

InTouch® HMI 数据管理指南

Invensys Systems, Inc.

修订版 A

最后修订日期：2007 年 8 月 6 日



版权声明

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

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

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

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

<http://www.wonderware.com>

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

商标

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

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

目录

欢迎.....	11
文档惯例.....	11
技术支持.....	12
第 1 章 数据管理概述	13
使用 InTouch 标记.....	14
InTouch 标记的类型	15
内存标记.....	16
I/O 标记.....	17
间接标记.....	18
其它标记.....	19
历史趋势标记	19
标记 ID 标记	19
SuperTag.....	19
废弃标记.....	20
标记属性.....	20
内存标记属性	21
I/O 标记属性	22
远程标记引用	24

第 2 章 使用标记名字典管理标记	25
规划标记的使用	26
创建新标记	27
配置标记属性	28
公共标记属性	28
标记名惯例	28
自动命名标记	29
标记注释	29
理解标记属性	30
值范围、度量单位及初始值	30
标记死区	30
标记值保留	32
I/O 连接	32
标记记录	33
创建离散标记	34
创建整型与实型标记	35
创建消息标记	36
创建 I/O 标记	36
修改标记	36
删除标记	37
打印标记列表与使用信息	37
第 3 章 系统标记	39
系统标记	39
第 4 章 使用标记点域来查看或更改标记属性	45
标记类型可用的点域	46
更改标记的值极限	54
查看原始值极限	55
.MinRaw 点域	55
.MaxRaw 点域	56
查看标记的原始值	57
.RawValue 点域	57
查看工程单位值极限	58
.MaxEU 点域	58
.MinEU 点域	59
更改标记的工程单位	60
.EngUnits 点域	60
按工程单位的格式查看标记值	61

.Value 点域	61
查看或更改离散标记消息	62
.OnMsg 点域	62
.OffMsg 点域	63
查看或更改标记的注释	64
.Comment 点域	64
第 5 章 使用 I/O 进行数据访问	65
支持的 InTouch 通讯协议	67
动态数据交换	67
Wonderware SuiteLink	67
SuiteLink 通讯问题疑难排解	67
设置访问名	68
删除访问名	71
使用 I/O 标记访问 I/O 数据	72
配置 I/O 标记的属性	72
指定离散 I/O 标记	72
指定整型与实型 I/O 标记	74
指定消息型 I/O 标记	76
设置 I/O 访问参数	76
在运行时检索 I/O 标记的有关信息	77
IOGetNode() 函数	77
IOGetApplication() 函数	78
IOGetTopic() 函数	78
在运行时动态更改 I/O 标记引用	79
.Reference 点域	79
IOSetItem() 函数	79
IOSetAccessName() 函数	81
将标记转换为远程引用	83
通过远程引用访问 I/O 数据	87
在运行时重定向远程引用	89
IOSetRemoteReferences() 函数	89
恢复引用	92
从 InTouch 中访问 Application Server 数据	93
将 Application Server 对象属性用于 InTouch 标记	93
从 InTouch 浏览应用程序服务器对象属性	94
Application Server 浏览器的限制	95
Application Server 对象中的特殊扩展	95
将 Application Server 数据类型映射到 InTouch 数据类型	97

Application Server 属性的读取 / 写入行为	99
配置 InTouch HMI 以便将 Galaxy 用作远程标记源	101
查看 I/O 标记的时间标签与质量信息	105
查看 I/O 标记的时间标签信息	105
.TimeDate 点域	105
.TimeDateString 点域	106
.TimeDateTime 点域	106
.TimeDay 点域	107
.TimeHour 点域	107
.TimeMinute 点域	108
.TimeMonth 点域	108
.TimeMsec 点域	109
.TimeSecond 点域	109
.TimeTime 点域	110
.TimeTimeString 点域	110
.TimeYear 点域	111
查看 I/O 标记的质量信息	112
质量数据格式	112
关于数据质量点域	113
.Quality 点域	113
.QualityLimit 点域	114
.QualityLimitString 点域	115
.QualityStatus 点域	115
.QualityStatusString 点域	116
.QualitySubstatus 点域	116
.QualitySubstatusString 点域	117
在运行时初始化与重置 I/O 连接	118
使用命令重新初始化 I/O 连接	118
使用脚本重新初始化 I/O 连接	120
IOReinitAccessName() 函数	120
IOReinitialize() 函数	121
IOStartUninitConversations() 函数	121
使用访问名的故障转移功能	122
配置故障转移	122
编辑故障转移对的访问名参数	123
删除访问名的故障转移	124
强制故障转移到备份访问名	124
故障转移表达式	124
IOForceFailover() 函数	125
临时禁用故障转移功能	126
禁用故障转移配置选项	126
IODisableFailover() 脚本函数	127

使用脚本检索故障转移的有关信息	128
IOGetAccessNameStatus() 函数	128
IOGetActiveSourceName() 函数	129
监视 I/O 连接的状态	130
使用 IOStatus 主题名	130
在 Excel 中使用主题名 IOStatus	132
监视 I/O 服务器通讯状态	132
从其它应用程序中访问 InTouch 标记数据	134
第 6 章 定义间接标记	135
在脚本中使用间接标记	136
给本地标记使用间接标记	137
给远程引用使用间接标记	138
第 7 章 定义可复用的标记结构	141
定义 SuperTag 模板	143
编辑 SuperTag 模板与成员标记	145
创建 SuperTag 实例	146
使用标记名字典创建 SuperTag 实例	146
使用标记名字典复制 SuperTag 实例	147
使用标记名字典将标记添加到 SuperTag 实例	148
创建 SuperTag 的其它方法	149
引用 SuperTag 成员	150
使用批量导入实用程序导入 SuperTag	150
第 8 章 减少标记使用	151
确定标记使用	152
确定标记计数	152
根据许可证确定最大远程标记数	153
查找标记的使用位置	154
保存与打印标记交叉引用列表	157
删除未使用的标记	160
第 9 章 记录标记值	161
配置历史记录功能	163
配置要记录历史的标记	163
配置常规记录属性	164

控制历史记录频率	166
在运行时启动与停止历史记录	167

第 10 章 绘制标记数据的趋势 169

InTouch 趋势的类型	170
理解历史趋势	170
理解实时趋势	171
在历史趋势中显示保存的标记值	171
使用历史趋势对象	171
创建历史趋势	172
配置要在历史趋势上显示的标记	173
配置历史趋势的时间跨度	175
配置历史趋势显示选项	176
在运行时更改趋势配置	177
使用点域控制历史趋势	179
.DisplayMode 点域	179
.MinRange 点域	180
.MaxRange 点域	181
.UpdateCount 点域	182
.UpdateInProgress 点域	183
.UpdateTrend 点域	184
.ChartLength 点域	185
.ChartStart 点域	186
.Pen1-8 点域	187
.TagID 点域	188
.ScooterLockLeft 点域	189
.ScooterLockRight 点域	190
.ScooterPosLeft 点域	191
.ScooterPosRight 点域	192
使用历史趋势向导	193
使用历史趋势向导创建趋势	194
配置要在趋势图上显示的标记	195
配置历史趋势时间跨度	195
配置显示选项	196
在运行时更改配置	197
使用脚本控制历史趋势向导	198
将趋势更新到当前时间	198
HTUpdateToCurrentTime() 函数	198
更改趋势配置	199
HTSelectTag() 函数	199

HTSetPenName() 函数	200
检索趋势与历史数据的有关信息	201
HTGetPenName() 函数	201
HTGetTimeAtScooter() 函数	202
HTGetTimeStringAtScooter() 函数	203
HTGetValue() 函数	204
HTGetValueAtScooter() 函数	205
HTGetValueAtZone() 函数	206
平移与缩放趋势	207
HTScrollLeft() 函数	207
HTScrollRight() 函数	208
HTZoomIn() 函数	209
HTZoomOut() 函数	210
打印趋势	211
PrintHT() 函数	211
排解趋势的错误	211
HTGetLastError() 函数	211
在趋势中显示实时值	212
使用实时趋势对象	213
创建实时趋势	213
配置要在实时趋势上显示的标记	214
配置实时趋势的时间跨度与更新速率	215
配置实时趋势的显示选项	216
在运行时打印趋势	218
配置趋势打印选项	218
显示来自其它 InTouch 节点或 Wonderware	
Historian 的历史标记值	219
配合使用 InTouch HMI 与 Wonderware Historian	220
对笔进行配置以显示远程趋势数据	221
使用标记浏览器将笔指定给远程历史供应器	222
使用 QuickScript 将笔指定给远程历史供应器	223

第 11 章 从其它应用程序访问历史标记值 225

使用 DDE 项目显示历史数据	226
使用 DDE 访问记录数据	228
使用 HistData 手工提取记录数据	228
创建 HistData “访问名”	228
创建 HistData 标记	229
创建 HistData 窗口	230
运行 HistData	232

使用历史数据向导来提取记录数据	232
从其它应用程序中访问历史数据	234
排解 HistData 错误	235
附录 A IEEE 十进制单位	237
在 InTouch HMI 中显示浮点数	238
附录 B InTouch 许可证	239
理解许可证标记计数	239
理解 InTouch 远程引用限制	242
远程标记计数函数	243
IORRGetSystemInfo() 函数	243
IORRWriteState() 函数	245
IORRGetItemActiveState() 函数	247
索引	249

欢迎

本文介绍如何采集、保存以及显示 InTouch 数据。

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

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

文档惯例

本文采用以下惯例：

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

技术支持

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

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

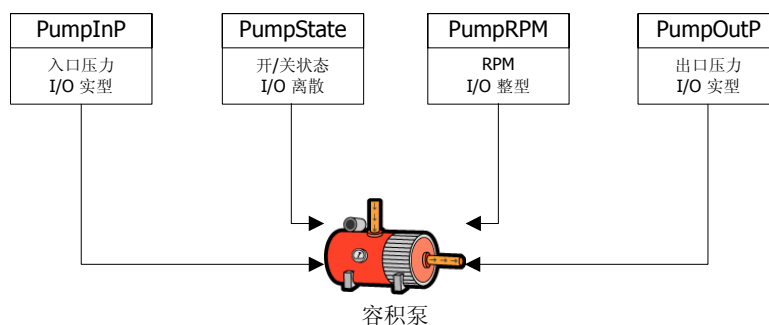
- 使用的操作系统的类型与版本。
- 有关如何重现问题的详细说明。
- 看到的错误消息的准确内容。
- Log Viewer 或任何其它诊断应用程序提供的任何相关输出列表。
- 为解决问题所作的尝试及其结果的详细说明。
- 如果遇到仍然存在的已知问题，请提供指定给该问题的“Wonderware 技术支持”案例号。

第 1 章

数据管理概述

Wonderware® InTouch® “人机界面” (HMI) 应用程序是生产环境中各种组件的图形化表示。工厂操作员使用这种图形界面来监视和管理生产过程。

下图显示一个泵的示例，它是生产过程中的一个组件。该泵有一些属性及关联的值。压力、RPM 及状态都是泵的属性，这些属性的值通过 HMI 进行监控。

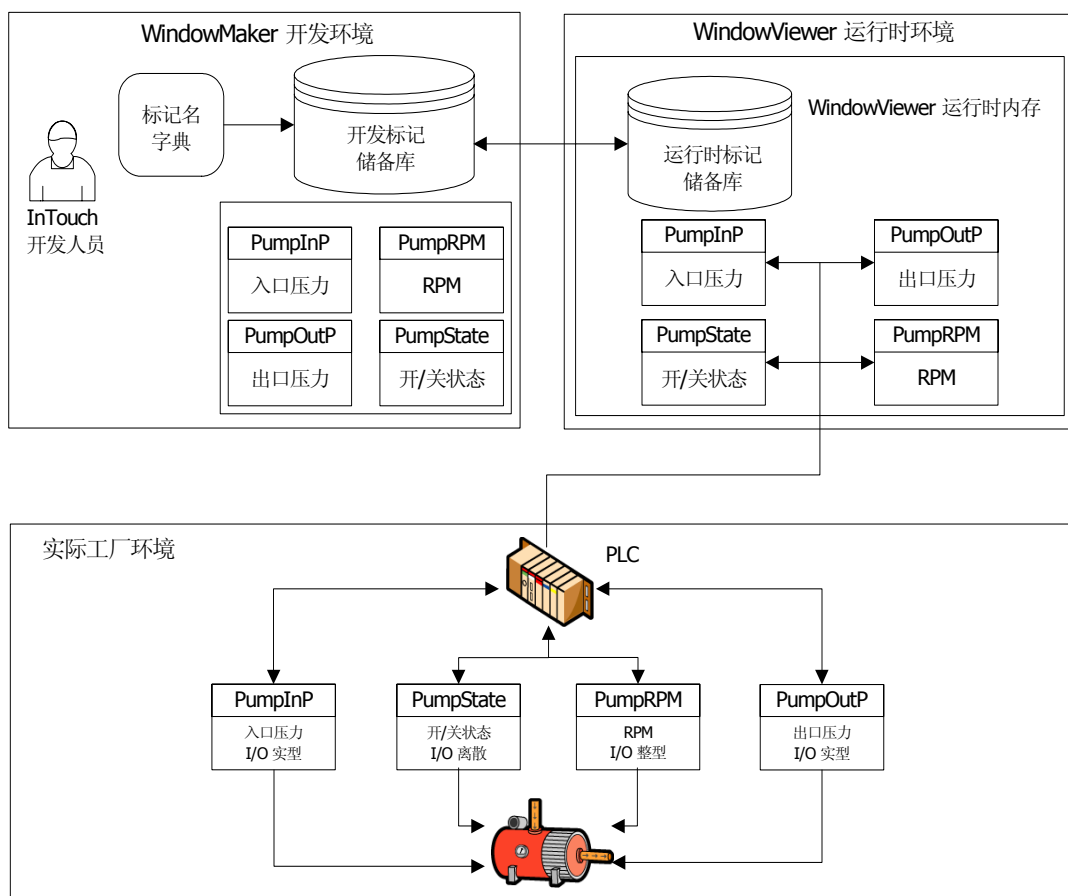


在 InTouch HMI 应用程序中，**标记** 代表数据项。通过使用标记，可以将特定的组件属性当作数据项从生产环境中访问。在上图中，**PumpState** 标记指出泵是打开还是关闭的。对于生产环境中的各种组件，如果要在 InTouch 应用程序中监视或控制其属性，则可以为它们创建标记。

对于从生产组件中采集的不同类型的数据，可以使用不同类型的标记。例如，PumpState 标记返回 On/Off 布尔值，以指出泵是正在运行还是已经停止。对于要成为应用程序一部分的数据类型，需要指定适当类型的 InTouch 标记。

使用 InTouch 标记

首先开始创建一个 InTouch 应用程序。然后使用 WindowMaker 中的“标记名字典”工具为应用程序定义标记。下图显示 InTouch 开发与运行时环境。



您使用“标记名字典”来指定标记的名称与类型。对于某些类型的标记，在“标记名字典”中还有其它一些选项，可以用于指定标记的附加属性。例如，I/O 型标记包含一些附加选项，用于指定与远程数据源的连接。

InTouch 应用程序在 WindowViewer 环境中运行。

WindowViewer 启动某个应用程序时，它从开发储备库中读取标记，然后将它们放入运行时内存。

InTouch 应用程序使用动画链接或脚本与放入运行时内存的标记进行通讯。InTouch 应用程序通过指定给标记的组件属性来跟踪当前值和其它状态信息。

InTouch 标记的类型

定义标记时，您根据所要关联的过程数据给标记指定特定的类型。例如，如果某个标记显示泵的 RPM，则将该标记的类型指定为整型标记。

在“标记名字典”中，使用**标记类型**对话框可以为创建的任何标记指定标记类型。



在指定标记类型之后，“标记名字典”列出所选标记类型的特定选项。

内存标记

内存标记定义 InTouch 应用程序中的内部系统常数与变量。例如，可以将某个内部常数定义为实数 3.414。在过程模拟中，内存标记可以充当计数器来控制后台 QuickScript 的动作。基于同该标记关联的计数，QuickScript 可以触发各种动画效果。内存标记也可以充当由其它程序来访问的计算变量。

基于同标记关联的过程数据，有四种类型的内存标记可以选择。

- **内存离散**

内存离散标记与过程组件的状态属性关联。指定给内存离散标记的值是两个可能的布尔状态，如：

- 0 或 1
- 假或真
- 开或关
- 高或低

- **内存整型（模拟）**

您可以将内存整型标记指定为 -2,147,483,648 与 2,147,483,647 之间的 32 位带符号整数。

- **内存实型（模拟）**

您可以将内存实型标记指定为 -3.4×10^{38} 与 3.4×10^{38} 之间的十进制浮点数。所有的浮点计算都按 64 位精度进行，但结果存储为 32 位十进制数。如需有关实数最大精度的详细信息，请参阅第 237 页的“IEEE 十进制单位”。

- **内存消息**

您可以将内存消息型标记指定为长度不超过 131 个单字节字符的文本字符串。

I/O 标记

I/O 标记从外部数据源中读取 InTouch 应用程序数据，或向其中写入数据。外部数据包含来自可编程控制器、过程计算机以及网络节点的输入和输出。I/O 标记数据值可以通过以下协议远程访问：

- Microsoft “动态数据交换” (Dynamic Data Exchange, 简称 DDE)
- Wonderware SuiteLink™

运行时内存中的 I/O 标记的值发生改变时，InTouch HMI 更新远程应用程序。反之，只要远程应用程序中相应数据项的值发生改变，便会更新 InTouch 中的 I/O 标记值。

根据与标记关联的过程数据，InTouch HMI 提供四种类型的 I/O 标记。这四种类型的 I/O 标记与内存标记类型非常类似。

- **I/O 离散**

I/O 离散标记与组件过程属性关联，这些属性的值使用两个可能的状态表示，如：

- 0 或 1
- 假或真
- 开或关
- 高或低

- **I/O 整型 (模拟)**

I/O 整型标记可以指定为 -2,147,483,648 与 2,147,483,647 之间的 32 位带符号整数。

- **I/O 实型 (模拟)**

I/O 实型标记可以指定为 -3.4×10^{38} 与 3.4×10^{38} 之间的十进制浮点数。所有的 I/O 实型标记浮点计算都按 64 位精度进行，但结果存储为 32 位数。如需有关 I/O 实数最大精度的详细信息，请参阅第 237 页的“IEEE 十进制单位”。

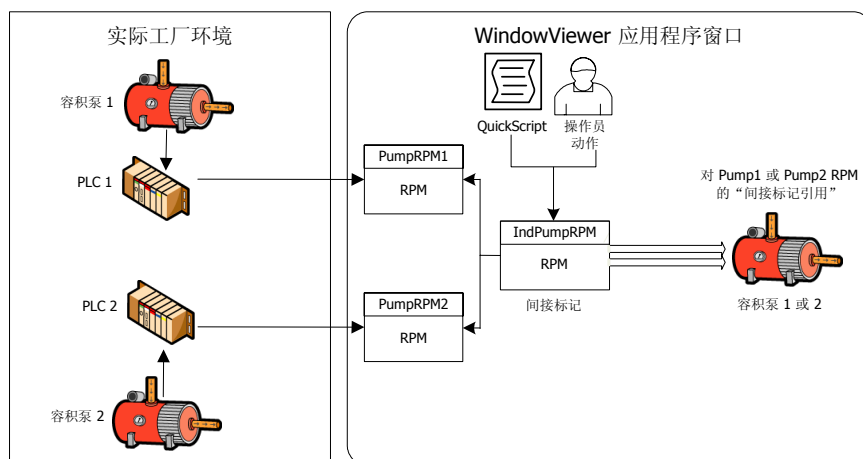
- **I/O 消息**

I/O 消息标记可以指定为最大长度不超过 131 个单字节字符的文本字符串。

间接标记

间接标记用作其它标记的“指针”。例如，您可以创建单个 InTouch 窗口，使用间接标记去显示来自多个不同组的标记的数据。

下图显示一个应用程序窗口示例，该窗口能够显示多个泵。您可以在一个窗口中使用间接标记来显示与单独的泵关联的不同源标记的值，而不必为每个泵创建单独的窗口。



QuickScript 或操作员动作将间接标记指向源标记。例如，以下脚本语句根据 PumpNo 标记的值，将两个 PumpRPM 标记指定给 IndPumpRPM 间接模拟标记。

```
IF PumpNo == 1 THEN
    IndPumpRPM.Name = "PumpRPM1";
ELSE
    IndPumpRPM.Name = "PumpRPM2";
ENDIF;
```

将某个间接标记的值赋给另一个源标记时，这个间接标记的作用就好像它自己是源标记。同原始的源标记和间接重复标记关联的这些值会一起进行同步。如果源标记的值发生改变，间接标记便会反映出这个变化。如果间接标记的值发生改变，源标记的值也会相应地改变。

您可以使用离散、模拟以及消息型的间接标记。这三种类型的间接标记类似于内存与 I/O 类型的标记。

如需有关间接标记的详细信息，请参阅第 135 页的“定义间接标记”。

其它标记

您可以使用针对特定、有限用途而设计的其它 InTouch 标记类型。这些标记可用于创建动态报警显示、创建历史趋势以及更改指定给历史趋势笔的标记。

历史趋势标记

“历史趋势”标记可用于引用历史趋势图。与历史趋势关联的所有“点域”都可以应用于“历史趋势”标记。

如需有关定义与使用“历史趋势”标记的详细信息，请参阅第 169 页的“绘制标记数据的趋势”。

标记 ID 标记

“标记 ID”标记从标记值绘制在 InTouch 历史趋势图上的标记中检索信息。通常，“标记 ID”标记可用于显示指定给特定趋势笔的标记的名称，或更改给趋势笔指定的标记。

您可以在 QuickScript 中包含一条语句，使用“标记 ID”类型的标记将某个新的标记指定给任何历史趋势中的笔。例如，以下 QuickScript 语句更改与历史趋势笔关联的标记：

```
HistTrend.Pen1=MyLoggedTag.TagID;
```

此 QuickScript 运行时，历史趋势的 Pen1 开始绘制 MyLoggedTag 的历史记录数据的趋势。

如需有关定义与使用“历史趋势”标记的详细信息，请参阅第 193 页的“使用历史趋势向导”。

SuperTag

SuperTag 是包含一系列相关标记的模板。例如，您可以创建一个 SuperTag 模板，该模板包含一系列指定给某个泵所有属性的标记。

在生产过程中有完全相同的设备时使用 SuperTag。您可以将 SuperTag 模板的实例指定给每个完全相同的过程项，而不必为设备中的每个组件都创建一组标记。

如需有关 SuperTag 的详细信息，请参阅第 141 页的“定义可复用的标记结构”。

废弃标记

通过使用“组变量”标记，可以利用 InTouch 的标准报警系统来创建动态报警显示、动态磁盘日志、动态打印。保留“组变量”标记仅为了与使用 InTouch 7.11 及更早版本开发的应用程序保持后向兼容性。请勿在使用比 InTouch 7.11 更新的版本开发的应用程序中使用“组变量”标记。

标记属性

每个 InTouch 标记类型都有一组属性，用于描述与标记关联的数据特性。与 InTouch 标记关联的四种主要数据类型有：

- 离散值
- 整数
- 实数
- 文本消息

在使用“标记名字典”创建标记时，所有的标记属性都会赋一个初始值。下图显示 I/O 整型标记的属性。

The screenshot shows the 'Tag Dictionary' (标记名字典) dialog box with the 'Main' (主要) tab selected. The tag name is 'PumpRPM' and its group is '\$System'. The type is 'I/O Integer' (I/O 整型) and it is set to 'Read/Write' (读写). The initial value (初始值) is 0, and the deadband (死区) is 0. The engineering unit (工程单位) is 'RPM'. The 'Confirm Model' (确认模型) is set to 'Condition' (条件). The 'Alarm Value' (报警值) is 0, and the 'Alarm Priority' (报警优先级) is 1. The 'Alarm Constraint Tag' (报警约束标记) is empty. The 'Alarm Comment' (报警注释) is empty. The 'Value Range' (值死区) is 0. The 'Change Rate' (变化率) is 0, and the 'Change Unit' (变化单位) is '%'. The 'Change Priority' (变化优先级) is 1, and the 'Change Constraint Tag' (变化约束标记) is empty.

从“标记名字典”中设置标记属性的初始值之后，可以使用点域来动态更改大多数标记属性。点域确定 InTouch 应用程序运行时可以由脚本监视或修改的标记属性。在脚本中可以将点域附加到标记的名称。

如需有关使用点域来动态更改标记属性的详细信息，请参阅第 45 页的“使用标记点域来查看或更改标记属性”。

内存标记属性

下表列出四种内存标记的属性。作为“标记字典”的选项，可以选择或修改每种属性。如需有关详细信息，请参阅第 27 页的“创建新标记”。

标记属性	离散	整型	消息	实型
% 偏差		●		●
%		●		●
确认模型	●	●		●
报警注释	●	●	●	●
报警组	●	●	●	●
报警约束	●	●		●
报警状态	●			
报警值		●		●
注释	●	●	●	●
死区		●		●
工程单位		●		●
偏差死区百分比		●		●
High		●		●
HiHi		●		●
初始值	●	●	●	●
记录数据	●	●		●
记录死区		●		●
记录事件	●	●	●	●
Lo		●		●
LoLo		●		●
最大长度			●	
主偏差		●		●
最大值		●		●
最小值		●		●
副偏差		●		●
关闭消息	●			
打开消息	●			

标记属性	离散	整型	消息	实型
优先级	●	●		●
变化率		●		●
只读	●	●	●	●
读写	●	●	●	●
保留参数		●		●
保留值	●	●	●	●
目标		●		●
值死区		●		●

I/O 标记属性

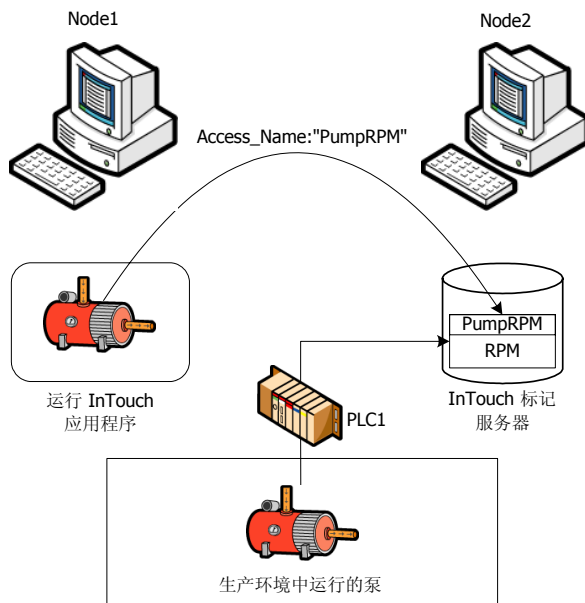
与内存标记类似，作为“标记名字典”的选项，可以选择或修改 I/O 标记属性。如需有关详细信息，请参阅第 27 页的“创建新标记”。

标记属性	离散	整型	消息	实型
% 偏差		●		●
%		●		●
访问名	●	●	●	●
确认模型	●	●		●
报警注释	●	●	●	●
报警组	●	●	●	●
报警约束	●	●		●
报警状态	●			
报警值		●		●
注释	●	●	●	●
转换		●		●
死区		●		●
工程单位				
偏差死区百分比		●		●
工程单位		●		●
High		●		●

标记属性	离散	整型	消息	实型
HiHi		●		●
初始值	●	●	●	●
输入转换	●			
项目	●	●	●	●
记录数据	●	●		●
记录死区		●		●
记录事件	●	●	●	●
Lo		●		●
LoLo		●		●
最大长度			●	
主偏差		●		●
最大工程单位		●		●
最大原始数据		●		●
最大值		●		●
最小工程单位		●		●
最小原始数据		●		●
最小值				
副偏差		●		●
关闭消息	●			
打开消息	●			
优先级	●	●		●
变化率		●		●
只读	●	●	●	●
读写	●	●	●	●
保留参数		●		●
保留值	●	●	●	●
平方根转换				
目标		●		●
将标记名用作项目名	●	●	●	●
值死区		●		●

远程标记引用

您可以创建分布式 InTouch 应用程序，让标记服务器在运行 InTouch 应用程序的节点之外的单独节点上运行。下图显示一个远程引用 PumpRPM 标记的 InTouch 应用程序，该标记来自另一个节点上运行的标记服务器。



创建 InTouch 应用程序引用远程节点上的标记有两种方法：

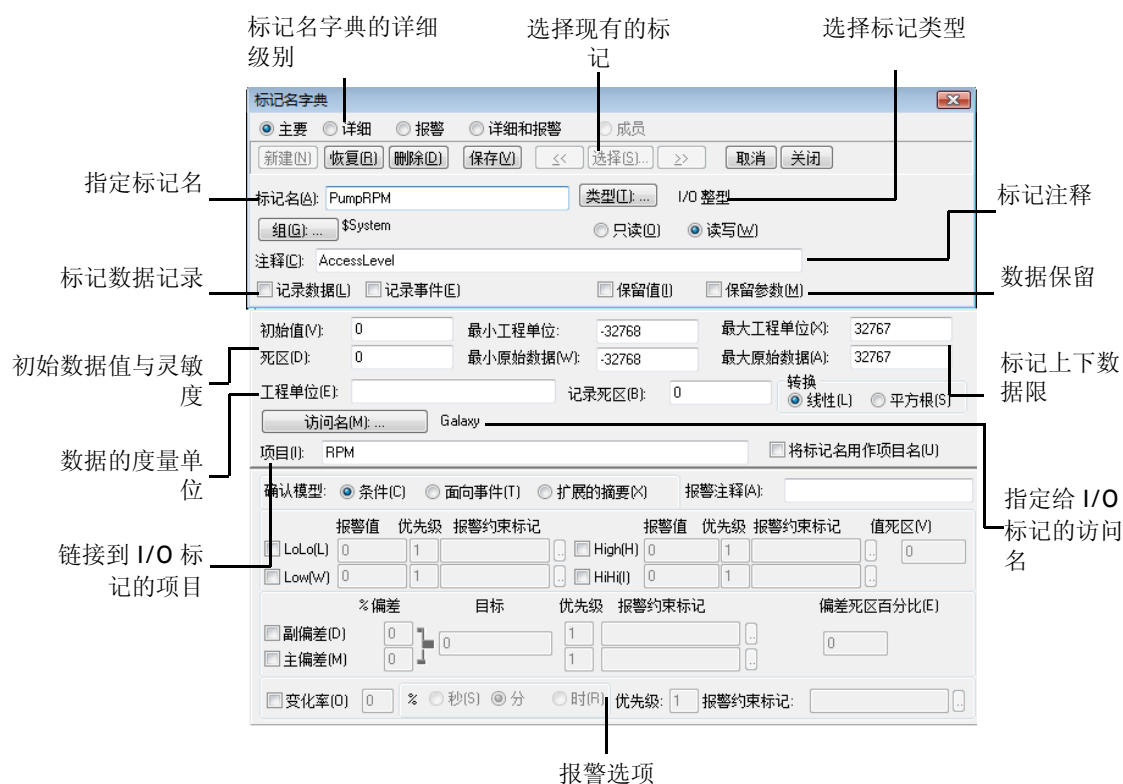
- 将 I/O 标记关联到将远程服务器指定为标记源的“访问名”。如需有关为 I/O 标记定义“访问名”的详细信息，请参阅第 68 页的“设置访问名”。
- 使用直接指向该“标记”的远程引用。例如，PLC1:PumpRPM。

如需有关详细信息，请参阅第 87 页的“通过远程引用访问 I/O 数据”。

第 2 章

使用标记名字典管理标记

通过使用“标记名字典”，可以为 InTouch 应用程序创建标记。
下图显示**标记名字典**对话框，它包含用于定义 I/O 标记属性的所有选项。



规划标记的使用

在最初的规划阶段，可以通过确定应用程序标记的关键需求来缩短开发时间。全面的规划可以缩短创建 InTouch 应用程序所需的时间。

在创建标记之前：

- 确定要在 InTouch 应用程序中呈现的过程的所有物理组件。

创建要在应用程序中代表数据源的组件属性的列表。

- 确定与每个组件属性关联的数据类型。

根据与组件属性关联的数据给每个标记指定一种标记类型。如需有关给标记指定数据类型的详细信息，请参阅第 27 页的“创建新标记”。

- 确定要合并到 InTouch 应用程序中的数据的特征。

评估每个标记的以下数据特征：

- 期望的数据值范围
- 指定给数据值的度量单位
- 初始数据值
- 死区值，用于设置将数据值视为已发生改变的阈值
- 要在标记值状态改变时显示的消息

如需有关定义标记数据特征的详细信息，请参阅第 30 页的“理解标记属性”。

- 制订标记命名惯例与标准。

复杂的应用程序通常要求使用许多标记。制订标准化的命名惯例，就应用程序中标记的组织结构提出相应的建议。如需有关标记命名惯例的详细信息，请参阅第 28 页的“标记名惯例”。

- 确定要保存的过程数据。

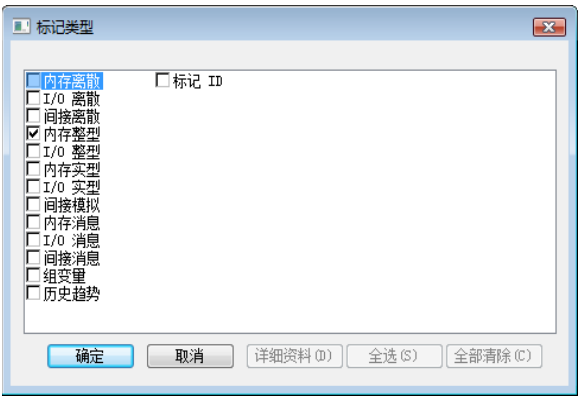
所选数据保存到日志文件。您可以使用记录的数据来创建历史趋势，以显示标记值随时间变化的情况。如需有关设置标记记录功能的详细信息，请参阅第 33 页的“标记记录”。

创建新标记

您可以使用 WindowMaker 的**标记名字典**创建标记。在开始之前，请分析工厂流程以确定需要在 InTouch 应用程序中创建的标记。

要创建新标记

- 1 在 WindowMaker 中打开 InTouch 应用程序。
- 2 在**特别**菜单上，单击**标记名字典**。
第一次打开“标记名字典”时，**标记名**框中出现 \$AccessLevel 系统标记的定义。在保存新标记之后，**标记名字典**显示最近保存的标记定义。
- 3 执行以下操作：
 - a 单击**新建**。此时清除**标记名**框。
 - b 输入新标记的名称。如需有关标记命名要求的详细信息，请参阅第 28 页的“标记名惯例”。
 - c 作为可选项，在**注释**框中输入关于新标记的注释。
- 4 单击**类型**。此时出现**标记类型**对话框，包含所支持的 InTouch 标记类型的列表。



- 5 从列表中选择标记类型，然后单击**确定**。此时再次出现“标记名字典”，并显示所选的标记类型。
- 6 如果需要，单击**详细**以查看所选标记类型的附加“标记名字典”选项。
- 7 在**标记名字典**对话框中，进一步指定各个标记选项。
如需有关指定标记属性的详细信息，请参阅第 28 页的“配置标记属性”。
- 8 单击**保存**。单击**关闭**以关闭**标记名字典**对话框。

配置标记属性

通过使用**标记名字典**对话框，可以指定每个标记定义中都会有的公共标记属性。您必须给每个标记指定一个名称。您可以添加可选的注释。标记名与注释都是所有标记的公共属性。



每种类型的 InTouch 标记都有唯一的数据属性。在选择标记类型之后，**标记名字典**对话框展开，根据所选的标记类型显示一组选项。

公共标记属性

您必须给每个标记指定唯一的名称。可选的注释可以是标记定义的一部分。给定义的每个标记指定适当的注释是一种很好的做法。

所有标记都属于某个报警组，报警组是另一个公共的标记属性。缺省条件下，所有标记都属于 **\$System** 报警组。如需有关将标记指定给其它报警组的详细信息，请参阅 *InTouch® HMI 报警与事件指南* 中的第 2 章“配置报警”。

标记名惯例

给标记命名时，如果需要许多包含相似属性的标记，请使用一致的命名惯例。

遵循以下这些 InTouch 标记名的命名惯例：

- 标记名不超过 32 个字符。
- 使用字母数字（A-Z、a-z、0-9）作为标记名的第一个字符。

标记名最好仅使用字母数字字符。

- 标记名至少使用一个英文字符。

(可选) 使用以下特殊字符:

- 连字符	! 感叹号	# 井号
\$ 美元号	% 百分号	& 与号
? 问号	@ At 号	_ 下划线

如果可能, 请避免在标记名中使用特殊字符, 除非应用程序确实需要。

- 避免在标记名中使用连字符 (-)。

连字符是 InTouch 标记名的有效字符。但是在逻辑或算术表达式中, InTouch 将连字符用作负号或减号运算符。例如, 表达式 $A=B-C$ 可以解释为 $A = B$ 减 C , 但也可以说是将名称为 $B-C$ 的标记指定给标记 A 。

- 请勿在标记名中使用空格。
- 请勿在标记名中使用可以解释为指数的数字。

例如, 不能将标记命名为 $125E4$, 因为它可以解释成底数的四次方。

- 请勿在标记名中使用可以解释为十六进制数的数字。

例如, 不能将标记命名为 $0x123B$, 因为它可以解释成十六进制数。

自动命名标记

在“标记名字典”中给标记命名时, InTouch HMI 跟踪所用的命名惯例。如果将标记命名为 Pump01、Pump02, 则 InTouch HMI 会建议将下一个标记命名为 Pump03。您可以接受或拒绝此名称。此命名帮助功能称为“自动编号”。

“自动编号”基于标记名中的最后一个连续数字。例如, 如果标记名是 PumpInP04LotB99A, 则 InTouch HMI 建议将下一个标记命名为 PumpInP04LotB100A, 而不是 PumpInP05LotB99A。

标记注释

创建标记时, 可以在标记名字典的注释框中, 输入最多 50 个字符的可选注释。

第一次访问“标记名字典”时, \$AccessLevel 系统标记的缺省注释出现在注释框中。删除此注释, 以防止它与创建的任何标记关联。

理解标记属性

在指定公共标记属性之后，必须定义正在创建的标记类型所特有的其它属性。下表按照标记类型显示内存标记的基本属性。

标记类型	唯一属性
离散	初始值、打开消息、关闭消息、注释
整型	初始值、最小值、死区、工程单位、最大值、记录死区、注释
实型	初始值、最小值、死区、工程单位、最大值、记录死区、注释
消息	最大长度、初始值、注释

I/O 标记又一些额外的属性，可用于建立网络通讯以及将网络设备上的原始数据转换成 InTouch 应用程序所用的规格化值。如需有关定义 I/O 标记的详细信息，请参阅第 72 页的“配置 I/O 标记的属性”。

值范围、度量单位及初始值

在 WindowViewer 中启动 InTouch 应用程序时，离散、整型、实型以及消息标记都会指定一个初始值。如果是离散标记，初始值是可能的二进制状态之一。对于整型与实型标记，初始值是应用程序启动时与该标记关联的数字。在“标记名字典”中指定初始值。

您可以将初始标记值指定为应用程序在 WindowViewer 中停止运行时标记的最后一个值。通过从“标记名字典”中选择保留值选项，可以将标记的最后一个活动值指定为应用程序再次启动时的初始值。

整型与实型标记包含一些属性，这些属性分别设置指定给标记的可能数值范围的上下限。整型与实型标记均包含定义范围上下限的最小值与最大值属性。

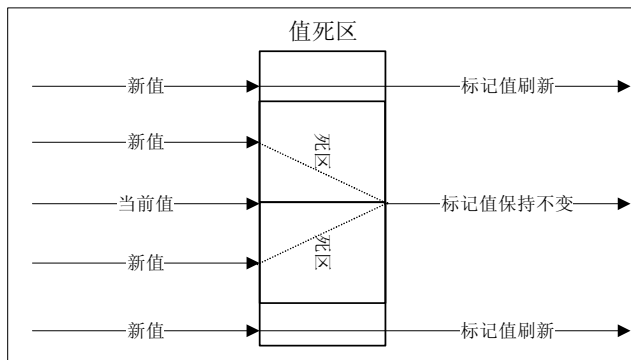
整型与实型标记还包含工程单位属性，可以指定一个工程单位标签来描述标记值的度量单位。例如，您可以将 PSI 指定为与泵压关联的整型标记的工程单位属性。

标记死区

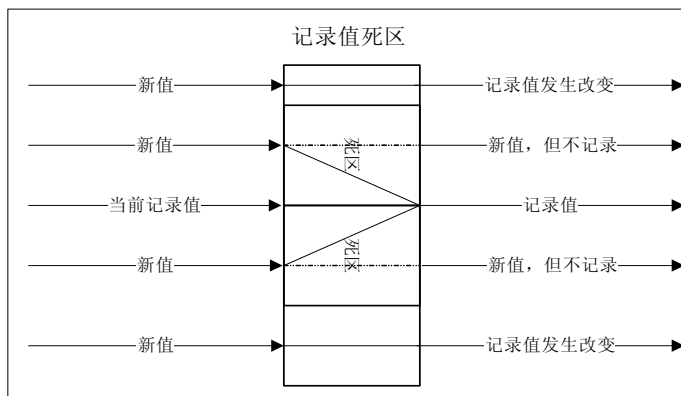
死区是标记值的灵敏度设置。死区通常与其值持续改变的 I/O 标记关联。死区过滤掉标记值中细小的瞬间改变，可以减少 InTouch 的数据处理量。

对于同整型与实型数据关联的标记，“标记名字典”包含两个死区属性。

- 值死区：值死区属性设置一个阈值；必须超过这个阈值，**WindowViewer** 才会刷新运行时内存中的标记值。下图显示当前标记值周围的绝对死区范围。



- 记录死区：记录死区设置一个阈值；必须超过这个阈值，才会将标记值写入日志文件。下图显示所记录的标记的当前值周围的死区。



只有死区以外的新标记值才会写入日志文件。死区范围内值的微小改变将会被忽略，而不会记录下来。

要设置标记死区

- 1 打开**标记名字典**对话框。
- 2 单击**选择**。此时出现**选择标记**对话框。当前为应用程序定义的标记会列出来。
- 3 从列表中选择整型或实型标记。
- 4 单击**确定**。选择实型或整型标记时，**标记名字典**对话框的详细资料部分显示一些附加的选项。

初始值(I):	0	最小值:	0	死区(D):	15
工程单位(E):	RPM	最大值(M):	2500	记录死区(B):	25

- 5 根据所选的标记类型，通过在**死区**框中输入整数或实数来设置值死区。
值死区设置的是以工程单位为单位的一个绝对阈值水平。
- 6 根据所选的标记类型，通过在**记录死区**框中输入整数或实数来设置记录死区。
同值死区类似，记录死区设置的是以工程单位为单位的一个绝对阈值。
- 7 单击**保存**以保存对死区所作的更改。
- 8 单击**关闭**以关闭“标记名字典”对话框。

标记值保留

标记名字典的详细资料部分包含两个属性，可以保留标记值和操作员对报警限所作的更改。

所有标记类型都包含一个**保留值**属性。选择**保留值**可以保留应用程序停止时标记的当前值。再次启动应用程序时，**WindowViewer** 将保留下来的值用作该标记的初始值。

WindowViewer 再次启动应用程序时，**WindowViewer** 不会将保留的值写入 I/O 设备。在“**I/O 服务器**”初次扫描提供数据的设备之后，I/O 值便会更新。

如果 **WindowViewer** 在运行，则无法为新的标记或现有的标记选择或清除**保留值**选项。选择此选项时，标记的初始值会不断更新，以反映它的当前值。**WindowViewer** 停止时，初始值设置为最后一个保留值。如果随后清除了此选项，则该标记的初始值设置为最后一个保留值。

整型与实型标记包含**保留参数**属性。选择**保留参数**可以保留应用程序运行时操作员对标记的报警限所作的任何更改。重新启动应用程序时，**WindowViewer** 将修改后的报警限用作报警限的初始值。

I/O 连接

所有类型的 I/O 标记都必须指定外部数据源的“访问名”与“项目名”。如需有关为 I/O 标记指定“访问名”与“项目名”的详细信息，请参阅第 76 页的“设置 I/O 访问参数”。

标记记录

在运行时，每次标记值的改变超过指定的记录死区时，WindowViewer 便可以将一个项目写入历史日志文件。WindowViewer 也可以按照固定的间隔将一些项目写入日志，而不论当前标记值如何。缺省条件下，这个固定间隔是一小时。

备注 如果需要更加可控、功能更多样的记录功能，可以考虑使用 Wonderware Historian 来存储 InTouch 历史数据。

标记名字典对话框包含一些单独的选项，可以将数据与事件记录到日志文件中。您可以设置一些值记录选项。如需有关设置事件记录功能的详细信息，请参阅 *InTouch® HMI 报警与事件指南* 中的第 2 章“配置报警”。

对于要写入历史日志文件的标记值，必须启用历史记录。如需有关设置常规记录属性的详细信息，请参阅第 163 页的“配置历史记录功能”。

对于整型与实型标记，可以在它们各自的详细资料对话框中设置“记录死区”。**记录死区**选项指定标记的值必须改变多少个工程单位，才会写入一个日志项。

要给标记配置记录功能

- 1 打开**标记名字典**。
- 2 选择要将其数据保存到日志文件的标记。
- 3 选择**记录数据**。

☐ 记录数据(L) ☐ 记录事件(E) ☐ 保留值(V) ☐ 保留参数(M)

- 4 如果希望记录由操作员、I/O、QuickScript 或操作系统对标记值所作的改变，请选择**记录事件**。在选择**记录事件**之后，会出现**优先级**框。

☒ 记录数据(L) ☒ 记录事件(E) 优先级: 999 ☐ 保留值(V) ☐ 保留参数(M)

优先级值确定标记的事件优先级。有效值是 1 到 999；其中 1 是最高优先级，999 是最低优先级。

- 5 单击**保存**，然后关闭“**标记名字典**”。

创建离散标记

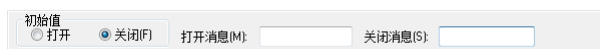
您可以指定离散标记，以显示在 InTouch 应用程序中运行的内部过程的二进制状态。离散标记必须指定一个初始值，即打开或关闭。此外，也可以指定过程与进入或脱离某种报警状态的标记转换关联时，在报警事件窗口中出现的消息。

以下步骤显示如何定义内存离散标记。I/O 离散标记指出可编程控制器、过程计算机的所有输入与输出以及网络节点数据的二进制状态。

如需有关设置 I/O 离散标记属性的详细信息，请参阅第 72 页的“指定离散 I/O 标记”。

要定义内存离散标记的初始值与消息

- 1 在**标记类型**对话框中，选择**内存离散**作为标记的类型。
- 2 如果需要，选择**标记名字典**对话框顶部的**详细**，以显示详细选项。此时出现**标记名字典**对话框的详细资料部分。



- 3 选择**打开**或**关闭**作为与标记关联的初始值。应用程序启动时，标记设置为这个初始值。
- 4 在**打开消息**与**关闭消息**框中输入标记进入或脱离某种报警状态时出现的消息。

这些消息可用在任何动画链接或脚本中，而不论该标记是否配置了报警。

- 如果定义在标记值等于 1 (On, True) 时处于活动状态的离散报警，则在**打开消息**框中输入的消息会出现在 ActiveX 报警显示的**值与报警限**列中。

标记的报警状态恢复正常时，在**关闭消息**框中输入的消息会出现在**值**列中，“打开消息”保持在**报警限**列中。

- 如果定义在标记值等于 0 (Off, False) 时处于活动状态的离散报警，则在**关闭消息**框中输入的消息会出现在 ActiveX 报警显示的**值与报警限**列中。

标记的报警状态恢复正常时，在**打开消息**框中输入的消息会出现在**值**列中，“关闭消息”保持在**报警限**列中。

- 5 将更改保存到标记中。

创建整型与实型标记

您可以指定整型与实型标记以显示 InTouch 应用程序中运行的过程的数值。

以下步骤显示如何定义内存与 I/O 整型和实型标记。内存整型与实型标记必须指定一个初始值。您还必须设置最小与最大数据范围。

如需有关设置 I/O 整型与实型标记的 I/O 属性的详细信息，请参阅第 74 页的“指定整型与实型 I/O 标记”。

重要 InTouch 实数的精度限制为最多八位。要避免可能发生的舍入错误，指定实型标记的属性时，不要超过八位精度。如需有关详细信息，请参阅第 237 页的“IEEE 十进制单位”。

要定义内存整型与实型标记值

- 1 在**标记类型**对话框中，将内存整型或内存实型指定为标记的类型。此时出现**标记名字典**对话框的详细资料部分。

初始值(I):	0	最小值:	0	死区(D):	15
工程单位(E):	PSI	最大值(M):	1500	记录死区(R):	25

- 2 设置整型与实型标记的属性。执行以下操作：
- 在**初始值**框中，输入应用程序启动时与标记关联的整数或实数。
 - 在**最小值**框中，输入标记的最小整数或最小实数。
最小值设置与内存整型与实型标记关联的数字可能的最小值。
 - 在**最大值**框中，输入标记的最大整数或最大实数。
最大值设置与内存整型与实型标记关联的数字可能的最大值。
 - 在**工程单位**框中，输入要用作标记工程单位的标签。
- 3 将更改保存到标记中。
- 如需有关设置标记的死区与记录死区属性的详细信息，请参阅第 30 页的“标记死区”。

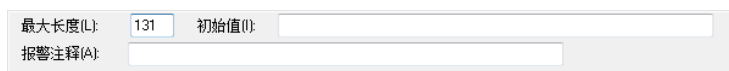
创建消息标记

您可以为内部与外部过程指定消息标记。这些标记包含一些属性，可以指定 WindowViewer 启动应用程序时出现的初始消息，以及可以从 Alarm Viewer 读取的注释。

如需有关定义 I/O 消息标记的详细信息，请参阅第 76 页的“指定消息型 I/O 标记”。

要定义内存消息标记

- 1 在**标记类型**对话框中，将内存消息指定为标记的类型。
- 2 如果需要，选择**详细**以显示**标记名字典**对话框的详细资料部分。



- 3 设置内存消息标记的属性。执行以下操作：
 - 在**最大长度**框中，输入一个整数，它是标记的消息中可以出现的最大字符数。或者，接受缺省长度 131 个字符，即消息的最大长度。
 - 在**初始值**框中，输入要在 WindowViewer 启动应用程序时指定给标记的消息文本。
 - 如果选择了消息标记的**记录事件**选项，则在**报警注释**框中，输入可以在 AlarmViewer 控件中读取的消息。
- 4 将更改保存到标记中。

创建 I/O 标记

I/O 标记包含一组公共属性，可以指定 InTouch 应用程序与外部过程之间的网络连接。**标记名字典**对话框的详细资料部分包含一些选项，可用于设置 I/O 标记的外部属性。

如需有关设置 I/O 属性的详细信息，请参阅第 65 页的“使用 I/O 进行数据访问”。

修改标记

修改标记同创建标记类似。您从“标记名字典”中选择要修改的标记。然后，执行与创建标记相同的步骤来更改标记的属性。

重要 在 InTouch 应用程序中使用了标记之后，要再修改它的类型并不容易。可选择的标记类型可能要局限于类似的数据类型。定义标记时，应该仔细选择正确的数据类型。

要修改标记

- 1 打开**标记名字典**对话框。

- 2 单击**选择**。此时出现**选择标记**对话框。当前为应用程序定义的一列标记会显示出来。
- 3 从列表中选择要修改的标记。
- 4 单击**确定**。此时**标记名字典**对话框显示为所选标记指定的值。
- 5 更改标记的属性。
- 6 单击**保存**以使用所作的更改来更新标记。
- 7 单击**关闭**以关闭“标记名字典”。

删除标记

系统会为给 InTouch 应用程序定义的所有标记维护一个计数。删除包含动画链接或脚本的窗口时，标记计数不会自动减少。与删除的窗口关联的标记被视作仍在使用的，且无法删除。

要删除不再使用的标记，必须关闭 WindowViewer，并更新本地与远程标记的使用计数。通过使用 InTouch 的“交叉引用”实用程序，可以确定标记的使用位置。如需有关使用“交叉引用实用程序”与更新标记计数的详细信息，请参阅第 151 页的“减少标记使用”。

在更新使用计数之后，可以删除标记。如需有关详细信息，请参阅第 160 页的“删除未使用的标记”。

打印标记列表与使用信息

在 WindowMaker 中，可以使用“WindowMaker 打印件”实用程序打印 InTouch 应用程序数据库、窗口及脚本的内容。

如需有关详细信息，请参阅第 157 页的“保存与打印标记交叉引用列表”。

第 3 章

系统标记

通过使用系统标记，可以提供与系统相关的信息和用于 InTouch 脚本的标准函数，例如日期与时间。系统标记是所有应用程序的一部分。

在“标记名字典”中，系统标记通过将美元号 (\$) 用作标记名的第一个字符进行标识。系统标记无法删除。您只能更改与系统标记关联的注释。

系统标记

下表介绍 InTouch 系统标记：

系统标记	用途	详细信息
\$AccessLevel	只读整型标记，指定与当前登录的操作员关联的访问级别。此信息可以在动画链接或脚本中用于控制操作员对特定 InTouch 功能的访问。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$ApplicationChanged	只读离散标记，指出 NAD 环境中的主应用程序是否已发生更改。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 2 章“分发应用程序”。
\$ApplicationVersion	只读实型标记，指定 WindowViewer 中运行的应用程序的当前版本号。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 2 章“分发应用程序”。

系统标记	用途	详细信息
\$ChangePassword	只写离散标记，设置为 1 时显示改变口令对话框。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$ConfigureUsers	只写离散标记，显示通用的配置用户对话框，以编辑安全性用户名列表。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$Date	只读整型标记，显示自 1970 年 1 月 1 日以来已过天数的整数部分。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。
\$DateString	只读消息标记，按照 Windows 区域和语言选项对话框中指定的格式显示日期。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。
\$DateTime	只读实型标记，显示自 1970 年 1 月 1 日以来已过天数的小数部分。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。
\$Day	只读整型标记，显示本月的当前日期值 (1-31)。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。
\$False	离散只读标记在表达式中返回 FALSE 值。从较早版本的 InTouch 应用程序更新到当前版本时，\$False 系统标记用于替换任何废弃系统标记的实例。	无进一步信息。
\$HistoricalLogging	可读写离散标记，用于在运行 InTouch 应用程序时开始与停止历史记录功能。	请参阅第 9 章“记录标记值”。
\$Hour	只读整型标记，按 0 到 23 的形式显示当前小时值。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。

系统标记	用途	详细信息
\$InactivityTimeout	只读离散标记，表示用户不活动期限已过。\$InactivityTimeout 的值可用于自动注销操作员。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$InactivityWarning	只读离散标记，表示不活动状态警告期限已过。 \$InactivityWarning 的值可用于向操作员发出不活动状态警告。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$Language	可读写整型标记，指定 InTouch 应用程序中所显示语言的语言 ID。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 6 章“在运行时切换语言”。
\$LogicRunning	可读写离散标记，用于启动与停止应用程序脚本。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$Minute	只读整型标记，显示当前的分钟值 (0-59)。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。
\$Month	只读整型标记，显示当前月份 (1-12)。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。
\$Msec	只读整型标记，显示当前毫秒值 (0-999)。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。
\$NewAlarm	可读写离散标记，表示发生新本地报警的时间。	请参阅 <i>InTouch® HMI 报警与事件指南</i> 中的第 6 章“在运行时控制标记与组的报警属性”。

系统标记	用途	详细信息
\$ObjHor	只读整型标记，显示屏幕上所选对象中心的水平像素位置。	请参阅 <i>InTouch® HMI 可视化指南</i> 中的第 4 章“设置对象动画效果”。
\$ObjVer	只读整型标记，显示屏幕上所选对象中心的垂直像素位置。	请参阅 <i>InTouch® HMI 可视化指南</i> 中的第 4 章“设置对象动画效果”。
\$Operator	只读消息标记，显示登录到 InTouch 应用程序的操作员的姓名。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$OperatorDomain	只读消息标记，包含应用程序在使用基于操作系统的安全性进行登录时所指定的域或机器名。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$OperatorDomainEntered	只写消息标记指定了操作员登录尝试 InTouch 应用程序的域。 您只有在将值指定给 \$PasswordEntered 系统标记后，登录尝试才会启动。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$OperatorEntered	可读写消息标记，指定尝试登录 InTouch 应用程序的操作员用户帐户名。 您只有在将值指定给 \$PasswordEntered 系统标记后，登录尝试才会启动。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$OperatorName	只读消息标记，在使用基于操作系统或 ArchestrA® 的身份验证时显示操作员的全名。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。

系统标记	用途	详细信息
\$PasswordEntered	只写消息标记，指定尝试登录 InTouch 应用程序的操作员的口令。 将值写入此标记时，使用 \$OperatorDomainEntered、\$OperatorEntered 及 \$PasswordEntered 系统标记尝试进行登录。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$Second	只读整型标记，显示当前秒值 (0-59)。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。
\$StartDdeConversations	可读写离散标记，用于在运行时启动未初始化的对话。	请参阅第 5 章“使用 I/O 进行数据访问”。
\$System	只读标记，用于确定根报警组。	请参阅 <i>InTouch® HMI 报警与事件指南</i> 中的第 1 章“报警与事件综述”。
\$Time	只读整型标记，显示自当天午夜起经过的毫秒数。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。
\$TimeString	只读消息标记，按照 Windows 区域和语言选项对话框中指定的格式显示当前时间。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。
\$VerifiedUserName	只读消息标记，包含经验证的用户的全名，或包含空值。	请参阅 <i>InTouch® HMI 应用程序管理与扩展指南</i> 中的第 5 章“保护 InTouch 安全”。
\$Year	只读整型标记，按四位数字的格式显示当前年份。	请参阅 <i>InTouch® HMI 脚本与逻辑指南</i> 中的第 6 章“内置函数”。

第 4 章

使用标记点域来查看或更改标记属性

每种类型的 InTouch 标记都有一组独特的属性，这些属性描述与标记关联的数据或可能的条件。点域确定标记属性。“标记名字典”中显示的差不多每个标记属性都有点域。某些点域是每种类型的 InTouch 标记所共有的。例如 .Name 点域总是与标记的名称关联。另一些点域则仅适用于特定标记类型的独特属性。

通过在脚本、表达式或用户输入中使用点域，可以在应用程序运行期间监视与修改标记的属性。下例显示在脚本或表达式中使用点域来访问标记属性的语法。

```
tag_name.property_dotfield
```

例如，要让操作员能够在应用程序运行期间更改 HiHi 报警限，可以创建一个“模拟 - 用户输入”触动链接。然后，将该链接应用于使用 Analog_Tag.HiHiLimit 点域表达式所定义的按钮。在运行期间，操作员可以单击该按钮，并为标记输入新的 HiHi 报警限值。

通过使用点域，可以输入和输出与标记关联的数据；此外，通过使用历史点域，可以修改正在运行的应用程序当前显示的历史趋势。例如，您可以在脚本中使用点域，让操作员能够修改历史趋势滚动方式、锁定或调整趋势上指示器的位置，或是将笔重新指定给新的标记。

标记类型可用的点域

每种类型的 InTouch 标记都有一组与它的独特属性关联的点域。
下表按字母顺序列出所有标记类型的点域。

点域	标记类型													
	内存				I/O				间接			其它		
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件
.Ack	●	●	●		●	●	●		●	●		●		
.AckDev		●	●			●	●			●		●		
.AckDsc	●				●				●			●		
.AckROC		●	●			●	●			●		●		
.AckValue		●	●			●	●			●		●		
.Alarm	●	●	●		●	●	●		●	●		●		
.AlarmAccess														●
.AlarmAckModel	●	●	●		●	●	●		●	●				
.AlarmClass														●
.AlarmComment	●	●	●	●	●	●	●	●	●	●	●	●	●	●
.AlarmDate														●
.AlarmDev		●	●			●	●			●		●		
.AlarmDevCount		●	●			●	●			●		●		
.AlarmDevDeadband		●	●			●	●			●				
.AlarmDevUnAckCount		●	●			●	●			●		●		
.AlarmDisabled	●	●	●		●	●	●		●	●		●		
.AlarmDsc	●				●				●			●		
.AlarmDscCount	●				●				●			●		
.AlarmDscDisabled	●				●				●			●		
.AlarmDscEnabled	●				●				●			●		
.AlarmDscInhibitor	●				●				●			●		
.AlarmDscUnAckCount	●				●				●			●		
.AlarmEnabled	●	●	●		●	●	●		●	●		●		

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件	标记 ID
.AlarmGroup														●	
.AlarmGroupSel														●	
.AlarmHiDisabled		●	●			●	●			●					
.AlarmHiEnabled		●	●			●	●			●					
.AlarmHiHiDisabled		●	●			●	●			●					
.AlarmHiHiEnabled		●	●			●	●			●					
.AlarmHiHiInhibitor		●	●			●	●			●					
.AlarmHiInhibitor		●	●			●	●			●					
.AlarmLimit														●	
.AlarmLoDisabled		●	●			●	●			●					
.AlarmLoEnabled		●	●			●	●			●					
.AlarmLoInhibitor		●	●			●	●			●					
.AlarmLoLoDisabled		●	●			●	●			●					
.AlarmLoLoEnabled		●	●			●	●			●					
.AlarmLoLoInhibitor		●	●			●	●			●					
.AlarmMajDevDisabled		●	●			●	●			●					
.AlarmMajDevEnabled		●	●			●	●			●					
.AlarmMajDevInhibitor		●	●			●	●			●					
.AlarmMinDevDisabled		●	●			●	●			●					
.AlarmMinDevEnabled		●	●			●	●			●					
.AlarmMinDevInhibitor		●	●			●	●			●					
.AlarmName														●	
.AlarmOprName														●	
.AlarmOprNode														●	
.AlarmPri														●	

点域	标记类型													
	内存				I/O				间接			其它		
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件
.AlarmProv														●
.AlarmROC		●	●			●	●			●				
.AlarmROCCount		●	●			●	●			●				
.AlarmROCDisabled		●	●			●	●			●				
.AlarmROCEnabled		●	●			●	●			●				
.AlarmROCIhibitor		●	●			●	●			●				
.AlarmROCUnAckCount		●	●			●	●			●				
.AlarmState														●
.AlarmTime														●
.AlarmTotalCount	●	●	●		●	●	●		●	●		●		
.AlarmType														●
.AlarmUnAckCount	●	●	●		●	●	●		●	●		●		
.AlarmUserDefNum1	●	●	●		●	●	●		●	●		●		
.AlarmUserDefNum1Set	●	●	●		●	●	●		●	●		●		
.AlarmUserDefNum2	●	●	●		●	●	●		●	●		●		
.AlarmUserDefNum2Set	●	●	●		●	●	●		●	●		●		
.AlarmUserDefStr	●	●	●		●	●	●		●	●		●		
.AlarmUserDefStrSet	●	●	●		●	●	●		●	●		●		
.AlarmValDeadband		●	●			●	●			●				
.AlarmValue			●				●			●		●		●
.AlarmValueCount		●	●			●	●			●		●		
.AlarmValueUnAckCount		●	●			●	●			●		●		
.Caption														●
.ChartLength													●	
.ChartStart													●	

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象 / 控件	标记 ID
.Comment	●	●	●	●	●	●	●	●	●	●	●	●	●		●
.DevTarget		●	●			●	●			●					
.DisplayMode													●		
.Enabled														●	
.EngUnits		●	●			●	●			●					
.Freeze														●	
.HiHiLimit		●	●			●	●			●					
.HiHiSet		●	●			●	●			●					
.HiHiStatus		●	●			●	●			●					
.HiLimit		●	●			●	●			●					
.HiSet		●	●			●	●			●					
.HiStatus		●	●			●	●			●					
.ListChanged														●	
.ListCount														●	
.ListIndex														●	
.LoLimit		●	●			●	●			●					
.LoLoLimit		●	●			●	●			●					
.LoLoSet		●	●			●	●			●					
.LoLoStatus		●	●			●	●			●					
.LoSet		●	●			●	●			●					
.LoStatus		●	●			●	●			●					
.MajorDevPct		●	●			●	●			●					
.MajorDevSet		●	●			●	●			●					
.MajorDevStatus		●	●			●	●			●					
.MaxEU		●	●			●	●			●					

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件	标记 ID
.MaxRange													●		
.MaxRaw		●	●			●	●			●					
.MinEU		●	●			●	●			●					
.MinorDevPct		●	●			●	●			●					
.MinorDevSet		●	●			●	●			●					
.MinorDevStatus		●	●			●	●			●					
.MinRange													●		
.MinRaw		●	●			●	●			●					
.Name	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
.NewIndex														●	
.NextPage														●	
.Normal	●	●	●		●	●	●		●	●		●			
.NumAlarms														●	
.OffMsg	●				●				●						
.OnMsg	●				●				●						
.PageNum														●	
.Pen1 至 .Pen8													●		
.PendingUpdates														●	
.PrevPage														●	
.PriFrom														●	
.PriTo														●	
.Quality	●	●	●	●	●	●	●	●	●	●	●				
.QualityLimit	●	●	●	●	●	●	●	●	●	●	●				
.QualityLimitString	●	●	●	●	●	●	●	●	●	●	●				
.QualityStatus	●	●	●	●	●	●	●	●	●	●	●				

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象 / 控件	标记 ID
.QualityStatusString	●	●	●	●	●	●	●	●	●	●	●				
.QualitySubstatus	●	●	●	●	●	●	●	●	●	●	●				
.QualitySubstatusString	●	●	●	●	●	●	●	●	●	●	●				
.QueryState														●	
.QueryType														●	
.RawValue	●	●	●		●	●	●		●	●		●			
.ReadOnly														●	
.Reference	●	●	●	●	●	●	●	●	●	●	●				
.ReferenceComplete	●	●	●	●	●	●	●	●	●	●	●				
.ROCPct		●	●			●	●			●					
.ROCSet		●	●			●	●			●					
.ROCStatus		●	●			●	●			●					
.ScooterLockLeft													●		
.ScooterLockRight													●		
.ScooterPosLeft													●		
.ScooterPosRight													●		
.Successful														●	
.SuppressRetain														●	
.TagID	●	●	●		●	●	●		●	●					
.TimeDate	●	●	●	●	●	●	●	●	●	●	●				
.TimeDateString	●	●	●	●	●	●	●	●	●	●	●				
.TimeDateTime	●	●	●	●	●	●	●	●	●	●	●				
.TimeDay	●	●	●	●	●	●	●	●	●	●	●				
.TimeHour	●	●	●	●	●	●	●	●	●	●	●				
.TimeMinute	●	●	●	●	●	●	●	●	●	●	●				

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象 / 控件	标记 ID
.TimeMonth	●	●	●	●	●	●	●	●	●	●	●				
.TimeMsec	●	●	●	●	●	●	●	●	●	●	●				
.TimeSecond	●	●	●	●	●	●	●	●	●	●	●				
.TimeTime*	●	●	●	●	●	●	●	●	●	●	●				
.TimeTimeString	●	●	●	●	●	●	●	●	●	●	●				
.TimeYear	●	●	●	●	●	●	●	●	●	●	●				
.TopIndex														●	
.TotalPages														●	
.UnAck	●	●	●		●	●	●		●	●		●			
.UpdateCount													●		
.UpdateInProgress													●		
.UpdateTrend													●		
.Value(Tagname)	●	●	●	●	●	●	●	●	●	●	●	●			●
.Value (Windows 控件)														●	
.Visible														●	

点域可以按其目标功能进行分类。如需有关点域功能类别的详细信息，请参阅以下几节：

类别	参阅
值与极限	第 54 页的“更改标记的值极限”。
报警参数	<i>InTouch® HMI 报警与事件指南</i> 中的第 6 章“在运行时控制标记与组的报警属性”
I/O	第 65 页的“使用 I/O 进行数据访问”。
分布式报警对象	<i>InTouch® HMI 报警与事件指南</i> 中的附录 A “使用分布式报警显示对象”
趋势显示	第 169 页的“绘制标记数据的趋势”。
窗口控件	<i>InTouch® HMI 可视化指南</i> 中的第 5 章“向导”

更改标记的值极限

来自 “I/O 服务器” 的原始输入数据会转换为适合 InTouch 应用程序的值范围。标记的值限定在由最小值与最大值确定的范围之间，它们分别由 “标记名字典” 中的**最小原始数据**与**最大原始数据**属性指定。

随后，这些原始值会转换成由**最小工程单位**与**最大工程单位**选项设置的工程单位范围。您可以使用一组点域来监视与修改标记的原始值与工程单位范围。

下表列出在应用程序运行期间监视或更改标记值的点域。

点域	可读写	显示
.MinRaw	只读	从 “I/O 服务器” 收到的原始值的低钳位设置。
.MaxRaw	只读	从 “I/O 服务器” 收到的原始值的高钳位设置。
.MinEU	只读	指定给标记的最小工程单位值。
.MaxEU	只读	指定给标记的最大工程单位值。
.EngUnits	可读写	从 “标记名字典” 的 工程单位 选项中给模拟标记指定的文本值。
.RawValue	只读	标记从 “I/O 服务器” 收到的实际离散或模拟值（未应用缩放）。
.Value	可读写	标记的当前值。
.OnMsg	可读写	离散标记的求值结果为 True 、 On 或 1 时指定给离散标记的消息。
.OffMsg	可读写	离散标记的求值结果为 False 、 Off 或 0 时指定给离散标记的消息。
.Comment	可读写	从 “标记名字典” 中指定的标记注释。

查看原始值极限

对于来自“I/O 服务器”的原始输入标记值，可以在 InTouch 应用程序中使用之前，可能需要进行钳位处理。钳位功能将原始值限制到所定义的上、下限范围内。**.MinRaw** 与 **.MaxRaw** 点域显示原始输入范围的上、下边界。

.MinRaw 点域

.MinRaw 点域显示给标记指定的“最小原始数据”下钳位设置。**.MinRaw** 点域的值来自在“标记名字典”中给 I/O 标记指定的**最小原始数据**值。任何低于此设置的原始值都会调整到这个最小值。

类别

标记

用法

tag_name.MinRaw;

参数

Tag_name

任何 I/O 整型、I/O 实型、及间接模拟标记的名称。

附注

这个只读的点域显示给“最小原始数据”下钳位设置指定的值。

数据类型

实型或整型（只读）。

有效值

任何模拟值。

示例

如果与泵入口压力关联的原始值超出由标记的**最小原始数据**与**最大原始数据**属性所设置的上、下值边界，则以下脚本显示一个错误窗口。

```
IF ((PumpInP.RawValue > PumpInP.MaxRaw) OR
    (PumpInP.RawValue < PumpInP.MinRaw)) THEN
    Show "Instrument Failure Window";
ENDIF;
```

另请参阅

.EngUnits, **.MinEU**, **.MaxEU**, **.MaxRaw**, **.RawValue**

.MaxRaw 点域

.MaxRaw 点域显示从“标记名字典”中的**最大原始数据**属性指定给 I/O 标记的“最大原始数据”的上钳位设置。任何超出此设置的原始数据值均限定为这个最大原始数据值。

类别

标记

用法

Tag_name.MaxRaw

参数

Tag_name

任何 I/O 整型、I/O 实型、及间接模拟标记的名称。

附注

这个只读的点域显示指定给“最大原始数据”上钳位设置的值。

数据类型

实型或整型（只读）。

有效值

任何模拟值。

示例

此脚本确定标记值是否超出正常操作范围，如果发生这种情况，则显示一个窗口。

```
IF ((Temp01.RawValue > Temp01.MaxRaw) OR
    (Temp01.RawValue <
    Temp01.MinRaw)) THEN
    Show "Instrument Failure Window";
ENDIF;
```

另请参阅

.EngUnits, .MinEU, .MaxEU, .MinRaw, .RawValue

查看标记的原始值

.RawValue 点域显示从 “I/O 服务器” 接收的所监视属性的实际离散或模拟值。原始值是在应用钳位与缩放来将值规格化为标记工程单位之前的实际输入值。

.RawValue 点域

.RawValue 点域显示 WindowViewer 从 “I/O 服务器” 接收的实际值。.RawValue 点域让您访问在 InTouch 应用缩放之前的 I/O 标记值。

类别

标记

用法

Tag_name.RawValue

参数

Tag_name

任何 I/O 整型、I/O 实型、I/O 离散、间接离散、及间接模拟标记的名称。

附注

这个只读点域用于显示 InTouch 应用缩放之前的实际离散或模拟 I/O 值。

数据类型

任何适合与 .RawValue 点域关联的标记类型的数据。例如，实型标记使用实数，离散标记使用离散值（只读）。

示例

原始泵入口压力低于或高于标记的最小与最大钳位极限时，以下脚本发出一则警告消息。

```
IF ((PumpInP.RawValue > PumpInP.MaxRaw) OR
    (PumpInP.RawValue < PumpInP.MinRaw)) THEN
    AlarmMessage = "Pump sensor is out of calibration or
    requires replacement.";
ENDIF;
```

另请参阅

.EngUnits, .MinEU, .MaxEU, .MinRaw, .MaxRaw

查看工程单位值极限

来自 “I/O 服务器” 的值在第一次进入 WindowViewer 时被视为原始数据。原始值可能需要缩放。InTouch HMI 在原始数据限定的输入值上执行算术转换，以便将它们缩放到标记的工程单位范围内。I/O 整型与实型标记类型包含**最小工程单位**与**最大工程单位**属性，分别显示工程单位范围的上、下边界。

.MaxEU 点域

.MaxEU 点域显示在 “标记名字典” 中给指定的标记所指定的最大工程单位值。

类别

标记

用法

Tag_name.MaxEU

参数

Tag_name

任何整型、实型或间接模拟标记。

附注

.MaxEU 点域用于将原始数据值缩放到给标记定义的工程单位范围内。它定义工程单位范围的上限。

数据类型

实型标记使用实型，整型标记使用整型（只读）。

有效值

依赖于指定的标记的类型。

示例

现场有一个 “可编程逻辑控制器” 来读取液位仪的数据。液位传送器发送 4 至 20 mA 范围内的一个信号。PLC 将这个信号转换为 0 到 4095 之间的整数值。这个值随后指定给 TankTwoLevel 标记。

显示原始数据值（0 到 4095 之间）并不能向操作员提供有用的数据。必须将此值缩放到适当的工程范围。

要完成这项工作，必须正确设置“最小工程单位”与“最大工程单位”字段。在本例中，如果原始数据值 0（现场读数是 4mA）转换为“0 加仑”，4095 这个值（现场读数是 20mA）转换为“100 加仑”，则需要进行以下设置以便在屏幕上显示正确的值：

```
TankTwoLevel.MinRaw = 0;
```

```
TankTwoLevel.MaxRaw = 4095;
```

```
TankTwoLevel.MinEU = 0;
```

```
TankTwoLevel.MaxEU = 100;
```

通过这些设置，现场的原始数据值为 4095 时，显示给操作员的值为 100。

另请参阅

.EngUnits, .MinEU, .MinRaw, .MaxRaw, .RawValue

.MinEU 点域

.MinEU 点域显示在“标记名字典”中给指定的标记所指定的最小工程单位值。

类别

标记

用法

```
Tag_name.MinEU
```

参数

Tag_name

任何整型、实型或间接模拟标记。

附注

.MinEU 点域用于将原始数据值缩放到为标记定义的工程单位范围内。它定义工程单位范围的下限。

数据类型

实型标记使用实型，整型标记使用整型（只读）。

有效值

依赖于指定的标记的类型。

示例

本例将给 Tag1 标记定义的工程单位范围指定给 AbsoluteTagRange 标记。

```
AbsoluteTagRange = (Tag1.MaxEU - Tag1.MinEU);
```

另请参阅

.EngUnits, .MaxEU, .MinRaw, .MaxRaw, .RawValue

更改标记的工程单位

您可以将点域与标记关联起来，以便确定指定给标记的工程单位的文本值。

.EngUnits 点域

.EngUnits 点域显示指定给有**工程单位**属性的模拟标记的文本值。 .EngUnits 点域将工程单位显示为文本值。

备注 写入此点域的值不可保留。

类别

标记

用法

Tag_name.EngUnits

参数

Tag_name

任何整型、实型或间接模拟标记。

数据类型

消息（可读写）。

附注

.EngUnits 点域不影响与标记关联的实际数据的缩放、转换或格式。

有效值

任意包含 0 到 31 个字符的字符串。

示例

如果标记的工程单位是摄氏度，则以下脚本调用华氏度转换函数。

```
IF Temperature.EngUnits == "Celsius" THEN
    CALL TempFConvert (Temperature);
ENDIF;
```

另请参阅

.MinEU, .MaxEU, .MinRaw, .MaxRaw, .RawValue

按工程单位的格式查看标记值

应用程序中没有明确指定点域时，给 InTouch 标记指定缺省的点域。

.Value 点域

.Value 点域按工程单位显示指定的标记的当前值。.Value 是隐式应用于所有标记的缺省 InTouch 点域。如果没有给标记指定点域，则缺省使用 .Value 点域。

类别

标记

用法

tag_name.Value

参数

tag_name

除“历史趋势”标记之外的任意类型的标记。

附注

很少需要使用 .Value 点域。不过在有些情况下，它可以使得计算或参数的使用更加明确。

数据类型

与指定的标记的类型相同（可读写）。

有效值

依赖于指定的标记的类型。

示例

下面的语句将内存整型 PumpRPM 标记的值设为等于 100：

```
PumpRPM.Value=100;
```

这在功能上等价于：

```
PumpRPM=100;
```

查看或更改离散标记消息

.OnMsg 与 .OffMsg 点域显示从“标记名字典”中给离散标记的开或关状态指定的消息。离散标记的打开与关闭消息是最多包含 15 个字符的短字符串。

.OnMsg 点域

.OnMsg 点域让您访问从“标记名字典”中给离散标记指定的“打开消息”。

类别

标记

用法

Tag_name.OnMsg

参数

Tag_name

任意离散标记。

数据类型

消息（可读写）。写入此点域的值不可保留。

有效值

任意包含 0 到 15 个字符的字符串。

示例

如果给间接 IndPumpState 标记的“打开消息”指定的字符串值是 "Pump1 running"，则下面的语句发出一条消息。

```
IF IndPumpState.OnMsg == "Pump1 running" THEN
    TypeOfTag = "The IndPumpState tag is assigned to
    Pump1.";
ENDIF;
```

另请参阅

.OffMsg

.OffMsg 点域

.OffMsg 点域让您可以访问从“标记名字典”中给离散标记指定的“关闭消息”。

类别

标记

用法

Tag_name.OffMsg

参数

Tag_name

任意离散标记。

数据类型

消息（可读写）。写入此点域的值不可保留。

有效值

任意包含 0 到 15 个字符的字符串。

示例

下面的语句根据 MyDiscrete 标记的状态将适当的字符串指定给 StateMessage 标记。

```
StateMessage=Dtext (MyDiscrete, MyDiscrete.OnMsg,  
    MyDiscrete.OffMsg);
```

另请参阅

.OnMsg

查看或更改标记的注释

只有从“标记名字典”中才能对标记注释作出永久更改。在应用程序运行时，可以使用 `.Comment` 点域指定另一个注释。这仅更改运行时会话期间的注释。它不会在“标记名字典”中永久性更改该标记的注释。关闭并重新启动 `WindowViewer` 之后，便会将原始注释指定给该标记。

`.Comment` 点域

`.Comment` 点域显示从“标记名字典”中给标记指定的注释。标记注释可以是最多包含 50 个字符的字符串。

类别

标记

用法

`Tag_name.Comment`

参数

`Tag_name`

任意标记名。

附注

在 `InTouch` 应用程序运行期间，使用 `.Comment` 点域可以修改与标记关联的注释。在应用程序停止之后，仍然将从“标记名字典”中指定的原始注释指定给标记。

数据类型

消息

有效值

任意 1 到 50 个字符的字符串。

示例

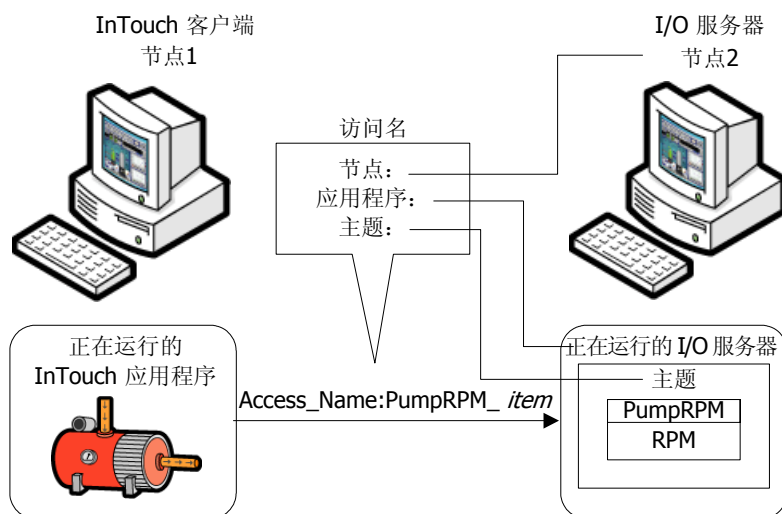
下面的语句通过将给标记指定的注释与标记的名称结合起来，而创建一条操作员消息。

```
OperatorMessage=PumpRPM.Name + " has a comment of:  
" + PumpRPM.Comment;
```


第 5 章

使用 I/O 进行数据访问

您可以开发分布式应用程序，使其中的各个 InTouch 系统功能组件位于不同的节点上。下图显示如何为另一个节点上存储的数据配置 I/O 请求。



您可以设置 InTouch 应用程序，使之能够使用第三方寻址惯例来确定另一个节点上存储的数据。此惯例包含节点、应用程序及主题名。要从远程节点获取数据，需要为 InTouch 应用程序配置指定这三个项目的“访问名”。

例如，如果希望从正在另一个节点上运行的远程 “I/O 服务器” 访问数据，“访问名” 需要包含以下内容：

访问名选项	描述
节点名	运行 “I/O 服务器” 程序的计算机的节点名。
应用程序名	节点上运行的 “I/O 服务器” 程序的名称。例如，DASMTEthernet 确定 Wonderware Mitsubishi Ethernet DAServer 应用程序名。如需有关与 Wonderware DAServer 关联的应用程序名的详细信息，请参阅 Wonderware DAServer 文档。
主题名	指定给 “I/O 服务器设备组” 的标签。

如果将 Excel 电子表格用作 InTouch 数据源，则可以按如下方式定义 “访问名”：

访问名选项	描述
节点名	运行 Excel 程序的计算机的节点名。
应用程序名	Excel 是 “应用程序名”。
主题名	包含所请求的数据的 Excel 工作簿与电子表格的名称。例如，[Book1]Sheet1。

除节点、应用程序、主题以及项目之外，还需要指定远程节点上的数据的类型。此信息确定在 “标记名字典” 中定义标记时给它指定的 I/O 类型。

支持的 InTouch 通讯协议

您可以配置 InTouch HMI 以使用 DDE 或 SuiteLink。

InTouch HMI 还支持 Wonderware ArchestrA Message Exchange（Wonderware ArchestrA 消息交换）通讯协议。

如需有关在 InTouch 应用程序中使用“消息交换”的详细信息，请参阅第 93 页的“从 InTouch 中访问 Application Server 数据”。

动态数据交换

“动态数据交换” (DDE) 通讯协议使 Windows 应用程序之间可以相互通讯。DDE 在两个当前正在运行的应用程序之间建立客户端 - 服务器关系。服务器应用程序提供数据，并接受由希望使用服务器数据的任何其它应用程序所发出的请求。发出请求的应用程序称为客户端。InTouch 应用程序可以既是客户端，又是服务器。

Wonderware SuiteLink

Wonderware SuiteLink 是专为工业应用设计且基于 TCP/IP 的协议。SuiteLink 提供数据完整性、高吞吐量以及一些简单的诊断过程。Microsoft Windows NT 4.0 及更高版本支持 SuiteLink 协议。

SuiteLink 并非 DDE 或 NetDDE 的替代品。客户端与服务器的每个连接都取决于网络要求。

SuiteLink 提供以下功能：

- “数值时间质量” (Value Time Quality, 简称 VTQ) 在传递给支持 VTQ 的客户端的所有数据值上都会插入一个时间标签与质量指示符。
- 通过 Microsoft Windows 操作系统的性能监视器，可以对数据吞吐量、服务器负载、计算机资源消耗以及网络传输等进行非常全面的诊断。
- 不管应用程序是在单个节点上，还是分布在很多节点上，都可以在应用程序之间始终维持很高的数据吞吐量。
- 网络传输协议是使用 Microsoft 标准 Winsock 接口的 TCP/IP。

SuiteLink 通讯问题疑难排解

如果遇到 SuiteLink 通讯问题，执行以下操作：

- 确认 Microsoft TCP/IP 可以在安装 InTouch HMI 的计算机上正常工作。

- 验证计算机节点名不超过 15 个字符。
- 确认 Wonderware SuiteLink 在安装 InTouch HMI 的计算机上是作为服务在运行。
- SuiteLink 在安装 InTouch 的过程中自动安装。SuiteLink 服务会自动启动。如果 SuiteLink 服务停止，必须重新启动它。

设置访问名

您必须将 InTouch I/O 标记或远程标记引用与“访问名”关联起来。“访问名”定义与另一个 I/O 数据源之间的通讯链接。每个“访问名”指定一个由节点名、应用程序名、主题组成的 I/O 地址。

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，或是指向本地“I/O 服务器”的局部地址。

InTouchView 显示专为在 ArchestrA Application Server 环境中使用的 HMI 应用程序的可视化界面。InTouchView 应用程序作为客户端运行，由 Application Server 充当提供大多数 HMI 功能的服务器。

InTouchView 应用程序仅提供全功能 InTouch 应用程序中的某些标准功能。InTouchView 应用程序无法连接到 ArchestrA Application Server Galaxy 之外的其它 I/O 数据源。使用 InTouchView 应用程序时，只能使用缺省的“访问名” Galaxy，且无法创建“访问名”。

要创建“访问名”

- 1 在特别菜单上，单击访问名。此时出现访问名对话框。
- 2 单击添加。此时出现添加访问名对话框。

该对话框用于配置新的访问名。其包含以下字段和选项：

- 访问名(M):** 文本输入框，右侧有“确定”按钮。
- 节点名:** 文本输入框，右侧有“取消”按钮。
- 应用程序名(A):** 文本输入框，右侧有“故障转移(F)”按钮。
- 主题名(T):** 文本输入框。
- 要使用的协议(P):** 包含三个单选按钮：DDE、SuiteLink (当前选中)、消息交换。
- 要对服务器提示时(W):** 包含两个单选按钮：提示所有项、只提示激活项 (当前选中)。
- 启用辅助数据源:** 位于底部，带有一个复选框。

- 3 设置添加访问名对话框中的各个属性。执行以下操作：
 - 在访问名框中，输入确定此“访问名”的名称。

- 如果数据在网络 “I/O 服务器” 上，请在**节点名**框中输入该远程服务器的节点名。
- 在**应用程序名**框中，输入将从中获取数据的 “I/O 服务器” 程序的**实际程序名**。

如果 I/O 数据源是 Wonderware DAServer，请输入 DAServer 程序的名称；不要包含该程序的 .exe 文件扩展名。

- 在**主题名**框中，输入希望访问的主题名。

主题名是特定于应用程序的数据元素子组。在数据来自 Wonderware DAServer 程序的情况下，主题名与在 DAServer 程序中给主题配置的名称相同。与 Microsoft Excel 通讯时，主题名必须是保存工作簿与电子表格时所指定的名称。例如， [Book1]Sheet1。

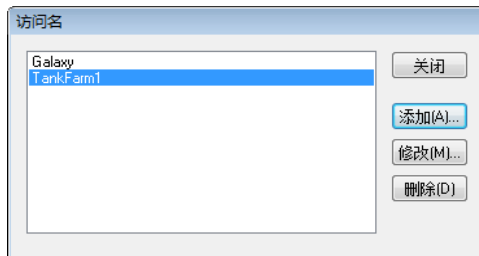
- 选择与 “I/O 服务器” 进行通讯的通讯协议。
- 选择轮询服务器上存储的信息的选项。

选项	定义
提示所有项	轮询所有的数据，不论这些数据是否在可见的窗口中，也不论是否已报警、已记录、已绘制到趋势图上或是用在脚本中。选择此选项会影响性能，因此建议不要使用。
只提示激活项	轮询部分数据，这些数据要么是显示在可见的窗口中，要么是已报警、已记录、已绘制到趋势图上或用在脚本中。
	备注 除非出现在可见的窗口中，否则不会轮询到按钮动作脚本。

- 4 如果希望选择一个辅助备份服务器，请选择**启用辅助数据源**。否则请转到步骤 5。

如果选择**启用辅助数据源**选项，则展开**添加访问名**对话框，以显示辅助数据源的配置字段。

- 完成各个选项。
- 给辅助数据源备份服务器指定所有的值之后，单击**确定**。
- 完成指定“访问名”的工作时，单击**确定**。此时再次出现**访问名**对话框，新的“访问名”已添加到列表中。



- 5 单击**关闭**。

如需有关设置辅助服务器以实现故障转移的详细信息，请参阅第 122 页的“使用访问名的故障转移功能”。

删除访问名

您可以删除不再需要的“访问名”。在删除“访问名”之前，确保满足以下条件：

- 没有标记与该“访问名”关联。
- WindowViewer 已停止。

要删除“访问名”

- 1 在**特别**菜单上，单击**访问名**。此时出现**访问名**对话框。当前的各个“访问名”会列出来。
- 2 要删除某个“访问名”，请从列表中选择它，然后单击**删除**。此时出现一条消息，要求确认是否应删除该“访问名”。
- 3 单击**是**。
- 4 单击**关闭**；如果要删除定义的其它“访问名”，请重复此操作程序。

使用 I/O 标记访问 I/O 数据

InTouch HMI 可以使用 I/O 标记来发送与接收本地或远程 Windows 应用程序的数据。每个 I/O 型标记引用都是“I/O 服务器”程序中的有效项目。您可以在“标记名词典”中定义不同类型的 I/O 标记。

配置 I/O 标记的属性

在“标记名词典”中可以定义不同类型的 I/O 标记。

指定离散 I/O 标记

I/O 离散标记指出可编程控制器、过程计算机的所有输入与输出以及网络节点数据的二值状态。

I/O 离散标记必须指定一个初始值，即“打开”或“关闭”。您也可以配置离散 I/O 标记，使之切换到二值数据源的相对值。您可以指定与标记关联的过程进入或脱离某种报警状态时，在报警事件窗口中出现的消息。

如需有关从“标记名字典”中创建标记的总体操作程序的详细信息，请参阅第 27 页的“创建新标记”。

要定义 I/O 离散标记

- 1 打开“标记名字典”并为新的标记指定一个名称。
- 2 从标记类型对话框中，将标记指定为 I/O 离散型。此时出现标记名字典对话框的详细资料部分。

The screenshot shows a dialog box for configuring a mark. It has several sections: 'Initial Value (I)' with radio buttons for 'Open' and 'Close (F)'; 'Input Conversion' with radio buttons for 'Direct (D)' and 'Inverted (I)'; 'Open Message' and 'Close Message (C)' text boxes; 'Access Name (M)' with a button to browse; 'Project (P)' with a dropdown menu showing 'Unassigned'; and a checkbox for 'Use Mark Name as Project Name (U)'.

3 执行以下操作：

- 选择**打开或关闭**作为与标记关联的初始值。

InTouch HMI 在应用程序启动时将此值指定给标记，但并不会将这个初始值写入 I/O 设备。

- 选择**直接**或**取反**作为要应用于从远程 I/O 标记接收到的值的输入转换。

输入转换	描述
直接	输入值从 “I/O 服务器” 程序中直接读取而不加改变。
取反	I/O 离散值在从服务器程序中读取时进行切换。例如，如果 I/O 输入是 0，则值自动设置为 1。

4 在打开消息**与**关闭消息**框中输入标记进入或脱离某种报警状态时出现的消息。**

这些消息可用在任何动画链接或脚本中，而不论该标记是否配置了报警。

- 如果定义在标记值等于 1 (On, True) 时处于活动状态的离散报警，则在**打开消息**框中输入的消息会出现在 ActiveX 报警显示的**值**与**报警限**列中。

标记的报警状态恢复正常时，在**关闭消息**框中输入的消息会出现在**值**列中，“打开消息”保持在**报警限**列中。

- 如果定义在标记值等于 0 (Off, False) 时处于活动状态的离散报警，则在**关闭消息**框中输入的消息会出现在 ActiveX 报警显示的**值**与**报警限**列中。

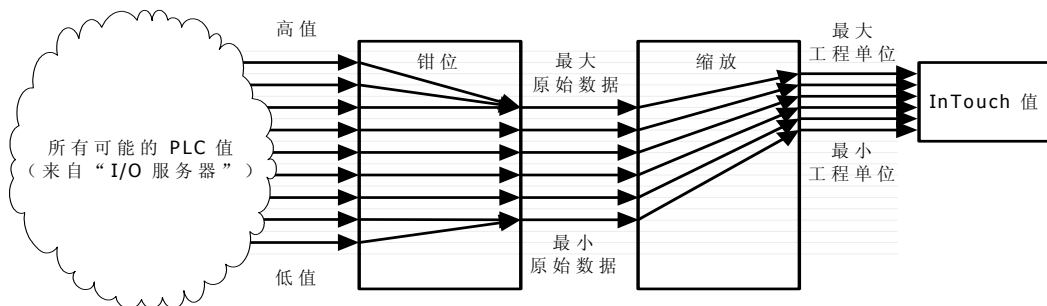
标记的报警状态恢复正常时，在**打开消息**框中输入的消息会出现在**值**列中，“关闭消息”保持在**报警限**列中。

5 将更改保存到标记中。

指定整型与实型 I/O 标记

您必须为 I/O 整型与实型标记指定一组属性，这些属性描述在 InTouch 应用程序与外部进程之间发送的数值数据。

InTouch HMI 对来自 PLC 的原始输入数据进行规格化处理。下图显示以下过程：限定原始的 I/O 数据值，然后将它们缩放到可以从 InTouch 应用程序中显示的工程单位值。



I/O 整型与实型标记包含某些属性，这些属性设置 PLC 发送的原始输入数据的最小与最大限制。InTouch HMI 会对低于或高于原始值范围的 I/O 值进行钳位处理。钳位功能将超出范围的值重新指定为最小或最大原始值。

I/O 整型与实型标记包含某些属性，这些属性将钳位后的原始值缩放到一定的工程单位范围。最小与最大工程单位属性设置缩放后的值的上、下边界。

定义整型与实型 I/O 标记时，可以指定缩放原始值以便计算工程单位值时要采用的转换类型。您可以选择“线性”或“平方根”。

对于线性缩放，结果是在最小与最大端点之间使用线性插值进行计算的。输入值的线性缩放算法是：

$$EUValue = (RawValue - MinRaw) * ((MaxEU - MinEU) / (MaxRaw - MinRaw)) + MinEU$$

输出值的线性缩放算法是：

$$RawValue = (EUValue - MinEU) * ((MaxRaw - MinRaw) / (MaxEU - MinEU)) + MinRaw$$

对于平方根缩放，最小与最大原始值用作插值。这对于缩放来自非线性设备（如压力传感器）的输入非常有用。输入值的平方根缩放算法是：

$$EUValue = \sqrt{RawValue - MinRaw} * ((MaxEU - MinEU) / \sqrt{MaxRaw - MinRaw}) + MinEU$$

输出值的平方根缩放算法是：

$$RawValue = \text{square}((EUValue - MinEU) * (\sqrt{MaxRaw - MinRaw} / (\sqrt{MaxEU - MinEU}))) + MinRaw$$

要定义整型与实型 I/O 标记

- 1 打开“标记名字典”并为新的标记指定一个名称。
- 2 从**标记类型**对话框中，将“I/O 整型”或“I/O 实型”指定为标记的类型。此时出现**标记名字典**对话框的详细资料部分。

- 3 执行以下操作：
 - 在**初始值**框中，输入在应用程序启动时与该标记关联的整数或实数。
应用程序并不会将这个初始值写入外部进程。
 - 在**最小工程单位**框中，输入标记的最小工程单位值。
 - 在**最大工程单位**框中，输入标记的最大工程单位值。
 - 在**最小原始数据**框中，输入原始 I/O 整型或实型数值的下钳位（最小值）。
 - 在**最大原始数据**框中，输入原始 I/O 整型或实型数值的上钳位（最大值）。
 - 在**工程单位**框中，输入要给标记的工程单位使用的标签。
 - 选择**线性**或**平方根**作为计算工程单位值时缩放原始值所采用的转换类型。
- 4 将更改保存到标记中。

指定消息型 I/O 标记

您可以指定一个 I/O 消息标记选项，以指定远程进程的网络地址。它的消息属性与内存消息标记相同。

要定义内存与 I/O 消息标记值

- 1 打开“标记名字典”并为新的标记指定一个名称。
- 2 从**标记类型**对话框中，选择 **I/O 消息** 作为标记的类型。此时出现**标记名字典**对话框的详细资料部分。

- 3 在**最大长度**框中，输入标记的消息中允许的最大字符数。在消息中最多可以输入 131 个字符。
- 4 在**初始值**框中，输入希望在 WindowViewer 启动应用程序时显示的文本字符串。
- 5 将更改保存到标记中。

设置 I/O 访问参数

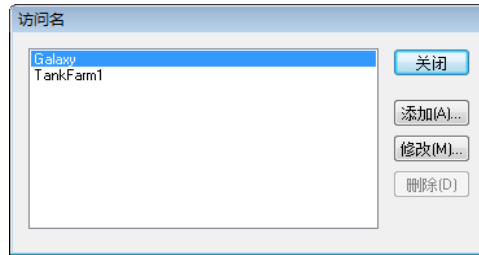
您可以在“标记名字典”中设置标记的 I/O 属性。这些属性确定与标记关联的外部数据源。

这些步骤仅说明如何从“标记名字典”中指定 I/O 属性。如需有关配置 Galaxy 与“访问名”的详细信息，请参阅第 65 页的“使用 I/O 进行数据访问”。

要设置标记的 I/O 属性

- 1 从**标记类型**对话框中，将标记指定为某种 I/O 标记类型。此时出现**标记名字典**对话框的详细资料部分。

- 2 单击**访问名**以定义或选择指定给标记的“访问名”。此时出现**访问名**对话框，显示 InTouch HMI 识别出的一列当前“访问名”。（Galaxy 是 ArchestrA 连接的缺省“访问名”）。



- 3 添加“访问名”或接受缺省值。
- 4 选择服务器程序中的数据点供 I/O 标记读取和写入数据。
 - 要在服务器程序中的某个过程数据点上读取和写入数据，请在**项目框**中输入“项目名”。例如，要从 PLC 的寄存器中读取某个值，请输入该寄存器的标识作为“项目名”。例如：

要将 Allen-Bradley PLC 的寄存器 1 用作“项目名”，请在**项目框**中输入 R1。

要将 Allen-Bradley PLC 的寄存器 1 的最低有效位用作“项目名”，请在**项目框**中输入 R1:0。
 - 要将标记用作项目，请选择**将标记名用作项目名**。

在运行时检索 I/O 标记的有关信息

您可以使用返回在“访问名”定义中指定的节点名、应用程序名以及主题名的函数编写脚本。

IOGetNode() 函数

IOGetNode() 函数将为特定“访问名”定义的节点地址返回给与脚本中的函数关联的标记。

类别

其它

语法

```
IOGetNode ("AccessName") ;
```

参数

AccessName

要返回其节点信息的现有“访问名”。

附注

您可以将“访问名”指定为字符串，或是使用其它 InTouch 标记或函数提供的字符串值。

示例

下例将 ModbusPLC1 “访问名” 的节点信息返回给 NodeName 标记。

```
NodeName = IOGetNode("ModbusPLC1");
```

IOGetApplication() 函数

IOGetApplication() 脚本函数将为特定 “访问名” 定义的应用程序名返回给某个标记，该标记指定为函数的参数。

类别

其它

语法

```
IOGetApplication("AccessName");
```

参数

AccessName

在其中定义应用程序的现有 “访问名”。

附注

您可以将 “访问名” 指定为字符串，或是使用其它 InTouch 标记或函数提供的字符串值。

示例

本例将为 ModbusPLC1 “访问名” 指定的应用程序名返回给 AppName 标记。

```
AppName = IOGetApplication("ModbusPLC1");
```

IOGetTopic() 函数

IOGetTopic() 脚本将为特定 “访问名” 定义的标题名返回给某个标记，该标记与脚本中的函数关联。

类别

其它

语法

```
IOGetTopic("AccessName");
```

参数

AccessName

要返回其主题名的 “访问名”。

附注

“访问名” 可以指定为字符串，或是使用其它 InTouch 消息标记或函数提供的字符串值。

示例

本例将访问名 ModbusPLC1 的主题信息返回给 TopicName 标记。

```
TopicName = IOGetTopic("ModbusPLC1");
```

在运行时动态更改 I/O 标记引用

InTouch HMI 使用动态引用查看只是临时需要它们的值的数据点，如在诊断应用程序中。通过“动态引用寻址”，可以使用单个标记名对多个数据源进行寻址。

您可以通过多种方法使用单个标记来动态引用多个数据源：

- 使用 I/O 标记的 .Reference 点域指定不同的“访问名”特征
- 使用 IOSetItem() 脚本函数设置 I/O 标记的 .Reference 点域
- 使用 IOSetAccessName() 脚本函数在运行时更改“访问名”的特征

.Reference 点域

通过将有效的引用指定给 I/O 标记的 .Reference 点域，可以实现“动态引用寻址”。通过使用 .Reference 点域来修改指定给 I/O 标记的“访问名”的特征，可以动态更改数据源。

.Reference 点域的语法是：

<code>tag.Reference=</code>	更改指定给标记的“访问名”与项目。
<code>“accessname.item”</code>	
<code>tag.Reference= “[.]item”</code>	更改指定给 I/O 标记的项目。
<code>tag.Reference= “accessname.”</code>	更改 I/O 标记的“访问名”。
<code>tag.Reference= “ ”</code>	使 I/O 标记不再活动。

每个 I/O 类型标记都有一个 .ReferenceComplete 点域。此离散点域的值指出 .Reference 点域中请求的项目是否会反映在 .Value 点域中。

应用程序在 WindowViewer 中启动时，.ReferenceComplete 域设置为假 (0)。.Value 点域经确认使用 .Reference 点域中指定的数据源进行更新时，.ReferenceComplete 值设置为真 (1)。如果 .Reference 点域发生更改，则 .ReferenceComplete 点域会自动设置为假 (0)，然后在更新新的值时更新为真 (1)。

IOSetItem() 函数

通过在脚本中使用 IOSetItem() 函数，可以实现“动态引用寻址”。IOSetItem() 包含一些参数，可以在运行时更改指定给 I/O 标记的 .Reference 点域的值。

类别

其它

语法

```
IoSetItem ("Tag", "AccessName", "Item");
```

参数*Tag*

使用英文双引号括起来的任何 InTouch I/O 标记。

AccessName

指定给 I/O 标记的“访问名”。

Item

指定给 I/O 标记的“项目”。

Tag、AccessName 以及 Item 参数可以指定为字符串，它们也可以是其它 InTouch 标记或函数提供的字符串值。

示例

在下例中，PumpInP1 标记的 .Reference 点域更改为指向 Excel “访问名”与 R1C1 项目。

```
IOSetItem("PumpInP1", "excel", "R1C1");
```

或者

```
Number = 1;
```

```
TagNameString = "PumpInP" + Text(Number, "#");
```

```
IOSetItem(TagNameString, "excel", "R1C1");
```

如果将“访问名”与“项目”的值都指定为空字符串(“”), 则可以使标记不再活动。例如, 通过以下方法可以使 PumpInP2 不再活动:

```
IOSetItem("PumpInP2", "", "");
```

如果只是给“项目”指定空值, 则标记的当前项目值保持不变, 它的“访问名”值会进行更新。例如, 以下代码将 PumpInP3 标记的“访问名”更改为 excel2, 而不影响它当前的“项目”值:

```
IOSetItem("PumpInP3", "excel2", "");
```

类似地, 如果只是给“访问名”指定空字符串, 则标记的当前“访问名”值保持不变, 它的“项目”值会进行更新。下例将 PumpInP4 标记的“项目”更改为 R1C2, 而不影响它当前的“访问名”值:

```
IOSetItem("PumpInP4", "", "R1C2");
```


IOSetAccessName() 函数

通过在脚本中使用 IOSetAccessName() 函数，可以实现“动态引用寻址”。IOSetAccessName() 在运行时修改 I/O 标记的“访问名”的应用程序名或主题名的特征。

备注 处理 IOSetAccessName() 函数时，在结束当前的对话并启动新的对话期间，会有一个时间上的延迟。在此期间，任何“插入”或写入新主题的尝试都将丢失。

类别

其它

语法

```
IOSetAccessName("AccessName", "NodeName", "AppName",
  "TopicName");
```

参数

AccessName

要将新的“应用程序名”与“主题名”值指定给它的现有“访问名”。实际的字符串或消息标记。

NodeName

要指定的新“节点名”。实际的字符串或消息标记。

AppName

要指定的新“应用程序名”。实际的字符串或消息标记。

TopicName

要指定的新“主题名”。实际的字符串或消息标记。

附注

指定给 AccessName、AppName 以及 TopicName 参数的值可以指定为字符串，或是使用其它 InTouch 标记或函数提供的字符串值。

如果使用添加访问名对话框为访问名故障转移配置了辅助数据源，则无法在运行时使用 IOSetAccessName() 函数来更改辅助数据源配置。

备注 在 WindowMaker 中创建“访问名”时，如果“访问名”是 SuiteLink 类型，则 InTouch HMI 会阻止“访问名”去访问相同的节点、应用程序以及主题。请勿允许 IOSetAccessName() 函数在运行时将“访问名”重定向到重复项。通过在运行时使用 IOSetAccessName() 函数，SuiteLink 类型的“访问名”可以重定向到重复的主题。重定向后的“访问名”将不起作用。

示例

通过使用以下脚本函数，可以将 MyAccess1 “访问名”更改为指向 Excel 应用程序与 [Book1]Sheet1 主题，而不影响当前的 NodeName:

```
IOSetAccessName("MyAccess1", "", "EXCEL", "[Book1]Sheet1");
```

如果给“主题”指定空字符串，则“访问名”的当前“应用程序”值会进行更新，它的“主题”值会保持不变。

例如，以下脚本将 MyAccess2 “访问名”的“应用程序名”更改为 EXCEL，而不影响当前的“主题”值：

```
IOSetAccessName("MyAccess2", "", "EXCEL", "");
```

如果仅仅给“应用程序名”指定空字符串，则标记的当前“主题”值会进行更新，但它的“应用程序”值会保持不变。例如，以下脚本将 MyAccess3 标记的“主题”更改为

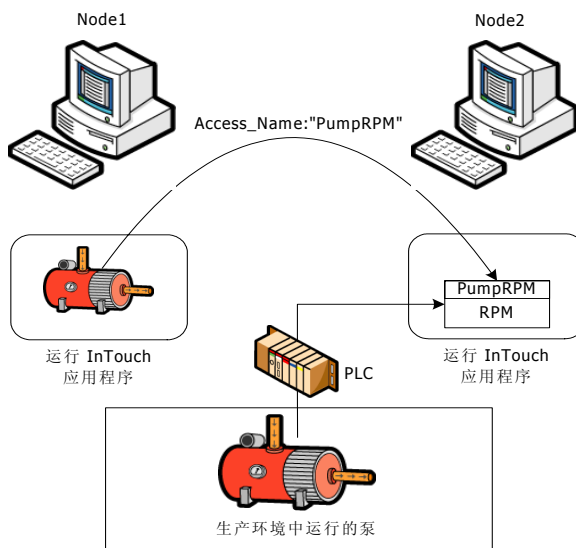
[Book2]Sheet1，而不影响它的当前“应用程序名”值：

```
IOSetAccessName("MyAccess3", "", "", "[Book2]Sheet1");
```

本例可用于必须采用 PLC 冗余配置的情况。

将标记转换为远程引用

您可以根据客户端 - 服务器架构创建分布式 InTouch 应用程序。客户端应用程序可以运行在使用其它远程节点上定义的标记的一个网络节点上。下图显示在 Node1 上运行的 InTouch 应用程序，它远程引用 Node2 上的 PumpRPM 标记。



在本例中，可以使用两种方法来检索 Node2 上 PumpRPM 标记的值：

- 在 Node1 的“标记字典”中创建一个 I/O 型标记，在同这个标记关联的“访问名”中 Node2 被用作节点名。
- 使用指向 PumpRPM 标记的直接远程引用。例如，PLC1:”PumpRPM”。

在窗口或 QuickScript 中，通过将“访问名”作为前缀附加到远程标记名之前，可以引用远程标记，具体形式如下：

```
access_name:"tag_name"
```

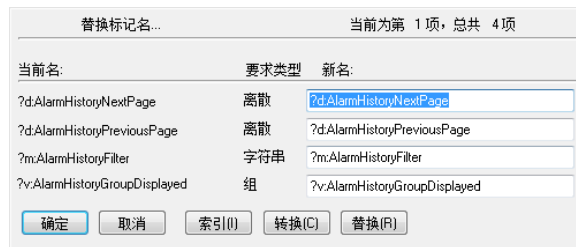
导入窗口或 QuickScript 时，可以将占位符标记转换为远程标记引用。例如，您可以将占位符标记转换为指向从中导入窗口的应用程序。这些标记不需要在本地“标记名字典”中定义。

您可以使用多种方法将本地标记转换为远程标记引用：

- 附加远程标记引用
- 转换与导入的窗口关联的占位符标记
- 启动“标记浏览器”，然后打开标记源的“标记名字典”以选择远程标记引用。

要手工将标记转换为远程标记引用

- 1 在 WindowMaker 中打开一个应用程序窗口。
- 2 选择同希望更改为远程标记引用的本地标记关联的对象。
- 3 在特别菜单上，单击**替换标记**。此时出现**替换标记名**对话框，显示与该对象关联的一系列标记。



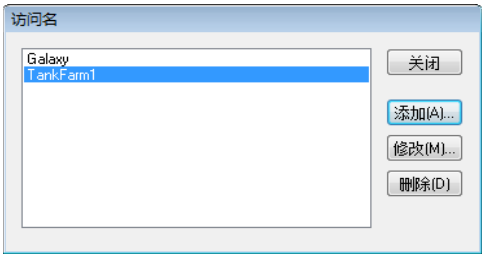
- 4 单击**索引**将索引添加到每个标记名。
- 5 单击**转换**。此时出现**转换**对话框，显示将标记转换为本地或远程引用标记的选项。
- 6 单击**远程**。此时出现**访问名**对话框。对话框中显示在本地 InTouch 应用程序中定义的所有“访问名”。
- 7 从列表选择一个“访问名”。
- 8 单击**关闭**。此时**替代标记**对话框中列出的所有标记都自动转换为远程标记引用，“访问名”会附加到标记名上。
- 9 单击**确定**。

要将导入的窗口的标记转换为远程引用

- 1 打开导入的窗口并选择所有的对象。
- 2 在特别菜单上，单击**替换标记**。此时出现**替换标记名**对话框。
- 3 单击**转换**。此时出现**转换**对话框。



- 4 单击**远程**。此时出现**访问名**对话框。对话框中显示在本地 InTouch 应用程序中定义的所有“访问名”。



- 5 从列表选择一个“访问名”。
要验证“访问名”是否正确配置，请单击**修改**。
如果当前尚未定义指向该标记源的“访问名”，请单击**添加**并定义它。“访问名”必须包含应用程序所在远程节点的名称。
- 6 单击**关闭**。此时**替代标记**对话框中列出的所有标记都自动转换为远程标记引用，“访问名”会附加到标记名上。



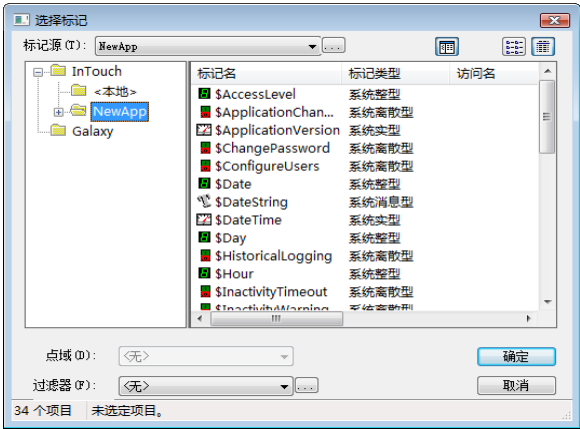
- 7 单击**确定**。

要在“标记浏览器”中选择远程标记引用

- 1 选择同要转换为远程标记引用的本地标记关联的对象。
- 2 在特别菜单上，单击**替换标记**。此时出现显示所选标记的**替换标记名**对话框。
- 3 删除**新名**框中要替换为远程标记引用的标记名。
- 4 双击**新名**框。此时出现**选择标记**对话框，显示与该应用程序关联的一系列标记。
- 5 使用“目录树”视图选择一个远程标记。



a 单击**目录树**图标，以便在左侧窗格中显示所有本地与远程“访问名”的层次结构化列表。



- b 选择一个远程“访问名”文件夹，以便在右侧窗格中显示已指定给它的标记。
- c 选择要用作远程引用的远程标记。
- d 单击**确定**。此时出现**替换标记名**对话框，**新名**框中的内容是所选的远程标记名。

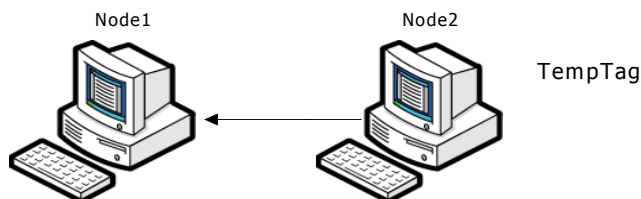


- 6 单击**确定**以关闭对话框并将远程标记关联到所选的对象。
- 7 针对要与远程引用关联的每个标记重复这些步骤。

通过远程引用访问 I/O 数据

InTouch HMI 使工厂自动化应用程序实现了真正的客户端 - 服务器架构。您可以设计这样的客户端应用程序，它不使用与运行 InTouch 应用程序位于相同节点的本地“标记名字典”中的任何标记。您可以在一个节点上运行应用程序，而此应用程序通过使用远程标记引用来使用远程节点的标记。

下图显示一个简单的示例，其中 TempTag 在 Node2 的本地定义：



在本例中，Node1 上运行的 InTouch 应用程序可以通过两种方法来检索 Node2 上 TempTag 的值：

- 在 Node1 的“标记名字典”中创建一个 I/O 型标记，在与该 I/O 标记关联的“访问名”中 Node2 被用作“节点名”。
- 使用一个直接指向 TempTag 的远程引用。例如，Node2:”TempTag”。

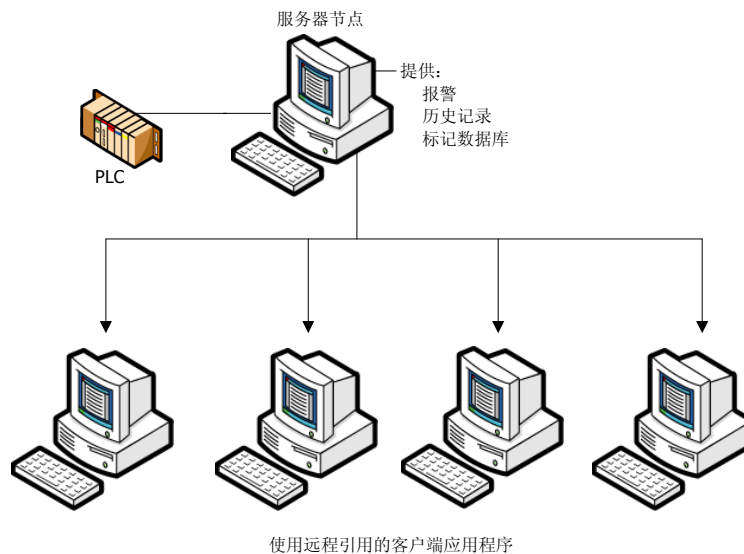
要在任何其它应用程序中直接引用远程标记，仅需要使用 *AccessName:item*。您不必在本地“标记名字典”中定义远程标记。远程引用还可以访问来自任何 I/O 数据源（如 Wonderware DAServer 或 Microsoft Excel）的数据。

您也可以远程引用 SuperTag。对 SuperTag 进行远程标记引用的有效语法如下：

Access_name:Parent_Instance\ChildMember\SubMember.

如需有关远程引用 SuperTag 的详细信息，请参阅第 150 页的“引用 SuperTag 成员”。

导入窗口或 QuickScript 时，可以将占位符标记转换为远程标记引用。您不必在本地“标记名字典”中定义标记。远程引用可以从网络上的任何 Wonderware 应用程序中进行访问。



在运行时重定向远程引用

通过使用脚本，可以在运行时重定向指向 InTouch 标记的 ArchestrA 对象引用或远程引用。根据满足的特定条件，或是直接通过操作员的动作，可以切换图形符号的对象实例。

IOSetRemoteReferences() 函数

在 InTouch 应用程序运行期间，可以使用 IOSetRemoteReferences() 脚本函数来重定向指向标记的 ArchestrA 对象引用或远程引用。IOSetRemoteReferences() 查找与指定的字符串匹配的所有远程引用，并根据指定的参数值来更改这些引用。您可以创建这样的脚本：它根据满足的条件或用户的动作来触发重定向引用的函数。

类别

其它

语法

IOSetRemoteReferences(*BaseAccess*, *NewAccess*, *MatchString*,
SubstituteString, *Mode*)

参数

BaseAccess

此字符串参数指定配置的原始“访问名”，此访问名要在引用中进行匹配。

NewAccess

新的“访问名”。新的“访问名”应用于满足以下条件的所有引用：原始“访问名”与 *BaseAccess* 提供的字符串匹配，并且原始“项目名”与 *MatchString* 值（如果有指定）匹配。

MatchString

引用中配置的原始“项目名”所要匹配的字符串。如果 *MatchString* 值是空字符串，则视为同任何“项目名”都匹配。

SubstituteString

要替换原始“项目名”的字符串。此字符串会替换 **MatchString** 值，以便为引用创建新的活动“项目名”。如果 **SubstituteString** 是空字符串，则不进行替换。

Mode

确定对配置的原始“项目名”与 **MatchString** 值进行比较时要使用的方法。匹配操作总是从“项目名”的开头进行。**Mode** 值为 0 时，指定必须匹配整个“项目名”，或是匹配到名称中的英文句号 (.)。**Mode** 值为 1 时，指定允许部分匹配，即便下一个字符不是英文句号。

附注

IOSetRemoteReferences() 不会在更改对象引用之前检查新的标记或“访问名”是否有效。

- **IOSetRemoteReferences()** 只更改远程引用。函数重定向满足以下条件的那些引用：配置的原始“访问名”与 **BaseAccess** 参数指定的值匹配，并且原始“项目名”与 **MatchString** 值匹配。
- 只需调用一次 **IOSetRemoteReferences()**，便会影响内存中的活动窗口中满足以下条件的所有远程引用：配置的原始名称字符串与指定给 **BaseAccess** 和 **MatchString** 参数的值匹配。
- 如果没有给 **BaseAccess** 参数指定任何值，则 **IOSetRemoteReferences()** 不会重定向任何远程引用。
- 如果 **MatchString** 参数为空，则 **IOSetRemoteReferences()** 重定向满足以下条件的所有远程引用：原始访问名与指定给 **BaseAccess** 参数的值匹配。

- **Mode** 参数设置为 0 时，仅替换完整对象名（或标记）或完整属性名（或点域）中的“项目名”。**MatchString** 参数的值必须与整个原始“项目名”匹配，或是匹配到英文句号前的一个字符。
- **Mode** 参数设置为 1 时，如果项目字符串以匹配项目字符串开头，则允许对项目字符串进行部分替换。也就是说，**MatchString** 必须与原始项目字符串的某个部分匹配，但这个部分必须从项目字符串的开头部分开始。匹配字符串中的最后一个字符后不必是英文句号。
- 给远程引用配置的原始名称保持不变。对 **IOSetRemoteReferences()** 的后续调用不需要识别当前的活动名称。对 **IOSetRemoteReferences()** 的调用可以按任何顺序进行。
- 如果让两个或更多的窗口指向同一个远程引用，则该远程引用的作用类似于 I/O 标记。对它进行重定向时，所有的窗口都会看到相同的内容。请勿使用单个名称来同时指向两个单独的目标。

备注 同时更改许多个引用（例如在“显示时”窗口脚本中）时，可能需要一些时间才能完成对所有引用的解析。

示例

下例将满足以下条件的所有远程引用重定向到 Galaxy “访问名”的 pump001 项目名：原始项目名与 pumpX 完全匹配。

```
IOSetRemoteReferences("Galaxy","", "pumpX",
    "pump001", 0);
```

下例将项目名与 pumpX 完全匹配的 Galaxy “访问名”更改为 TagServer1。此外，项目名也更改为 p2。

```
IOSetRemoteReferences("Galaxy", "TagServer1",
    "pumpX", "p2", 0);
```

下例在项目名是 pumpX 时将 TagServer1 “访问名”更改为 TagServer2。此外，项目名也更改为 backpump3。

```
IOSetRemoteReferences("TagServer1","TagServer2",  
    "pumpX","backpump3",0)
```

下例将 TagServer1 “访问名”的 Tank 项目名更改为 Plant。

```
IOSetRemoteReferences("TagServer1","", "Tank",  
    "Plant",1)
```

因为没有给 *BaseAccess* 参数指定任何值，下例不会重定向任何远程引用。

```
IOSetRemoteReferences("", "Galaxy", "pumpX",  
    "pump001", 0);
```

恢复引用

如果 *NewAccess* 参数为空（即没有指定的值），则 *IOSetRemoteReferences()* 会将活动的“访问名”恢复为原始的基本“访问名”。

如果 *MatchString* 参数为空（即没有指定的值），则 *IOSetRemoteReferences()* 会将活动的“项目名”恢复为原始“项目名”。

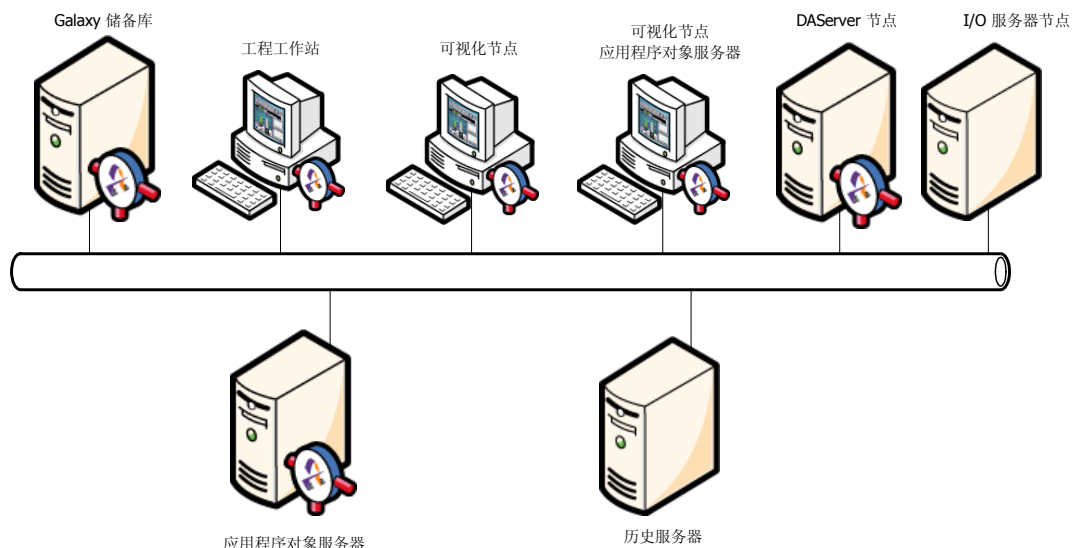
备注 即便 *SubstituteString* 不为空，但如果 *MatchString* 为空，则“项目名”也会恢复为原始“项目名”。在名称的开头插入文本是不允许的。例如，运行脚本 *IOSetRemoteReferences*(“Access1”，“”，“”，“Valve”，0); 时，并不会在任何原始“项目名”的开头或结尾附加字符串 *Valve*。

如果 *SubstituteString* 为空（即没有指定任何值），*IOSetRemoteReferences()* 会将活动的“项目名”恢复为原始的基本“项目名”。通过使用非空的 *MatchString* 与空的 *SubstituteString*，可以根据指定的原始基本访问名选择一部分远程引用，并恢复它们的原始“项目名”。

从 InTouch 中访问 Application Server 数据

ArchestrA 为整套 Wonderware 产品提供一组公共服务和底层架构。通过选择这组产品中的某些产品并使用模块化的 ArchestrA 组件，可以构建工厂自动化与信息系统。

Application Server 提供一组用于构建工厂自动化应用的服务。Application Server 服务分布在系统中的一组节点上。



通常，InTouch HMI 与 Application Server 相互配合，共同为操作员与之交互的应用程序提供可视化界面，从而对工厂的生产过程进行管理。

将 Application Server 对象属性用于 InTouch 标记

您可以将 InTouch 标记用于同 Application Server 对象属性进行交互，以便在 InTouch 应用程序与 Application Server 数据储备库之间传输数据。

InTouch HMI 支持的通讯协议包括“消息交换”。WindowViewer 运行 InTouch 应用程序时，“消息交换”将 WindowViewer 视为匿名引擎。

这种匿名性质表示 InTouch 应用程序没有任何属性可供其它“消息交换”客户端访问。在 Application Server Galaxy 中，WindowViewer 未配置成或当作 AutomationObject 进行管理，也不视为 AutomationObject。InTouch HMI 仅使用“消息交换”来预订 Application Server 中的那些活动项目。

您可以使用 InTouch 的“标记浏览器”将 Galaxy 选作远程标记的标记源，以及浏览 Galaxy 的域名空间。Application Server 对象属性或是属性的特性可以用在远程引用中，或是用作 InTouch I/O 标记的项目。

如需有关将 Application Server 对象用作远程标记源的详细信息，请参阅第 101 页的“配置 InTouch HMI 以便将 Galaxy 用作远程标记源”。

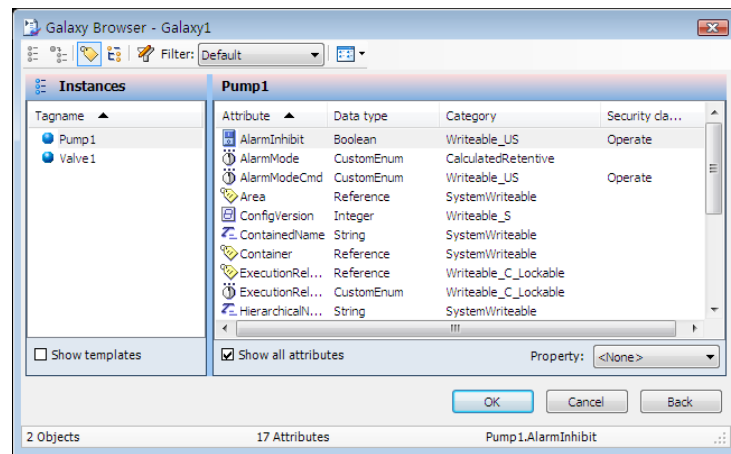
WindowMaker 中提供一个预先配置好的“访问名” Galaxy 供“消息交换”来访问。Galaxy “访问名”仅与 ArchestrA 环境中的 InTouch HMI 相关。Galaxy “访问名”没有用户可以配置的属性。

从 InTouch 浏览应用程序服务器对象属性

要从 InTouch HMI 浏览并选择“应用程序服务器”属性，必须首先在 InTouch HMI 中设置 Galaxy 的标记源。如需有关详细信息，请参阅第 101 页的“配置 InTouch HMI 以便将 Galaxy 用作远程标记源”。

从 InTouch 选择标记对话框中选择标记源。

InTouch 的 **Galaxy Browser**（Galaxy 浏览器）对话框列出目标 Galaxy 中的所有对象。您可以展开一个对象来查看它包含的对象或运行时可以访问属性。**Galaxy Browser**（Galaxy 浏览器）对话框不显示那些以“_”（标记为隐藏）开头的属性，也不显示任何 **QualifiedStruct** 类型的属性。



要从 **Galaxy Browser**（Galaxy 浏览器）对话框返回到普通的 InTouch 选择标记对话框，请单击 **Back**（返回）。

Application Server 浏览器的限制

使用 InTouch 的 **Attribute Browser**（属性浏览器）对话框查看 Application Server 对象时，存在以下限制：

- 只能查看到单个 Galaxy 的域名空间中的运行时可见属性。这包括在对象的 TagName 及其 HierarchicalName 之间切换的功能。

如果对象属性满足以下要求，则可以从 **Attribute Browser**（属性浏览器）对话框中选择它：

- 在运行时可见
- 是 AutomationObject 中当前选中的属性
- 属性名的 “.” 后面没有跟着 “_” 的情况
- **Attribute Browser**（属性浏览器）对话框仅显示 InTouch HMI 所支持的 Application Server 对象属性数据类型。如需有关支持的数据类型的详细信息，请参阅第 97 页的“将 Application Server 数据类型映射到 InTouch 数据类型”。
- InTouch 的 **Attribute Browser**（属性浏览器）对话框不显示会导致 InTouch “项目名” 超过 95 个字符这一最大限制的任何属性。
- 通过将“标记名.属性名[下标]”用作引用，可以在 InTouch HMI 中显示或检索自动化对象数组元素。使用下标 -1 可以显示或检索所有数组元素的值。
- 您可以使用“标记浏览器”来选择对象属性的特性。缺省条件下，选择属性时会选择 Value 特性。

Application Server 对象中的特殊扩展

WindowMaker 的“标记浏览器”与 WindowViewer 中的“消息交换”客户端可以给每个 Application Server 对象属性添加特殊扩展，并能识别它们。通过这些扩展，InTouch HMI 可以访问无法以其它方式获取的信息。

这些特殊扩展是可选的，并且 InTouch 应用程序不需要使用它们。不过，经常处理状态与质量信息的应用程序需要用到这些扩展。

这些项目扩展了属性的域名空间，使之包含一些

WindowViewer 可以提供给应用程序脚本与窗口的附加属性。例如，通过引用“TIC101.PV.#ReadSts”，可以访问有关 TIC101.PV 预订情况的 MxStatus 信息。此信息对于显示由“消息交换”提供的扩展信息非常有用。

在 Application Server 中，这些属性不像具名元素那样作为对象属性而存在。这些属性是客户端抽象层中提供的客户端扩展，可使得对象属性对于 InTouch HMI 而言是可见的。下表介绍 InTouch HMI 的这些属性扩展：

属性扩展	数据类型	用途
无	强制转换结果	缺省的扩展。表示未提供任何扩展。项目根据 InTouch 的要求使用强制转换的值数据类型进行读取 / 写入。如果客户端预订下文介绍的 #ReadSts 或 #WriteSts 项目，则可以获取失败的读取 / 写入信息。示例：“Pump1.PV”。
.#VString 仅限浮点 / 双精度属性： .#VString1 .#VString2 .#VString3 .#VString4	字符串（读取 / 写入）	设置对带 “.#VString” 后缀的引用的预订。这是底层引用。读取和写入都可以正常工作时，将底层引用的当前值作为字符串返回。如果 UserGetAttribute 返回错误的状态，此项目根据 MxStatus 返回简短的状态描述字符串，而不返回值。这些简短的状态描述字符串是： “?Pending” - 待处理 “?Warning” - 警告 “?Comms” - 通讯错误 “?Config” - 配置错误 “?Oper” - 操作错误 “?Security” - 安全性错误 “?Software” - 软件错误 “?Other” - 其它错误

对于 .#VString，如果状态正常但质量不佳，则此项目返回 “消息交换” 所提供的质量描述字符串，而不返回值。

仅当 UserGetAttribute 的质量和状态都正常，或是质量正常而状态不定时，才会将值作为字符串返回。这可能需要强制转换，“消息交换” 返回的数据类型不是字符串时便是如此。质量或状态不定时，值会带一个后缀 “?”。例如，“3.27?” 或 “True?”。

将 Application Server 数据类型映射到 InTouch 数据类型

Application Server 包括某些属性与数据类型，它们并未直接映射到 InTouch 标记所支持的四种基本数据类型。

下表显示客户端抽象层如何映射数据类型，以便进行读取和写入操作。它还显示 Galaxy 字典展现给 InTouch 的数据类型。

属性 特性数据类型	InTouch 数据类型	备注
Float	实型 - 32 位	直接传递。
Double	实型 - 32 位	如果双精度值是 IEEE NAN，则转换成浮点型 IEEE NAN。如果这会发生上溢，则将 Quality 设置为“不佳”，并传递浮点型 IEEE NAN。如果双精度小数值比最小的浮点小数 1.17549E-38 还小，则将它视为浮点数 0.0，并将 Quality 设置为“良好”。
Boolean	离散	假 = 0、真 = 1。
Integer	实型 - 32 位	直接传递。
String（总是 Unicode）	消息 - MBCS （多字节字符集编码）	如果字符串对于 InTouch 而言太长，则截断它并将 Quality 设置为“不定”。每个 Unicode 字符的两个字节都保持不变。
Time	消息 - MBCS	按照语言环境设置成适当格式的字符串。使用 MxValue 来转换字符串。
ElapsedTime	实型	作为浮点数传递，以秒计。 MxValue 支持强制转换为这种类型。
MxDataType	消息 - MBCS	传递字符串。
MxSecurityClassification	消息 - MBCS	传递字符串。
MxQuality	消息 - MBCS	传递字符串。
MxReference	消息 - MBCS	将引用字符串仅当作 Unicode 传递。
MxCategorizedStatus	消息 - MBCS	传递字符串。
MxQualifiedStruct	不支持。	不支持。

属性 特性数据类型	InTouch 数据类型	备注
MxQualifiedEnum	消息 - MBCS	传递“枚举”型字符串。应用程序可以通过引用 #EnumOrdinal 来访问整型序数。例如， “Pump1.PV.#EnumOrdinal”。
字符串数组	消息 - MBCS (只读)	将数组的每个元素放入一个由逗号分隔的字符串，如： “String1,String2,String3” 最长为 InTouch 字符串值的最大限制。如果这会发生截断情况，则发送给 InTouch HMI 的关联质量是“不定”。您无法写入整个字符串数组，但可以写入数组的单独元素。
所有数组	整型、实型、消息、离散	仅支持预订数组的单个元素。在这种情况下，将采用上文所述的转换。否则返回空字符串，并且质量为“不佳”。
MxInternationalizedText	消息	这在运行时作为字符串类型来访问。

Application Server 属性的读取 / 写入行为

系统写入“自动化对象”属性时，属性的写入状态最初设置为“?Pending”。

完成写入时，#WriteSts 字符串更新为写入操作的结果。如果成功完成写入，则 #WriteSts 值设置为空字符串。如果写入操作返回错误并处于待处理状态，即便读取操作上仍然在继续更新预订，#WriteSts 项目也会继续显示最新的写入状态。

您也可以使用 #VString1 到 #VString4 这些项目将浮点值或双精度值转换为字符串格式，数字 N 表示要返回的小数位数。例如，“3.1234”是 #VString4 的字符串。通过使用不带数字的 #VString 项目，可以将浮点值或双精度值舍入成整数值，然后将它作为字符串值返回。

属性扩展	数据类型	用途
.#EnumOrdinal	整型（读取 / 写入）	包含“限定枚举”类型属性的当前读取序数值。这是一种为枚举表达式返回整数（而不是返回字符串）的方法。
.#ReadSts	字符串（只读）	<p>包含项目预订情况的当前读取状态，字符串将与之串联起来。它显示为“TIC101.PV.#ReadSts”。它由“消息交换”作为字符串提供。它的值可以是如下内容之一：</p> <p>“?Config” - 配置错误</p> <p>“?Comms” - 通讯错误</p> <p>“?Oper” - 操作错误</p> <p>“?Pending” - 待处理</p> <p>“?Warning” - 警告</p> <p>“?Security” - 安全性错误</p> <p>“?Software” - 软件错误</p> <p>“?Other” - 其它错误</p> <p>备注 如果未预订关联的项目（例如，TIC101.PV），则返回的字符串为空。</p>

属性扩展	数据类型	用途
.#WriteSts	字符串（只读）	<p>包含此字符串要串联的项目的最新写入状态，例如 Pump1.Cmd.#WriteSts。它由“消息交换”作为字符串提供。如果字符串为空，则说明上次对项目的写入操作成功。否则，#WriteSts 可以是以下各个值之一：</p> <p>“?Config” - 配置错误</p> <p>“?Comms” - 通讯错误</p> <p>“?Oper” - 操作错误</p> <p>“?Pending” - 待处理</p> <p>“?Warning” - 警告</p> <p>“?Security” - 安全性错误</p> <p>“?Software” - 软件错误</p> <p>“?Other” - 其它错误</p> <hr/> <p>备注 如果未预订关联的项目（例如，TIC101.PV），则返回的字符串为空。</p>

配置 InTouch HMI 以便将 Galaxy 用作远程标记源

您可以使用 InTouch 的“标记浏览器”选择 Application Server 对象作为标记源，也可以使用它来浏览 Galaxy 数据库。Application Server 对象的属性或属性的特性可以用在远程引用中，或是用作 InTouch I/O 标记的项目。

InTouch 应用程序作为客户端运行，为 Application Server 应用程序提供可视化界面时，必须在 InTouch 应用程序所在的相同节点上安装 Application Server Bootstrap 与 WinPlatform 对象。

要从 InTouch HMI 中浏览 Galaxy 域名空间，还需要安装“ArchestrA 集成开发环境” (IDE)。

InTouch HMI 使用平台的“消息交换”功能来查看 Galaxy 域名空间，并提供更好的数据通讯。

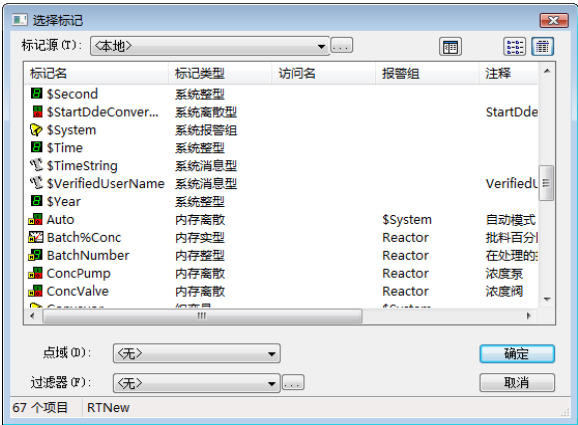
要配置 InTouch 将 Galaxy 用作远程标记源

- 1 在 WindowMaker 中打开一个应用程序窗口。
- 2 双击一个文本对象。此时出现动画链接对话框。

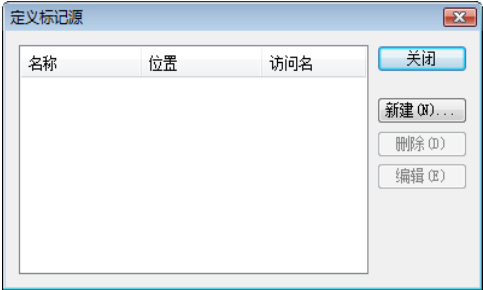


- 3 在值显示区域中，单击模拟。此时出现一个用于插入表达式的对话框。
- 4 删除表达式框中的任何表达式。

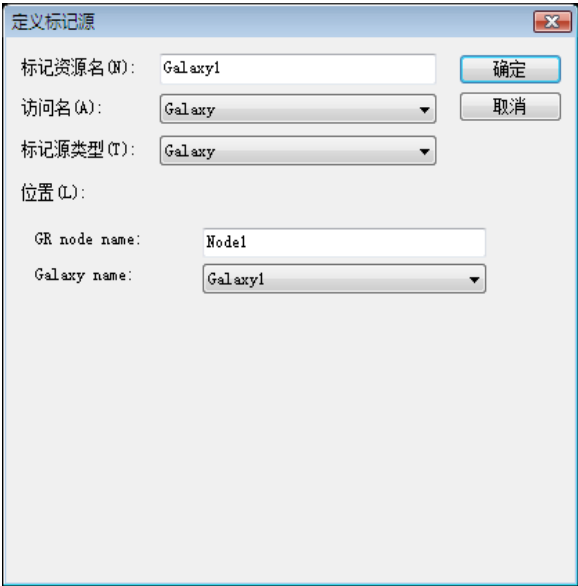
5 在表达式框内双击。此时出现**选择标记**对话框。



6 单击**标记源**框右侧的按钮。此时出现**定义标记源**对话框。



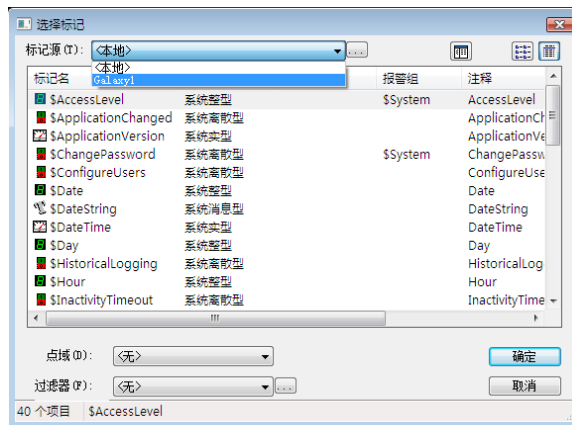
7 单击**新建**以显示**定义标记源**对话框。



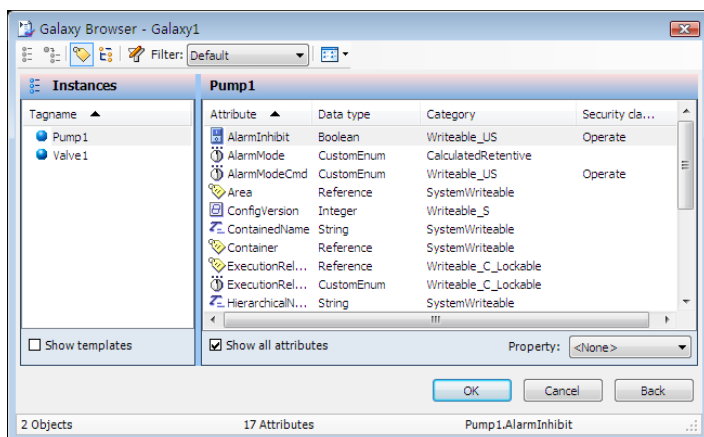
- 8 为定义标记源对话框中的各个框输入值。执行以下操作：
 - a 在**标记资源名**框中，输入远程 Galaxy 标记源的名称。
 - b 在**访问名**框中，从列表中选择 Galaxy。
 - c 在**标记源类型**框中，从列表中选择 Galaxy。
 - d 在**位置**区域中，输入 “Galaxy 储备库节点” 的名称，然后从列表中选择 Galaxy。
 - e 单击**确定**。定义标记源对话框显示您在它的列表中定义的远程标记源。



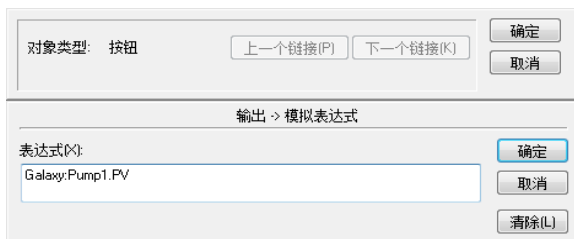
- 9 单击**关闭**以关闭定义标记源对话框。此时**选择标记**对话框在**标记源**框的列表中显示新的标记源。



- 10 从**标记源**框中选择所创建的新标记源。此时 **Galaxy 浏览器**对话框打开，并在其左侧窗格中显示标记列表。



- 11 从 **Galaxy 浏览器**对话框的左侧窗格中选择一个标记。此时 **Galaxy 浏览器**对话框的右侧窗格会进行刷新，以显示所选标记的属性。
- 12 单击要使用的属性，然后单击**确定**。此时出现**输出 -> 模拟表达式**对话框，并且其**表达式**框中会出现一个表达式。



- 13 确认该字符串表达式是否正确。
表达式采用以下形式：
`Galaxy:tag_name.attribute_name`
示例：
`Galaxy:Pump1.PV`

- 14 单击**确定**以关闭**输出 -> 模拟表达式**对话框。
- 15 根据需要配置其余的对象链接。
- 16 单击动画链接对话框中的**确定**。
- 17 单击**运行时**。此时文本对象显示所配置的标记属性的值。

查看 I/O 标记的时间标签与质量信息

InTouch HMI 在传递给支持 VTQ 的客户端的所有数据值上都会插入数值、时间及质量 (VTQ) 指示符。InTouch 将一组点域用作数据质量的指示符，它们对于疑难排解非常有用。

- .Value 点域包含指定的标记的值。这也是每个 InTouch 标记的缺省点域。如果没有指定其它点域，则假定使用 .Value 点域。
- Time 点域是时间标签，表示标记上次更新的时间。
- Quality 点域显示指定给 I/O 标记的数据值的可靠性。

查看 I/O 标记的时间标签信息

Time 点域按以下格式指定给标记：

Tag_name.Time_Dotfield

.TimeDate 点域

.TimeDate 点域显示自 1970 年 1 月 1 日到上次从 “I/O 服务器” 更新标记值之间已经过的天数的整数部分。

类别

标记

用法

Tag_name.TimeDate

参数

Tag_name

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

整型（只读）。

另请参阅

.TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear

.TimeDateString 点域

.TimeDateString 点域显示从 “I/O 服务器” 中更新标记值时的日期与时间。

类别

标记

用法

Tag.TimeDateString

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

消息（只读）。

另请参阅

.TimeDate, .TimeDay, .TimeDateTime, .TimeHour,
.TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond,
.TimeTime, .TimeTimeString, .TimeYear

.TimeDateTime 点域

.TimeDateTime 点域显示自 1970 年 1 月 1 日到上次从 “I/O 服务器” 更新标记值之间已经过的天数的小数部分。

类别

标记

用法

Tag.TimeDateTime

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

实型（只读）。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeHour,
.TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond,
.TimeTime, .TimeTimeString, .TimeYear

.TimeDay 点域

.TimeDay 点域显示当月自上次从 “I/O 服务器” 更新标记值之后已经过去的天数。

类别

标记

用法

Tag.TimeDay

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 1 到 31。

另请参阅

.TimeDate, .TimeDateString, .TimeDateTime, .TimeHour,
.TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond,
.TimeTime, .TimeTimeString, .TimeYear

.TimeHour 点域

.TimeHour 点域显示当天自上次从 “I/O 服务器” 更新标记值之后已经过去的小时数。

类别

标记

用法

Tag.TimeHour

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 0 到 23。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime,
.TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond,
.TimeTime, .TimeTimeString, .TimeYear

.TimeMinute 点域

.TimeMinute 点域显示上次从 “I/O 服务器” 更新标记值时的分钟数部分。

类别

标记

用法

Tag.TimeMinute

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 0 到 59。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime,
.TimeHour, .TimeMsec, .TimeMonth, .TimeSecond,
.TimeTime, .TimeTimeString, .TimeYear

.TimeMonth 点域

.TimeMonth 点域显示上次从 “I/O 服务器” 更新标记值时所属的月份 (1-12)。

类别

标记

用法

Tag.TimeMonth

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 1 到 12。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime,
.TimeHour, .TimeMinute, .TimeMsec, .TimeSecond,
.TimeTime, .TimeTimeString, .TimeYear

.TimeMsec 点域

.TimeMsec 点域显示上次从 “I/O 服务器” 更新标记值时的毫秒部分。

类别

标记

用法

Tag.TimeMsec

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 0 到 999。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear

.TimeSecond 点域

.TimeSecond 点域显示上次从 “I/O 服务器” 更新标记值时的秒数部分。

类别

标记

用法

Tag.TimeSecond

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 0 到 59。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeTime, .TimeTimeString, .TimeYear

.TimeTime 点域

.TimeTime 点域显示从 “I/O 服务器” 更新标记值时的时间标签，使用从午夜之后已经过去的毫秒数来表示。

类别

标记

用法

Tag.TimeTime

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 0 到 86399999。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTimeString, .TimeYear

.TimeTimeString 点域

.TimeTimeString 点域显示从 “I/O 服务器” 中更新标记值时的时间。

类别

标记

用法

Tag.TimeTimeString

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

消息（只读）。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeYear

.TimeYear 点域

.TimeYear 点域显示从 “I/O 服务器” 更新标记值时的四位数的年份。

类别

标记

用法

Tag.TimeYear

参数

Tag

任何离散、整型、实型、消息、间接模拟或间接消息标记。

数据类型

整型（只读）。

有效值

以四位数形式表示的任何年份。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString.

查看 I/O 标记的质量信息

您可以使用一组质量点域来确保在 “I/O 服务器” 与 InTouch 应用程序之间发送的数据的完整性。质量点域代表项目的数据值的质量状态。通过此质量属性，可以非常轻松地监视在网络节点之间发送的 InTouch 数据的完整性。

Wonderware 数据质量标准基于建议的标准 “过程控制 OLE” (OLE for Process Control, 简称 OPC), 后者又基于各种 “现场总线数据质量规格”。

质量数据格式

通过将值指定给一组 InTouch .Quality 点域, “I/O 服务器” 可以就发送给客户端的数据质量报告六个互异的状态。Quality 点域的低八位 (最低有效字节) 目前以三个位域的形式定义: “质量” (Q)、“子状态” (S) 以及 “限制” (L), 它们的格式如下: QQSSSLL。客户端应用程序无法与服务器通讯时, .Quality 点域的值 为 0。

下表显示 “I/O 服务器” 使用 .Quality 点域报告的数据质量状态:

质量状态	十进制值	十六进制值	最高有效字节 XXXXXXXX	最低有效字节 QQSSSLL	质量	质量子状态	限制
良好	192	0x00C0	00000000	11000000	Q=3	S=0	L=0
上嵌位 (超出范围)	86	0x0056	00000000	01010110	Q=1	S=5	L=2
下嵌位 (超出范围)	85	0x0055	00000000	01010101	Q=1	S=5	L=1
不能转换	64	0x0040	00000000	01000000	Q=1	S=0	L=0
通讯失败	24	0x0018	00000000	00011000	Q=0	S=6	L=0
不能访问点	4	0x0004	00000000	00000100	Q=0	S=1	L=0

关于数据质量点域

.Quality 点域表示上次收到数据时数据值的质量。仅当数据改变由 “I/O 服务器” 提供时， SuiteLink 与 DDE 协议才会向客户端（例如， WindowViewer）发送更新的质量。因此仅当收到新的数据值时，才能观察到质量改变。某些 “I/O 服务器” 可以在与数据关联的质量发生改变时，随当前的数据值一起发送更新的质量。

使用 SuiteLink 与 DDE 协议时，可能无法直接引用服务器项目值的质量。要这样做，“I/O 服务器” 必须直接支持 Item.Quality。没有这种支持，项目便无法继续给出提示，因此 .Quality 点域的值永远保持为 0。

TestProt “I/O 服务器” 模拟器不直接支持 Item.Quality。在使用 Quality（质量）菜单命令修改质量时，该模拟器不发送新的数据值。

如果要观察 I/O 项目的数据质量，但 “I/O 服务器” 又不直接支持 Item.Quality 的寻址，请定义一个查看服务器项目的 InTouch I/O 标记，然后监视该 InTouch 标记的 .Quality。如果超出了标记限制，请考虑在脚本中使用 IOSetRemoteReferences() 函数来动态调整 I/O 点。

SuiteLink 与 DDE 协议不会将连接状态或 “I/O 服务器” 状态的其它改变诠释为发送给客户端的质量项目。因此，项目的质量不一定要指出当前的数据服务器状态，也不一定要指出客户端 - 服务器连接的当前状态。“I/O 服务器” 进程可以停止，而 .Quality 域并不改变。如果通讯链接丢失，.Quality 域的值不一定会改变。

使用 DDE 或 SuiteLink 内部状态项目来监视 “I/O 服务器” 连接。

.Quality 点域

.Quality 点域以数值形式评估 “I/O 服务器” 所提供的数据的质量。

类别
标记

用法

Tag.Quality

参数

Tag

任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

整型（只读）。

有效值

值的范围是 0 到 255。

另请参阅

.QualityLimit, .QualityStatus, .QualitySubstatus

示例

```
IF IOTag.Quality <> 192 THEN
    LogMessage("This data is not Good, assuming
    high-byte of 2-byte quality is zero");
    LogMessage("Better to check .QualityStatus to avoid
    this assumption");
ENDIF;
```

.QualityLimit 点域

.QualityLimit 点域显示所连接的 “I/O 服务器” 提供的数据值的质量限制。

类别

标记

用法

Tag.QualityLimit

参数

Tag

任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

整型（只读）。

有效值

0 = 无限制

1 = 低限制

2 = 高限制

3 = 常数

另请参阅

.Quality

.QualityLimitString 点域

.QualityLimitString 点域显示所连接的 “I/O 服务器” 提供的
数据值的质量限制字符串。

类别

标记

用法

Tag.QualityLimitString

参数

Tag

任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

消息（只读）。

另请参阅

.Quality, .QualityLimit

.QualityStatus 点域

.QualityStatus 点域显示 “I/O 服务器” 所提供的数据值的整数
质量状态。

类别

标记

用法

Tag.QualityStatus

参数

Tag

任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

整型（只读）。

有效值 (SSSS)

0 = 不佳

1 = 不定

3 = 良好

示例

```
IF IOTag.QualityStatus <> 3 THEN  
    LogMessage("This data is not Good!");  
ENDIF;
```

另请参阅

.Quality, .QualitySubStatus

.QualityStatusString 点域

.QualityStatusString 点域显示 “I/O 服务器” 所提供的数据值的质量状态字符串。

类别

标记

用法

`Tag.QualityStatusString`

参数

Tag

任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

消息（只读）。

另请参阅

.QualityStatus, .QualitySubStatus, .Quality

.QualitySubstatus 点域

.QualitySubstatus 点域显示 “I/O 服务器” 所提供的数据值的质量子状态。

类别

标记

用法

`Tag.QualitySubstatus`

参数

Tag

任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

整型（只读）。

有效值 (SSSS) 与 (QQ)

“不佳” 质量 (QQ=0) 的子状态 (SSSS)。

0 = 非具体

1 = 配置错误

2 = 未连接

3 = 设备故障

4 = 传感器故障

5 = 最后已知值

6 = 通讯故障

7 = 停止服务

“不定”质量 (QQ=1) 的子状态 (SSSS)。

- 0 = 非具体
- 1 = 最后可用值
- 4 = 传感器不精确
- 5 = 超出工程单位
- 6 = 次正常

“良好”质量 (QQ=3) 的子状态 (SSSS)。

- 0 = 非具体
- 6 = 本地重写

另请参阅

.QualityStatus, .QualitySubStatus, .Quality

.QualitySubstatusString 点域

.QualitySubstatusString 点域显示 “I/O 服务器” 所提供的数值的质量子状态字符串。

类别

标记

用法

Tag.QualitySubstatusString

参数

Tag

任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

消息（只读）。

另请参阅

.QualityStatus, .QualitySubstatus, .Quality

在运行时初始化与重置 I/O 连接

WindowViewer 在启动 InTouch 应用程序时会启动所有的 I/O 对话。您也可以在 InTouch 应用程序运行期间手工重新启动 I/O 对话。通过使用 WindowViewer 命令或脚本，可以在运行时初始化与重置 I/O 连接。

您也可以指定 I/O 连接应该根据它们的缺省值来重新初始化。如果选择此选项，则重新初始化“访问名”时使用缺省设置，而忽略当前的设置。要根据“访问名”重新初始化 I/O 对话，InTouch 应用程序必须已经定义“访问名”。

使用命令重新初始化 I/O 连接

WindowViewer 中的**特别**菜单包含一组命令，可用于重新初始化所有的 I/O 对话或是选择特定的 I/O 对话。

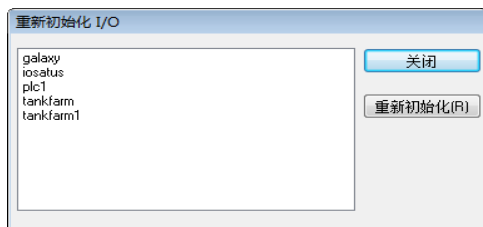
您可以使用 InTouch 的缺省设置来重新初始化“访问名”。使用缺省的重新初始化设置时，“访问名”会忽略指定给节点名、应用程序名以及主题的当前值。此时使用原始的“访问名”设置来重新初始化“访问名”。

要在运行时重新初始化所有的“访问名”

- 1 在**特别**菜单上，单击**重新初始化 I/O**。
- 2 单击**全部重新初始化**。此时重新初始化所有的“访问名”。

要在运行时重新初始化所选的“访问名”

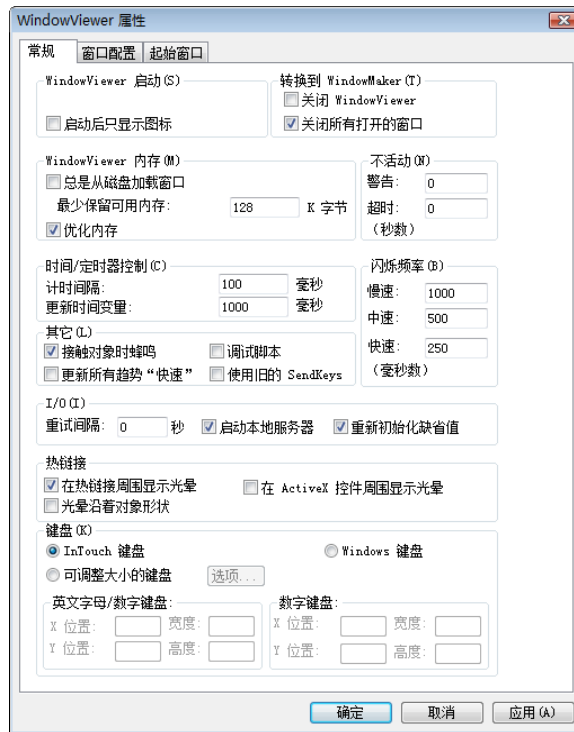
- 1 在**特别**菜单上，单击**重新初始化 I/O**，然后单击**选择**。此时出现**重新初始化 I/O**对话框，它显示一个“访问名”列表。



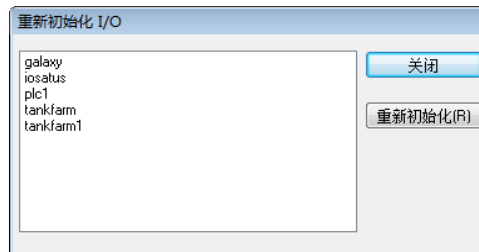
- 2 单击要重新初始化的一个或多个“访问名”，然后单击**重新初始化**。此时重新初始化所选的“访问名”。

要使用缺省设置重新初始化“访问名”

- 1 在 WindowMaker 中打开一个应用程序。
- 2 在特别菜单上，单击配置，然后单击 WindowViewer。此时出现“WindowViewer 属性”对话框，它显示一个选项列表。



- 3 在常规选项卡上，选择 I/O 区域中的重新启动初始化缺省值复选框。
- 4 单击确定。
- 5 在 WindowViewer 中打开应用程序。
- 6 在特别菜单上，单击重新启动初始化 I/O，然后单击选择。此时出现重新启动初始化 I/O 对话框。



- 7 选择要重新初始化的一个或多个“访问名”，然后单击重新启动初始化。此时忽略节点名、应用程序名以及主题的当前设置。同时使用原始的“访问名”设置来重新启动初始化“访问名”。

使用脚本重新初始化 I/O 连接

通过创建包含以下函数的脚本，可以重新初始化与一个或多个“访问名”的 I/O 连接：

- `IOReinitAccessName()`
- `IOReinitialize()`
- `IOStartUninitConversations()`

IOReinitAccessName() 函数

`IOReinitAccessName()` 函数初始化与特定“访问名”的 I/O 连接。

类别

I/O 通讯

语法

```
IOReinitAccessName("AccessName", Default);
```

参数

AccessName

要重新初始化的“访问名”。

Default

Default = 1。I/O 重新初始化过程使用从 WindowMaker 中指定的原始“访问名”缺省值。

Default = 0。I/O 重新初始化过程使用指定给“访问名”的当前节点、应用程序以及主题值。

附注

缺省设置由“访问名”配置面板中的各个设置以及 WindowViewer 配置中的各个设置（重试间隔、启动本地服务器、重新初始化缺省值）确定。

示例

本例使用指定给节点、应用程序以及主题的缺省值重新初始化与 `AccessName1` 的 I/O 连接。

```
IOReinitAccessName("AccessName1", 1);
```

本例使用指定给节点、应用程序以及主题的当前值重新初始化与 `AccessName2` 的 I/O 连接。

```
IOReinitAccessName("AccessName2", 0);
```


IOReinitialize() 函数

IOReinitialize() 函数首先关闭为 InTouch 应用程序定义的所有活动的 I/O 连接，然后重新启动它们。

类别

其它

语法

```
IOReinitialize();
```

参数

无。

附注

IOReinitialize() 函数与 WindowViewer 特别菜单上的**重新初始化 I/O** 命令执行相同的操作。

示例

本例关闭任何活动的 I/O 连接，然后重新启动为 InTouch 应用程序定义的所有 I/O 连接。

```
IOReinitialize();
```

IOStartUninitConversations() 函数

WindowViewer 开始运行 InTouch 应用程序时，会自动处理启动请求以启动所有的 I/O 对话。如果 “I/O 服务器” 程序没有响应 WindowViewer 的启动请求，则可以使用 IOStartUninitConversations() 脚本函数来强制 WindowViewer 再次试图启动 I/O 对话。

类别

其它

语法

```
IOStartUninitConversations();
```

参数

无。

附注

IOStartUninitConversations() 函数与 WindowViewer 特别菜单上的**启动未初始化的对话**命令执行相同的操作。

示例

本例强制 WindowViewer 再提交一个初始化请求，以启动为 InTouch 应用程序定义的所有 I/O 连接。

```
IOStartUninitConversations();
```

使用访问名的故障转移功能

您可以指定让 InTouch HMI 在主 “I/O 服务器” 遇到通讯问题时自动切换到辅助 “I/O 服务器”。这就是所谓的 I/O 故障转移。

配置故障转移

您可以指定让 InTouch 应用程序在无法继续与主 “I/O 服务器” 通讯时切换到用于故障转移的辅助 “I/O 服务器”。

设置故障转移时，需要指定故障转移死区。故障转移死区是从主 “访问名” 切换到辅助 “访问名” 之前延迟的秒数。表达式为真所持续的时间或是 I/O 通讯故障所持续的时间超过死区时段长度时，InTouch HMI 便会触发故障转移。故障转移死区设置为 0 或是为空时，检测到 I/O 通讯故障后会立即触发故障转移。

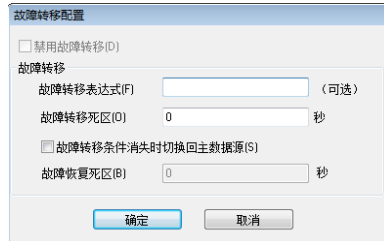
要配置 “访问名” 的故障转移

- 1 如果需要，停止 WindowViewer。
- 2 在**特别菜单**上，单击**访问名**。此时出现**访问名**对话框，显示一系列定义的所有 “访问名”。
- 3 从列表中选择要添加故障转移服务器的 “访问名”。
- 4 单击**修改**。此时出现**修改访问名**对话框。
- 5 单击**启用辅助数据源**。此时展开**修改访问名**对话框。

- 6 执行以下操作：
 - 在**节点名**框中，输入辅助 “I/O 服务器” 的节点名。

- 在**应用程序名**框中，输入将从中获取数据的辅助“I/O 服务器”程序的程序名。
- 在**主题名**框中，输入要从辅助 I/O 数据源中访问的主题名。
- 在**要使用的协议**区域，选择 **DDE** 或 **SuiteLink** 作为辅助“I/O 服务器”通讯协议。
- 在**要对服务器提示时**区域中，为辅助 I/O 数据源选择**提示所有项**或**只提示激活项**。

7 单击**故障转移**。此时出现**故障转移配置**对话框。



- 8 在**故障转移表达式**框中输入可选的故障转移表达式，或双击该框以选择一个标记。如需有关故障转移表达式的详细信息，请参阅第 124 页的“强制故障转移到备份访问名”。
- 9 在**故障转移死区**框中，输入以秒为单位的故障转移死区长度。
- 10 如果希望启用在故障转移条件消失之后从辅助“访问名”切换回主“访问名”，选择**故障转移条件消失时切换回主数据源**。
缺省值是不切换回主“访问名”。如果选择**故障转移条件消失时切换回主数据源**，则**故障转移配置**对话框中的**故障恢复死区**选项变为可以选择的。
- 11 从**故障恢复死区**中，输入以秒为单位的故障恢复死区长度。
在表达式与任何关联的 I/O 通讯故障消失后所持续的时间超过死区时段之后，InTouch HMI 便会触发故障恢复，以恢复到主“访问名”。表达式留为空白或是为 0 时，I/O 通讯故障条件消失后便会立即发生故障恢复。
- 12 单击**确定**以关闭**故障转移配置**对话框。
- 13 单击**确定**以关闭**修改访问名**对话框。

编辑故障转移对的访问名参数

要编辑作为故障转移对的一部分的“访问名”参数，必须已经使用辅助 I/O 数据源给“访问名”配置故障转移。

要编辑故障转移对的“访问名”参数

- 1 如果需要，停止 WindowViewer。
- 2 在**特别**菜单上，单击**访问名**。此时出现**访问名**对话框。
- 3 选择“访问名”对，然后单击**修改**。此时出现**修改访问名**对话框，显示主“访问名”与辅助“访问名”的参数。

- 4 更改主“访问名”与辅助“访问名”的“访问名”参数。
- 5 单击**确定**以关闭**修改访问名**对话框。
- 6 单击**关闭**以关闭**访问名**对话框。

删除访问名的故障转移

要删除“访问名”的故障转移，必须已经使用辅助 I/O 数据源给“访问名”配置了故障转移。

要删除“访问名”对的故障转移

- 1 在**特别**菜单上，单击**访问名**。
- 2 选择“访问名”对，然后单击**修改**。此时出现**修改访问名**对话框。
- 3 清除**启用辅助数据源**复选框。
- 4 单击**确定**。此时“访问名”对的故障转移功能便已禁用。

强制故障转移到备份访问名

您可以在没有遇到故障转移的情况下，手工在“访问名”的主数据源与辅助数据源之间切换。这就是所谓的**强制故障转移**。要强制进行故障转移，必须已经使用辅助 I/O 数据源给“访问名”配置了故障转移。

您可以使用故障转移表达式或 `IOForceFailover()` 脚本函数来强制进行故障转移。

故障转移表达式

故障转移配置对话框显示一个**故障转移表达式**选项，用以包含触发故障转移的标记或表达式。在下图显示的对话框中，**故障转移表达式**中输入的值是内存离散标记 `failover`。



将 `failover` 表达式设置为真（例如，通过将 `failover` 标记设置为真），可以将“访问名”从主（假）I/O 数据源切换到辅助（真）I/O 数据源。

IOForceFailover() 函数

IOForceFailover() 脚本函数在“访问名”的主数据源与辅助数据源之间进行切换。每次激活脚本函数时，活动的 I/O 节点在主节点与辅助节点之间切换。

通常，IOForceFailover() 函数是与按钮或另一个窗口对象关联的脚本中的一部分。操作员从应用程序窗口中选择该对象，以便强制进行故障转移。操作员再次单击该对象之后，IOForceFailover 函数强制将 I/O 连接切换回之前活动的 I/O 节点。

类别

I/O 通讯

语法

```
IOForceFailover("AccessName");
```

参数

AccessName

配置了故障转移的“访问名”。

示例

Acc1 “访问名”有“主”数据源与“辅助”数据源，其中“主”数据源处于活动状态。脚本运行时，Acc1 发生故障转移并切换到“辅助”数据源。

```
IOForceFailOver("Acc1");
```

临时禁用故障转移功能

您可以手工禁用在“访问名”的主 I/O 节点与辅助 I/O 节点之间进行切换的故障转移功能。InTouch 系统的各个组件已经启动但尚未准备就绪的很短时间，便是需要临时禁用故障转移的典型情况。在各个组件都稳定之后，便可以恢复故障转移切换功能。

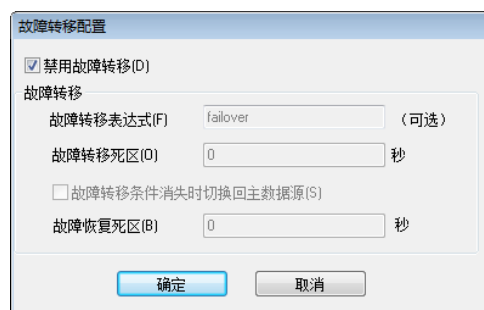
要禁用“访问名”的故障转移，必须已经使用辅助 I/O 数据源给“访问名”配置了故障转移。

您可以通过两种方法来手工禁用故障转移切换功能：

- 从**故障转移配置**对话框中选择**禁用故障转移**。
- 运行包含 IODisableFailover() 函数的脚本。

禁用故障转移配置选项

故障转移配置对话框包含一个**禁用故障转移**选项，可防止“访问名”在主 I/O 节点与辅助 I/O 节点之间进行切换。在下图显示的对话框中，**禁用故障转移**选项设置为活动状态。



您必须编辑“访问名”定义才能将**禁用故障转移**选项设置为活动状态。只要“访问名”的该选项处于活动状态，便会禁用故障转移。

IODisableFailover() 脚本函数

您可以在脚本中使用 IODisableFailover() 函数给指定的“访问名”禁用故障转移功能。IODisableFailover() 禁用除 IOForceFailover() 脚本函数方法之外的所有故障转移方法的切换功能。

类别

I/O 通讯

语法

IODisableFailover ("AccessName",Option);

参数

AccessName

配置了故障转移的“访问名”。

Option

1 = 禁用故障转移

0 = 启用故障转移

附注

“访问名”可以指定为字符串，也可以是使用其它 InTouch 标记或函数提供的字符串值。

示例

本例禁用 ModbusPLC1 “访问名”的故障转移功能。

```
IODisableFailover ("ModbusPLC1",1)
```

本例启用 ModbusPLC1 “访问名”的故障转移功能。

```
IODisableFailover ("ModbusPLC1",0)
```

使用脚本检索故障转移对的有关信息

您可以编写包含一些函数的脚本，这些函数返回“访问名”的主 I/O 数据源、辅助 I/O 数据源以及活动的 I/O 数据源的状态。通常，操作员在强制进行故障转移之前，会运行脚本以确定“访问名”的辅助 I/O 数据源的状态。

要创建脚本来返回“访问名”信息，必须已经使用辅助 I/O 数据源给“访问名”配置了故障转移。

IOGetAccessNameStatus() 函数

IOGetAccessNameStatus() 脚本函数返回一个整数，指出“访问名”的主 I/O 数据源、辅助 I/O 数据源或活动的 I/O 数据源的连接状态。

通常，IOGetAccessNameStatus() 返回值与整型标记关联。标记的值可以驱动离散值显示动画链接，向操作员显示“访问名”活动的 I/O 数据源、主 I/O 数据源以及辅助 I/O 数据源的状态。

类别
其它

语法

```
Result=IOGetAccessNameStatus ("AccessName", Mode);
```

参数

AccessName

要返回其状态的现有“访问名”。

Mode

指定给此参数的值确定查询故障转移对哪个“访问名”的当前状态。

- 0 - “访问名”的活动 I/O 数据源的状态
- 1 - “访问名”的主 I/O 数据源的状态
- 2 - “访问名”的辅助 I/O 数据源的状态

结果

返回值	描述
-1	“访问名”中存在配置错误。“访问名”不存在，或未定义“访问名”的辅助 I/O 数据源。
0	未成功连接到所请求的 I/O 数据源。
1	成功连接到所请求的 I/O 数据源。

附注

IOGetAccessNameStatus() 函数通常在脚本中用于确定辅助 IO 数据源（当前在不活动状态）的状态。在强制进行故障转移之前，操作员运行此脚本以验证辅助连接的状态。

示例

本例返回 ModbusPLC1 “访问名”的辅助 I/O 数据源的状态。
返回值与 ANStatus 标记关联。

```
ANStatus = IOGetAccessNameStatus ("ModbusPLC1",2)
```

IOGetActiveSourceName() 函数

IOGetActiveSourceName() 脚本函数返回访问名是否当前使用其主数据源或辅助数据源。

通常，IOGetActiveSourceName() 函数包含在与按钮或另一个窗口对象相关联的脚本中。随后，操作员从应用程序窗口中选择该对象，以请求应用程序的“I/O 服务器”的状态。

类别

其它

语法

```
Result=IOGetActiveSourceName ("AccessName");
```

参数**AccessName**

要返回其数据源名的现有“访问名”。

附注

IOGetActiveSourceName() 返回一个字符串，指出正在轮询“访问名”的主节点还是辅助节点。IOGetActiveSourceName() 函数可能的返回值如下：

Primary	正在轮询“访问名”的主节点。
Secondary	正在轮询“访问名”的辅助或故障转移节点。
Null	“访问名”的主节点与辅助节点都处于不活动状态。

示例

在本例中，返回值（Primary、Secondary 或 Null）指定给 ActiveServer 消息标记，用于确定 ModbusPLC1 “访问名”的当前活动节点。

```
ActiveServer = IOGetActiveSourceName ("ModbusPLC1");
```

监视 I/O 连接的状态

WindowViewer 包含一个内置的主题 **IOStatus**，用于监视 InTouch 应用程序与“**I/O 服务器**”（同 PLC 通讯）之间的特定 I/O 对话的状态。

备注 在 7.0 之前的 InTouch 版本中，此主题名是 **DDEStatus**。

您可以设置 **IOStatus** 主题以监视 I/O 对话。

使用 IOStatus 主题名

您可以预备 **IOStatus** 主题，以便监视“**WindowViewer**”与“**I/O 服务器**”之间的 I/O 通讯情况。在本例中，**WindowViewer** 与 Wonderware Simulate “**I/O 服务器**”通讯，后者又与“**I/O 服务器**”中定义的主题名为 **PLC1** 的 PLC 进行通讯。

备注 **Simulate** 是一个用作培训工具的通用型 Wonderware DAServer。**Simulate** 位于 `c:\program files\common files\ArchestraA` 文件夹。

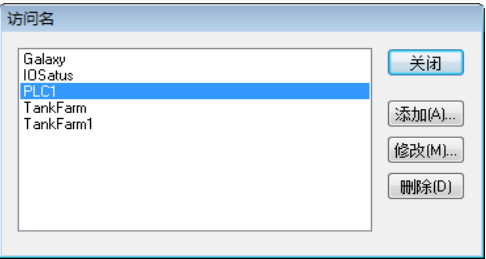
要监视 I/O 通讯的状态

- 1 在 **WindowMaker** 中打开一个应用程序。
- 2 打开“**标记名字典**”。
- 3 创建一个“**I/O 离散**”标记。

使用 **IOStatus** 监视 I/O 对话时，必须为要监视的“访问名”定义至少一个 I/O 型标记。



- 4 单击**访问名**以便将标记指定给“访问名”的定义，此定义将 **IOStatus** 定义为主题名。



请注意，**PLC1** 的“访问名”定义当前已经存在。

- 5 选择 **PLC1**，然后单击**修改**。

在本例中，由于标记与**主题名**相同，因此很容易找到包含正确**主题名**的“访问名”。

- 6 单击**取消**以关闭对话框，并返回到最初的**访问名**对话框。

- 7 单击**添加**。此时出现**添加访问名**对话框。

- 8 执行以下操作：

- a 在**访问名**框中，输入 **IOStatus**。
- b 由于打算要从 **WindowViewer** 中监视状态，因此请在**应用程序名**框中输入 **View**。
- c 在**主题名**框中，输入 **IOStatus** 作为 InTouch 内部主题。
- d 选择**只提示激活项**。

- 9 单击**确定**以关闭对话框。此时再次出现最初的访问名对话框，并且列表中显示新的“访问名” **IOStatus**：



- 10 单击**关闭**以关闭对话框，并将新的访问名关联到 **I/O 离散** 标记。

- 11 在**项目**框中，为要监视的实际主题名输入“访问名”。在本例中是 **PLC1**。



- 12 由于标记与**主题名**相同，因此可以选择**将标记名用作项目名**，并自动将它作为**项目**进行输入。

备注 使用内置的主题 **IOStatus**（在 InTouch 7.0 版之前是 **DDEStatus**）监视 I/O 对话时，在**访问名**框中输入的名称总是同时用于**项目**。

在 Excel 中使用主题名 IOStatus

通过输入相同的信息作为电子表格单元格的公式，可以使用 Excel 来监视 I/O 状态。例如，要监视与前面的操作程序所介绍的相同主题，请在单元格中输入以下公式：

```
=view|IOStatus!'PLC1'
```

监视 I/O 服务器通讯状态

对于使用的每个主题名，都可以使用内置的离散项 **Status** 来监视与“**I/O 服务器**”程序之间的通讯状态。发生通讯故障时，**Status** 项目设置为 0。与“**I/O 服务器**”程序之间正常通讯时，**Status** 项目设置为 1。

备注 使用 **IOStatus** 项目来监视主题的状态时，所监视的主题至少必须有一个活动的 I/O 点。

从 InTouch HMI 中，通过定义标记并将它关联到给设备配置的主题（通过将 **Status** 用作“项目名”），可以读取服务器通讯的状态。例如，如果 WindowViewer 正在使用 Wonderware Modbus DAServer 与 PLC 进行通讯，则“访问名”定义类似于下例：

要监视与 PLC1 主题之间的所有通讯的状态，请创建以下标记定义：

在 Excel 中，通过在单元格中输入以下公式，可以读取 PLC 通讯的状态：

```
=SIMULATE | PLC1 ! ' STATUS '
```

从其它应用程序中访问 InTouch 标记数据

其它应用程序向 InTouch HMI 请求数据值时，它必须知道三个 I/O 地址项。请遵循这些 InTouch I/O 地址惯例。

VIEW (*应用程序名*) 确定包含数据元素的 InTouch 运行时程序。

TAGNAME (*主题名*) 是读取 / 写入标记时总是要使用的字。

ActualTagname (*项目名*) 是在 InTouch 的“标记名字典”中给项目定义的实际标记。

例如，要从 Excel 访问 InTouch HMI 中的数据值，需要为将写入数据的单元格指定一个“DDE 远程调用”公式：

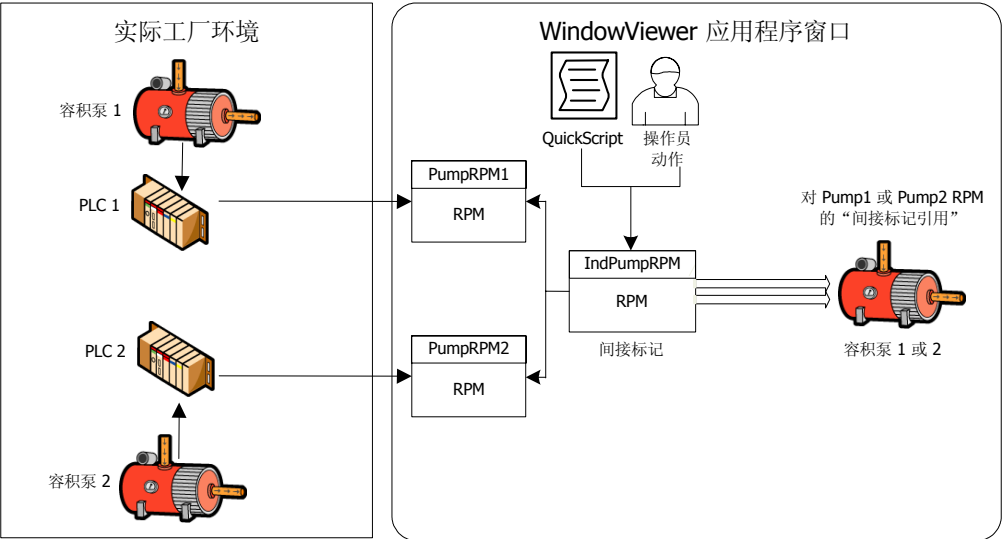
```
=VIEW|TAGNAME!'ActualTag_name'
```

第 6 章

定义间接标记

通过使用间接标记，可以创建包含一些窗口对象的应用程序，这些对象显示多个标记的值。

下图显示应用程序窗口中的一个泵对象。该泵对象根据从间接标记中设置的值来代表两个可能的过程泵。QuickScript 或操作员动作负责选择与间接标记关联的源标记。



间接标记可以最大限度缩短开发时间。由于单个窗口对象可以代表生产环境中运行的多个过程，因此可以创建更少的应用程序窗口。

在脚本中使用间接标记

您可以使用脚本将输入源标记指定给间接标记。通过将源标记的名称指定给间接标记的 `.Name` 点域，可以将输入源标记指定给该间接标记。

例如，如果创建一个间接模拟标记 `IndPumpRPM`，则通过使用与下例类似的脚本语句，可以将两个 `PumpRPM` 源标记指定给该间接模拟标记：

```
IF PumpNo == 1 THEN
    IndPumpRPM.Name = "PumpRPM1";
ELSE
    IndPumpRPM.Name = "PumpRPM2";
ENDIF;
```

间接标记赋值脚本可以由应用程序事件或操作员动作（如单击窗口按钮）触发。

将某个间接标记的值赋给另一个源标记时，此间接标记的作用就好像它自己是源标记。如果源标记的值发生改变，间接标记便会反映出这个变化。如果间接标记的值发生改变，源标记的值也会相应地改变。

由于间接标记的 `.Name` 点域是一个简单的字符串，因此可以在运行时动态指定间接标记目标。例如，如果创建一个“数据改变” `QuickScript`，每次 `Number` 标记的值发生改变时该脚本都会运行，则指定给间接 `IndPumpRPM` 标记的源标记也会相应地改变：

```
IndPumpRPM.Name = "PumpRPM" + Text(Number, "#" );
```

此脚本运行时，模拟标记 `Number` 的值转换为文本，并附加到字符串 `PumpRPM` 上。如果 `Number` 等于 1，则这将间接标记 `IndPumpRPM` 的名称设置为 `PumpRPM1`。

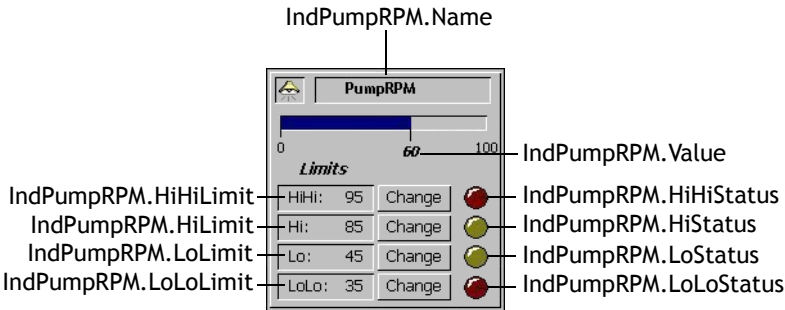
间接模拟型标记同时可用于整型与实型标记。只要标记类型相同，间接标记可以映射到任何其它标记。

您也可以将保留属性指定给间接标记。利用这种保留功能，当应用程序再次启动时，间接标记会保留最新的标记赋值情况。

给本地标记使用间接标记

间接标记通常用于本地“标记名字典”中定义的标记。间接本地标记可用于创建显示本地标记的多个属性的可视化对象。例如，您可以在应用程序窗口中创建一个面板。该面板包含一些可选项，这些可选项链接到指定给不同点域的间接本地标记。在下例中，操作员修改链接到本地间接标记 PumpRPM 的点域报警限。

下图显示一个面板，该面板包含一些指向泵 RPM 报警限的动画链接。间接标记将不同本地标记的属性指定给报警限面板。



要将面板重定向到适当的标记，请在 QuickScript 中包含一条语句。

```
Indirect_tag_name.Name = "tag_name";
```

在这个示例脚本中，tag_name 是本地“标记名字典”中定义的实际标记的名称。该脚本运行时，与这个本地标记关联的所有点域值都可以通过间接标记由应用程序对象进行访问。

给远程引用使用间接标记

远程间接标记引用与本地标记引用不同。远程引用的语法是：

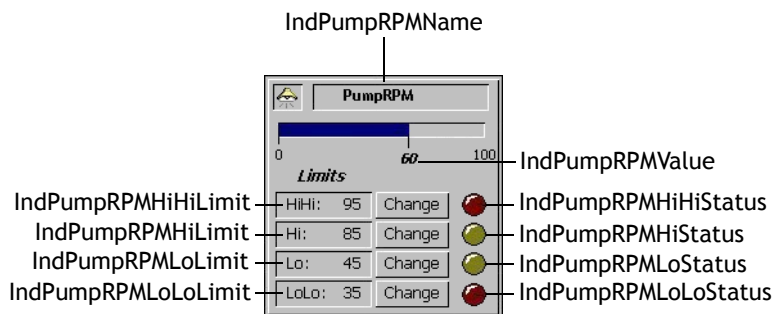
AccessName:Item

其中

- *AccessName* 是任何有效的 InTouch “访问名”。
- *Item* 是任何有效的 “项目名”，此 “项目名” 受 “访问名” 定义中指定的 “I/O 服务器” 支持。

使用远程引用时，服务器给客户端返回一个值，而不是一个标记结构。该值包含时间标签与质量标签。因此，除与值、时间以及质量相关的标记点域之外，指定给远程引用的间接标记无法访问任何其它标记点域。例如，间接标记无法通过远程引用来访问标记属性以指定报警限。

一个可能的解决方案是创建包含一组间接标记的面板。下图显示一个可修改泵的报警限的面板。



在本例中，面板使用 10 个间接标记，这些标记都与隐式 .Value 点域关联。报警面板重定向到远程 InTouch 节点 TagServer1 上的远程引用标记 IndPumpRPM。InTouch “访问名” 配置如下：

访问名：TagSource1
 节点名：TagServer1
 应用程序名：View
 主题名：Tagname

要将该面板重定向到远程引用标记 PumpRPM，请运行以下 QuickScript：

```
IndPumpRPMName.Name = "TagSource1:PumpRPM.Name";
IndPumpRPMValue.Name = "TagSource1:PumpRPM";
IndPumpRPMHiHiLimit.Name = "TagSource1:PumpRPM.HiHiLimit";
IndPumpRPMHiLimit.Name = "TagSource1:PumpRPM.HiLimit";
IndPumpRPMLoLimit.Name = "TagSource1:PumpRPM.LoLimit";
IndPumpRPMLoLoLimit.Name = "TagSource1:PumpRPM.LoLoLimit";
```

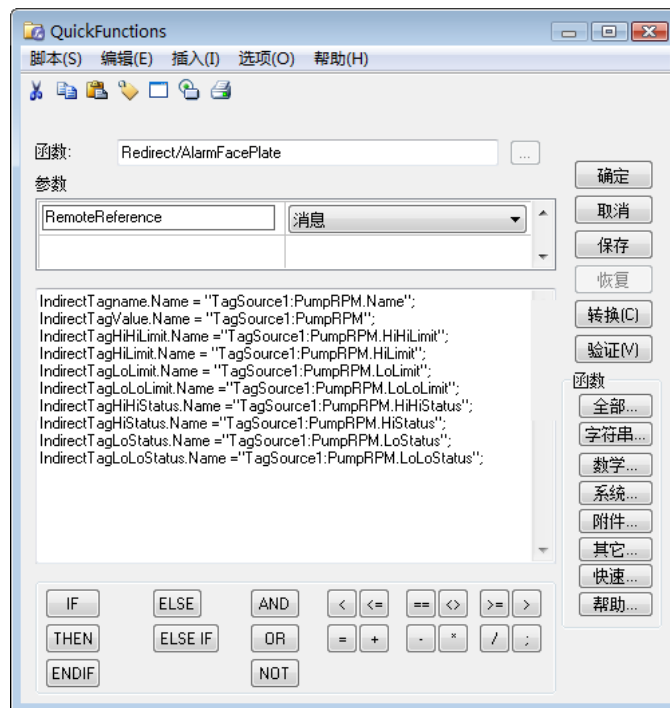
```

IndPumpRPMHiHiStatus.Name =
    "TagSource1:PumpRPM.HiHiStatus";
IndPumpRPMHiStatus.Name = "TagSource1:PumpRPM.HiStatus";
IndPumpRPMLoStatus.Name = "TagSource1:PumpRPM.LoStatus";
IndPumpRPMLoLoStatus.Name =
    "TagSource1:PumpRPM.LoLoStatus";

```

每次重定向面板时必须运行此脚本。另一个解决方案是创建一个 **InTouch QuickFunction**，允许您编写单个脚本并将远程引用的名称传递给它。通过使用调用相同 **QuickFunction** 的多个面板，可以减少脚本的编码量。

例如，通过使用一组类似的脚本命令，可以定义一个 **QuickFunction RedirectAlarmFacePlate**：



您可以调用 **RedirectAlarmFacePlate** 函数来处理整个重定向过程。要完成这项工作，必须使用另一个 **InTouch QuickScript** 来调用该函数。例如：

```
CALL RedirectAlarmFacePlate ("TagSource1:PumpRPM");
```

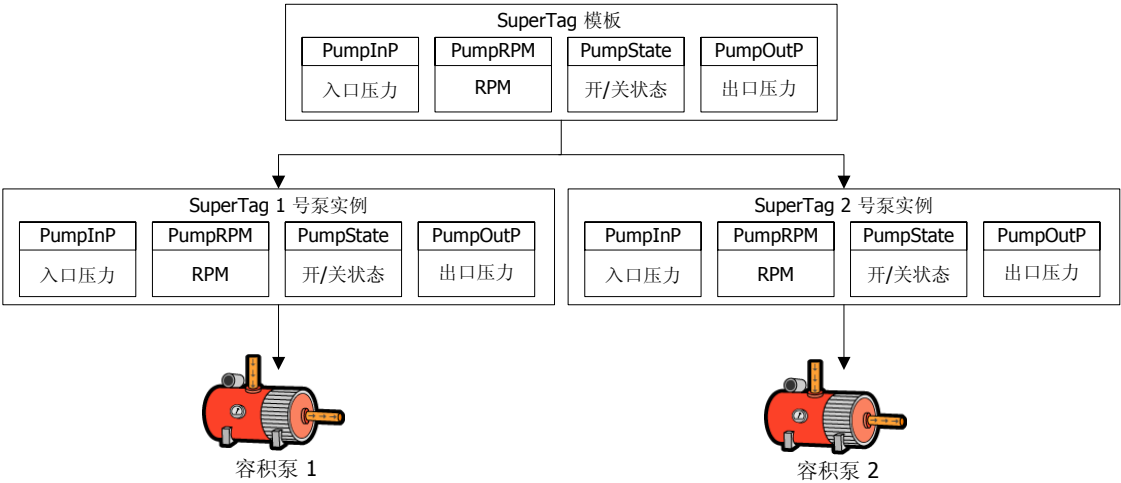

第 7 章

定义可复用的标记结构

SuperTag 是一组相关标记的模板。属于某个 SuperTag 模板的标记与生产过程中某个组件的公共属性关联。

SuperTag 可以节省开发时间。您可以复制一个 SuperTag 模板，然后为具有相同属性的所有过程组件创建单独的实例，而不必为生产过程中的每个组件都创建一组标记。

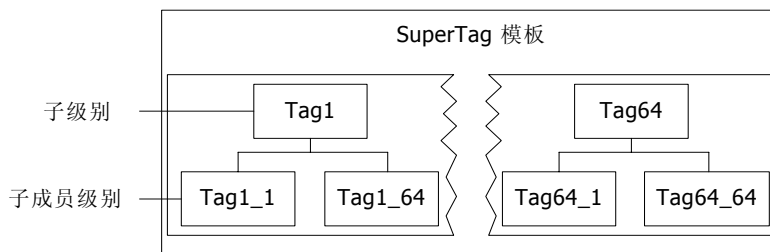
下图显示一个 SuperTag 模板，该模板包含一组与泵数据关联的相关标记。此模板可以复制，以便为过程中所有相同的泵创建该 SuperTag 的实例。



属于 SuperTag 实例的所有标记的行为都与普通标记完全相同。成员标记可以指定为 InTouch 离散、整型、实型及消息等数据类型。这些标记支持趋势、报警以及所有的标记点域。

SuperTag 模板可以按两个嵌套级别来组织管理自己的成员标记。SuperTag 模板最多可以包含 64 个内嵌的子标记。每个子标记最多可以包含 64 个子成员标记。这可以在 SuperTag 模板中总共提供 4095 个标记。

下图显示标记在 SuperTag 模板中的组织方式。



一个 SuperTag 父模板嵌入另一个 SuperTag 模板时，内嵌的标记就变成了子成员。

创建 SuperTag 父模板之后，“标记名字典”在**标记类型**对话框中将它列为一种标记类型。在新建标记时，可以将该模板当作一种标记类型来直接选择。您不必重新启动 WindowMaker，便可以定义一些标记来使用新创建的 SuperTag 类型。

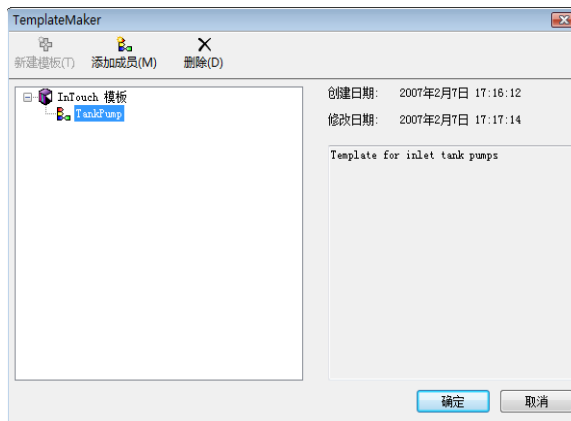
所有的 SuperTag 模板都保存在 C:\Documents and Settings\All Users\Application Data\Wonderware\InTouch 文件夹下的 supertag.dat 文件中。

定义 SuperTag 模板

您可以使用 **TemplateMaker** 实用程序来创建、编辑以及删除 **SuperTag** 模板及其成员标记。创建 **SuperTag** 模板之后，“标记字典”在**标记类型**对话框中将它与间接标记形式的模板列为标记类型。

要创建 SuperTag 模板

- 1 在**特别**菜单上，单击 **TemplateMaker**。此时出现 **TemplateMaker** 对话框。
- 2 单击**新建模板**。此时出现**新建模板**对话框。
- 3 在**名称**框中，为 **SuperTag** 模板输入唯一的名称。
模板名最多可包含 10 个字母数字字符。模板名的第一个字符必须是字母。
- 4 作为可选项，在**描述**框中输入描述该模板的注释。
- 5 单击**确定**。此时再次出现 **TemplateMaker** 对话框，且在窗口中列出新模板。添加模板之后，**添加成员**与**删除**按钮变为活动状态。



要将标记添加到 SuperTag 模板

- 1 打开 **TemplateMaker** 对话框。
- 2 从列表中选择 **SuperTag** 模板。
- 3 单击**添加成员**以显示**新建成员标记**对话框。
- 4 执行以下操作：
 - 在**名称**框中，输入要添加到该模板的成员标记的名称。
成员标记名最多可包含 10 个字母数字字符。第一个字符必须是字母。

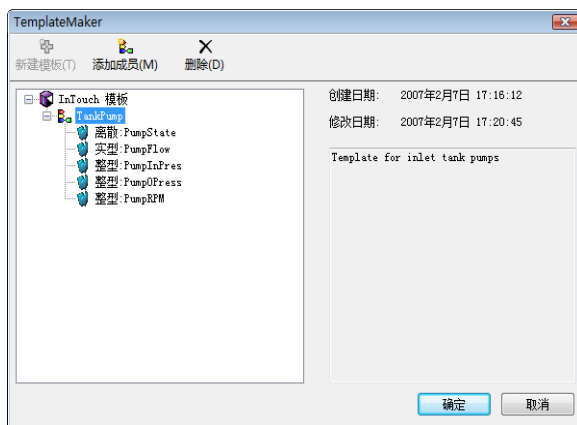
备注 如果要使用远程引用，请仔细考虑标记名的长度。如需有关 **SuperTag** 引用限制的详细信息，请参阅第 150 页的“引用 **SuperTag** 成员”。

- 在**类型**框中，输入该成员的标记类型，或单击**类型**并从列表中选择标记类型。

标记类型可以是离散、整型、实型、消息，也可以是另一个 SuperTag 模板。缺省条件下，在 TemplateMaker 中定义成员标记时，所有的成员标记都设置为内存型。不过在“标记名字典”中定义模板实例时，必须指定是让模板成员保持为内存标记还是更改为 I/O 标记。

- 作为可选项，在**注释**框中输入描述该成员标记的注释。

- 单击**确定**。新成员标记的名称出现在 TemplateMaker 对话框中 SuperTag 父模板的下方。



- 标记名包含指定给该标记的所选标记类型作为前缀，并使用冒号进行分隔。

示例：

离散: PumpState

- 重复这些步骤将所有的标记都添加到 SuperTag 模板。

TemplateMaker 对话框在 SuperTag 模板名列出添加的所有标记。

创建 SuperTag 模板之后，**标记类型**对话框将 SuperTag 模板及其间接形式作为可选标记类型同时显示出来。



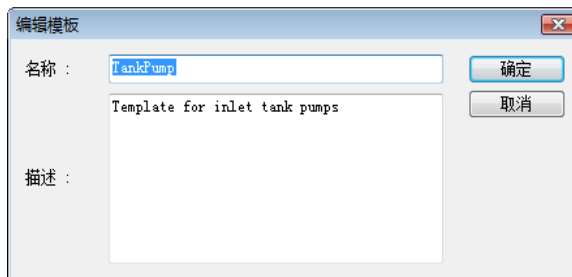
编辑 SuperTag 模板与成员标记

您可以随时修改 SuperTag 模板或成员标记。从某个模板衍生的现有 SuperTag 实例不会继承对模板所作的更改。不过，所有新的实例都将继承修改后的模板上所发生的更改。

您可以删除整个 SuperTag 模板，也可以删除属于该模板的所选成员标记。已删除的模板不再列在“标记名字典”的**标记类型**对话框中。

要编辑现有的 SuperTag 模板或成员标记

- 1 打开 **TemplateMaker** 对话框。
- 2 双击要编辑的 SuperTag 模板名或成员标记。此时出现**编辑模板**或**编辑成员标记**对话框。该对话框显示 SuperTag 模板或成员标记的当前定义。



- 3 进行更改。
- 4 单击**确定**。
- 5 此时 **TemplateMaker** 对话框进行刷新，并显示进行编辑更改的日期与时间。

要删除 SuperTag 模板或成员标记

- 1 打开 **TemplateMaker** 对话框。
- 2 选择要删除的 SuperTag 模板名或成员标记。
- 3 单击**删除**。此时出现一条消息，要求确认是否删除所选的项目。
- 4 单击**是**以删除所选的名称。
此时 **TemplateMaker** 列表进行刷新，并清除已删除的项目。

重要 SuperTag 模板实例的成员标记无法删除。例如，如果 PumpRPM 是 SuperTag 模板 TankPump 的成员标记，则无法从任何 TankPump 实例中删除它。您只能从 SuperTag 模板中删除标记。

创建 SuperTag 实例

SuperTag 模板与模板实例不同。实例是 SuperTag 模板在 InTouch 应用程序中实际实现的版本。

模板与实例之间最重要的区别是父模板名将替换为实例标记的名称。子模板名与子成员标记不发生改变。

使用标记名字典创建 SuperTag 实例

您可以使用“标记名字典”来创建 SuperTag 实例。定义新的 SuperTag 实例时，“标记名字典”自动创建所有的成员标记与子成员标记。

要从模板创建 SuperTag 实例

- 1 打开“标记名字典”。
- 2 单击**新建**。
- 3 在**标记名**框中，为新的 SuperTag 实例输入所需的名称。
SuperTag 实例名最多包含 10 个字符。实例名与普通的 InTouch 标记遵循相同的命名规则。
- 4 单击**类型**以显示**标记类型**对话框。
- 5 从列表中选择 SuperTag 模板名。



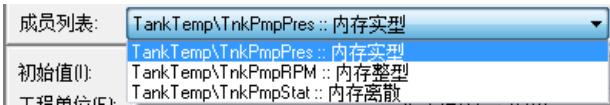
- 6 单击**确定**。此时**标记名字典**展开，以显示附加选项。



此时在**标记名**框中输入的新标记变成所选 SuperTag 模板中所有成员标记的父对象。

- 7 设置标记的属性。执行以下操作：

- a 在**成员列表**框中，从 SuperTag 模板列表中选择**一个**标记。



- b 从**数据访问**中选择**内存**或 **I/O**，以显示相应的 “内存” 或 I/O 详细资料对话框。
 - c 像对待标准的 InTouch 标记那样输入详细资料。
 - d 从列表中选择其余的成员标记并配置它们。
- 8 为 SuperTag 实例的成员标记指定所有的详细资料之后，单击**关闭**。

使用标记名字典复制 SuperTag 实例

您可以使用 “标记名字典” 来复制现有的实例。在复制实例之后，这些标记立即就可以在动画链接与 InTouchQuickScript 中使用。

要从 “标记名字典” 中复制 SuperTag 实例

- 1 打开 “标记名字典”。
- 2 单击**选择**以显示**选择标记**对话框，列出为应用程序定义的一系列标记。
- 3 从列表中选择要用作新实例的模板的 SuperTag 实例。
- 4 单击**确定**。此时所选模板的名称出现在**标记名**框中。
- 5 单击**新建**。此时出现一条消息，要求确认是否复制 SuperTag 实例。
- 6 单击**是**。此时出现**输入名称**对话框，提示输入新 SuperTag 的名称。
- 7 根据标准的标记命名惯例输入最多包含 10 个字符的名称。
- 8 单击**确定**。此时新的 SuperTag 实例出现在 “标记名字典” 中。
- 9 如果需要，像对待普通的 InTouch 标记那样编辑各个成员标记。
- 10 单击**关闭**。

使用标记名字典将标记添加到 SuperTag 实例

您可以使用“标记名字典”将某个标记添加为现有 SuperTag 实例的成员。

添加标记时，输入 SuperTag 实例的准确名称，后跟反斜杠 (\) 分隔符以及新成员标记的名称。例如：

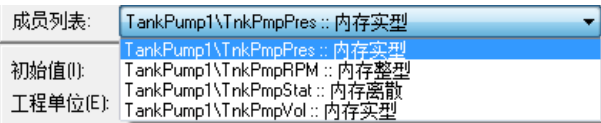
Pump_8\PumpSTS

备注 如果打算使用 ArchestrA 批量导入实用程序（ArchestrA 批量导入实用程序）将标记从 InTouch 应用程序移植到 Application Server，请参阅第 150 页的“使用批量导入实用程序导入 SuperTag”以获取有关替换标准反斜杠分隔符的详细信息。

要将标记添加到 SuperTag 实例

- 1 打开“标记名字典”。
- 2 要将标记添加到 SuperTag 实例，请执行以下操作：
 - a 单击**新建**。
 - b 在**标记名**框中，输入 SuperTag 实例的准确名称，后跟反斜杠 (\) 分隔符和新成员标记的名称。
 - c 单击**类型**。
 - d 为要添加到实例的新成员标记选择标记类型。
 - e 单击**确定**。此时出现该成员标记类型的详细资料对话框。
 - f 像对待普通的 InTouch 标记那样输入所需的详细资料。
- 3 单击**保存**。
- 4 单击**选择**。
- 5 选择添加了成员标记的 SuperTag 实例。
- 6 单击**确定**。

在**成员列表**框中，属于该 SuperTag 模板的所有成员标记都会列出。



此时新成员标记出现在列表中。

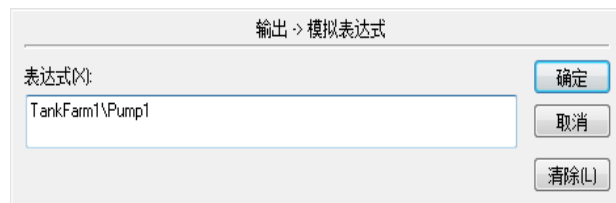
创建 SuperTag 的其它方法

除 TemplateMaker 外，还可以使用以下方法创建 SuperTag：

- 在“动画链接表达式”输入框中
- 在 InTouch QuickScript 中
- 在使用 InTouch DBLoad 实用程序加载到应用程序的外部文件中

备注 使用其它方法创建成员时，TemplateMaker 不在 SuperTag 模板定义中显示该成员。

通过动画表达式或 InTouch QuickScript 创建 SuperTag 时，必须使用有效的 SuperTag 格式。例如：



备注 如果在动画表达式或 QuickScript 中指定的 SuperTag 实例与成员标记当前尚未定义，则程序会提示您定义标记。单击**确定**。此时出现“标记名字典”，并显示所创建的 SuperTag 实例与成员标记。

以下是有效语法的示例：

```
ParentInstance\ChildMember  
ParentInstance\ChildMember\Submember
```

以下是无效语法的示例：

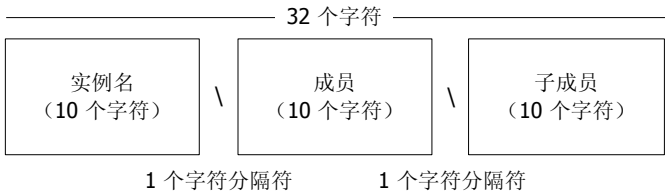
```
ParentInstance\  
ParentInstance\ChildMember\
```

如果使用的格式无效，则出现错误消息框，指出 SuperTag 语法包含错误。

引用 SuperTag 成员

InTouch 标记名最多可以包含 32 个字符。每个 SuperTag 的“父实例\子成员\隶属成员”最多可以包含 32 个字符。标记名的最大长度使得在对 SuperTag 的引用上存在着一定的限制。

SuperTag 引用最多可以包含两个模板（父实例\子成员），引用深度为一个成员，具体如下图所示。



SuperTag 模板中的每个成员都可以使用当前用于访问标准 InTouch 标记类型的点域的标准格式进行访问。InTouch 中凡是可以使用标准标记的场合也都支持 SuperTag 引用语法。例如，有效的 SuperTag 点域引用是：

```
TankFarm\Tank1\Pump1RPM.RawValue
```

远程标记引用同样支持 SuperTag。例如，有效的 SuperTag 远程引用是：

```
PLC1:"TankFarm\Tank1\Pump1RPM.RawValue"
```

使用批量导入实用程序导入 SuperTag

您可以使用“ArchestrA 批量导入实用程序”将标记定义从 InTouch 转换为“Application Server”对象结构。

“ArchestrA 批量导入实用程序”可以更有效地将 InTouch 标记移植到“Application Server”。除标准的 InTouch 标记之外，“ArchestrA 批量导入实用程序”还可以移植 SuperTag。

如果打算将 SuperTag 从 InTouch 应用程序移植到“Application Server”，请将 SuperTag 引用中的反斜杠字符 (\) 替换为受支持的 ArchestrA 符号，如下划线 (_)。

示例：

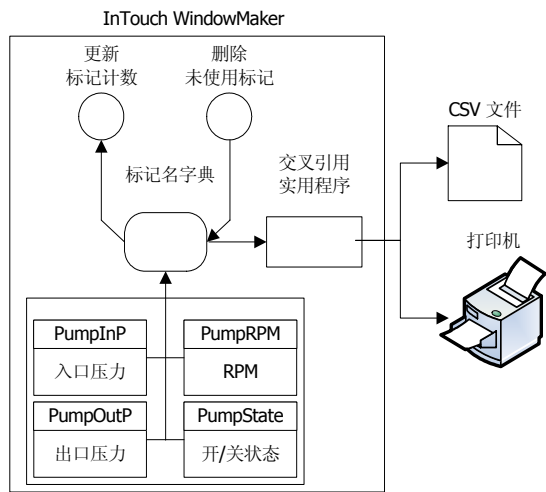
```
TankFarm_Tank1_Pump1RPM.RawValue
```

第 8 章

减少标记使用

InTouch 应用程序中可使用的最大标记数取决于您的 Wonderware 许可证。Wonderware 许可证防止正在运行的应用程序的并发标记计数超出最大限制。

您可以跟踪 InTouch 应用程序中的标记使用。下图显示 "交叉引用" 实用程序如何通过分析 "标记名字典" 的内容来生成标记使用情况报告。



您应该维持运行 InTouch 应用程序所需的最少标记数。维持最少标记计数的一种方法是删除未使用的标记。您必须在删除未使用的标记之前更新标记计数。

确定标记使用

InTouch HMI 为本地“标记名字典”中定义的所有标记维持一份使用计数。标记计数不包括内部系统标记。

远程标记不在“标记名字典”中定义。每次在应用程序内引用远程标记时，InTouch 标记计数均会递增。如需有关 InTouch 如何统计远程标记引用的详细信息，请参阅第 153 页的“根据许可证确定最大远程标记数”。

在删除未使用的标记之前，确认执行以下操作：

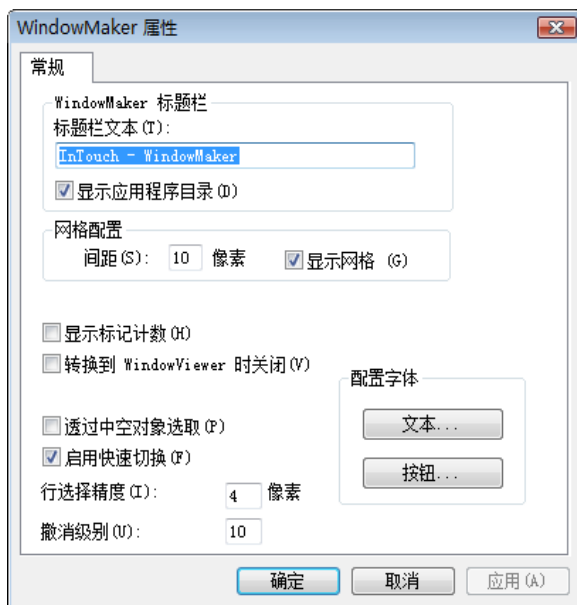
- 关闭 WindowViewer。
- 更新本地标记计数和对远程标记的引用。
- 生成“交叉引用”标记报告。
- 使用“交叉引用”实用程序来查找应用程序中的标记使用。
- 在 WindowMaker 中保存应用程序。

确定标记计数

您可以更新应用程序“标记名字典”中当前定义的本地标记的计数。您也可以更新应用程序中引用的其它节点上远程标记的计数。

要显示本地标记计数

- 1 在 WindowMaker 中打开一个 InTouch 应用程序。
- 2 在“工具”视图中，展开配置列表，然后选择 WindowMaker。此时出现 WindowMaker 属性对话框。



- 3 选择显示标记计数。

选择此选项时，必须读取整个“标记名字典”才能更新显示的标记计数。更新到“标记名字典”可能需要更长的时间。

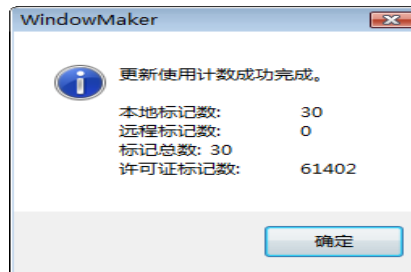
- 4 单击**确定**。此时出现一条消息，指出必须重新启动 WindowMaker 才能使配置更改生效。
- 5 关闭 WindowMaker。
- 6 在 WindowMaker 中重新启动应用程序。
应用程序“标记名字典”中所定义的本地标记数显示在 WindowMaker 菜单栏的右侧。

文件(F) 编辑(E) 查看(V) 排列(A) 文本(T) 线条(L) 特别(S) 窗口(W) 帮助(H) [30 个已定义的标记]

要更新包括远程引用在内的标记总数。

- 1 在 WindowMaker 中打开一个 InTouch 应用程序。
- 2 关闭所有打开的窗口。
- 3 在**特别**菜单上，单击**更新使用计数**。此时出现一条消息，要求确认是否更新使用计数。
- 4 单击**是**。

计算完成之后，对话框显示应用程序的本地标记数以及远程标记引用数。



该对话框还包括标记总数和基于 InTouch 许可证的最大标记使用数。

根据许可证确定最大远程标记数

Wonderware 产品许可证决定了您的 InTouch 应用程序中可以使用的标记数。许可证不仅决定本地标记计数，还决定了引用远程节点上标记源的标记数。

如需有关 InTouch 许可证如何决定应用程序中可以使用的远程引用标记数的详细信息，请参阅附录 B “InTouch 许可证”。

查找标记的使用位置

通过使用“交叉引用实用程序”，可以生成一份在线报告来显示 InTouch 应用程序中的标记使用情况。

“交叉引用实用程序”报告显示以下位置中的本地标记、远程标记以及 SuperTag：






- 动画链接
- 向导
- 所有类型的 InTouch 脚本
- QuickFunction
- ActiveX 控件
- 可选的 InTouch 组件，如 SPC Pro、“SQL 访问管理器”以及 Recipe Manager。

该报告列出在其中定义了标记的窗口的名称，以及链接到该标记的所有对象的窗口坐标。该报告还可用于查看包含指定标记的任何 QuickScript 或 QuickFunction。

您可以搜索该报告来查找特定的项目。搜索通配符可供指定更广泛的搜索准则，以搜索仅与部分搜索字符串匹配的标记名。

- 在字符串中使用星号 (*) 可以搜索名称与星号前面及 / 或后面的那些字符匹配的所有标记。例如，搜索字符串 Prod* 可以找到 ProdPump1 与 ProdTank2。
- 在搜索字符串中使用问号 (?) 替换单个字符。例如，搜索字符串 Pump? 可以找到 Pump1 与 Pump2。

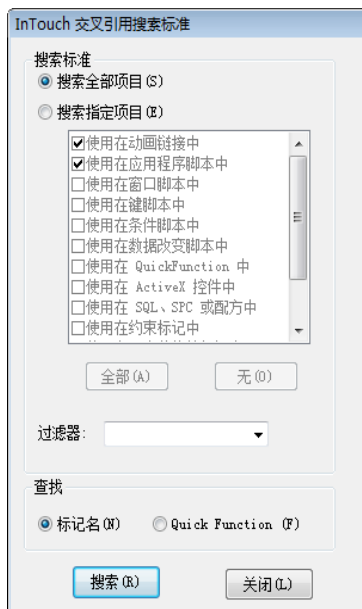
“InTouch 交叉引用实用程序”报告中出现一些图标，以显示状态与使用情况。

图标	描述
	标记或 SuperTag 定义在应用程序的“标记名字典”中，但未指定给某个对象。
	标记或 SuperTag 用在动画链接或 InTouch QuickScript 中。
	标记或 SuperTag 指定给某个动画链接。
	标记或 SuperTag 用在某个“应用程序”脚本中。
	对于选择的所有脚本，都会显示此图标。双击脚本名可以查看标记。

图标	描述
	标记或 SuperTag 用在某个“窗口”脚本中。双击窗口名可以查看脚本。
	标记或 SuperTag 用在某个“数据改变”脚本中。双击脚本名可以查看它。
	标记或 SuperTag 用在某个“条件”脚本中。选择图标可以展开视图，以显示脚本的条件及其类型。例如，\$Hour==12 On True。双击脚本名可以查看它。
	标记或 SuperTag 用在某个“键”脚本中。双击图标可以展开视图，以显示指定给脚本的键与脚本类型。例如，F2 On Key Down。
	标记或 SuperTag 用在某个 QuickFunction 中。双击图标可以展开视图，以显示使用标记或 SuperTag 的 QuickFunction 的名称。双击脚本名可以查看它。
	标记或 SuperTag 用在某个“ActiveX 事件”脚本中。双击图标可以展开视图，以显示“ActiveX 事件”脚本。
	按窗口进行交叉引用时，此图标出现在使用了所显示的标记或 SuperTag 的窗口的名称前面。双击图标可以查看该窗口中使用的所有标记。
	标记或 SuperTag 用在 SPC Pro 应用程序中。双击图标可以显示使用了标记或 SuperTag 的 SPC 数据集的名称。
	显示的标记或 SuperTag 用在 SQL 应用程序中。双击图标可以显示使用了标记或 SuperTag 的“SQL 绑定列表”的名称。
	显示的标记或 SuperTag 用在 Recipe Manager 应用程序中。
	此标记用作报警约束。双击图标可以显示用作报警约束的所有标记的名称。

要使用“交叉引用”实用程序创建报告

- 1 在 WindowMaker 中打开一个 InTouch 应用程序。
- 2 在特别菜单上，单击交叉引用。此时出现 InTouch 交叉引用搜索标准对话框。



- 3 在**搜索标准**区域中，选择搜索所有标记，或者将搜索范围限制为指定的 InTouch 应用程序组件。
 - 单击**搜索全部项目**，以搜索全部标记或 SuperTag 项目。
 - 单击**搜索指定项目**，将标记搜索限制为特定的 InTouch 应用程序组件。对于要在其中搜索标记的那些组件，选择它们旁边的复选框。
- 4 在**过滤器**框中输入字符串，进一步将搜索限制为名称与此字符串匹配的标记。如果不希望按标记名设置搜索过滤器，请将**过滤器**框留为空白。

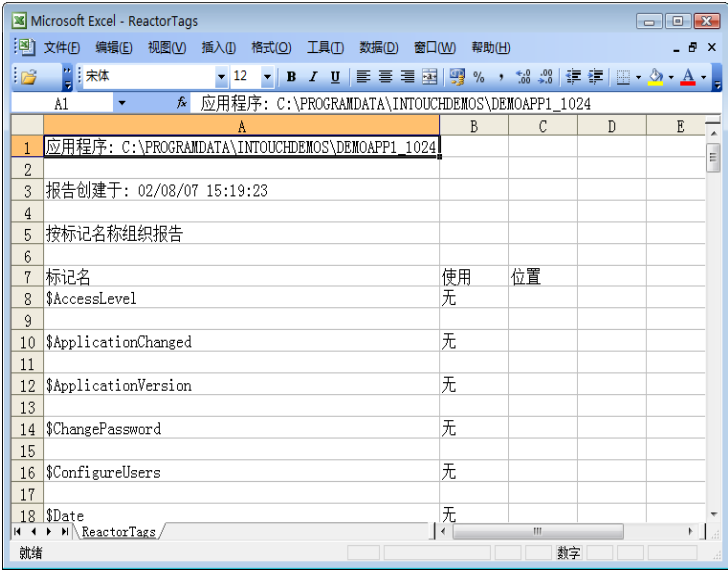
- 5 单击**搜索**以开始对标记进行交叉引用搜索。
- 搜索完成之后，会有一个报告列出符合搜索标准的所有标记。如果不使用过滤器，则报告显示当前应用程序“标记名字典”中定义的所有标记。



保存与打印标记交叉引用列表

您可以将交叉引用标记列表保存到文件，然后使用任何支持 .csv 文件格式的应用程序查看它。

交叉引用标记列表文件按名称、在应用程序中的使用方式以及在窗口的名称列出所有标记。您可以使用 Excel 或支持 .csv 文件的任何其它程序打开交叉引用标记列表文件。

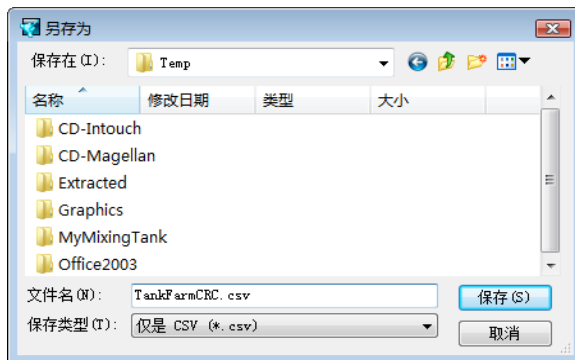


您也可以打印“标记名字典”的内容。通过打印“标记名字典”的内容，可以显示应用程序中使用的数据库项、窗口以及脚本。您也可以指定查看其中每一项的详细资料级别。

您可以将此报告发送到打印机，或将它保存到文件。

要保存交叉引用文件

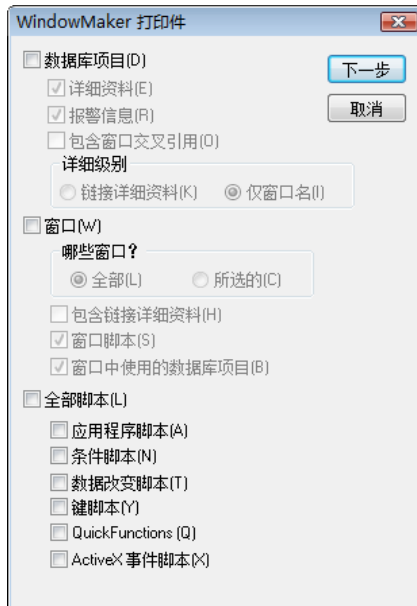
- 1 在 **InTouch 交叉引用实用程序** 对话框中，单击**另存为**。此时出现**另存为**对话框。



- 2 指定文件的名称与位置。
- 3 单击**保存**。此时标记使用情况文件保存在指定的文件夹中。

要打印“标记字典”的内容

- 1 在 WindowMaker 中打开一个 InTouch 应用程序。
- 2 在 WindowMaker 文件菜单上，单击**打印**。此时出现 **WindowMaker 打印件** 对话框。



- 3 如果希望打印“标记名字典”中的标记信息，请选择**数据库项目**。

如果选择**数据库项目**，以下选项变为活动状态：

- 选择**详细资料**，以便在报告中包含标记的详细资料。
- 选择**报警信息**，以便在报告中包含标记报警信息。
- 选择**包含窗口交叉引用**，以打印包含窗口交叉引用的所有“标记名字典”项目。如果选择此选项，请指定要打印的详细资料级别。

链接详细资料打印标记所在位置与动画链接的详细资料。

仅窗口名仅打印交叉引用的窗口的名称。

- 4 选择**窗口**，以打印应用程序窗口中使用的数据库项目的列表。如果选择**窗口**，请指定要打印的窗口选项：
 - **全部**打印应用程序中所有窗口的数据库项目。
 - **所选的**仅打印特定窗口的数据库项目。此时出现**要打印的窗口**对话框。选择应用程序中要打印的窗口，然后单击**确定**。
 - **包含链接详细资料**打印窗口的链接详细资料。
 - **窗口脚本**打印与窗口关联的脚本。
 - **窗口中使用的数据库项目**打印窗口中使用的标记。
- 5 选择**全部脚本**打印应用程序中使用的所有脚本。

通过清除**全部脚本**复选框，可以仅打印所选类型的脚本。然后，对于希望打印的每种脚本类型，选择对应的复选框。
- 6 单击**下一步**。此时出现**选择输出目标**对话框。
- 7 选择此选项以打印“标记名字典”的内容，或将输出发送到文本文件。
- 8 单击**打印**。

删除未使用的标记

在更新使用计数之后，可以从 InTouch 应用程序中删除未使用的标记。您可以从“标记名字典”中一次删除一个标记，也可以一次删除多个标记。

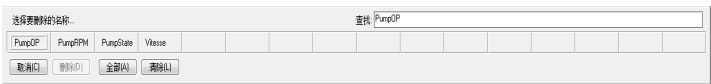
从应用程序中删除包含标记的窗口或在链接脚本中更改标记时，标记计数不自动更新。InTouch 将这些标记视为正在使用，因此会阻止删除它们。

在可以删除这些标记之前，必须先更新标记计数以便从总数中删除未使用的标记。如需有关更新标记计数的详细信息，请参阅第 152 页的“确定标记计数”。

注意 仅用于报警的标记没有使用计数，因此可能会不小心删除。要确保使用计数中包含仅用于报警的标记，需要在窗口或 QuickScript 中使用它们。

要删除多个未使用的标记

- 1 如果 WindowViewer 正在运行，请关闭它。
- 2 在 WindowMaker 中打开一个 InTouch 应用程序。
- 3 在特别菜单上，单击**删除未使用标记**。此时出现**选择要删除的名称**对话框，列出未使用的各个标记。



- 4 单击一个或多个标记名以选择要删除的标记。您也可以通过单击**全部**来选择列出的所有标记。
- 5 单击**删除**。此时出现一条消息，要求确认是否删除所选的标记。
- 6 单击**确定**进行确认。

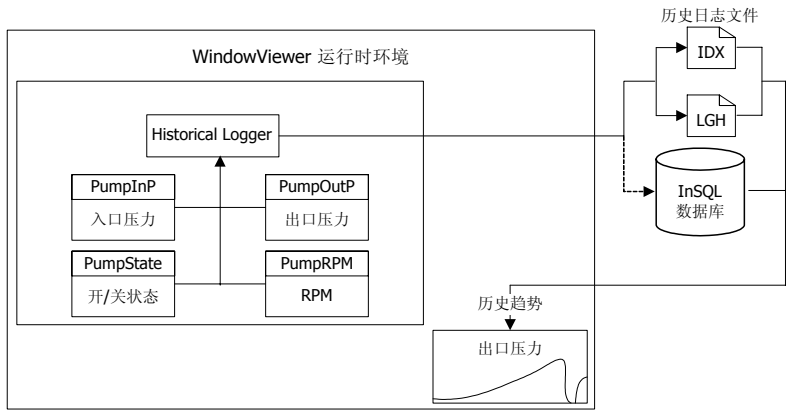
第 9 章

记录标记值

InTouch 应用程序运行期间，其标记值可以记录并永久保存。您可以使用这些日志数据来创建历史趋势图，以显示一段时间内工厂生产过程的某些方面。

备注 对于大型应用程序，或是要创建详细报告时，应使用 Wonderware Historian 保存 InTouch 历史数据。如需有关配置历史记录功能的详细信息，请参阅 Wonderware Historian 文档。

下图显示一个泵的标记数据，这些数据保存在历史日志文件或可选的 Wonderware Historian 数据库中。每次标记值的变化超出指定的记录死区范围时，InTouch Historical Logger 便写入一个日志项。



InTouch HMI 创建两个日志文件。一个文件包含按专有格式存储的记录数据。另一个文件是这些数据的索引。

这两个日志文件的名称按以下格式指定：

YYMMDD00.LGH 和 **YYMMDD00.IDX**

其中：

- YY** 创建日志文件的年份的后两位数字。
- MM** 创建文件的月份的两为数字 (01-12)
- DD** 创建文件的日期的两位数字 (01-31)
- 00** 日志文件名中的常数值 00

每天的记录周期从午夜开始并在午夜结束。**Historical Logger** 在午夜将最后一些项目写入活动的日志文件，并将它们进行归档。此时创建要在第二天使用的两个新文件，并在其中记录数据。

日志文件会保存指定的天数。超过保留期限的日志文件将会删除。如需有关设置日志文件保留天数的详细信息，请参阅第 164 页的“配置常规记录属性”。

配置历史记录功能

为 InTouch 应用程序配置历史记录功能时，需要执行三个主要任务：

- 配置要记录历史的标记
- 配置 InTouch 应用程序的常规记录属性
- 作为可选项，设置历史记录频率

配置要记录历史的标记

您可以从“标记名字典”中选择要记录历史的标记。所选标记的值发生改变时，**Historical Logger** 根据每个标记的记录死区与当前值来确定是否应该将一个项目写入日志。

如果将某个标记从要记录更改为不记录，则与该标记关联的数据将不再保存到日志文件。重新启用记录时会恢复记录。不过历史趋势会在禁用记录功能的期间显示一段空隙。

在 **WindowViewer** 运行应用程序期间，对标记记录功能所作的更改会被忽略。只有停止并重新启动正在运行的应用程序之后，对标记记录所作的更改才会生效。

您可以从“标记名字典”中为每个单独的标记配置记录功能。

要给标记启用历史记录功能

- 1 如果需要，停止 **WindowViewer** 中正在运行的应用程序。
- 2 使用 **WindowMaker** 打开该应用程序。
- 3 打开“标记名字典”。
- 4 从“标记名字典”的列表中选择要记录其数据的标记。

5 选择记录数据复选框。



标记名字典对话框包含与记录功能紧密关联的其它标记属性。

- **记录死区** 设置一个工程单位阈值，标记的值必须超过这个值才会写入日志文件。只有超出死区的新值才会写入日志文件。死区范围内的微小值变化将被忽略。
- **最小工程单位与最大工程单位** 属性将原始值缩放到一定的工程单位范围内。最小及最大工程单位属性设置缩放值的上下边界。

最小 / 最大工程单位确定趋势中显示的日志值的范围边界。缺省条件下，InTouch 历史趋势显示 0-100% 工程单位范围内的日志数据。

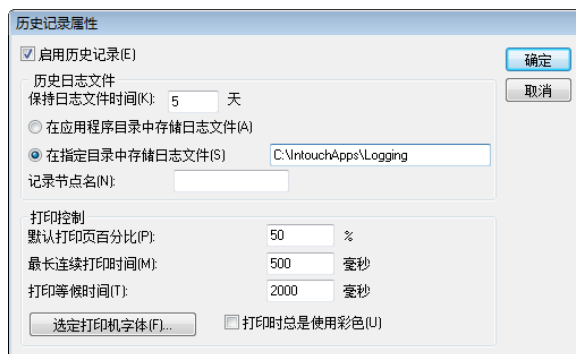
- 6 单击**保存**。
- 7 重复这些步骤，为要记录数据的每个标记启用记录功能。
- 8 完成时**关闭**，以关闭“标记名字典”对话框。

配置常规记录属性

您可以设置要应用于所选应用程序的常规记录属性。

要配置分布式或历史记录功能

- 1 如果需要，请关闭 WindowViewer 中正在运行的 InTouch 应用程序。
- 2 打开 **WindowMaker**。
- 3 在“工具”视图上，展开**配置**，然后选择**历史记录**。此时出现**历史记录属性**对话框。



如需有关设置趋势打印的详细信息，请参阅第 218 页的“在运行时打印趋势”。

4 选择**启用历史记录**复选框。

5 在**保持日志文件时间**框中，输入当前日期之前要保留日志文件的天数。

当天以及指定的保留期限内的日志文件都会保留下来。超过保留期限的日志文件将会删除。将此值设置为 0 时，则会无限期保留所有日志文件。

示例：

将保留期限设置为五天，并从当月的第一天开始记录。在当月的第七天，保留的是前五天的日志文件。当月第一天创建的日志文件会被删除。

设置保存记录数据的天数时，要考虑磁盘空间使用情况。如果硬盘可用空间不足，历史记录功能会停止。您必须释放磁盘空间才能恢复记录功能。

6 选择要在其中保存日志文件的文件夹位置。

历史记录属性对话框包含两个可用于设置日志文件存储文件夹位置的选项。

备注 存储日志数据的文件夹路径与文件名最多可以包含 55 个字符。

选择在**应用程序目录中存储日志文件**，以便将日志文件保存在创建记录数据的 InTouch 应用程序的相同文件夹中。

选择在**指定目录中存储日志文件**，以指定要用于存储日志文件的另一个文件夹。您可以按以下格式指定要用于存储日志文件的文件夹：

- Windows 文件夹路径，如 C:\History Log Files
- “通用命名惯例”（Universal Naming Convention，简称 UNC）路径，如 \\node\share\directory。

如果要将日志文件保存到分布式节点上，则必须按 UNC 路径格式来指定目录。

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 节点都会尝试将历史数据写入主应用程序。为避免这种情况，请将每个 NAD 节点上的历史数据配置为写入本地目录，而不是主应用程序节点。

7 在**记录节点名**框中，输入某个计算机的节点名，创建日志文件的 InTouch 应用程序在该计算机上运行。

8 单击**确定**以保存设置。

对记录功能的配置所作的更改将立即生效。记录功能将在下次运行应用程序时开始记录数据。

控制历史记录频率

作为可选项，您可以基于两个条件来指定要记录的项：

- 只要标记值的变化超出记录死区值一个工程单位值，则立即写入一个日志项。
- 按固定的间隔写入记录的所有标记的当前值。记录的所有标记对应的日志项都会写入日志，而不论它们的当前值如何。缺省的固定间隔为 60 分钟。

您可以接受缺省的固定记录间隔，也可以在 `intouch.ini` 文件中添加两个参数来更改这个间隔。

- *ForceLogging*

ForceLogging 以分钟为单位指定固定记录间隔的长度。*ForceLogging* 可以设置为 5 到 120 分钟之间的某个值。缺省值是 *ForceLogging=60*。

- *ForceLogCurrentValue*

ForceLogCurrentValue 指定按固定的记录间隔记录当前标记值，即使它仍然在上一个记录值的记录死区范围内。设置为 0 时，再次记录上一个记录值。缺省值是 *ForceLogCurrentValue=0*。

下例显示包含这两个记录参数的示例 `intouch.ini` 文件。

```
WinFullScreen=1
WinWidth=808
AlarmBufferSize=5000
ForceLogging=5
ForceLogCurrentValue=1
```

在本例中，标记值按照五分钟的间隔写入“历史日志”文件。

要修改历史记录频率

- 1 关闭 WindowMaker 与 WindowViewer。
- 2 在与 InTouch 应用程序相同的文件夹中找到 intouch.ini 文件。
- 3 编辑 intouch.ini 文件。
- 4 在 *ForceLogging* 语句中插入 5 与 120 之间的一个值。
- 5 插入 *ForceLogCurrentValue=1* 语句。
- 6 保存更改并关闭 intouch.ini 文件。
- 7 重新启动 WindowViewer。

在运行时启动与停止历史记录

应用程序正在运行时，可以使用 WindowViewer 特别菜单中的命令来手工停止然后再重新启动历史记录功能。

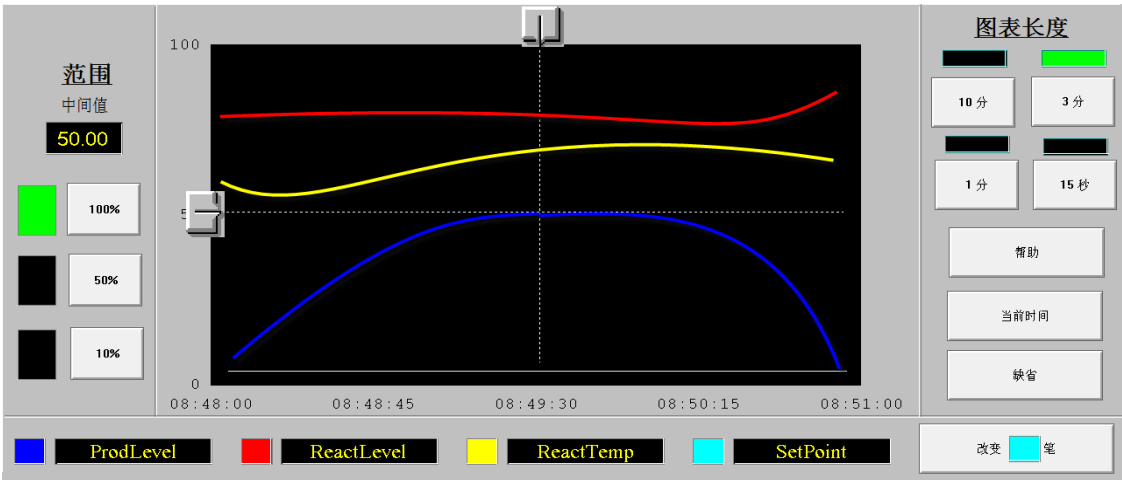
- **停止历史记录**命令可以在当前应用程序会话期间停止记录功能。记录功能在当前会话期间一直处于停止状态，直到再次手工启动它。
- 使用**停止历史记录**选项手工停止记录功能之后，**重新启动历史记录**命令可以重新启动记录功能。

您也可以在应用程序中添加一个按钮，然后编写一个包含 **\$HistoricalLogging** 系统标记的 QuickScript 来启动与停止历史记录功能。记录功能在将 **\$HistoricalLogging** 的值指定为 1 时启动。将 **\$HistoricalLogging** 的值指定为 0 时停止记录。如需有关 **\$HistoricalLogging** 系统标记的详细信息，请参阅第 39 页的“系统标记”。

第 10 章

绘制标记数据的趋势

您可以创建一些趋势图，以图形化的方式显示从 InTouch 应用程序采集的数据。WindowMaker 包含一组可用于创建历史与实时趋势的实用程序与向导。下例显示一个 InTouch “平均 / 散点” 趋势的示例。



您也可以使用一组趋势控件。通过使用这些控件，可以选择要在趋势中显示的数据及其在趋势中的显示方式。

您可以配置实时与历史趋势。这两种类型的趋势都包含一些配置选项，可用于设置趋势的数据采集间隔和视觉外观。

InTouch 趋势的类型

历史趋势显示过去采集并存储在 InTouch 数据储备库中的记录数据。

通过使用分布式历史系统，从可访问的网络节点上的任何 InTouch 历史记录文件中，都可以检索到历史数据。分布式历史系统将远程记录数据库涵括在内，从而扩展了历史趋势的检索能力。

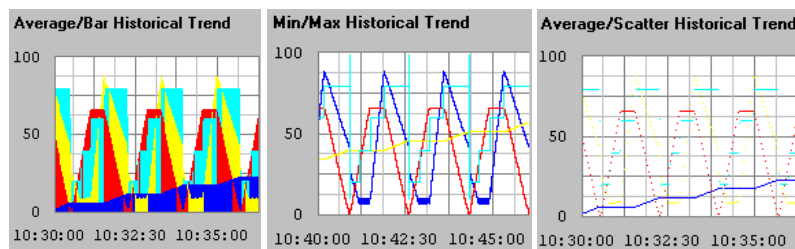
实时趋势会持续进行更新，可以在数据产生之后相对较短的时间段内显示它们。您可以使用 WindowMaker 的“实时趋势”工具在窗口中创建趋势对象。如果安装了可选的 16-Pen Trend（16 笔趋势）工具，则也可以创建实时趋势来最多显示 16 个标记的数据。

理解历史趋势

历史趋势显示过去一个连续时间段内的数据。与实时趋势不同，历史趋势只能通过脚本或操作员动作进行更新。

历史趋势使用图形化的表示法来最多显示八个标记的数据。通过将标记指定给趋势笔，可以指定要在历史趋势中显示的数据。

下图显示三种类型的 InTouch 历史趋势。



- “均值 / 棒图”历史趋势以条形图的形式显示一定时间间隔内数据点的平均值。
- “最小 / 最大”历史趋势以工程单位的百分比为纵坐标，以时间宽度为横坐标来显示所发生的变化。它强调的是时间流与变化率，而不是变化量。
- “平均 / 散点”历史趋势显示每个趋势时间间隔内数据点的平均值。

您可以创建一些被称作指示器的图形游标，以便根据指示器在趋势中的当前位置来访问趋势数据的详细信息。例如，操作员将指示器放在趋势上有可见数据的区域时，要在该位置上绘制趋势的所有数据库值的时间和值都会显示出来。

您也可以创建一些按钮或游标，以放大或缩小指示器间的区域；也可以缩放到数据范围，如最大值到最小值。此时可以显示完整图表或指示器之间区域的平均值和标准偏差。

历史趋势也可以按任何时间量进行滚动。您可以创建一些自定义的刻度，并将它们链接到 **.MinEU** 与 **.MaxEU** 点域，以创建一个趋势，按工程单位显示整个数据集范围。

理解实时趋势

实时趋势显示当前正在运行的 **InTouch** 应用程序中的数据。实时趋势持续进行更新。实时趋势图绘制最多与四个本地标记或表达式关联的当前数据值。

您可以：

- 创建实时趋势
- 给趋势选择标记
- 指定趋势的时间跨度与更新间隔
- 配置趋势的显示选项

在历史趋势中显示保存的标记值

通过使用以下任何一种 **WindowMaker** 实用程序，都可以创建历史趋势：

- 历史趋势工具
- 历史趋势向导
- 16-Pen Trend（16 笔趋势）向导（可选）

此外，您可以采用 **ActiveFactory** 趋势，以显示保存到 **Wonderware Historian** 数据库中的 **InTouch** 历史趋势数据。

使用历史趋势对象

您可以使用 **WindowMaker** 的“历史趋势”工具来创建和配置趋势。您可以

- 创建历史趋势
- 给趋势选择标记
- 指定趋势的时间跨度与更新间隔
- 配置趋势的显示选项

创建历史趋势

您可以使用“历史趋势”工具在窗口中创建一个趋势对象。第一次创建历史趋势对象时，使用 InTouch 缺省配置设置。

配置历史趋势之后，WindowMaker 使用最新的配置值作为新趋势对象的初始设置。

您可以在窗口边框内绘制任意大小的趋势图。

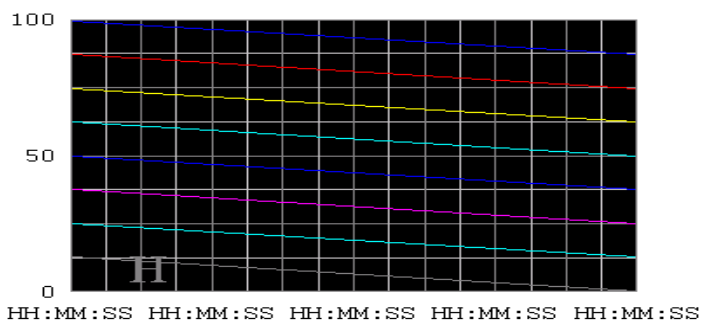
要创建历史趋势

- 1 在 WindowMaker 中，打开要在其中放置一个历史趋势的窗口。



- 2 从“绘图”工具栏中，单击**历史趋势**按钮。

- 3 将鼠标移到要放置历史趋势的窗口区域上。沿着对角线拖动鼠标，根据所需的趋势大小来创建一个长方形。此时“历史趋势”对象出现在窗口中。



- 4 如果需要，使用对象手柄来调整趋势的高度与宽度。

配置要在历史趋势上显示的标记

历史趋势笔创建指定时间段内所记录数据的图形化表示。您将趋势笔指定给采集历史数据的标记。

历史趋势最多支持八支笔。

备注 WindowViewer 必须关闭。否则无法选择“笔”框。

如果配置了远程历史供应器，则可以从这些供应器中选择标记。如需有关设置远程历史供应器的详细信息，请参阅 *InTouch® HMI 应用程序管理与扩展指南* 中的第 2 章“分发应用程序”。

备注 您也可以配置一个 Wonderware Historian 历史供应器，以便使用图形化的方式来显示历史趋势数据。为获得更多的功能性以及其它的绘图选项，请使用 ActiveFactory 趋势工具基于 Wonderware Historian 数据库中保存的 InTouch 历史数据来创建趋势。

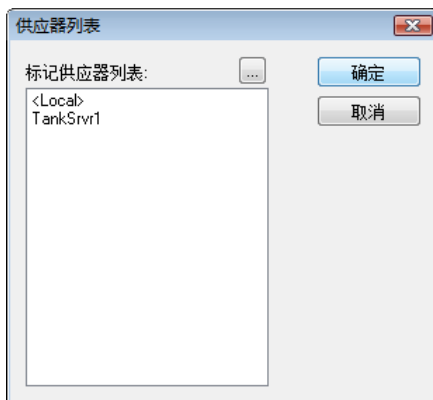
要配置希望在历史趋势上显示的标记

- 1 双击窗口中的趋势对象。此时出现**历史趋势配置**对话框。

- 2 在**历史标记**框中，输入要给趋势使用的标记。
该标记必须定义为“历史趋势”类型。您必须给 InTouch 应用程序中的每个历史趋势指定一个不同的“历史趋势”标记。
- 如果输入的标记当前尚未在“标记名字典”中定义，则出现一个对话框，询问是否要创建标记。如果选择**确定**以定义标记，则出现**标记名字典**对话框。
- 3 在**标记名**区域的一个或多个**笔**框中，指定现有的本地或远程标记的名称。
- 4 要直接指定现有的本地或远程标记，请单击**笔**框并输入标记的名称。

5 要浏览到所要指定的标记：

a 在**笔框**中双击。此时出现**供应器列表**对话框。



b 选择要给笔使用的标记供应器。

c 单击**确定**以显示一个列出所选供应器的标记的对话框。

d 从列表中双击某个标记来选择它。

6 对于指定了标记的每支笔，双击旁边的颜色框以显示调色板。单击要给笔使用的颜色。

7 在**宽度**框中，为趋势中显示的每支笔输入以像素为单位的线条宽度。

8 对于要指定给历史趋势笔的每个标记，重复步骤 3 到 7。

9 如果需要，选择**允许运行时更改**复选框，以允许操作员在应用程序运行期间配置历史趋势。

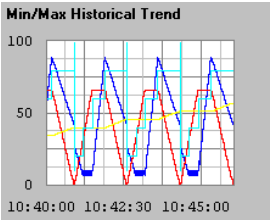
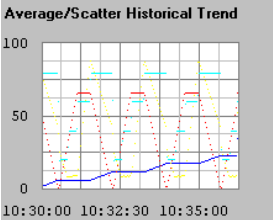
如需有关在运行时更新历史趋势的详细信息，请参阅第 177 页的“在运行时更改趋势配置”。

配置历史趋势的时间跨度

您可以配置历史趋势的时间跨度。

要配置历史趋势的时间跨度

- 1 如果需要，双击趋势对象以显示**历史趋势配置**对话框。
- 2 在**图表时间区域**的**初始时间跨度**中，输入希望在趋势的 x 横坐标轴上出现的时间长度。
- 3 选择时间单位：“秒”、“分”、“小时”或“日”。
例如，如果在**初始时间跨度**中输入 8，然后选择**小时**，则趋势上显示的时间跨度为 8 小时。
- 4 在**初始显示模式**区域中，选择 WindowViewer 最初显示包含趋势的窗口时出现的历史趋势类型。

初始显示模式	描述
最小 / 最大	<p>图表以工程单位的百分比为纵坐标，以时间宽度为横坐标来显示所发生的变化。</p> 
平均	<p>趋势时间段内的每个像素都显示该时间段内标记的平均值。</p> 

- 5 转至第 176 页的“配置历史趋势显示选项”，以配置历史趋势的视觉外观。

配置历史趋势显示选项

您可以配置历史趋势的视觉外观。

要配置历史趋势显示选项

- 1 如果需要，双击趋势对象。此时出现**历史趋势配置**对话框。
- 2 设置颜色选项。执行以下操作：
 - 在**颜色**区域中，单击**图表颜色**框以打开调色板。
 - 单击调色板中的某种颜色，以便将它用作趋势的背景色。
白色是缺省的背景色。任何其它背景色都会增加打印趋势所需的时间。
 - 选择**边框颜色**以打开调色板。
 - 单击调色板中的某种颜色，以便将它用作趋势的边框颜色。
- 3 设置时间刻度选项。执行以下操作：
 - 在**时间刻度**区域的主**刻度数**中，输入趋势的主时间刻度数。
此时主时间刻度出现在趋势的时间横坐标轴上。主时间刻度之间的最大时间间隔是 65536 秒，即 18 小时 12 分 16 秒。
 - 选择主刻度线的颜色。
 - 在**副 / 主刻度**中，输入每个主时间刻度内的副时间刻度数。
副时间刻度数应该是主刻度的偶数倍。例如，如果主刻度设置为 60 秒，则在**副 / 主刻度**中输入值 2 时，会将副时间刻度设置为 30 秒。
 - 选择副刻度线的颜色。
 - 选择**顶部标签**或**底部标签**，以便指定时间标签在趋势上的位置。
 - 如果使用时间标签，请在**主刻度 / 时间标签**框中输入每个时间标签中的主时间刻度线数。
 - 选择时间刻度标签的颜色。
 - 选择显示为主时间刻度标签的时间单位。

月 (MM)	时 (HH)
天 (DD)	分 (MM)
年 (YY)	秒 (SS)
- 4 在**值刻度**区域中，配置趋势纵坐标轴的外观。
值刻度选项的配置方法与**时间刻度**选项的方法相同。纵坐标轴根据所有标记的工程单位来指定趋势中出现的数据值范围。
- 5 单击**确定**以保存对配置所作的更改，然后关闭**历史趋势配置**对话框。

在运行时更改趋势配置

如果在配置历史趋势时选择**允许运行时更改**选项，则操作员可以在运行期间配置历史趋势。操作员使用一个对话框来配置趋势，该对话框在从显示的窗口中选择趋势之后出现。

要在运行时配置历史趋势

- 1 在运行期间单击历史趋势。此时出现**历史趋势设置**对话框。



历史趋势设置对话框包含以下配置项：

- 图表开始**：月(M) 日(D) 年(Y) 时(H) 分(M) 秒(S)，当前显示为 10 / 15 / 97 17 : 48 : 00。
- 显示模式**：
 - ☒ 最小/最大(M)
 - ☐ 平均/散点(V)
 - ☐ 均值/棒图(T)
- 图表长度(L)**：3，单位选择为天(A)、小时(H)、分(I)、秒(C)，当前选中分(I)。
- 图表范围(E)**：最小: 0 % 最大: 100 %。
- 标记**：
 - 笔 #1 ... ProdLevel
 - 笔 #2 ... ReactLevel
 - 笔 #3 ... ReactTemp
 - 笔 #4 ... SetPoint
 - 笔 #5 ... 未指定的 ...
 - 笔 #6 ... 未指定的 ...
 - 笔 #7 ... 未指定的 ...
 - 笔 #8 ... 未指定的 ...

对话框右侧有 **确定**、**取消** 和 **打印(P)** 按钮。

- 2 在**图表开始**区域中，输入历史趋势数据采集间隔的开始日期与时间。
- 3 在**显示模式**区域中，选择历史趋势图表的类型。
趋势显示模式对性能有一定的影响。决定趋势性能的主要因素是趋势中显示的线条的长度。线条越长，生成趋势所需的时间也越长。
线条宽度也会对性能产生影响。绘制宽线条花费的时间要明显长许多。创建“最小/最大”或“平均/散点”趋势比创建“均值/棒图”要更为快速。
- 4 在**图表长度**区域中，输入要在趋势上显示的时段，然后选择度量单位。
例如，如果输入 2 并选择**小时**，则趋势持续时间为 2 小时。
- 5 在**图表范围**区域中，输入历史趋势纵坐标轴上显示的工程单位范围的百分比。
趋势的缩放范围是使用某个百分比范围定义的要绘制趋势的标记的一段工程单位范围。值的范围是从 0 到 100。例如，对于所选的标记，如果要绘制它在 40% 到 60% 工程单位范围内的变化趋势，请分别在**最小**与**最大**范围框中输入 40 和 60。
- 6 在**标记**区域中，单击笔号以指定标记。
此时出现**选择标记**对话框，显示可以指定给历史趋势笔的一系列标记。

- 7 在 QuickScript 或按钮中使用以下语句，以便操作员可以更新图表：

```
Hist_TrendTag.UpdateTrend = 1;
```

- 8 在 QuickScript 中或按钮上使用以下任何一个函数：

```
HTUpdateToCurrentTime(Hist_Tag);
```

```
HTScrollLeft(Hist_Tag,Percent);
```

```
HTScrollRight(Hist_Tag,Percent);
```

```
HTZoomIn(Hist_Tag,LockString);
```

```
HTZoomOut(Hist_Tag,LockString);
```

```
HTSetPenName(Hist_Tag,PenNum,Tagname);
```

如需有关使用包含趋势函数的脚本的详细信息，请参阅第 198 页的“使用脚本控制历史趋势向导”。

- 9 更改以下任何一个趋势标记点域：

```
.ChartStart
```

```
.ChartLength
```

```
.MaxRange
```

```
.MinRange
```

```
.Pen1-.Pen8
```

如需有关给历史趋势使用点域的详细信息，请参阅第 179 页的“使用点域控制历史趋势”。

使用点域控制历史趋势

在运行时，可以使用点域来管理历史趋势。

.DisplayMode 点域

.DisplayMode 点域指定用于显示标记值的趋势格式。

类别

历史

用法

```
tag_name.DisplayMode
```

参数

tag_name

任何“历史趋势”标记。

数据类型

模拟（可读写）。

有效值

1 = 显示每个采样周期中发生的最小 / 最大值（缺省）。

2 = 在散点历史趋势中显示每个采样周期的平均值。

3 = 在棒图历史趋势中显示每个采样周期的平均值。

示例

此语句指定历史趋势中的值由 "HistTrend_Tag" 表示，且设置为条形图历史趋势的格式。

```
HistTrend_Tag.DisplayMode=3;
```

另请参阅

.ChartLength, .ChartStart

.MinRange 点域

.MinRange 点域指定给“历史”趋势中的每个标记显示的标记工程单位范围的最小百分比。

类别

历史。

用法

`tag_name.MinRange`

参数

tag_name

任何“历史趋势”标记。

附注

历史趋势可以同时显示多个类型的标记。由于不同类型的标记有不同的工程范围，要按工程单位来指定值范围的最小与最大边界非常困难。因此，最小与最大范围值使用每个标记工程范围的百分比来表示。这样，无论标记的真正工程范围如何，历史趋势都显示该标记的指定百分比的特定工程范围。

数据类型

实型（可读写）。

有效值

.MaxRange 与 .MinRange 点域的极限介于 0 到 100 之间。

.MinRange 总是小于 .MaxRange。如果将小于 0 或大于 100 的值指定给这两个点域中的任何一个，则该值将调整为 0 或 100。如果 .MinRange 大于或等于 .MaxRange，则趋势不显示任何数据。

示例

此示例点域语句将历史趋势的最小百分比范围设置为 HistTrend 标记可能的工程单位范围的 25%。

```
HistTrend.MinRange=25
```

另请参阅

.ChartStart, .ChartLength, .DisplayMode, .EngUnits, .MinEU, .MaxEU, .MaxRange, .MinRaw, .MaxRaw, .RawValue

.MaxRange 点域

.MaxRange 点域指定给“历史”趋势中的每个标记显示的工程单位范围的最大百分比。

类别

历史。

用法

tag_name.MaxRange

参数

tag_name

任何“历史趋势”标记。

附注

历史趋势可以同时显示许多种类型的标记。由于标记可以有不同的工程范围，要按工程单位来指定值范围的最小与最大边界非常困难。因此，最小与最大范围值使用每个标记工程范围的百分比来表示。这样，无论标记的真正工程范围如何，历史趋势都显示该标记的指定百分比的工程范围。

数据类型

实型（可读写）。

有效值

.MaxRange 与 .MinRange 点域的极限介于 0 到 100 之间。

.MinRange 总是小于 .MaxRange。如果将小于 0 或大于 100 的值指定给这两个点域中的任何一个，则该值将调整为 0 或 100。如果 .MinRange 大于或等于 .MaxRange，则趋势不显示任何数据。

示例

此示例点域语句将历史趋势的最大范围设置为 Hist Trend 标记可能的工程单位范围的 75%。

```
HistTrend.MaxRange=75
```

另请参阅

.ChartStart, .ChartLength, .DisplayMode, .EngUnits,
.MinEU, .MaxEU, .MinRange, .MinRaw, .MaxRaw,
.RawValue

.UpdateCount 点域

每次历史趋势更新时，.UpdateCount 点域将计数值递增一次。
.UpdateCount 点域可以用作其它函数的触发器。

类别

历史。

用法

HistTrendTag.UpdateCount

参数

HistTrendTag

指定了趋势的名称的“历史趋势”标记。

数据类型

整型（只读）。

有效值

任何正整数。

示例

本例使用 HTGetValueAtScooter() 函数来检索右指示器当前位置上 Pen1 的值。对任何函数参数所作的更改都会导致对该函数进行重新求值。完成更新时，.UpdateCount 的值会递增，此语句将重新求值。

```
MyRealTag=HTGetValueAtScooter(  
    MyHistTrendTag,MyHistTrendTag.UpdateCount, 2,  
    MyHistTrendTag.ScooterPosRight, 1, "PenValue");
```

另请参阅

.UpdateInProgress, .UpdateTrend

.UpdateInProgress 点域

.UpdateInProgress 点域指出历史趋势更新操作的当前状态。如果正在检索历史，则此点域的值设置为 1；否则此点域设置为 0。

类别

历史。

用法

HistTrendTag.UpdateInProgress

参数

HistTrendTag

指定了趋势的名称的“历史趋势”标记。

附注

只要历史趋势要求提供新数据，此点域的值便设置为 1。在过程结束之后，.UpdateInProgress 重置为 0。.UpdateInProgress 可以用于同历史趋势有关的函数。

如果操作员将趋势滚动到当前显示的时段以外的区域，则检索历史数据需要花费一些时间。.UpdateInProgress 点域提供了一种方法来提示操作员目前正在检索要求提供的数据。如果没有这个反馈信息，操作员可能不会了解到目前正在更新趋势。

数据类型

离散（只读）。

有效值

0 = 未在更新

1 = 正在更新

示例

.UpdateInProgress 点域通常用作“历史趋势”滚动按钮上的或旁边的文本对象上的可视化链接中的表达式。通过以下消息值显示动画链接，可以使用 .UpdateInProgress 点域在检索数据时将 "Busy" 显示到窗口上。

```
DText(HistTrend1.UpdateInProgress,"Busy","Ready")
```

另请参阅

.UpdateCount, .UpdateTrend

.UpdateTrend 点域

.UpdateTrend 点域触发对历史趋势的更新。在按钮动作脚本中使用 .UpdateTrend 点域时，操作员可以在运行时手工更新趋势。

类别

历史。

用法

HistTrendTag.UpdateTrend

参数

HistTrendTag

指定了趋势的名称的“历史趋势”标记。

附注

历史趋势不自动更新。要更新图表并显示指定的标记的当前值，必须更改 .ChartStart 或 .ChartLength 点域之一。通过在按钮动作脚本中使用此点域，操作员可以在运行时更新图表。如果要更改与历史趋势关联的其它点域，则也可以在 QuickScript 中使用此点域。

您只能将 .UpdateTrend 点域设置为 1 这个值。

数据类型

离散（只写）。

有效值

1

示例

本例触发与 MyHistTrendTag 标记关联的历史趋势使用所有参数的当前值进行更新。

```
MyHistTrendTag.UpdateTrend=1;
```


.ChartLength 点域

.ChartLength 点域指定“历史”趋势中显示的时间长度。

类别

历史。

用法

HistTrendTag.ChartLength

参数

HistTrendTag

指定了趋势的名称的“历史趋势”标记。

附注

指定给 .ChartLength 的值以秒为单位指定图表的长度。长度定义为“历史趋势图表”上当前显示的时间量。更具体一些说，从“历史趋势图表”检索“图表长度”的计算公式是：

$$\text{ChartLength} = (\text{图表右侧的日期 / 时间标签}) - (\text{图表左侧的日期 / 时间标签});$$

由于“日期/时间标签”使用自 1970 年 1 月 1 日午夜以来的秒数来表示，因此计算结果为图表左右两侧之间显示的时间秒数之差。

对 .ChartLength 进行加减运算时，都使用秒来表示时间。因此，要从当前的 .ChartLength 中减去两小时，则必须在执行计算之前先将小时转换为秒。例如：

$$(2 \text{ 小时}) * (60 \text{ 分钟} / \text{小时}) * (60 \text{ 秒} / \text{分钟}) = 7200 \text{ 秒}。$$

数据类型

整型（可读写）。

有效值

任何正整数。

示例

本例将历史趋势的长度强制设置为 1 小时。

```
HtTag.ChartLength=3600 {60 minutes * 60
seconds/minute};
```

本例将趋势向左滚动 50%。

```
HtTag.ChartStart=HtTag.ChartStart -
HtTag.ChartLength / 2;
```

本例将图表向左滚动 10%。

```
HtTag.ChartStart=HtTag.ChartStart - (.10 *
HtTag.ChartLength);
```

另请参阅

.ChartStart

.ChartStart 点域

.ChartStart 点域可以用于设置或验证历史趋势的开始（左侧）日期 / 时间标签的值。

类别

历史。

用法

HistTrendTag.ChartStart

参数

HistTrendTag

指定了趋势的名称的“历史趋势”标记。

附注

这个可读写点域用于设置或验证历史趋势的开始日期 / 时间标签的值。 .ChartStart 点域使用自 1970 年 1 月 1 日午夜以来所经过的秒数来表示。起始点定义为历史趋势上的第一个日期 / 时间标签。

数据类型

整型（可读写）。

有效值

任何正整数。

示例

以下语句使图表向右滚动一分钟。

```
HtTagname.ChartStart=HtTagname.ChartStart + 60;
```

另请参阅

.ChartLength

.Pen1-8 点域

.Pen1-8 点域将记录的标记指定给某个历史趋势笔。

类别

历史

用法

```
HistTrendTag{.Pen1 | .Pen2 | .Pen3 | .Pen4 | .Pen5  
| .Pen6 | .Pen7 | .Pen8};
```

参数

HistTrendTag

指定了趋势的名称的“历史趋势”标记。

附注

使用 .Pen1-8 点域采用以下格式将标记指定给趋势笔：

HistTrend.PenX = Tag_Name.TagID

其中 X 是 1 到 8 之间的整数。

建议尽量使用 HTSetPenName() 与 HTGetPenName() 函数。

备注 只有本地标记可以指定给 .PenX 点域。无法使用 provider.tag 表示法。provider.tag 仅可以用于 HTSetPenName() 函数。

了解这些点域的工作原理的一个良好的参考是，放置到屏幕上并进行了分解的历史趋势向导。

数据类型

TagID（可读写）。

有效值

点域数据类型是 TagID 类型。这就意味着，只有标记的句柄可以指定给 .Pen1-8 点域。您无法将标记的名称直接指定给 .Pen1-8 点域。您必须使用以下语法将与标记关联的 .TagID 点域关联到 Pen1-8 点域：

```
HistTrendTag.Pen1=LoggedTag.TagID;
```

一般而言，TagID 类型标记仅可赋给另一个 TagID 标记。它无法与任何其它标记类型配合使用，除非其它标记中添加了 .TagID 点域扩展名。

尽管 .Pen1-8 点域被视为可读写，但无法直接在屏幕上显示它们的值。

示例

下例将新标记指定给与“历史趋势”标记关联的历史趋势的 .Pen5 点域。要将所记录的标记指定给 .Pen5 点域，必须将 .TagID 点域附加到所记录的标记的名称中。

```
HistTrendTag.Pen5=PumpPress.TagID;
```

从上例开始，可以显示指定给 `HistTrendTag.Pen5` 的标记的名称。对于操作员来说，创建一个图例来显示指定给每个趋势笔的标记，这是非常有用的信息。

您无法在“消息显示”链接中显示指定给 `HistTrendTag.Pen5` 的标记。`.Pen5` 点域的实际值是一个整数，表示 `WindowViewer` 中的内存位置，这对于显示目的而言毫无意义。您需要创建一个名称为 `Pen05` 的 `TagID` 型标记。将以下语句放到上例中的语句之下：

```
Pen05=HistTrendTag.Pen5;
```

在第一个示例中，`PumpPress` 标记指定给 `HistTrendTag` 的 5 号笔。在本例中，`Pen05` 被指定 `HistTrendTag` 的 `Pen5` 的值，而 `HistTrendTag` 则是 `PumpPress` 标记的 `TagID`。

`.Pen1-8` 这些点域是一些指针，它们指向与选择要在趋势中显示的笔关联的标记。`.Pen1-8` 这些点域是一种特殊的数据类型，即 `.TagID`。执行指定之后，您可以使用 `TagID` 标记的 `.Name` 点域显示标记的名称。

.TagID 点域

`.TagID` 点域可以同 `.Pen1 - .Pen8` 点域配合使用，以便将标记指定给历史趋势笔。

类别

历史标记。

用法

```
tag_name.TagID
```

参数

tag_name

任何离散、整型、实型、间接离散或间接模拟标记。

附注

`.TagID` 点域提供标记句柄并主要用于将标记指定给历史趋势中的笔。

数据类型

`TagID`（只读）。

示例

本例使用 `.TagID` 点域将 `PumpRPM` 标记指定给历史趋势的 6 号笔。

```
HistTrendTag.Pen6=PumpRPM.TagID;
```

另请参阅

`.Pen1-.Pen8`

.ScooterLockLeft 点域

.ScooterLockLeft 点域指定操作员是否可以将右指示器进一步移到历史趋势上左指示器当前位置的左侧。

类别

历史。

用法

```
HistTrendTag.ScooterLockLeft
```

参数

HistTrendTag

指定了趋势的名称的“历史趋势”标记。

附注

一般而言，应该阻止操作员将右指示器进一步移到左指示器当前位置的左侧。左指示器锁定时，每次将右指示器移到它的左侧时，它便迫使右指示器处在与它相同的位置。

数据类型

离散（可读写）。

有效值

0 = False。右指示器可以进一步移到历史趋势上左指示器当前位置的左侧

1 = True。右指示器无法移到历史趋势上左指示器当前位置的左侧。

示例

下例阻止将右指示器移到历史趋势上左指示器当前位置的左侧。

```
HistTrendTag.ScooterLockLeft=1;
```

另请参阅

.ScooterPosRight、.ScooterPosLeft、.ScooterLockRight

.ScooterLockRight 点域

.ScooterLockRight 点域指定操作员是否可以将左指示器进一步移到历史趋势上右指示器当前位置的右侧。

类别

历史。

用法

HistTrendTag.ScooterLockRight

参数

HistTrendTag

指定了趋势的名称的“历史趋势”标记。

附注

一般而言，应该阻止操作员将左指示器进一步移到右指示器当前位置的右侧。右指示器锁定时，每次左指示器移到它的右侧时，它便迫使左指示器处在与它相同的位置。

数据类型

离散（可读写）。

有效值

0 = False。左指示器可以移到历史趋势上右指示器当前位置的右侧。

1 = True。左指示器无法移到历史趋势上右指示器当前位置的右侧。

示例

下例阻止将左指示器移到历史趋势上右指示器当前位置的右侧。

```
HistTrendTag.ScooterLockRight=1;
```

另请参阅

.ScooterPosRight、.ScooterPosLeft、.ScooterLockLeft

.ScooterPosLeft 点域

.ScooterPosLeft 点域动态控制历史趋势上左指示器的位置。

类别

历史

用法

HistTrendTag.ScooterPosLeft

参数

HistTrendTag

指定了趋势的名称的“历史趋势”标记。

附注

这个可读写点域动态控制左指示器的位置。您可以在 **QuickScript** 函数中使用此点域来检索左指示器的当前位置；或者将一个值指定给此点域，以便将左指示器调整到趋势上的另一个位置。

此点域最常与一组 **HTGetValue()** 函数配合使用。这些函数必须指定要查询哪个历史趋势以及趋势指示器的当前位置。

数据类型

实型（可读写）。

有效值

0.0 到 1.0；其中 0.0 是历史趋势图表的最左侧，1.0 是历史趋势图表的最右侧。

示例

下例重新调整左指示器的位置。左指示器移到距离历史趋势图表（当前与 **MyHistTrendTag** 标记关联的）左侧 34% 图表总长的位置。

```
MyHistTrendTag.ScooterPosLeft=.34;
```

在以下语句中，**QuickScript** 函数 **HTGetValueAtScooter()** 检索左指示器当前位置上 1 号笔的值。对函数参数列表中的任何值所作的更改都会导致对函数重新进行求值。每次左指示器的位置发生改变时，此语句都会重新求值。

```
MyRealTag=HTGetValueAtScooter  
  (MyHistTrendTag,MyHistTrendTag.UpdateCount, 1,  
   MyHistTrendTag.ScooterPosLeft, 1, "PenValue");
```

另请参阅

.ScooterPosRight、.ScooterLockLeft、.ScooterLockRight

.ScooterPosRight 点域

可读写的 .ScooterPosRight 点域动态控制右指示器的位置。

类别

历史。

用法

HistTrendTag.ScooterPosRight

参数

HistTrendTag

指定了趋势的名称的“历史趋势”标记。

附注

这个可读写点域动态控制右指示器的位置。您可以在 **QuickScript** 函数中使用此点域来检索右指示器的当前位置。您也可以将某个值指定给此点域，以便将右指示器移到趋势上的另一个位置。

此点域最常与 **HTGetValue()** 函数配合使用。这些函数必须指定要查询哪个历史趋势以及趋势指示器的当前位置。

数据类型

实型（可读写）

有效值

0.0 到 1.0；其中 0.0 是历史趋势图表的最左侧，1.0 是历史趋势图表的最右侧。

示例

以下语句给右指示器指定一个新的位置。右指示器移到距离历史趋势图表（与 **MyHistTrendTag** 标记关联的）左侧 34% 图表总长的位置。

```
MyHistTrendTag.ScooterPosRight=.34;
```

以下语句使用 **QuickScript** 函数 **HTGetValueAtScooter()** 检索右指示器新位置上 1 号笔的值。对函数参数列表中的任何变量所作的更改都会导致对函数重新进行求值。每次右指示器的位置发生改变时，此语句都会重新求值。

```
MyRealTag=HTGetValueAtScooter(MyHistTrendTag,  
    MyHistTrendTag.UpdateCount, 2,  
    MyHistTrendTag.ScooterPosRight, 1, "PenValue");
```

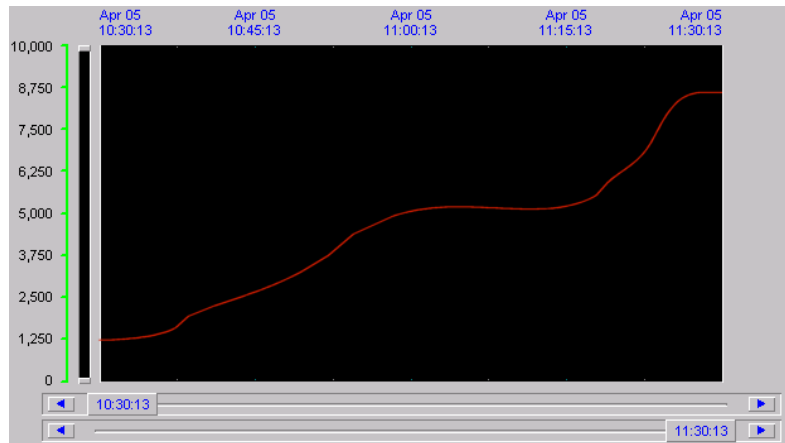
另请参阅

.ScooterPosLeft,. ScooterLockLeft, .ScooterLockRight

使用历史趋势向导

“历史趋势向导”自动创建一个历史趋势。除了手工将标记指定给历史趋势笔，此向导还使用一些标准的值来自动配置历史趋势。

下图显示使用“历史趋势向导”创建的标准趋势。该趋势图包含被称作指示器的游标，以显示趋势图中特定位置上的数据，或放大到所选的趋势数据范围。



要将缩放与移动功能或一些笔控件添加到历史趋势中，请使用趋势“缩放 / 平移”面板以及“趋势笔图例”向导。

您可以创建与配置历史趋势。您可以：

- 使用一些向导来创建历史趋势
- 给趋势选择标记
- 配置历史趋势时间跨度
- 使用 QuickScript 控制趋势

使用历史趋势向导创建趋势

您可以使用“历史趋势向导”的自动化功能来创建标准的历史趋势。其它部分将介绍如何使用向导选项来手工配置历史趋势。

要使用向导创建历史趋势

1 从 WindowMaker 打开一个要放置历史趋势的窗口。



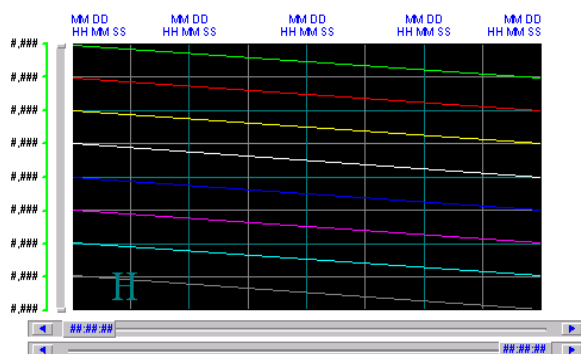
2 在 WindowMaker 菜单栏上，单击**向导**图标。此时出现**向导选择**对话框。

3 从向导列表中选择 **Trends**（趋势）。此时**向导选择**对话框的右侧面板中显示一组趋势向导图标。



4 选择**含指示器和刻度的历史趋势**向导，然后单击**确定**。此时**向导选择**对话框关闭，您的窗口再次出现。

5 将光标移到窗口上希望放置历史趋势左上角的位置。通过单击将趋势放置到窗口中。



6 双击该趋势。此时出现**历史趋势图表向导**对话框。

7 单击**建议**。此时“历史趋势图表向导”自动将一些缺省配置值指定给该趋势。

剩余的唯一配置任务就是将标记指定给趋势笔。

配置要在趋势图上显示的标记

在“历史趋势图表向导”中将标记指定给趋势笔同“历史”与“实时”工具中的过程类似。

要从“历史趋势图表向导”中指定标记

- 1 如果需要，双击历史趋势。此时出现**历史趋势图表向导**对话框。
- 2 单击**笔**。此时出现**趋势笔**对话框。



- 3 在**笔**框中输入现有的本地标记的名称。

备注 WindowViewer 必须关闭。否则无法选择“笔”框。

如果在**笔**框中双击，则出现**选择标记**对话框，显示应用程序中指定了**记录数据**选项的一系列标记。通过从**选择标记**对话框中选择标记，可以将它指定给笔。

- 4 如果希望更改笔的缺省颜色，请单击每支笔旁边的颜色框，然后选择另一种颜色。否则跳过此步骤并接受缺省颜色。
- 5 单击“确定”以关闭**趋势笔**对话框。
- 6 单击**确定**以关闭**历史趋势图表向导**对话框。

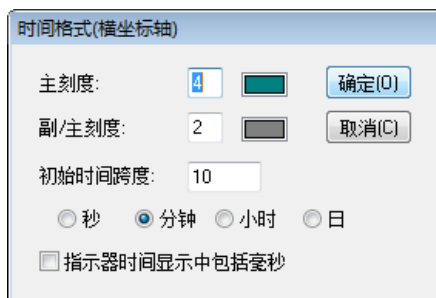
配置历史趋势时间跨度

历史趋势图表向导对话框包含一个选项，可用于手工配置使用“历史趋势向导”创建的趋势中所显示的时间跨度。您可以手工配置趋势的时间，而不是接受“历史趋势向导”的缺省配置。

要配置历史趋势的时间跨度

- 1 双击历史趋势。此时出现**历史趋势图表向导**对话框。

- 2 单击**时间**。此时出现**时间格式**对话框。



- 3 配置时间格式。执行以下操作：

- a 在**主刻度**框中，输入趋势的时间横坐标轴上显示的主时间刻度数。
- b 在**副 / 主刻度**框中，输入每个主刻度内的副时间刻度数。
- c 在**初始时间跨度**框中，输入在趋势的横坐标轴上显示的时间段的长度。应用程序在 **WindowViewer** 中运行时，使用“历史趋势向导”创建的趋势可以进行更新。操作员可以更改趋势时间段的长度。但历史趋势总是从**时间格式**对话框中设置的时间段开始。
- d 选择趋势时间段的度量单位：秒、分钟、小时、日。
- e 作为可选项，可以在显示的指示器时间中包含毫秒下例显示的指示器游标中包含附加到当前时间的毫秒。。

11:30:13.000

- 4 单击**确定**以关闭**时间格式**对话框。
- 5 单击**确定**以关闭**历史趋势图表向导**对话框。

配置显示选项

历史趋势图表向导对话框包含一个可配置水平趋势垂直单位的选项。您可以手工配置趋势纵坐标轴上显示的主、副数值刻度。

要使用“历史趋势图表向导”配置显示选项

- 1 如果需要，双击历史趋势。此时出现**历史趋势图表向导**对话框。
- 2 单击**值**。此时出现**值格式**对话框，其中包含一些可配置趋势数值纵坐标轴的选项。



- 3 配置值格式。执行以下操作：
 - a 在**主刻度**框中，输入趋势的纵坐标轴上显示的主数值刻度数。单击颜色框以访问调色板，然后单击要指定给数值坐标轴主刻度线的颜色。
 - b 在**副 / 主刻度**框中，输入每个数值坐标轴主刻度内可见的副刻度数。单击颜色框以访问调色板，然后单击要指定给数值坐标轴副刻度线的颜色。
 - c 在**值格式**框中，输入趋势的数值纵坐标轴中出现的数值格式。缺省数字格式是 **#,###**。
 - d 在**趋势范围**框中，输入趋势中出现的标记工程单位的上、下百分比边界。
- 4 单击**确定**以关闭**值格式**对话框。
- 5 单击**确定**以关闭**历史趋势图表向导**对话框。

在运行时更改配置

在配置历史趋势时，如果选择了**允许运行时更改**选项，则操作员可以在运行时对历史趋势的某些方面进行更改。

要在运行时配置历史趋势

- 1 单击 WindowViewer 中的趋势。此时出现**历史趋势设置**对话框。



- 2 在**图表开始**区域中，输入图表的开始日期与时间。
- 3 在**显示模式**区域中，选择历史趋势的类型。
- 4 在**图表长度**区域中，输入趋势上显示的时间长度，然后选择该长度的时间增量。

- 5 在**图表范围**区域中，输入作为趋势垂直范围的工程单位范围百分比。
- 6 在**标记**区域中，单击每个**笔 #** 以便将标记指定给趋势笔。此时出现**选择标记**对话框，并显示那些启用了记录功能的标记。
- 7 双击标记的名称将它指定给趋势笔。
- 8 单击**确定**以保存在运行时对趋势所作的更改。

使用脚本控制历史趋势向导

您可以给趋势对象或动画链接表达式使用 **QuickScript** 函数，以便在运行时控制历史趋势。例如，您可以使用 **QuickScript** 将趋势更新到当前时间、将标记指定给趋势笔、将笔连接到图表、重新绘制网格，以及删除或重新绘制指示器。

将趋势更新到当前时间

您可以创建脚本来更新历史趋势以显示最新标记数据。

HTUpdateToCurrentTime() 函数

HTUpdateToCurrentTime() 函数检索并显示结束时间等于当前时间的数据。开始时间等于结束时间减去图表的宽度。

类别

历史

语法

```
HTUpdateToCurrentTime(Hist_Tag);
```

参数

Hist_Tag

指定了历史趋势的名称的历史趋势标记

示例

以下语句检索并显示当前时间 **Trend1** 历史标记的数据：

```
HTUpdateToCurrentTime("Trend1");
```

如果当前时间是下午 3:04，趋势的宽度是 60 秒，则新的结束时间是下午 3:04。新的趋势开始时间是下午 3:03。

更改趋势配置

您可以使用以下这些脚本函数来更改指定给历史趋势笔的标记：

- HTSelectTag() 函数
- HTSetPenName() 函数

HTSelectTag() 函数

HTSelectTag() 函数打开**选择标记**对话框，以便操作员将不同的标记指定给趋势笔。

备注 **选择标记**对话框仅列出定义了历史记录功能（在“标记名字典”中选择了**记录数据**选项）的那些标记。

类别

历史

语法

```
HTSelectTag();
```

附注

HTSelectTag() 函数仅显示从“标记名字典”中选择了**记录数据**选项的那些标记。不过也可以使用“标记浏览器”的过滤器来显示较少的一组标记。例如，所有以“A”开头的标记。此函数返回所选的标记，并可以用作给笔指定标记的函数参数。

示例

以下 QuickScript 导致在 WindowViewer 中出现**选择标记**对话框。然后用户可以从列表中选择标记。此标记由名称为 HistTrend 的“历史对象”指定给 1 号笔。

```
HTSetPenName("HistTrend",1,HTSelectTag());
```

另请参阅

HTSetPenName()

HTSetPenName() 函数

HTSetPenName() 函数将一个不同的标记指定给趋势的某支笔。

类别

历史

语法

```
HTSetPenName (Hist_Tag, PenNum, Tagname);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

PenNum

代表趋势的笔号（从 1 到 8）的整型标记或值。

Tagname

指定给笔的新标记的名称。

附注

此 QuickScript 函数是在运行时从分布式历史供应器添加标记的唯一方法。

试图取消指定趋势笔时，可能会看到以下错误消息：

```
VIEW /UpdateData: Invalid DBS.TAGNAME handle - 0
```

如果试图取消指定先前按照 *histprovider.tag_name* 的形式指定给某个远程标记的笔，则会发生此错误。要解决此错误，请创建一个选择了**记录数据**选项的本地标记。然后使用以下脚本来取消指定这支笔：

```
HTSetPenName ( "HistTrend", 1, "localtag" );
{assigns the pen to a locally logged
 tag---localtag}

HistTrend.Pen1=None;
{unassigns the pen}
```

其中，None 是 TagID 类型标记。

示例

以下语句将 OutletPressure 标记指定给 Trend1 的 3 号笔。

```
HTSetPenName ("Trend1", 3, "OutletPressure");
```

以下语句将 HistPrv1.Tag1 标记指定给 Trend1 的 TrendPen4。

```
HTSetPenName ("Trend1", TrendPen4, "HistPrv1.Tag1");
```

另请参阅

HTSelectTag()

检索趋势与历史数据的有关信息

您可以创建一些脚本在运行从历史趋势中检索信息。使用以下函数：

- HTGetPenName() 函数
- HTGetTimeAtScooter() 函数
- HTGetTimeStringAtScooter() 函数
- HTGetValue() 函数
- HTGetValueAtScooter() 函数
- HTGetValueAtZone() 函数
- HTScrollLeft() 函数
- HTScrollRight() 函数
- HTZoomIn() 函数
- HTZoomOut() 函数

HTGetPenName() 函数

HTGetPenName() 函数返回当前指定给历史趋势的指定笔号的标记的名称。

类别

历史

语法

```
MessageResult=HTGetPenName (Hist_Tag, UpdateCount,  
PenNum) ;
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

UpdateCount

代表趋势的 .UpdateCount 点域的整数。此参数值起着数据改变触发器的作用，可以重新对函数求值。

PenNum

代表趋势的笔号（从 1 到 8）的整型标记或值。

示例

以下语句检索指定给 Trend1 趋势的 2 号笔的标记的名称，然后将该名称放入 TrendPen 消息标记中：

```
TrendPen=HTGetPenName ("Trend1",  
Trend1.UpdateCount, 2) ;
```

HTGetTimeAtScooter() 函数

HTGetTimeAtScooter() 返回某个样本时间（使用自 GMT 时间 1970 年 1 月 1 日 00:00:00 以来经过的秒数来表示），该样本在 *ScootNum* 与 *ScootLoc* 参数所指定的指示器位置上。

类别

历史

语法

```
IntegerResult=HTGetTimeAtScooter(Hist_Tag,  
UpdateCount, ScootNum, ScootLoc);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

UpdateCount

代表趋势的 .UpdateCount 点域的整数。

ScootNum

代表左指示器或右指示器的整数。

1 = 左指示器

2 = 右指示器

ScootLoc

代表趋势中 .ScooterPosRight 或 .ScooterPosLeft 位置上的值的实数。

附注

指定给 UpdateCount、ScootNum、及 ScootLoc 参数的值所发生的任何变化都会导致对该表达式重新求值。这确保了在检索新的数据或移动指示器之后会对表达式重新进行求值。

示例

以下语句检索 Trend1 趋势的当前左指示器位置上的值的时间（以秒为单位）：

```
HTGetTimeAtScooter("Trend1",Trend1.UpdateCount,1,  
Trend1.ScooterPosLeft);
```

HTGetTimeStringAtScooter() 函数

HTGetTimeStringAtScooter() 函数为指定的指示器位置上的样本返回一个包含日期 / 时间的字符串。

类别

历史

语法

```
MessageResult=HTGetTimeStringAtScooter(Hist_Tag,
    UpdateCount, ScootNum, ScootLoc, Format_Text);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

UpdateCount

代表趋势的 .UpdateCount 点域的整数。

ScootNum

代表左指示器或右指示器的整数。

1 = 左指示器

2 = 右指示器

ScootLoc

代表趋势中 .ScooterPosRight 或 .ScooterPosLeft 位置上的值的实数。

Format_Text

指定要使用的日期 / 时间格式的字符串。以下是可以接受的

Format_Text 字符串：

"Date"、"Time"、"DateTime"、"DOWShort"（例如 Wed）
以及 "DOWLong"（例如 Wednesday）。

附注

指定给 UpdateCount、ScootNum、及 ScootLoc 参数的值所发生的任何变化都会导致对该表达式重新求值。这确保了在检索新的数据或移动指示器之后会对表达式重新进行求值。字符串的格式确定了返回值的内容。

示例

以下语句检索 Trend1 趋势右指示器当前位置上的值的日期与时间。该值按照 "Time" 格式存储在 NewRightTimeString 消息标记中。

```
NewRightTimeString=HTGetTimeStringAtScooter
    ("Trend1",Trend1.UpdateCount,2,
    Trend1.ScooterPosRight,"Time");
```

HTGetValue() 函数

HTGetValue() 函数为指定的趋势笔返回所请求的类型的值。

类别

历史

语法

```
RealResult=HTGetValue(Hist_Tag,UpdateCount,  
    PenNum,ValType_Text);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

UpdateCount

代表趋势的 .UpdateCount 点域的整数。

PenNum

代表趋势的笔号（从 1 到 8）的整型标记或值。

ValType_Text

代表要返回的值的类型的字符串：

PenAverageValue = 整个趋势的平均值。

PenMaxValue = 整个趋势的笔的最大值。

PenMinValue = 整个趋势的笔的最小值。

PenMaxEU = 整个趋势的最大工程单位值。

PenMinEU = 整个趋势的最小工程单位值。

PenStdDev = 整个趋势的标准偏差。

附注

函数按实数形式返回所请求的值。

示例

以下语句获取从 PumpPress 趋势中检索的 2 号笔数据的标准偏差。该值存储在 LeftHemisphereSD 内存实型标记中：

```
LeftHemisphereSD=HTGetValue("PumpPress",  
    PumpPress.UpdateCount,2,"PenStdDev");
```

HTGetValueAtScooter() 函数

HTGetValueAtScooter() 函数返回指定的指示器位置、趋势以及笔号处的所请求的类型的样本值。UpdateCount 参数在函数处理完毕之后导致对表达式进行求值。

类别

历史

语法

```
RealResult=HTGetValueAtScooter(Hist_Tag,
                                UpdateCount,ScootNum,ScootLoc,PenNum,
                                ValType_Text);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

UpdateCount

代表趋势的 .UpdateCount 点域的整数。

ScootNum

代表左指示器或右指示器的整数。

1 = 左指示器

2 = 右指示器

ScootLoc

代表趋势的 .ScooterPosRight 或 .ScooterPosLeft 点域的实数。

PenNum

代表笔号（从 1 到 8）的整型标记或值。

ValType_Text

代表要返回的值的类型的字符串：

PenValue = 指示器位置处的值。

PenValid = 如果该值无效，则返回 0；否则返回 1。

在 HTGetValueAtScooter() 函数中使用 ValType_Text 参数时，使用列出的有效类型之一。

示例

如果值是一个实际的样本，则以下函数返回 1；如果值对于右指示器当前位置上 Trend1 趋势的 3 号笔而言无效，则以下函数返回 0：

```
HTGetValueAtScooter("Trend1",Trend1.UpdateCount,
                    2,Trend1.ScooterPosRight,3, "PenValid");
```

HTGetValueAtZone() 函数

HTGetValueAtZone() 函数为指定的趋势笔返回左、右指示器位置之间某个数据的所请求的类型的值。

类别

历史

语法

```
RealResult=HTGetValueAtZone(Hist_Tag, UpdateCount,
    Scoot1Loc, Scoot2Loc, PenNum, ValType_Text);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

UpdateCount

代表趋势的 .UpdateCount 点域的整数。它只用作对函数进行求值的触发器。

Scoot1Loc

代表趋势的 .ScooterPosLeft 点域的实数。它只用作对函数进行求值的触发器。

Scoot2Loc

代表趋势的 .ScooterPosRight 点域的实数。它只用作对函数进行求值的触发器。

PenNum

代表趋势的笔号（从 1 到 8）的整型标记或值。

ValType_Text

代表要返回的值的类型的字符串。

PenAverageValue = 指示器之间的区域的平均值。

PenMaxValue = 指示器之间的区域的最大值。

PenMinValue = 指示器之间的区域的最小值。

PenMaxEU = 指示器之间的区域的最大工程单位值。

PenMinValue = 指示器之间的区域的最小工程单位值。

PenStdDev = 指示器之间的区域的标准偏差。

附注

返回代表给定类型计算值的实数。给 Scoot1Loc 和 Scoot2Loc 指定常数值没有任何效果，它们仅用于对函数进行求值的触发器。此函数直接使用趋势标记的 .ScooterPosLeft 和 .ScooterPosRight 点域值，而不论给 Scoot1Loc 与 Scoot2Loc 参数指定的值如何。

示例

下面的语句计算 Trend1 趋势 1 号笔的左、右指示器之间的数据的平均值。值存储在 AvgValue 内存实型标记中：

```
AvgValue=HTGetValueAtZone("Trend1",
    Trend1.UpdateCount,Trend1.ScooterPosLeft,
    Trend1.ScooterPosRight,1,"PenAverageValue");
```

平移与缩放趋势

您可以创建一些 **QuickScript**，在其中包含在运行时从历史趋势选择特定数据的函数。

HTScrollLeft() 函数

HTScrollLeft() 函数将趋势的起始时间值设置为比当前起始时间早趋势总时间跨度的一定百分比。其效果是按照指定的趋势总时间跨度百分比向左滚动图表，以回到较早前的时间。

类别

历史

语法

```
HTScrollLeft(Hist_Tag,Percent);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

Percent

代表要向左滚动的图表时间跨度百分比的实数（0.0 到 100.0）。

示例

以下语句将日期 / 时间向左滚动 **PumpPress** 趋势总宽度的 10%：

```
HTScrollLeft("PumpPress",10.0);
```

如果当前显示从中午 12:00:00 点开始、显示宽度为 60 秒，则新趋势将从上午 11:59:54 开始（在处理函数之后）。

HTScrollRight() 函数

HTScrollRight() 函数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。其效果是按照指定的趋势宽度百分比向右滚动图表的日期 / 时间。

类别

历史

语法

```
HTScrollRight(Hist_Tag, Percent);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

Percent

代表要向右滚动的图表百分比的实数（0.0 到 100.0）。

示例

以下语句将 PumpPress 趋势向右滚动 20%:

```
HTScrollRight("PumpPress", 20.0);
```

如果当前显示从中午 12:00:00 点开始、显示宽度为 60 秒，则新趋势将从下午 12:00:12 开始（在处理函数之后）。

HTZoomIn() 函数

HTZoomIn() 函数计算新的图表宽度和开始时间。如果趋势的指示器在趋势的左侧和右侧，则新图表的宽度将等于旧图表的宽度除以二。新的开始时间基于 *LockString* 参数的值进行计算。

如果指示器不在趋势的左侧和右侧，HTZoomIn() 函数将按指示器定义的区域来缩放趋势，而忽略 *LockString* 参数。

类别

历史

语法

```
HTZoomIn(Hist_Tag, LockString);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

LockString

代表缩放类型的字符串：

StartTime 将开始时间设置为与缩放前相等

Center 将中心时间设置为与缩放前相等

EndTime 将结束时间设置为与缩放前相等

附注

如果指示器位置不在趋势的左侧和右侧，则新图表的宽度将是 .ScooterPosLeft 和 .ScooterPosRight 位置之间的时间。在这种情况下不会使用 LockString 的值。最小图表宽度是一秒。在缩放之后，指示器位置设置为 .ScooterPosLeft=0.0、.ScooterPosRight=1.0。

示例

以下语句将显示画面缩小为原来的二分之一，并将 Trend1 趋势的开始时间保持不变。Trend1.ScooterPosRight 等于 1.0，Trend1.ScooterPosLeft 等于 0.0。如果在缩放前开始时间是下午 1:25:00，并且图表宽度是 30 秒，则新的开始时间仍然保持在 1:25:00。新的图表宽度是 15 秒。

```
HTZoomIn("Trend1", "StartTime");
```

HTZoomOut() 函数

HTZoomOut() 函数计算新的图表宽度和开始时间。新图表的宽度等于旧图表宽度乘以二。新的开始时间基于 *LockString* 参数的值进行计算。

类别

历史

语法

```
HTZoomOut(Hist_Tag, LockString);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

LockString

代表缩放类型的字符串：

StartTime = 将开始时间设置为与缩放前相等

Center = 将中心时间设置为与缩放前相等

EndTime = 将结束时间设置为与缩放前相等

附注

当前的指示器位置对 HTZoomOut() 没有影响。在函数缩放完成之后，新的指示器位置设置为 .ScooterPosLeft=0.0、.ScooterPosRight=1.0。

示例

以下语句将显示画面放大为原来的两倍，并将 Volume 趋势的中心时间保持不变。如果在缩放前开始时间是下午 2:15:00，且图表宽度是 30 秒，则缩放后现在的开始时间是 2:14:45。图表宽度变为 60 秒，趋势的中心仍是 2:15:15。

```
HTZoomOut("Volume", "Center");
```

打印趋势

您可以在脚本中使用 `PrintHT()` 函数打印 WindowViewer 屏幕上当前可见的历史趋势。

PrintHT() 函数

`PrintHT()` 函数打印屏幕上当前可见的历史趋势。通常，`PrintHT()` 函数与历史趋势窗口上包含的某个屏幕按钮关联。操作员单击该按钮可以打印包含当前值的可见历史趋势。

类别

历史

语法

```
PrintHT(Trend_Tag);
```

参数

Trend_Tag

历史趋势标记。

示例

本例打印屏幕上当前可见的 PumpPress 历史趋势。

```
PrintHT(PumpPress);
```

排解趋势的错误

您可以创建 QuickScript 来验证成功检索的数据是否出现在历史趋势中。使用 `HTGetLastError()` 函数来排解趋势的错误。

HTGetLastError() 函数

`HTGetLastError()` 函数可以在脚本中用来确定上一次为指定的历史趋势笔进行数据检索时是否发生了错误。

类别

历史

语法

```
[Result=]HTGetLastError(Hist_Tag, UpdateCount,  
    PenNum);
```

参数

Hist_Tag

指定了趋势的名称的“历史趋势”标记。

UpdateCount

代表趋势的 .UpdateCount 点域的整数。

PenNum

代表趋势的笔号（从 1 到 8）的整型标记或值。

Result

指定给某个标记的整数，代表对指定的笔进行的上一个脚本函数调用的状态。

0 = 没有错误

1 = 一般服务器错误

2 = 旧的请求

3 = 文件错误

4 = 未加载服务器

5 = 函数内传递的趋势 / 笔不存在。

6 = 数据库中不存在此趋势标记

7 = 传递给函数的笔号无效（不在 1 到 8 之间）。

8 = 未给该笔号指定标记，或是未指定启用了记录功能的标记。

示例

以下语句检索上次检索 Trend1 趋势 3 号笔数据的状态，并将结果指定给 ResultCode 整型标记。

```
[ResultCode=]HTGetLastError("Trend1",
    Trend1.UpdateCount,3);
```

在动画“模拟值显示” QuickScript 中，会使用以下语句：

```
HTGetLastError("Trend1",Trend1.UpdateCount,3);
```

在趋势中显示实时值

您可以通过两种方法来创建实时趋势。“实时趋势”对象提供一套标准控件来选择数据、设置时间范围，以及指定图形的实际外观。InTouch 还包含 16-Pen Trend（16 笔趋势）向导，这是可用于创建实时和历史趋势的可选控件。如需有关使用 16-Pen Trend（16 笔趋势）向导创建实时趋势的详细信息，请参阅 *InTouch® HMI 辅助组件指南* 中的第 4 章“使用 16-Pen Trend 向导”。

使用实时趋势对象

您可以创建实时趋势在应用程序中显示当前值。

创建实时趋势

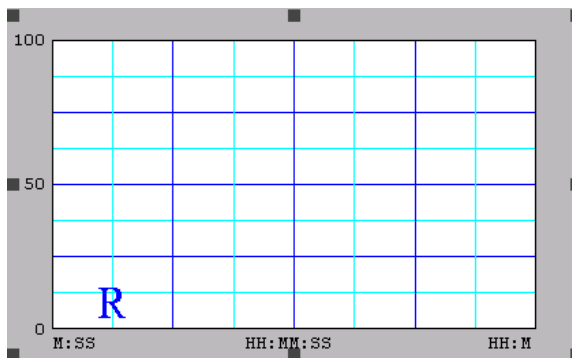
您可以使用“实时趋势”工具在窗口中创建一个趋势对象。第一次粘贴实时趋势对象时，WindowMaker 使用缺省设置。在配置实时趋势之后，WindowMaker 将上次配置的值用作任何新创建的实时趋势对象的初始设置。

您可以在窗口边框内绘制任意大小的趋势图。

要创建实时趋势



- 1 从“绘图”工具栏中选择**实时趋势**工具。
- 2 将鼠标移到要放置实时趋势的窗口区域上。沿着对角线拖动鼠标，根据所需的趋势大小来创建一个长方形。此时窗口中出现“实时趋势”对象。



- 3 如果需要，使用对象手柄来调整趋势的高度与宽度。

配置要在实时趋势上显示的标记

实时趋势笔创建当前数据的图形化表示，这些数据可来自任何本地标记或包含一个或多个本地标记的表达式。您可以配置这些在实时趋势中显示标记数据的笔。

要配置实时趋势标记

- 1 双击窗口中的趋势对象。此时出现**实时趋势配置**对话框。



- 2 在**表达式**区域中，输入本地标记的名称或包含一个或多个本地标记的表达式。

如果在**笔**框中双击，则出现**选择标记**对话框，显示为应用程序定义的一系列标记。通过从**选择标记**对话框中选择标记，可以将它指定给笔。

- 3 单击指定了标记的每支笔旁边的颜色框以显示调色板。
- 4 单击要指定给笔的颜色。
- 5 在**宽度**框中，为趋势中显示的每支笔输入以像素为单位的线条宽度。

选择大于 1 的线条宽度会增加更新或打印趋势所需的时间。

- 6 如果要使趋势仅当它显示在活动窗口中时才进行更新，请选择**只在内存中时才更新**复选框。

如果不选择此选项，则即便窗口关闭，趋势也在不断更新。持续更新趋势时会导致系统性能降低。

- 7 保持**实时趋势配置**对话框打开，转到“配置实时趋势的时间跨度与更新速率”中介绍的下一个操作程序。

配置实时趋势的时间跨度与更新速率

您可以配置实时趋势的时间跨度与更新速率。

要设置实时趋势的时间跨度与更新速率

- 1 如果需要，双击趋势对象。此时出现**实时趋势配置**对话框。
- 2 在**时间区域**的**时间跨度**框中，输入要在趋势的 **x** 横坐标轴上显示的时间长度。
- 3 选择趋势时间的度量单位。
 - 秒
 - 分
 - 小时

例如，如果在**时间跨度**框中输入 30，然后选择**分**，则图表上显示的时间跨度为 30 分钟。

- 4 在**样本区域**的**间隔**框中，输入对趋势表达式进行求值并更新图表的间隔时间量。
- 5 选择间隔的度量单位。
 - 毫秒
 - 秒
 - 分
 - 小时

例如，如果在**间隔**框中输入 10，然后选择**秒**，则实时趋势每 10 秒更新一次。

- 6 保持**实时趋势配置**对话框打开，转到“配置实时趋势的显示选项”中介绍的下一个操作程序。

配置实时趋势的显示选项

您可以配置实时趋势的视觉外观。

要配置实时趋势的显示选项

- 1 双击趋势对象。此时出现**实时趋势配置**对话框。
- 2 在**颜色**区域中配置颜色。执行以下任何一项操作：
 - 单击**图表颜色**框以打开调色板。为趋势选择背景颜色。
白色是缺省的背景色。任何其它背景颜色都会显著增加打印趋势需要的时间。
 - 单击**边框颜色**框以打开调色板。为趋势选择边框颜色。
- 3 在**时间刻度**区域中，配置时间刻度。执行以下操作：
 - 在**主刻度数**框中，输入趋势主时间刻度数。主时间刻度出现在趋势的时间水平坐标轴上。
主时间刻度数必须是**主刻度 / 时间标签**值的偶数倍。例如，刻度数 20 是**主刻度 / 时间标签**值 4 的偶数倍。

时间刻度(D)

主刻度数: 4 [Color Selection]

副/主刻度: 2 [Color Selection]

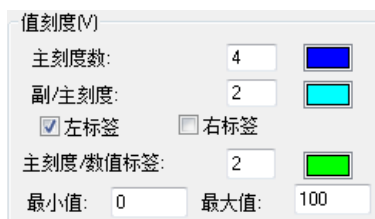
☐ 顶部标签 ☒ 底部标签

主刻度 / 时间标签: 2 [Color Selection]

HH:MM:SS 显示: ☒ HH ☒ MM ☒ SS

- 选择主刻度线的颜色。
- 在**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。
副时间刻度数应该是主刻度时段的偶数倍。例如，如果主刻度时段设置为 60 秒，在**副 / 主刻度**中输入值 2，则创建两个为期 30 秒的副时间刻度。
- 选择副刻度线的颜色。
- 选择**顶部标签**或**底部标签**复选框，以便指定时间标签在趋势上的位置。
您可以同时选择这两个选项，在趋势的顶部与底部都放上时间标签。将两个选项留为空白会从趋势的横坐标轴中删除时间标签。
- 如果使用时间标签，请在**主刻度 / 时间标签**中输入每个时间标签中的主时间刻度线数。主刻度数必须是**主刻度 / 时间标签**值的偶数倍。
选择时间刻度标签的颜色。
- 选择作为时间主刻度标签一部分显示的时间单位。
时 (HH)
分 (MM)
秒 (SS)

- 在**值刻度**区域中，配置趋势纵坐标轴的外观。



值刻度选项的配置方法与**时间刻度**选项的类似。y 坐标轴上的主、副刻度显示数据值（而非时间）的幅度。纵坐标轴根据所有标记的工程单位来指定趋势中出现的数据值范围。

要给数值主、副刻度显示小数点，请在**最小值**与**最大值**选项中输入实数。例如，0.00 到 100.00。

- 单击**选择显示字体**。此时出现**字体**对话框，它包含一些选项，可以设置趋势中出现的文本的字体、字形及大小。
- 单击**确定**。

在运行时打印趋势

有多个因素决定着趋势打印的速度。最主要的因素是打印页上趋势的大小。趋势的显示模式也会影响打印性能。“最小 / 最大”或“平均 / 散点”趋势的打印速度可以比“平均 / 棒图”趋势的快许多。而且，趋势上的线条越长、越宽，所需的打印时间也越长。

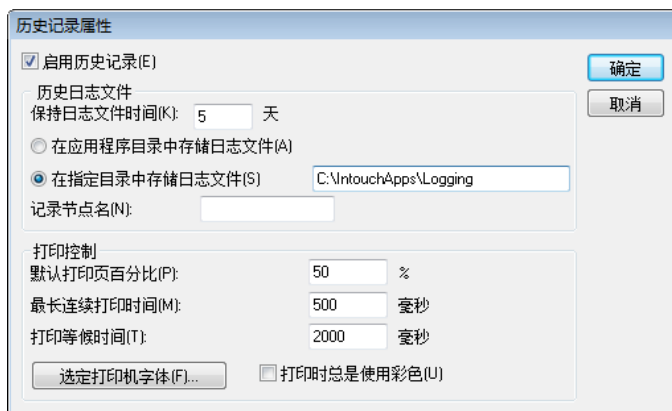
影响打印性能的另一个因素是趋势的背景颜色。在大多数情况下，白色背景在打印时要比彩色背景快许多。

配置趋势打印选项

您可以配置确定如何打印趋势的一些选项。

要配置历史趋势打印

- 1 在**特别菜单**上，单击**配置**，然后单击**历史记录**。此时出现**历史记录属性**对话框。



- 2 在**打印控制**区域的**默认打印页百分比**框中，指定用于打印趋势的页面百分比。

如果输入 50，趋势将打印到垂直和水平方向的一半页面上。按 50% 打印的趋势所花费的时间要比整页打印趋势少许多。

作为另一个打印选项，您可以使用 `PrintWindow()` `QuickScript` 函数。

- 3 在**最长连续打印时间**框中，输入以毫秒为单位的处理时间片断。

时间片断代表分配给计算机处理器以便在前台运行打印模块进程并打印趋势的时段。时间片段越长，趋势的打印速度越快，但这是以牺牲计算机上运行的其它进程为代价的。

- 4 在**打印等候时间**框中，以毫秒为单位输入打印模块在处理器时间片断之间等待的时间。

处理器时间片断之间的等待时段越短，趋势的打印速度越快。

- 5 单击**选择打印机字体**。此时出现**字体**对话框。选择趋势出现中的文本的特征。

- 6 单击**确定**以保存打印配置并关闭**历史记录属性**对话框。

显示来自其它 InTouch 节点或 Wonderware Historian 的历史标记值

如果要使用远程存储的数据来创建历史趋势，则必须在 InTouch 历史供应器列表中注册远程供应器。此列表指定每个历史供应器的名称与网络位置。只要历史趋势笔指向远程历史供应器上的标记，便会引用这些名称。

您可以定义一个远程历史供应器，将历史趋势笔指定给远程位置上存储的标记数据。您可以：

- 配置远程历史供应器。
- 对笔进行配置以显示远程历史供应器的数据。
- 通过使用“标记浏览器”，将笔指定给远程历史供应器上存储的标记数据。
- 使用 QuickScript 将笔指定给远程标记。

如需有关使用远程历史供应器的数据的详细信息，请参阅 *InTouch® HMI 应用程序管理与扩展指南* 中的第 2 章“分发应用程序”。

配合使用 InTouch HMI 与 Wonderware Historian

Wonderware Historian 是一种实时关系型数据库，专门针对工业应用而设计。作为可选项，您可以将 Historical Logger 配置成将 InTouch 历史数据存储到 Wonderware Historian 数据库中。

备注 如需有关将 InTouch 历史数据记录到数据库的详细信息，请参阅 Wonderware Historian 文档。如需有关为 InTouch HMI 设置远程历史供应器的详细信息，请参阅 *InTouch® HMI 应用程序管理与扩展指南* 中的第 2 章“分发应用程序”。

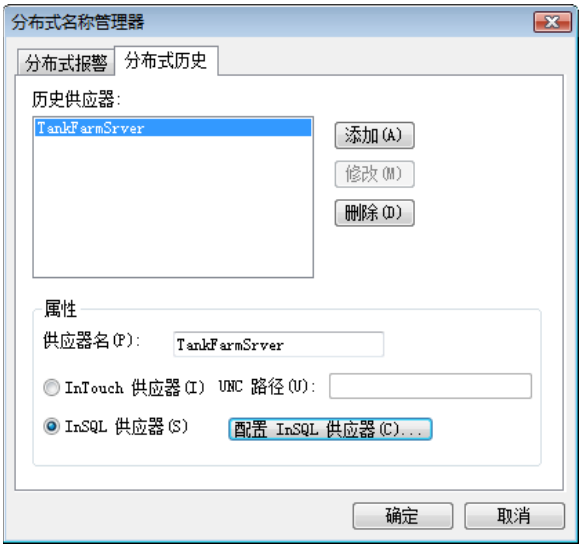
如果使用 Wonderware Historian 存储历史数据，则必须从 WindowMaker 中使用“分布式名称管理器”来指定同数据库的连接。

要配置与 Wonderware Historian 数据库的连接

- 1 打开 WindowMaker。
- 2 在“工具”视图中，展开**配置**列表。
- 3 选择**分布式名称管理器**。此时出现**分布式名称管理器**对话框。
- 4 在**分布式历史**选项卡上的**供应器名**框中，输入 **InSQL** 作为新的供应器。
- 5 选择 **InSQL 供应器**。

- 6 单击**配置 InSQL 供应器**。此时出现 **InSQL History Provider 属性**对话框。
 - a 在**数据源**框中，输入安装了 Wonderware Historian 的服务器的节点名。
 - b 输入 Wonderware Historian 数据库用户帐户的名称。
 - c 在**口令与再次输入口令**框中，输入用户帐户的口令。
 - d 单击**测试**以验证与 Wonderware Historian 数据库的连接。此时出现一条消息，指出与数据库的连接是否成功。
- 7 单击**确定**以关闭对话框。

- 8 单击**确定**以关闭 **InSQL History Providers 属性**对话框。此时 Wonderware Historian 节点出现在**历史供应器**列表中。



- 9 单击**确定**以关闭**分布式名称管理器**对话框。

对笔进行配置以显示远程趋势数据

历史趋势同时可以显示本地和远程历史供应器的标记数据。您可以指定趋势笔来显示远程历史供应器的数据。

要显示远程历史供应器中的标记

- 1 双击历史趋势以显示**历史趋势配置**对话框。
- 2 在每支笔的**标记名**框中，输入对远程历史供应器的引用。远程历史供应器的引用格式是：

history_provider_name.tag_name

示例：

TankFarm1.Pump1RPM

历史趋势的每支笔都可以引用不同的远程历史供应器。

- 3 单击**确定**以保存对配置所作的更改。

备注 .TagID 点域无法用在远程历史供应器的标记引用中。

使用标记浏览器将笔指定给远程历史供应器

以下操作程序介绍如何使用“标记浏览器”将趋势笔指定给来自远程历史供应器的标记数据。通过使用“标记浏览器”来选择标记，消除了手工输入每个标记名的必要，从而减少了出错的可能性。

在“访问名”中指定的远程节点名不必是标记所在节点的实际名称。但必须创建“访问名”以便将远程历史供应器定义为标记源。如需有关创建“访问名”的详细信息，请参阅第 68 页的“设置访问名”。

要将远程历史供应器定义为标记源

- 1 创建一个指定历史供应器所在节点名的“访问名”。
- 2 双击历史趋势以打开**历史趋势配置**对话框。
- 3 双击某支笔的**标记名**输入框，以显示**选择标记**对话框。
- 4 单击**定义标记源**以便将远程历史供应器定义为标记源。
- 5 单击**标记源**箭头，从列表中选择新的远程历史供应器标记源；或单击**目录树视图**按钮，并在目录树视图窗格中选择标记源。此时**选择标记**对话框刷新，显示来自所选的远程历史供应器的标记。
- 6 选择要指定给历史笔的标记并单击**确定**。此时再次出现**历史趋势配置**对话框，所选的标记按照此格式列在该笔的**标记名**框中：*AccessName:Item*。
- 7 使用您在**分布式名称管理器**中定义的历史供应器名替代 AccessName: 部分。

例如，HistPrv1.tag_name

这个过程看似麻烦，但在“标记浏览器”中将历史供应器定义为标记源之后，每次双击另一个标记名输入框时，只要在**标记浏览器**中双击标记名，然后将 AccessName: 部分替换成历史供应器名即可。

备注 在 WindowViewer 中，如果允许在运行时更改历史趋势，则用户单击笔按钮更改标记名时，会出现“标记浏览器”，但只能访问应用程序的本地标记。

使用 QuickScript 将笔指定给远程历史供应器

在 InTouch 应用程序正在运行期间，您可以配置趋势笔以便显示来自远程历史供应器的标记数据。创建一个 QuickScript，在 HTSetPenName() 函数中指定远程历史供应器标记引用。例如：

```
HTSetPenName("HistTrendTag", 1,  
    "HistPrv1.Boiler1");
```

在本例中，数字 1 指定历史趋势中的笔号，该笔绘制来自 HistPrv1 远程历史供应器的 **Boiler1** 远程标记的值。

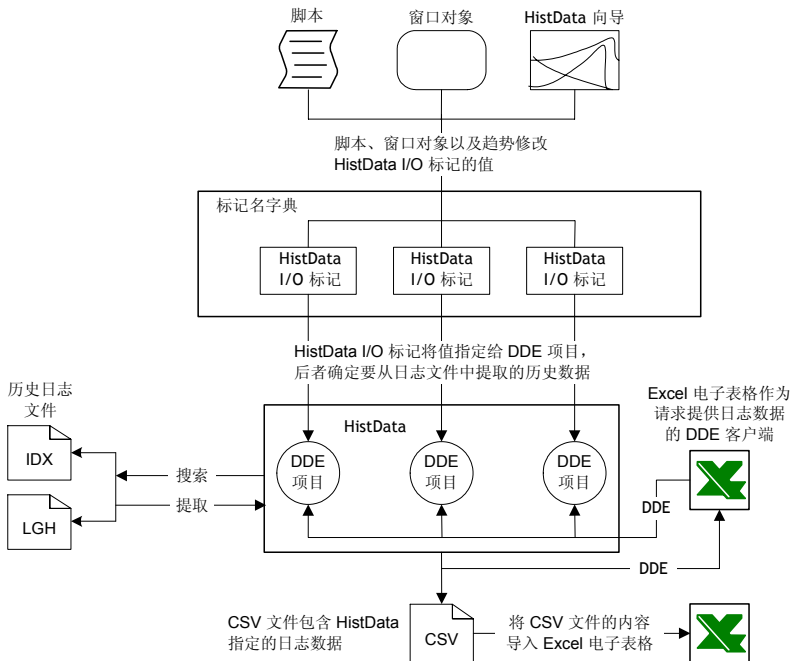
备注 对于远程历史供应器，不支持运行时的**历史趋势设置**对话框与**笔点域**。

第 11 章

从其它应用程序访问历史标记值

您可以使用 InTouch HistData 实用程序将数据从历史日志文件中提取到逗号分隔值 (.csv) 文件。Excel 这类的应用程序可以作为 DDE 客户端直接从 HistData 中提取 InTouch 记录数据，或从 HistData 实用程序创建的输出文件中导入记录数据。

下图显示将所选历史记录数据保存到文件或 DDE 客户端应用程序的过程。



使用 DDE 项目显示历史数据

HistData 程序包含一组 DDE 项目，指定如何从日志文件中提取历史数据。这些项目是 HistData 内部数据库的一部分。您可以将值指定给每个项目。

下表概括了在 HistData 程序中定义的 HistData 项目。

项目	数据类型	描述
DATADIR	消息	包含历史日志文件的文件夹的路径。
DBDIR	消息	包含 “InTouch 标记名字典” 内容的文件夹的路径。
STARTDATE	消息	从日志文件中提取数据的开始日期。开始日期的格式为 MM/DD/YY。
STARTTIME	消息	从日志文件中提取数据的开始时间。开始时间的格式为 24 小时制的 HH:MM:SS 格式。
DURATION	消息	<p>从日志文件中进行数据采集的间隔长度。DURATION 可以表示为：</p> <ul style="list-style-type: none"> • 星期 (w) • 天 (d) • 小时 (h) • 分 (m) • 秒 (s) <p>DURATION 周期可以指定为小数。例如，DURATION=0.5m 相当于 30 秒。如果要求提供一个样本，请将 DURATION 设置为 0。</p>
INTERVAL	消息	<p>数据采集间隔的时间长度。INTERVAL 可以表示为星期、天、小时、分以及秒。INTERVAL 周期的时间单位与 DURATION 周期的时间单位相同。</p> <p>间隔可以指定为小数。例如，INTERVAL=0.25d 表示 6 个小时。</p> <p>DURATION 或 INTERVAL 的最大周期为六个星期。六个星期的最大周期适用于指定给 DURATION 或 INTERVAL 的任何时间值。例如，42 是 DURATION 或 INTERVAL 周期的最大天数。</p>
FILENAME	消息	某个文件的名称与文件夹位置，该文件包含从历史日志文件中提取的数据。

项目	数据类型	描述
WRITEFILE	整型	一个标识，指出输出文件的 HistData 写入操作状态。设置为 1 时，HistData 将请求的数据写入 FILENAME “项目名” 所指定的文件。文件更新完成时，WRITEFILE 自动重置成 0。
ERROR	消息	一个字符串，它包含从日志文件中提取数据时发生的最后一个错误的描述。STATUS 设置为 1 时，ERROR 字符串设置为 “无”。STATUS 设置为 0 时，ERROR 字符串包含错误消息。
TAGS1, TAGS2, ...	消息	<p>包含一个或多个标记名的字符串，这些标记的数据从将日志文件中提取。</p> <p>TAGS 字符串在 WindowViewer 中最多可以包含 131 个字符，在 Excel 中最多可以包含 255 个字符。通过添加名称为 Tagsn 的标记项，可以给该字符串附加内容以满足更长的请求，其中 n 代表一个递增的整数。</p> <p>如果某个标记需要额外的标记文本，请在字符串末尾输入一个加号 (+)。</p> <p>例如：</p> <pre>TAGS="\$Date,ProdLevel,ProdTemp,+" TAGS1= "ReactLevel,Temp,GasLevel,+" TAGS2="MotorStatus"</pre> <p>不允许使用重复的标记名，每个标记字符串的最大长度为 512 个字节。</p>
PRINTTAGNAMES	离散	一个标识，指出标记的名称是否放到关联的值列的上方。设置为 1 时，打印标记名。设置为 0 时，不打印标记名。
DATA	消息	此项目按逗号分隔值的格式在 HistData 程序中保存请求的数据。它由其它应用程序用来通过 DDE 进行数据的 ADVISE 或 REQUEST。
STATUS	离散	HistData 最新操作的状态。值为 1 表示 HistData 从日志文件中成功地提取了历史数据。值为 0 表示发生了一个错误。
SENDDATA	整型	<p>表示 HistData 更新操作状态的标识。设置为 1 时，HistData 使用所请求的数据更新 DATA 项。更新完成时，SENDDATA 自动重置为 0。</p> <p>如果使用 SENDDATA 时收到错误消息，指出请求的数据过多，请缩短 DURATION 期限或减少请求的标记数。不允许使用重复的标记名，每个标记字符串的最大长度为 512 个字节。</p>

使用 DDE 访问记录数据

您可以使用两种方法将记录数据提取到输出文件中。

- 如果要将八个或更多标记的历史记录数据保存到输出文件，则可以使用手工方法。
- 如果仅需要保存映射到当前指定给历史趋势的笔的记录数据，请使用“历史数据向导”。

使用 HistData 手工提取记录数据

您可以将历史数据手工提取到输出文件中。按以下顺序完成各个步骤。

- 创建 HistData “访问名”
- 为 HistData 创建 I/O 标记
- 创建 HistData 窗口
- 运行 HistData

创建 HistData “访问名”

为了让 InTouch 从 HistData 程序中索取数据，必须定义一个“访问名”。

要定义访问名

- 1 在**特别菜单**上，单击**访问名**。此时出现**访问名**对话框。
- 2 单击**添加**。此时出现**添加访问名**对话框。

该对话框用于配置 HistData 的访问名。其包含以下字段和选项：

- 访问名(M):** 文本框，输入 ViewStream1。
- 节点名:** 文本框，输入 TankFrmSrvr。
- 应用程序名(A):** 文本框，输入 HistData。
- 主题名(T):** 文本框，输入 ViewStream1。
- 要使用的协议(P):** 包含三个单选按钮：DDE (已选中)、SuiteLink 和 消息交换。
- 要对服务器提示时(W):** 包含两个单选按钮：提示所有项 (已选中) 和 只提示激活项。
- 启用辅助数据源:** 复选框，当前未选中。

对话框右侧包含三个按钮：**确定**、**取消** 和 **故障转移(F)**。

- 3 在**访问名**框中，输入一个最多包含 32 个字母数字字符的名称。指定给访问名与主题名的值应该相同。
- 4 在**节点名**框中，输入日志文件当前所在节点的名称。
- 5 在**应用程序名**框中，输入不含 .exe 文件扩展名的 **HistData**。
- 6 在**主题名**框中，输入在**访问名**框中指定的名称。访问名与主题名应该相同。

- 7 选择 **DDE** 作为通讯协议。
- 8 在**要对服务器提示时**区域中，只要使用 **HistData**，便选择**提示所有项**。
- 9 单击**确定**。

创建 **HistData** 标记

在定义“访问名”之后，创建以下 I/O 型标记，以生成包含日志文件数据的输出文件。将上一步中创建的“访问名”指定给标记。

标记	I/O 标记类型	项目
HDWDATADIR	消息	DataDir
HDWDBDIR	消息	DbDir
HDWDURATION	消息	Duration
HDWERROR	消息	Error
HDWFILENAME	消息	FileName
HDWINTERVAL	消息	Interval
HDWSTARTDATE	消息	StartDate
HDWSTARTTIME	消息	StartTime
HDWSTATUS	消息	Status
HDWTAGS, HDWTAGS1, HDWTAGS2	消息	Tags
PRINTTAGNAMES	离散	PrintTagNames
HDWWRITEFILE	整型	WriteFile

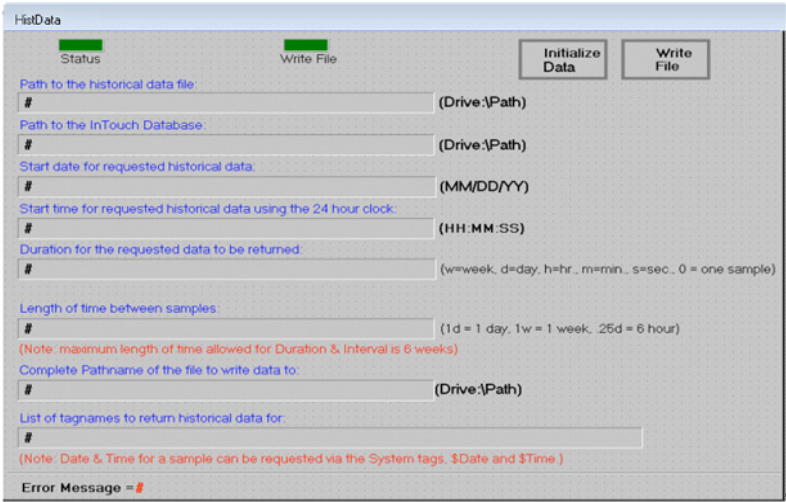
备注 “历史数据向导”自动创建除 **PRINTTAGNAMES** 标记以外的这些标记。

如果要将记录数据发送到 **Data** 项目中，以便可以从其它应用程序访问它，请创建两个额外的标记。而且“历史数据向导”不会自动创建 **HDWSendData** 及 **HDWData** 标记。

标记	I/O 标记类型	项目
HDWSendData	离散	SendData
HDWData	消息	Data

创建 HistData 窗口

在创建 I/O 型标记之后，创建一个名为 **HistData** 的新窗口，类似于下例：

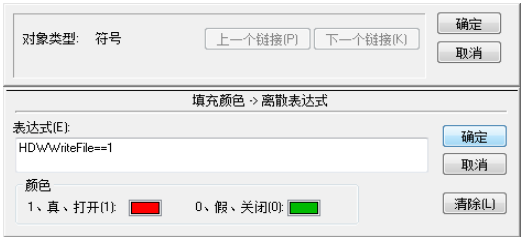


符号链接到一个用户输入链接。例如，# 符号有一个指向 HDWDataDir 标记的“用户输入 / 字符串”链接。该用户输入链接可用于在运行时更改标记的值。

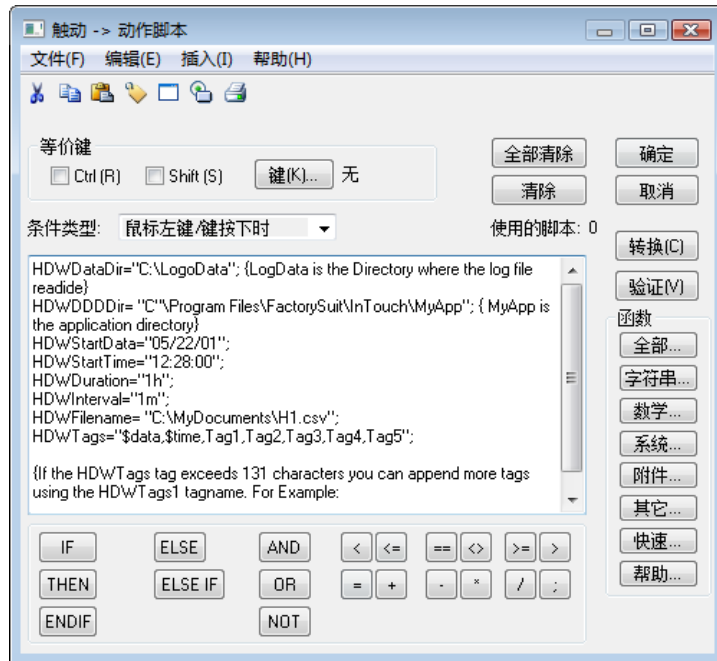
Status（状态）按钮链接到一个基于 HDWStatus 标记的“填充色—离散”表达式。



Write File（写入文件）按钮链接到一个基于 HDWWriteFile 标记的“填充色—离散”表达式。

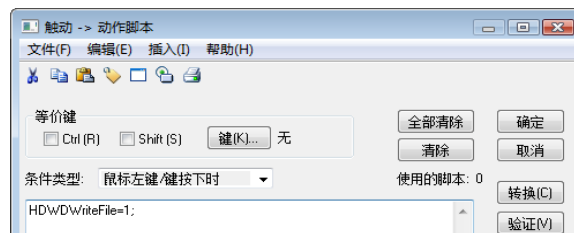


Initialize Data（初始化数据）按钮链接到“触动按钮”动作脚本。



单击 **Initialize Data**（初始化数据）按钮时，HistData 项将使用所需的值进行初始化。如果需要，也可以通过使用“用户输入链接”在运行时更改这些值。

Write File（写入文件）按钮链接到“触动按钮”动作脚本：



单击 **WriteFile**（写入文件）按钮时，生成输出文件。

运行 HistData

在创建 HistData 窗口之后，完成以下步骤以便在 WindowViewer 中运行它。

要运行 HistData 窗口

- 1 启动 HistData 并将它最小化。
- 2 启动 WindowViewer 并打开 HistData 窗口。
- 3 单击 **Initialize**（初始化）按钮并根据需要修改 HistData 项目。
- 4 单击 **WriteFile**（写入文件）按钮。


如果操作成功，**Status**（状态）值将变为 **ON**（打开），并呈现与 **ON**（打开）状态关联的颜色。如果操作不成功，**Status**（状态）的值将变为 **OFF**（关闭），并显示 **Error Message**（错误消息）以指明故障原因。

使用历史数据向导来提取记录数据

您可以创建一个输出文件，包含在历史趋势中出现的记录数据。InTouch 包含“历史数据向导”，可以将从日志文件中提取数据的步骤自动化。

由于 HistData 输出文件写入历史趋势中显示的日志数据，因此该文件只能包含当前指定给历史趋势笔的标记的数据。

使用“历史数据向导”来提取记录数据

- 1 启动 WindowMaker。
- 2 打开一个包含历史趋势的窗口。
- 3 从菜单栏中单击向导工具。此时出现**向导选择**对话框。
- 4 从左侧面板中选择**趋势**。
- 5  从右侧面板中选择**历史数据向导**图标，然后单击**确定**。
- 6 将鼠标指针移到要放置 HistData 对象的趋势窗口区域。
- 7 通过单击将 HistData 对象放置在趋势窗口中。“历史数据向导”创建一个窗口对象，其中包括一个按钮，以及**文件名**框，它显示创建输出文件的位置的路径。

保存至文件 文件名 c:\intouch\datafile.csv

- 8 双击趋势窗口中放置的“历史数据向导”对象。此时出现历史数据面板向导对话框。



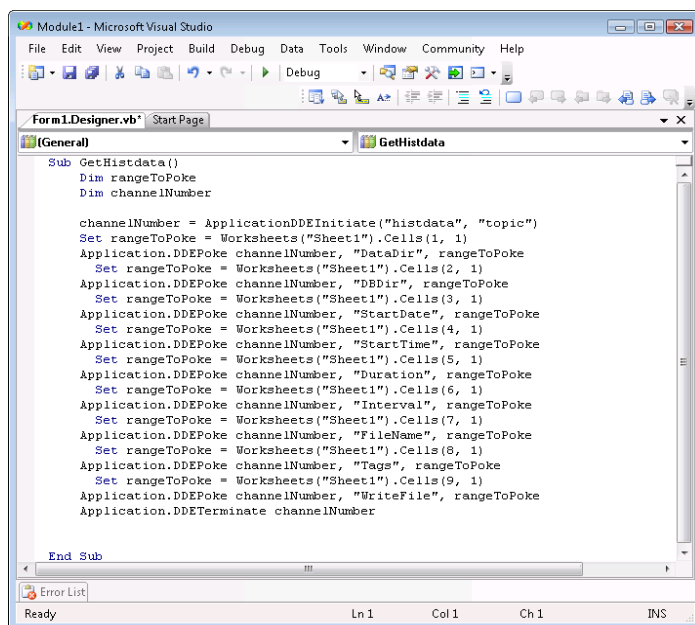
- 9 在历史趋势框中，输入“历史趋势”标记的名称。
- 10 在写入每一 CSV 文件的记录数框中，输入要写入输出文件的记录数。
- 11 单击确定。“历史数据向导”创建一组使用 HDW 前缀的标记。
“历史数据向导”创建第 229 页的“创建 HistData 标记”中列出的标记。“历史数据向导”将这些标记指定给 HisDataViewSt “访问名”。
- 12 使用 WindowViewer 运行历史趋势窗口。
- 13 单击保存至文件（它是 HistData 窗口对象的一部分）。HistData 在窗口对象中显示的文件夹位置创建输出文件。

从其它应用程序中访问历史数据

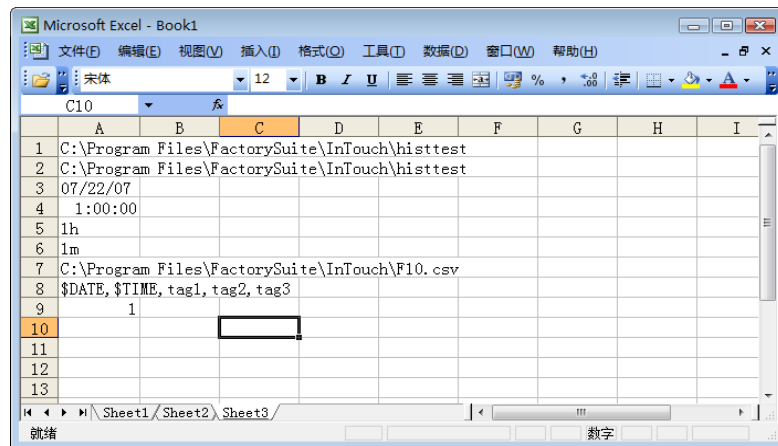
您可以编写 Excel 宏，以便从历史数据文件提取数据。

HistData 程序会对宏中的 INITIATE、POKE 及 TERMINATE 函数作出响应。带关键字（内部数据库项目）的 POKE 函数设置定义查询的参数。在正确指定查询之后，会运行宏向 HistData 文件请求提供所选的历史数据。

下例显示使用 VBA 编写的一个宏：



在上述示例中，要插入的数据在 Sheet1 中。下例显示要插入的数据：



排解 HistData 错误

使用 HistData 提取记录数据时，可能会看到发生一些错误。下表在左侧一栏列出典型的 HistData 问题或错误消息。右侧一栏描述该问题的可能原因及解决方案。

错误消息或条件	原因与 / 或解决方案
错误消息： 请求的数据太多 - 请缩短持续时间或减少标记名数。	SendData 项目请求过多的数据时，便会发生此错误。如果唯一目的是创建一个输出文件，用它来储存日志文件的数据，请不要使用 SendData 项目。
错误消息： 无法打开文件 C:\FILES1\HISTDATA.CSV	文件夹路径不存在，或文件夹路径的拼写不正确。
错误消息： 无法打开文件 C:\FILES\	未定义输出文件。
错误消息： DATADIR 项目无效	DataDir 项目指定的目标文件夹路径不存在。验证文件夹路径的拼写是否正确。
错误消息： STARTDATE 项目无效	StartDate 项目包含无效的开始日期格式。在 Windows 中，将计算机的日期格式更改为 mm/dd/yy。
错误消息： 未找到日志文件	DataDir 项目指定的路径中没有所请求的日期的日志文件。
错误消息： 在数据库中找不到标记名 TAGi	在应用程序的“标记名字典”中不存在所请求的标记。验证标记名的拼写是否正确。
错误消息： 在 C:\IT6.0B\HISTEST 中找不到 tagname.x	tagname.x 文件不存在或已损坏。

错误消息或条件	原因与 / 或解决方案
未创建 .csv 输出文件且没有出现错误。	<ul style="list-style-type: none"> • HistData 没有运行。 • Tags 项没有列出已指定要记录的任何标记。 • HDWWritefile 定义不正确。确保标记是整型标记, “DDE 访问名” 正确且项目是 WriteFile。此外, 确保 MinEU=MinRaw 且 MaxEU=MaxRaw 时未进行缩放。
.csv 输出文件包含日期与时间标签, 但不包含所请求的标记的任何记录数据。	在所请求的时段内, 这些记录的“历史”记录中没有任何项目。显示一个历史趋势, 以验证日志文件在所请求的时段内是否包含数据。
WWLogger 包含以下消息: DDE HistData Viewstream1 错误! WriteFile: 插入被服务器拒绝。	由于错误会给 HistData 项目指定值, 因此每次创建 .csv 文件失败时, 都会将错误消息写入 WWLogger。如果尝试将 WriteFile 项目的值设置为 0, 或尝试写入错误项目, 则也会出现此错误。
日志文件中应该包含许多条记录时, .csv 文件只包含一个记录。	可能给 Interval 项目指定了不正确的值, 导致采集间隔非常小。也可能由于 Duration 项目使用了不正确的格式, 如 HDWDuration=1- (未指定增量)。

附录 A

IEEE 十进制单位

InTouch HMI 采用 “电气电子工程师学会” (Institute of Electrical and Electronics Engineers, 简称 IEEE) 754 标准, 将 32 位二进制值转换为十进制浮点数。

在 16 位 “可编程逻辑控制器” (Programming Logic Controllers 简称 PLC) 中, 符合 IEEE 754 标准的 32 位数字存储为两个 16 位字。对于低位与高位十六进制字, PLC 中的浮点寄存器通常采取顺序编号机制。目前这代 32 位个人计算机使用单个 32 位寄存器。寄存器的位编号机制与两个顺序的 16 位寄存器采用相同的格式。

要在 InTouch 应用程序中使用浮点数, “I/O 服务器” 必须能够转换两个 16 位 PLC 寄存器中存储的值。InTouch 总是将原始的 PLC 寄存器值当作单独的整数, 因此必须进行位转换。对于这两个寄存器整数, 不能执行布尔 AND 运算, 也不能将它们转换为实数。InTouch 无法对双整数寄存器执行类型转换。

在 InTouch HMI 中显示浮点数

InTouch HMI 使用 IEEE 32 位浮点格式在应用程序中显示实数。IEEE 浮点格式只是实际实数的一个近似值。除非实数是二的偶数次方，否则不能用 IEEE 32 位浮点格式完全准确地表示。IEEE 32 位浮点数的精度约为十进制小数点后八位。

要在 InTouch 应用程序中显示实数时，确保该数字不超过八位。以下浮点格式显示 InTouch 应用程序中有效的实数：

`#.#####`

`###.#####`

`#####.##`

任何超过八位的浮点数都会发生舍入错误。

示例

InTouch 应用程序显示实数 2.3。但 2.3 这个数字不是二的偶数次方，因此不能使用 IEEE 32 位浮点格式精确地表示为 8 个以上小数位的十进制数。

要确保从应用程序中将 2.3 这个值显示为 ASCII 字符 2.3，该数字不得超过八位。如果该数字超过最大值八位，则产生得数字可能显示为 2.29999999 或 2.30000001。

附录 B

InTouch 许可证

Wonderware 提供多种 InTouch 许可证。InTouch 许可证基于应用程序运行时可以使用的可变标记数。您需要理解 InTouch 许可证方案如何计算标记。您可以使用一组函数来计算应用程序中预期的远程引用标记数。

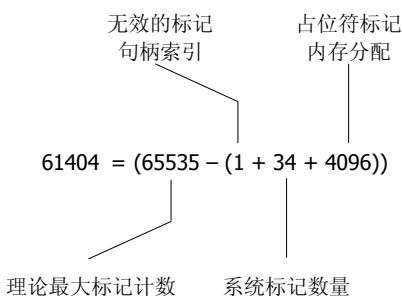
重要 许可证数量可随时更改。

理解许可证标记计数

在 InTouch 应用程序运行时，标记句柄存储在 64K 内存的数据库中。每个标记都必须指定一个句柄。标记句柄由 WindowViewer 初始化与使用，但在应用程序停止之后，标记句柄不会永久保存到磁盘。

此运行时数据库理论上可以存储 65535 个标记句柄，包括本地标记与引用远程标记源的标记。InTouch 应用程序使用的同时活动标记数永远不会超过这个运行时数据库中的可用句柄数。InTouch 许可证确定可以给多少个本地与远程标记指定运行时数据库中的句柄。

此外，应用程序中的最大活动标记数受限于运行时数据库的功能限制。实际最大标记计数小于理论最大标记句柄数 **65535**。下图显示 InTouch 的实际最大标记计数。最大潜在标记计数中要减去一组常数。



- 保留无效标记句柄位，以指出 WindowViewer 运行时数据库中是否出现无效的句柄值。
- InTouch 10 包含 34 个系统标记，这些标记无法由用户自定义的标记来替换。如果将 7.11 或更早版本的应用程序移植到当前版本的 InTouch，则系统标记计数是 37。
- 在配置时，会保留 4096 个数据库句柄以存储占位符标记。在配置期间导入窗口、脚本或符号时，占位符标记会指定给这个内存段。在运行时，所有这 4096 个占位符句柄都可以指定给远程引用标记。

根据应用程序中可以使用的最大本地标记与远程引用标记数，Wonderware 提供多种 InTouch 许可证选项。

下表显示 Wonderware 针对 InTouch 的一些典型许可证选项。除表中显示的示例之外，Wonderware 还提供更多的许可证选项。

最大本地 标记计数	最大远程引用标记计数	最大本地标记计数
512	256 个粘滞标记	768
8192	4096 个粘滞标记	12288
61404	60000 个活动标记	60000 (60K)

在表中显示的第一个示例中，您可以创建最多使用 512 个本地标记（许可证的最大限制）的应用程序。您最多可以添加 256 个远程引用标记。许可证限制总计标记计数为 768，即本地与远程引用计数之和。

表中所示的前两个许可证示例按粘滞方式实行远程引用标记计数。粘滞标记是一种远程标记引用，在运行时 WindowViewer 接收到远程引用的数据改变通知时，它才进行绑定。

WindowViewer 在运行时更新远程标记引用，最多可以达到 InTouch 许可证的最大限制。WindowViewer 不会更新超出许可证限制的任何额外远程标记引用。关闭窗口时，WindowViewer 不会减少远程引用标记计数。在应用程序运行时，每个远程引用标记计数都保持其粘滞性。

超出 InTouch 许可证的最大远程引用标记计数时，会出现一条消息。达到许可证最大值之后，与无效的远程引用标记关联的值不会在应用程序中进行更新。您必须停止后再重新启动应用程序，然后才可以打开包含一个或多个远程标记引用的其它窗口，这些引用尚未同根据许可证限制而统计的那些标记关联起来。

表中显示的 60K 许可证有些名不副实。最大标记数并不限于 60000。相反，根据 InTouch 运行时数据库所实现的限制，远程引用与本地标记的最大可能数量会略大于 60K。

使用 60K 许可证时，实现的本地与远程引用标记限制是：

- 可能的本地标记总数

$$61404 = 65535 - (4096 + 1 + 34)$$

- 可能的远程引用标记总数

$$\text{最大值} = 65535 - (1 + 34 + \# \text{ 本地标记})$$

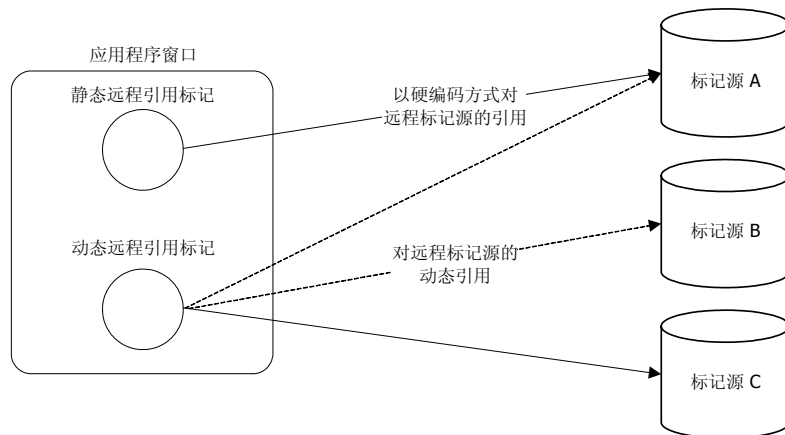
因此，使用 60K 许可证运行应用程序时，可以有效地使用每个潜在的远程引用来交换本地标记数据库中的每个用户自定义标记。在运行时配置中，可能的远程引用标记永远不会少于 4096 个。

理解 InTouch 远程引用限制

InTouch 远程引用标记分为两种类型。静态远程引用在您从“标记字典”中定义标记时明确指定到固定的远程地址。应用程序开始运行时，静态远程引用指定给标记数据库中的标记句柄。在应用程序运行时，静态远程引用标记计数保持其粘滞性。

动态远程引用在应用程序运行时解析目标地址。如果动态远程引用标记指定了数据库句柄，则通过在脚本中使用 `.Reference` 点域或 `IOSetRemoteReference()` 函数，可以在运行时更改目标地址。

下图显示 60K 许可证下运行的一个 InTouch 应用程序示例，它没有粘滞的远程引用标记。在应用程序运行时，静态远程引用标记计数会保持粘滞性。但是，动态远程引用标记计数仅用于活动标记源。以前与远程标记源的连接不粘滞，也不计入远程引用计数或总计标记计数。



InTouch 60K 许可证不使用在动态远程标记引用数上施加限制的粘滞标记计数。这使得应用程序可以在运行期间动态访问 60K 以上的标记。动态远程引用的标记使用计数随着包含远程引用的窗口的打开与关闭而上下波动。但是，应用程序同时使用的活动标记数不能比运行时标记数据库所实施的限制更多。

从 I/O 标记的远程引用不包含在 InTouch 许可证的粘滞远程引用计数中。I/O 标记的远程引用可以更改无限次，而不会根据粘滞远程引用限制进行统计。

您可以设置带主目标地址与辅助目标地址的 InTouch “访问名”。在应用程序运行时，辅助地址通常用作备份或故障转移标记源。应用程序在“访问名”的主标记源与辅助标记源之间切换时，不论“访问名”寻址如何变化，其粘滞远程引用标记计数不会递增。

远程标记计数函数

InTouch HMI 包含一组函数，可以验证应用程序是否符合许可证的远程标记要求。您可以编写一些包含这些函数的临时脚本，在部署到生产环境之前测试与确定应用程序是否有任何潜在的许可证问题。验证应用程序不存在任何许可证问题之后，请删除这些脚本。

IORRGetSystemInfo() 函数

IORRGetSystemInfo() 函数返回正在运行的 InTouch 应用程序的标记计数。IORRGetSystemInfo() 函数根据参数值返回一个数值，该数值可以是：

- InTouch 许可证指定的最大远程标记地址数
- 在 InTouch 应用程序运行期间根据许可证统计的远程标记地址数
- InTouch 应用程序中当前激活的远程标记数
- 正在运行的 InTouch 应用程序中可用的远程标记数
- 当前禁止的远程引用标记数
- 正在运行的 InTouch 应用程序中的本地标记数
- 如果函数调用期间发生错误，或 *Option* 参数指定为无效的值，则为 -1。

类别
其它

语法
`IORRGetSystemInfo(Option);`

参数

Option

整型标记或整型常数，指定要返回的远程引用标记计数的类型。可能的值有：

- 1 根据 InTouch 许可证返回允许的最大远程标记地址数。本地 I/O 标记不记入远程标记计数。在 InTouch 应用程序运行时，此数字为常数。

2 返回根据许可证限制统计的独特远程标记地址数，这些地址是在 InTouch 应用程序运行时激活的。本地 I/O 标记不记入远程标记计数。

如果许可证允许使用 60000 以上的远程引用标记，则不论激活了多少远程标记地址，此数字都可能是 0。在无限制许可证下运行时，WindowViewer 不统计激活的远程标记地址。

在具有远程标记限制的许可证下运行应用程序时，此计数会递增，直至达到远程标记许可证计数的限制。达到远程标记限制之后，便无法激活更多的远程标记地址。此时仅能重新激活当前已激活的地址。通过使用将 *Option* 参数设置为 3 的 IORRWriteState，可以获取根据许可证限制统计的远程引用标记的列表。

3 返回 InTouch 应用程序中当前已激活的远程引用标记数。

4 返回可以在 InTouch 应用程序中激活且不会导致缺少远程标记句柄的远程引用标记数。应用程序在运行时，此计数通常会发生变化。停止与启动脚本、打开与关闭包含远程引用标记的窗口时，都会影响远程引用标记计数。

此数字可以少于许可证上剩余的数字，尤其是在许可证没有限制的情况下。发生这种情况是因为存在一个内部限制来确定同时可以有多少个活动的远程引用标记。

5 返回 InTouch 应用程序中当前处于禁用状态的远程标记数。

6 返回 InTouch 应用程序中当前的本地标记数。

示例

以下示例返回 InTouch 应用程序正在运行时根据许可证限制统计的远程引用标记数。返回的远程引用标记计数被指定 RRTagCount 整型标记的值。

```
RRTagCount = IORRGetSystemInfo(2);
```

IORRWriteState() 函数

IORRWriteState() 函数将应用程序远程标记的当前状态信息保存到文本文件。如果不存在该文件，此函数会创建它。每次运行包含此函数的脚本时，都会将新的信息附加到该文件中。

您可以指定将哪些远程标记信息保存到文件。另外，此函数的返回值会指出信息是否已成功添加到文件中。

类别
其它

语法
`IORRWriteState(FilePath, Option, " ");`

参数

FilePath
文本文件的完整文件夹路径，该文件包含有关应用程序远程标记的信息。*FilePath* 参数可以是字符串常量或消息标记。

Option
整型标记或整型常数，指定写入文件的远程标记计数信息。可能的值有：

- | | |
|---|---|
| 1 | 当前远程标记地址的列表。此信息还包括每个远程标记的状态、激活时间及停用时间。 |
| 2 | 当前所有活动地址的列表，包括激活时间。 |
| 3 | 已激活并根据许可证进行统计的所有远程引用标记地址的列表。

激活新的远程标记地址时，新的项目会添加到列表中。达到许可证限制时，不再向列表中添加更多的项目。不过，如果远程标记许可证限制是无限的，则不向此列表中添加任何地址。 |
| 4 | 由于远程引用标记计数超出许可证限制，或由于达到了内部标记句柄限制而未激活的当前地址的列表。如果远程标记许可证限制是无限的，则不会返回与许可证相关的地址。

如果许可证是无限的，则列表包含由于实施限制而当前未处于活动状态的所有项目。如果此列表中的任何项目变为活动状态，则它会被从列表中删除。某个项目由于许可证限制而停用时，它会被从列表中删除。此列表在 InTouch 应用程序运行时进行更新。 |

" " (空字符串)
此参数保留下来供将来使用，但在脚本的 IORRWriteState() 函数中必须包含它。

结果
下面是一些示例，用来解释由 IORRWriteState() 函数调用保存到文件的信息。

当前地址列表

输出文件中的下面这行代码显示完全激活且可以更新的远程引用标记的示例。这行代码显示地址 **65535** 指定给了

TestProt:di000 远程引用标记：

```
65535 <TestProt:di000> (RAA) {C:5/23/2007 9:58:35 AM}
      {A:5/23/2007 9:58:35 AM}
```

远程引用标记名后面是圆括号括起来的三个标识：

- 第一个标识 **R** 表示标记的远程引用已经成功解析。如果标记的远程引用仍然等待处理，则指定给第一个标识的值是 **X**。
- 第二个标识表示远程标记当前处于活动状态 **A**，还是处于不活动状态 **D**。
- 第三个标识表示由于许可证限制该地址是允许的 **A**，还是不允许的 **D**。

C 后面的日期表示创建远程标记的时间。**A** 后面的日期显示最近激活标记的时间。如果远程引用标记已在 **InTouch** 应用程序运行时停用，这行代码会在最后包含一个停用时间。

输出文件中的下面这行代码显示超出 **InTouch** 许可证标记计数限制的活动远程引用标记的示例。标记的远程引用已成功解析，且标记当前处于活动状态：

```
65414 <TestProt:di121> (RAD) {C:5/23/2007 9:58:35 AM}
      {A:5/23/2007 9:58:35 AM}
```

但 **InTouch** 应用程序中不更新标记值，这是因为指定给远程引用标记的地址超出了 **InTouch** 许可证的标记计数限制。

活动地址列表

输出文件中的下面这行代码显示完全激活的远程标记的示例，指定给该标记的值会更新 **InTouch** 应用程序：

```
65429 <TestProt:di106> (A) {C:5/23/2007 9:58:35 AM}
      {A:5/23/2007 9:58:35 AM}
```

第一个数字是远程标记句柄，其后是地址，然后是表示允许的标识 **A**，或表示不允许的标识 **D**。输出行中标识的后面是创建时间、最近激活时间以及最近停用时间。如果该远程标记在应用程序运行期间尚未停用过，则停用时间不会出现。

输出文件中的下面这行代码显示超出许可证限制的活动远程标记的示例：

```
65342 <TestProt:di193> (D) {C:5/23/2007 9:58:35 AM}
      {A:5/23/2007 9:58:35 AM}
```

许可的地址

列表中的下面这行代码显示指定给远程引用标记的地址以及它添加到列表的时间。

<testprot:di000> {C:5/23/2007 9:58:36 AM}

拒绝的地址

由于实施限制或由于标记计数超出许可证最大值，被拒绝的地址出现在列表中。

本例显示超出许可证限制的远程标记地址：

testprot:di125 [1] (L) {F:5/23/2007 9:58:39 AM} {R:5/23/2007 9:58:39 AM}

地址与计数一起列出，指出尝试了多少次去引用该项目。标帜指出该地址是由于超出许可证限制 (L)，还是由于内部实施限制 (I) 而出现在列表中。两个时间分别代表它第一次添加到列表的时间与最近访问的时间。

示例

本例将当前激活的远程标记地址写入到 c:\intouch\data 文件夹下的文件中。Return Value 标记会被指定给一个整数，指出函数调用是否成功地将远程标记信息写入文件中。

```
Return Value = IORRWriteState("c:\intouch\data", 2,
    "");
```

IORRGetItemActiveState() 函数

IORRGetItemActiveState() 函数返回指定的远程标记地址的状态。

类别

其它

语法

IORRGetItemActiveState(*ItemPath*, *Option*);

参数

ItemPath

ItemPath 是代表所关注的地址的字符串。 *ItemPath* 可以是字符串常量或消息标记。

Option

整型标记或整型常数，指定要返回的远程引用标记计数的类型。可能的值有：

- | | |
|---|---|
| 1 | 确定当前远程标记地址目前是否处于活动状态。
如果是当前的活动地址，则返回值为 1。如果不是当前地址，则返回值为 -1。如果是当前地址但并非处于活动状态，则返回值为 0。 |
|---|---|

-
- | | |
|-------|---|
| 2 | 确定当前远程标记地址在应用程序运行期间是否激活过。

如果是当前地址，并且至少激活过一次，则返回值为 1。如果不是当前地址，则返回值为 -1。如果是当前地址但从未激活过，则返回值为 0。 |
| <hr/> | |
| 3 | 确定当前远程标记地址是否停用过。

如果不是当前地址，则返回值为 -1。如果是当前地址但从未停用过，则返回值为 0。 |
| <hr/> | |
| 4 | 确定当前远程标记地址是否已禁用。

如果是当前地址，且至少停用过一次，则返回值为 1。如果不是当前地址，则返回值为 -1。如果地址未禁用，则返回值为 0。如果是当前地址，并且已禁用，则返回值为 1。 |
| <hr/> | |
| 5 | 确定地址是否在允许的列表中。

如果地址不在列表中，则返回值为 0。如果地址在列表中，则返回值为 1。 |
| <hr/> | |
| 6 | 确定地址是否在拒绝的列表中。

如果地址不在列表中，则返回值为 0。如果地址在列表中，则返回值为 1。 |
-

示例

本例确定 TestProt:di000 远程标记地址当前是否处于活动状态：

```
ReturnValue = IORRGetItemActiveState("TestProt:di000",  
1);
```

本例确定 TestProt:di121 远程标记地址当前是否已禁用：

```
ReturnValue = IORRGetItemActiveState("TestProt:di121",  
4);
```

本例确定当前是否根据许可证限制来统计 TestProt:di001 远程标记地址。

```
ReturnValue = IORRGetItemActiveState("TestProt:di001",  
5);
```


索引

符号

\$AccessLevel 系统标记 27, 39
\$ApplicationChanged 系统标记 39
\$ApplicationVersion 系统标记 39
\$Date 系统标记 40
\$DateTime 系统标记 40
\$Day 系统标记 40
\$HistoricalLogging 系统标记 40, 167
\$Hour 系统标记 40
\$InactivityTimeout 系统标记 41
\$InactivityWarning 系统标记 41
\$Language 系统标记 41
\$LogicRunning 系统标记 41
\$Minute 系统标记 41
\$Month 系统标记 41
\$Msec 系统标记 41
\$NewAlarm 系统标记 41
\$ObjHor 系统标记 42
\$ObjVer 系统标记 42
\$Operator 系统标记 42
\$OperatorDomain 系统标记 42
\$OperatorDomainEntered 系统标记 42
\$OperatorName 系统标记 42
\$PasswordEntered 系统标记 43
\$Second 系统标记 43

\$StartDdeConversations 系统标记 43
\$System 系统标记 43
\$Time 系统标记 43
\$TimeString 系统标记 43
\$VerifiedUserName 系统标记 43
\$Year 系统标记 43

A

Application Server 93
 将数据类型映射到 InTouch 数据类型 97–98
 描述 93
 属性的读取 / 写入行为 99–100
ArcheStrA 批量导入实用程序
 描述 150
 移植 SuperTag 150

B

保留参数 32
标记
 保留参数 32
 标记 ID 类型 19
 查看时间标签与质量信息 105–111
 查看质量信息 112–117
 初始值 30

- 从导入的窗口中将标记转换为远程引用 85
- 从其它应用程序中访问数据 134
- 打印 37
- 定义 I/O 离散 72–73
- 定义 I/O 消息 76
- 定义 I/O 整型与实型 75
- 定义内存消息 36
- 定义内存整型与实型 35
- 动态远程引用 242
- 度量单位 30
- 废弃类型 20
- 更新本地标记计数 152
- 更新本地计数 153
- 更新远程计数 153
- 工程单位 75
- 公共属性 28–29
- 规划使用 26
- 恢复远程引用 92
- I/O 标记的输入转换 73
- I/O 消息的详细资料
 - 项目 77
- I/O 型属性 22–??
- 间接 135–139
- 检索远程引用 87
- 将 Galaxy 用作远程标记源 101–104
- 将原始数据值转换为工程单位值 74
- 交叉引用搜索标准 156
- 静态远程引用 242
- 句柄 239
- 类型 15–19
- 理论许可证最大数量 153
- 历史趋势类型 19
- 描述 13
- 命名惯例 28–29
- 内存型属性 21
- 配置记录功能 33
- 配置历史记录功能 163–164
- 平方根缩放 75
- 其它类型 19
- 确定使用 152
- SuperTag 详细资料
 - 数据访问 147
- 删除 37

- 删除未使用的 160
- 设置 I/O 属性 76
- 设置死区 32
- 生命周期 14
- 使用“动态引用寻址” 79–80
- 手工将标记转换为远程引用 84
- 添加到 SuperTag 实例 148
- 通过远程引用访问 I/O 数据 87–88
- 线性缩放 75
- 修改 36
- 许可证中的占位符计数 240
- 远程引用 24
- 在运行时动态更改 I/O 引用 79–82
- 在运行时检索信息 77–78
- 在运行时重定向远程引用 89–91
- 在“标记浏览器”中选择远程引用 86
- 值保留 32
- 值范围 30
- 值死区 30–31
- 注释 29
- 转换为远程引用 83–86
- 自动编号的名称 29
- 最大工程单位 75
- 最大原始值 75
- 最小工程单位 75
- 最小原始值 75
- 标记的命名惯例 28–29
- 标记名点域
 - .Comment 64
 - .MaxEU 点域 58, 59
 - .MinRaw 点域 55
 - .OnMsg 点域 62
 - .RawValue 点域 57
 - .Value 点域 61
- 标记名字典 145
- 标记创建综述 14
- 创建 I/O 离散标记 72–73
- 创建 I/O 消息标记 76
- 创建 SuperTag 实例 146–147
- 创建新标记 27
- 创建整型与实型 I/O 标记 75
- 打印内容 158
- 复制 SuperTag 实例 147

更新标记计数 152–153
 I/O 标记的属性 22–23
 将标记添加到 SuperTag 实例 148
 列出 SuperTag 142
 内存标记的属性 21–22
 设置 I/O 访问参数 76–77
 设置公共标记属性 28–29
 设置记录死区 30
 设置死区 32
 设置值范围、度量单位及初始值 30
 设置值死区 30
 输入标记注释 29
 修改 SuperTag 成员标记的特征 144
 选择标记类型 15
 指定标记属性综述 20
 自动给标记命名 29

C

.ChartLength 点域 185
 .ChartStart 点域 186
 .Comment 点域 64
 窗口
 打印 37
 将导入的标记转换为远程引用 85
 目录树视图 86

D

DDEStatus 主题名 130
 DDE, 请参阅 “动态数据交换”
 .DisplayMode 点域 179
 电气电子工程师学会 237
 点域
 .ChartLength 点域 185
 .ChartStart 点域 186
 .Comment 点域 64
 .DisplayMode 点域 179
 .EngUnits 点域 60
 列表 46–52
 .MaxEU 点域 58, 170, 171
 .MaxRange 点域 181
 .MaxRaw 点域 55
 .MinEU 点域 58, 59, 170, 171
 .MinRange 点域 179
 .MinRaw 点域 55

描述 45
 .Name 点域 136
 .OffMsg 点域 62
 .OnMsg 点域 62
 .Pen1-8 点域 187
 .Quality 点域 113
 .QualityLimit 点域 114
 .QualityLimitString 点域 115
 .QualityStatus 点域 115
 .QualityStatusString 点域 116
 .QualitySubstatus 点域 116
 .QualitySubstatusString 点域 117
 .RawValue 点域 57
 .Reference 点域 79
 .ReferenceComplete 点域 79
 .ScooterLockLeft 点域 189
 .ScooterLockRight 点域 190
 .ScooterPosLeft 点域 191
 .ScooterPosRight 点域 192
 .TagID 点域 188
 .TimeDate 点域 105
 .TimeDateString 105
 .TimeDateTime 点域 106
 .TimeDay 点域 107
 .TimeHour 点域 107
 .TimeMinute 点域 108
 .TimeMonth 点域 108
 .TimeMsec 点域 109
 .TimeSecond 点域 109
 .TimeTime 点域 110
 .TimeTimeString 点域 110
 .TimeYear 点域 111
 .Value 点域 61
 .UpdateCount 点域 182
 .UpdateInProgress 点域 183
 .UpdateTrend 点域 184
 用于本地间接标记 137
 语法 45
 值极限组 54
 动态数据交换
 DDEStatus 主题名 130
 Excel 远程引用公式 134
 访问记录数据 228–232

更新标记数据的质量状态 113

HistData 项目 226–227

设置“访问名”协议 123

通讯协议 67

动态引用寻址

描述 79

使用 IOSetAccessName 函数实现 81

使用 IOSetItem 函数实现 79–80

使用 .Reference 点域实现 79

E

.EngUnits 点域 60

F

ForceLogCurrentValue 参数 166

ForceLogging 参数 166

访问名

编辑故障转移对的参数 123–124

创建 68–70

从 WindowViewer 中重新初始化 118

辅助备份服务器 70

获取故障转移对的状态 128–129

间接标记远程引用 138

检索故障转移对的有关信息 128

离散值显示动画链接 128

临时禁用故障转移 126–127

轮询信息 69

描述 65

配置辅助故障转移服务器 122–123

强制故障转移 124–125

确定应用程序 78

确定主题 78

删除 71

删除故障转移 124

使用故障转移功能 122–124

使用脚本重新初始化 I/O 连接 120–121

使用命令重新初始化 I/O 连接 118–119

使用缺省设置重新初始化 119

通讯协议 69

许可证注意事项 242

选择服务器程序中的数据点 77

应用程序名 69

在运行时从 WindowViewer 中重新初始化所选的“访问名” 118

指定给标记 76

主题名 69

浮点数 237

H

HistData

创建标记 229

创建窗口 230–231

创建访问名 228

从其它应用程序中访问数据 234

DDE 项目 226–227

描述 225

排解错误 235–236

使用向导来提取记录数据 232–233

手工提取记录数据 228–232

运行以提取历史数据 232

HTGetLastError() 函数 211

HTGetPenName() 函数 201

HTGetTimeAtScooter() 函数 202

HTGetTimeStringAtScooter() 函数 203

HTGetValue() 函数 204

HTGetValueAtScooter() 函数 205

HTGetValueAtZone() 函数 206

HTScrollLeft() 函数 207

HTScrollRight() 函数 208

HTSelectTag() 函数 199

HTSetPenName() 函数 200

HTUpdateToCurrentTime() 函数 198

HTZoomIn() 函数 209

HTZoomOut() 函数 210

函数

HTGetLastError() 函数 211

HTGetPenName() 函数 201

HTGetTimeAtScooter() 函数 202

HTGetTimeStringAtScooter() 函数 203

HTGetValue() 函数 204

HTGetValueAtScooter() 函数 205

HTGetValueAtZone() 函数 206

HTScrollLeft() 函数 207

HTScrollRight() 函数 208

HTSelectTag() 函数 199

HTSetPenName() 函数 200

HTUpdateToCurrentTime() 函数 198

- HTZoomIn() 函数 209
 - HTZoomOut() 函数 210
 - IOGetActiveSourceName() 函数 129
 - IOGetApplication() 函数 78
 - IOGetNode() 函数 77, 78
 - IOGetTopic() 函数 78
 - IOReinitialize() 函数 121
 - IORRGetItemActiveState() 247–248
 - IORRGetSystemInfo() 243–244
 - IORRWriteState() 245–247
 - IOSetAccessName() 函数 81
 - IOSetItem() 函数 79
 - IOSetRemoteReferences() 函数 89–92
 - IOStartUninitConversations() 函数 121
 - PrintHT() 函数 211
- I**
- I/O 寻址 134
 - I/Status
 - 主题名 130
 - IEEE, 请参阅 “电气电子工程师学会”
 - InTouchEvent 68
 - IOGetActiveSourceName() 函数 129
 - IOGetApplication() 函数 78
 - IOGetNode() 函数 77, 78
 - IOGetTopic() 函数 78
 - IOReinitialize() 函数 121
 - IORRGetItemActiveState() 函数 247–248
 - IORRGetSystemInfo() 函数 243–244
 - IORRWriteState() 函数 245–247
 - IOSetAccessName() 函数 81
 - IOSetItem() 函数 79
 - IOSetRemoteReferences() 函数 89–91
 - IOStartUninitConversations() 函数 121
 - IOStatus 主题
 - 监视 I/O 通讯的状态 130–132
 - 监视 I/O 通讯状态 132
 - 描述 130
 - 用于 Excel 132
- J**
- 记录死区
 - 描述 30
 - 设置 32
 - 间接标记
 - 串联名称 136
 - 赋给源标记 136
 - 描述 135
 - 用于本地标记 137
 - 用于远程引用 138–139
 - 交叉引用实用程序
 - 报告 154–155
 - 创建报告 156–157
 - 过滤器通配符 156
 - 将报告保存到文件 157, 158
 - 描述 151
 - 搜索全部项目 156
 - 搜索指定项目 156
 - 均值 / 棒图历史趋势 170
- L**
- 历史记录功能
 - 控制记录频率 166–167
 - 描述 161
 - 配置常规属性 164–165
 - 配置与 Wonderware Historian 数据库的连接 220–221
 - 数据采集周期 162
 - 文件保留 162
 - 文件命名惯例 162
 - 在运行时停止与启动 167
 - 历史趋势
 - 创建 172
 - 打印 218
 - 将标记指定给笔 173–174
 - 描述 170
 - 配置时间跨度与更新速率 175
 - 配置显示选项 176
 - 平均显示模式 175
 - 使用点域将标记指定给笔 187–188
 - 使用点域控制显示 179–192
 - 使用点域设置指示器属性 189–192
 - 使用脚本进行控制 198–211
 - 使用 “历史趋势图表向导” 配置显示选项 196–197
 - 使用 “历史趋势向导” 创建 194
 - 使用 “历史趋势向导” 指定标记 195
 - 选择显示模式 175

- 运行时配置 197–198
 - 在运行时更改配置 177–178
 - 在运行时使用点域进行配置 178
 - 在运行时使用脚本函数进行配置 178
- 最小 / 最大显示模式 175
- 历史趋势向导
 - 创建趋势 194
 - 将标记指定给笔 195
 - 描述 193
 - 配置趋势的显示选项 196–197
 - 设置趋势的时间跨度 195–196
- 联系技术支持 12

M

- .MaxEU 点域 58, 170, 171
- .MaxRange 点域 181
- .MaxRaw 点域 55
- .MinEU 点域 58, 170, 171
- MinEU 点域 59
- .MinRange 点域 179
- .MinRaw 点域 55
- 目录树视图 86

N

- .Name 点域 136

O

- .OnMsg 点域 62

P

- .Pen1-8 点域 187
- PrintHT() 函数 211
- 平方根缩放 75
- 平均 / 散点历史趋势 170
- 平均历史趋势 175

Q

- .Quality 点域 113
- .QualityLimit 点域 114
- .QualityLimitString 点域 115
- .QualityStatus 点域 115
- .QualityStatusString 点域 116
- .QualitySubstatus 点域 116
- .QualitySubstatusString 点域 117

趋势

- 均值 / 棒图 170
- 类型 170–171
- 历史 171–211
- 描述 169
- 平均 / 散点 170
- 实时 212–217
- 最小 / 最大历史 175
- 最小 / 最大历史趋势 170

R

- .RawValue 点域 57
- .Reference 点域 79
- .ReferenceComplete 点域 79

S

- .ScooterLockLeft 点域 189
- .ScooterLockRight 点域 190
- .ScooterPosLeft 点域 191
- .ScooterPosRight 点域 192
- SuiteLink
 - 更新标记数据的质量状态 113
 - 功能 67
 - 描述 67
 - 设置“访问名”协议 122
 - 诊断通讯问题 67

SuperTag

- 编辑 145
- 编辑模板 145
- 创建模板 143
- 创建实例 146
- 从模板创建实例 146–147
- 定义模板实例 146
- 定义实例 146
- 复制实例 147
- 将标记添加到模板 143–144
- 将标记添加到实例 148
- 类型 144
- 描述 19, 141
- 模板 141
- 模板结构 142
- 其它创建方法 149
- 删除 145
- 删除模板 145

- 实例 141
- 使用远程标记引用 150
- 使用“ArchestrA 批量导入”实用程序进行移植 150
- 使用“标记名字典”创建实例 146
- 引用成员 150
- 有效语法 150
- 远程引用语法 88
- 实时趋势
 - 创建 213
 - 描述 171
 - 配置 214
 - 配置视觉外观 216–217
 - 设置时间跨度与更新速率 215
- 数值时间质量
 - 描述 67
 - 使用点域查看 112–113

T

- .TagID 点域 188
- .TimeDate 点域 105
- .TimeDateString 点域 105
- .TimeDateTime 点域 106
- .TimeDay 点域 107
- .TimeHour 点域 107
- .TimeMinute 点域 108
- .TimeMonth 点域 108
- .TimeMsec 点域 109
- .TimeSecond 点域 109
- .TimeTime 点域 110
- .TimeTimeString 点域 110
- .TimeYear 点域 111
- 通讯协议
 - DDE 67
 - FastDDE 67
 - NetDDE 67
 - SuiteLink 67
 - 消息交换 93

U

- .UpdateCount 点域 182
- .UpdateInProgress 点域 183
- .UpdateTrend 点域 184

V

- .Value 点域 61
- VTQ, 请参阅“数值时间质量”

W

- WindowViewer
 - 记录频率 33
 - IOStatus 主题 130
 - 描述 14
 - 使用 Application Server 93
 - 使用缺省设置初始化“访问名” 119
 - 在运行时初始化与重置 I/O 连接 118
 - 重新初始化所选的“访问名” 118–121
 - 重新初始化所有的“访问名” 118
- Wonderware Historian
 - 配置历史记录连接 220
 - 显示远程标记值 219
- 文档惯例 11
- 无效的标记句柄 240

X

- 系统标记
 - \$Date 系统标记 40
 - \$datestring 系统标记 40
 - \$DateTime 系统标记 40
 - \$Day 系统标记 40
 - \$HistoricalLogging 167
 - \$HistoricalLogging 系统标记 40
 - \$Hour 系统标记 40
 - \$InactivityTimeout 系统标记 41
 - \$InactivityWarning 系统标记 41
 - \$Language 系统标记 41
 - \$LogicRunning 系统标记 41
 - \$Minute 系统标记 41
 - \$Month 系统标记 41
 - \$Msec 系统标记 41
 - \$NewAlarm 系统标记 41
 - \$ObjHor 系统标记 42
 - \$ObjVer 系统标记 42
 - \$Operator 系统标记 42
 - \$OperatorDomain 系统标记 42
 - \$OperatorDomainEntered 系统标记 42
 - \$OperatorName 系统标记 42
 - \$PasswordEntered 系统标记 43

- \$Second 系统标记 43
- \$StartDdeConversations 系统标记 43
- \$System 系统标记 43
- \$Time 系统标记 43
- \$TimeString 系统标记 43
- \$VerifiedUserName 系统标记 43
- \$Year 系统标记 43
- 许可证计数 240

线性缩放 75

消息交换 93

许可证

- 标记句柄 239
- 理论最大标记计数 153
- 无效的标记句柄 240
- 系统标记计数 240
- 远程引用标记计数 242
- 占位符标记计数 240

Y

远程标记计数 153

远程历史供应器

- 定义为标记源 222
- 将标记指定给笔 221
- 使用脚本将标记指定给笔 223

远程引用

- 访问 I/O 数据 87–88
- 描述 24
- 用于间接标记 138–139
- 在运行时重定向 89–92
- 指定 SuperTag 150
- 指定间接标记 138–139
- 转换标记 83–86

Z

质量点域

- 更新状态 113
- 描述 112
- 数据格式 112

指示器

- 描述 171
- 使用 .ScooterLockRight 点域锁定 190
- 使用 .UpDateCount 点域检索笔的值 182

值死区

- 描述 30
- 设置 32

自动编号的标记名 29

组变量标记 20

最小 / 最大历史趋势 170, 175