需用八位来表示，因而将波特率除以 8 便得到每秒传输的字符数 (cps)。例如，300 波特相当于 37.5 cps， 1200 波特相当于 150 cps，2400 波特相当于 300cps 等。

Beta 测试

由选定的“典型”用户对软件或硬件产品进行预测试，以便在产品投放到市场以前发现缺陷。

二进制

一种以 2 为基数的数字系统；或某一包含仅有两种可能的选项或条件的特征或属性。

二进制码

通过使用位 - 即 0 或 1 来对数据编码的编码系统。

二-十进制(BCD 码)

用一组 4 位二进制 **1** 和 **0** 来表示十进制数字的计算机编码系统。

BIOS

Basic **I**nput/**O**utput **S**ystem（基本输入/输出系统）的字首缩写。在一些操作系统中，指根据特定计算机进行自定义的程序部分。

位

二进制数；数字的二进制表述中的一位（1 或 0），是计算机或相关设备所能识别的最小信息单位。一个字节或计算机字由若干个位组成。

位图

部分显示设备表面的内存映像。在 Windows 中，位图实际上是包含指向该内存映像的指针的数据结构，外加显示设备的信息。位图所需的内存量与设备有关，由颜色能力和设备象素分辨率决定。

启动

从存储设备中读取指令到计算机内存，以启动或重启动计算机系统。这涉及到加载部分操作系统到计算机主内存。如果计算机已打开，称为“热启动”；否则，称为“冷启动”。

边界

包围当前活动窗口的线段。当边界上显示双向箭头时，该窗口的大小可以通过拖拉边界来调整。

缓冲区 存储从一个设备传送到另一设备的数据所需的临时存储空间。缓冲区用于补偿硬件处理数据时的不同速率；例如，用于存储等待打印的文件，以释放 CPU 给其它任务，因为它处理数据的速率更快。

总线 传送数据的通道或路径。

按钮 出现在对话框中的大圆边矩形按钮或小圆形按钮。用光标箭头单击可选择按钮的选项或命令。

字节 计算机将一组相邻的二进制数作为一个单位来运行。通常字节由 8 位二进制位组成。

CD-ROM 光盘只读存储器；一种可以容纳超过 650 MB 数据的光盘格式。

复选框 出现于对话框中的可打开或关闭的小矩形框。复选框通常与可设置的多个选项关联。要设置复选框选项，移动到复选框并单击鼠标按钮。当出现 **X** 时，复选框即被选中。如为空白，则未被选中。

客户端 与服务器交换数据的机器；与服务器共享数据的软件程序。

客户端/服务器 一个客户机依赖于服务器运行的网络。

剪贴板 存储要复制或移动到另一应用程序或窗口的数据（文本、位图、图形对象等）的空间。

关闭 从屏幕上移除应用程序窗口和图标并释放应用程序所用内存。要关闭应用程序，选择*控制/关闭*命令。应用程序一旦关闭，必须重新运行才能再次使用。

命令 执行特定动作的字或词组，通常出现在菜单中。

命令按钮 带有标签的圆角矩形，其标签描述某特定动作，例如“**确定**”、“**取消**”或“**关闭**”。一旦选取，命令按钮即可执行动作。

命令键 用于执行不同功能的键盘键。

COM 端口 个人计算机上的串行通讯端口。

命令行 跟在任意 MS-DOS 命令后面的参数字符串，包括初始化应用程序的命令。命令行中的参数会传递给 MS-DOS 函数或启动时的程序。

计算机图形 概指图形或图表外观的一般术语，与字母及数字不同，它显示在显示屏或硬拷贝到输出设备上。

串联 将两个或多个字符链结成一个字符串，或将一行显示内容与下一行相连。

CONFIG.SYS 当系统启动或重启动时由 MS-DOS 处理的 ASCII 文本文件。它允许用户配置操作系统的某些方面，例如内置磁盘缓冲器数、可同时打开的文件数等。

控制台 在服务器上运行的管理进程。

控件名称 参见“**Windows 控件**”。

裁剪 在计算机图形中，切除图像的某些部分。

CSV 剪贴板使用逗号分隔变量在应用程序间传送文本栏和数据。CSV 数据项如同用逗号分隔的变量文本。虽然 Microsoft Excel 是创建 CSV 剪贴板数据的主要程序，但许多 DOS 应用程序都支持这种格式。

当前文件 在应用程序中运行的文件。

数据库 逻辑上相关的记录或文件的集合。数据库将许多记录合并成数据记录的公共集合并将之作为单个核心文件使用。

缺省 系统自动选取某一选项、命令或设备。例如，当打开对话框时，对话框的某个命令按钮已被选取。这表明它是缺省值，如按 **<Enter>** 键，它将被自动选取。通过选择另一相应选项、命令或设备可改写缺省值。

设备驱动程序 控制计算机与打印机、显示器或鼠标等设备的相互作用的程序。设备驱动程序使计算机能够使用设备。

DHCP 动态主机配置协议。TCP/IP 协议用于自动分配 IP 地址以及为网络客户端配置 TCP/IP。

对话框 当执行某一命令前 Windows 需要进一步的信息时所出现的窗口。例如，如果选定“**文件**”菜单上的“**保存**”命令，将出现一个对话框，询问用以保存文件的名称。

目录 将文件分组以便存取的结构。目录如同显示文件位置的地址。目录可以包含文件或文件的子目录。

离散值 仅有两种状态的变量：'1'（真、开）或 '0'（假、关）。

磁盘操作系统 (DOS) 是一种操作系统程序存储在磁盘的操作系统。通常用于跟踪文件、存储和检索文件、分配存储空间，以及管理与磁盘存储有关的其它控制功能。

显示 数据在屏幕上的物理表示。

混合 使用不同彩色点的并列组合，可形成另一单色的错觉。

文档 一个必须连续打印的打印机输出单位，即其它输出不会穿插在该文件中。文档也称为报告。应用程序必须指示每个文档的开头和结尾。

域 在 Windows 2000 中，指共享安全策略及用户帐号数据库的一组计算机。Windows 2000 域与互联网域名不同。

域名 在 Active Directory 中，指为共享公共目录的网络计算机集合指定的名称；在互联网中，指唯一确定主机的文本名称。一台机器可以具有多个域名，但给定的域名只能指向一台机器。域名通过 DNS 域名服务器解析为 IP 地址。

域名系统 (DNS)	将域名解析为 IP 地址的 TCP/IP 网络（包括互联网）服务。这使得用户在查询远程系统时可以采用易于记忆的名称如 FinanceServer 或 Ourbusiness.com，而不必使用象 123.45.67.89 这样的 IP 地址。
DRA	动态地址引用，允许通过点域在运行时动态地控制 I/O 标记名的访问名和项名。它还允许您通过 QuickScript 函数来控制访问名属性。
DRC	动态分辨率转换，它使每个视图节点能够按用户定义的选项来缩放应用程序，包括自定义分辨率。定义分辨率可在 WindowViewer 编译应用程序时进行，而不需要 WindowMaker。
驱动器	A-Z 范围内的一个字母，后跟冒号(:) 来指示逻辑磁盘驱动器。
动态数据交换(DDE)	DDE 是应用程序之间的数据通道，无须用户介入或监视就能完成。在 Windows 环境中，DDE 通过一组消息类型、处理这些消息类型的推荐程序（协议），以及某些新定义的数据类型来实现。按照协议，独立编写的应用程序可以相互之间传递数据，而无须用户的介入。例如 InTouch 和 Excel。
动态主机配置协议	参见 **DHCP**。
ENTER 键	键盘上执行某一语句或命令的键。某些键盘上使用 RETURN 键。

企业

用于概述所有业务经营单位，包括所有远程办公室和分支机构的一个术语。

Ethernet

1976 年由施乐公司开发的一个局域网协议。Ethernet 支持 10 Mbps 的数据传输速率，使用总线拓朴及粗细不等的同轴光纤双铰电缆进行传输。新发展的 Ethernet 称为高速 Ethernet，支持高达 100 Mbps 的数据传输速率，而最新的千兆位 Ethernet 更支持 1000 Mbps 的数据传输率。

事件

事件与 ActiveX 控件关联并通过 ActiveX 容器发生。您可以设计某个动作并通过创建 ActiveX 事件脚本将其与事件关联，以便在运行时 (WindowViewer) 执行 ActiveX 控件事件。例如，**Control.click (shift)**. **FileViewer.DoubleClick (name)**。参见“**属性**”和“**方法**”。

表达式

描述数字、带操作符的数字、变量及上述类型的组合的一般术语：6、3+6、n+10 等都是表达式。

扩展

从窗口中选取多个项目。要选取多个项目，按住 SHIFT 键直至选取所有项目。

扩展名

文件名结尾的句点及三个字母。扩展名确定文件所包含的信息类型。例如，扩展名为 .EXE、.BAT 表明该文件包含一个应用程序。

FactorySuite	Wonderware 的软件包，包括 InTouch（及其所有附加程序和公用程序）、InControl 及其 I/O Server、IndustrialSQL Server、FactorySuite Web Server、其它许多 I/O Server 程序、Productivity Pack、NetDDE for Windows 以及 NetDDE Extensions for Windows NT。
高速 Ethernet	参见 **Ethernet**。
文件	将信息保存或存储到硬盘或软盘以备日后使用的一种机制。文件也指由用户创建的任何文档或数据库，如字处理文档、电子表格等。每个文件显示在其各自的窗口中，在大多数情况下，文件名会显示在窗口顶端的标题栏中。
文件名	文件名由不超过 8 个字符的基本名称和三个字符的扩展名组成，例如 INTOUCH.EXE。
格式化	准备磁盘以便存储信息的过程。格式化磁盘将擦除任何先前存储的数据。格式化是对象改编的术语。在大多数 Windows 应用程序中，可用的格式包括文本、位图等。
正式域名 (FQDN)	包括所有网域名称的域名，可回溯到根域以清楚地揭示域名空间树中的位置。正式域名的一个例子是 Sales.Europe.Wonderware.com。
千兆位 Ethernet	参见 **Ethernet**。
图形对象	视觉化对象，例如表示视觉界面的滚动栏、位图或图标。图形对象可以通过应用程序或应用程序所使用的 Windows 来创建。

帮助
解释如何使用 Windows 应用程序的在线说明。“帮助”菜单会显示特定的帮助主题。

辉亮
指示对象已被选定，并且将受下一个动作或命令的影响。辉亮选定的对象会反白显示。选定的图标呈白色外形并显示应用程序名。

HMI
人机界面；允许操作员控制制造过程的一个软件程序，也称为 MMI。例如 Wonderware InTouch 就是一个 HMI。

ICA
独立计算结构体系。由 Citrix Systems 公司开发的允许瘦客户机访问服务器的远程显示服务协议。

不活动
未被选取的窗口或图标。
参见**“选择”**。

插入点
当用户进行输入时用于插入文本的位置。插入点通常显示为一闪烁的垂直线（光标），常出现在工作区域或对话框中。输入的文本显示在插入点的左侧，当输入文本时，光标将被推向右侧。

整数
由正整数、负整数和零组成的数集的任意成员。例如：–59、-3、0。

互联网
全部使用 TCP/IP 进行互联的庞大网络群，由二十世纪六十年代晚期及七十年代早期的 ARPANET 发展而得。互联网将大约七万个独立网络连成一个全球网络。

I/O
Input/Output（输入/输出）的缩写。

IP（Internet 协议）
用于网络通讯的公共协议。

IP 编号或地址

以句点分隔的四部分数字（例如 123.45.67.89），唯一确定互联网上的一台机器。互联网上的每台机器都有一个唯一的 IP 编号；如果机器没有 IP 编号，则不能出现在互联网上。大多数机器还具有一个或多个域名，以方便人们记忆。

Java

网页中使用的类似于 C 和 C++ 的高级编程语言，提供制作独特网页的动画及其它高级特性。

键加速器

执行菜单命令的特殊键盘序列。例如，Ctrl + A。
参见“快捷键”。

列表框

对话框中的一个框，列出命令的所有可用选项。例如，磁盘上的文件名列表。通常在从列表框中选取项目后，需选取“确定”按钮。如果列表框中有多个可选项，则会出现一个垂直滚动条。选择列表中第一项旁边的向下箭头，将可显示列表框中的其它选项。

负载平衡

对服务器群集中的服务器进行配置，以平均分配各个服务器上的负载需求。

局域网 (LAN)

一组互相连接的电脑，通常彼此靠近（如处于同一幢建筑或某一建筑的同一层），以便可以在其间传递数据。

本地变量

您可以在脚本内声明本地变量来存储临时结果，并用临时脚本值来创建复杂的计算而不影响或减少授权标记名计数并可提高性能。在同一脚本内，本地变量或标记名可交替使用。

登录

进入计算机系统的行为，例如“登录到网络并收阅电子邮件”。

登录名 您用于访问计算机系统的帐号名。与口令不同，登录名是公开的。

宏 单一符号编程语言语句，可以转译成一组汇编语句。

最大化 使某个窗口或图标布满整个屏幕。要最大化窗口，选取*控制/最大化*命令，或单击窗口右上角的最大化框。参见“**最小化**”和“**恢复**”。

MB Megabyte（兆字节）的缩写；一百万字节；1000KB。

Megabyte 实际为 1,048,576 字节或 1024 千字节；或取约值一百万字节或一千千字节。

菜单 菜单是可用的 Windows 和应用程序命令组的列表。菜单标题出现在窗口顶部的菜单栏。要选取命令，可先显示菜单，然后选取所需的命令。

菜单栏 列出应用程序菜单名的水平栏。菜单栏显示在窗口标题栏的下方。虽然某些菜单（和命令）在许多应用程序中通用，但每个 Windows 应用程序都有其特殊的菜单栏。

消息框 应用程序藉以显示错误消息或其他重要信息的特别对话框。当错误发生或应用程序需要信息来完成特定动作或命令时，消息框将向用户发出警告。

方法 方法与 ActiveX 控件有关。它们类似于可从 ActiveX 容器调用的脚本函数。例如：

Browser.Navigate("URLPageName"), **Engine.start()**。参见“**属性**”和“**事件**”。

毫秒 千分之一秒，简写为 ms 或 msec。

最小化 将窗口转换成图标。要最小化窗口，选取*控制/最小化*命令，或单击窗口右上角的最小化框。参见“**最大化**”和“**恢复**”。

镜像 显示或建立与原始方向完全相反的图像。例如，按其 x 或 y 轴翻转图形。

模式 操作的方法或条件。

取模 取除法的余数的数学函数。数字 *x* 对 *n* 取余得到 *x/n* 的整数余数。例如，200 对 47 取余得到 200/47 的余数为 12。

MS/DOS **M**ICRO**S**OFT **D**ISK **O**PERATING **S**YSTEM（微软磁盘操作系统）的缩写，是由 IBM 个人计算机及兼容机使用的标准操作系统。由 Microsoft, Inc. 开发。

多任务 计算机同时操作两项或多项功能或任务的能力。

NAD 网络应用程序开发或 NAD 是一个结合基于客户机和基于服务器结构优点的结构。NAD 提供应用程序更新的自动通知，并将更新后的应用程序自动分发到运行节点。

域名空间	按照命名惯例定义的名称或名称组；解析指定名称的任意有限区域。互联网使用等级域名空间将域名划分成不同类别，称为顶级域名，如 .com、.edu 和 .gov。
NetBIOS 增强用户接口 (NetBEUI)	需要少量内存但不能路由的小型快速通讯协议。通过路由器链接的远程地点不能使用 NetBEUI 来进行通讯。
节点	包含与其下某个或多个项目的链接的树形结构上的位置。在局域网中，指可以与网络上的其它设备进行通讯的设备。在群集技术中，指运行作为群集一员的 Windows 2000 Advanced Server 的电脑。
NTFS 文件系统	Windows 2000 和 Windows NT 本地文件系统。支持长文件名、各种共享文件许可以及事务处理日志，如果操作系统损毁，允许完成任何未完成的文件相关任务。
对象	数据集。对象有几种格式：位图图像，文本、实时和历史趋势图等。
离线	与计算机的中央处理器不直接通讯的设备或装置；未连接到计算机的设备。
操作数	据以操作的数量或数据项。
操作系统	控制计算机程序执行的软件，可提供计划任务、调试、输入/输出控制、存储分配等。缩写为 OS。
操作符	在过程描述中，指示对操作数执行的动作。

选项按钮

显示在对话框中的一个小圆形按钮。通过选取选项按钮可设置选项，但在一组相关选项按钮中，只能选取一个按钮。当选项按钮被选取时，会显示一个黑点，如果未被选取，选项按钮显示为空白。

选项按钮组

对话框中的一组相关选项。一次只能从一组按钮中选取一个选项按钮。

数据包

通过网络发送的信息的基本单位。每个数据包包含目标地址、发送者地址、错误控制信息和数据。数据包的大小和信息取决于所使用的协议类型。

页面

页面指文件中选定和存储的信息块。例如，Microsoft Word 的一段文字可以作为一页，Microsoft Excel 的图表也可以作为一页。在同一文件中，页面的格式可以不同。页面添加到文件中时将被编号。

调色板

计算机图形系统中的可用颜色集。

奇偶校验

在字节、字符或字中添加一位，以根据系统逻辑，确保总有一个奇数位或偶数位。如果因硬件失败而导致传输过程中某个位丢失，可以通过奇偶校验检测到丢失的位。只要字节、字符或字的内容不变，这种位组合就存在。

粘贴

将某些对象从剪贴板插入到文档或文件。某些应用程序（包括 InTouch）可能包含执行此任务的剪贴命令。如果在窗口中运行其它标准应用程序，Windows 会将粘贴命令添加到控制菜单中。

路径

藉以查找特定文件的文件层次结构。如果在当前或指定的驱动器或目录下未找到程序或批处理文件，系统将指定一个或多个要按顺序搜索的磁盘驱动器和（或）目录路径。系统按路径中显示的顺序搜索驱动器和（或）目录路径。

路径名

系统内目录或文件的位置描述。路径名包括驱动器名、冒号(:)，后跟目录或子目录名，然后再跟文件名。各个名称与前面的名称之间用反斜杠 (\) 隔开。如果未指定路径名，将使用缺省驱动器和目录。

象素

图形单元格。“图形元素”的简称。视觉显示屏按行和列分成极小的点、方形或单元格，每个元素都是一个象素；可储存或显示的显示屏幕网格的最小单位。计算图形通常由矩形象素方阵组成。图形的分辨率通常用显示屏中的象素数表示。例如，560 x 720 象素的图形要比 275 x 400 象素的图形清晰得多。

插入

用于将某个值置于（插入）计算机存储器的指定位置的指令。

轮询

某些计算机/终端系统使用的通讯控制方式，计算机通过它依次询问连接到公共传输媒介的许多设备，以确定是否有要传递的信息。

端口

外围设备藉以通讯的计算机部分；CPU 与外围设备之间的连接。

优先权	规定表达式中哪个操作符需首先执行的规则。
过程控制	使用计算机来控制工业过程，例如精炼石油和钢铁生产。
过程控制计算机	过程控制系统中使用的计算机，通常在指令容量、字长和精确度方面受到限制。设计用于非空调环境下的连续作业。
处理	当前控制处理器的应用程序。一旦收到消息，应用程序将取得处理器控制权。它一直保留处理器控制权直到消息处理完毕。
属性	属性与 ActiveX 控件关联并能与 InTouch 标记名相关联。特定 ActiveX 控件的可配置属性由 ActiveX 控件的设计者决定。某些属性是单方向的，意味着属性决定标记名的值，或标记名的值决定属性。而其它属性为双向的，意味着值可通过标记名或属性进行设置。
协议	控制计算机系统或应用程序之间的信息交换的一组规则或约定。
已分布的应用程序	在服务器之间平等共享的服务器群集中的应用程序。
队列	等待计算机执行的一组项目。项目的安排决定处理的优先级。例如，等待打印的文件。

QuickFunction	QuickFunction 指可以从其它脚本或表达式编写及调用的脚本。每个 QuickFunction 最多可用 16 个参数。它们存储于创建它的应用程序中。QuickFunction 可定义为异步函数，即当执行时在主要 WindowViewer（运行时）过程的后台中运行。
QuickScript	InTouch 内创建的脚本。InTouch QuickScript 功能允许您在指定条件满足时执行命令和逻辑操作。例如，按下键、打开窗口，改变值等。
RAM	随机存储器。计算机的主要存储空间。
寄存器	中央处理器使用的高速设备，用于临时存储处理期间的临时结果或少量数据。
远程桌面协议 (RDP)	随 Windows 终端服务一起安装的缺省连接协议。RDP 允许为每个服务器配置各种不同的设置。客户端通常控制大部分的设置。
远程标记名	存在于远程标记源内但被本地 InTouch 应用程序引用的标记名。通过远程标记名引用，无需使用本地标记名字典中的任何标记名就可以设计客户应用程序。
恢复	双击可将图标恢复成最大尺寸窗口。要恢复窗口，从控制菜单选择恢复命令，或单击窗口右上方的恢复框。参见“**最大化**”和“**最小化**”。

保留标记名 保留其当前值并在每次系统出现故障时使用该值作为重启后的初始值的标记名。

运行 启动应用程序。运行命令可指定应用程序的参数。通过双击其名称或图标也可以运行应用程序。

运行中 “运行中”的应用程序是在系统内作为任务“运行”的应用程序，可以接收消息并且通常为用户所知。从初始化到终止，应用程序始终处于运行状态，但不总是在*进行处理*。参见“**处理**”。

运行时 控制单元检索数据及在数学逻辑单元中执行实际处理的时间，也指程序执行的时间。

保存 将文件或对文件所做的更改保存到磁盘上。

缩放 改变图像大小的过程。

滚动 上下或左右移动数据或文本以查看屏幕未显示的部分。

滚动条 在窗口底部或右边出现的栏。使用滚动条移动窗口，可以查看一个屏幕不能全部显示的更多信息。窗口右边的滚动条可以垂直滚动；窗口底部的滚动条可以水平滚动。

滚动框

滚动条中的小白框。滚动框反映窗口内的信息相对整个文件内容所处的位置。例如，如果滚动框在滚动条的中部，那么窗口中的文本或数据即处于文件的中部。可以通过鼠标拖动滚动条内的滚动框来翻阅文件。参见“**滚动条**”。

串行端口

计算机中的输入/输出端口，通过串行端口一次只能传输或接收一位的数据。在大多数情况下，个人计算机通过 RS232C 串行接口端口来传递串行数据。

服务器

指为网络上的其它计算机提供服务的计算机。例如，文件服务器即是为客户机提供文件。

服务器群集

共享责任的一组互相连接的服务器，通常配置为当一个或多个服务器崩溃时可以继续处理。

服务

一种特殊类型的程序，它“享有优先级”并在系统内以很低的级别运行。服务程序在后台自动运行，不需要用户登录。由于 Windows NT 操作系统是一种安全操作系统，常规程序不允许直接访问硬件（如硬盘）或其它系统对象（如事件批）。服务程序可替其它常规程序访问硬件和系统对象。例如，Wonderware 记录器、WindowViewer 都可以作为 NT 服务程序来运行。

电子表格

在单元格矩阵中处理数据和公式的程序。例如 Microsoft Excel。

语句

计算机语言中的指令表达式。

独立	单一自含式计算机系统，与连接到并依赖于远程计算机系统的计算机相反。独立式计算机可以自行操作，而不必依赖其它设备。
字符串	作为单个数据项连接的一系列字符或位。
子目录	子目录位于目录内，是将文件分组以便存取的结构。子目录如同显示文件位置的地址。
SuperTag	InTouch 支持 SuperTag 模板结构，允许您定义一个名叫 SuperTag 的复合标记名类型。SuperTag 模板最多可容纳 64 个标记名成员和两个嵌套级别。参见“**TemplateMaker**”。
同步传输	以固定速率传输字位的数据传输。为了保证同步，传送器和接收器使用同一时钟信号。参见“**异步**”。
语法	规定语言结构及其表达式的规则。
任务	任务是正在执行的应用程序。任务与“进程”同义。
标记名	在标记名字典（InTouch 数据库）中定义的变量的名称。
TemplateMaker	允许您创建 SuperTag 模板的 InTouch 实用程序。参见“**SuperTag**”。
终端	允许您发送命令给另一计算机的设备。最简单的终端通常是键盘、显示屏和一些简单的电路。个人计算机中通常使用终端软件，这种软件模拟物理终端，允许您输入命令给另一计算机。

终端服务器	带有处理终端数据的多用户操作系统的服务器。
终端服务高级客户端	参见“**TSAC**”
文本框	输入执行命令所需信息的一个框。文本框通常出现在对话框中。
瘦客户机	不带用于访问服务器的硬盘的终端。
平铺窗口	在平铺窗口中，窗口的大小、形状及其在屏幕上的位置由 Windows 决定。平铺窗口是唯一不能覆盖其它窗口的一类窗口，它可以放入图标区域并且可以带有菜单。每个应用程序通常只创建一个平铺窗口。应用程序创建的其它窗口通常为层叠或弹出式窗口。
时间间隔	时间单位。
标题栏	每个窗口顶部的栏，包含应用程序名及该应用程序所使用的文档和文件名（InTouch 提供了一个选项来隐藏标题栏）。标题栏也可用于在屏幕上移动窗口，方法是在抓住标题栏的同时拖动鼠标。
切换	指任何具有两个稳定状态的设备。与“触发器”同义。
触摸屏	用户可以通过手指或其它物体触动指定区域来输入指令的显示屏。
传输控制协议/Internet 协议 (TCP/IP)	网络间用于在 Internet 上互相通讯的协议。

TSAC	Microsoft 的终端服务高级客户端是一个基于 Win32® 的 ActiveX® 控件，可用于在 Microsoft® Internet Explorer 内运行终端服务会话。通过使用 TSAC，您现在可以在 Internet 上运行全功能的 InTouch 应用程序，同时可以达到与局域网相同的性能和速度。
UNIX	设计由许多计算机用户同时使用的、内嵌 TCP/IP 的计算机操作系统，是 Internet 服务器最常用的操作系统。
视图区域	Windows 应用程序中的视图区域（又称工作空间）显示文件的一页。参见“**工作区域**”。
窗口	应用程序工作和查看所用的屏幕上的矩形区域。屏幕上一次可打开多个窗口，并且这些窗口可单独调整大小和移动。
Windows®	由 Microsoft 开发的一个操作环境。
Windows 应用程序	专为 Microsoft Windows 操作环境设计的应用程序，它使用所有的 Windows 特性，例如菜单、滚动条和图标等。
工作区域	显示应用程序本身及所有其它已打开窗口的应用程序窗口区域。
x 轴	坐标平面上的水平轴。
y 轴	坐标平面上的垂直轴。

索引

符号

$AccessLevel, 2-98, 2-107, 2-110, 2-111, 2-113, 4-13, 4-105
$AlarmLogging, 4-105
$AlarmPrinterError, 4-105
$AlarmPrinterNoPaper, 4-105
$AlarmPrinterOffline, 4-105
$AlarmPrinterOverflow, 4-105
$ApplicationChanged, 4-105
$ApplicationVersion, 4-105
$ChangePassword, 2-111, 4-105
$ConfigureUsers, 2-111, 4-105
$Date, 4-106
$DateString, 4-106
$DateTime, 4-106
$Day, 4-106
$HistoricalLogging, 4-106, 8-16
$Hour, 4-106
$InactivityTimeout, 2-98, 2-113, 4-106
$InactivityWarning, 2-97, 2-113, 4-106
$LogicRunning, 4-106
$Minute, 4-106
$Month, 4-106
$Msec, 4-106
$NewAlarm, 4-106
$ObjHor, 4-106
$ObjVer, 4-106
$Operator, 2-107, 2-111, 4-107
$OperatorEntered, 2-111, 4-107
$PasswordEntered, 2-111, 4-107
$Second, 4-107
$StartDdeConversations, 4-107
$System, 4-13, 4-107
$Time, 4-107
$TimeString, 4-107
$Year, 4-107
.Ack, 4-110, 7-63
.AckDev, 4-110, 4-111, 7-63
.AckDsc, 7-63
.AckROC, 4-110, 7-63
.AckValue, 4-110, 7-63
.Alarm, 4-110, 7-63
.AlarmAccess, 4-110, 7-63
.AlarmAckModel, 4-110, 7-64
.AlarmClass, 4-110, 7-64
.AlarmComment, 4-110, 7-64
.AlarmDate, 4-110, 7-64
.AlarmDev, 4-110, 7-64
.AlarmDevCount, 7-64
.AlarmDevDeadband, 4-111, 7-64
.AlarmDevUnAckCount, 7-64
.AlarmDisabled, 4-111, 7-64
.AlarmDsc, 4-111, 7-64
.AlarmDscCount, 4-111
.AlarmDSCCount, 7-64
.AlarmDscDisabled, 4-111, 7-64
.AlarmDscEnabled, 4-111, 7-64
.AlarmDscInhibitor, 4-111, 7-65
.AlarmDscUnAckCount, 4-111, 7-65
.AlarmEnabled, 4-111, 7-65
.AlarmGroup, 7-65
.AlarmGroupSel, 4-111, 7-65
.AlarmHiDisabled, 4-111, 7-65
.AlarmHiEnabled, 4-111, 7-65
.AlarmHiHiDisabled, 4-111, 7-65
.AlarmHiHiEnabled, 4-112, 7-65
.AlarmHiHiInhibitor, 4-112, 7-65
.AlarmHiInhibitor, 4-111, 7-65
.AlarmLimit, 4-112, 7-65
.AlarmLoDisabled, 4-112, 7-65
.AlarmLoEnabled, 4-112, 7-65
.AlarmLoInhibitor, 4-112, 7-65
.AlarmLoLoDisabled, 4-112, 7-65
.AlarmLoLoEnabled, 4-112, 7-66
.AlarmLoLoInhibitor, 4-112, 7-66
.AlarmMajDevDisabled, 4-112, 7-66
.AlarmMajDevEnabled, 4-112, 7-66
.AlarmMajDevInhibitor, 4-112, 7-66
.AlarmMinDevDisabled, 4-112, 7-66
.AlarmMinDevEnabled, 4-112, 7-66
.AlarmMinDevInhibitor, 4-113, 7-66
.AlarmName, 4-113, 7-66
.AlarmOprName, 4-113, 7-66
.AlarmOprNode, 4-113, 7-66
.AlarmPri, 4-113, 7-66
.AlarmProv, 4-113, 7-66
.AlarmROC, 4-113, 7-66
.AlarmROCCount, 7-66
.AlarmROCDisabled, 4-113, 7-66
.AlarmROCEnabled, 4-113, 7-67
.AlarmROCInhibitor, 4-113, 7-67
.AlarmROCUnAckCount, 7-67
.AlarmState, 4-113, 7-67
.AlarmTime, 4-113, 7-67
.AlarmTotalCount, 4-113, 7-67
.AlarmType, 4-113, 7-67
.AlarmUnAckCount, 4-113, 7-67
.AlarmUserDefNum1, 4-114, 7-67
.AlarmUserDefNum2, 4-114, 7-67
.AlarmUserDefStr, 4-114, 7-67
.AlarmValDeadband, 4-114, 7-68
.AlarmValue, 4-114, 7-68
.AlarmValueCount, 4-114, 7-68
.AlarmValueUnAckCount, 4-114, 7-68
.ChartLength, 4-114
.ChartStart, 4-115, 8-29
.Comment, 4-115
.DevTarget, 4-115, 7-68
.DisplayMode, 4-115, 8-29
.EngUnits, 4-115

.Freeze, 4-115, 7-68
.HiHiLimit, 4-115, 7-68
.HiHiStatus, 4-115, 7-68
.HiLimit, 4-115, 7-68
.HiStatus, 4-115, 7-68
.ListChanged, 4-115, 7-69
.LoLimit, 4-115, 7-68
.LoLoLimit, 4-115, 7-68
.LoLoStatus, 4-115, 7-68
.LoStatus, 4-115, 7-68
.MajorDevPct, 4-115, 7-68
.MajorDevStatus, 4-115, 7-68
.MaxEU, 4-16, 4-115
.MaxRange, 4-115, 8-29
.MaxRaw, 4-104, 4-115
.MinEU, 4-16, 4-115
.MinorDevPct, 4-115, 7-68
.MinorDevStatus, 4-115, 7-68
.MinRange, 4-115, 8-29
.MinRaw, 4-104, 4-115
.Name, 4-116, 7-69
.Normal, 4-116, 7-69
.NumAlarm, 7-69
.OffMsg, 4-116
.OnMsg, 4-116
.Pen1 - 8, 8-29
.Pen1-.Pen8, 4-116
.PendingUpdates, 7-69
.Quality, 4-116
.QualityLimit, 4-116
.QualityLimitString, 4-116
.QualityStatus, 4-116
.QualityStatusString, 4-116
.QualitySubstatus, 4-116
.QualitySubstatusString, 4-117
.RawValue, 4-117
.Reference, 4-60, 4-117
.ReferenceComplete, 4-62, 4-117
.ROCPct, 4-117, 7-69
.ROCStatus, 4-117, 7-69
.ScooterLockLeft, 4-117, 8-30
.ScooterLockRight, 4-117, 8-30
.ScooterPosLeft, 4-117, 8-30, 8-44
.ScooterPosRight, 4-117, 8-30, 8-44
.SuppressRetain, 4-117, 7-69
.TagID, 4-117, 8-30
.TimeDate, 4-118
.TimeDateString, 4-118
.TimeDateTime, 4-118
.TimeDay, 4-118
.TimeHour, 4-118
.TimeMinute, 4-118
.TimeMonth, 4-118
.TimeMsec, 4-118
.TimeSecond, 4-118
.TimeTime, 4-118
.TimeTimeString, 4-118
.TimeYear, 4-118
.Unack, 4-110, 4-119
.UpdateCount, 4-119, 8-30
.UpdateInProgress, 4-119, 8-30
.UpdateTrend, 4-119, 8-30
.Value, 4-119

A

Abs(), 6-61
Ack, 6-64
ACK 模型
  扩展的摘要, 7-25, 7-34
ActivateApp(), 6-62
ActiveX, 0-1
ActiveX 事件脚本对话框, 6-25
ActiveX 控件, 2-5
  InTouch不支持的, 2-77
  This Event, 2-91
  从工具栏删除, 2-80
  从应用程序删除, 2-78
  方法, 2-76, 2-86
  关联方向, 2-85
  关联符号, 2-85
  在 InTouch 中使用, 2-77
  安装, 2-78
  导入事件脚本, 2-95
  访问方法/属性, 2-87
  更改 ActiveX 控件名, 2-82
  事件, 2-76
  事件参数, 2-91
  事件脚本, 2-92, 6-2, 6-24
  将标记名与属性关联, 2-83
  标记名/属性关联, 2-85
  重复使用事件脚本, 2-93
  容器, 2-76
  配置属性, 2-83
  控件名, 2-81, 2-91
  控件浏览器对话框, 2-87, 2-91
  粘贴到窗口, 2-79
  属性对话框, 2-81
  编辑, 2-77
ActiveX 控件/容器, 0-1
ActiveX 控件浏览器对话框, 6-26
ActiveX 控制安装对话框, 2-78
ActiveX 控制属性对话框, 2-76
Alarm DB Logger, 7-82, 7-83, 7-84, 7-85, 7-86, 7-88, 7-159
Alarm Disablement, 7-30
AlarmSuite 历史迁移实用程序, 7-164
  移植 AlarmSuite 历史数据, 7-164
AlarmSuite 报警记录数据库视图, 7-176
almAckAll, 6-64
almAckDisplay, 6-64
almAckRecent, 6-64

almAckSelect, 6-64
almDefQuery, 6-64
almMoveWindow, 6-64
almQuery, 6-64
almSelectAll, 6-64
almSelectItem, 6-64
almShowStats, 6-64
AND 运算符, 6-50
API 调用, 7-3
ArcCos(), 6-61
ArcSin(), 6-61
ArcTan(), 6-61

B

Beta 测试, 0-3
BIOS, 0-4
bla, 4-47

C

CD-ROM, 0-5
ChangePassword(), 6-64
Client, 10-5
  Command, 10-5
COM 端口, 0-6
CONFIG.SYS, 0-6
Cos(), 6-61
CSV, 0-7
CSV文件转储到对话框, 4-122

D

DBDump, 4-121
  CSV 文件, 4-122
  分类分组输出, 4-122
DBLoad, 4-121, 4-124
  关键字, 4-132
  创建 SuperTag 实例, 4-126
  空白字符串, 4-127
  数据库记录模板
    Type 和关键字条目, 4-133
    设置字段缺省值, 4-132
    重置字段缺省值, 4-132
  数据库输入文件, 4-125
  数据库输入文件格式, 4-125
  操作模式
    MODE=ASK, 4-129
    MODE=IGNORE, 4-131
    MODE=REPLACE, 4-129
    MODE=TERMINATE, 4-131
    MODE=TEST, 4-131
    MODE=UPDATE, 4-129
DDE
  赋值（一次性）. *参见* WWPoke()
DHCP, 0-7
DialogStringEntry(), 6-64
DialogValueEntry(), 6-64
DRA, 0-9
DRC, 3-22, 3-24, 0-9
DText(), 6-59

E

ENTER 键, 0-9
Enterprise Manager, 7-168, 7-174
ErrorNumber, 6-75
Ethernet, 0-10
Excel
  在 Excel 中使用 HistData, 8-60
Exp(), 6-61

F

FactoryFocus, xv
FactorySuite, 0-11
FAT, 3-15
FileCopy(), 6-62
FileDelete(), 6-62
FileMove(), 6-62
FileReadFields(), 6-62
FileReadMessage(), 6-62
FileWriteFields(), 6-62
FileWriteMessage(), 6-62
FOR-NEXT 循环脚本, 6-38, 6-51
  五秒时间限制, 6-40
  屏幕更新, 6-39
  退出控制结构, 6-42
  嵌套 FOR-NEXT 循环, 6-39
  嵌套控制结构, 6-41
  循环执行后的循环变量值, 6-41

G

GetNodeName(), 6-65
GetPropertyD(), 6-65
GetPropertyI(), 6-65
GetPropertyM(), 6-65
GMT, 3-25

H

HCL, 10-17
Hide, 6-65
HideSelf, 6-65
HistData 实用程序, 8-53
  HistData 数据库, 8-53
  在 Excel 中使用 HistData, 8-60
  在 InTouch 中使用 HistData, 8-56
HMI, 0-12
HTGetLastError(), 6-65, 8-31
HTGetPenName(), 6-65, 8-31
HTGetTimeAtScooter(), 6-65, 8-31
HTGetTimeStringAtScooter(), 6-65, 8-31
HTGetValue(), 6-65, 8-31

HTGetValueAtScooter(), 6-65, 8-31
HTGetValueAtZone(), 6-65, 8-31
HTScrollLeft(), 6-65, 8-31
HTScrollRight(), 6-65, 8-32
HTSelectTag(), 6-65
HTSetPenName(), 6-66, 8-32
HTUpdateToCurrentTime(), 6-66, 8-32
HTZoomIn(), 6-66, 8-32
HTZoomOut(), 6-66, 8-32

I

I/O, 0-12
  实型标记名类型, 4-4
  标记名类型, 4-4
  消息型标记名类型, 4-4
  离散型标记名类型, 4-4
  整型标记名类型, 4-4
I/O 消息, 4-24
I/O 编址, 9-5
I/O 数据源, 3-11
I/OStatus 主题名, 9-13
I/O服务器
  监视服务器通讯的状态, 9-17
I/O通讯, 9-1
ICA, 10-4, 0-12
InfoAppActive(), 6-63
InfoAppTitle(), 6-63
InfoDisk(), 6-63
InfoFile(), 6-63
InfoInTouchAppDir(), 6-63
InfoResources(), 6-63
InSql History Provider 属性对话框, 8-37
Int(), 6-61
InTouch DDE 地址, 9-5
InTouch SuperTag 语法, 4-35
InTouch 交叉引用视图选项对话框, 4-83
InTouch 交叉引用搜索标准对话框, 4-78
InTouch 访问名, 3-10
  创建, 9-6
  删除, 9-9
  修改, 9-9
InTouch 特性
  FactoryFocus, xv
  分布式历史记录, xv
  分布式警报系统, xii
  动态分辨率转换, xv
  动态引用编址, xv
  网络应用程序开发, xv
INTOUCH.INI, xx
IOGetApplication, 6-66
IOGetNode, 6-66
IOGetTopicIOReinitialize, 6-66
IOReinitialize, 6-66
IOSetAccessName, 6-66
IOSetItem, 4-62, 6-66
IP 编号或地址, 0-13
IP（Internet 协议）, 0-12
IsAnyAsyncFunctionBusy(), 6-35, 6-63

J

Java, 0-13

L

Log(), 6-61
LogMessage(), 6-66
LogN(), 6-61

M

MB, 0-14
Megabyte, 0-14
MS/DOS, 0-15

N

NAD, 3-5, 3-17, 8-34, 0-15
  更新功能, 3-20
NAD 系统, 7-4
NetBEUI, 3-8
NetBIOS 增强用户接口 (NetBEUI), 0-16
NOT 运算符, 6-50
NT 用户管理员, A-5
NT 网域结构, 3-9
NTFS, 3-15
NTFS 文件系统, 0-16
NT服务
  WindowViewer, 2-104

O

OCX, 2-76
OLE 控件, 2-76
OR 运算符, 6-50

P

Pi(), 6-61
PlaySound(), 6-66
PrintHT(), 6-66
PrintScreen(), 6-67
PrintWindow(), 6-67

Q

QuickFunction, 6-2, 6-27, 0-20
  IsAnyAsyncFunctionBusy() 函数, 6-35
  异步, 6-27, 6-33
QuickScript, 0-20
  ActiveX 事件, 6-2
  QuickFunction, 6-2
  一般步骤, 6-4

动作按钮, 6-2
向导, 6-2
在运行时停止, 2-101, 2-102
导入 ActiveX 事件脚本, 2-95
应用程序, 6-2
条件, 6-2
窗口, 2-14, 6-2
数据改变, 6-2
键, 6-2
QuickScript 函数
TseGetClientId(), 10-24
TseQueryRunningOnClient(), 10-24
TseQueryRunningOnConsole(), 10-24
QuickScripts
导入, 2-23
QuickScripts 函数
almAckAll, 7-59
almAckDisplay, 7-59
almAckGroup, 7-59
almAckPriority, 7-59
almAckRecent, 7-59
almAckSelect, 7-59
almAckSelectedGroup, 7-59
almAckSelectedPriority, 7-59
almAckSelectedTag, 7-59
almAckTag, 7-59
almDefQuery, 7-62
almMoveWindow, 7-61
almQuery, 7-62
almSelectAll, 7-61
almSelectGroup, 7-61
almSelectionCount, 7-61
almSelectItem, 7-61
almSelectPriority, 7-61
almSelectTag, 7-61
almSetQueryByName, 7-62
almShowStats, 7-61
almSuppressAll, 7-60
almSuppressDisplay, 7-60
almSuppressGroup, 7-60
almSuppressPriority, 7-60
almSuppressRetain, 7-60
almSuppressSelected, 7-60
almSuppressSelectedGroup, 7-60
almSuppressSelectedPriority, 7-60
almSuppressSelectedTag, 7-60
almSuppressTag, 7-60
almUnSelectAll, 7-61
almUnSuppressAll, 7-60

R

RAM, 0-20
RDP, 10-4, 10-5
RDP 客户端的硬件要求, 10-18
ReloadWindowViewer(), 6-67
RestartWindowViewer(), 6-67
Round(), 6-61
RTU 报警记录, 7-12

S

SendKeys, 6-67
SetPropertyD(), 6-67
SetPropertyI(), 6-67
SetPropertyM(), 6-67
Sgn(), 6-61
Show, 6-67
ShowAt(), 6-67
ShowTopLeftAt(), 6-67
Sin(), 6-61
SPC, 7-29
SQL Server, 7-82, 7-83, 7-85
SQL 服务器, 7-165
Sqrt(), 6-61
StartApp(), 6-63
StringASCII(), 6-59
StringChar(), 6-59
StringFromIntg(), 6-59
StringFromReal(), 6-59
StringFromTime(), 6-59
StringInString(), 6-59
StringLeft(), 6-59
StringLen(), 6-59
StringLower(), 6-59
StringMid(), 6-59
StringReplace(), 6-60
StringRight(), 6-60
StringSpace(), 6-60
StringTest(), 6-60
StringToIntg(), 6-60
StringToReal(), 6-60
StringTrim(), 6-60
StringUpper(), 6-60
SuiteLink, 3-7, 9-1, 9-3
Supertag, 4-2
TemplateMaker, 4-37
创建
DBLoad, 4-126
创建父模板, 4-37
创建成员标记名, 4-39
在标记名字典中创建, 4-52
在标记浏览器中访问, 4-72
有效语法, 4-35
删除, 4-47
使用远程标记名引用, 4-35
定义实例, 4-48
类型, 4-39
结构, 4-36, 4-44
添加成员, 4-39
等级, 4-44
编辑, 4-47
SuperTag, 0-23

其它创建方法, 4-51
SuperTags
创建, 4-34
定义成员标记名, 4-25
标记名类型, 4-8

T

Tan(), 6-61
TCO, 10-5
TCP/IP, 3-9, 10-5
TemplateMaker, 4-37, 0-23
Text Box 控件向导, 2-67
Text(), 6-60
Trunc(), 6-61
TSAC, 0-24
TseGetClientId(), 10-24
TseQueryRunningOnClient(), 10-24
TseQueryRunningOnConsole(), 10-24

U

UNC 路径, 3-6, 8-37, 8-42
UNIX, 0-25

V

VGA 显卡, 10-18
VTQ, 3-7, 9-2

W

wcAddItem(), 6-67
wcClear(), 6-67
wcDeleteItem(), 6-67
wcDeleteSelection(), 6-67
wcErrorMessage, 6-75
wcErrorMessage(), 6-68
wcFindItem(), 6-68
wcGetItem(), 6-68
wcGetItemData(), 6-68
wcInsertItem(), 6-68
wcLoadList(), 6-68
wcLoadText(), 6-68
wcSaveList(), 6-68
wcSaveText(), 6-68
wcSetItemData(), 6-68
WindowMaker
GUI, 1-2
打印输出对话框, 1-16, 6-58
状态栏, 1-24
标尺, 1-23
调色板, 1-25, 2-51
属性, 2-6
属性对话框, 2-6
WindowMaker 工具栏
全部隐藏, 1-15
向导/ActiveX, 1-17, 2-24
改变浮动工具栏的大小, 1-14
固定, 1-13
视图, 1-20, 2-24
显示/隐藏固定工具栏, 1-14
绘图对象, 2-24
绘图对象工具栏, 1-19
格式, 1-17, 2-53
浮动, 1-13
浮动固定工具栏, 1-14
通用, 2-10, 2-16, 2-24
排列, 1-13, 1-21, 2-24, 2-36
WindowMaker 程序元素, 1-1
Windows 2000
兼容硬件表 (HCL), 10-17
Windows 应用程序, 0-25
WindowViewer
在转换到 WindowMaker 时关闭所有打开的窗口, 2-97
自定义, 2-96
作为图标启动, 2-97
保留最大化, 2-102
复制应用程序, 3-21
显示/隐藏菜单栏, 2-101
选择起始窗口, 2-103
逻辑菜单, 2-101
属性, 2-97
禁止转换到 WindowMaker, 2-101
WindowViewer 属性
对话框, 2-97
Wonderware Logger, 2-7
自动启动, 2-7
Wonderware SuiteLink 通讯协议, 3-7, 9-1, 9-3
Wonderware 服务用户对话框, 2-105
WWControl(), 6-69
WWExecute(), 6-69
WWPoke(), 6-69
WWRequest(), 6-69

X

x 轴, 0-25

Y

y 轴, 0-25

二划

二-十进制, 0-3
二进制, 0-3
二进制码, 0-3

三划

上下文相关菜单运行时特征复选框, 7-97
上下文相关菜单选项

列明细按钮, 7-103
状态栏, 7-101
显示模式, 7-97
检索按钮复选框, 7-102
辅助排序对话模式, 7-95
激活刷新菜单, 7-94
激活复位菜单, 7-96
激活排序菜单, 7-95
下划线工具, 1-18
下拉式列组合框, 2-70
下拉式组合框, 2-70
千兆位 Ethernet, 0-11
子目录, 0-23
子状态, 7-5
工作区域, 0-25
工具栏工具
三维按钮, 1-19
下划线, 1-18
中心点对齐, 1-21, 2-38
分解单元, 1-22, 2-44
分解符号, 1-21, 2-44
历史趋势, 1-19, 8-7
文本对象, 1-19, 2-55
文本颜色, 1-18
水平/垂直线, 1-19
水平间距, 1-21, 2-40
水平翻转, 1-22, 2-42
长方形, 1-19
右对齐, 1-18, 1-21, 2-37
对齐网格, 1-20
对角线, 1-19
左对齐, 1-18, 1-21, 2-36
打开窗口, 2-16
打印, 1-16
全屏幕模式, 1-20
关闭窗口, 2-17
多边形, 1-19
多边线, 1-19
字体, 1-18, 2-56
字体放大, 1-18
字体缩小, 1-18
位图, 1-19, 2-47, 2-49, 2-51
应用程序浏览器, 1-20
改变对象形状工具, 1-22, 2-34
制作单元, 1-21, 2-44
制作符号, 1-21, 2-44
垂直间距, 1-21, 2-40
垂直居中对齐, 1-21, 2-38
垂直翻转, 1-22, 2-42
实时趋势, 1-19, 8-2
居中, 1-18
居中对齐, 1-21, 2-37
底部对齐, 1-21, 2-38
线颜色, 1-18
顶部对齐, 1-21, 2-37
保存所有窗口, 1-16
保存窗口, 2-16
复制对象, 1-16, 2-29, 2-30
恢复, 1-16, 2-27
标尺, 1-20
逆时针旋转, 2-41
选择器, 1-19
顺时针旋转, 1-22, 2-41
圆角长方形, 1-19
透明颜色, 1-18
剪切对象, 1-16, 2-30
斜体, 1-18
粗体, 1-18
粘贴对象, 1-16, 2-31
隐藏所有工具栏, 1-20
椭圆, 1-19
窗口背景色, 1-18
填充色, 1-18
新建窗口, 1-17, 1-18, 1-19, 1-20, 1-21, 1-22
置后, 1-21, 2-39
置前, 1-21, 2-39
撤消, 1-16, 2-27
工程单位, 4-16, 8-6
已分布的应用程序, 0-19
已知问题和限制, 10-14

四划

不活动, 0-12
为报警配置视图节点, xxii
为真时条件脚本, 6-20, 6-21
为真期间条件脚本, 6-20, 6-21
为笔选择一个标记对话框, 8-23
为假时条件脚本, 6-20, 6-21
为假期间条件脚本, 6-20, 6-21
互联网, 0-12
五秒时间限制, 6-40
从工具栏对话框删除向导, 2-65, 2-80
允许 CTRL-Break 停止脚本, 2-101, 2-102
公共数据源, 3-10
内存
实型标记名类型, 4-3
消息型标记名类型, 4-3
离散型标记名类型, 4-3
整型标记名类型, 4-3
内部系统 $Tagnames
$ChangePassword, 4-105
$ConfigureUsers, 4-105
内部系统 $Tagnames, 4-105
$AlarmLogging, 4-105

$AlarmPrinterNoPaper, 4-105
$AlarmPrinterOffline, 4-105
$AlarmPrinterOverflow, 4-105
$ApplicationChanged, 4-105
$ApplicationVersion, 4-105
$Date, 4-106
$DateString, 4-106
$DateTime, 4-106
$Day, 4-106
$HistoricalLogging, 4-106
$Hour, 4-106
$InactivityTimeout, 4-106
$InactivityWarning, 4-106
$LogicRunning, 4-106
$Minute, 4-106
$Month, 4-106
$Msec, 4-106
$NewAlarm, 4-106
$ObjHor, 4-106
$ObjVer, 4-106
$Operator, 4-107
$OperatorEntered, 4-107
$PasswordEntered, 4-107
$Second, 4-107
$StartDdeConversations, 4-107
$System, 4-107
$Time, 4-107
$TimeString, 4-107
$Year, 4-107
内部系统$Tagnames
$AccessLevel, 4-105
$AlarmPrinterError, 4-105
冗余, 3-5
分发应用程序, 8-35
分布对象, 2-40
分布式
历史记录, 3-26
应用程序与时区, 3-25
报警多项选择, 7-36
报警消息颜色, 7-36
分布式历史系统, 8-1, 8-33
历史记录, 8-42
远程历史供应器, 8-41
配置 InSql 为历史供应器, 8-37
配置远程历史供应器, 8-39, 8-40
配置供应器列表, 8-36
分布式名称管理器 – 分布式报警对话框, 8-36
分布式名称管理器对话框 – 分布式报警标签, 7-21
分布式报警对象
右击菜单, 7-37
创建, 7-43
状态栏, 7-36
使用准则, 7-42
调整大小, 7-35
配置, 7-33, 7-44
配置查询缺省值, 7-52
配置消息, 7-47
配置通用属性, 7-44
配置确认注释, 7-53
配置颜色, 7-50
滚动条, 7-35
分布式报警系统, 7-1, 7-3
抑制报警, 7-60
报警组和报警组列表, 7-21
显示报警统计, 7-55
监视本地报警, 7-54
监视属性, 7-57
通过提交报警查询来检索报警信息, 7-62
控制报警的显示, 7-61
控制属性, 7-58
脚本函数, 7-57
确认报警, 7-59
分布式报警系统点域
.AlarmGroup, 7-57
.NextPage, 7-58
.NumAlarms, 7-58
.PageNum, 7-58
.PrevPage, 7-58
.PriFrom, 7-57
.PriTo, 7-57
.ProvidersReq, 7-58
.ProvidersRet, 7-58
.QueryState, 7-57
.QueryType, 7-57
.Successful, 7-58
.TotalPages, 7-58
分布式报警系统特点, 7-3
分布式报警数据库视图, 7-166
在 Enterprise Manager 中查看视图定义, 7-168
视图与存储过程简介, 7-166
调用视图, 7-168
分布报警对象
下一页/上一页控制, 7-35
分解单元工具, 1-22, 2-44
分解符号工具, 1-21, 2-44
分辨率, 11-6
切换, 0-24
切换按钮, 5-18
切换显示模式, 1-20
区域, 7-6
升级, 10-7
历史记录
存储记录文件, 8-17
启用记录, 8-16
历史记录属性对话框, 8-16, 8-28, 8-42
历史记录频率, 8-19
历史点域

.ChartLength, 4-114
.ChartStart, 4-115, 8-29
.DisplayMode, 8-29
.MaxRange, 8-29
.MinRange, 8-29
.Pen1 - 8, 8-29
.ScooterLockLeft, 4-117, 8-30
.ScooterLockRight, 4-117, 8-30
.ScooterPosLeft, 4-117, 8-30
.ScooterPosRight, 4-117, 8-30
.TagID, 8-30
.UpdateCount, 4-119, 8-30
.UpdateInProgress, 4-119, 8-30
.UpdateTrend, 4-119, 8-30
历史脚本函数, 8-31
历史趋势, 8-1
工程单位, 8-6
历史趋势对象, 8-6
打印趋势, 8-27
记录标记名, 8-15
向导, 8-12
在运行时更新, 8-26
在运行时配置, 8-20
自动更改系统时间, 8-48
时间刻度, 8-9
指示器, 8-43
取得区域间的数值, 8-44
显示数值, 8-44
缩放, 8-45
显示模式, 8-9, 8-21
平均/散点, 8-21
均值/棒图, 8-22
最小/最大, 8-21
配置历史趋势对象, 8-8
历史趋势设置对话框, 8-20
历史趋势图表向导对话框, 8-13
历史趋势指示器, 8-6
历史趋势标记名类型, 4-5
历史趋势点域, 8-30
历史趋势配置对话框, 8-8
开发快速切换, 2-9
文本工具
下划线, 1-18
文本颜色, 1-18
右对齐, 1-18
左对齐, 1-18
字体, 1-18, 2-56
字体放大, 1-18
字体缩小, 1-18
居中, 1-18
粗体, 1-18
文本对象, 2-2
下划线, 1-18
右对齐, 1-18
对齐, 2-53
左对齐, 1-18
创建, 2-55
字体放大, 1-18
字体缩小, 1-18
更改文本, 2-57
居中, 1-18
显示数值, 2-55
选择
字体磅值, 2-56
选择字体, 1-18, 2-56
选择颜色, 1-18
格式化, 2-53
斜体, 1-18
粗体, 1-18
替换部分, 2-58
编辑, 1-31
数值, 2-53, 2-54
文本对象工具, 1-19, 2-55
文本框, 0-24
文本框控件向导, 2-66
文本颜色工具, 1-18
文件, 0-11
文件名, 0-11
文件访问, 3-15
文字终端, 10-6
文档, 0-8
方向链接, 5-36, 5-39
方法, 0-15
比较 (<, >, <=, >=, ==, <>) 运算符, 6-50
水平/垂直线工具, 1-19
水平间距工具, 1-21, 2-40
水平游标详细资料对话框, 8-44
水平游标链接, 5-15
水平填充百分比链接, 5-34
水平翻转工具, 1-22, 2-42
父模板, 4-37
计时间隔, 2-98
计算机图形, 0-6
长方形工具, 1-19
队列, 0-19

五划

世界调整时间, 3-25
主偏差, 4-31
主题名, 9-4
仪器故障监视, 4-104
加法 (+) 运算符, 6-49
加载CSV文件自对话框, 4-124
占位符标记名, 2-20, 2-23
发布应用程序, 11-2
不包括开发文件, 11-4

用多种分辨率, 11-6
作者名, 11-4
软件包名, 11-4, 11-5
描述, 11-4
只能运行选项, 11-4
可见性链接, 5-36
右击菜单
对话框, 1-31
对象, 1-30, 2-25
浮动工具栏, 1-31
窗口, 1-29, 2-10
右对齐工具, 1-18
处理, 0-19
外围设备, 10-17
失效
链接, 5-36
对齐工具
中心点, 1-21, 2-38
右边缘, 1-21, 2-37
左边缘, 1-21, 2-36
垂直居中, 1-21, 2-38
居中, 1-21, 2-37
底边缘, 1-21, 2-38
顶边缘, 1-21, 2-37
对齐对象
中心点, 1-21, 2-38
右边缘, 1-21, 2-37
左边缘, 1-21, 2-36
垂直居中, 1-21, 2-38
居中, 1-21, 2-37
底边缘, 2-38
底部, 1-21
顶边源, 2-37
顶部, 1-21
对齐网格, 2-7, 2-45
工具, 1-20
对齐对象, 2-45
对角线工具, 1-19
对话框, 0-7
ActiveX 事件脚本编辑器, 6-25
ActiveX 控件安装, 2-78
ActiveX 控件浏览器, 2-87, 6-26
ActiveX 控件属性, 2-81
CSV文件转储到, 4-122
InSql History Provider 属性, 8-37
InTouch 交叉引用视图选项, 4-83
InTouch 交叉引用搜索标准, 4-78
WindowMaker 打印输出, 1-16, 6-58
WindowMaker 属性, 2-6
WindowViewer 属性, 2-97
Wonderware 服务用户, 2-105
为笔选择一个标记, 8-23
从工具栏删除向导, 2-65, 2-80
分布式名称管理器, 7-21, 8-36
历史记录属性, 8-16, 8-28, 8-42
历史趋势设置, 8-20
历史趋势图表向导, 8-13
历史趋势配置, 8-8
水平游标详细资料, 8-44
加载CSV文件自, 4-124
对象类型, 5-2
打印窗口, 4-91
节点属性
应用程序开发, 2-105
关于, 1-38
动画选择, 5-3
向导/ActiveX 安装, 2-62
向导选择, 1-17, 2-64, 2-79
字体, 2-56
导出窗口, 2-20
导出操作问题, 2-21
当触摸时显示窗口, 5-21
访问名, 9-6
应用程序开发, 2-105
应用程序动作脚本, 6-12
报警配置, 7-44, 7-47, 7-50, 7-52, 7-53, 8-8
报警属性, 7-33
改变口令, 2-109
条件脚本编辑器, 6-20
定义标记过滤器, 4-76
实时趋势配置, 8-3
保存窗口, 2-18
标记名字典, 4-12, 4-49
主要, 4-11, 4-12, 4-49
成员, 4-11
报警, 4-11
详细, 4-11
详细资料
I/O 离散型, 4-19
I/O 模拟型, 4-21
SuperTag 成员, 4-25, 4-50
内存消息型, 4-18, 4-19, 4-24
内存离散型, 4-15
内存模拟型, 4-16
标记源, 4-73
浏览文件夹, 2-19, 2-22, 2-63
选择 ActiveX 脚本, 2-93
选择标记名, 5-6
选择标记类型, 4-12, 4-49
重复名称 (DBLoad), 4-130
离散型报警状态, 4-28, 4-30
起始窗口, 2-103
通用特性, 1-32
配置用户, 2-108

添加访问名, 9-7
添加颜色, 1-26
脚本编辑器
应用程序脚本, 6-12
窗口脚本, 6-13
触动动作脚本, 6-17
键脚本, 6-15
替代标记, 4-96
替换文本, 2-59, 4-103
替换字符串, 2-57, 2-58
登录, 2-110
窗口配置, 2-100
窗口属性, 2-11
编辑自定义颜色, 2-52
数据改变脚本编辑器, 6-22
新成员标记, 4-39, 4-42, 4-43
新模板, 4-37, 4-41
对象, 0-16
ActiveX 控件, 2-5
分层, 2-39
文本, 2-2
水平间距, 2-40
对齐网格, 2-45
闪烁速度, 2-99
全部选定, 2-27
向导, 2-4
设置动画, 5-9
位图, 2-3
删除/清除, 2-32
删除多边形顶点, 2-35
删除多边线顶点, 2-35
删除轮廓, 2-60
单元, 2-4
单元 - 翻转, 2-44
垂直间距, 2-40
放大半径, 2-33
线条, 2-2
复杂, 2-3
复制, 2-30
复制链接, 2-32
按钮, 2-2, 2-3
选择, 2-26
选择多个, 2-27
选择组, 2-27
调整大小, 2-26
旋转, 2-41
添加多边形顶点, 2-35
清除链接, 2-32
符号, 2-3
粘贴, 2-31
粘贴链接, 2-32
属性, 2-2
趋势, 2-3
填充图案, 2-2
简单, 2-2
缩小半径, 2-33
撤消选定组, 2-27
翻转, 2-42
对象大小链接, 5-30
对象分层, 2-39
对象类型对话框, 5-2
对象接触时蜂鸣, 2-98
对象镜像, 2-42
左对齐工具, 1-18
左移位 (SHL) 运算符, 6-49
左移位 (SHR) 运算符, 6-49
平均分布对象, 2-40
平铺窗口, 0-24
打开窗口, 2-16
打开窗口工具, 2-16
打印
历史趋势, 8-27
交叉引用文件, 4-89
标记名字典详细资料, 4-90
脚本, 6-58
窗口信息, 1-16
数据库信息, 1-16
打印工具, 1-16
打印窗口对话框, 4-91
本地工作目录, 3-18
本地变量, 0-13
本地标记计数, 2-8, 4-9, 4-93, 4-94, 4-95
正式域名 (FQDN), 0-11
用箭头键移动对象, 1-37
电子表格, 0-22
目录, 0-8
节点, 0-16
节点属性对话框, 2-105
记录死区, 4-14, 4-16
记录事件, 4-14
记录数据, 4-14
边界, 0-4
闪烁对象速度, 2-99
闪烁链接, 5-36, 5-37

六划

交叉引用实用程序, 4-78
打印 交叉引用文件, 4-89
创建搜索过滤器, 4-79
过滤器通配符, 4-79
改变搜索标准, 4-83
图标, 4-81
通过标记名交叉引用, 4-84, 4-86
搜索所有事件, 4-78

搜索指定的事件, 4-78
任务, 0-23
企业, 0-10
优化内存, 2-97
优先权, 0-19
优先级, 7-5
传输要发布的应用程序, 11-2
传输控制协议, 10-5
传输控制协议/Internet 协议 (TCP/IP), 0-24
全屏幕模式工具, 1-20
关于对话框, 1-38
关闭, 0-6
关闭时应用程序脚本, 6-11
关闭窗口, 2-17
关闭窗口工具, 2-17
列表框, 0-13
列表框控件向导, 2-66, 2-67, 2-69
创建
ActiveX 事件脚本, 2-92, 6-24
I/O 项目, 9-10
InTouch SuperTag, 4-34
QuickFunction, 6-29
SuperTag 实例
DBLoad, 4-126
SuperTag 模板结构, 4-36
分布式报警对象, 7-43
文本对象, 2-55
用户输入触动链接, 5-11
字符串, 5-11, 5-14
离散, 5-11, 5-12
模拟, 5-11, 5-13
异步 QuickFunction 脚本, 6-33
自定义安全登录窗口, 2-111
自定义调色板, 1-26
访问名, 9-6
报警组, 7-17
报警组列表, 7-21
指示器, 8-43
显示链接
文本颜色, 5-24, 5-26, 5-28, 5-29
对象大小, 5-22, 5-30
位置, 5-22, 5-32
其它, 5-22
方向, 5-36, 5-39
可见性, 5-36
失效, 5-36
闪烁, 5-36, 5-37
线颜色, 5-24, 5-26, 5-28, 5-29
值显示, 5-22
字符串, 5-42, 5-45
离散型, 5-42
模拟型, 5-42, 5-45
填充百分比, 5-22, 5-34
填充颜色, 5-24, 5-26, 5-28, 5-29
颜色链接
报警, 5-23, 5-29
离散, 5-23
离散报警, 5-28
离散填充颜色, 5-24
模拟, 5-23
模拟填充颜色, 5-26
透明位图, 2-51
游标触动链接
离散, 5-15
数据库输入文件, 4-125
新窗口, 2-11
触动按钮触动链接
动作, 5-17, 5-19
显示窗口, 5-17, 5-21
离散值, 5-17, 5-18
隐藏窗口, 5-17, 5-21
创建SuperTag的其它方法, 4-51
创建分布式应用程序, 3-1
创建动画链接, 5-1
创建标记名服务器应用程序, 4-57
创建数据库记录模板, 4-132
动作按钮脚本, 6-2
动作触动按钮链接, 5-17, 5-19
动态分辨率转换, 3-22, 3-24
动态引用编址
.Reference, 4-60
.ReferenceComplete, 4-62
使用 IOSetItem 改变引用, 4-62
验证引用, 4-62
动态主机配置协议, 0-9
动态配置远程历史供应器, 8-41
动态数据交换, 0-9
动画链接
功能, 5-2
对象类型对话框, 5-2
用户输入触动链接
字符串, 5-14
离散, 5-12
游标, 5-15
模拟, 5-13
访问标记名
点域, 5-7
访问标记浏览器, 5-6
应用颜色链接, 5-5
指定等价键, 5-3
显示链接, 5-22
文本颜色, 5-24, 5-26, 5-28, 5-29
对象大小, 5-22, 5-30
位置, 5-22, 5-32
其它, 5-22
方向, 5-36, 5-39

可见性, 5-36
失效, 5-36
闪烁, 5-36, 5-37
线颜色, 5-24, 5-26, 5-28, 5-29
值显示, 5-22
字符串, 5-42, 5-45
离散型, 5-42
模拟型, 5-42, 5-45
填充百分比, 5-22, 5-34
填充颜色, 5-24, 5-26, 5-28, 5-29
颜色链接
离散, 5-23
离散报警, 5-28
离散填充颜色, 5-24
模拟, 5-23
模拟报警, 5-23, 5-29
模拟填充颜色, 5-26
选择对话框, 5-3
触动按钮触动链接
动作, 5-17, 5-19
显示窗口, 5-17, 5-21
离散值, 5-17, 5-18
隐藏窗口, 5-17, 5-21
触动链接, 5-10
用户输入, 5-10
游标, 5-10
触动按钮, 5-10
协议, 0-19
压缩文件, 11-2
同步传输, 0-23
同或 (|) 运算符, 6-50
后台/背景, 0-3
向导, 2-4
从工具栏删除, 2-65
安装, 2-62
删除, 2-62
添加到工具栏, 2-64
粘贴到窗口, 2-64
向导/ActiveX 工具栏, 1-17, 2-24
向导工具, 1-17, 2-64, 2-79, 2-80
向导/ActiveX 安装对话框, 2-62
向导选择对话框, 1-17, 2-64, 2-79
向导脚本, 6-2
在 Enterprise Manager 中查看存储过程定义, 7-174
在 WindowViewer 中隐藏光标, 2-102
在应用程序浏览器中导航, 1-4
在运行时更新历史趋势, 8-26
在运行时停止脚本, 2-101, 2-102
在运行期间应用程序脚本, 6-11
在按下期间脚本, 6-17
在显示期间窗口脚本, 6-13
多边形
删除顶点, 2-35
改变形状, 1-22, 2-34
添加顶点, 2-35
多边形工具, 1-19
多边线
删除顶点, 2-35
改变形状, 1-22, 2-34
添加顶点, 2-35
多边线工具, 1-19
多任务, 0-15
多种分辨率, 11-6
多监视器系统, 3-24
字节, 0-2, 0-5
字体工具, 1-18, 2-56
字体对话框, 2-56
字体放大工具, 1-18
字体缩小工具, 1-18
字体磅值, 2-56
字符串, 0-23
字符串函数, 6-59
字符串值显示链接, 5-45
字符串脚本函数, 6-6
字符串输入链接, 5-14
存储过程, 7-166
安全, 10-6
安全性
$AccessLevel, 2-107, 2-110
$ChangePassword, 2-109
$ConfigureUsers, 2-111
$Operator, 2-107
$OperatorEntered, 2-111
$PasswordEntered, 2-111
内部标记名, 2-107
失效动画链接, 2-107
自动注销, 2-112
自定义登录窗口, 2-111
访问级别, 2-106
改变口令, 2-109
注销, 2-111
配置用户, 2-108
配置闲置, 2-112
登录, 2-110
管理员, 2-106
安装
ActiveX 控件, 2-78
向导, 2-62
终端服务, 10-20
规划, 10-19
安装模式, 10-20
导入
ActiveX 事件脚本, 2-95
InTouch QuickScript, 4-59, 6-54
自定义调色板, 1-28

位图, 2-47
图像, 2-47
窗口, 2-23, 4-59
转换占位符标记名为远程标记名引用, 4-99
窗口脚本, 2-23
导出
自定义调色板, 1-28
导出窗口, 2-19
占位符标记名, 2-20, 2-23
问题, 2-21
导出窗口对话框, 2-20
导出操作问题, 2-21
导出操作问题对话框, 2-21
异步, 0-2
异步 QuickFunction 脚本, 6-33
异或 (^) 运算符, 6-50
当前文件, 0-7
当触摸时显示窗口对话框, 5-21
执行脚本, 10-15
扩展, 0-10
扩展名, 0-10
扩展的标记名支持, 4-2, 4-9
扩展的摘要报警, 7-25
死区, 4-16
约束器标记名, 4-26, 4-29, 4-32
网络应用程序开发, 3-5, 3-17, 0-15
DDE 访问名, 9-7
I/O 数据源, 3-11
分发应用程序, 8-35
访问名, 3-10
使用分布式历史系统, 8-34
指向 I/O 数据源的本地地址, 3-13
配置文件的 UNC 路径, 3-6
配置网络资源, 3-6
网络段, 3-8
网络结构, 3-2
网络适配器, 10-17
网络疑难排解, 3-8
自动分发主/从应用程序, 3-5
自动更改系统时间, 8-48
自动注销系统, 2-112
自定义
NAD 更新功能, 3-20
分辨率, 3-22
运行时环境, 2-96
您的开发环境, 2-6
自定义分辨率, 0-9
自定义调色板, 1-28
色调, 1-27
行选择精度, 2-9
许可证查看实用程序, 1-38
设备驱动程序, 0-7
设置 WindowViewer 的通用属性, 2-6, 2-97
设置对象动画, 5-9
设置时区, 3-25
访问, 0-1
访问名, 3-10
主题名, 9-7
创建, 9-6
删除, 9-9
应用程序名, 9-7
修改, 9-9
项目名, 9-12
访问名对话框, 9-6
负载平衡, 0-13
过程控制, 0-19
过程控制计算机, 0-19
页面, 0-17

七划

串行端口, 0-22
串联, 0-6
位, 0-4
位与 (&) 运算符, 6-49
位图, 2-46, 0-4
工具, 2-47, 2-49, 2-51
从剪贴板粘贴, 2-49
导入, 2-47
设置原始尺寸, 2-48
容器, 2-51
透明, 1-27, 2-46, 2-51
透明颜色工具, 1-18
旋转, 2-46
粘贴, 2-49
位图工具, 1-19
位图对象, 2-3
位域, 4-120
位置链接, 5-32
作为图标启动 WindowViewer, 2-97
作者名, 11-4
删除
ActiveX – 从应用程序, 2-78
ActiveX 控件 - 从工具栏, 2-80
不用的标记名, 4-93
对象轮廓, 2-60
向导 - 从工具栏, 2-65
字典中标记名, 4-92
访问名, 9-9
报警组, 7-20
图形对象, 2-32
标记浏览器过滤器, 4-77
窗口, 2-17
启用历史记录, 8-16
启动, 0-4

启动本地 I/O, 10-15
启动时应用程序脚本, 6-11

六划

吸管工具, 1-27, 2-52

七划

宏, 0-14
局部变量, 6-36
  有效语法, 6-36
  数据类型, 6-37
局域网 (LAN), 0-13
应用程序, 0-2
应用程序开发对话框, 2-105
应用程序发布器, 11-2, 0-2
应用程序动作脚本对话框, 6-12
应用程序名, 9-4
应用程序复制, 3-21
应用程序浏览器, 1-1, 1-2, 1-3
  TemplateMaker, 1-8
  右击菜单, 1-3
  交叉引用, 1-8
  导航, 1-4
  应用程序, 1-8
  折叠, 1-4, 1-5
  显示, 1-4
  标记名字典, 1-8
  展开, 1-4, 1-5
  添加应用程序, 1-3, 1-10
  隐藏, 1-4
  窗口, 1-5
应用程序浏览器工具, 1-20
应用程序脚本, 6-2
应用程序脚本：关闭时, 6-11
应用程序脚本：在运行期间, 6-11
应用程序脚本：启动时, 6-11
应用程序脚本编辑器对话框, 6-12
应用程序编辑锁定, 3-21
快速切换, 2-9, 2-97
快捷方式和加速键, 1-36
快捷键, 0-1
抑制分布式报警, 7-60
报警, 7-1, 7-5, 0-2
  优先级, 4-14, 4-28, 4-32
  约束器标记名, 4-26, 4-29, 4-32
  报警列表, 7-21
  报警注释, 4-28, 4-31, 4-33
  报警组
    创建, 7-17
    删除, 7-20
  报警组分层结构, 7-16
  报警点域, 7-63
报警类型
  报警值, 7-7
  变化率, 7-7
  偏差, 7-7
事件类型
  ALM, 7-14
  DDE, 7-14
  LGC, 7-14
  OPR, 7-14
  RTN, 7-14
  SYS, 7-14
  USER, 7-14
事件类型
  ACK, 7-14
  EVT, 7-14
定义离散型报警条件, 4-28, 4-30
组, 4-13
格式化报警/事件消息, 7-47
监视本地报警, 7-54
离散型
  报警状态, 4-28
配置分布式报警/事件对象, 7-33
确认, 7-56
模拟型
  主偏差, 4-31
  报警值, 4-31
  定义模拟型报警条件, 4-30
  类型, 4-31
  值死区, 4-31
  偏差死区百分比, 4-32
  偏差百分比, 4-31
  副偏差, 4-31
模拟型变化率, 4-32
报警与事件历史数据库视图, 7-171
报警历史, 7-8
报警历史数据库视图, 7-169
报警计数器数据库存储过程, 7-173
报警打印机, 7-81
报警打印程序, 7-75
报警记录, 7-27
报警优先级, 7-15
报警抑制, 7-31
报警抑制 QuickScripts 函数
  almSuppressAll, 7-60
  almSuppressDisplay, 7-60
  almSuppressGroup, 7-60
  almSuppressPriority, 7-60
  almSuppressRetain, 7-60
  almSuppressSelected, 7-60
  almSuppressSelectedGroup, 7-60
  almSuppressSelectedPriority, 7-60
  almSuppressSelectedTag, 7-60
  almSuppressTag, 7-60
  almUnSuppressAll, 7-60

报警状态, 7-30
报警供应器, 7-26
报警刺激, 7-30
报警服务器节点, 2-99
报警组, 7-6
报警显示控制 QuickScript 函数
almSelectTag, 7-61
almUnSelectAll, 7-61
报警显示控制 QuickScripts 函数
almMoveWindow, 7-61
almSelectAll, 7-61
almSelectGroup, 7-61
almSelectionCount, 7-61
almSelectItem, 7-61
almSelectPriority, 7-61
almShowStats, 7-61
报警查询
修改现有查询, 7-39, 7-41
选择现有查询, 7-39, 7-41
选择查询收藏夹, 7-39
配置查询收藏夹, 7-39
添加新的指定查询, 7-40
报警查询 QuickScript 函数, 7-62
almDefQuery, 7-62
almQuery, 7-62
almSetQueryByName, 7-62
报警点域
.Ack, 4-110
.AckDate, 4-110
.AckDev, 4-110
.AckDevCount, 4-111
.AckDevUnAckCount, 4-111
.AckDsc, 4-110, 4-111
.AckDscCount, 4-111
.AckDscDisabled, 4-111
.AckDscEnabled, 4-111
.AckDscInhibitor, 4-111
.AckDscUnAckCount, 4-111
.AckROC, 4-110
.AckValue, 4-110
.Alarm, 4-110
.AlarmAccess, 4-110
.AlarmAckModel, 4-110
.AlarmClass, 4-110
.AlarmComment, 4-110
.AlarmDev, 4-110
.AlarmDevDeadband, 4-111
.AlarmDisabled, 4-111
.AlarmEnabled, 4-111
.AlarmGroupSel, 4-111
.AlarmHiDisabled, 4-111
.AlarmHiEnabled, 4-111
.AlarmHiHiDisabled, 4-111
.AlarmHiHiEnabled, 4-112
.AlarmHiHiInhibitor, 4-112
.AlarmHiInhibitor, 4-111
.AlarmLimit, 4-112
.AlarmLoDisabled, 4-112
.AlarmLoEnabled, 4-112
.AlarmLoInhibitor, 4-112
.AlarmLoLoDisabled, 4-112
.AlarmLoLoEnabled, 4-112
.AlarmLoLoInhibitor, 4-112
.AlarmMajDevDisabled, 4-112
.AlarmMajDevEnabled, 4-112
.AlarmMajDevInhibitor, 4-112
.AlarmMinDevDisabled, 4-112
.AlarmMinDevEnabled, 4-112
.AlarmMinDevInhibitor, 4-113
.AlarmName, 4-113
.AlarmOprName, 4-113
.AlarmOprNode, 4-113
.AlarmPri, 4-113
.AlarmProv, 4-113
.AlarmROC, 4-113
.AlarmROCDisabled, 4-113
.AlarmROCEnabled, 4-113
.AlarmROCInhibitor, 4-113
.AlarmState, 4-113
.AlarmTime, 4-113
.AlarmTotalCount, 4-113
.AlarmType, 4-113
.AlarmUnAckCount, 4-113
.AlarmUserDefNum1, 4-114
.AlarmUserDefNum2, 4-114
.AlarmUserDefStr, 4-114
.AlarmValDeadband, 4-114
.AlarmValue, 4-114
.AlarmValueCount, 4-114
.AlarmValueUnAckCount, 4-114
.DevTarget, 4-115
.Freeze, 4-115
.HiHiLimit, 4-115
.HiHiStatus, 4-115
.HiLimit, 4-115
.HiStatus, 4-115
.ListChanged, 4-115
.LoLimit, 4-115
.LoLoLimit, 4-115
.LoLoStatus, 4-115
.LoStatus, 4-115
.MajorDevPct, 4-115
.MajorDevStatus, 4-115
.MinorDevPct, 4-115
.MinorDevStatus, 4-115
.ROCPct, 4-117
.ROCStatus, 4-117
.SuppressRetain, 4-117
.Unack, 4-110, 4-119
分布式
.AlarmGroup, 7-57
.NextPage, 7-58
.NumAlarms, 7-58

.PageNum, 7-58
.PrevPage, 7-58
.PriFrom, 7-57
.PriTo, 7-57
.ProvidersReq, 7-58
.ProvidersRet, 7-58
.QueryState, 7-57
.QueryType, 7-57
.Successful, 7-58
.TotalPages, 7-58
报警类型, 7-13
报警套件, 7-4
报警通知, 7-30
报警配置, 7-28
报警配置对话框, 7-44, 7-47, 7-50, 7-52, 7-53
报警高速缓存, 7-27
报警接收器, 7-3, 7-26
报警属性对话框, 7-33
报警确认 QuickScripts 函数
almAckAll, 7-59
almAckDisplay, 7-59
almAckGroup, 7-59
almAckPriority, 7-59
almAckRecent, 7-59
almAckSelect, 7-59
almAckSelectedGroup, 7-59
almAckSelectedPriority, 7-59
almAckSelectedTag, 7-59
almAckTag, 7-59
报警确认模型, 7-24
报警数据库视图 ActiveX 控件, 7-89
上下文相关菜单选项, 7-93
方法和事件, 7-130
字体页, 7-110
安装, 7-90, 7-91
事件页, 7-133
卸载, 7-90
查询页, 7-123
选择页, 7-114
通用页, 7-92
控件名称页, 7-91
属性页, 7-128
数据库页, 7-111
颜色页, 7-104
报警数据库恢复公用程序, 7-160
恢复数据库配置, 7-160
选择需恢复的数据库, 7-162
配置需恢复的数据库, 7-160
报警数据库恢复公用程序文件名配置, 7-162
报警数据库清除/归档公用程序, 7-154
通用属性配置, 7-154
配置清除/归档时间间隔, 7-158
清除/归档报警数据库, 7-155
清除/归档配置, 7-158
清除/归档数据库配置, 7-157
报警禁止, 7-30
报警摘要, 7-8
改变
安全登录口令, 2-109
改变口令对话框, 2-109
改变对象形状工具, 1-22, 2-34
改变浮动工具栏的大小, 1-14
时区, 3-25
时间间隔, 0-24
更改 Active X 控件名, 2-82
更改文件名, 11-6
更新
本地应用程序 - WindowViewer 启动时, 3-18
时间变量, 2-98
使用计数, 4-9, 4-93
所有趋势"Fast", 2-98
条件脚本, 6-2
为真时, 6-20, 6-21
为真期间, 6-20, 6-21
为假时, 6-20, 6-21
为假期间, 6-20, 6-21
条件脚本编辑器对话框, 6-20
状态栏
显示/隐藏, 1-24
系统脚本函数, 6-6, 6-62
运行, 0-21
运行中, 0-21
运行时, 0-21
运行时快速切换, 2-9, 2-97
运行时环境
自定义, 2-96
运算
大于, 6-46
大于或等于, 6-46
小于, 6-46
小于或等于, 6-46
不等于, 6-46
加和链接, 6-46
右移位, 6-46
左移位, 6-46
同或, 6-46
异或, 6-46
位与, 6-46
取反, 6-46
取余数, 6-46
除, 6-46
乘, 6-46
乘方, 6-46
减, 6-46
逻辑 NOT, 6-46
逻辑或, 6-46

等于, 6-46
赋值, 6-46
运算 取补, 6-46
运算符优先级
最低优先级, 6-47
最高优先级, 6-47
运算符描述
AND、OR 和 NOT, 6-50
比较 (<, >, <=, >=, ==, <>), 6-50
加 (+), 6-49
左移位 (SHL), 6-49
左移位 (SHR), 6-49
同或 (|), 6-50
异或 (^), 6-50
位与 (&), 6-49
取反 (-), 6-48
取余数 (MOD), 6-49
取补 (~), 6-48
括号 ( ), 6-48
除 (/), 6-49
乘 (*), 6-49
乘方 (**), 6-48
减 (-), 6-49
赋值 (=), 6-50
运算数, 6-44, 6-46
远程
历史供应器, 8-41
报警节点, 2-99
标记计数, 2-8, 4-9, 4-93, 4-94, 4-95
标记名引用, 4-2, 4-54, 4-98, 4-99, 4-101
手工转换, 4-98
导入窗口, 4-99
有效语法, 4-54, 4-55
标记浏览器, 4-101
远程标记名, 0-20
远程桌面协议, 10-5
远程桌面协议 (RDP), 0-20
远程控制客户机会话, 10-7
远程管理模式, 10-7
连接到终端服务器, 10-26
附加程序脚本函数, 6-7
驱动器, 0-9

八划

事件, 7-1, 7-5, 0-10
格式化报警/事件消息, 7-47
事件历史数据库视图, 7-170
事件计数器数据库存储过程, 7-175
事件类型, 7-14
使用
ActiveX 控件, 2-76
ActiveX 控件方法, 2-86
ActiveX 控件事件参数, 2-91
I/OStatus 主题名, 9-13
I/OStatus 主题名 – 在 Excel 中, 9-16
InSql 作为历史供应器, 8-37
IOSetItem改变引用, 4-62
SuiteLink, 9-3
WindowMaker 帮助, 1-38
WindowMaker 窗口, 2-10
分布式报警显示监视本地报警, 7-54
文本对象, 2-53
动态引用, 4-61
向导, 2-61
安全内部标记名, 2-107
局部变量, 6-36
远程标记名引用, 4-59
图形对象, 2-24
图像和位图, 2-46
线条和轮廓, 2-60
标记名字典字典创建 SuperTag, 4-52
标记浏览器访问远程历史供应器, 8-40
浮动/固定工具栏, 1-13
使用旧的 SendKeys, 2-99
使用两种报警系统, 7-10
使用标记名作为项目名, 4-20
供应器列表, 8-36
其它函数, 6-64
IOSetItem, 4-62
其它标记名类型
历史趋势, 4-5
组变量, 4-5
标记 ID, 4-5
其它脚本函数, 6-7
其它链接
方向, 5-36, 5-39
可见性, 5-36
失效, 5-36
闪烁, 5-36, 5-37
函数
TseGetClientId, 10-24
TseQueryRunningOnClient, 10-24
TseQueryRunningOnConsole, 10-24
分布式报警系统
对象操纵函数
almMoveWindow, 6-64
对象操纵函数
almSelectAll, 6-64
almSelectItem, 6-64
almShowStats, 6-64
报警查询函数
almQuery, 6-64
报警确认函数
Ack, 6-64
almAckAll, 6-64
almAckDisplay, 6-64

almAckRecent, 6-64
almAckSelect, 6-64
almDefQuery, 6-64
历史函数
HTGetLastError(), 6-65, 8-31
HTGetPenName(), 6-65, 8-31
HTGetTimeAtScooter(), 6-65, 8-31
HTGetTimeStringAtScooter(), 6-65, 8-31
HTGetValue(), 6-65, 8-31
HTGetValueAtScooter(), 6-65, 8-31
HTGetValueAtZone(), 6-65, 8-31
HTScrollLeft(), 6-65, 8-31
HTScrollRight(), 6-65, 8-32
HTSelectTag(), 6-65
HTSetPenName(), 6-66, 8-32
HTUpdateToCurrentTime(), 6-66, 8-32
HTZoomIn(), 6-66, 8-32
HTZoomOut(), 6-66, 8-32
字符串函数, 6-59
DText(), 6-59
StringASCII(), 6-59
StringChar(), 6-59
StringFromIntg(), 6-59
StringFromReal(), 6-59
StringFromTime(), 6-59
StringInString(), 6-59
StringLeft(), 6-59
StringLen(), 6-59
StringLower(), 6-59
StringMid(), 6-59
StringReplace(), 6-60
StringRight(), 6-60
StringSpace(), 6-60
StringTest(), 6-60
StringToIntg(), 6-60
StringToReal(), 6-60
StringTrim(), 6-60
StringUpper(), 6-60
Text(), 6-60
系统函数, 6-62
ActivateApp(), 6-62
FileCopy(), 6-62
FileDelete(), 6-62
FileMove(), 6-62
FileReadFields(), 6-62
FileReadMessage(), 6-62
FileWriteFields(), 6-62
FileWriteMessage(), 6-62
InfoAppActive(), 6-63
InfoAppTitle(), 6-63
InfoDisk(), 6-63
InfoFile(), 6-63
InfoInTouchAppDir(), 6-63
InfoResources(), 6-63
IsAnyAsyncFunctionBusy(), 6-35, 6-63
StartApp(), 6-63
其它函数, 6-64
Ack(), 6-64
ChangePassword(), 6-64
DialogStringEntry(), 6-64
DialogValueEntry(), 6-64
GetNodeName(), 6-65
GetPropertyD(), 6-65
GetPropertyI(), 6-65
GetPropertyM(), 6-65
Hide, 6-65
HideSelf, 6-65
IOGetApplication, 6-66
IOGetNode, 6-66
IOGetTopic, 6-66
IOReinitialize, 6-66
IOSetAccessName, 6-66
IOSetItem(), 6-66
LogMessage(), 6-66
PlaySound(), 6-66
PrintHT(), 6-66
PrintScreen(), 6-67
PrintWindow(), 6-67
ReloadWindowViewer(), 6-67
RestartWindowViewer(), 6-67
SendKeys, 6-67
SetPropertyD(), 6-67
SetPropertyI(), 6-67
SetPropertyM(), 6-67
Show, 6-67
ShowAt(), 6-67
ShowTopLeftAt(), 6-67
wcAddItem(), 6-67
wcClear(), 6-67
wcDeleteItem(), 6-67
wcDeleteSelection(), 6-67
wcErrorMessage(), 6-68
wcFindItem(), 6-68
wcGetItem(), 6-68
wcGetItemData(), 6-68
wcInsertItem(), 6-68
wcLoadList(), 6-68
wcLoadText(), 6-68
wcSaveList(), 6-68
wcSaveText(), 6-68
WWControl(), 6-69
WWExecute(), 6-69
WWPoke(), 6-69
WWRequest(), 6-69
插入脚本, 6-6
窗口控件脚本函数
wcSetItemData(), 6-68
数学函数, 6-61
Abs(), 6-61
ArcCos(), 6-61
ArcSin(), 6-61
ArcTan(), 6-61
Cos(), 6-61
Exp(), 6-61

Int(), 6-61
Log(), 6-61
LogN(), 6-61
Pi(), 6-61
Round(), 6-61
Sgn(), 6-61
Sin(), 6-61
Sqrt(), 6-61
Tan(), 6-61
Trunc(), 6-61
制作单元工具, 1-21, 2-44
制作符号工具, 1-21, 2-44
单元, 2-4
分解, 1-22, 2-4, 2-44
制作, 1-21, 2-44
单选钮, 2-66, 2-67, 2-72
参数, 0-2
取反(-) 运算符, 6-48
取反按钮, 5-18
取余数 (MOD) 运算符, 6-49
取补 (~) 运算符, 6-48
取模, 0-15
命令, 0-6
命令行, 0-6
命令按钮, 0-6
命令键, 0-6
固定工具栏, 1-14
图形对象, 0-11
分层, 2-39
水平间距, 2-40
对齐网格, 2-45
闪烁速度, 2-99
全部选定, 2-27
合并, 1-21, 2-44
设置动画, 5-9
删除/清除, 2-32
删除多边形顶点, 2-35
删除多边线顶点, 2-35
删除轮廓, 2-60
单元 - 翻转, 2-44
垂直间距, 2-40
放大半径, 2-33
复制, 2-30
复制链接, 2-32
选择, 2-26
选择多个, 2-27
选择组, 2-27
调整大小, 2-26
旋转, 2-41
添加多边形顶点, 2-35
清除链接, 2-32
粘贴, 2-31
粘贴链接, 2-32
缩小半径, 2-33
撤消选定组, 2-27
翻转, 2-42
图形对象工具
三维按钮, 1-19
历史趋势, 1-19, 8-7
文本对象, 1-19, 2-55
水平/垂直线, 1-19
长方形, 1-19
对角线, 1-19
多边形, 1-19
多边线, 1-19
位图, 1-19, 2-47, 2-49, 2-51
实时趋势, 1-19, 8-2
选择器, 1-19
圆角长方形, 1-19
椭圆, 1-19
图像
JPEG, 2-46
PCX, 2-46
TGA, 2-46
导入, 2-47
垂直间距工具, 1-21, 2-40
垂直游标链接, 5-15
垂直填充百分比链接, 5-34
垂直翻转工具, 1-22, 2-42
备份, 0-3
备份应用程序, xvii
奇偶校验, 0-17
定义
I/O 项目, 9-10
SuperTag 成员标记名详细资料, 4-25
SuperTag 模板实例, 4-48
交叉引用搜索标准, 4-78
标记名报警条件, 4-26
标记名详细资料
I/O 消息型, 4-24
I/O 离散型, 4-19
内存消息型, 4-18
内存离散型, 4-15
内存模拟型, 4-16
标记浏览器标记源, 4-73
标记浏览器搜索过滤器, 4-76
项目名, 9-12
离散型标记名报警条件, 4-28, 4-30
新标记名, 4-10
模拟型标记名报警条件, 4-30
定义标记名详细资料, 4-15
定义标记过滤器对话框, 4-76
定标 I/O 标记名, 4-104
实用程序
HistData, 8-53
实时趋势, 8-1

实时趋势对象, 8-2
配置, 8-3
配置实时趋势对象, 8-3
增强性能, 8-5
实时趋势配置对话框, 8-3
建立终端会话, 10-26
所需最少可用内存, 2-97
放开时脚本, 6-15, 6-17
服务, 0-22
服务器, 0-22
服务器群集, 0-22
波特, 0-2
波特率, 0-3
注销
应用程序, 2-111
注销会话, 10-26
直接访问基于网络的文件, 3-6
直接按钮, 5-18
线条对象, 2-2
线菜单, 2-60
线颜色工具, 1-18
组合框控件向导, 2-66, 2-67
组变量标记名类型, 4-5
终端, 0-23
终端会话, 10-5
终端服务
安装, 10-20
连接, 10-26
函数
TseGetClientId, 10-24
TseQueryRunningOnClient, 10-24
TseQueryRunningOnConsole, 10-24
建立会话, 10-26
注销会话, 10-26
组件, 10-20
客户端生成器文件, 10-20
结束会话, 10-26
配置客户端设置, 10-27
授权, 10-20
断开会话连接, 10-26
终端服务高级客户端, 0-24, 0-25
终端服务器, 10-5, 0-24
安装模式, 10-20
终端服务器概念, 10-4
表达式, 0-10
视图工具栏, 1-20, 2-24
对齐网格工具, 1-20
应用程序浏览器工具, 1-20
标尺工具, 1-20
隐藏所有工具栏工具, 1-20
视图与存储过程的列定义, 7-167
视图区域, 0-25
转换为屏幕视频分辨率, 3-23
转换占位符标记名, 4-97, 6-57
转换旧应用程序, xvii
转换标记名为远程引用, 4-98
轮询, 0-18
软件包名, 11-4, 11-5
更改, 11-6
饱和度, 1-27

九划

保存, 0-21
脚本, 6-10
窗口, 2-16
保存所有窗口工具, 1-16
保存窗口工具, 2-16
保存窗口对话框, 2-18
保留参数, 4-14
保留标记名, 0-21
保留值, 4-14
修改
访问名, 9-9

八划

变化率, 7-29

九划

复杂对象, 2-3
ActiveX 控件, 2-5
向导, 2-4
单元, 2-4
按钮, 2-3
符号, 2-3
趋势, 2-3
复杂脚本, 6-51
复制
对象到 Windows 剪贴板, 2-30
窗口, 2-18
链接, 2-32
复制对象工具, 1-16, 2-29, 2-30
复选框, 0-5
复选框控件向导, 2-66, 2-67
客户端, 0-5
RDP 硬件要求, 10-18
客户端/服务器, 0-5
将 WindowViewer 作为 NT 服务运行, 2-104
将对象置后, 2-39
将对象置前, 2-39
屏幕更新, 6-39
屏幕显示
分辨率, 11-6
屏幕显示大小, 11-6
带宽, 0-3

帮助, 0-12
总体拥有成本, 10-5
总线, 0-5
总是从磁盘加载窗口, 2-97
恢复, 0-20
  层数, 1-16, 2-9
  堆栈, 2-9
  脚本, 6-10
恢复工具, 1-16, 2-27
括号 ( ) 运算符, 6-48
指示器, 8-43
  取得区域间的数值, 8-44
  显示数值, 8-44
  缩放, 8-45
指向 I/O 数据源的本地地址, 3-13
指向 I/O 数据源的全局地址, 3-11
指向文件数据源的本地地址, 3-16
指向文件数据源的全局地址, 3-15
按下时脚本, 6-15, 6-17
按下期间键脚本, 6-15
按钮, 0-5
  切换, 5-18
  取反, 5-18
  复位, 5-18
  离散, 5-18
  置位, 5-18
按钮对象, 2-2, 2-3
星号, 0-2
显示, 0-8
  标记使用计数, 4-94
  数值, 2-53, 2-54, 2-55
显示/隐藏可见网格, 1-20, 2-7, 2-45
显示/隐藏固定工具栏, 1-14
显示/隐藏所有固定工具栏, 1-20
显示/隐藏标尺, 1-20
显示时窗口脚本, 6-13
显示标记计数, 2-8
显示标记名计数, 4-94
显示框
  选择键, 5-4, 6-19
显示窗口触动按钮链接, 5-17, 5-21
显示链接, 5-22
  文本颜色, 5-24, 5-26, 5-28, 5-29
  对象大小, 5-22, 5-30
  位置, 5-22, 5-32
  其它, 5-22
    方向, 5-36, 5-39
    可见性, 5-36
    失效, 5-36
    闪烁, 5-36, 5-37
  线颜色, 5-24, 5-26, 5-28, 5-29
  值显示, 5-22
    字符串, 5-42, 5-45
    离散型, 5-42
    模拟型, 5-42, 5-45
  填充百分比, 5-22, 5-34
  填充颜色, 5-24, 5-26, 5-28, 5-29
  颜色链接
    离散, 5-23
    离散报警, 5-28
    离散填充颜色, 5-24
    模拟, 5-23
    模拟报警, 5-23, 5-29
    模拟填充颜色, 5-26
查看工具栏
  全屏幕模式工具, 1-20
标尺, 1-23
  刻度标志, 1-23
  显示/隐藏, 1-23
标尺工具, 1-20
标记 ID 标记名类型, 4-5
标记名, 0-23
  .fields, 4-69
  I/O 消息型详细资料
    初始值, 4-24
  I/O 离散详细资料
    访问名, 4-19
  I/O 离散型详细资料, 4-19
    初始值, 4-19
    使用标记名作为项目名, 4-20
    项目, 4-20, 4-24
    输入转换, 4-19
  I/O 模拟型详细资料, 4-21
    工程单位, 4-22
    记录死区, 4-23
    死区, 4-21
    访问名, 4-22, 4-24
    初始值, 4-21
    转换, 4-22
    最大工程单位, 4-21
    最大原始数据, 4-21
    最小原始数据, 4-21
  SuperTag 详细资料, 4-25
    数据访问, 4-50
  SuperTag 类型, 4-8
  内存消息详细资料
    初始值, 4-18
  内存消息型详细资料, 4-18
    最大长度, 4-18, 4-24
  内存离散型详细资料
    开消息, 4-15, 4-19
    关消息, 4-15, 4-19
    初始值, 4-15
  内存离散型详细资料, 4-15
  内存模拟型详细资料, 4-16

占位符标记名, 2-20, 2-23
只读, 4-13
打印详细资料, 4-90
本地标记名使用计数, 4-9
记录死区, 4-14
记录事件, 4-14
记录标记名, 4-14
记录数据, 4-14
交叉引用, 4-2, 4-78
交叉引用搜索标准, 4-78
优先级, 4-14
在脚本中查找, 6-9
在脚本中替换, 6-9
扩展的支持, 4-9
有效字符, 4-10, 4-49, 4-52
自动索引, 4-10
位域, 4-120
删除标记名, 4-92, 4-93
报警注释, 4-28, 4-31, 4-33
报警组, 4-13
更新使用计数, 4-93
远程标记名, 4-54
远程标记名计数, 4-95
远程标记名使用计数, 4-9
使用计数, 4-9
使用连字符, 4-10
定义
  报警条件, 4-26, 4-30
  新标记名, 4-10
定义详细资料, 4-15
定标, 4-104
转换为远程引用, 4-99
转换占位符, 4-97
转换占位符标记名为远程引用, 4-98
转换–平方根, 4-22
转换–线性, 4-22
保留参数, 4-14
保留值, 4-14
显示使用计数, 4-94
选择 .field, 4-69
选择点域, 4-109
离散型报警详细资料, 4-28, 4-30
  优先级, 4-28
  报警状态, 4-28
读/写, 4-13
插入脚本, 6-8
替换, 4-96, 4-103
最大工程单位值, 4-104
最小工程单位值, 4-104
模拟型报警详细资料
  优先级, 4-32
  死区, 4-31
  报警类型, 4-31
  变化率, 4-32
  值, 4-31
  偏差死区百分比, 4-32
  偏差百分比, 4-31
  副/主偏差, 4-31
模拟型详细资料
  .Max EU, 4-16
  .Min EU, 4-16
  工程单位, 4-16
  记录死区, 4-16
  死区, 4-16
  初始值, 4-16
标记名 .Fields
  .Min Raw, 4-104
  .最大原始数据, 4-104
标记名字典, 4-1
  创建 SuperTag, 4-52
  扩展的标记名支持, 4-2
  远程标记名引用, 4-2
  标记名交叉引用, 4-2
  标记名类型, 4-3
  标记浏览器, 4-2
  特殊功能, 4-2
标记名字典Supertag, 4-2
标记名字典对话框, 4-12, 4-49
标记名字典实用程序, 4-121
  DBDump, 4-121
  DBLoad, 4-121
标记名定义
  I/O 类型, 9-10
标记名点域, 4-108
  .Ack, 4-110
  .AckDev, 4-110
  .AckDevCount, 4-111
  .AckDevUnAckCount, 4-111
  .AckDsc, 4-110
  .AckROC, 4-110
  .AckValue, 4-110
  .Alarm, 4-110
  .AlarmAccess, 4-110
  .AlarmAckModel, 4-110
  .AlarmClass, 4-110
  .AlarmComment, 4-110
  .AlarmDate, 4-110
  .AlarmDev, 4-110
  .AlarmDevDeadband, 4-111
  .AlarmDisabled, 4-111
  .AlarmOprNode, 4-113
  .AlarmPri, 7-66
  .ChartStart, 8-29
  .Comment, 4-115
  .DisplayMode, 4-115, 8-29
  .EngUnits, 4-115
  .MaxEU, 4-115

.MaxRange, 4-115, 8-29
.MaxRaw, 4-115
.MinEU, 4-115
.MinRange, 4-115, 8-29
.MinRaw, 4-115
.Name, 4-116
.Normal, 4-116
.OffMsg, 4-116
.OnMsg, 4-116
.Pen1 - 8, 8-29
.Pen1-.Pen8, 4-116
.Quality, 4-116
.QualityLimit, 4-116
.QualityLimitString, 4-116
.QualityStatus, 4-116
.QualityStatusString, 4-116
.QualitySubstatus, 4-116
.QualitySubstatusString, 4-117
.RawValue, 4-117
.Reference, 4-60, 4-117
.ReferenceComplete, 4-62, 4-117
.ScooterLockLeft, 8-30
.ScooterLockRight, 8-30
.ScooterPosLeft, 8-30, 8-44
.ScooterPosRight, 8-30, 8-44
.TagID, 4-117, 8-30
.TimeDate, 4-118
.TimeDateString, 4-118
.TimeDateTime, 4-118
.TimeDay, 4-118
.TimeHour, 4-118
.TimeMinute, 4-118
.TimeMonth, 4-118
.TimeMsec, 4-118
.TimeSecond, 4-118
.TimeTime, 4-118
.TimeTimeString, 4-118
.TimeYear, 4-118
.Unack, 4-110
.UpdateCount, 8-30
.UpdateInProgress, 8-30
.UpdateTrend, 8-30
.Value, 4-119
.选择, 4-109
插入脚本, 6-8
标记名类型
I/O, 4-4
I/O实型, 4-4
I/O消息型, 4-4
I/O离散型, 4-4
I/O整型, 4-4
内存, 4-3
内存实型, 4-3
内存消息型, 4-3
内存离散型, 4-3, 4-15, 4-19, 4-24
内存整型, 4-3, 4-21
其它
历史趋势, 4-5
组变量, 4-5
标记 ID, 4-5
标记许可证, 2-8, 4-93, 4-94, 4-95
标记浏览器, 4-2, 4-67
无限选择模式, 4-67, 4-69
过滤选择模式, 4-67, 4-68
过滤器, 4-76
删除过滤器, 4-77
状态栏, 4-68
定义标记源, 4-73
视图, 4-70
详细, 4-71
树, 4-72
选择远程标记名引用, 4-101
选择点域, 4-109
标记类型
内存
内存离散型, 4-18
内存整型, 4-16
标记授权, 4-9
标记源对话框, 4-73
标准报警子系统, 7-9
标题栏, 0-24
标题栏文本, 2-7
活动应用程序, 0-1
浏览文件夹对话框, 2-19, 2-22, 2-63
独立, 0-23
独立式应用程序, 3-1, 3-2
结构
NAD, 3-5
基于服务器, 3-4
基于客户, 3-3
绘图对象工具栏, 1-19, 2-24
三维按钮工具, 1-19
历史趋势工具, 1-19, 8-7
文本对象工具, 1-19, 2-55
水平/垂直线工具, 1-19
长方形工具, 1-19
对角线工具, 1-19
多边形工具, 1-19
多边线工具, 1-19
位图工具, 1-19, 2-47, 2-49, 2-51
实时趋势工具, 1-19, 8-2
选择器工具, 1-19
圆角长方形工具, 1-19
椭圆工具, 1-19
统一程序布署, 10-7
胖客户机, 10-4
语句, 0-22
语法, 0-23
退出控制结构, 6-42

逆时针旋转工具, 2-41
选择
  文本对象的字体, 2-56
  文本对象的字体大小, 2-56
  对象组, 2-27
  导入窗口, 2-23
  图形对象, 2-26
  线宽, 2-60
  线样式, 2-60
  起始窗口, 2-103
  窗口中的多个对象, 2-27
  窗口中的所有对象, 2-27
选择 ActiveX 脚本对话框, 2-93
选择标记名对话框, 5-6
选择标记类型对话框, 4-12, 4-49
选择键对话框, 5-4, 6-19
选择器工具, 1-19
选项按钮, 0-17
选项按钮组, 0-17
选项钮, 2-66, 2-67, 2-72
重试开始, 2-99
重复名称
  对话框, 4-130
重复使用 ActiveX 事件脚本, 2-93
重绘对象, 2-29
重新启动历史日志, 8-16
重置按钮, 5-18
除法 (/) 运算符, 6-49
项目名, 9-4
顺时针旋转工具, 1-22, 2-41

十划

乘方 (**) 运算符, 6-48
乘法 (*) 运算符, 6-49
值, 7-28
值死区, 4-31
值显示链接
  字符串, 5-42, 5-45
  离散型, 5-42
  模拟型, 5-42, 5-45
圆角长方形工具, 1-19
格式工具栏, 1-17, 2-53
  下划线工具, 1-18
  文本颜色工具, 1-18
  右对齐工具, 1-18
  左对齐工具, 1-18
  字体工具, 1-18, 2-56
  字体放大工具, 1-18
  字体缩小工具, 1-18
  居中工具, 1-18
  线颜色工具, 1-18
  透明工具, 1-18
  斜体工具, 1-18
  粗体工具, 1-18
  窗口背景色工具, 1-18
  填充色工具, 1-18
格式化, 0-11
格式化文本对象, 2-53
框式样, 2-13
浮动工具栏, 1-14
浮动固定工具栏, 1-14
消息框, 0-14
热备份
  创建报警映射 .csv 文件, 7-150
  同步实例, 7-152
  导入文件所需要的详细信息, 7-144
  导入报警记录映射 .csv 文件, 7-151
  设置报警记录的关键码域, 7-142
  删除现有的热备份对, 7-141
  更改缺省 ini 文件, 7-137
  运行热备份, 7-152
  使用实例, 7-147
  配置热备份对, 7-147
  添加新的热备份对, 7-138
  编辑现有的热备份对, 7-140
热备份 ：映射报警记录, 7-144
热备份：映射报警记录, 7-145
热备份：映射报警记录关键码域, 7-149
热备份：选择报警记录关键码域, 7-142
热备份与同步, 7-134
  有关热备份对的注意事项, 7-135
热备份配置, 7-136
监视
  多个输入设备状态, 9-19
  服务器通讯的状态, 9-17
监视
  I/O 对话的状态, 9-13
离线, 0-16
离散, 7-28
离散型报警状态对话框, 4-28, 4-30
离散值, 0-8
离散值显示链接, 5-42
离散值触动按钮链接, 5-17, 5-18
离散输入链接, 5-12
缺省, 0-7
调用存储过程, 7-174
调色板, 0-17
  创建自定义颜色, 1-26
  导入/导出, 1-28
  导入/导出自定义调色板, 1-28
  自定义颜色, 1-28
  色调, 1-27
  吸管工具, 1-27, 2-52
  典型颜色, 1-25

饱和度, 1-27
亮度, 1-27
调试脚本, 2-98
调整大小
图形对象, 2-26
调整多边线或多边形的形状, 2-34
起始窗口, 2-103
起始窗口对话框, 2-103
透明位图, 2-51
透明颜色工具, 1-18, 2-51
通用工具栏, 2-10, 2-24
打开窗口工具, 2-16
打印工具, 1-16
关闭窗口工具, 2-17
保存所有窗口工具, 1-16
保存窗口工具, 2-16
复制对象工具, 1-16, 2-29, 2-30
恢复工具, 1-16, 2-27
剪切对象工具, 1-16, 2-30
粘贴对象工具, 1-16, 2-31
新建窗口工具, 1-17, 1-18, 1-19, 1-20, 1-21, 1-22
撤消工具, 1-16, 2-27
通用窗口对话框特性, 1-32
通讯协议
DDE, 9-2
FastDDE, 9-2
NetDDE, 9-2
SuiteLink, 9-2
配置
ActiveX 控件, 2-81
ActiveX 控件属性, 2-83
InSql 为历史供应器, 8-37
InTouch 为多监视器系统, 3-24
InTouch 的公共数据源, 3-10
NAD, 3-17
WindowMaker 属性, 2-6
WindowViewer, 2-97
WindowViewer 作为 NT 服务, 2-104, A-7
WindowViewer 窗口, 2-100
分布式历史记录, 8-42
分布式历史供应器, 8-36
分布式报警对象, 7-44
分布式报警对象, 7-33, 7-44, 7-47, 7-50, 7-52, 7-53
历史趋势 - 在运行时, 8-20
历史趋势对象, 8-8
文件的 UNC 路径, 3-6
节点属性, 3-17
闪烁对象速度, 2-99
动态分辨率转换, 3-22
网络应用程序开发, 3-17
网络资源, 3-6
网格, 2-45
报警服务器节点, 2-99
系统权限, A-5
系统闲置, 2-112
远程历史供应器, 8-39, 8-40, 8-41
实时趋势对象, 8-3
标题栏文本, 2-101
要记录的标记名, 8-15
窗口控件向导, 2-73
操作员口令, 2-109
操作员的安全级别, 2-108
配置用户对话框, 2-108
配置用户命令, 2-108
配置客户端设置, 10-27
验证脚本, 6-9
高度, 1-27
高速 Ethernet, 0-11

十一划

偏差, 7-28
偏差百分比, 4-31
减小圆角对象的半径, 2-33
减法 (-) 运算符, 6-49
剪切
对象到 Windows 剪贴板, 2-30
对象链接, 2-32
剪切对象工具, 1-16, 2-30
剪贴板, 0-5
副偏差, 4-31
域, 0-8
域名, 0-8
域名系统 (DNS), 0-9
域名空间, 0-16
基于服务器的结构, 3-4
基于客户的结构, 3-3
寄存器, 0-20
弹出类型窗口, 2-12
排列工具栏, 1-13, 1-21, 2-24, 2-36
中心点对齐工具, 1-21, 2-38
分解单元工具, 1-22, 2-44
分解符号工具, 1-21, 2-44
水平间距工具, 1-21, 2-40
水平翻转工具, 1-22, 2-42
右对齐工具, 1-21, 2-37
左对齐工具, 1-21, 2-36
改变对象形状工具, 1-22, 2-34
制作单元工具, 1-21, 2-44
制作符号工具, 1-21, 2-44
垂直间距工具, 1-21, 2-40
垂直居中对齐工具, 1-21, 2-38
垂直翻转工具, 1-22, 2-42
居中对齐工具, 1-21, 2-37

底部对齐工具, 1-21, 2-38
顶部对齐工具, 1-21, 2-37
逆时针旋转工具, 2-41
顺时针旋转工具, 1-22, 2-41
置后工具, 1-21, 2-39
置前工具, 1-21, 2-39
排列对象
水平间距, 1-21, 2-40
在窗口中, 2-36
垂直间距, 1-21, 2-40
置后, 1-21, 2-39
置前, 1-21, 2-39
控件名, 2-66
控件名称, 0-6
控制分布式报警的显示, 7-61
控制水平和垂直间距, 2-40
控制台, 0-6
控制异步 QuickFunction 脚本, 6-35
推广版 InTouch 许可证, 2-15
斜体工具, 1-18
断开会话连接, 10-26
旋转
文本对象, 2-53
位图, 2-46
图形对象, 2-41
水平翻转, 1-22, 2-42
垂直翻转, 1-22, 2-42
逆时针, 2-41
顺时针, 1-22, 2-41
图像, 2-46
检查影响系统性能的外围设备, 10-17
毫秒, 0-15
混合, 0-8
添加
向导到工具栏, 2-64
应用程序到应用程序浏览器, 1-10
添加访问名对话框, 9-7
添加颜色对话框, 1-26
清除
图形对象, 2-32
清除链接, 2-32
符号
分解, 1-21, 2-44
制作, 1-21, 2-44
符号对象, 2-3
粗体工具, 1-18
粘贴, 0-17
ActiveX 控件到窗口, 2-79
对象 - 从 Windows 剪贴板, 2-31
对象链接, 2-32
向导到窗口, 2-64
位图 - 从 Windows 剪贴板, 2-49
图形对象, 2-31
粘贴对象工具, 1-16, 2-31
脚本
ActiveX 事件, 6-2
ActiveX 事件脚本, 6-24
FOR-NEXT, 6-51
FOR-NEXT 循环脚本, 6-38, 6-39, 6-40, 6-41, 6-42
QuickFunction, 6-2, 6-27
异步, 6-33
有效语法, 6-31
参数表达式, 6-30
参数数据类型匹配, 6-31
数据类型, 6-30
文本
全部选定, 6-5
复制, 6-5
剪切, 6-5
粘贴, 6-6
比较
大于, 6-52
大于或等于, 6-52
小于, 6-52
小于或等于, 6-52
不等于, 6-52
等于, 6-52
打印脚本, 6-58
关键字
AS, 6-36
CALL, 6-27, 6-31
DIM, 6-36
RETURN, 6-27
创建 QuickFunction, 6-29
动作按钮, 6-2
动作触动按钮
按下时, 5-19
按下期间, 5-19
释放时, 5-19
向导, 6-2
在运行时停止, 2-101, 2-102
导入, 6-54
导入 ActiveX 事件脚本, 2-95
异步, 6-27
删除, 6-5
局部变量, 6-36
有效语法, 6-36
数据类型, 6-37
应用程序, 6-2
关闭时, 6-11
在运行期间, 6-11
启动时, 6-11
条件, 6-2
条件脚本

为真时, 6-20, 6-21
为真期间, 6-20, 6-21
为假时, 6-20, 6-21
为假期间, 6-20, 6-21
运算
取反, 6-46
取补, 6-46
运算
大于, 6-46
大于或等于, 6-46
小于, 6-46
小于或等于, 6-46
不等于, 6-46
加和连接, 6-46
右移位, 6-46
左移位, 6-46
同或, 6-46
异或, 6-46
位与, 6-46
取余数, 6-46
除, 6-46
乘, 6-46
乘方, 6-46
减, 6-46
逻辑 NOT, 6-46
逻辑或, 6-46
等于, 6-46
赋值, 6-46
运算符优先级
最低优先级, 6-47
最高优先级, 6-47
运算符描述
AND、OR 和 NOT, 6-50
比较 (<, >, <=, >=, ==, <>), 6-50
加 (+), 6-49
左移位 (SHL), 6-49
左移位 (SHR), 6-49
同或 (|), 6-50
异或 (^), 6-50
位与 (&), 6-49
取反 (-), 6-48
取余数 (MOD), 6-49
取补 (~), 6-48
括号 ( ), 6-48
除 (/), 6-49
乘 (*), 6-49
乘方 (**), 6-48
减 (-), 6-49
赋值 (=), 6-50
运算数, 6-44, 6-46
保存, 6-10
复杂脚本, 6-51
屏幕更新, 6-39
恢复, 6-10
指定等价键, 6-19
退出控制结构, 6-42
样式和语法, 6-44
验证, 6-9
基本语法, 6-44
脚本编辑器错误消息, 6-70
嵌套 FOR-NEXT 循环, 6-39
嵌套控制结构, 6-41
循环执行后的循环变量值, 6-41
插入
函数, 6-6
标记名, 6-8
窗口名, 6-9
窗口, 2-14, 6-2
窗口脚本
在显示期间, 6-13
显示时, 6-13
隐藏时, 6-13
数据改变, 6-2
脚本, 6-22
简单脚本, 6-45
触动按钮动作脚本, 6-17
键, 6-2
缩进/取消缩进文本, 6-4
撤消上次动作, 6-5
脚本：文本：清除, 6-6
脚本函数, 2-75, 6-68. *参见*函数
帮助, 6-7
类型
全部, 6-6
字符串, 6-6
系统, 6-6
附加程序, 6-7
其它, 6-7
数学, 6-6
脚本函数主题, 6-7
脚本编辑器
一般步骤, 6-4
菜单, 0-14
菜单栏, 0-14
象素, 0-18
隐藏时窗口脚本, 6-13
隐藏所有工具栏, 1-15
隐藏所有工具栏工具, 1-20
隐藏窗口触动按钮链接, 5-17, 5-21

十二划

属性, 0-19
动态, 2-2
静态, 2-2

嵌套 FOR-NEXT 循环, 6-39
嵌套控制结构, 6-41
强制记录参数, 8-19
循环执行后的循环变量值, 6-41
循环脚本
  FOR-NEXT, 6-51
插入, 0-18
插入点, 0-12
替代标记对话框, 4-96
替换
  文本, 2-58
  标记名, 4-103
替换文本对话框, 2-59, 4-103
替换字符串对话框, 2-57, 2-58
替换标记名, 4-96
替换类型窗口, 2-12
最大, 0-14
最小化, 0-15
椭圆工具, 1-19
游标链接, 5-15
登录, 0-13
  到应用程序, 2-110
登录对话框, 2-110
登录窗口, 2-111
硬盘速度, 10-17
确认, 7-5
确认分布式报警, 7-59
确认报警, 7-56
确认模型
  扩展的摘要, 4-28, 4-30
  条件, 4-28, 4-30
  面向事件, 4-28, 4-30
程序元
  工具栏, 1-13
  向导/ActiveX 工具栏, 1-17
  应用程序浏览器, 1-3
  状态栏, 1-24
  视图工具栏, 1-20
  绘图对象工具栏, 1-19
  格式工具栏, 1-17
  浮动/固定工具栏, 1-13
  调色板, 1-25, 2-51
  排列工具栏, 1-21
窗口, 0-25
  X 轴定位, 2-13
  Y 轴定位, 2-13
  打开, 2-16
  关闭, 2-17
  创建新窗口, 2-11
  导入, 2-23
  导入脚本, 2-23
  导出, 2-19
  删除, 2-17
  其它注释, 2-11
  命名, 2-11
  保存, 2-16
  复制, 2-18
  标题栏, 2-13
  类型, 2-12
  背景色, 2-12
  宽度, 2-13
  框式样, 2-13
  高度, 2-13
  维, 2-13
  脚本, 2-14
    导入, 2-23
窗口背景色工具, 1-18
窗口配置对话框, 2-100
窗口控件向导, 2-66, 2-67
  .Caption, 2-74
  .Enabled, 2-74
  .ListCount, 2-74
  .ListIndex, 2-74
  .NewIndex, 2-74
  .ReadOnly, 2-74
  .TopIndex, 2-74
  .Value, 2-74
  .Visible, 2-74
  下拉式列表组合框, 2-70
  下拉式组合框, 2-70
  列表框, 2-66, 2-67, 2-69
  使用准则, 2-68
  单选（选项）钮, 2-72
  单选钮, 2-66, 2-67
  组合框, 2-66, 2-67
  复选框, 2-66, 2-67
  选项钮, 2-66, 2-67
  获得和设置属性, 2-74
  配置, 2-73
  控件名, 2-66, 2-67
  属性, 2-74
  简单组合框, 2-70
窗口控件脚本函数
  wcSaveText(), 6-68
  错误消息, 6-75
窗口脚本, 6-2
  在显示期间, 2-14, 6-13
  显示时, 2-14, 6-13
  隐藏时, 2-14, 6-13
窗口脚本编辑器对话框, 6-13
窗口属性对话框, 2-11
等价键, 5-3, 6-19
缓冲区, 0-5
编辑
  Supertag 模板, 4-47
  文本对象, 2-57

编辑自定义颜色对话框, 2-52
裁剪, 0-7
赋值 (=) 运算符, 6-50
赋值语句, 0-2
赋值操作符, 0-2
趋势
  历史, 8-1
  历史趋势工具, 1-19, 8-7
  实时, 8-1
  实时趋势工具, 1-19, 8-2
趋势对象, 2-3
辉亮显示, 0-12
集中管理, 10-7

十三划

填充对象, 2-2
填充百分比链接, 5-34
填充色工具, 1-18
数学函数, 6-61
数学脚本函数, 6-6
数据包, 0-17
数据吞吐量, 9-2
数据库, 0-7
数据库视图, 7-166
数据库输入文件
  空白字符串, 4-127
  格式, 4-125
  操作模式, 4-129
数据改变脚本, 6-2, 6-22
数据改变脚本编辑器对话框, 6-22
数据时间质量, 3-7, 9-2
数据源地址, 3-10
新成员标记对话框, 4-39, 4-42, 4-43
新建窗口工具, 1-17, 1-18, 1-19, 1-20, 1-21, 1-22
新模板
  对话框, 4-37
新模板对话框, 4-41
滚动, 0-21
滚动条, 0-21
滚动框, 0-22
禁用
  ALT 键 – WindowViewer, 2-102
  CTRL-ESC 键 - WindowViewer, 2-102
简单对象, 2-2
  文本, 2-2
  线条, 2-2
  按钮, 2-2, 2-3
  填充图案, 2-2
简单组合框, 2-70
简单脚本, 6-45
置后工具, 1-21, 2-39
置位按钮, 5-18
置前工具, 1-21, 2-39
触动动作脚本编辑器对话框, 6-17
触动按钮动作脚本, 6-17
触动链接. 参见动画链接
触摸屏, 0-24
路由器, 3-8
路径, 0-18
路径名, 0-18
错误消息
  脚本, 6-70
  窗口控件脚本函数, 6-75
键
  键脚本
    放开时, 6-15
    按下时, 6-15
    按下期间, 6-15
键加速器, 0-13
键盘快捷方式, 1-36
键盘箭头键, 1-37
键脚本, 6-2
  放开时, 6-15
  按下时, 6-15
  按下期间, 6-15
键脚本编辑器对话框, 6-15

十四划

模式, 0-15
模拟, 7-28, 0-2
模拟值显示链接, 5-45
模拟输入链接, 5-13
瘦客户机, 10-4, 10-6, 0-24
磁盘操作系统, 0-8
端口, 0-18
算法, 0-2
精确对齐, 1-23
缩放, 0-21

十五划

增大圆角对象的半径, 2-33
增强实时趋势性能, 8-5
撤消
  对象编辑, 2-27
  层数, 1-16, 2-9
  堆栈, 2-9
撤消工具, 1-16, 2-27
撤消层数, 1-16, 2-9
撤消选定
  选定对象组, 2-27
颜色工具
  线, 1-18
  透明颜色, 1-18

窗口背景, 1-18
填充, 1-18

十六划

操作系统, 0-16
操作符, 0-16
操作数, 0-16
整型, 0-12
激活快速切换, 2-9
镜像, 0-15

十八划

翻转对象, 2-42
水平, 1-22, 2-42
垂直, 1-22, 2-42
翻转单元, 2-44
覆盖类型窗口, 2-12