

# InTouch® HMI 应用程序管理与扩展 指南

Invensys Systems, Inc.

修订版 A

上次修订日期：2007 年 8 月 6 日



## 版权声明

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

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

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

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

<http://www.wonderware.com>

对产品文档如有任何意见或建议，请发送电子邮件到 [productdocs@wonderware.com](mailto: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 A<sup>2</sup>、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
<b>第 1 章 管理 InTouch 应用程序.....</b>	<b>13</b>
关于 InTouch 应用程序管理器.....	14
托管的 InTouch 应用程序.....	14
启动应用程序管理器.....	15
从应用程序管理器中启动 ArchestrA IDE .....	16
创建 InTouch 应用程序 .....	17
在 WindowMaker 与 WindowViewer 中打开应用程序.....	19
修改 InTouch 应用程序 .....	20
从应用程序管理器中删除 InTouch 应用程序 .....	20
自定义应用程序管理器窗口.....	21
查找 InTouch 应用程序 .....	21
将应用程序发布到远程节点.....	22
发布的文件的内容 .....	23
发布独立的 InTouch 应用程序.....	24
将较早版本的 InTouch 应用程序迁移到当前版本.....	26
转换旧的报警显示 .....	27

<b>第 2 章 分发应用程序 .....</b>	<b>29</b>
支持的 InTouch 架构 .....	30
单机架构 .....	30
基于客户端的架构 .....	30
基于服务器的架构 .....	31
网络应用程序开发 (NAD) .....	32
网络化应用程序的规划注意事项 .....	33
网络化应用程序的 I/O 数据访问 .....	33
使用全局 I/O 地址 .....	33
使用本地 I/O 地址 .....	34
Wonderware SuiteLink .....	34
访问共享的文件 .....	35
使用文件数据的全局地址 .....	35
使用文件数据的本地地址 .....	36
通过 UNC 访问共享的文件 .....	36
在分布式环境中记录数据 .....	37
配置远程历史供应器 .....	39
动态配置远程历史供应器 .....	40
配置分布式历史记录 .....	41
特殊网络的注意事项 .....	42
针对 NAD 配置 InTouch 应用程序 .....	42
执行 NAD 自动更新 .....	44
执行 NAD 手工更新 .....	44
\$ApplicationChanged 系统标记 .....	44
\$ApplicationVersion 系统标记 .....	45
RestartWindowViewer() 函数 .....	46
ReloadWindowViewer() 函数 .....	46
应用程序编辑锁定 .....	47
在 NAD 更新期间对应用程序更改 .....	47
在运行时调整应用程序分辨率 .....	48
使用终端服务 .....	49
终端服务器应用程序的规划注意事项 .....	50
终端服务环境中的报警 .....	50
终端服务环境中的安全性 .....	51
终端服务环境中的 I/O .....	51
终端服务环境中的脚本执行 .....	52
终端服务环境中的其它限制 .....	52
使用脚本检索关于 InTouch 客户端会话的信息 .....	53
TseGetClientId() 函数 .....	53
TseGetClientNodeName() 函数 .....	53
TseQueryRunningOnConsole() 函数 .....	53
TseQueryRunningOnClient() 函数 .....	54

### 第 3 章 管理 InTouch 服务 ..... 55

作为服务运行 WindowViewer .....	56
将 WindowViewer 配置成作为服务运行 .....	56
手工启动服务 .....	57
停止服务 .....	58
配置 InTouch 服务的用户帐户 .....	58
InTouch 服务疑难排解 .....	59
查看服务的错误消息 .....	59
服务用户帐户问题的疑难排解 .....	60
停用提示的 I/O 项目 .....	60
InTouch 服务的注册表项 .....	61

### 第 4 章 导出与导入标记定义、窗口以及脚本 ..... 63

导出标记定义 .....	63
查看导出的标记定义 .....	65
导入标记定义 .....	65
标记名字典导入文件格式 .....	66
创建导入文件模板 .....	67
设置字典导入文件的操作模式 .....	68
:MODE=REPLACE .....	68
:MODE=UPDATE .....	69
:MODE=ASK .....	70
:MODE=IGNORE .....	70
:MODE=TERMINATE .....	70
:MODE=TEST .....	70
设置访问名与报警组 .....	71
:IOAccess 关键字属性 .....	71
:AlarmGroup 关键字属性 .....	72
定义标记类型关键字与属性 .....	74
标记关键字属性 .....	75
:MemoryDisc 关键字属性 .....	81
:IODisc 关键字属性 .....	82
:MemoryInt 关键字属性 .....	82
:IOInt 关键字属性 .....	85
:MemoryReal 关键字属性 .....	87
:IOReal 关键字属性 .....	89
:MemoryMsg 关键字属性 .....	91
:IOMsg 关键字属性 .....	92
:GroupVar 关键字属性 .....	93
:HistoryTrend 关键字属性 .....	93

:TagID 关键字属性 .....	93
:IndirectDisc 关键字属性 .....	94
:IndirectAnalog 关键字属性 .....	94
:IndirectMsg 关键字属性 .....	95
在导入文件中使用空字符串 .....	95
使用字段的缺省值 .....	96
创建 SuperTag 实例 .....	96
使用 DBLoad 导入标记定义 .....	97
导入窗口 .....	98
转换导入窗口的占位符标记 .....	100
导出窗口 .....	101
导入脚本 .....	103
转换导入的脚本中的占位符标记 .....	104
导入的窗口与脚本中的标记占位符 .....	106

## 第 5 章 保护 InTouch 安全 ..... 109

InTouch 安全性功能 .....	110
配置不活动超时 .....	110
\$InactivityTimeout 系统标记 .....	112
\$InactivityWarning 系统标记 .....	112
锁定系统键 .....	113
启用 DisableKeys() 函数 .....	115
在运行时隐藏菜单项 .....	115
基于身份验证与授权的安全性 .....	118
比较身份验证与授权 .....	118
不同的身份验证安全模式 .....	118
使用基于 InTouch 的安全性 .....	118
使用基于操作系统的安全性 .....	119
使用基于 ArchestrA 的安全性 .....	119
管理用户并设置授权级别 .....	120
配置 InTouch 安全性身份验证与授权 .....	120
在运行时更改 InTouch 操作员口令 .....	121
设置基于操作系统的身份验证与授权 .....	122
设置基于 ArchestrA 的安全性 .....	123
AddPermission() 函数 .....	123
ChangePassword() 函数 .....	124
\$AccessLevel 系统标记 .....	124
\$ChangePassword 系统标记 .....	125
\$ConfigureUsers 系统标记 .....	126

登录与注销 .....	127
登录到采用 InTouch 安全机制的应用程序 .....	127
登录到采用操作系统安全机制的应用程序 .....	127
登录到采用 ArchestrA 安全机制的应用程序 .....	128
从 InTouch 应用程序中注销 .....	128
创建自定义的登录窗口 .....	128
PostLogonDialog() 函数 .....	129
LogonCurrentUser() 函数 .....	129
Logoff() 函数 .....	130
AttemptInvisibleLogon() 函数 .....	130
\$OperatorEntered 系统标记 .....	131
\$PasswordEntered 系统标记 .....	132
\$OperatorDomainEntered 系统标记 .....	132
基于操作员或访问级别启用与禁用功能 .....	133
InvisibleVerifyCredentials() 函数 .....	133
检索当前登录的操作员的有关信息 .....	134
GetAccountStatus() 函数 .....	134
IsAssignedRole() 函数 .....	135
QueryGroupMembership() 函数 .....	136
\$OperatorName 系统标记 .....	136
\$OperatorDomain 系统标记 .....	137
\$Operator 系统标记 .....	137
\$VerifiedUserName 系统标记 .....	137
安全系统标记与函数的摘要 .....	138

## 第 6 章 在运行时切换语言 ..... 141

为运行时语言切换配置语言 .....	142
为配置的语言更改字体设置 .....	143
添加运行时语言切换功能 .....	143
SwitchDisplayLanguage() 函数 .....	146
\$Language 系统标记 .....	146
导出应用程序文本进行脱机翻译 .....	146
将文本导出到现有的字典文件 .....	148
翻译导出的字典文件 .....	148
导入翻译的字典文件 .....	150
导出报警注释进行翻译 .....	151
理解两个英文字符的应用程序 ID .....	151
导出报警注释 .....	151

导出到现有的报警注释文件 .....	152
编辑字典文件 .....	153
导入翻译的报警注释 .....	155
在运行时测试语言切换功能 .....	156
将本地化版的文件分发到网络应用程序开发客户端 .....	157

## 第 7 章 在运行时查看应用程序 .....

关于 WindowViewer .....	159
自定义运行时环境 .....	159
配置常规 WindowViewer 属性 .....	159
配置 WindowViewer 的视觉特征 .....	162
使用 WindowViewer 窗口 .....	163
常用对话框功能 .....	163
从 WindowViewer 中打开窗口 .....	164
从 WindowViewer 中关闭窗口 .....	165
从 WindowViewer 转换到 WindowMaker .....	165
InTouchView .....	166
创建新的 InTouchView 应用程序 .....	167
在 InTouch 与 InTouchView 之间转换应用程序 .....	168
将 InTouchView 应用程序转换为 InTouch 应用程序 .....	168
将 InTouch 应用程序转换为 InTouchView 应用程序 .....	169
InTouchView 许可证 .....	170

## 第 8 章 设置多监视器系统 .....

多监视器配置 .....	172
单显卡配置 .....	172
单显卡配置的特征 .....	172
单显卡驱动程序的特征 .....	173
多显卡配置 .....	173
多显卡配置的特征 .....	173
多显卡驱动程序的特征 .....	174
规划多监视器应用程序 .....	175
选择多监视器显卡 .....	175
确定应用程序屏幕分辨率 .....	175
确定显示应用程序的监视器数 .....	175
确定应用程序窗口的位置 .....	176
窗口显示在强制的位置 .....	176
可以手工移动窗口 .....	177



窗口根据环境自动放置 .....	177
开发多监视器 InTouch 应用程序 .....	177
配置多监视器参数 .....	177
配置屏幕分辨率转换.....	178
部署应用程序与验证多监视器设置 .....	179
在运行时验证多监视器支持 .....	179

## 第 9 章 在 Tablet PC 上使用 InTouch..... 181

注解可视化屏幕与将它当作电子邮件消息发送 .....	182
给窗口添加注解 .....	182
选择、复制及删除窗口注解 .....	183
保存、打印及使用电子邮件发送注解的窗口 .....	183
AnnotateLayout() 函数 .....	184
更改屏幕方向 .....	184

## 附录 A 从 INTOUCH.ini 文件中自定义应用程序设置 ..... 187

自定义的 INTOUCH.ini 参数 .....	187
设置自定义的记录属性 .....	188
设置记录频率 .....	188
记录远程引用的标记 .....	189
禁用 WindowMaker 快捷菜单 .....	189
设置自定义的 WindowViewer 属性 .....	189
添加脚本循环计时器.....	190
根据不同屏幕分辨率缩放 InTouch 窗口 .....	190
设置打印等待时段的长度 .....	190
记录报警注释 .....	190
设置 16 笔趋势的绘制模式 .....	191
调整数字小键盘的大小 .....	191
调整模拟与字符串用户输入链接的输入字段的大小 ....	191

## 索引..... 193



# 欢迎

本文介绍如何管理与扩展 InTouch 应用程序。本文涉及如何创建应用程序，如何在 WindowMaker 与 WindowViewer 中打开它们，如何将它们从一台计算机移到另一台计算机；以及如何在网络中发布它们。您也可以扩展应用程序以使之在特定的环境中运行，例如，在多个显示器上以及在 Tablet PC 上。

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。如需有关完成这些任务的帮助，请参阅 Microsoft 文档。

## 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。
<b>粗体</b>	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

## 管理 InTouch 应用程序

管理 InTouch 应用程序时，您可以：

- 创建或删除 InTouch 应用程序。请参阅第 17 页的“创建 InTouch 应用程序”与第 20 页的“从应用程序管理器中删除 InTouch 应用程序”。
- 在 WindowMaker 或 WindowViewer 中打开应用程序。请参阅第 19 页的“在 WindowMaker 与 WindowViewer 中打开应用程序”。
- 搜索应用程序。请参阅第 21 页的“查找 InTouch 应用程序”。
- 将应用程序移到不同的计算机上。请参阅第 22 页的“将应用程序发布到远程节点”。
- 将应用程序发布到多个计算机上。请参阅第 2 章“分发应用程序”。
- 管理 InTouch 服务。请参阅第 3 章“管理 InTouch 服务”。
- 导入或导出标记定义、窗口及脚本。请参阅第 4 章“导出与导入标记定义、窗口以及脚本”。
- 配置安全性。请参阅第 5 章“保护 InTouch 安全”。

您可以按照以下方式扩展应用程序：

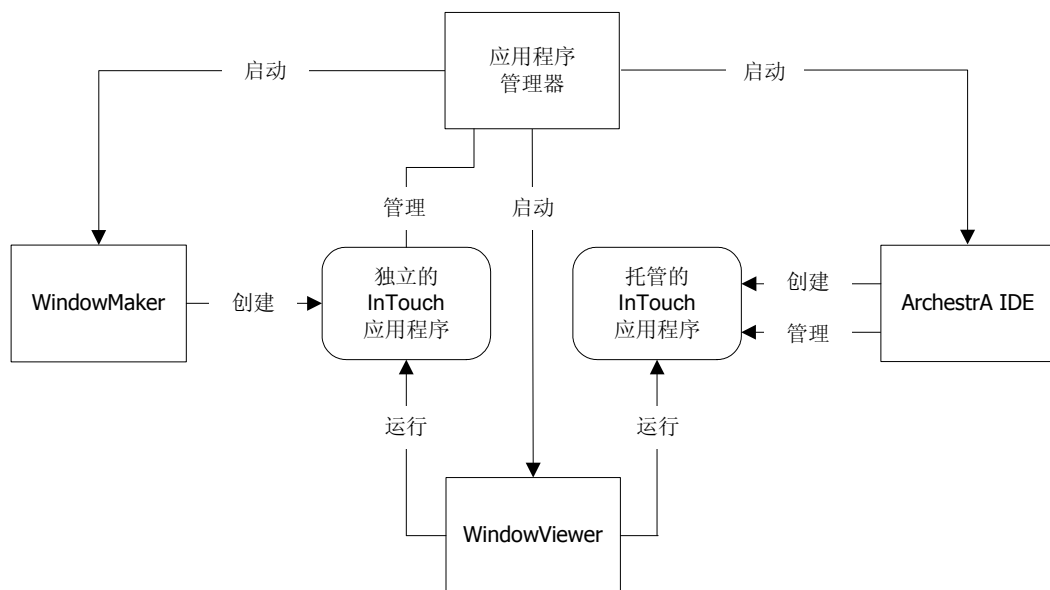
- 将文本字符串与报警注释翻译成不同的语言。请参阅第 6 章“在运行时切换语言”。
- 将应用程序集成到 ArchestrA 中。请参阅第 14 页的“托管的 InTouch 应用程序”与第 7 章“在运行时查看应用程序”。

- 将应用程序显示到多个监视器上。请参阅第 8 章 “设置多监视器系统”。
- 在 Tablet PC 上使用应用程序。请参阅第 9 章 “在 Tablet PC 上使用 InTouch”。

## 关于 InTouch 应用程序管理器

您可以使用 “InTouch 应用程序管理器” 管理大多数全局性任务，如创建、删除以及修改 InTouch 应用程序。

“应用程序管理器” 显示当前的 InTouch 应用程序列表。您可以从该列表中选择要在 WindowMaker 或 WindowViewer 中打开的应用程序。

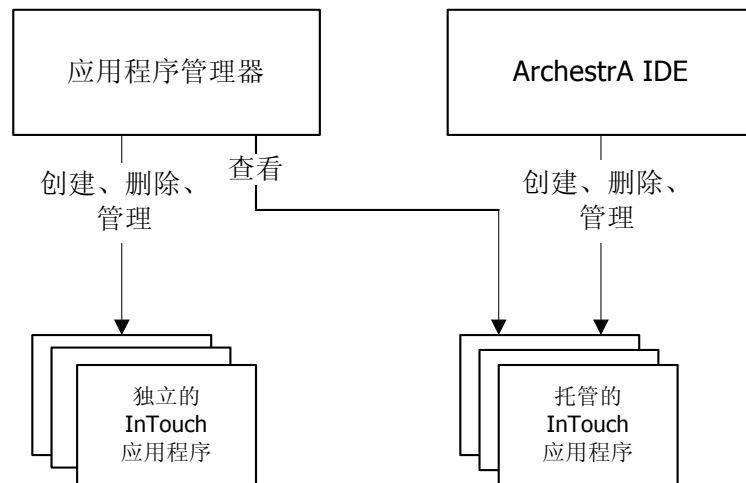


## 托管的 InTouch 应用程序

如果 “Archestra 集成开发环境” (IDE) 与 InTouch HMI 安装在相同的计算机上，则可以使用它来管理 InTouch 应用程序。

这些应用程序称为托管的 InTouch 应用程序。与完全由 “InTouch 应用程序管理器” 管理的独立的 InTouch 应用程序不同，托管的应用程序更多地集成到 Archestra 环境，并支持高级图形。

您可以从“应用程序管理器”中启动 ArchestrA IDE。托管的 InTouch 应用程序作为“被托管对象”出现在“InTouch 应用程序管理器”中，并且仅能通过从 IDE 中启动 WindowMaker 进行编辑。



如需有关详细信息，请参阅 *InTouch® HMI 与 ArchestrA® 集成指南* 中的第 1 章“关于 InTouch HMI 与 ArchestrA 集成”。

## 启动应用程序管理器

您可以从开始菜单，或使用计算机桌面上的快捷方式来启动“应用程序管理器”。

### 第一次启动“应用程序管理器”

- 1 在任务栏上，单击**开始**，指向**程序**，指向 **Wonderware**，然后单击 InTouch。此时出现“欢迎使用 InTouch 应用程序管理器”向导。



- 2 单击**下一步**。此时出现下一页。

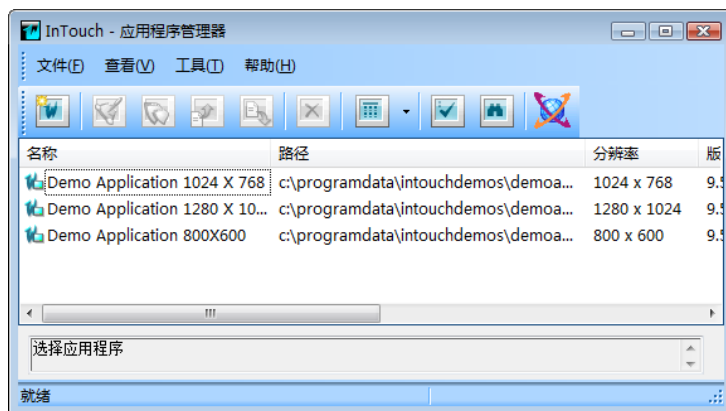


此时显示缺省的应用程序文件夹。例如，C:\Documents and Settings\用户名\My Documents\我的 InTouch 应用程序。对于运行 Microsoft Windows Vista 的计算机，缺省的应用程序文件夹是 C:\用户\用户名\Documents\我的 InTouch 应用程序。

- 3 要选择不同的文件夹或创建新的文件夹，请单击**浏览**，选择或创建文件夹，然后单击**确定**。

InTouch 程序将缺省文件夹更改为指定的文件夹。此缺省文件夹应用于计算机上的所有 InTouch 用户。

- 4 单击**完成**。此时出现“应用程序管理器”。



窗口显示您使用“应用程序管理器”在计算机上创建或找到的 InTouch 应用程序。**版本**列显示上一次用于保存应用程序的 InTouch 版本。

## 从应用程序管理器中启动 ArchestraA IDE

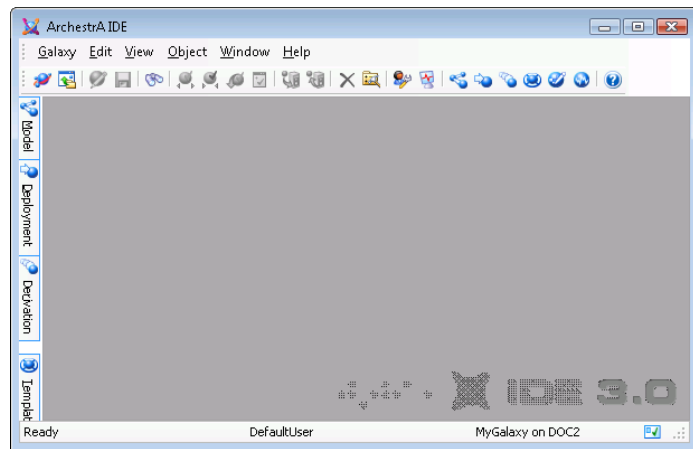
您可以切换到“ArchestraA 集成开发环境”(IDE)来创建或编辑托管的 InTouch 应用程序。



**备注** Application Server Bootstrap 与 IDE 必须与 InTouch HMI 安装在相同的计算机上，才能从“应用程序管理器”中启动 ArchestrA IDE。

要从“应用程序管理器”中启动 ArchestrA IDE

- 1 启动“应用程序管理器”。
- 2 在文件菜单上，单击 **ArchestrA IDE**。此时出现 Application Server 的 **Connect to Galaxy**（连接到 Galaxy）对话框。
- 3 连接到现有的 Galaxy 或创建新的 Galaxy。在连接到 Galaxy 之后，出现 **ArchestrA IDE** 对话框。



- 4 创建或编辑指定为托管的 InTouch 应用程序的对象。

## 创建 InTouch 应用程序

您可以使用“应用程序管理器”创建新的 InTouch 应用程序。应用程序路径不得超过 114 个字符，包括网络驱动器盘符、冒号及所有的反斜杠。如果超过限制，则无法在 WindowMaker 中打开应用程序。

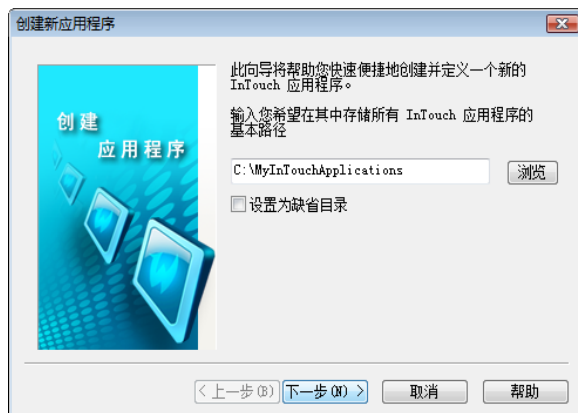
INTOUCH.ini 文件是在您创建应用程序时创建的。INTOUCH.ini 文件包含应用程序的缺省配置设置。配置应用程序时，新设置会写入 INTOUCH.ini 文件。

创建应用程序之后，您可以将现有的 INTOUCH.ini 文件复制到应用程序文件夹。这样，您就不必在每次创建新的应用程序时配置自定义的 InTouch 参数。

### 要创建新的应用程序



- 1 在文件菜单上，单击**新建**。此时出现**创建新应用程序**向导。



- 2 配置应用程序文件夹。如果指定的文件夹不存在，程序会提示您创建它。
  - 单击**浏览**以指定一个文件夹，而不使用缺省值。
  - 选择**设置为缺省目录**复选框，使指定的文件夹成为缺省值。
- 3 单击**下一步**。此时出现向导的下一页。



- 4 输入应用程序文件夹名。

5 单击下一步。此时出现向导的下一页。



6 配置应用程序详细信息。

- 在**名称**框中，为应用程序输入唯一的名称。
- 在**描述**框中，输入可选的描述，最多可以包含 255 个字符。
- 选择 **InTouchView 应用程序** 复选框，以创建 InTouchView 应用程序。如需有关 InTouchView 的详细信息，请参阅第 159 页的“在运行时查看应用程序”。

7 单击**完成**。

## 在 WindowMaker 与 WindowViewer 中打开应用程序

在 WindowViewer 中打开新的应用程序之前，必须先在 WindowMaker 中打开它。

### 要在 WindowMaker 中打开应用程序

- 1 在“应用程序管理器”窗口中选择应用程序。
- 2 在**文件**菜单上，单击 **WindowMaker**。



**提示** 您也可以双击应用程序以便在 WindowMaker 中打开它。

### 要在 WindowViewer 中打开应用程序

- 1 在“应用程序管理器”窗口中选择应用程序。
- 2 在**文件**菜单上，单击 **WindowViewer**。



## 修改 InTouch 应用程序

您可以重命名应用程序以及更改应用程序属性。

### 要重命名应用程序

- 1 选择列表中的应用程序。
- 2 在文件菜单上，单击**重命名**。

### 要修改应用程序属性

- 1 选择列表中的应用程序。
- 2 在文件菜单上，单击**属性**。此时出现**属性**对话框。



- 3 配置应用程序属性。
  - 在**名称**框中，为应用程序输入新的名称。
  - 在窗口中，为应用程序输入另一段描述。
- 4 单击**确定**。

## 从应用程序管理器中删除 InTouch 应用程序

从“应用程序管理器”中删除应用程序时，应用程序文件保留在计算机上。

### 要删除应用程序

- 1 选择列表中的应用程序。
- 2 在文件菜单上，单击**删除**。
- 3 在出现的消息中，单击**是**。



## 自定义应用程序管理器窗口

您可以隐藏工具栏与状态栏。您也可以更改在“应用程序管理器”窗口中列出应用程序的方式。应用程序可以显示为大图标、小图标、简要列表或详细列表。

### 要隐藏工具栏

- 在**查看**菜单上，单击**工具栏**使之不显示选中标记。

### 要隐藏状态栏

- 在**查看**菜单上，单击**状态栏**使之不显示选中标记。



### 要更改应用程序列表的视图

- 在**查看**菜单上，单击适当的命令。

## 查找 InTouch 应用程序

您可以搜索现有的 InTouch 应用程序。“应用程序管理器”显示找到的任何应用程序。

如果完整路径超过 114 个字符（包括网络驱动器盘符、冒号及所有的反斜杠），则无法在 WindowMaker 中打开应用程序。如果应用程序超过此字符限制，请给路径中的各个文件夹重命名，或是将应用程序移到另一个位置。



### 要查找应用程序

- 在**工具**菜单上，单击**查找应用程序**。此时出现**浏览文件夹**对话框。



- 选择要在其中搜索应用程序的文件夹。
- 单击**确定**。

## 将应用程序发布到远程节点

通过使用“应用程序发布器”，可以创建压缩的自解压数据包文件，包含在另一台计算机上安装 InTouch 应用程序所需的所有相关文件与安装过程。

您可以使用“应用程序发布器”发布独立的 InTouch 应用程序。您可以使用 ArchestrA IDE 发布托管的 InTouch 应用程序。

有两个选项可供发布应用程序：

- **仅限运行时。**在仅限运行时的软件包中，包含运行应用程序而不是编辑应用程序所需的文件。
- **设计时与运行时。**在设计时与运行时的软件包中，包含编辑与运行应用程序所需的所有文件。由于某些运行时文件（如编译的 \*.www 文件）可以从设计时文件重新创建，因此它们被排除在外。

您可以将已发布的应用程序发到 Web 服务器上，以便可以下载并安装它们。对于发布的应用程序，会显示以下软件包信息：

- 软件包描述
- 发布者姓名
- 发布的文件名（可执行文件）
- 应用程序分辨率

例如：

描述	Dairy Processing Application
发布者	Navin Johnson
文件名	Dairy.exe / Video Resolution... (1024x768)
描述	Dairy Processing Application
发布者	Navin Johnson
文件名	Dairy_2.exe / Video Resolution... (800x600)

## 发布的文件的内容

下表列出对于发布的所有独立的 InTouch 应用程序，都会包含的文件夹、文件以及会排除的文件。

包含的文件夹	包含的文件	排除的文件
主应用程序文件夹	所有	备份文件。这些文件有 .?bk 文件扩展名。
	使用这些扩展名的文件： .win, .dat, .lgh, .idx, .log, .fsm, .stg, .\$\$\$	不在“特殊目录”列表中的子文件夹
	retentiv.x retentiv.d retentiv.a retentiv..s（两个点） retentiv.h wm.ini db.ini linkdefs.ini tbox.ini group.def itocx.cfg	appedit.lok 文件，表示在 WindowMaker 中打开了应用程序。
	名称为 SSD_*.xml 形式的任何文件。	编译的窗口文件，使用 .www 文件扩展名。
Dictionary 子文件夹，用于运行时语言切换功能	使用 .xml 扩展名的所有文件。	
Symbol 子文件夹	所有文件与子文件夹。	
	wiz.ini 文件，如果安装了向导。	
	向导可执行文件的副本。	
	.dll 文件、 .wdo 文件、 .wdf 文件	

对于仅限运行时的应用程序，会排除文件名为 SSD\_\*.xml 的所有文件。

## 发布独立的 InTouch 应用程序

使用“应用程序发布器”发布独立的 InTouch 应用程序。

如果希望发布的应用程序在特定的屏幕分辨率下运行，在发布之前，请将原始应用程序设置成该分辨率。

要发布托管的 InTouch 应用程序，请使用 ArchestrA IDE。

### 要发布 InTouch 应用程序

- 1 启动“应用程序发布器”。
  - a 打开 WindowMaker。
  - b 显示“经典视图”并展开工具窗格。
  - c 展开应用程序。
  - d 双击 **Application Publisher**。

此时出现 InTouch 应用程序发布器 - 第一步 / 共四步对话框。





- 2 单击下一步。此时出现 InTouch 应用程序发布器 - 第二步 / 共四步对话框。



- 3 配置软件包详细信息。
- 在**输入作者名**框中，输入应用程序联系人的姓名。姓名限制为 256 个字符。
  - 在**描述**框中，输入应用程序的描述。限制为 256 个字符。
  - 在**软件包名**框中，为发布的应用程序软件包输入唯一的名称。限制为 32 个字符。如果使用现有软件包的名称，则会覆盖掉现有的软件包。
- 4 单击下一步。此时出现 InTouch 应用程序发布器 - 第三步 / 共四步对话框。



- 5 配置关于发布的详细信息。
- 在文本框中，输入 InTouch 应用程序文件夹的路径。缺省路径是 WindowMaker 应用程序文件夹。
  - 选择**仅限运行时**复选框，以便在发布的文件中排除 WindowMaker 开发文件。

- 6 单击下一步。此时出现 InTouch 应用程序发布器 - 第四步 / 共四步对话框。



- 7 配置应用程序软件包可执行文件的详细信息。
- 在第一个框中，验证可执行文件名是否正确。缺省条件下，可执行文件名与软件包名相同。
  - 在第二个框中，输入要保存可执行文件的文件夹的路径；或是单击浏览以选择不同的文件夹。缺省条件下，可执行文件保存在当前的临时文件夹中。
- 8 单击完成。

## 将较早版本的 InTouch 应用程序迁移到当前版本

您可以将旧版 InTouch HMI 开发的应用程序迁移到当前版本。试图使用 WindowMaker 或 WindowViewer 打开旧版的应用程序时，会显示两个对话框：

- 第一个对话框要求确认是否要将旧版应用程序迁移到当前版本的 InTouch HMI。
- 第二个对话框要求提供信息，以便在将旧版应用程序迁移到当前版本的 InTouch HMI 之前，先创建一个备份副本。

在确认要将旧版应用程序迁移到当前版本的 InTouch HMI 之后，可以看到**备份配置**对话框。您必须指定要用于创建备份副本的文件夹，以及是否要从备份中排除任何文件。



要更改缺省备份路径（< 应用程序目录 >\Bak），请清除**使用缺省备份路径**框。然后在**备份路径**框中，输入要保存备份的文件夹的路径。如果该文件夹不存在，则必须创建它，然后再创建备份。

在**忽略文件**框中，可以指定要从备份中排除的任何文件。缺省条件下，应用程序目录中的所有文件都会备份。输入每个文件名，并使用分号 (;) 隔开。或者使用标准通配符（'\*' 与 '?'），以根据名称中的公共字符来排除一组文件。

## 转换旧的报警显示

在 WindowViewer 中打开使用 InTouch 7.11 之前的版本构建的应用程序时，会出现一个对话框，提示您运行 WindowMaker 以转换该应用程序。如果继续转换，则所有的“标准报警对象”都会转换为包含缺省值的“分布式报警对象”。颜色、字体、表达式及报警查询等设置不会保留下来。



## 第 2 章

# 分发应用程序

通常，分布式应用程序有一个中央开发工作站、中央数据存储区以及多个客户端工作站。您可以使用 InTouch 的“网络应用程序开发”（Network Application Development，简称 NAD）来构建与维护分布式应用程序。NAD 使得多个客户端工作站可以维护单个应用程序的一份副本，而不致限制该应用程序的开发。在应用程序发生更改时，客户端工作站自动收到通知。

您可以创建单机、基于客户端以及基于服务器的 InTouch 应用程序。

您也可以使用 ArchestrA IDE 来管理与部署 InTouch 应用程序。如需有关配合使用 ArchestrA IDE 与 InTouch HMI 的详细信息，请参阅 *InTouch® HMI 与 ArchestrA® 集成指南* 中的第 1 章“关于 InTouch HMI 与 ArchestrA 集成”。

## 支持的 InTouch 架构

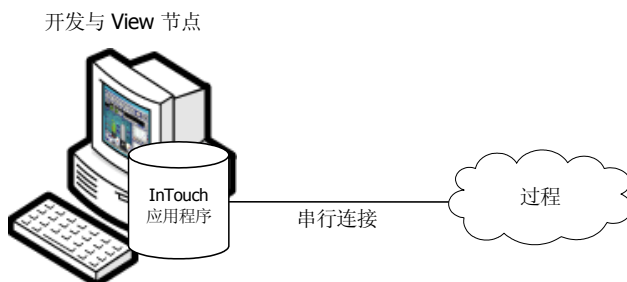
支持的 InTouch 网络架构包括：

- 单机
- 基于客户端
- 基于服务器
- NAD

### 单机架构

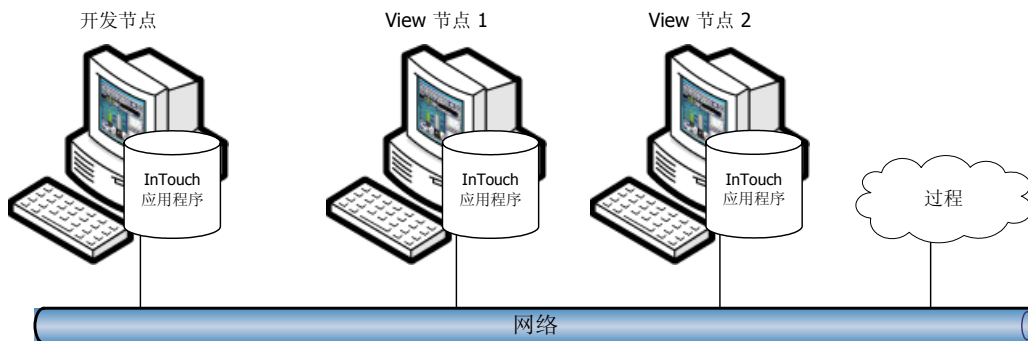
单机应用程序通常由一个充当主操作员界面的非联网计算机组成。此计算机直接连接（如使用串行电缆）到工业过程。

在这种架构中，您在一台计算机上开发 InTouch 应用程序。您可以将应用程序复制到另一台计算机上进行修改，然后再复制回原来的计算机上。



### 基于客户端的架构

在基于客户端的架构中，运行 WindowViewer 的每台计算机（View 节点）上，或网络服务器某个独特的位置上，都有某个 InTouch 应用程序的一份唯一的副本。在下例中，应用程序在开发节点上进行开发与测试，然后再复制到每个 View 节点上。



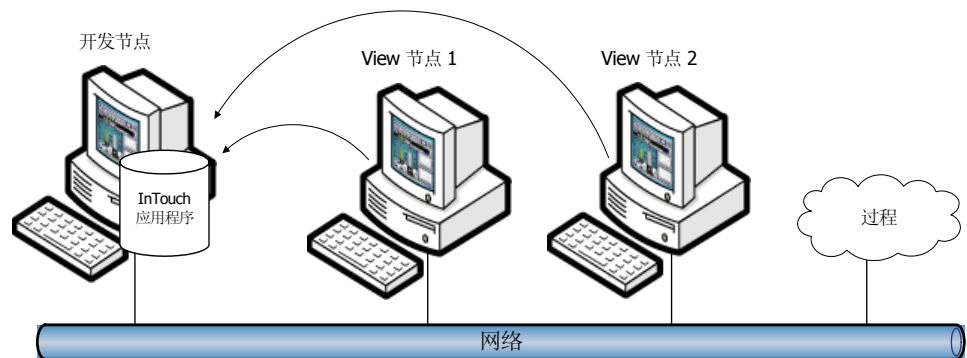
由于每个节点均可自足，因此具有内在的冗余特性；并且，在可以使用的 View 节点数上，不存在任何限制。

每个 View 节点都必须有完全相同的一份应用程序，并且对任何网络数据源（如 “I/O 服务器” 或 Wonderware Historian）都必须有相同的访问权限。不过，每个 View 节点都需要与共享的服务器维持一个单独的对话，这样就可能导致增加了网络负载。

您可以在开发节点上修改与测试应用程序，而不致影响正在运行的过程。不过，您必须将修改后的应用程序分发到各个 View 节点。您必须先在本机关闭每个 View 节点，将新的应用程序复制到其中，然后再重新启动。

## 基于服务器的架构

基于服务器的架构将一个公用的 InTouch 应用程序分发到多个 View 节点。在下图中，两个 View 节点从开发节点访问相同的 InTouch 应用程序。



对于每个 View 节点：

- 逻辑驱动器必须映射到开发节点的共享网络驱动器上。
- 共享的应用程序必须向 InTouch 程序注册。
- 计算机对应用程序引用的任何数据源都必须有完全相同的访问权限。通过使用脚本组合来确定节点名，并基于该名称来更改每个数据位置，可以有多种方法来定义数据源的位置。

在这种架构中，只需维护单个应用程序。应用程序发生变化并重新启动 WindowViewer 时，View 节点会自动更新。

这种架构的缺点包括：

- 应用程序的开发受到限制
- 如果开发节点崩溃，则没有冗余备份可以使用
- 所有节点都必须使用相同的屏幕分辨率

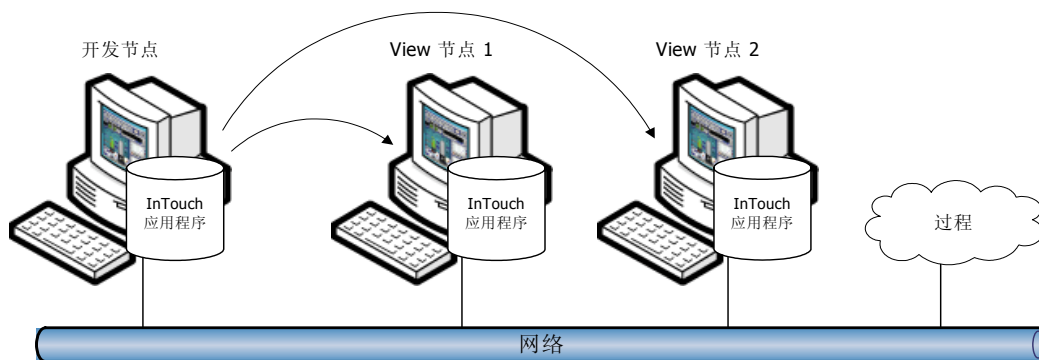
## 网络应用程序开发 (NAD)

在“网络应用程序开发”(NAD)架构中，在中央网络位置（通常是开发节点）上维护应用程序的主副本。每个 View 节点都会将应用程序复制到用户定义的位置并从那里运行它。

在通知客户端应用程序有更改（使用 WindowMaker 特别菜单上的**通知客户端**命令）时，会在应用程序目录中设置一个标帜，然后由 View 节点读取它。

您可以配置 View 节点处理应用程序更改的方式。这可以是忽略更改，也可以是自动关闭并重新启动 View 节点，从而重新加载主应用程序。

在下图中，两个 View 节点都从开发节点注册了主应用程序，但实际上还是在各自的本地计算机上运行它。



**备注** 如果将应用程序配置成将历史数据写入主应用程序节点的应用程序目录，则所有的 NAD 节点都会尝试将它们的历史数据写入主应用程序。为避免这种情况，在每个 NAD 节点上，将历史数据配置成写入本地目录，而不是主应用程序节点。

如果将一个复杂的大型应用程序分发到许多个节点，则在最初下载时，系统响应时间可能会明显变慢。不过对于更新，这种情况会得到优化。对于较慢的网络或串行连接，应用程序传输可能会是个问题。

此外请注意其它的一些网络限制，如过滤掉特定类型的网络通信与地址的路由器用户。



## 网络化应用程序的规划注意事项

在构建 InTouch 应用程序时，无论采取何种架构，都必须慎重考虑：

- 对 I/O 数据源的访问。
- 对共享文件的访问。
- 记录数据的位置。
- 任何特殊的网络要求。

### 网络化应用程序的 I/O 数据访问

InTouch HMI 使用“访问名”来引用实时 I/O 数据。每个“访问名”都等价于一个由节点名、应用程序以及主题组成的 I/O 地址。在分布式应用程序中，I/O 引用可以设置为网络“I/O 服务器”的全局地址，也可以是本地“I/O 服务器”的本地地址。

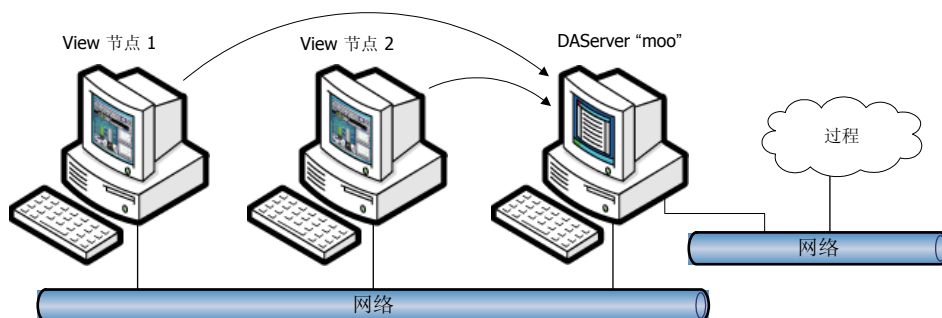
**备注** InTouchView 限制为单个 Galaxy “访问名”。您无法为 InTouchView 创建其它“访问名”。如需有关 InTouchView 的限制的详细信息，请参阅第 159 页的“在运行时查看应用程序”。

View 节点对数据源的访问权限必须和开发节点相同。

#### 使用全局 I/O 地址

I/O 数据的全局地址允许所有 View 节点共享一个基于网络的公共“I/O 服务器”。这样便没有必要使用多个“I/O 服务器”，但容错能力会下降，并可能导致整体性能降低。

在下图中，两个 View 节点在运行一份相同的应用程序。这两个 View 节点引用相同的 I/O 数据源。由于每个应用程序都使用该数据源的全限定 I/O 地址，因此所有的引用都指向同一个“I/O 服务器”。



您可以设置 InTouch 应用程序，使之能够使用第三方寻址惯例在“访问名”中确定另一个节点上存储的数据元素。“访问名”寻址惯例包括远程数据所在位置的节点名、应用程序名以及主题名。InTouch 应用程序通过结合使用“访问名”与项目名来获

取远程数据。如需有关为远程 “I/O 服务器” 定义 “访问名” 的详细信息，请参阅 *InTouch® HMI 数据管理指南* 中的第 5 章 “使用 I/O 进行数据访问”。

---

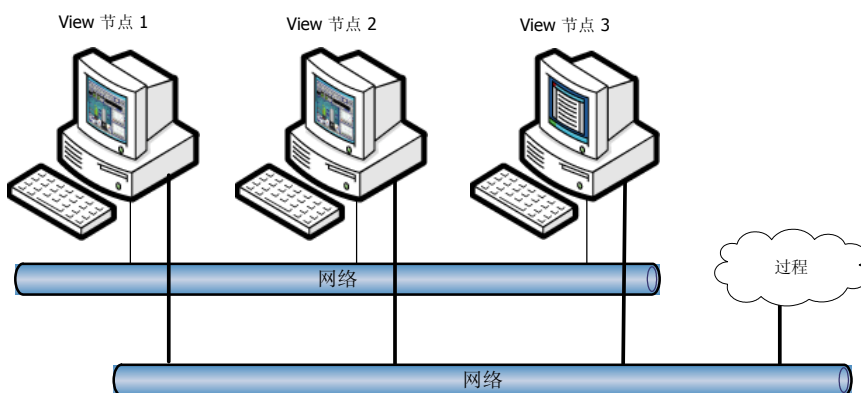
**备注** 在 WindowMaker 中创建 “访问名” 时，如果 “访问名” 使用 SuiteLink 协议，软件会阻止 “访问名” 去访问相同的节点、应用程序及主题。在运行时，请勿使用 `IOSetAccessName()` 函数将 “访问名” 重定向到重复的对象，否则重定向的 “访问名” 将无法正常工作。

---

### 使用本地 I/O 地址

每个 View 节点都有自己的 “I/O 服务器” 时，使用 I/O 数据的本地地址。这种架构提供容错操作，在网络中断时，每个 View 节点都能继续独立运行。

在下图中，两个 View 节点运行相同应用程序的不同副本，并引用各自的 I/O 数据源。由于每个应用程序都给该数据源使用本地 I/O 地址，因此每个引用都指向本地 “I/O 服务器”。



使用本地 “I/O 服务器” 会显著增加过程连接网络的负载。例如，由于必须单独处理每个节点的请求，所以三个节点产生的流量将三倍于单个节点所产生的流量。

如需有关为本地 “I/O 服务器” 定义 “访问名” 的详细信息，请参阅 *InTouch® HMI 数据管理指南* 中的第 5 章 “使用 I/O 进行数据访问”。

### Wonderware SuiteLink

Wonderware SuiteLink 通讯协议基于 TCP/IP 协议。由于提供以下功能，SuiteLink 通常用于一些高速工业应用：

- “数值时间质量” (Value Time Quality, 简称 VTQ)，其中的时间标签及质量指示符与传递给支持 VTQ 的客户端的所有数据值关联。InTouch HMI 是支持 VTQ 的客户端，其标记数据在传输时带有 VTQ 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，还是分布在大量的节点上，都可以在应用程序之间始终维持很高的数据吞吐量。

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

## 访问共享的文件

在分布式应用程序中，文件引用可以设置为：

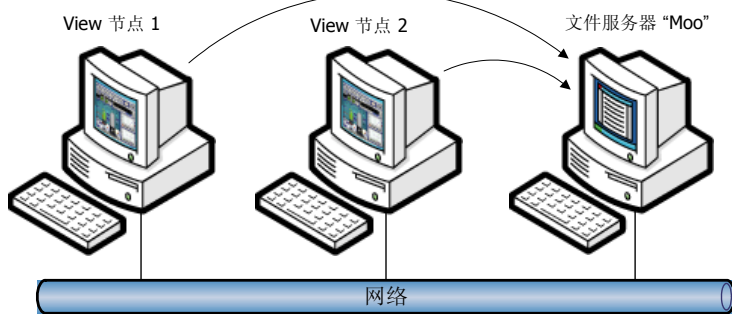
- 网络文件服务器的全局地址。
- 本地文件的本地地址。

View 节点对数据源的访问权限必须与开发节点相同。

### 使用文件数据的全局地址

您可以设置文件数据的全局地址，以便所有的 View 节点都共享一组基于公共网络的文件。这就给这些文件提供了单源维护，但它的容错性能要比本地副本差。

在下图中，两个 View 节点分别运行一份相同的应用程序，但引用相同的配方文件。由于每个应用程序都使用映射成该文件全限定网络路径的盘符，因此所有的引用都指向相同的文件。



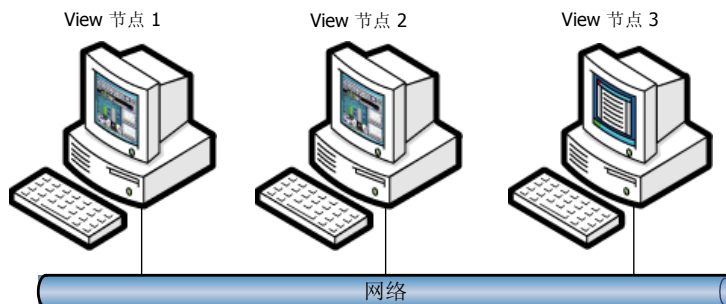
### 要设置共享的文件

- 1 将网络驱动器映射到包含所引用的文件的共享路径。例如，G:\Directory\Recipe.csv，其中 “G:\” 是引用 \\Moo\Share 的映射盘符。您必须在每个 View 节点上映射这个相同的驱动器。
- 2 在脚本中，引用共享的路径。例如：  

```
RecipeSelectRecipe( "G:\Directory\Recipe.csv" ,
    "review" , "RecipeName" );
```

### 使用文件数据的本地地址

每个 View 节点都有自己的一份文件副本时，可以使用文件数据的本地地址。在下图中，三个 View 节点分别运行一份相同的应用程序，并引用配方文件的本地副本。



在本例中，本地地址是：

C:\Directory\Recipe.csv

其中 “C:\” 是本地驱动器。

在脚本中，引用本地路径。例如：

```
RecipeSelectRecipe( “C:\Directory\Recipe.csv” , “review” ,  
    “RecipeName” );
```

这种架构有容错功能。不过，您必须将任何文件更改都复制到所有的 View 节点上。

任何文件访问权限都只能为 “只读”，并且不允许修改本地文件。

### 通过 UNC 访问共享的文件

在通常输入文件路径的任何地方（如应用程序目录项、配置项以及分布式报警），都可以使用 “通用命名惯例”（Universal Naming Convention，简称 UNC）。如果使用 UNC 名，则不需要创建映射的驱动器。

UNC 地址采用 \\Node\Share\Path 形式，其中：

- Node 是包含文件共享的计算机的名称。
- Share 是指定给该计算机上共享文件夹的逻辑名称。
- Path 是跟共享有关的该文件的普通路径。

**备注** 如果使用 Wonderware SuiteLink，则节点名的长度限制为 15 个字符。

在通过 UNC 访问文件之前，必须在要访问的计算机上创建文件共享。如需有关详细信息，请参阅 Windows 文档。

例如，假设有一台网络名是 "EngineRm" 的计算机，并且您已使用共享名 "Root" 共享了该计算机的根驱动器 "C:\\"。要设置到 "C:\IT\Apps\Boiler" 应用程序的 UNC 路径，必须使用以下 UNC：

\\EngineRm\Root\IT\Apps\Boiler

如果 "Boiler" 目录自身已作为 "Boiler" 共享，则 UNC 可以简化为：

\\EnginerRm\Boiler

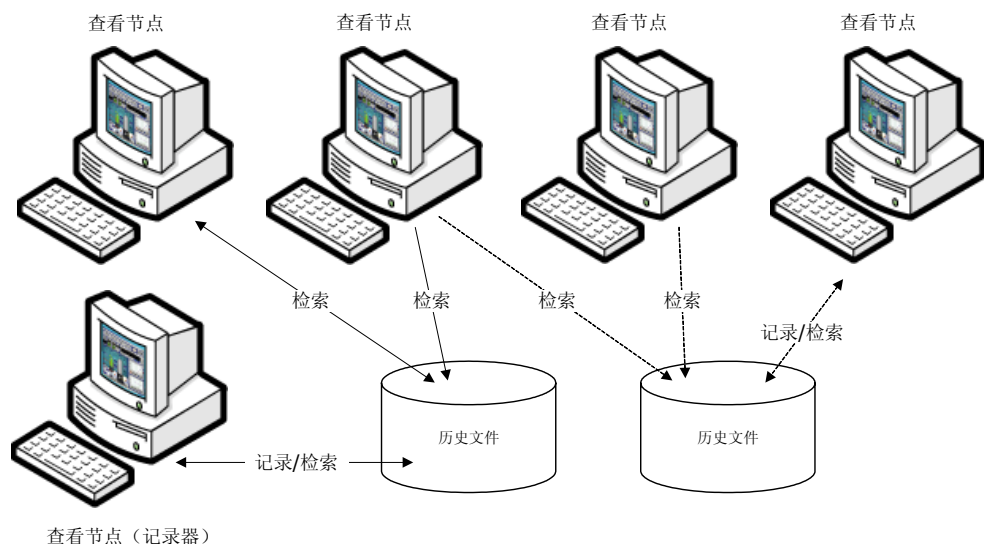
如果该共享是 PATH 环境变量指定的路径，则不要求提供路径。

**备注** 如果需要写入由 UNC 地址引用的文件，则该共享必须是可读写的共享，即使它位于本地节点。如果创建由口令保护的共享，则除非先设置了网络驱动器映射，否则将不能使用 UNC 来访问该共享。您可以使用 Windows 的“资源管理器”从远程节点来设置驱动器映射。

## 在分布式环境中记录数据

通过使用 InTouch 分布式历史系统，可以从网络上的任何 InTouch 应用程序中检索历史数据。此系统还允许同时从多个历史数据库远程检索数据。这些数据库都称为历史供应器。

只有一个 InTouch 节点可以写入分布式历史文件。不过，查看该文件内容的 InTouch 节点数则没有任何限制。



从历史文件检索数据的远程节点可能看不到最近一个小时的数据（基于 Logger 节点的时间）。远程趋势只能查看已写入记录节点的磁盘的数据。

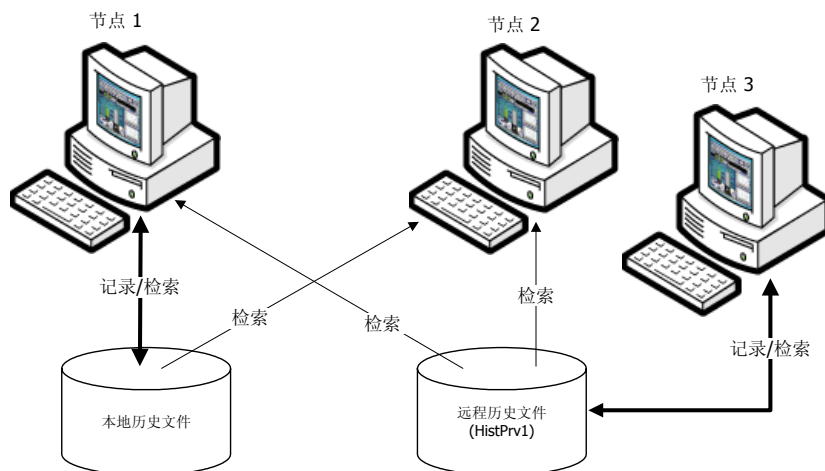
对于选择了“记录数据”的每个标记名，在给该标记名采集 22 个样本之后，其数据会自动写入磁盘。如果调用 `HTUpdateToCurrentTime()` 函数，则无论采集多少个样本，数据都会写入磁盘。缺省条件下，数据每小时写入磁盘一次。通过将下面这行代码添加到 `INTOUCH.ini` 文件，可以更改此时间间隔：

```
ForceLogging=X;
```

其中 **X** 是分钟数，可设置为 5 和 120 之间的任意时间间隔。

**备注** 使用“分布式历史”系统时，Wonderware NetDDE Helper 服务必须正在运行。

下图显示一个典型的分布式历史系统的配置，它使用“网络应用程序开发”(NAD) 来分发应用程序。



节点 1 与 2 包含同一份 InTouch 应用程序；不过该应用程序配置为只允许节点 1 写入本地历史文件，而任何一个节点都可以从本地历史文件或远程历史文件中检索。节点 3 也正在写入远程历史文件位置并从中进行检索。给节点 3（历史供应器）指定的名称为 **HistPrv1**。节点 1 既是开发工作站又是运行时工作站，而节点 2 只是运行时工作站。

执行以下主要步骤以创建这种类型的应用程序：

- 1 创建一个历史供应器列表。请参阅第 39 页的“配置远程历史供应器”。
- 2 创建并配置历史趋势对象。如需有关详细信息，请参阅 *InTouch® HMI 数据管理指南* 中的第 10 章“绘制标记数据的趋势”。
- 3 配置应用程序的分布式记录功能。请参阅第 41 页的“配置分布式历史记录”。

#### 4 分发应用程序。请参阅第 42 页的“针对 NAD 配置 InTouch 应用程序”。

您可以手工或通过使用 NAD 来分发应用程序。在分发应用程序时，历史供应器列表文件作为应用程序的一部分进行分发。

分发应用程序之后，可以运行 View 节点，并同时检索本地标记名和远程历史供应器的标记名。尽管应用程序将在所有的 View 节点上运行，但只有记录节点才将数据记录到历史日志文件；而其它节点则只能从中读取。

### 配置远程历史供应器

您必须给要用于 InTouch HMI 的每个远程历史供应器指定一个名称与网络位置。您可以使用远程 InTouch 历史供应器，也可以使用远程 Wonderware Historian 历史供应器。

**备注** 您无法给 InTouchView 应用程序配置远程历史供应器。如需有关 InTouchView 应用程序的限制的详细信息，请参阅第 166 页的“InTouchView”。

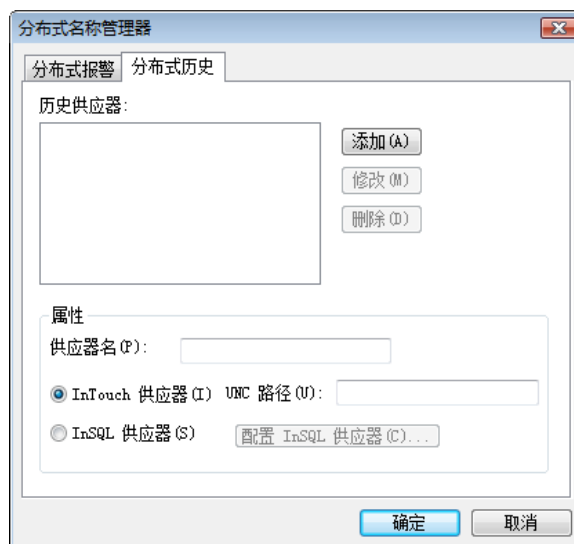
在本地 InTouch 应用程序被认为是历史供应器的情况下，不必为应用程序去定义它。

如果在应用程序中引用了未定义的历史供应器，则 WindowViewer 会忽略该引用并将一条错误消息写入 Logger。

HistData 实用程序无法从 Wonderware Historian 供应器中检索历史信息。

### 要配置历史供应器

- 1 在特别菜单上，指向配置，然后单击分布式名称管理器。此时出现分布式名称管理器对话框。
- 2 单击分布式历史选项卡。



- 3 在供应器名框中，为新的历史供应器输入希望使用的名称。



供应器名的长度不得超过 16 个字母数字字符。

**4 要配置 InTouch 历史供应器，执行以下操作：**

- a** 单击 **InTouch 供应器**。
- b** 在 **UNC** 框中，输入 InTouch 应用程序目录的 UNC 路径，然后单击**添加**。

UNC 路径格式为：

\\node\_name\volume\_name\directory\

如果 UNC 位置受口令保护，必须先使用 Windows 的“资源管理器”来建立节点连接。

**5 要配置 Wonderware Historian 历史供应器，执行以下操作：**

- a** 单击 **InSQL 供应器**。
- b** 单击**配置 InSQL 供应器**。此时出现 **InSql History Provider 属性**对话框。



- c** 在**数据源**框中，输入 Wonderware Historian 数据库所在节点的名称，最多包含 35 个字符。
- d** 在**用户**框中，输入登录帐户的用户名。用户帐户必须拥有检索数据的数据库权限。
- e** 在**口令**与**再次输入口令**框中，输入登录帐户的口令。
- f** 单击**测试**以验证与 Wonderware Historian 的连接。出现消息时，单击**确定**。
- g** 单击**确定**以关闭 **InSql History Provider 属性**对话框。

**6 单击**确定**。**

### 动态配置远程历史供应器

在运行时，您也可以动态配置历史趋势的远程历史供应器，方法是创建一个脚本，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");
```

其中，1 指定将绘制指定的远程历史供应器标记的趋势笔。



对于远程历史供应器，运行时**历史趋势设置**对话框与 .Pen 点域不受支持。

## 配置分布式历史记录

只有一个 InTouch 节点可以写入历史文件。不过可以有多个 InTouch 节点查看该文件。

**备注** 您无法给 InTouchView 应用程序配置历史记录。如需有关 InTouchView 应用程序的限制的详细信息，请参阅第 166 页的“InTouchView”。

### 要配置分布式历史记录

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



- 2 选择**启用历史记录**复选框以打开全局标记记录功能。
- 3 选择**在指定目录中存储日志文件**，然后在输入框中输入日志文件存储位置的路径。  
您必须输入有效的“通用命名惯例”(UNC) 路径。例如，  
\\Node\Share\Path  
如果使用 NAD，必须确保路径指向应用程序文件夹之外的某个文件夹。
- 4 在**记录节点名**框中，输入要记入历史日志文件的节点的名称。此设置只允许将此处指定的节点记录到文件中。
- 5 单击**确定**。

**备注** 在选择了**启用历史记录**选项的应用程序分发到 WindowViewer 节点时，该节点会检查此选项以确定是否应该记录它。如果选择了**启用历史记录**，则可能的设置是：  
字段等于“节点”的名称 - 启用记录功能  
字段不等于“节点”的名称 - 禁用记录功能

## 特殊网络的注意事项

如果使用低速网络，并因此导致 InTouch HMI 启动或保存信息要花费很长时间，则可以在 NAD 客户端上修改 win.ini 设置：

ViewNadClearNADCopyDirectory=0

ViewNADCopyApplicationOnStartup=1

ViewNADOnApplicationChanged=3（或 4）

ViewNADThreadPriority=2

对于 ViewNADOnApplicationChanged 参数，设置为 3 对应于“**InTouch 应用程序管理器**”中**节点属性**对话框上的**将更改加载到 WindowViewer** 选项。设置为 4 对应于**提示用户将更改加载到 WindowViewer** 选项。这些设置可以让应用程序继续执行，而 NAD 下载则在单独的执行线程上同时进行。

NAD 对应用程序执行更新时，它只是从主应用程序复制更改过的文件。NAD 不复制用于运行时语言切换功能的 SmartSymbol 设计时字典文件。

## 针对 NAD 配置 InTouch 应用程序

“网络应用程序开发”或简称 NAD 是一种特殊架构，它兼备基于客户端和基于服务器的架构的优点。NAD 不仅提供自动通知应用程序更改的功能，而且还可以自动将更新的应用程序分发到各个 View 节点。

针对 NAD 配置应用程序时，必须指定 WindowViewer 要将主应用程序复制到哪个文件夹。

- 如果这是开发节点，则可以输入本地文件夹路径，如 c:\InTouch\NAD。您也可以输入远程网络 UNC 路径，如 \\node\share\path。对于大多数文件存储都在一个中央位置的基于文件服务器的网络，这非常方便。
- 如果这是客户端节点（仅限运行时），则通常使用本地文件夹路径。

我们建议尽量使用本地文件夹，以避免因网络延迟与故障而影响 WindowViewer 的操作。

---

**注意** 不要使用根文件夹或指向根文件夹的 UNC 路径名。在复制主应用程序目录之前，View 节点将删除指定的目标应用程序文件夹中的所有文件与子文件夹。因此，切勿使用主应用程序文件夹的路径或指向主应用程序文件夹的 UNC 路径。

---

如果不指定文件夹，WindowViewer 会在 WindowViewer 启动的文件夹中自动创建一个名称为 NAD 的本地子文件夹。此 NAD 文件夹应视为临时文件夹，并且除了由 NAD 自身复制的那些文件之外，不得将其它文件保存到在其中。

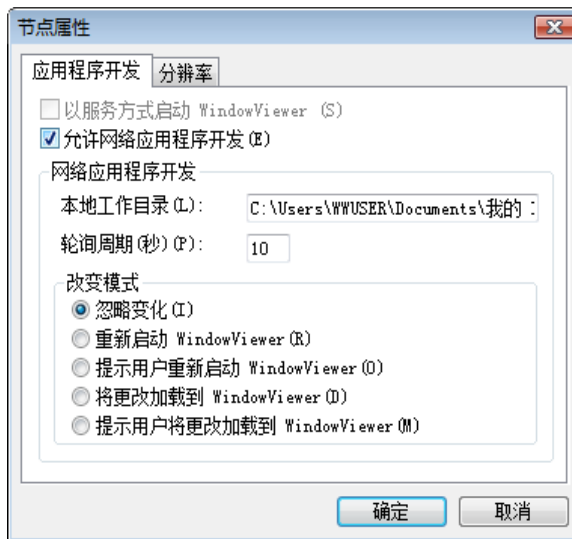
如果将 WindowViewer 当作服务运行，则无法使用 NAD 更新功能。如需有关详细信息，请参阅第 56 页的“作为服务运行 WindowViewer”。

### 要针对 NAD 配置应用程序

1 启动“应用程序管理器”。



2 在工具菜单上，单击节点属性。此时出现节点属性对话框。



3 选择允许网络应用程序开发复选框。

4 在本地工作目录框中，输入希望 WindowViewer 将主应用程序所复制到的文件夹的路径。

5 在轮询周期（秒）框中，输入以秒计的间隔，View 节点将按照此间隔来检查开发节点是否有更新。

注意不要将这个值设置得过小。如果 WindowViewer 过于频繁地检查主应用程序，则会干扰它为运行中的应用程序提供服务。

6 在改变模式区域中，选择相应的选项确定在主应用程序发生更改时 WindowViewer 所要采取的操作。

- 单击忽略变化，让 WindowViewer 节点忽略开发节点上所作的任何更改。
- 单击重新启动 Window Viewer，让 WindowViewer 节点复制更新的主应用程序（如果配置成如此），然后重新启动自身。

- 单击**提示用户重新启动 WindowViewer**，以便向操作员显示应用程序已发生更改的消息。操作员可以重新启动 WindowViewer 并更新应用程序，也可以继续使用当前的应用程序。
- 单击**将更改加载到 WindowViewer**，以便将开发节点上发生的更改动态加载到 WindowViewer 中。对于大规模的更新，这可能会影响性能。
- 单击**提示用户将更改加载到 WindowViewer**，以便向操作员显示应用程序已发生更改的消息。此消息提示操作员加载这些更改。

7 单击**确定**。

## 执行 NAD 自动更新

在应用程序开发期间，您可以启动 NAD 自动更新。

运行**通知客户端**命令时，会设置一个标识，向所有的远程 View 节点通知主应用程序已经发生更改。客户端可以根据为每个节点定义的**改变模式**选项自动启动更新过程。

**要执行自动更新**

- 1 在 WindowMaker 中打开应用程序。
- 2 在**特别菜单**上，单击**通知客户端**。

## 执行 NAD 手工更新

您可以编写脚本供操作员在他们工作的 View 节点上手工启动 NAD 更新。

要使用 NAD 手工更新应用程序，必须在**节点属性**对话框中，将**改变模式**选项设置为**忽略变化**。如需有关详细信息，请参阅第 42 页的“针对 NAD 配置 InTouch 应用程序”。

在脚本中使用以下系统标记与函数执行 NAD 手工更新：

- \$ApplicationChanged 系统标记
- \$ApplicationVersion 系统标记
- RestartWindowViewer() 函数
- ReloadWindowViewer() 函数

### \$ApplicationChanged 系统标记

发出信号，指出“网络应用程序开发” (NAD) 架构中的主应用程序已经更改。

**类别**

应用程序

## 用法

\$ApplicationChanged

## 附注

每次通过选择 WindowMaker 特别菜单上的**通知客户端**生成更新信号时，此系统标记便更改为 1。在更新应用程序时，\$ApplicationChanged 重置为 0。此标记可用于生成一条消息，通知操作员主应用程序已经发生更改。

您也可以在数据改变脚本中使用 \$ApplicationChanged 系统标记，以构建节点更新通知脚本。此脚本可以启动您自己的对话框或停止正在运行的进程。然后您可以使用 ReloadWindowViewer() 函数来启动更新过程。

## 数据类型

离散（只读）

## 示例

在数据改变脚本的标记名框中使用以下语句会导致运行脚本的主体。脚本主体可能显示一个窗口，提醒用户重新启动 WindowViewer 以便使更改生效。

\$ApplicationChanged

## 另请参阅

\$ApplicationVersion

## \$ApplicationVersion 系统标记

包含应用程序的当前版本号。每次发生可以保存或撤消的更改时，此数字都会更改。

## 类别

应用程序

## 用法

\$ApplicationVersion

## 附注

与 \$ApplicationVersion 系统标记关联的值设置为 InTouch 应用程序的当前版本。每次应用程序发生可以保存或撤消的更改时，版本都会更改。此标记可用于生成一条消息，通知操作员主应用程序已经发生更改。

## 数据类型

实型（只读）

## 示例

如果用在模拟显示链接中，则此系统标记显示 WindowViewer 中运行的应用程序的当前版本。

\$ApplicationVersion

## 另请参阅

\$ApplicationChanged

### RestartWindowViewer() 函数

关闭 WindowViewer，复制更新的主应用程序（如果配置成如此），然后重新启动 WindowViewer。

#### 类别

系统

#### 语法

```
RestartWindowViewer();
```

#### 附注

在未使用“网络应用程序开发” (NAD) 自动更新函数时，可以使用此函数来更新应用程序。

使用 \$ApplicationChanged 系统标记来确定何时发生 NAD 更新。

您使用**通知客户端**命令来启动 NAD 更新。不过，操作员可能需要延迟更新时间。您可以将此函数用于某个按钮动作脚本，以便操作员可以在方便时重新启动 WindowViewer。

您也可以使用 ReloadWindowViewer() 函数，它可以在不关闭 WindowViewer 的情况下更新 View 节点。

#### 另请参阅

\$ApplicationChanged, ReloadWindowViewer()

### ReloadWindowViewer() 函数

使用更新的 NAD 主应用程序来动态更新 WindowViewer，而无需中断服务。

#### 类别

系统

#### 语法

```
ReloadWindowViewer();
```

可供用户控制 WindowViewer 的重新加载。

#### 附注

在未使用“网络应用程序开发” (NAD) 自动更新函数时，可以使用此函数来更新应用程序。

使用 \$ApplicationChanged 系统标记来确定何时发生 NAD 更新。

您使用**通知客户端**命令来启动 NAD 更新。不过，操作员可能需要延迟更新时间。您可以将此函数用于按钮动作脚本，以便操作员可以在方便时在 WindowViewer 中重新加载应用程序。

#### 另请参阅

\$ApplicationChanged

## 应用程序编辑锁定

为防止多个开发人员同时编辑某个应用程序，在编辑会话中 WindowMaker 锁定该应用程序。如果试图打开锁定的应用程序，则会显示一条错误消息。当前编辑该应用程序的节点的名称包含在消息中。

如果 WindowMaker 在加载应用程序时异常关闭，则可能不会删除 appedit.lock 文件。您可以通过从应用程序目录中删除 appedit.lock 文件来手工解除锁定。

## 在 NAD 更新期间对应用程序更改

WindowViewer 节点更新应用程序时，它会尽可能地在复制过程中保留主应用程序的属性（只读、系统、隐藏，等等）。

WindowViewer 也会复制主应用程序的所有文件与子文件夹，以下文件除外：\*.WVW、\*.DAT、\*.LGH、\*.IDX、\*.LOG、\*.LOK、\*.FSM、\*.STG、\*.DBK、\*.CBK、\*.HBK、\*.KBK、\*.LBK、\*.NBK、\*.OBK、\*.TBK、\*.WBK、\*.XBK、\*.\$\$\$、RETENTIV.X、RETENTIV.D、RETENTIV.A、RETENTIV.S、RETENTIV.H、RETENTIV.T、SSD\_、WM.INI、DB.INI、LINKDEFS.INI、TBOX.INI、GROUP.DEF 以及 ITOCX.CFG。

---

**备注** WindowViewer 以递归方式删除目标应用程序文件夹中的所有文件与子文件夹（运行时语言切换功能所需的那些除外）。此文件夹应视作临时文件夹。不能存放其它文件。

---

开始更新之后，NAD 客户端会创建一个列表，列出客户端应用程序目录中出现的本地文件与子目录。NAD 客户端一边在主文件列表中查找更新，一边从本地列表中删除每个主文件对应的客户端文件。本地列表中剩余的项目是废弃的文件与子目录，应该从应用程序中删除掉。

下载的所有文件将复制到临时子目录 NAD\_Temp 中。仅在重试次数限制范围内成功复制所有的新文件以及更新的文件之后，才会将这些文件从 NAD\_Temp 复制到应用程序目录。如果 NAD 客户端不得不放弃更新，正在运行的应用程序并不会因为只引入部分新文件或更新的文件而损坏。

如果下载所有的新文件与更新的文件之后无法联系到 NAD 主应用程序，则通过从 NAD\_Temp 中复制更新并删除废弃的文件，仍可以完成更新。这确保不会仅仅因为丢失连接而无法确认文件是否存在于主应用程序中便将它们删除掉。

NAD 可以检测出在下载应用程序期间是否对主应用程序进行过其它的更改。如果发生这种情况，NAD 将放弃下载应用程序。如果在最近的更新之后运行**通知客户端**命令，NAD 将在下一个轮询周期自动开始下载最新的应用程序。否则，它将等到发出下一条**通知客户端**命令之后再开始下载应用程序。

## 在运行时调整应用程序分辨率

通过使用“动态分辨率转换”（Dynamic Resolution Conversion，简称 DRC），可以使创建的分布式应用程序能够在不同的屏幕分辨率下运行。

每个 View 节点都可以适当地调整应用程序，包括调整到自定义的分辨率。这种比例调整可以在 WindowViewer 编译应用程序时进行，并不要求使用 WindowMaker。每个 View 节点都可以使用不同的 DRC 设置，因此每个 View 节点都必须配置自己的设置。

**注意** 如果不使用 DRC 来调整应用程序，则仅当节点的屏幕分辨率与应用程序开发节点的屏幕分辨率完全相同时，WindowViewer 才会运行应用程序。如果分辨率不同，WindowViewer 会提示操作员运行 WindowMaker，以便按该节点的分辨率转换应用程序。如果已设置主应用程序目录的 UNC 路径，请务必小心，因为这只会修改原始应用程序。

### 要给应用程序配置 DRC

- 1 启动**应用程序管理器**。
- 2 在工具菜单上，单击**节点属性**。此时出现**节点属性**对话框。
- 3 单击**分辨率**选项卡。



- 4 如果希望 WindowViewer 在本地缩放主应用程序，请选择**允许 WindowViewer 动态改变分辨率**复选框。



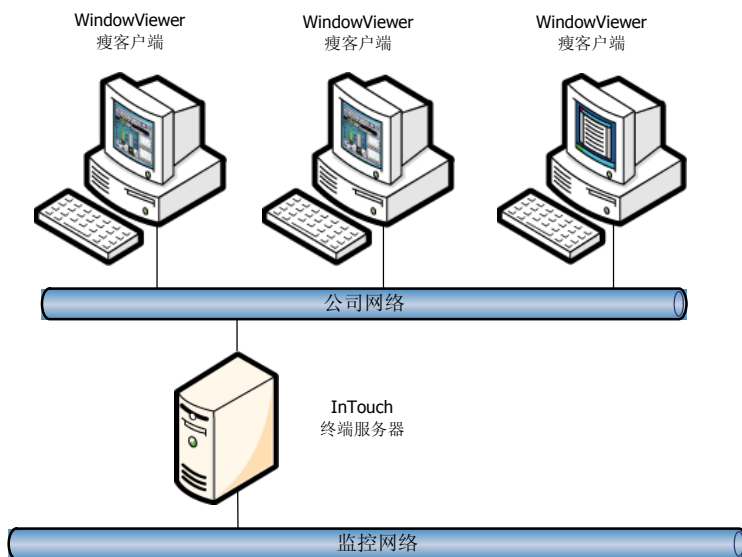
5 在**动态分辨率**区域中，选择以下操作之一：

- 如果希望 WindowViewer 以开发时的分辨率运行应用程序而忽略节点的分辨率，请选择**使用应用程序分辨率**。例如，如果应用程序在 800x600 下开发，而节点的分辨率是 1024x768，则 WindowViewer 不动态缩放应用程序。相反，应用程序分辨率保持为 800x600。
- 如果希望 WindowViewer 以节点分辨率运行应用程序而忽略开发时应用程序的分辨率，请选择**转换为屏幕视频分辨率**。例如，如果节点以 800x600 运行，而应用程序是在 1280x1024 下开发的，则 WindowViewer 动态调整应用程序，使之适合节点的 800x600 分辨率。
- 如果希望 WindowViewer 以**像素宽度 [X]** 与**像素高度 [Y]**（必须是整数值）框中指定的特定分辨率运行应用程序，请选择**自定义分辨率**。应用程序的分辨率和节点的分辨率都会被忽略。例如，如果**像素宽度 [X]** 与**像素高度 [Y]** 分别设置为 512 和 384，则应用程序将动态缩放，以适合节点屏幕上的 512x384 像素区。

6 单击**确定**。

## 使用终端服务

“终端服务”是 Microsoft Windows Server 操作系统中包含的一种可配置的服务，这些系统从服务器集中运行基于 Windows 的应用程序。在“终端服务”中，客户端计算机访问服务器节点，其中有多 InTouch 应用程序的实例在同时运行。



“终端服务”环境由三个部分组成：

- **终端服务服务器** 服务器管理每个客户端会话的计算资源，并为客户端用户提供唯一的环境。服务器接收并处理远程客户端上执行的所有键击与鼠标操作，并将操作系统与应用程序的所有显示输出传递给相应的客户端。所有的“终端服务”应用程序处理都在服务器上发生。
- **远程桌面协议（Remote Desktop Protocol，简称 RDP）**。“远程桌面协议”（RDP）客户端应用程序将输入数据（如键击、鼠标移动等）传递给服务器。
- **客户端**。“终端服务”客户端不执行本地应用程序处理，它仅显示应用程序输出。通过运行 Windows 程序菜单上的**终端服务客户端**命令，可以从客户端访问“终端服务”。与“终端服务器”连接之后，客户端环境看起来与 Windows 服务器的一样。由于应用程序不在本机运行，因此是完全透明的。

如需有关“终端服务”的详细信息（包括功能与优点），请参阅 Microsoft 文档。

## 终端服务器应用程序的规划注意事项

---

**重要** 我们建议先在测试服务器上安装应用程序，然后再在生产环境中部署。

---

在安装“终端服务”之前：

- 确定需要在服务器上安装的客户端应用程序（如 Wonderware InTouch）。
- 确定客户端的硬件要求。
- 确定支持客户端所需的服务器配置。
- 确定“终端服务”以及要运行的其它应用程序所需的许可证。
- 理解 InTouch 应用程序的某些方面（如报警、安全性、I/O 及脚本）在“终端服务”下如何运行。

### 终端服务环境中的报警

通过给 Terminal Services for InTouch 使用“分布式报警系统”，不同终端会话上运行的客户端可以选择显示哪些报警以及如何显示。

“报警供应器”通过唯一确定应用程序及应用程序实例的名称来识别它们。“报警供应器”或“报警接收器”向“分布式报警系统”注册之后，此信息将可供“分布式报警系统”使用。

运行“报警供应器”的节点则通过在系统中唯一确定该计算机节点的名称进行识别。在该计算机节点上启动“分布式报警系统”的实例之后，此信息便会提供给它。

在记录报警事件时，节点与完整的“报警供应器”名可以确定报警的来源。

在“终端服务”环境中确认报警时，记录的“操作员节点”是运行操作员所使用的“终端服务”会话的客户计算机的名称。如果无法检索计算机的节点名，则使用节点的 IP 地址。

---

**备注** “报警供应器”在“终端”会话上不受支持。它们只在“终端控制台”上受支持。

---

## 终端服务环境中的安全性

使用应用程序安全性来保证 InTouch 应用程序、Wonderware Historian 及其它机密信息系统的安全。

- 使用 \$Operator 系统标记来确保应用程序安全性。然后，您可以通过将这些函数链接到内部标记，以控制操作员对特定功能的访问。

如需有关使用 \$Operator 系统标记的详细信息，请参阅第 109 页的“保护 InTouch 安全”。

- 使用更新的 TseGetClientId() 函数取代 GetNodeName() 函数，来确定客户端计算机。使用“终端服务”时，GetNodeName() 函数返回终端服务器的名称，而不是客户端计算机的名称。

使用安全性跟踪来监视非法入侵企图。如果怀疑系统遭受任何形式的攻击，则可以启用日志功能，以记录一系列可跟踪的事件。通常安全性日志 / 跟踪功能要求使用大量的处理资源，所以缺省条件下它是禁用的。

---

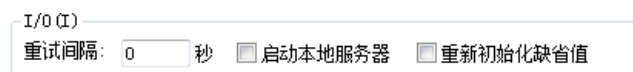
**注意** 安全性跟踪要求使用大量的资源。在评估试验服务器时启用跟踪功能，以准确估计 InTouch 应用程序硬件要求。

---

## 终端服务环境中的 I/O

InTouch HMI 无法在“终端服务”环境中启动“I/O 服务器”。根据 view 会话的启动顺序，可能需要使用 IOREinitialize() 函数。在启动从这些服务器读取数值的应用程序之前，必须确保所有的服务器（I/O 设备或 view 应用程序）都在运行。

为避免在 WindowViewer 启动时接收到“正在初始化 I/O”的错误消息，请清除 **WindowViewer 属性**对话框中**常规**选项卡上的**启动本地服务器**复选框。




---

**备注** 由于“终端服务”架构的关系，将 WindowViewer 作为“Windows 服务”来运行并不是受支持的配置。

---

终端服务环境中的脚本执行

“终端服务”中运行的所有应用程序都使用相同的定时参考（服务器时钟），因此在存在大量的 CPU 加载操作期间，脚本可能不会运行。CPU 加载异常可能会由视频处理负载过重引起，在多个应用程序定义了相同的脚本触发器（例如 End-of-Shift 事件）时，也可能会发生此种情形。因此，如果服务器正忙于处理许多客户端的脚本，则在定时器通常可以启动脚本的时间间隔之内，它可能无法启动其它客户端上的脚本。这可能会使脚本无法在客户端上运行。

为确保脚本正确运行，请将使用公用触发器的脚本合并到一起，并将它们移到一个应用程序（如标记服务器）中。这是执行试验布署的主要原因之一。通过试验部署，您可以有机会执行压力测试，以确定所选的硬件配置是否够用。

终端服务环境中的其它限制

下表介绍在终端服务器上运行应用程序时的一些限制以及建议的解决方案。

功能	是否支持?	注释
WindowViewer	是	WindowViewer 在“终端服务”下作为服务运行时不受支持。
AlarmSuite Logger	否	使用标记服务器（仅限单机）。
到 I/O 设备或 MS Office（例如 Excel）的 DDE	否	使用标记服务器（控制台或单机）。这包括 DDE QuickScript: WWExecute()、WWpoke() 以及 WWRequest()
自 MS Office 的 DDE（例如 Excel 中配置的热链接）	是	Excel 与 InTouch HMI 必须在相同的会话中运行。
历史趋势	是	使用标记服务器或 NAD 来记录数值。多个会话可能会读取相同的历史文件，但仅有一个控制台可以写入历史文件。
InTouch Alarm Logger	是	--
MEM OLE 自动化	否	--
打印报警	否	--
保留标记	是	必须使用 NAD。
SPC Pro	否	不支持。
SQL Access (ODBC)	是	数据库必须位于单独的计算机上。

功能	是否支持?	注释
到 I/O 设备或其它 InTouch 应用程序的 SuiteLink	是	与另一个 view 会话通讯时，应包含“终端服务器”节点名，并将所需会话的 IP 地址追加到应用程序名。例如，view10.103.25.6。“I/O 服务器”在客户端会话中不受支持。

## 使用脚本检索关于 InTouch 客户端会话的信息

您可以给“终端服务”使用以下 InTouch QuickScript 函数。

- TseGetClientId() 函数
- TseGetClientNodeName() 函数
- TseQueryRunningOnConsole() 函数
- TseQueryRunningOnClient() 函数

### TseGetClientId() 函数

如果 View 应用程序在“终端服务器”客户端上运行，则返回字符串形式的客户端 ID（客户端的 TCP/IP 地址）。此客户端 ID 供内部使用，以生成 SuiteLink 服务器名与 logger 文件名。否则 TseGetClientId() 函数返回一个空字符串。

#### 语法

```
MessageResult=TseGetClientId();
```

#### 示例

客户端 IP 地址 10.103.202.1 保存到 MsgTag 标记中。

```
MsgTag=TseGetClientID();
```

### TseGetClientNodeName() 函数

如果给运行 View 应用程序的“终端服务器”客户端指定的名称 Windows 能够识别，则返回客户端节点名。否则 TseGetClientNodeName() 函数返回一个空字符串。

#### 语法

```
MessageResult=TseGetClientNodeName();
```

#### 示例

客户端节点名作为指定给 MsgTag 标记的值返回。

```
MsgTag=TseGetClientNodeName();
```

### TseQueryRunningOnConsole() 函数

TseQueryRunningOnConsole() 函数可以从脚本中运行，以显示 View 应用程序是否正在“终端服务”控制台上运行。

**语法**

```
Result=TseQueryRunningOnConsole();
```

**返回值**

如果 View 应用程序在“终端服务”控制台上运行，则返回非零整数值。否则 TseQueryRunningOnConsole() 函数返回零。

**示例**

如果 WindowViewer 在“终端服务”控制台上运行，则 IntTag 设置为 1。

```
IntTag=TseQueryRunningOnConsole();
```

**TseQueryRunningOnClient() 函数**

如果 View 应用程序在“终端服务”客户端上运行，则返回非零整数值。否则返回零。

**语法**

```
Result=TseQueryRunningOnClient();
```

**返回值**

如果 View 不在“终端服务”客户端上运行，则返回 0。

**示例**

如果 WindowViewer 在“终端服务”客户端上运行，则 IntTag 设置为 1。

```
IntTag=TseQueryRunningOnClient;
```

## 第 3 章

# 管理 InTouch 服务

服务是执行无人照管的特定后台系统功能的某种 Windows 进程，它没有用户界面，也不要求用户登录。

以下启动选项可供 Windows 服务使用：

- **自动。**Windows 重新启动时，服务自动启动而无需任何用户干预。
- **手动。**用户或应用程序进程必须明确启动服务。
- **已禁用。**禁止服务启动。这对于疑难排解非常有用。

服务启动时不影响 Windows 安全系统。

InTouch HMI 包含以下 Windows 服务：

- Alarm DB Logger
- Alarm DB Purge-Archive
- Wonderware NetDDE Helper
- Wonderware SuiteLink
- Wonderware WindowViewer

## 作为服务运行 WindowViewer

如果将 WindowViewer 配置成作为服务运行，WindowViewer 会在安装了该应用程序的计算机启动时自动启动。

WindowViewer 服务在后台运行，直到有用户登录进来，此时 WindowViewer 打开。这有以下优点：

- 大多数灾难恢复计划都要求在恢复供电之后，立即启动关键的计算机系统。Microsoft Windows 服务器可以在供电恢复之后自动重新启动。WindowViewer 作为服务运行时，工厂自动化系统可以立即开始运行。计算机重新启动时，WindowViewer 中上次打开的 InTouch 应用程序自动启动。
- WindowViewer 继续记录历史数据、搜集报警消息、处理脚本、充当“I/O 服务器”、作为 I/O 客户端写入数值、甚至作为不同的操作员进行登录与注销。

WindowViewer 作为服务运行时，如果试图使用快捷方式图标或通过单击 Windows 开始菜单上的 WindowViewer 来启动 WindowViewer，则出现一条消息。此消息描述在 WindowViewer 配置成作为服务运行时，它在启动方面所受到的限制。

---

**重要** 在 Windows Vista 操作系统上不支持 WindowViewer 作为服务来运行。

---

## 将 WindowViewer 配置成作为服务运行

WindowViewer 作为 Windows 服务运行时，可以在操作员注销之后提供连续的操作，以及在系统启动期间自动启动，而无须操作员干预。这样便可以让无人工作站启动 WindowViewer，而不会影响操作系统的安全性。

---

**重要** 在 Windows Vista 操作系统上不支持 WindowViewer 作为服务来运行。

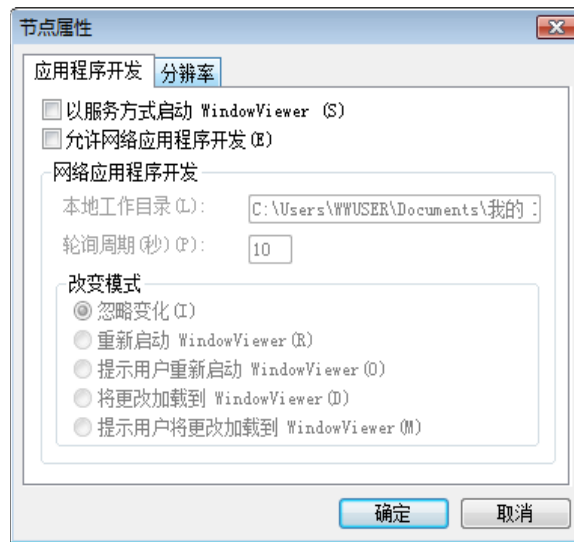
---

**要将 WindowViewer 配置成作为服务启动**

- 1 启动 InTouch。此时出现“InTouch 应用程序管理器”。



- 2 在工具菜单上，单击**节点属性**。此时出现**节点属性**对话框。



- 3 选择**以服务方式启动 WindowViewer** 复选框，以便将 WindowViewer 配置为作为服务自动运行。
- 4 单击**确定**。

## 手工启动服务

您可以使用 Windows 的“控制面板”手工启动 Wonderware WindowViewer 服务。

WindowViewer 不出现在“服务”控制面板中，除非您将它配置成作为服务运行。如需有关详细信息，请参阅第 56 页的“将 WindowViewer 配置成作为服务运行”。

### 要使用“控制面板”启动 WindowViewer 服务

- 1 启动“控制面板”。
- 2 双击**管理工具**，然后双击**服务**。此时出现**服务**对话框。
- 3 在“详细信息”面板中，使用鼠标右键单击 **Wonderware WindowViewer** 服务，然后单击**启动**。

### 要使用命令提示符启动 WindowViewer 服务

- 1 启动命令提示符。
- 2 输入 Net Start view。

View 是 Wonderware WindowViewer 可执行程序文件的名称。

## 停止服务

您可以使用“控制面板”手工停止 WindowViewer 服务。

### 要使用“控制面板”停止 WindowViewer 服务

- 1 启动“控制面板”。
- 2 双击**管理工具**，然后双击**服务**。此时出现**服务**对话框。
- 3 在“详细信息”面板中，使用鼠标右键单击 **Wonderware WindowViewer**，然后单击**停止**。

### 要使用命令提示符停止服务

- 4 启动命令提示符。
- 5 输入 Net Stop View。

View 是 WindowViewer 可执行程序文件的名称。

## 配置 InTouch 服务的用户帐户

缺省条件下，Windows 服务使用本地系统帐户运行。InTouch 服务要求使用拥有某些管理权限的用户帐户，本地系统帐户可能没有这些权限。

安装 InTouch HMI 时，如果还没有创建帐户，则可以指定一个所有 ArchestrA 服务都在其下运行的管理帐户。此帐户被视为 ArchestrA 主帐户。InTouch 服务使用 ArchestrA 主帐户来自动启动。

---

**备注** 主帐户也称为模拟帐户。模拟帐户是一个用户或用户组帐户，它提供访问您的站点中或服务器中受限资源“区域”的权限。

---

如果希望更改主帐户，请使用 ArchestrA Change Network Account 实用程序。

---

**注意** 更改主帐户会影响所有的 ArchestrA 服务，不仅仅是 InTouch 服务。

---

### 要更改 ArchestrA 主帐户

- 1 在开始菜单上，指向**程序**，指向 **Wonderware**，指向 **Common**，然后单击 **Change Network Account**（更改网络帐户）。此时出现**更改网络帐户**对话框。
- 2 更改用户帐户。如需有关详细信息，请参阅“更改网络帐户”文档。
- 3 单击**确定**。

## InTouch 服务疑难排解

**重要** 在 Windows Vista 操作系统上不支持 WindowViewer 作为服务来运行。

如果某项服务可以启动与否取决于其它一些服务是否已启动，则 Windows 会在启动该服务之前验证这些作为前提条件的服务是否正在运行。

根据运行 WindowViewer 的要求，注意以下相关性：

- 如果计划使用“分布式报警”或“分布式历史”，或者是您打算访问网络 DDE 数据，则必须运行 Wonderware NetDDE Helper 服务。

Wonderware NetDDE Helper 服务也取决于 Network DDE 与 Network DDE DSDM 服务安装并配置成“手动”还是“自动”启动。安装期间，Wonderware NetDDE Helper 服务配置成“手动”启动。WindowViewer 在计算机启动时自动启动此服务。

- 如果需要 WindowViewer 作为 SuiteLink 服务器或客户端运行，则必须运行 Wonderware SuiteLink 服务。

Wonderware SuiteLink 服务还要求安装 Microsoft TCP/IP。

- 如果希望在运行 WindowViewer 时存储任何消息或错误，则必须确保安装了 ArchestrA Logger 服务。

Wonderware SuiteLink 与 ArchestrA Logger 服务都应该安装并配置为在自动启动时运行。

## 查看服务的错误消息

使用 Windows 的“事件查看器”来排解同服务相关的错误消息。例如，您可能会看到“一个或多个服务无法启动...”。

Windows 的“事件查看器”列出在启动 Windows 服务时发生的提示性消息、警告或错误。如需有关“事件查看器”的详细信息，请参阅 Microsoft 文档。

您可以看到 InTouch 服务启动失败而导致的任何警告或错误消息。如果“事件查看器”指出 Wonderware WindowViewer 服务启动失败，最有可能的原因是所依赖的必要服务不在运行。

## 服务用户帐户问题的疑难排解

如果无法安装 InTouch 服务，或安装 InTouch HMI 之后无法启动，则运行它们的用户帐户可能有问题。

### 要排解服务用户帐户问题的疑难

- 1 打开 Windows 的“用户管理器”窗口，并创建一个新的主用户帐户。  
此用户帐户必须拥有本地计算机上的管理权限，以便以服务方式启动 InTouch 组件。如果在域名列表中看不到您计算机的节点名，请手工输入节点名。  
如需有关详细信息，请参阅第 58 页的“配置 InTouch 服务的用户帐户”。
- 2 确认计算机的节点名不超过 14 个字符。如果节点名包含下划线字符 ( ) 或短划线 (-)，请删除它们。
- 3 在安装期间，如果提示输入域名，请输入计算机的节点名而不是域名。然后输入步骤 1 中创建的用户名以及口令。
- 4 如果已经安装 InTouch HMI，则仍可以指定域名、用户名，以及运行 ArchestrA Change Network Account 实用程序所需的口令。
- 5 重新启动计算机。
- 6 使用任何有效的用户帐户登录到网络域。即便是域停机，也不会影响在本地计算机上运行的 InTouch 应用程序。

## 停用提示的 I/O 项目

启动 Windows 操作系统时，配置成自动启动的服务将在“后台”启动，而不会在桌面上出现可见的用户界面。在这种情况下，服务在系统环境下运行。操作员登录到系统时，对于在系统环境下运行的任何服务，如果有关联的用户界面，则会自动出现在桌面上。在这种情况下，服务在桌面环境下运行。

如果将 WindowViewer 服务配置成自动启动，则在启动操作系统时，服务在系统环境中运行。然后，在用户登录时，WindowViewer 服务继续运行，但是会转换到桌面环境中，WindowViewer 用户界面自动出现。

如果有一些 InTouch “访问名”在定义时打开了**只提示激活项**选项，并且有一些 I/O 标记只在特定的 InTouch 应用程序窗口中是活动的（这些标记未在应用程序中的任何其它地方使用），则可能“停用”那些标记。例如，如果 WindowViewer 作为服务运行，您使用脚本关闭应用程序窗口，则该窗口会自动从内存中释放，从而终止与那些标记的链接。

## InTouch 服务的注册表项

InTouch 服务作为注册表项列在 Windows 注册表中:

Wonderware SuiteLink:

- HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\SLS
- HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\slssvc
- HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\SuiteLink

Wonderware NetDDE Helper:

- HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\WWNetDDE

Wonderware WindowViewer:

- HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\VIEW



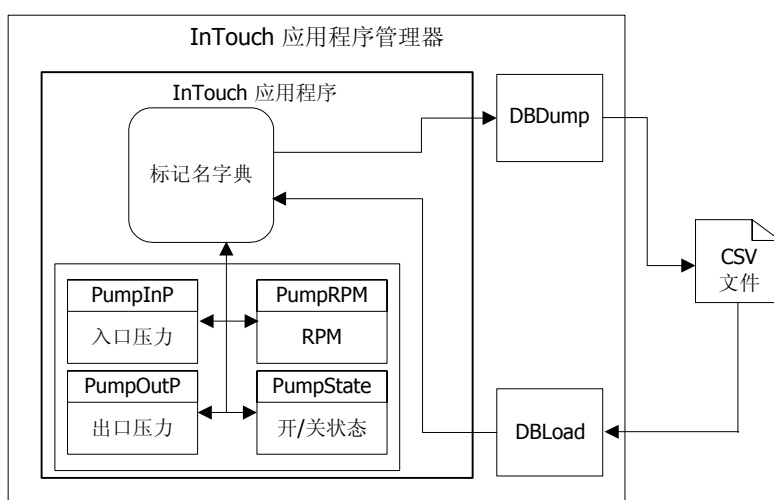
## 第 4 章

# 导出与导入标记定义、窗口以及脚本

通过导入或导出现有应用程序中的部分或所有组件，可以更快地创建 InTouch 应用程序。您可以从现有的应用程序中将标记定义、窗口以及脚本导入到新的应用程序。

## 导出标记定义

下图显示在中间导出文件与应用程序的“标记名字典”之间导出与导入标记定义的步骤。



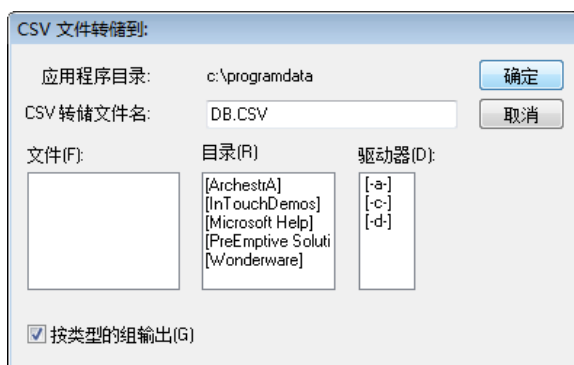
您使用“应用程序管理器”中的 DBDump 实用程序将“标记名字典”的内容导出到“逗号分隔值”（Comma Separated Value，简称 CSV）文件。您可以使用 Microsoft 的“记事本”

或 Microsoft Excel 来查看与编辑导出的文件。在进行编辑之后，使用同样是“应用程序管理器”实用程序的 DBLoad 实用程序，将标记定义导入 InTouch 应用程序。

要能够导出标记定义，必须先将应用程序转换为 InTouch HMI 软件的当前版本。

### 要导出标记定义

- 1 关闭 WindowMaker 与 WindowViewer。
- 2 启动“应用程序管理器”。此时应用程序管理器对话框显示一个 InTouch 应用程序列表。
- 3 从列表中选择应用程序。
- 4 单击 DBDump 图标。此时出现 **CSV 文件转储到：**对话框。

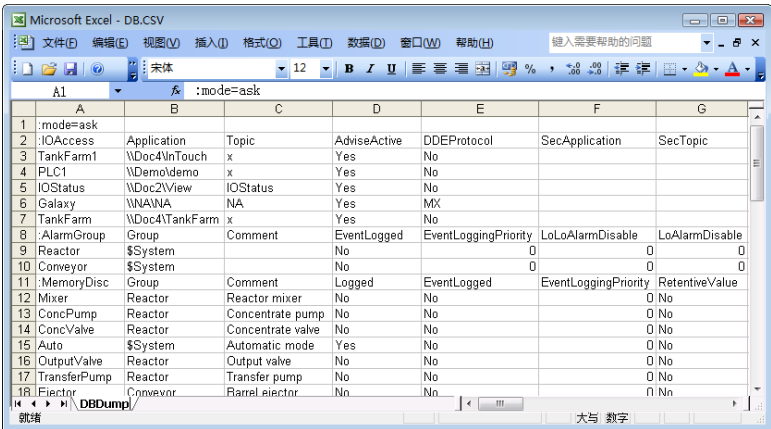


- 5 在 **CSV 转储文件名**框中，输入带 .csv 文件扩展名的文件名。
- 6 选择导出文件中数据组的类型。
  - 选择**按类型的组输出**复选框，以便在导出文件中按标记类型对数据进行分组。这是缺省值。
  - 清除**按类型的组输出**以便按标记名的字母顺序将输出内容保存到导出文件。
- 7 单击**确定**，以便将“标记名字典”的内容保存到所选的文件。此时出现一条消息，指出内容已成功保存到文件。



## 查看导出的标记定义

如果使用 Microsoft Excel 来查看使用 DBDump 实用程序创建的导出文件，则每条数据记录都出现在单独的电子表格单元格中。



A	B	C	D	E	F	G
1	:mode=ask					
2	:IOAccess	Application	Topic	AdviseActive	DDEProtocol	SecApplication
3	TankFarm1	\Doc4WinTouch	x	Yes	No	SecTopic
4	PLC1	\DemoVdemo	x	Yes	No	
5	IOStatus	\Doc2View	IOStatus	Yes	No	
6	Galaxy	\NANA	NA	Yes	MX	
7	TankFarm	\Doc4TankFarm	x	Yes	No	
8	:AlarmGroup	Group	Comment	EventLogged	EventLoggingPriority	LoLoAlarmDisable
9	Reactor	\$System	No		0	0
10	Conveyor	\$System	No		0	0
11	:MemoryDisc	Group	Comment	Logged	EventLogged	EventLoggingPriority
12	Mixer	Reactor mixer	No	No		RetentiveValue
13	ConcPump	Reactor	Concentrate pump	No	No	0 No
14	ConcValve	Reactor	Concentrate valve	No	No	0 No
15	Auto	\$System	Automatic mode	Yes	No	0 No
16	OutputValve	Reactor	Output valve	No	No	0 No
17	TransferPump	Reactor	Transfer pump	No	No	0 No
18	Ejector	Conveyor	Barrel ejector	No	No	0 No

此文件由关键字及其属性以及“标记名字典”中的数据组成，这些数据在关键字属性下方按列顺序进行排列。

请注意示例 Excel 电子表格中的 :MemoryDisc 关键字。此关键字标识从“标记名字典”中导出的内存离散标记。内存离散标记的属性出现在电子表格相同的行、单独的列中。例如，Logged 属性列显示是否记录内存离散标记的数据。

关键字与属性行的正下方是导出的标记及其关联的属性。在 Excel 电子表格示例中，OutputValve 是不记录数据的内存离散标记。

您可以使用支持 .csv 文件格式的任何程序来查看或编辑 DBDump 创建的导出文件。通常使用 Excel，这是因为它的分栏电子表格格式使得标记数据更便于组织。但是如果您偏向于使用它原来的逗号分隔字符串格式来查看或编辑文件的内容，则也可以使用 Microsoft “记事本”。

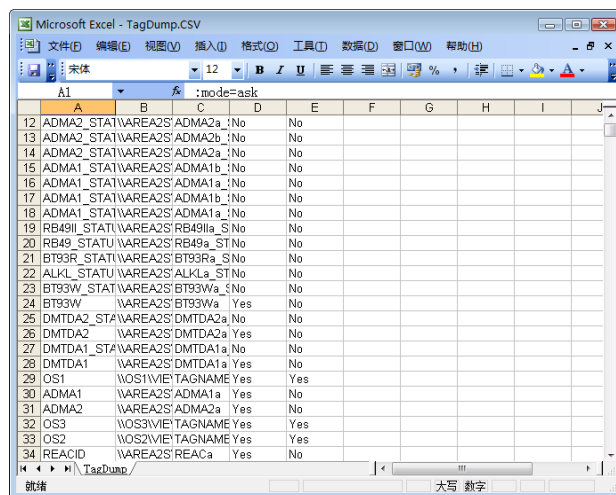
## 导入标记定义

您可以使用“应用程序管理器”中的 DBLoad 实用程序，将 .csv 标记定义文件导入应用程序的“标记名字典”。您可以导入原来使用 DBDump 实用程序创建的定义文件。您也可以创建自己的导入文件。

您还可以使用 DBLoad 实用程序替代 InTouch TemplateMaker 来创建 SuperTag 实例。如需有关详细信息，请参阅第 96 页的“创建 SuperTag 实例”。

## 标记名字典导入文件格式

您可以使用支持 .csv 文件格式的任何程序来手工创建 DBLoad 导入文件。如果使用 **Excel** 创建导入文件，则每一项都放在单独的电子表格单元格中。这使得它更便于阅读，并且可以减少出错的可能。



如需有关创建导入文件的详细信息，请参阅第 67 页的“创建导入文件模板”。

DBLoad 导入文件包含一组关键字，它们将“访问名”、报警组以及标记数据在文件中组织起来。

- 所有的关键字前面都有一个冒号 (:)。
- 如果要续行, 请在行尾输入一个反斜杠 (\)。
- 要输入注释, 请在前面加上一个分号 (;)。

下表列出 DBLoad 导入文件中的关键字。表格中按使用 DBDump 创建文件时指定关键字的顺序列出这些关键字。但是您可以在文件中按任何顺序指定关键字。

关键字	描述
:mode	指定将 DBLoad 文件的内容导入应用程序的“标记名字典”时如何处理重复的标记记录。
:IOAccess	为 InTouch 应用程序定义的访问名。
:AlarmGroup	为 InTouch 应用程序定义的报警组。
:MemoryDisc	内存离散标记。
:IODisc	I/O 离散标记。
:MemoryInt	内存整型标记。

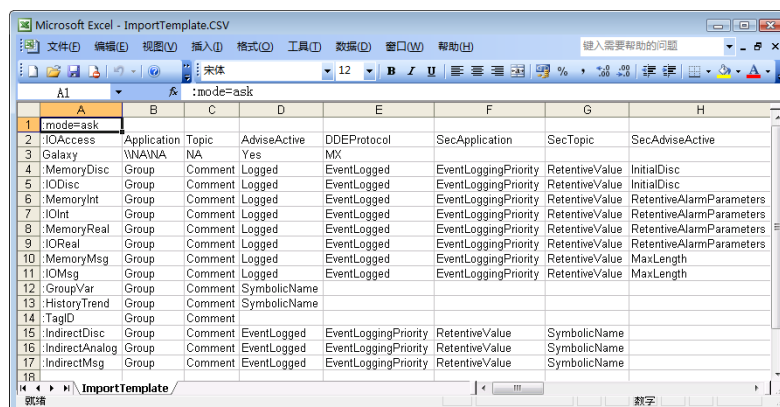
关键字	描述
:IOInt	I/O 整型标记。
:MemoryReal	内存实型标记。
:IOReal	I/O 实型标记。
:MemoryMsg	内存消息标记。
:IOMsg	I/O 消息标记。
:GroupVar	“组变量”标记。
:HistoryTrend	“历史趋势”标记。
:TagID	“标记 ID”标记
:IndirectDisc	间接离散标记。
:IndirectAnalog	间接模拟标记。
:IndirectMsg	间接消息标记。

每个关键字都包含指定“访问名”、报警组以及标记的属性的一组关联属性。例如，:IOAccess 关键字包含一组属性，用于指定应用程序、主题以及通讯协议，这些都是每个 InTouch “访问名”的属性。

## 创建导入文件模板

您可以使用任何支持 .csv 文件格式的应用程序来手工创建“标记名字典”导入文件。但是创建整个导入文件可能费时又容易出错。将现有的 .csv 文件用作模板则既快捷又可靠。

下图显示 DBDump 创建的模板导入文件。图中显示从某个既没有窗口也没有标记的 InTouch 应用程序中创建的文件。产生的文件只包含所需的關鍵字和没有标记数据的属性。



在创建模板之后，您可以随后在确定标记类型的关键字下方手工添加标记数据。在与标记类型关键字关联的相应属性列中，插入标记的属性。

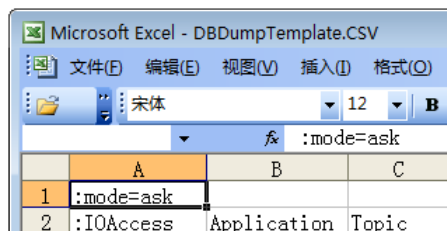
#### 要创建模板导入文件

- 1 打开“应用程序管理器”。
- 2 创建一个新的 InTouch 应用程序。  
如需有关创建应用程序的步骤的详细信息，请参阅第 17 页的“创建 InTouch 应用程序”。
- 3 从“应用程序管理器”显示的列表中，选择新的应用程序。
- 4 使用 DBDump 实用程序导出应用程序“标记名字典”的内容。  
如需有关导出标记的详细信息，请参阅第 63 页的“导出标记定义”。
- 5 编辑文件以插入要导入的标记数据。

## 设置字典导入文件的操作模式

您必须指定从导入文件将数据加载到应用程序“标记名字典”时，DBLoad 如何处理重复的标记记录。

如果使用通过 DBDump 创建的导入文件模板，则文件的第一行包含 :mode 关键字。例如，您可以将值 ask 指定给 Excel 应用程序 A1 单元格中的 :mode 关键字。



您可以将以下这些值指定给 :mode 关键字：

```
:MODE=REPLACE
:MODE=UPDATE
:MODE=ASK
:MODE=IGNORE
:MODE=TERMINATE
:MODE=TEST
```

#### :MODE=REPLACE

如果遇到重复的标记，则 DBLoad 实用程序删除“标记名字典”中现有的标记，并使用导入文件中同名的标记来替换它。

**:MODE=UPDATE**

如果遇到重复的标记，则 DBLoad 实用程序仅在导入文件中明确指定数据的情况下才会覆盖“标记名字典”中现有的标记定义。“标记名字典”中与该标记关联的其它所有数据都保持不变。

如果字段存在于记录中，并且已输入内容或已通过“:KEYWORD=value”机制进行设置，则这些此字段被视为明确定义的。如果字段未在记录中指定，并且使用“:KEYWORD=”命令重置过关键字，则当前字段值不更新。

下例显示采用更新模式将导入文件加载 / 合并到“标记名字典”时产生的结果：

```
:Mode=update
:Group=Group1
:IODisc,Group,DConversion
Tagname1,Group2,
; Tagname1's Group updated to Group2 only
Tagname2,,
; Tagname2's Group updated to Group1 and the DConversion left as is
Tagname3,,Reverse
; Tagname3's Group updated to Group1 and the DConversion to
  "Reverse"
; the following line "resets" the Group field to its default value
:Group=
; Data field "Group" is reset to its default value
Tagname4,,
; Tagname4 will be left alone
```

如果要更改标记类型且标记正在使用，则标记类型必须兼容。例如，如果应用程序正在使用标记，则现有的历史趋势标记无法更改为“I/O 整型”。同样，如果应用程序中的输入链接正在使用标记，则该标记无法更改为 ReadOnly=yes。由于存在这些限制，运行 DBLoad 实用程序之前，请先更新目标应用程序的使用计数。

### :MODE=ASK

在加载“标记名字典”期间遇到重复标记时，DBLoad 会停止。此时出现**重复名称**对话框，显示一个列表，列出处理重复标记的各个选项。这是缺省导入模式。



用于处理重复项的选项包括：

- 单击**用新信息替换现有信息**，以便将现有的标记记录替换为导入文件中的记录。
- 单击**用新信息更新现有信息**，以便仅在导入文件中明确定义字段的情况下才覆盖现有的标记记录。
- 单击**将名称更改为**，以便将导入标记的名称替换为**重复名称**对话框的方框中所输入的名称。
- 单击**忽略此项**，以忽略标记并继续导入文件的内容。
- 单击**放弃加载**，以取消导入过程。

### :MODE=IGNORE

DBLoad 导入实用程序忽略重复标记，并继续处理导入文件的剩余记录。

### :MODE=TERMINATE

遇到重复标记时，DBLoad 导入操作停止。

### :MODE=TEST

DBLoad 在导入文件中扫描错误，而不尝试将标记定义加载到“标记名字典”。DBLoad 生成一份报告，使用导入文件中的行号与位置指出任何格式错误。

使用 `:mode=test` 运行 DBLoad，以确定导入文件中的任何错误。纠正所有错误之后，在运行 DBLoad 之前，将 `mode` 关键字的值更改为 `:mode=replace` 或 `:mode=update`。

设置访问名与报警组

DBLoad 导入文件包含一些关键字，它们指定 InTouch 应用程序中定义的“访问名”与报警组。

:IOAccess 关键字属性

:IOAccess 关键字标识为 InTouch 应用程序定义的“访问名”。:IOAccess 关键字包含一组属性，用于描述定义的 InTouch “访问名”的特征。

下图显示如何使用 :IOAccess 关键字在 Excel 电子表格中定义“访问名”。各个属性在单独的电子表格列中从左到右进行指定。

:IOAccess		Topic		DDEProtocol		SecTopic		
关键字	属性	属性	属性	属性	属性	属性	属性	
	A	B	C	D	E	F	G	H
1	:mode=ask							
2	:IOAccess	Application	Topic	AdviseActive	DDEProtocol	SecApplication	SecTopic	SecAdviseActive
3	T7	View	T7	Yes	No			
4	M22	RSLINK	M22	Yes	Yes			
5	IOStatus	View	IOstatus	Yes	Yes			
6	Galaxy	WNA/NA	NA	Yes	MX			

Application  
属性

AdviseActive  
属性

SecApplication  
属性

下表显示与 :IOAccess 关键字关联的属性的列表。此表格按照使用模板导入文件（用 DBDump 实用程序创建）时指定的顺序列出各个属性。

字符串位置	属性	可接受的值	缺省值
1	Application	为“访问名”定义的应用程序名。	无
2	Topic	为“访问名”定义的主题名。	无
3	AdviseActive	要从服务器轮询哪些信息 No = 提示所有项 Yes = 只提示激活项	Yes
4	DDEProtocol	为“访问名”定义的通讯协议。 No = Suitelink Yes = DDE MX = 消息交换	No
5	SecApplication	为“访问名”的辅助数据源定义的应用程序名	无
6	SecTopic	为“访问名”的辅助数据源定义的主题名。	无

字符串位置	属性	可接受的值	缺省值
7	SecAdviseActive	何时轮询辅助服务器上存储的信息 NO = 提示所有项 YES = 只提示激活项	无
8	SecDDEProtocol	为“访问名”的辅助数据源定义的通讯协议 NO = Suitelink YES = DDE MX = 消息交换	无
9	FailoverExpression	值为“真”时将访问名切换到辅助数据源的故障转移表达式	无
10	FailoverDeadband	开始向辅助数据源进行故障转移之前的秒数的整数部分，由“访问名”定义。	无
11	DFOFlag	禁用故障转移标帜 Yes = “禁用故障转移标帜”已设置 No = “禁用故障转移标帜”未设置	无
12	FBDFlag	切换回“主数据源”标帜 YES = 故障转移条件消失之后切换回“主数据源” NO = 故障转移条件消失之后不切换回“主数据源”	无
13	FailbackDeadband	故障转移条件消失之后切换回主“访问名”之前的秒数的整数部分	无

### :AlarmGroup 关键字属性

DBLoad 导入文件包含某个关键字，它用于标识为 InTouch 应用程序定义的报警组。:AlarmGroup 关键字包含一组属性，用于描述 InTouch 应用程序的报警组的特征。

下表显示与 :AccessGroup 关键字关联的属性的列表。此表格按以下顺序列出各个属性：使用以 DBDump 实用程序创建的模板导入文件时指定的顺序。

字符串位置	属性	可接受的值	缺省值
1	Group	报警组的名称	\$System



字符串位置	属性	可接受的值	缺省值
2	Comment	指定给报警组的注释 任何文本字符串	无
3	EventLogged	启用或禁用事件记录 Yes 或 On = 启用事件记录 No 或 Off = 禁用事件记录	否
4	EventLoggingPriority	指定给事件的优先级 1 到 999, 0 = 不记录	0
5	LoLoAlarmDisable	禁用或启用 LoLo 报警 0 = 启用 LoLo 报警 1 = 禁用 LoLo 报警	0
6	LoAlarmDisable	禁用或启用 Low 报警 0 = 启用 Low 报警 1 = 禁用 Low 报警	0
7	HiAlarmDisable	禁用或启用 High 报警 0 = 启用 High 报警 1 = 禁用 High 报警	0
8	HiHiAlarmDisable	禁用或启用 HiHi 报警 0 = 启用 HiHi 报警 1 = 禁用 HiHi 报警	0
9	MinDevAlarmDisable	禁用或启用 “副偏差” 报警 0 = 启用 “副偏差” 报警 1 = 禁用 “副偏差” 报警	0
10	MajDevAlarmDisable	禁用或启用 “主偏差” 报警 0 = 启用 “主偏差” 报警 1 = 禁用 “主偏差” 报警	0
11	RocAlarmDisable	禁用或启用 “变化率” 报警 0 = 启用 ROC 报警 1 = 禁用 ROC 报警	0

字符串位置	属性	可接受的值	缺省值
12	DSCAlarmDisable	禁用或启用离散报警 0 = 启用离散报警 1 = 禁用离散报警	0
13	LoLoAlarmInhibitor	用于抑制 LoLo 报警的标记的名称 标记引用：任何离散或模拟标记	无
14	LoAlarmInhibitor	用于抑制 Low 报警的标记的名称 标记引用：任何离散或模拟标记	无
15	HiAlarmInhibitor	用于抑制 High 报警的标记的名称 标记引用：任何离散或模拟标记	无
16	HiHiAlarmInhibitor	用于抑制 HiHi 报警的标记的名称 标记引用：任何离散或模拟标记	无
17	MinDevAlarmInhibitor	用于抑制”副偏差“报警的标记的名称 标记引用：任何离散或模拟标记	无
18	MajDevAlarmInhibitor	用于抑制”主偏差“报警的标记的名称 标记引用：任何离散或模拟标记	无
19	RocAlarmInhibitor	用于抑制“变化率”报警的标记的名称 标记引用：任何离散或模拟标记	无
20	DSCAlarmInhibitor	用于抑制离散报警的标记的名称 标记引用：任何离散或模拟标记	无

## 定义标记类型关键字与属性

标记记录从标识标记类型的关键字行开始。每个标记关键字都包含一组唯一的属性，用于指定同标记类型关联的数据的特征。

在下例中，:IODisc 关键字标识 I/O 离散标记类型。关键字行中其余的值确定与 I/O 离散标记关联的数据的属性。本例使用“记事本”按原始的逗号分隔字符串格式显示该文件的内容。

```
:IODisc,Group,Comment,Logged,EventLogged,
EventLoggingPriority,RetentiveValue,InitialDis,
OffMsg,OnMsg,AlarmState,AlarmPri,DConversion,
```

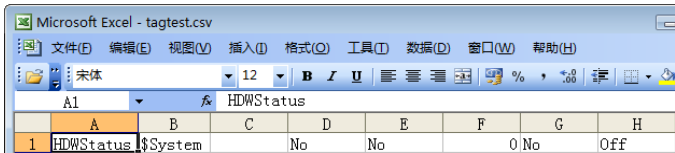
AccessName,ItemUseTagname,ItemName,ReadOnly,  
AlarmComment,AlarmAckModel,DSCAlarmDisable,  
DSCAlarmInhibitor,SymbolicName

标记类型关键字行下面有一些单独的行，这些行使用一组属性值指定该类型的标记。在下例中，导入文件中的 HDWStatus 标记属于 “I/O 离散” 标记类型。

"HDWStatus","\$System","",No,No,0,No,Off","",",,1,Direct,"HistaViewstr",No,"Status",No,"",0,0,"","

记录使用引号确定空字符串。

下图在 Excel 电子表格中显示相同的导入文件数据。由于导入文件中没有指定标记注释，因此 **Comment** 单元格为空白。



标记关键字属性

下表列出与 InTouch 标记关键字关联的所有属性。表格包含一些列，描述与每个标记属性关联的数据类型及其缺省值。

只要跟随的标记数据与相应的属性相匹配，这些标记属性可以在 DBLoad 导入文件中按任何顺序指定。例如，如果在 Excel 导入文件中插入 :IODisc 关键字，则所有的 I/O 离散标记的工程单位都必须放在与 EngUnits 属性相同的 Excel 列中。

属性	可接受的值	缺省值
AccessName	指定给标记的 InTouch “访问名”	无
AlarmAckModel	报警确认模型 整型 0 = 条件 1 = 面向事件 2 = 扩展的摘要	0
AlarmComment	指定给标记的报警注释 文本字符串	无
AlarmDevDeadband	标记偏差报警死区 实型	无
AlarmPri	指定给标记的报警优先级 1 到 999	1

属性	可接受的值	缺省值
AlarmState	标记报警状态 On、Off 或无	无
AlarmValueDeadband	标记报警死区 实型	0
Comment	指定给标记的注释 文本字符串	无
Conversion	标记值转换 “线性”或“平方根”	线性
Deadband	指定给标记的值死区 实型	0
DevTarget	标记偏差目标值 实型	0
DSCAlarmDisable	禁用或启用离散报警 0 = 启用离散报警 1 = 禁用离散报警	0
DSCAlarmInhibitor	用于抑制离散报警的标记的名称	无
EngUnits	指定给标记的工程单位 文本字符串	无
EventLogged	启用或禁用事件记录 Yes 或 On = 启用事件记录 No 或 Off = 禁用事件记录	否
EventLogging	启用或禁用标记事件记录 No 或 Off = 禁用记录功能 Yes 或 On = 启用记录功能	否
EventLoggingPriority	指定给事件的优先级 1 到 999, 0 = 不记录	0
Group	标记所属报警组的名称	\$System
HiAlarmDisable	禁用或启用 High 报警 0 = 启用 High 报警 1 = 禁用 High 报警	0

属性	可接受的值	缺省值
HiAlarmInhibitor	用于抑制 High 报警的标记的名称 任何离散或模拟标记	无
HiAlarmPri	指定给 High 报警的优先级 1 到 999	1
HiAlarmState	启用或禁用 High 报警 No 或 Off = 禁用 Yes 或 On = 启用	否
HiAlarmValue	指定给标记的 High 报警点 实型	0
HiHiAlarmDisable	禁用或启用 HiHi 报警 0 = 启用 HiHi 报警 1 = 禁用 HiHi 报警	0
HiHiAlarmInhibitor	用于抑制 HiHi 报警的标记的名称 任何离散或模拟标记	无
HiHiAlarmPri	指定给 HiHi 报警的优先级 1 到 999	1
HiHiAlarmState	启用或禁用 HiHi 报警 No 或 Off = 禁用 Yes 或 On = 启用	否
HiHiAlarmValue	指定给标记的 HiHi 报警点 实型	0
InitialDisc	指定给离散标记的初始值 0、Off、False 或 No = 关闭 1、On、True 或 Yes = 打开	0
InitialMessage	初始标记消息 文本字符串	无
InitialValue	指定给标记的初始值 实型	0
ItemName	指定给标记的项目的名称 文本字符串	无

属性	可接受的值	缺省值
ItemUseTagName	启用或禁用将标记名用作项目名选项 No 或 False = 禁用 Yes 或 True = 启用	否
LoAlarmDisable	禁用或启用 Low 报警 0 = 启用 Low 报警 1 = 禁用 Low 报警	0
LoAlarmInhibitor	用于抑制 Low 报警的标记的名称 任何离散或模拟标记	无
LoAlarmPri	指定给 Low 报警的优先级 1 到 999	1
LoAlarmState	启用或禁用 Low 报警 No 或 Off = 禁用 Yes 或 On = 启用	否
LoAlarmValue	指定给标记的 Low 报警点 实型	0
LogDeadband	指定给标记的记录死区 实型	0
Logged	启用或禁用标记值记录 No 或 Off = 禁用记录功能 Yes 或 On = 启用记录功能	否
LoLoAlarmDisable	禁用或启用 LoLo 报警 0 = 启用 LoLo 报警 1 = 禁用 LoLo 报警	0
LoLoAlarmInhibitor	用于抑制 LoLo 报警的标记的名称 任何离散或模拟标记	无
LoLoAlarmPri	指定给 LoLo 报警的优先级 1 到 999	1
LoLoAlarmState	启用或禁用 LoLo 报警 No 或 Off = 禁用 Yes 或 On = 启用	否

属性	可接受的值	缺省值
LoLoAlarmValue	指定给标记的 LoLo 报警点 实型	0
MajDevAlarmDisable	禁用或启用 “主偏差” 报警 0 = 启用 “主偏差” 报警 1 = 禁用 “主偏差” 报警	0
MajDevAlarmInhibitor	用于抑制” 主偏差 “报警的标记的名称 任何离散或模拟标记	无
MajorDevAlarmPri	指定给 “主偏差” 报警的优先级 1 到 999	1
MajorDevAlarmState	启用或禁用 “主偏差” 报警 No 或 Off = 禁用 Yes 或 On = 启用	否
MajorDevAlarmValue	指定给标记的主偏差报警百分比 实型	0
MaxEU	指定给标记的最大工程单位值 实型	32767
MaxLength	最大消息长度 实型	131
MaxRaw	指定给标记的最大原始值 实型	32767
MaxValue	指定给标记的最大值 实型	32767
MinDevAlarmDisable	禁用或启用 “副偏差” 报警 0 = 启用 “副偏差” 报警 1 = 禁用 “副偏差” 报警	0
MinDevAlarmInhibitor	用于抑制” 副偏差 “报警的标记的名称 任何离散或模拟标记	无
MinEU	指定给标记的最小工程单位值 实型	-32768
MinorDevAlarmPri	指定给 “副偏差” 报警的优先级 1 到 999	1

属性	可接受的值	缺省值
MinorDevAlarmState	启用或禁用副偏差报警 No 或 Off = 禁用 Yes 或 On = 启用	否
MinorDevAlarmValue	指定给标记的副偏差报警百分比 实型	0
MinRaw	指定给标记的最小原始值 实型	-32768
MinValue	指定给标记的最小值 实型	-32768
OffMsg	离散标记 “关闭” 消息 文本字符串	无
OnMsg	离散标记 “打开” 消息 文本字符串	无
ReadOnly	标记值只读或可读写 Yes = 只读 No = 可读写	否
RetentiveAlarmParameters	启用或禁用标记 “保留” 参数 No 或 Off = 禁用 Yes 或 On = 启用	否
RetentiveValue	启用或禁用标记 “保留值” 0、Off、False 或 No = 禁用 1、On、True 或 Yes = 启用	否
RocAlarmDisable	禁用或启用 “变化率” 报警 0 = 启用 ROC 报警 1 = 禁用 ROC 报警	0
RocAlarmInhibitor	用于抑制 “变化率” 报警的标记的名称 任何离散或模拟标记	无
ROCArmPri	指定给 “变化率” 报警的优先级 1 到 999	1



属性	可接受的值	缺省值
ROCArmState	启用或禁用“变化率”报警 No 或 Off = 禁用 Yes 或 On = 启用	否
ROCArmValue	以百分比表示的标记值变化 实型	0
ROCTimeBase	计算变化率的测量周期 Sec、Min 或 Hr	Min

### :MemoryDisc 关键字属性

DBLoad 导入文件包含 :MemoryDisc 关键字，用于定义可导入到“标记名字典”的内存离散标记。下表列出与内存离散标记属性关联的 :MemoryDisc 关键字的属性。

此表格按使用 DBDump 创建导入文件时指定的 :MemoryDisc 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串 位置	属性
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	InitialDisc
8	OffMsg
9	OnMsg
10	AlarmState
11	AlarmPri
12	AlarmComment
13	AlarmAckModel
14	DSCAlarmDisable
15	DSCAlarmInhibitor

### :IODisc 关键字属性

DBLoad 导入文件包含 :IODisc 关键字，用于定义可导入到“标记名字典”的 I/O 离散标记。下表列出与 I/O 离散标记属性关联的 :IODisc 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IODisc 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串 位置	属性
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	InitialDisc
8	OffMsg
9	OnMsg
10	AlarmState
11	AlarmPri
12	Conversion
13	AccessName
14	ItemUseTagname
15	ItemName
16	ReadOnly
17	AlarmComment
18	AlarmAckModel
19	DSCAlarmDisable
20	DSCAlarmInhibitor

### :MemoryInt 关键字属性

DBLoad 导入文件包含 :MemoryInt 关键字，用于定义可导入到“标记名字典”的内存整型标记。下表列出与内存整型标记属性关联的 :MemoryInt 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :MemoryInt 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串 位置	属性
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue
12	MinValue
13	MaxValue
14	Deadband
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHiAlarmPri

字符串 位置	属性
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCArmState
36	ROCArmValue
37	ROCArmPri
38	ROCTimeBase
39	AlarmComment
40	AlarmAckModel
41	LoLoAlarmDisable
42	LoAlarmDisable
43	HiAlarmDisable
44	HiHiAlarmDisable
45	MinDevAlarmDisable
46	MajDevAlarmDisable
47	RocAlarmDisable
48	LoLoAlarmInhibitor
49	LoAlarmInhibitor
50	HiAlarmInhibitor
51	HiHiAlarmInhibitor
52	MinDevAlarmInhibitor
53	MajDevAlarmInhibitor
54	RocAlarmInhibitor

### :IOInt 关键字属性

DBLoad 导入文件包含 :IOInt 关键字，用于定义可导入到“标记名字典”的 I/O 整型标记。下表列出与 I/O 整型标记属性关联的 :IOInt 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IOInt 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串 位置	属性
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue
12	MinEU
13	MaxEU
14	Deadband
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri

字符串 位置	属性
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHiAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCArmState
36	ROCArmValue
37	ROCArmPri
38	ROCTimeBase
39	AlarmComment
39	MinRaw
40	MaxRaw
41	Conversion
42	AccessName
43	ItemUseTagname
44	ItemName
45	ReadOnly
46	AlarmComment
47	AlarmAckModel
48	LoLoAlarmDisable
49	LoAlarmDisable
50	HiAlarmDisable
51	HiHiAlarmDisable
52	MinDevAlarmDisable

字符串 位置	属性
53	MajDevAlarmDisable
54	RocAlarmDisable
55	LoLoAlarmInhibitor
56	LoAlarmInhibitor
57	HiAlarmInhibitor
58	HiHiAlarmInhibitor
59	MinDevAlarmInhibitor
60	MajDevAlarmInhibitor
61	RocAlarmInhibitor

### :MemoryReal 关键字属性

DBLoad 导入文件包含 :MemoryReal 关键字，用于定义可导入到“标记名字典”的内存实型标记。下表列出与内存整型标记属性关联的 :MemoryReal 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :MemoryReal 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串 位置	属性
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue
12	MinValue
13	MaxValue

字符串 位置	属性
14	Deadband
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHiAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCArmState
36	ROCArmValue
37	ROCArmPri
38	ROCTimeBase
39	AlarmComment
40	AlarmAckModel
41	LoLoAlarmDisable
42	LoAlarmDisable



字符串 位置	属性
43	HiAlarmDisable
44	HiHiAlarmDisable
45	MinDevAlarmDisable
46	MajDevAlarmDisable
47	RocAlarmDisable
48	LoLoAlarmInhibitor
49	LoAlarmInhibitor
50	HiAlarmInhibitor
51	HiHiAlarmInhibitor
52	MinDevAlarmInhibitor
53	MajDevAlarmInhibitor
54	RocAlarmInhibitor

### :IOReal 关键字属性

DBLoad 导入文件包含 :IOReal 关键字，用于定义可导入到“标记名字典”的 I/O 实型标记。下表列出与 I/O 实型标记属性关联的 :IOReal 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IOReal 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串 位置	属性
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband

字符串 位置	属性
10	EngUnits
11	InitialValue
12	MinEU
13	MaxEU
14	Deadband
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHiAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCArmState
36	ROCArmValue
37	ROCArmPri
38	ROCTimeBase

字符串 位置	属性
39	MinRaw
40	MaxRaw
41	Conversion
42	AccessName
43	ItemUseTagname
44	ItemName
45	ReadOnly
46	AlarmComment
47	AlarmAckModel
48	LoLoAlarmDisable
49	LoAlarmDisable
50	HiAlarmDisable
51	HiHiAlarmDisable
52	MinDevAlarmDisable
53	MajDevAlarmDisable
54	RocAlarmDisable
55	LoLoAlarmInhibitor
56	LoAlarmInhibitor
57	HiAlarmInhibitor
58	HiHiAlarmInhibitor
59	MinDevAlarmInhibitor
60	MajDevAlarmInhibitor
61	RocAlarmInhibitor

### :MemoryMsg 关键字属性

DBLoad 导入文件包含 :MemoryMsg 关键字，用于定义可导入到“标记名字典”的内存消息标记。下表列出与内存消息标记属性关联的 :MemoryMsg 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :MemoryMsg 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串位置	属性
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	MaxLength
8	InitialMessage
9	AlarmComment

:IOMsg 关键字属性

DBLoad 导入文件包含 :IOMsg 关键字，用于定义可导入到“标记名字典”的 I/O 消息标记。下表列出与 I/O 消息标记属性关联的 :IOMsg 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IOMsg 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串位置	属性
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	MaxLength
8	InitialMessage
9	AccessName
10	ItemUseTagname

字符串 位置	属性
11	ItemName
12	ReadOnly
13	AlarmComment

### :GroupVar 关键字属性

DBLoad 导入文件包含 :GroupVar 关键字，用于定义可导入到“标记名字典”的“组变量”标记。下表列出与“组变量”标记属性关联的 :GroupVar 关键字属性。

**备注** InTouch “组变量”标记已废弃。包含 :GroupVar 关键字仅仅是为了支持旧有的应用程序。

此表格按使用 DBDump 创建导入文件时指定的 :GroupVar 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串 位置	属性
1	Group
2	Comment

### :HistoryTrend 关键字属性

DBLoad 导入文件包含 :HistoryTrend 关键字，用于定义可导入到“标记名字典”的 HistTrend 标记。下表列出与 HistTrend 标记属性关联的 :HistoryTrend 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :HistoryTrend 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串 位置	属性
1	Group
2	Comment

### :TagID 关键字属性

DBLoad 导入文件包含 :TagID 关键字，用于定义可导入到“标记名字典”的“标记 ID”标记。下表列出与“标记 ID”标记属性关联的 :TagID 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :TagID 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串位置	属性
1	Group
2	Comment

**:IndirectDisc 关键字属性**

DBLoad 导入文件包含 :IndirectDisc 关键字，用于定义可导入到“标记名字典”的间接离散标记。下表列出与间接离散标记属性关联的 :IndirectDisc 关键字属性。

此表按使用 DBDump 创建导入文件时指定 :IndirectDisc 关键字属性的顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串位置	属性
1	Group
2	Comment
3	EventLogging
4	EventLoggingPriority
5	RetentiveValue

**:IndirectAnalog 关键字属性**

DBLoad 导入文件包含 :IndirectAnalog 关键字，用于定义可导入到“标记名字典”的间接模拟标记。下表列出与间接模拟标记属性关联的 :IndirectAnalog 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IndirectAnalog 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串位置	属性
1	Group
2	Comment
3	EventLogging
4	EventLoggingPriority

字符串位置	属性
5	RetentiveValue

:IndirectMsg 关键字属性

DBLoad 导入文件包含 :IndirectMsg 关键字，用于定义可导入到“标记名字典”的间接消息标记。下表列出与间接消息标记属性关联的 :IndirectMsg 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IndirectMsg 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅第 75 页的“标记关键字属性”。

字符串位置	属性
1	Group
2	Comment
3	EventLogging
4	EventLoggingPriority
5	RetentiveValue

在导入文件中使用空字符串

对于字典导入文件而言，包含空字符串的字段与没有数据的字段之间是有区别的。可以指定空字符串的关键字属性包括：

Comment	EngUnits	OffMsg
InitialMessage	OnMsg	Application
ItemName	Topic	

在下例中，空字符串使用英文双引号 ( " ") 表示：

```
:Comment="HI"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,"",$System
```

其中：

Tagname1 的 Comment 字段的值为 Hi， Tagname2 的 Comment 字段的值为空注释。

Microsoft Excel 在保存文件时忽略表示空字符串的英文双引号，这会产生以下结果：

```
:Comment="HI"
```

```
:MemoryDisc,Comment,Group
Tagname1,, $System
Tagname2,, $System
```

要确保空字符串在 Excel 中能起作用，请在单元格中输入一个空格作为属性值。

## 使用字段的缺省值

您可以使用关键字给某条记录的特定字段设置缺省值。缺省值是该标记类型的原始 InTouch 值。例如，某个内存离散标记使用 **Group=\$System**、**EventLogging=Off**、**InitialValue=Off** 作为缺省值。

例如：

```
:KEYWORD=value
```

这就为所有后续的数据记录设置了所引用字段的缺省值。使用此功能可以为许多记录中保持不变的字段设置缺省值。如果某个字段定义了缺省值，则在记录中没有该值的数据时会使用缺省值。

例如，如果设置 **:GROUP=Reactor\_Site**，则 **GROUP** 列中包含空白项的所有标记都会指定给 **Reactor\_Site** “报警组”。如果标记的 **GROUP** 列中输入了 **\$System**，则此标记仍指定给 **\$System** “报警组”。

您可以通过忽略方程中的值将某个单独的关键字重置为原始缺省值。例如，**:GROUP=**。

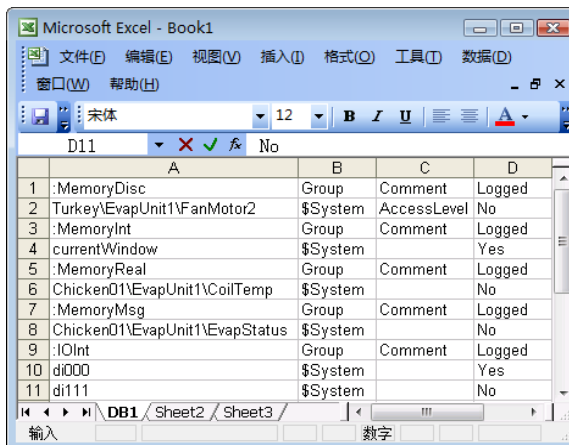
要重置所有关键字，请使用 **:RESET** 命令。此命令不含参数，并且会影响使用此命令之后处理的文件中的所有项目。

## 创建 SuperTag 实例

您可以使用“应用程序管理器”中的 **DBLoad** 实用程序来创建 **SuperTag**。不过，创建的 **SuperTag** 实例不反映到 **TemplateMaker** 中的 **SuperTag** 模板定义。



您必须使用有效的 SuperTag 格式，并且 SuperTag 实例数据记录必须以该标记类型的有效关键字开头。例如：



	A	B	C	D
1	:MemoryDisc	Group	Comment	Logged
2	Turkey\EvapUnit1\FanMotor2	\$System	AccessLevel	No
3	:MemoryInt	Group	Comment	Logged
4	currentWindow	\$System		Yes
5	:MemoryReal	Group	Comment	Logged
6	Chicken01\EvapUnit1\CoilTemp	\$System		No
7	:MemoryMsg	Group	Comment	Logged
8	Chicken01\EvapUnit1\EvapStatus	\$System		No
9	:IOInt	Group	Comment	Logged
10	di000	\$System		Yes
11	di111	\$System		No

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\

如果使用无效格式，则会出现一条错误消息。

导入包含 SuperTag 实例的 CSV 文件时，这些实例自动添加到“标记字典”中，并且可以立即在动画链接与 InTouch QuickScript 中使用。

## 使用 DBLoad 导入标记定义

使用 DBLoad 导入文件的内容时，所有标记定义都导入到所选 InTouch 应用程序的“标记字典”中。

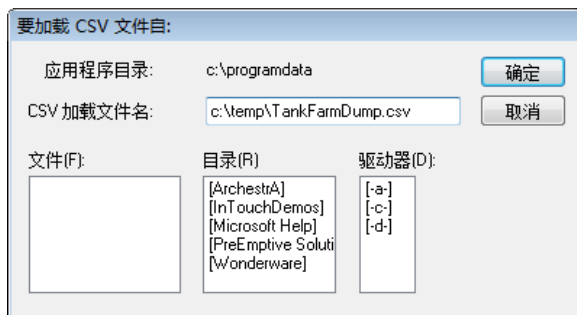
如果导入失败，则出现一条消息，说明失败的原因。错误消息写入 Logger 中。

### 要将标记定义导入 InTouch 应用程序

- 1 关闭 WindowMaker 与 WindowViewer。
- 2 对于要从导入文件中加载标记定义的“标记字典”，备份其所属的应用程序。
- 3 启动应用程序管理器。
- 4 从列表中选择其“标记字典”将接收导入的标记定义的应用程序。



- 5 单击 **DBLoad** 图标。此时会出现一条消息，要求确认是否已备份 InTouch 应用程序。
- 6 单击 **是** 以确认应用程序已备份。此时出现 **要加载 CSV 文件自：** 对话框。



- 7 在 **CSV 加载文件名** 框中，找到并选择要导入的文件。
- 8 单击 **确定**。

根据 **DBLoad** 是将新的还是现有的标记定义导入“标记名字典”中，下一个步骤可能有所不同。

- 如果导入新的标记定义，则将新标记数据加载到应用程序的“标记字典”。此时出现一条消息，确认已成功加载并合并数据。
- 如果导入现有的标记定义，**:mode** 关键字设置为 **:mode=ask** 并且导入文件包含重复标记，则导入将停止。此时显示用于处理重复标记的选项，您也可以取消导入。如需有关这些关键字选项的详细信息，请参阅第 68 页的“设置字典导入文件的操作模式”。

## 导入窗口

从现有的 InTouch 应用程序将窗口导入当前应用程序时，由于可以复用先前创建的窗口、对象以及窗口脚本，因此可以缩短开发时间。

要能够导入窗口，必须先将应用程序转换为 InTouch HMI 软件的当前版本。

缺省条件下，将为与导入窗口关联的标记创建占位符。在导入之后，您可以将占位符转换为本地标记或远程标记引用。如需有关详细信息，请参阅第 106 页的“导入的窗口与脚本中的标记占位符”。如果目标应用程序中已经存在关联的标记，则可以在导入期间选择使用这些标记。

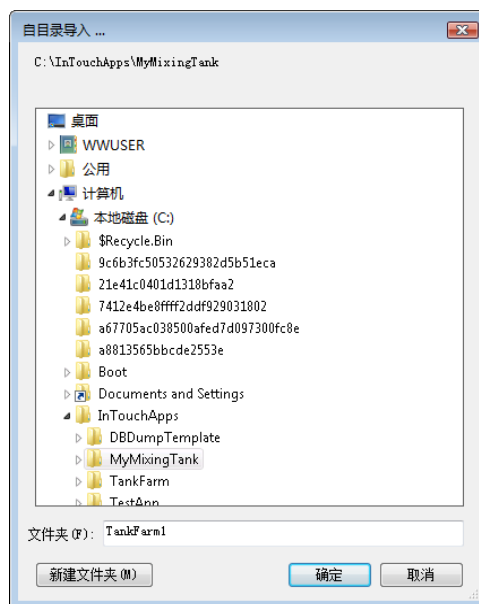
导入包含 **SmartSymbol** 的窗口并且选择使用现有的标记时，InTouch HMI 仍然为恢复的符号保留占位符，即便标记存在于目标应用程序中也是如此。

从包含 SuperTag 的应用程序中导入窗口时，只有窗口中实际使用的 SuperTag 实例才会导入新应用程序。整个 SuperTag 模板结构不会导入。例如，如果应用程序中定义了数百个 SuperTag 成员标记，其中只有 50 个用在导入的窗口中，则仅导入这 50 个标记。

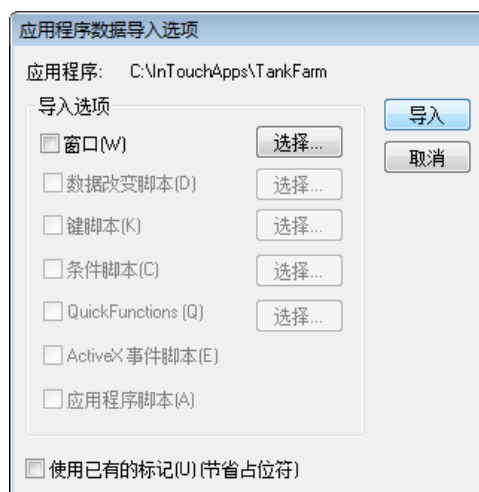
**重要** 如果使用除导入或导出之外的方法移动 InTouch 窗口文件，则可能破坏应用程序“标记名字典”的内容。

### 要导入窗口

- 1 关闭当前应用程序中的所有窗口。
- 2 在文件菜单上，单击导入。此时出现自目录导入对话框。



- 3 选择包含要导入窗口的应用程序所在的文件夹。
- 4 单击确定。此时出现应用程序数据导入选项对话框。



- 5 选择**窗口**复选框，然后单击**选择**以选择要导入的各个单独窗口。
- 6 如果应用程序中已经存在与导入窗口关联的标记，并且您希望使用这些标记而不是占位符，请选择**使用已有的标记（节省占位符）**。
- 7 单击**导入**。
- 8 将占位符标记转换为本地标记或远程标记引用。如需有关详细信息，请参阅第 100 页的“转换导入窗口的占位符标记”。
- 9 如果导入的窗口包含一个或多个向导，双击每个向导以打开其属性面板。如果导入的窗口包含一个或多个 SmartSymbol，编辑每个 SmartSymbol 并创建新的实例。

转换导入窗口的占位符标记

从当前应用程序导入或导出窗口或 QuickScript 时，与该窗口或 QuickScript 关联的所有标记都会随窗口一起传输。但是，这些标记并不添加到新应用程序的“标记名字典”。相反，除非在导入时选择了**节省占位符**选项，否则标记自动标记为占位符标记。您必须转换这些占位符标记，如果需要，请在新应用程序的“标记名字典”中定义它们。

要转换窗口的标记

- 1 在 WindowMaker 中打开窗口。
- 2 按 F2 以选择窗口中的所有对象。
- 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。



- 4 单击**转换**。此时出现**转换**对话框。



- 5 转换这些标记。
- 单击**本地**，以便将占位符转换为本地标记。程序会提示您在“标记名字典”中定义每个标记。
  - 单击**远程**，以便将占位符标记转换为远程标记引用。此时出现**访问名**对话框。选择“访问名”，然后单击**关闭**。
- 转换之后，**替换标记名**对话框中显示新的标记。

替换标记名... 当前为第 1 项，总共 13 项

当前名:	要求类型	新名:
Auto	离散	PLC1:Auto
ConcPump	离散	PLC1:ConcPump
ConcValve	离散	PLC1:ConcValve
Mixer	离散	PLC1:Mixer
OutputValve	离散	PLC1:OutputValve
PassWord	字符串	PLC1:PassWord
ProdLevel	模拟	PLC1:ProdLevel
ReactLevel	模拟	PLC1:ReactLevel
ReactTemp	模拟	PLC1:ReactTemp
StreamValve	模拟	PLC1:StreamValve

确定

取消

索引(I)

转换(C)

替换(R)

上一页(P)

下一页(N)

- 6 单击**确定**。

## 导出窗口

您可以导出应用程序窗口，以便：

- 创建或维护一个包含所有窗口的库应用程序。
- 在另一个应用程序中创建远程标记引用。

要能够导出窗口，必须先将应用程序转换为 InTouch HMI 软件的当前版本。

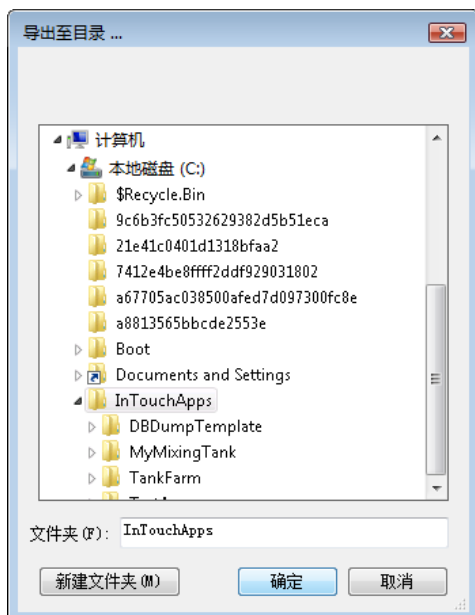
导出窗口时，与该窗口关联的所有对象与动画链接都将导出。与窗口中的对象关联的标记转换为占位符标记，以防止覆盖目标应用程序中现有的标记。如需有关转换占位符标记的详细信息，请参阅第 100 页的“转换导入窗口的占位符标记”。

**重要** 如果使用除导入或导出之外的方法移动 InTouch 窗口文件，则可能破坏应用程序“标记名字典”的内容。

### 要导出窗口

- 1 关闭当前应用程序中的所有窗口。

- 2 在文件菜单上，单击**导出窗口**。此时出现**导出至目录**对话框。



- 3 选择要将窗口导出到其中的应用程序所在的文件夹。
- 4 单击**确定**。此时出现**要导出的窗口**对话框。



- 5 选择要导出的窗口。
- 6 单击**确定**。

如果出现问题，此时出现**导出操作出现问题**对话框。单击要执行的操作对应的选项，然后单击**确定**。

## 导入脚本

通过将现有 **QuickScript** 从 **InTouch** 应用程序导入当前应用程序，可以节省开发时间。

要能够导入脚本，必须先将应用程序转换为 **InTouch HMI** 软件的当前版本。

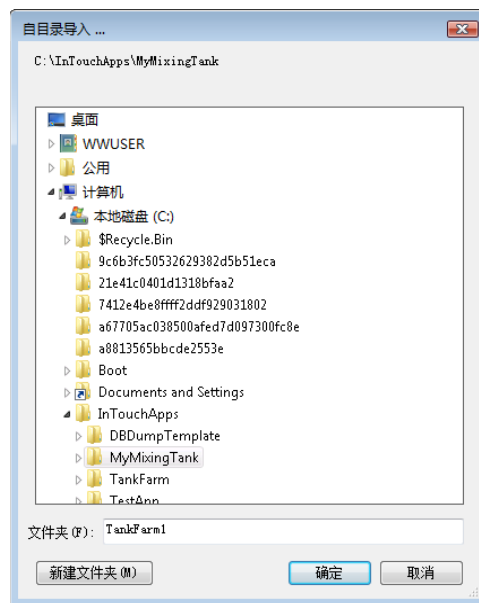
缺省条件下，将为与导入的 **QuickScript** 关联的标记创建占位符。在导入之后，您可以将占位符转换为本地标记或远程标记引用。如需有关详细信息，请参阅第 106 页的“导入的窗口与脚本中的标记占位符”。如果目标应用程序中已经存在关联的标记，则可以在导入期间选择使用这些标记。

要导入窗口脚本，必须导入整个窗口。

要让导入的“**ActiveX** 事件”脚本在目标应用程序中正确运行，必须还要在目标应用程序中使用原先为其创建脚本的相同的 **ActiveX** 控件与事件，并且必须将它加载到内存中。如果关闭包含 **ActiveX** 控件的窗口，则与该窗口关联的任何脚本（“**ActiveX** 事件”脚本或 **QuickScript**）都无法正确运行。

### 要导入 QuickScript

- 1 关闭当前应用程序中的所有窗口。
- 2 在文件菜单上，单击**导入**。此时出现**自目录导入**对话框。



- 3 选择包含要导入脚本的应用程序所在的文件夹。

- 4 单击**确定**。此时出现**应用程序数据导入选项**对话框。



- 5 选择要导入的 QuickScript 类型旁边的复选框，然后单击**选择**以选择要导入的各个单独脚本。

**备注** 要导入窗口脚本，必须导入整个窗口。如需有关详细信息，请参阅第 98 页的“导入窗口”。

- 6 如果应用程序中已经存在与导入脚本关联的标记，并且您希望使用这些标记而不是占位符，请选择**使用已有的标记（节省占位符）**。
- 7 单击**导入**。如果应用程序中包含使用相同名称的脚本，则程序会提示您选择覆盖、忽略或重命名。
- 8 将占位符标记转换为本地标记或远程标记引用。如需有关详细信息，请参阅第 104 页的“转换导入的脚本中的占位符标记”。

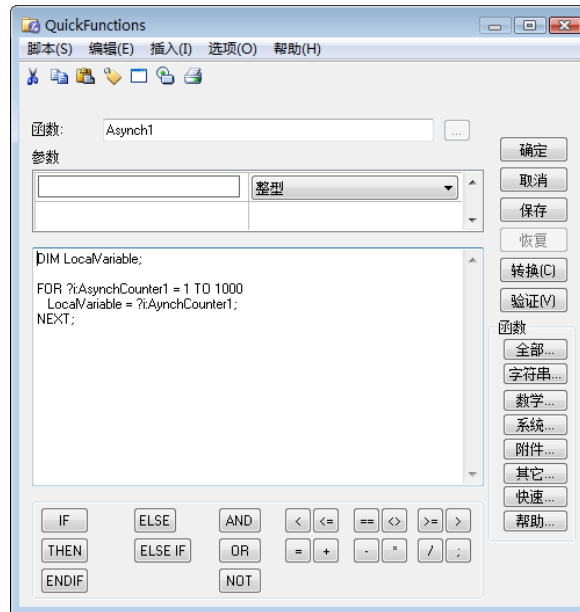
## 转换导入的脚本中的占位符标记

从当前应用程序导入或导出 QuickScript 时，与该 QuickScript 关联的所有标记都会一起传输。但是，这些标记并不添加到新应用程序的“标记名字典”。相反，它们会自动标为占位符标记。您必须转换这些占位符标记，如果需要，请在新应用程序的“标记名字典”中定义它们。



### 要转换导入的脚本中的占位符标记

- 1 在**特别**菜单上，指向**脚本**，然后单击导入的 QuickScript 类型。此时出现 QuickScript 编辑器，显示文件中与所选的脚本类型对应的第一个 QuickScript。



- 2 单击**转换**。此时出现**转换**对话框。



- 3 转换这些标记。
  - 单击**本地**，以便将占位符标记转换为本地标记。程序会提示您在“标记名字典”中定义每个标记。
  - 单击**远程**，以便将占位符标记转换为远程标记引用。此时出现**访问名**对话框。选择“访问名”，然后单击**关闭**。
- 4 转换标记之后，单击 QuickScript 编辑器中的**确定**。

## 导入的窗口与脚本中的标记占位符

导入窗口或 QuickScript 时，您可以配置如何处理关联的标记。

- 使用占位符标记。

缺省条件下，导入的标记会转换为“占位符”（或“索引”）标记。最多允许使用 4096 个占位符。

占位符标记包含一个三字符的前缀。例如，如果原始标记为 WaterHeater，则占位符标记为 ?d:WaterHeater。

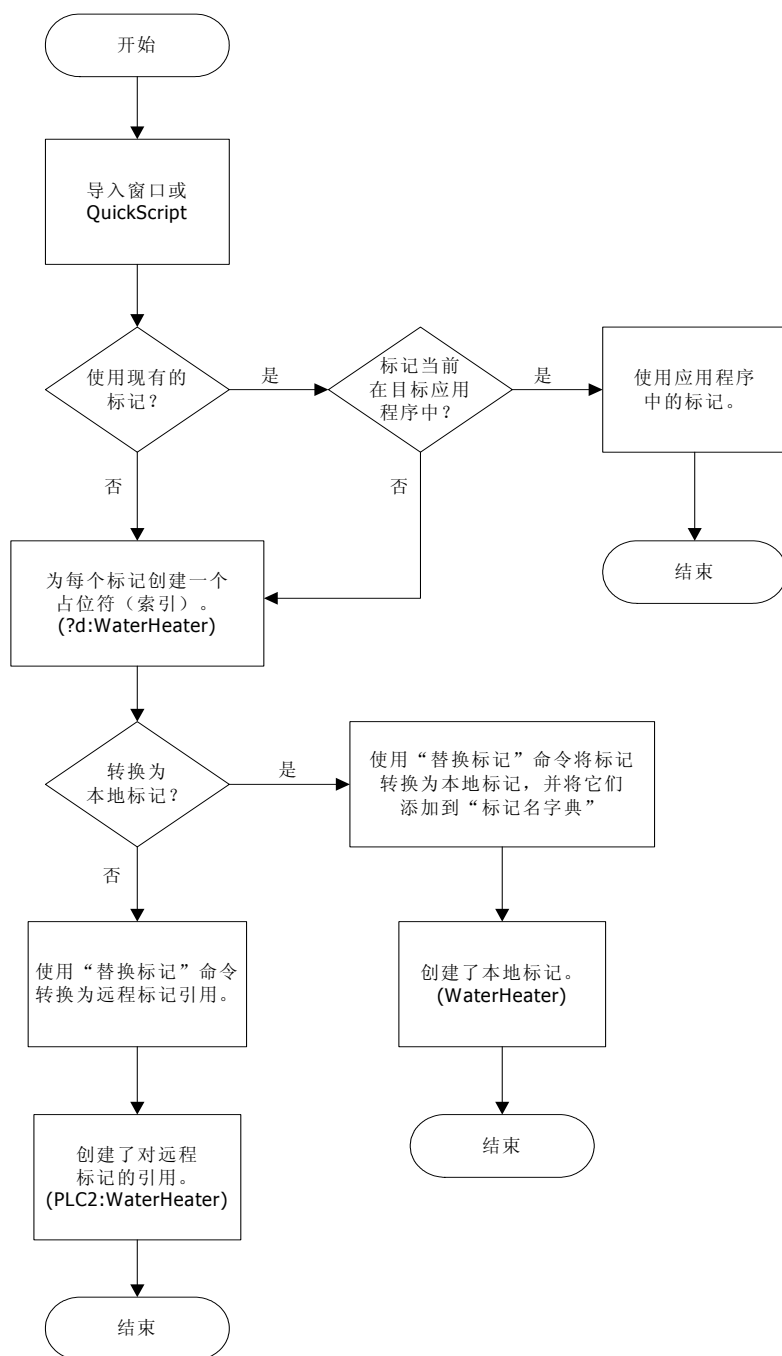
如果导入的标记包含 30、31 或 32 个字符，占位符前缀仍会添加到标记的开头，并且不截短现有标记的长度。例如，仅对占位符标记而言，32 个字符的标记增加到 35 个字符长。对于标准标记，不支持这种标记长度的增加。

要在应用程序中使用占位符标记，必须执行以下操作之一：

- 将此标记转换为常规（本地）标记并在“标记名字典”中定义它。
  - 将它转换为远程标记引用。远程标记的示例有 PLC2:WaterHeater。远程标记引用使得应用程序可以立刻接收来自远程标记服务器的数据，而不必在本地“标记名字典中”定义单个标记。
- 使用现有的标记。

在导入期间，如果选择使用现有的标记，则 InTouch HMI 确认“标记名字典”中是否已经存在导入的标记。如果标记已经存在，则该标记作为全限定标记导入。通过使用此选项，可以减少占位符的总数，以便您导入标记数据库较大的应用程序。

下图介绍如何处理导入的窗口与 QuickScript 中的标记。





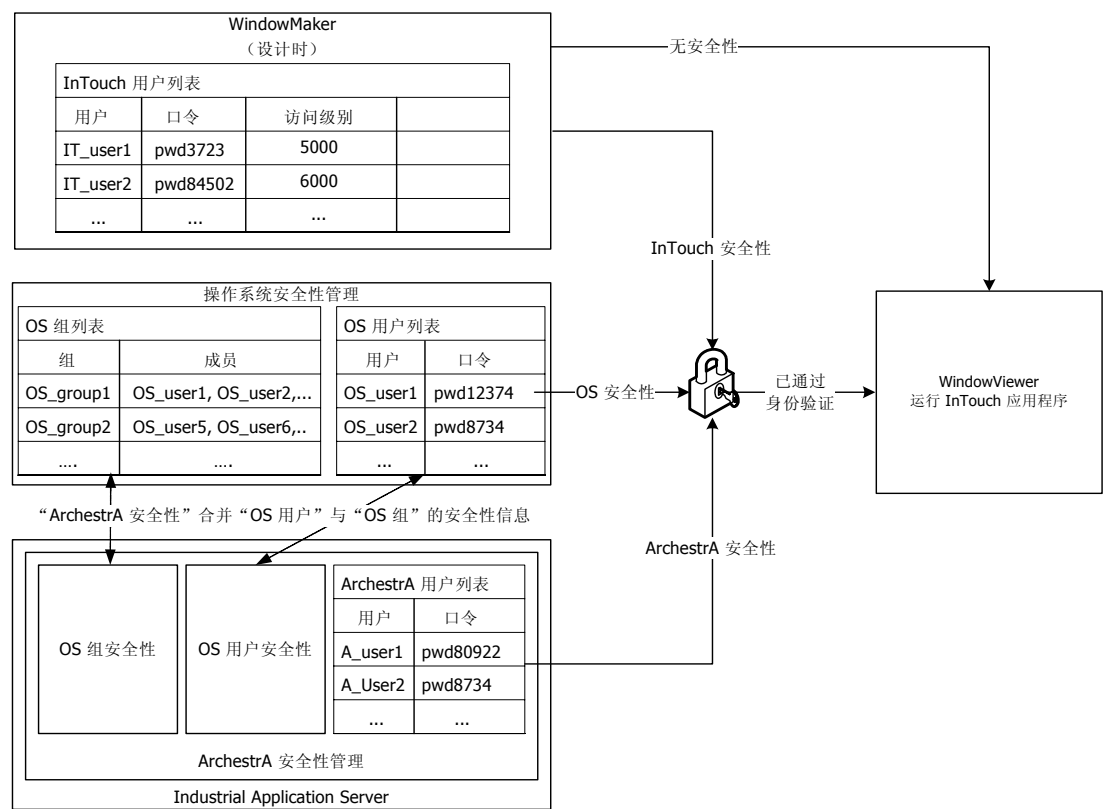
## 第 5 章

# 保护 InTouch 安全

您可以使用以下机制保护 InTouch 应用程序：

- 传统的基于 InTouch 的安全性
- 基于操作系统的安全性
- 基于 ArchedrA 的安全性

下图显示三种安全性之间的关系。



## InTouch 安全性功能

要在运行时保护 InTouch 应用程序，可以：

- 设置不活动超时时段
- 锁定键
- 隐藏菜单

### 配置不活动超时

您可以将 WindowViewer 配置成从 InTouch 应用程序自动注销不活动的操作员。因为不活动而注销操作员之后，操作员必须再次登录。通过设置自动不活动注销时段，可以在操作员离开工作站而导致其无人照管时，防止他人未经授权访问您的 InTouch 应用程序。

有一个计时器测量操作员未与正在运行的 InTouch 应用程序进行交互的时段。每次操作员使用鼠标或任何其它输入设备输入数据时，该计时器会重置。

**备注** 不活动计时器不会因 Active-X 控件、“OLE 自动化”控件及 SPC 向导而重置。

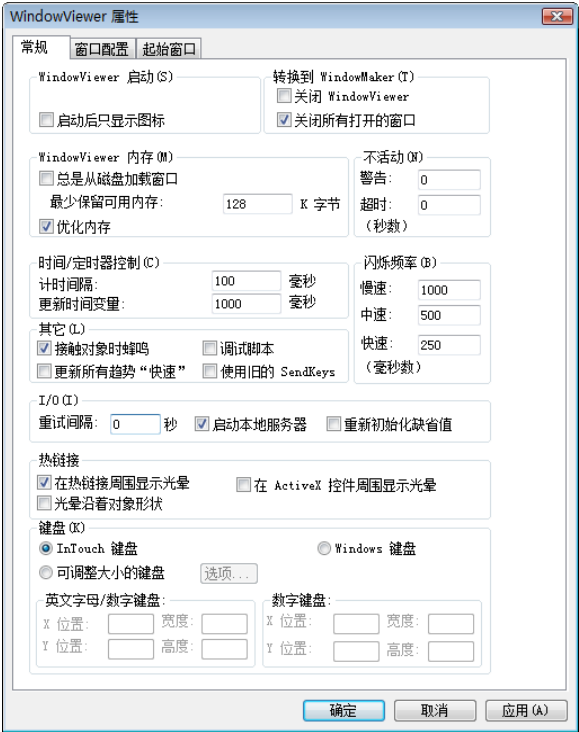
自动注销操作员的过程分为三个步骤：

- 1 操作员的不活动时段超过指定的警告时段时，WindowViewer 将 \$InactivityWarning 系统标记设置为 1。您可以在条件 QuickScript 中使用 \$InactivityWarning 标记显示一个窗口，以警告操作员因不活动而将要注销的情况。操作员通过在指定的超时时段发生之前作出响应以保持登录。操作员采取某种操作时，\$InactivityWarning 标记与不活动计时器重置为零。
- 2 如果操作员在不活动警告之后未能作出响应，到达超时时段时，\$InactivityTimeout 系统标记设置为 1。  
\$InactivityTimeout 为 1 时，WindowViewer 将登录的操作员姓名设置为保留名 None，并将 \$AccessLevel 安全性标记设置为 0。
- 3 如果 \$InactivityTimeout 标记设置为 1，您可以编写一个脚本以自动注销操作员。

您可以在警告功能之外独立使用超时功能。

要配置不活动超时

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



- 2 在**不活动**区域中，配置警告与超时值。执行以下操作：

- 在**警告**框中，输入在 `$InactivityWarning` 标记设置为 1 之前可以经过的秒数。
- 在**超时**框中，输入在 `$InactivityTimeout` 标记设置为 1 之前可以经过的秒数。

3 单击**确定**。

4 要在不活动警告时间过去之后显示名为 "Warning - Logoff Pending" 的窗口，请创建以 "`$InactivityWarning`" 为条件并使用以下脚本主体的条件脚本：

```
Show "Logoff Pending";
```

5 要在不活动超时时间过去之后注销用户并显示名为 "Logged Off" 的窗口，请创建以 "`$InactivityTimeout`" 为条件并使用以下脚本主体的条件脚本：

```
LogOff();  
Show "Logged Off";
```

### `$InactivityTimeout` 系统标记

指出为不活动配置的时间已经过去。

**类别**

安全性

**用法**

`$InactivityTimeout`

**附注**

不活动计时器的时间过去时设置为 1。如需有关设置注销时间的详细信息，请参阅第 110 页的“配置不活动超时”。

---

**备注** 不活动计时器不会因 Active-X 控件、OLE 自动化控件及 SPC 向导而重置。

---

**数据类型**

离散（只读）

**另请参阅**

`$InactivityWarning`

**示例**

下例是一个“为真时”条件脚本：

```
If $InactivityTimeout == 1 THEN  
    Show "Logged Off";  
ENDIF
```

**另请参阅**

`$InactivityWarning`

### `$InactivityWarning` 系统标记

指出为警告用户注销即将发生所配置的时间已过去。

**类别**

安全性



**用法**  
\$InactivityWarning

**附注**  
不活动警告的时间过去时设置为 1。不活动计时器仅通过鼠标单击或键盘活动进行重置。如需有关设置注销警告的详细信息，请参阅第 110 页的“配置不活动超时”。

---

**备注** 不活动计时器不会因 Active-X 控件、OLE 自动化控件及 SPC 向导而重置。

---

**数据类型**  
离散（只读）

**示例**  
下例是一个“为真时”条件脚本。

```
If $InactivityWarning == 1 THEN
    Show "Logoff Pending";
ENDIF;
```

**另请参阅**  
\$InactivityTimeOut

# 锁定系统键

通过运行 InTouch 应用程序的计算机上禁用系统键，可以限制操作员访问 Windows 标准函数。例如，您可以阻止操作员使用 Windows CTRL+ALT+DEL 组合键来显示**任务管理器**对话框。禁用系统键可以阻止操作员从 InTouch HMI 切换到其它 Windows 应用程序。

WindowViewer 有一些键过滤器选项，可以设置系统键在 InTouch 应用程序启动时的缺省状态。键过滤器在活动时禁用系统键。

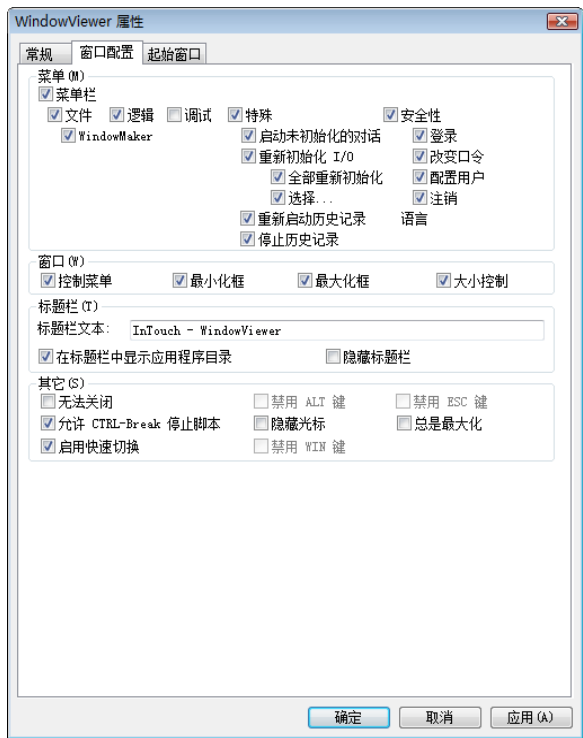
系统键的禁用取决于您希望各种 InTouch 用户去完成哪些任务。大多数功能键应该对操作员禁用。管理员仍然需要这些功能键来执行自己的 InTouch 任务。

您可以编写一个脚本，根据登录到 WindowViewer 的用户的访问级别来启用或禁用系统键。您可以在脚本中使用 EnableDisableKeys() 函数，以便有选择性地启用或禁用一些 Windows 功能键。

## 要启用键过滤器

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

2 单击窗口配置选项卡。



3 在其它区域中，禁用 WindowViewer 系统键。执行以下操作：

- 清除启用快速切换复选框，以便从 WindowViewer 中删除可以将用户切换到 WindowMaker 的开发按钮。
- 选择禁用 ALT 键复选框，以便在运行 InTouch 应用程序的计算机上禁用 ALT 键。
- 选择禁用 WIN 键复选框，以便在运行 InTouch 应用程序的计算机上禁用 WIN 键。
- 选择禁用 ESC 键复选框，以便在运行 InTouch 应用程序的计算机上禁用 ESC 键。

4 单击确定。

5 编写一个在 WindowViewer 开始运行 InTouch 应用程序时运行的脚本。

此脚本应该包含一些语句，根据登录到 WindowViewer 的用户访问级别来动态执行键的锁定或释放。

在脚本中包含 EnableDisableKeys() 函数以启用 / 禁用 ALT、ESC 及 WIN 键。EnableDisableKeys() 函数会根据离散参数值来启用或禁用系统键：

```
EnableDisableKeys(AltKey, EscKey, WinKey);
```

参数值为 1 时，将启用键过滤器来禁用该键。

## 启用 DisableKeys() 函数

启用 / 禁用 Alt、Escape 及 Windows 键的键过滤器。

类别

View

语法

EnableDisableKeys(*AltKey*, *EscKey*, *WinKey*);

参数

*AltKey*

启用或禁用 Alt 键的键过滤器的整数：

1 = 启用过滤器（禁用 Alt 键）

0 = 禁用过滤器（启用 Alt 键）

*EscKey*

启用或禁用 Escape 键的键过滤器的整数：

1 = 启用过滤器（禁用 Esc 键）

0 = 禁用过滤器（启用 Esc 键）

*WinKey*

启用或禁用 Windows 键的键过滤器的整数：

1 = 启用过滤器（禁用 Win 键）

0 = 禁用过滤器（启用 Win 键）

附注

禁用 Alt 键时也会禁用 Win+L 组合键（锁定 Windows 桌面）。Win+L 是涉及 Alt 键的另一个组合键的快捷键。因此，禁用 Alt 键时也会禁用这个特定的快捷键。

示例

```
EnableDisableKeys(0,0,0); // enable all three keys
```

```
EnableDisableKeys(1,1,1); // disable all three keys
```

```
EnableDisableKeys(0,0,1); // enable Alt and Escape keys, disable  
Windows key.
```

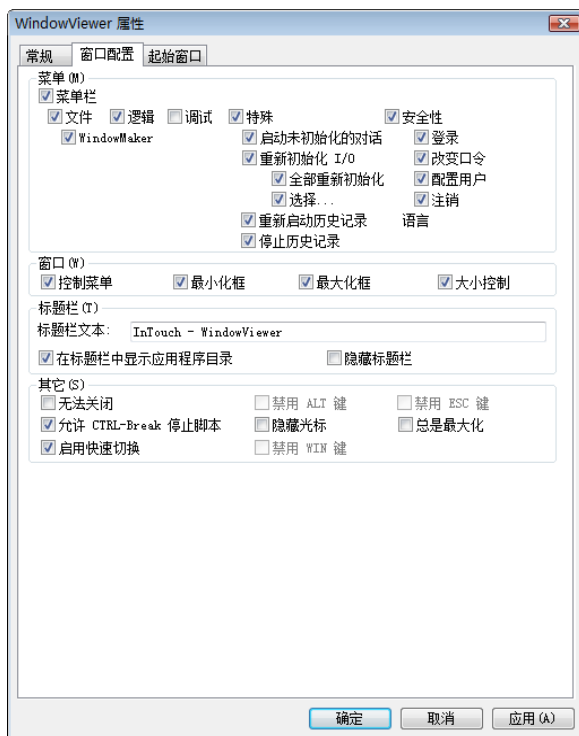
## 在运行时隐藏菜单项

您可以在运行 InTouch 应用程序时，通过隐藏 WindowViewer 菜单与命令来限制操作员访问它们。

要在运行时隐藏菜单项

- 1 在特别菜单上，指向配置，然后单击 WindowViewer。此时出现 WindowViewer 属性对话框。

## 2 单击窗口配置选项卡。



## 3 在菜单区域中，选择希望对操作员隐藏的 WindowViewer 菜单与命令。执行以下操作：

- 清除 **WindowMaker** 复选框使 **WindowMaker** 命令在 **WindowViewer 文件** 菜单中不可用。清除此选项不影响快速切换到 **WindowMaker**。
- 清除**逻辑**复选框，以隐藏包含启动与停止 QuickScript 的命令的 **WindowViewer 逻辑**菜单。

**备注** 您可以使用 **\$LogicRunning** 系统标记使操作员能够启动与停止所有的 **QuickScript**。

如果选择**允许 CTRL-Break 停止脚本**选项，则无论是否显示**逻辑**菜单，操作员都可以停止运行所有的 **QuickScript**。

当前正在执行的异步 **QuickFunction** 无法停止。不过，您可以阻止操作员启动新的异步 **QuickFunction**。

- 如果正在测试应用程序，请选择**调试**复选框。否则清除**调试**复选框，以便在运行时隐藏**调试**菜单。
- 清除**特殊**菜单项以阻止操作员停止正在进行的 InTouch 功能，如记录与 I/O 连接。
- 清除**安全性**复选框，以阻止操作员更改与安全性相关的选项。

- 4 在**窗口**区域中，选择希望在 WindowViewer 中提供给操作员窗口控件。这些选项会影响运行 InTouch 应用程序的窗口。执行以下操作：
  - 清除**控制菜单**复选框，以隐藏可以关闭、最小化、最大化窗口及调整窗口大小的这些控件。
  - 清除**最小化框**复选框，以阻止操作员最小化窗口。
  - 清除**最大化框**复选框，以阻止操作员最大化窗口。
  - 清除**大小控制**复选框，以阻止操作员调整窗口大小。
- 5 在**标题栏**区域中，配置运行 InTouch 应用程序的窗口的标题栏。执行以下操作：
  - 在**标题栏文本框**中，输入要在 WindowViewer 标题栏中显示的标题。
  - 选择**显示应用程序目录**复选框，以便在标题栏中包含 InTouch 应用程序文件夹的路径。
  - 选择**隐藏标题栏**复选框，以隐藏窗口的标题栏。
- 6 在**其它**区域中，执行以下操作：
  - 选择**无法关闭**复选框，以阻止操作员关闭正在运行 InTouch 应用程序的 WindowViewer 窗口。选择此选项时禁用窗口的**关闭按钮**。  
如果希望隐藏**关闭按钮**，请清除**窗口**区域中的**控制菜单**复选框。
  - 清除**允许 CTRL-Break 停止脚本**复选框，以禁用能够让操作员停止 QuickScript 的 CTRL + BREAK 组合键。

---

**备注** 当前正在执行的异步 QuickFunction 无法停止。不过，可以阻止执行新的异步 QuickFunction。

---

- 选择**隐藏光标**复选框，以便在运行时隐藏鼠标光标。如果在设计触摸屏应用程序，这会很有用。
  - 选择**总是最大化**复选框，让运行 InTouch 应用程序的窗口保持最大化显示在操作员屏幕上。
- 7 单击**确定**。
  - 8 重新启动 WindowViewer 以应用更改。

## 基于身份验证与授权的安全性

InTouch 安全性是一个分为两个步骤的过程，第一步确定尝试使用应用程序的人员是否为有效用户。第二步确定通过身份验证的用户拥有哪些 InTouch 权限。

### 比较身份验证与授权

身份验证是验证用户身份的过程。通常，操作员会在使用 InTouch 应用程序之前输入用户名与口令进行验证。作为身份验证过程的一部分，所有三种类型的安全性都会在登录过程中验证用户的凭证。

授权是确定通过身份验证的用户是否可以访问所请求的资源的过程。通常，对 InTouch 功能的访问权限是基于用户在组中的成员关系或是基于所指定的访问级别来授予的。

### 不同的身份验证安全模式

所有类型的 InTouch 安全性都在登录过程中使用用户名与口令来验证用户。每种类型的安全性都在身份验证过程中提供一种不同的机制来验证用户名与口令。

#### 使用基于 InTouch 的安全性

给应用程序应用安全性属于可选项。缺省条件下，InTouch 应用程序不采用安全措施。不过，通过将功能链接到内部标记，可以对允许操作员执行哪些功能作出限制。此外，在应用程序上设立安全机制时，可以创建审核跟踪线索，将报警与事件同登录到 InTouch HMI 的操作员关联起来。

安全性基于操作员通过输入用户名与口令登录到 InTouch 应用程序来进行验证。您必须给每个操作员指定用户名、口令及访问级别。

缺省条件下，创建新的应用程序时，用户名设置为 Administrator，访问级别为 9999，允许访问所有的安全性命令。缺省的管理员口令是 wonderware。口令的最大字符数量是 29。

将新用户名添加到安全性列表并重新启动 WindowMaker 或 WindowViewer 之后，缺省用户名自动重置为 None，访问级别为 0，这会阻止访问 WindowMaker 与 WindowViewer 中的配置用户命令。不过，Administrator 帐户与口令保持不变，并且仍可以使用。

操作员登录到应用程序之后，访问任何受保护的功能时，系统都会根据给链接到该功能的内部安全性标记所指定的值去验证操作员的口令和访问级别，以确定是否授予访问权限。

## 使用基于操作系统的安全性

基于操作系统的身份验证方法从 Windows 操作系统中继承了某些帐户策略的实行方式，而其它一些策略则是在 InTouch HMI 中实行。口令策略（如最大与最小口令寿命，以及最小口令长度）则是由操作系统实行的。

在安装期间使用的用户名会作为操作系统的一部分发挥作用。

Windows 域必须设置所需的帐户策略，以实行这些标准。

InTouch HMI 实行不活动超时时段策略。

在基于操作系统身份验证的方法中，可以从与 Windows “网络域”或“工作组”关联的用户列表中选择用户名。每个用户名都指定有一个访问级别，确定该用户对给定活动所拥有的权限。操作系统在内部管理口令，因此 InTouch HMI 不在存放应用程序的节点上存储口令。

基于操作系统的安全性使用 InTouch AddPermission() 脚本函数来定义与维护一份用户及其相应访问级别的列表。此列表在执行 Addpermission() 调用之后创建，并写入磁盘。包含身份验证详细信息的文件不会复制到 NAD 客户端节点上。

通过执行 WindowViewer 特别菜单中安全性下的登录菜单命令（如果显示特别菜单），操作员可以登录到应用程序；或者，您也可以使用链接到内部安全性标记的触控输入对象来创建自定义的登录窗口。

用于在应用程序上设立安全机制的命令位于 WindowMaker 与 WindowViewer 中特别菜单的安全性下。这些安全性命令用于登录与注销应用程序、更改口令，以及配置有效用户名、口令、访问级别的列表。

例如，通过指定登录的操作员的访问级别必须大于 2000，可以控制对某个窗口的访问、某个对象的可见性，等等。

## 使用基于 ArchestrA 的安全性

将某个节点配置成使用 ArchestrA 安全性时，InTouch HMI 使用 Application Server 中的方法与对话框来执行登录与注销操作。用户在 Application Server 的 Galaxy “储备库”节点上配置。如需有关详细信息，请参阅 Application Server 文档。

ArchestrA 安全性使您可以轻松地定义用户并指定允许执行的操作。这些安全权限是使用自动化对象基于用户可以执行的操作来定义的。基本方法包含以下步骤：

- 1 定义安全性模型。
- 2 根据安全模型组织整理要保护的自动化对象。
- 3 根据安全性模型定义用户。

系统管理员通过创建相应的用户配置文件来定义系统用户。然后，通过从安全性模型中预定义的用户角色列表中进行选择，系统管理员可以将一个或多个角色指定给每个用户。

如果在基于 ArchestrA 的安全性模式下使用 InTouch，口令的最大字符数是 31。

InTouchView 用户通过基于口令的登录帐户进行正常的身份验证。

## 管理用户并设置授权级别

要给需要使用 InTouch HMI 的一组用户设施安全机制，必须：

- 给每个用户指定用户名与口令身份验证凭证。
- 给每个用户指定 InTouch 授权级别（访问级别）。

## 配置 InTouch 安全性身份验证与授权

对于每个操作员，您需要指定用户名、口令以及访问级别。

**None** 与 **Administrator** 名称是保留名，只有 **Administrator** 的口令可以更改（缺省值是 wonderware）。给应用程序配置用户名之后，更改 **Administrator** 口令。**Administrator** 的缺省访问级别 (9999) 是最高的，允许访问所有的 InTouch 功能，包括**配置用户**命令。

您也可以将一个“用户输入 - 离散”按钮链接到 **\$ConfigureUsers** 标记，以允许经过授权且访问级别大于或等于 9000 的用户访问**配置用户**对话框，以便编辑安全性用户名列表。操作员单击该按钮时，**\$ConfigureUsers** 标记的值设置为 1，此时出现**配置用户**对话框。操作员关闭对话框时，系统将该值重置为 0。这是一个仅用于写入操作的系统离散标记。

---

**备注** **\$ConfigureUsers** 标记仅在安全性类型设置为 InTouch 时起作用。对于基于 ArchestrA 与基于操作系统的安全性，它不起作用。

---

**要给应用程序操作员配置安全性**

- 1 在 **WindowMaker** 特别菜单上，指向**安全性**，然后单击**登录**。
- 2 使用 InTouch 管理员帐户进行登录。



- 3 在**特别**菜单上，指向**安全性**，然后单击**配置用户**。此时出现**配置用户**对话框。



- 4 要添加安全性帐户，请执行以下操作：
- a 在**用户名**框中，输入要指定给操作员的名称。
  - b 在**口令**框中，输入操作员口令，最多可以包含 29 个字符。
  - c 在**访问级别**框中，输入操作员的访问级别（最低 = 0 到最高 = 9999）。
  - d 单击**添加**将用户名添加到 InTouch 安全性列表。
- 5 要更改用户名，请选择名称，执行任何更改，然后单击**更新**。
- 6 要删除用户名，请选择名称，然后单击**删除**。
- 7 单击**确定**。

## 在运行时更改 InTouch 操作员口令

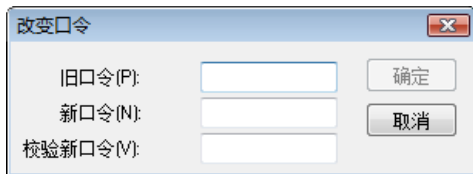
操作员可以在运行时使用 WindowViewer 中的**特别**菜单来更改他们的口令。

如果不打算在 WindowViewer 中显示**特别**菜单，则可以创建一个离散按钮，并将它链接到 \$ChangePassword 内部标记。

\$ChangePassword 标记的值设置为 1 时，出现**改变口令**对话框。操作员然后可以更改自己的口令。操作员关闭对话框时，系统将 \$ChangePassword 的值重置为 0。这是一个仅用于写入操作的系统离散标记。

### 要更改操作员口令

- 1 在**特别**菜单上，指向**安全性**，然后单击**改变口令**。此时出现**改变口令**对话框。



- 2 配置口令。执行以下操作：

- 在旧口令框中，输入旧口令。
- 在新口令框中，输入新口令。
- 在校验新口令框中，再次输入新口令。

3 单击**确定**。

## 设置基于操作系统的身份验证与授权

基于操作系统的安全性根据经过授权的 Windows 用户组的列表来验证 InTouch 用户。您可以在本地计算机或 Active Directory 服务器上创建 Windows 用户组。您必须通过将 Windows 用户添加到特定的组来将它们与这些组关联起来。如需有关创建用户组的详细信息，请参阅 Windows 操作系统文档。

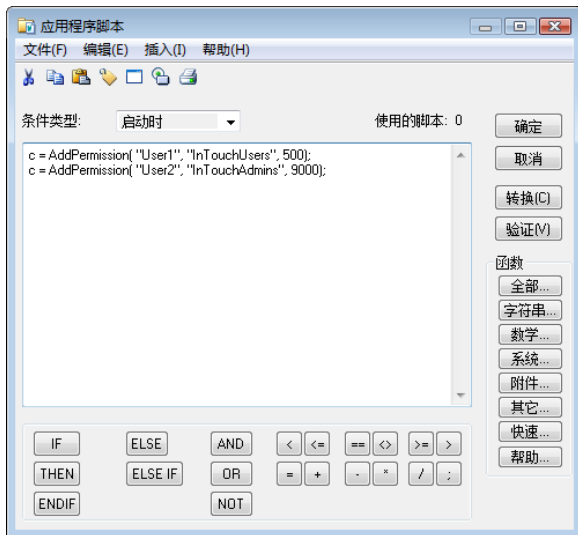
然后通过脚本中使用 AddPermission() 函数，您可以将 InTouch 访问级别指定给 Windows 组。AddPermission() 函数通常在应用程序启动时调用，这样在用户准备登录时，WindowViewer 便可以识别所有经授权的用户组。

通常在创建 InTouch 应用程序之后，立即指定基于操作系统的安全性。

将 InTouch 应用程序配置成使用操作系统身份验证与 InTouch 内部授权之后，特别 ... 安全性菜单上的**改变口令**、**登录**、**配置用户**及**注销**等命令便不再可用。

要设置基于操作系统的安全性并配置访问级别

- 1 在 WindowMaker 特别菜单上，指向**安全性**，指向**选择安全性类型**，然后单击**操作系统**。
- 2 在特别菜单上，指向**脚本**，然后单击**应用程序脚本**。此时出现**应用程序脚本**对话框。



- 3 在**条件类型**列表中，单击**启动时**。

- 4 使用 `AddPermission()` 函数指定组名与相应的访问级别。  
`AddPermission()` 的参数是操作系统（或域）、组名以及访问级别。
- 5 单击**确定**。

## 设置基于 ArchestrA 的安全性

ArchestrA 安全性系统是一个全局功能，它应用于 Galaxy 数据库中的每个对象。它是用户同 Galaxy 中的对象与函数之间的一个基于关系的系统。此系统是基于安全性角色（配置、系统管理以及运行时权限）与安全性组，以确定特定安全性角色的对象级运行时权限。安全性系统的配置在“集成开发环境”（Integrated Development Environment，简称 IDE）中完成，并通过其自身的编辑器应用于每个对象。

将 InTouch 应用程序配置成使用 ArchestrA 身份验证之后，WindowMaker 中**特别 ... 安全性**菜单上的**改变口令、登录、配置用户**以及**注销**命令便不再可用。

### 要设置基于 ArchestrA 的安全性

- 1 在 WindowMaker 中打开窗口。
- 2 在**特别**菜单上，指向**安全性**，指向**选择安全性类型**，然后单击**ArchestrA**。

## AddPermission() 函数

将特定的 InTouch 访问级别指定给本地系统或域上给定的用户组。在调用 `AddPermission()` 函数之后，属于该组的用户登录到 InTouch HMI 时，该用户获得指定的访问级别。

### 类别

安全性

### 语法

```
DiscreteTag=AddPermission( "Domain", "Group", AccessLevel);
```

### 参数

#### *Domain*

组所在的域或本地计算机的名称。

#### *Group*

Windows 用户组。

#### *AccessLevel*

您希望与给定的组关联的 InTouch 访问级别。

**附注**

仅对于操作系统安全性有效。调用此函数时，将在指定的域或工作组中检查是否存在指定的组。如果成功，则返回 **TRUE**，指定的“访问级别”与组关联，以便在随后的用户登录操作中使用。在所有其它情况中（即，如果为任何一个参数指定了无效的值），则返回 **FALSE**。

此函数通常配置成在应用程序启动时运行。它不影响当前登录的用户。只有在成功调用 **AddPermission()** 函数之后登录的用户，才会获得与自己的组关联的访问级别。

**示例**

```
DiscreteTag=AddPermission("corporate_hq", "InTouchAdmins", 9000);
DiscreteTag=AddPermission("johns01", "InTouchUsers", 5000);
```

**另请参阅**

**PostLogonDialog()**, **InvisibleVerifyCredentials()**,  
**IsAssignedRole()**, **AttemptInvisibleLogon()**,  
**QueryGroupMembership()**

## ChangePassword() 函数

显示**改变口令**对话框，允许登录的操作员更改自己的口令。

**类别**

安全性

**语法**

```
[Result=]ChangePassword();
```

**返回值**

返回以下整数值之一：

0 = 已按过“取消”。

1 = 已按过“确定”。

**附注**

如果操作员使用触摸屏，则操作员可以使用字母数字键盘输入新口令。

**示例**

以下脚本可以放到按钮上，或是从条件脚本或数据改变脚本中调用。

```
Errmsg=ChangePassword();
```

## \$AccessLevel 系统标记

定义当前登录的用户的访问级别。

**类别**

安全性

**用法**

**\$AccessLevel**

**附注**

此标记的值由 InTouch HMI 中指定给当前登录的用户的安全性配置文件的访问级别确定。在 WindowViewer 中，此配置文件可以使用**配置用户**菜单来访问。

\$AccessLevel 的实际数值对于 WindowViewer 没有任何意义，除非是 9000 或更大的值，这代表管理员权限，并且会在 WindowViewer 中启用**安全性**菜单。\$AccessLevel 系统标记可用于在系统中进一步自定义安全性。

**数据类型**

整型（只读）

**有效值**

0 到 9999

**示例**

以下语句用于在一个可视化链接中，根据登录的用户的访问级别来确定是否让某个对象（如按钮）变为可见：

```
$AccessLevel >= 2000;
  {Objects can have a "disable" link associated with them, with the
   expression based on $AccessLevel.}
$AccessLevel < 5411;
```

```
IF $AccessLevel <=500 THEN
```

```
  Show "Access Denied"; {popup window denying access}
```

```
ELSE
```

```
  Show "Access Granted"; {popup window granting access}
```

```
ENDIF;
```

**另请参阅**

\$Operator, \$OperatorEntered, \$PasswordEntered;  
\$ConfigureUsers

## \$ChangePassword 系统标记

显示**改变口令**对话框。

**类别**

安全性

**用法**

\$ChangePassword

**附注**

将此值设置为 1 以显示**改变口令**对话框。对话框关闭时，\$ChangePassword 系统标记的值重置为 0。如果不将此系统标记的值设置为 1，则结果未定义。

**数据类型**

离散（只写）。

**有效值**

1

**示例**

您可以创建一个打开**改变口令**对话框的离散按钮。给该按钮指定一个选择了“置位”选项的离散按钮链接。按下该按钮时，\$ChangePassword 系统标记设置为 1，并打开**改变口令**对话框。

**另请参阅**

\$AccessLevel, \$OperatorEntered, \$PasswordEntered, \$Operator, \$ConfigureUsers

## \$ConfigureUsers 系统标记

显示**配置用户**对话框。

**类别**

安全性

**用法**

\$ConfigureUsers

**附注**

此函数仅对于 InTouch 安全性有效。

将值设置为 1 以打开**配置用户**对话框。

对话框关闭时，此系统标记的值重置为 0。如果不将此系统标记的值设置为 1，则结果未定义。

用户必须有 >9000 的 \$AccessLevel 才可以显示此对话框。

**数据类型**

离散（只写）。

**有效值**

1

**示例**

您可以创建一个打开**配置用户**对话框的离散按钮。给该按钮指定一个选择了“置位”选项的离散按钮链接。按下按钮时，\$ConfigureUsers 系统标记设置为 1，**配置用户**对话框打开。

**另请参阅**

\$Operator, \$OperatorEntered, \$ChangePassword, \$PasswordEntered, \$AccessLevel

## 登录与注销

根据用于保护应用程序的安全性的类型，从 InTouch 应用程序进行的登录与注销也存在着差异。

### 登录到采用 InTouch 安全机制的应用程序

如果登录信息不正确或是无效，则出现一条消息，指出登录尝试不成功。

如果登录成功，则 \$AccessLevel 系统标记设置为与用户（在 InTouch 安全性用户列表中）关联的预定义值。

---

**备注** 您也可以使用 PostLogonDialog() 函数显示登录对话框。如需有关详细信息，请参阅第 129 页的“PostLogonDialog() 函数”。

---

#### 要登录到应用程序

- 1 在**特别**菜单上，指向**安全性**，然后单击**登录**。此时出现登录对话框。
- 2 在**名称**框中，输入您的用户名。
- 3 在**口令**框中，输入您的口令。
- 4 单击**确定**。

### 登录到采用操作系统安全机制的应用程序

用户登录到 InTouch 应用程序时，出现一个要求提供以下信息的对话框：

- 用户名
- 口令
- 域或本地计算机名

域 / 用户名组合传递给操作系统以验证用户的凭证。此时，不管是否启用操作系统缓冲区，都会尝试一次登录。如果没有缓冲区（例如，由于网络中断），因而用户无法登录，但先前使用缓冲区时该用户曾通过身份验证，则会从本地的 InTouch 缓冲区获取该用户的全名与访问级别。

如果成功通过所有安全性检查，该用户会被视为已登录到 InTouch HMI，相关的数据结构（如 \$Operator）会进行更新。否则显示一条错误消息。

如果操作员从未成功登录过，并且域不可用，则登录尝试会失败。此时 InTouch HMI 将一个系统事件记录到错误日志中。

如果口令过期，则显示一条错误消息。在操作员单击**确定**之后，出现**更改过期的口令**对话框，这样操作员便可以更改口令，并可以使用新口令再次尝试登录。

## 登录到采用 ArchestrA 安全机制的应用程序

用户通常会通过输入有效的用户名与口令来登录和注销采用 ArchestrA 安全机制的 InTouch 应用程序。

如果 InTouch 应用程序已配置成 ArchestrA 安全性 "None", 则使用缺省用户的登录凭证, 并且不会提示操作员登录。以下过程假设系统已经配置成 ArchestrA 身份验证模式, 如 "Galaxy"、"OS User based" (基于操作系统用户)、"OS Group based" (基于操作系统用户组)。

### 要登录

- 1 启动采用 ArchestrA 安全机制的 InTouch 应用程序。此时出现登录对话框。
- 2 输入有效的用户名与口令。如果系统无法验证您的身份, 则会再次提示您登录。

在系统验证您的登录凭证之后, 会根据安全性模型中同您关联的角色 / 权限来授予对将来所有操作的访问权限。

## 从 InTouch 应用程序中注销

操作员在完成工作之后从 InTouch 应用程序中注销。您也可以将应用程序配置成在一定时间内未有任何动作时自动注销操作员。如需有关详细信息, 请参阅第 110 页的“配置不活动超时”。

### 要从应用程序中注销

- ◆ 在**特别菜单**上, 指向**安全性**, 然后单击**注销**。

## 创建自定义的登录窗口

如果 WindowViewer 中未显示**特别菜单**, 则可以创建自定义的登录窗口让操作员登录到应用程序。

### 要创建自定义的登录窗口

- ◆ 将 \$OperatorEntered、\$PasswordEntered、\$OperatorDomainEntered 系统标记链接到用户输入对象, 或在脚本中使用它们以设置用户名、口令及域名。这些标记是仅用于写入操作的内部消息型标记。

只有在安全模式基于操作系统时, 才要求使用

\$OperatorDomainEntered 标记。否则会忽略此标记。如果安全模式基于操作系统, 并且 \$OperatorDomainEntered 为空, 则视为指向本地计算机。



将某个值写入 \$PasswordEntered 系统标记时，会发生使用 \$OperatorEntered、\$PasswordEntered 及 \$OperatorDomainEntered 系统标记值进行的登录尝试。如果仅将值写入 \$OperatorEntered 或 \$OperatorDomainEntered 系统标记，则不会发生登录。

如果输入有效，\$AccessLevel 与 \$Operator 内部标记会设置为它们的预定义值（在安全性用户列表中配置）。

您也可以将一个“用户输入 - 离散”按钮链接到 \$ConfigureUsers 标记，以允许经过授权且访问级别大于或等于 9000 的用户访问配置用户对话框，以便编辑安全性用户名列表。操作员单击该按钮时，\$ConfigureUsers 标记的值设置为 1，此时出现配置用户对话框。操作员关闭对话框时，系统将该值重置为 0。（这是一个仅用于写入操作的系统离散标记）。

---

**备注** \$ConfigureUsers 标记仅在安全性类型设置为 InTouch 时起作用。对基于 Archestra 的安全性，它不起作用。

---

## PostLogonDialog() 函数

显示 InTouch 登录对话框并返回 TRUE。

**类别**  
安全性

**语法**  
DiscreteTag=PostLogonDialog();

**示例**  
DiscreteTag=PostLogonDialog();

**另请参阅**  
InvisibleVerifyCredentials(), AttemptInvisibleLogon(), IsAssignedRole(), QueryGroupMembership(), AddPermission()

## LogonCurrentUser() 函数

使用当前登录到 Windows 操作系统的用户帐户登录到 InTouch。

**类别**  
安全性

**语法**  
DiscreteTag = LogonCurrentUser();

**返回值**  
如果登录失败，则返回 -1，并且不更改指定给 \$Operator、\$OperatorName、\$OperatorDomain 及 \$AccessLevel 的值。

**附注**

仅给基于操作系统的安全性使用此函数。

**示例**

```
DiscreteTag = LogonCurrentUser();
```

**另请参阅**

PostLogonDialog(), InvisibleVerifyCredentials(),  
IsAssignedRole(), AttemptInvisibleLogon(),  
QueryGroupMembership(), AddPermission()

## Logoff() 函数

从 InTouch 应用程序中注销用户。

**类别**

安全性（只写）

**语法**

```
DiscreteTag = LogOff();
```

**附注**

注销当前登录的用户，并且将当前用户状态设置为缺省的 none 操作员。

**示例**

```
DiscreteTag = LogOff();
```

**另请参阅**

PostLogonDialog(), InvisibleVerifyCredentials(),  
IsAssignedRole(), AttemptInvisibleLogon(),  
QueryGroupMembership(), AddPermission()

## AttemptInvisibleLogon() 函数

AttemptInvisibleLogon() 函数可以用在脚本中，使用所提供的凭证将用户登录到 InTouch。此时不要求该用户输入口令或用户 ID。

**类别**

安全性

**语法**

```
DiscreteTag=AttemptInvisibleLogon( "UserId", "Password", "Domain"  
);
```

**参数**

*UserId*

有效的用户帐户名。

*Password*

用户的口令。

### Domain

用户所属的本地计算机、工作组或域的名称。只有在当前安全性类型基于操作系统时，此参数才适用。

### 返回值

如果身份验证成功，则返回 **TRUE**。否则返回 **FALSE**。

### 附注

尝试使用所提供的凭证登录到 InTouch HMI。

- 如果登录尝试成功，则返回 **TRUE**，并且相应地更新 **\$OperatorDomain**、**\$OperatorName**、**\$AccessLevel** 及 **\$Operator** 等系统标记。
- 如果登录尝试失败，则返回 **FALSE**，并且当前登录的用户（如果存在）将继续是当前用户。

*Domain* 参数仅对基于操作系统的安全性是有效的。如果在使用 **ArchestrA** 安全模式，并且如果 **ArchestrA** 安全性又在使用基于操作系统的安全性，则 *UserId* 参数应该包含带域名或计算机名的全限定用户名。

### 示例

安全性基于操作系统时：

```
DiscreteTag=AttemptInvisibleLogon("UserId", "Password", "Domain");
```

安全性基于 InTouch 或基于 **ArchestrA** 时：

```
DiscreteTag=AttemptInvisibleLogon("UserId", "Password", "");
```

### 另请参阅

**PostLogonDialog()**, **InvisibleVerifyCredentials()**,  
**IsAssignedRole()**, **QueryGroupMembership()**,  
**AddPermission()**

## \$OperatorEntered 系统标记

用于输入有效的用户名。

### 类别

安全性

### 用法

**\$OperatorEntered**

### 附注

您可以使用此标记名来创建自定义的登录窗口。您可以将触控输入对象与 / 或 **QuickScript** 链接到此标记，以设置登录帐户的用户名。

---

**备注** **\$OperatorEntered** 系统标记有效时，**\$AccessLevel** 与 **\$Operator** 系统标记设置为它们的预定义值。

---

### 数据类型

消息（只写）。

**另请参阅**

\$AccessLevel, \$Operator, \$PasswordEntered,  
\$ChangePassword, \$ConfigureUsers

## \$PasswordEntered 系统标记

用于输入有效的口令。

**类别**

安全性

**用法**

\$PasswordEntered

**附注**

\$PasswordEntered 系统标记总是读为空字符串。绑定到此系统标记的显示链接总是为空。因为此标记总是返回一个空字符串，所以数据改变脚本从不触发此标记。您可以使用此标记名来创建自定义的登录窗口。您可以将触控输入对象与 / 或脚本链接到此标记，以设置用户的口令。

---

**备注** \$PasswordEntered 有效时，\$AccessLevel 与 \$Operator 系统标记设置为它们的预定义值。

---

**数据类型**

消息（只写）。

**另请参阅**

\$AccessLevel, \$Operator, \$OperatorEntered,  
\$ChangePassword, \$ConfigureUsers

## \$OperatorDomainEntered 系统标记

域名由操作员输入。

**类别**

安全性

**附注**

只要 \$PasswordEntered 标记发生更改，便会尝试进行登录，而不显示任何对话框。登录尝试将 \$\*Entered 标记用于输入用户名，将 \$OperatorDomainEntered 的字符串值用作域名（只有在当前模式是基于操作系统的安全性时才使用）。如果模式并非基于操作系统，则忽略此标记。

**数据类型**

字符串

**示例**

```
$OperatorEntered == "john";  
$OperatorDomainEntered == "Corporate_HQ";  
$PasswordEntered == "password";
```

另请参阅  
\$Operator

## 基于操作员或访问级别启用与禁用功能

在给应用程序采用安全机制之后，可以将 \$AccessLevel 与 \$Operator 安全性标记用在按钮、动画链接表达式或 QuickScript 中，以控制是否允许登录的操作员执行特定的应用程序功能。

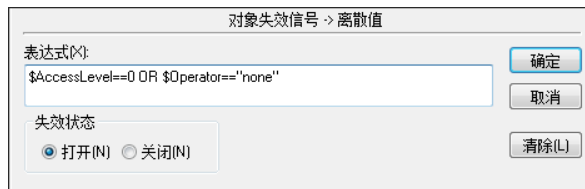
例如，要根据登录的用户的访问级别来显示对象，则可以在可见性动画链接表达式中使用以下语句：

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

或者也可以使用 IF 语句来限定脚本：

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

通过使用失效动画链接，也可以基于内部安全性标记的值来控制对象的触动功能。例如：



通过使用此表达式，如果无人登录，则该对象或按钮会被禁用，以防意外操作。

## InvisibleVerifyCredentials() 函数

InvisibleVerifyCredentials() 函数可以用在同步 QuickScript 中，以验证给定用户的凭证，而不必将用户登录到 InTouch HMI。

类别  
安全性

语法

```
AnalogTag=InvisibleVerifyCredentials( "UserId", "Password",  
    "Domain" );
```

参数

*UserId*

作为本地计算机、工作组或域的一部分的 Windows 操作系统用户帐户名。

*Password*

帐户口令。

*Domain*

帐户的 Windows 域。

**附注**

如果提供的用户、口令及域组合有效，则以整数的形式返回与用户关联的相应访问级别。否则返回 -1。

---

**备注** InvisibleVerifyCredentials() 函数必须从同步 QuickScript 中运行。如果从异步 QuickScript 中运行，此函数总是返回 -1。

---

此函数不更改当前登录的用户。Domain 参数仅对基于操作系统的安全性有效。如果使用 ArchestrA 安全模式，并且如果 ArchestrA 安全性又使用基于操作系统的安全性，则 UserId 参数应该包含带域名或计算机名的全限定用户名。

**示例**

```
AnalogTag=InvisibleVerifyCredentials( "john", "Password",
    "corporate_hq" );
```

**另请参阅**

PostLogonDialog(), AttemptInvisibleLogon(),  
IsAssignedRole(), QueryGroupMembership(),  
AddPermission()

## 检索当前登录的操作员的有关信息

对于任何安全性系统，跟踪都是一项主要功能。您可以使用一组安全性系统标记来确定登录到 InTouch 应用程序的用户、用户登录的域，以及尝试登录的时间。

### GetAccountStatus() 函数

返回用户口令过期之前的天数。

**类别**

安全性

**语法**

```
Result=GetAccountStatus(Domain, UserID);
```

**参数***Domain*

用户帐户所在的域或本地计算机的名称。

*UserID*

作为本地计算机、工作组或域的一部分的 Windows 用户帐户名。

返回值

此函数也返回以下值：

结果	描述
-1	帐户口令已过期
-2	帐户口令永不过期
-3	帐户已锁定
-4	帐户已禁用
-5	帐户信息无效

附注

此脚本函数用于基于操作系统的安全性。不要将此函数用于 ArchestrA 安全模式。

如果 GetAccountStatus() 函数用于 ArchestrA 安全性，则脚本会尝试直接从域控制器检索帐户信息。只要 ArchestrA Galaxy “储备库”使用相同的域的操作系统安全性，此函数便可以起作用。

示例

Status = GetAccountStatus("Corporate\_HQ","Operator");

IsAssignedRole() 函数

确定当前登录的用户是不是特定用户角色的成员。仅适用于 ArchestrA 安全性。

类别  
安全性

语法

DiscreteTag=IsAssignedRole( "RoleName" );

参数

RoleName  
与 Application Server 用户关联的角色。

附注

仅对 ArchestrA 安全模式有效，适用于当前登录的用户。如果用户当前已登录，并且拥有在 Galaxy IDE 中指定的 RoleName 角色，则返回 TRUE。否则返回 FALSE。

示例

DiscreteTag=IsAssignedRole( "Administrators" );

另请参阅

AttemptInvisibleLogon(), PostLogonDialog(), InvisibleVerifyCredentials(), QueryGroupMembership(), AddPermission()

## QueryGroupMembership() 函数

确定当前登录的用户是不是特定用户组的成员。仅适用于操作系统安全性。

**类别**

安全性

**语法**

```
DiscreteTag=QueryGroupMembership( "Domain", "Group" );
```

**参数**

*Domain*

组所在的域或本地计算机的名称

*Group*

组的名称。

**附注**

仅对操作系统安全模式有效，适用于当前登录的用户。如果用户当前已登录，并且是域上某个组的成员，则返回 **TRUE**。否则返回 **FALSE**。

**QueryGroupMembership()** 函数在基于操作系统的安全性中可以正常使用；对于基于 **ArchestrA** 的安全性，仅当 **ArchestrA** 安全性设置为基于操作系统的安全性时，它才能正常使用。

**示例**

```
DiscreteTag=QueryGroupMembership( "corporate_hq",
    "InTouchAdmins" );

DiscreteTag=QueryGroupMembership( "JohnS01", "InTouchUsers" );
```

**另请参阅**

**PostLogonDialog()**, **InvisibleVerifyCredentials()**, **IsAssignedRole()**, **AttemptInvisibleLogon()**, **AddPermission()**

## \$OperatorName 系统标记

如果使用基于操作系统或 **ArchestrA** 的身份验证并且某人登录后未注销，则包含操作员的全名。否则此标记包含登录的用户名称（与 **\$Operator** 标记的内容相同）。

**类别**

安全性

**数据类型**

字符串（只读）

**示例**

```
IF $OperatorName <> "" THEN
    {Configure some defaults}
ENDIF;
```

**另请参阅**

**\$Operator**



## \$OperatorDomain 系统标记

根据使用的安全性，包含不同的值：

- 如果选择基于操作系统的安全性，并且操作员已成功登录，则 \$OperatorDomain 标记包含登录时指定的域或节点名。
- 如果选择 ArchestrA 安全性，并且用户已登录，则 \$OperatorDomain 包含 "ArchestrA"。
- 如果选择 InTouch 安全性，则 \$OperatorDomain 标记包含字符串 "InTouch"。
- 如果选择 "None"，则它是一个空字符串 ("")。

### 类别

安全性

### 数据类型

字符串

### 示例

```
IF $OperatorDomain == "PRODUCTION" THEN
    {Allow change to setpoint}
ELSE
    {Change denied}
ENDIF;
```

### 另请参阅

\$Operator

## \$Operator 系统标记

包含登录的用户的登录名。

### 类别

安全性

### 数据类型

字符串

## \$VerifiedUserName 系统标记

如果成功调用 InvisibleVerifyCredentials() 函数，并且如果安全性模式设置为基于操作系统或基于 ArchestrA Application Server 的安全性，则包含通过验证的用户的全名。如果调用失败，则此系统标记设置为空。

### 类别

安全性

### 用法

\$VerifiedUserName

**附注**  
\$VerifiedUserName 系统标记更改时（调用 InvisibleVerifyCredentials() 函数时），生成一个事件。

**数据类型**  
消息（只读）。

**有效值**  
用户的全名。

**示例**  
Tag = InvisibleVerifyCredentials("john","password", "Plant\_Floor");{ If the call is successful, the \$VerifiedUserName is set to "John Smith" and an Operator Event is generated. If the above call is not successful, \$VerifiedUserName is set to "" }

**另请参阅**  
InvisibleVerifyCredentials()、\$OperatorName、\$Operator

## 安全系统标记与函数的摘要

下表显示在各种安全模式下您可以使用的安全系统标记与函数。

	InTouch 安全性	操作系统安全性	Archestra 安全性
\$AccessLevel	是	是	是
\$ChangePassword	是	是	是
\$ConfigureUsers	是	否	否
\$InactivityTimeout	是	是	是
\$InactivityWarning	是	是	是
\$Operator	是	是	是
\$OperatorDomain	否	是	是 *
\$OperatorDomainEntered	否	是	是 *
\$OperatorEntered	是	是	是
\$OperatorName	是	是	是
\$PasswordEntered	是	是	是
\$VerifiedUserName	否	是	是
AddPermission()	否	是	否
AttemptInvisibleLogon()	是	是	是
ChangePassword()	是	否	否
EnableDisableKeys()	是	是	是

	InTouch 安全性	操作系统安全性	ArchestrA 安全性
GetAccountStatus()	否	是	是 *
InvisibleVerifyCredentials()	否	是	是 *
IsAssignedRole()	否	否	是
Logoff()	是	是	是
LogonCurrentUser()	否	是	否
PostLogonDialog()	是	是	是
QueryGroupMembership()	否	是	是 *

\* ArchestrA 安全性基于操作系统用户或用户组时



## 第 6 章

# 在运行时切换语言

您可以开发能够在运行时切换为另一种语言的应用程序。

要启用运行时语言切换功能，必须完成以下任务：

- 为应用程序配置多种语言。
- 导出应用程序文本进行脱机翻译。
- 翻译一个或多个导出的字典文件。
- 导入一个或多个已翻译的字典文件。

作为运行时语言切换设置的一部分，您也可以将报警注释与报警域等内容本地化。除切换文本字符串的运行时语言之外，还可以配置 AlarmViewer 与 AlarmDBView 控件中报警注释、报警状态、报警类型以及报警类的运行时语言切换。

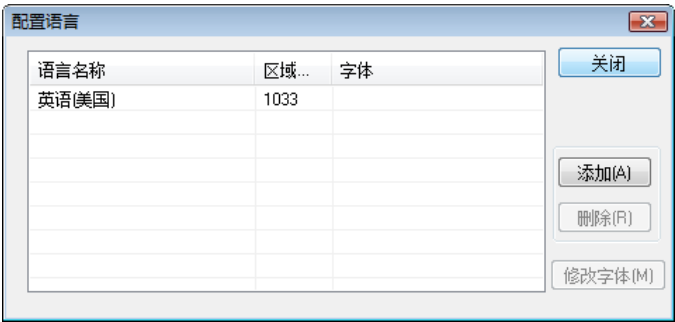
## 为运行时语言切换配置语言

每个 InTouch 应用程序都有一种用于开发此应用程序的基本语言。您必须配置希望支持的任何其它语言。

**备注** 如果结合使用语言切换与“网络应用程序开发”(NAD)，建议将 NAD 客户端节点的改变模式设置为“重新启动 WindowViewer”或“提示用户重新启动 WindowViewer”，而不是“将更改加载到 WindowViewer”或“提示用户将更改加载到 WindowViewer”。

### 要为运行时语言切换配置语言

- 1 在 WindowMaker 中，打开要配置语言的应用程序。
- 2 在特别菜单上，指向语言，然后单击配置语言。此时出现配置语言对话框。



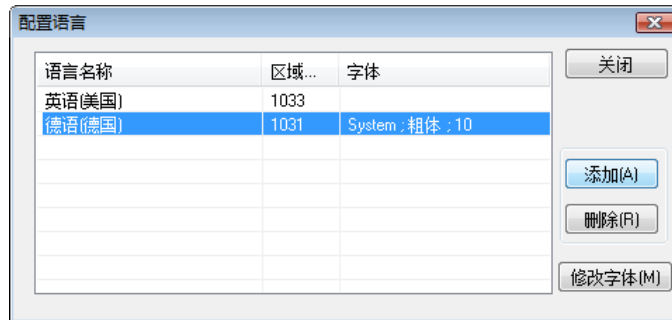
配置语言对话框显示应用程序的基本语言。

- 3 单击添加。此时出现添加语言对话框。



- 4 指定语言与字体设置。通过配置字体设置，可以为翻译的文本定义缺省字体属性。
  - 在按名称或语言环境 ID 列表中，单击要添加的语言。如果按名称选择语言，则对应的语言环境 ID 出现在语言环境 ID 列表中，反之亦然。
  - 单击字体。此时出现字体对话框。配置字体，然后单击确定。

- 单击**确定**以关闭**添加语言**对话框。此时配置的语言列在**配置语言**对话框中。



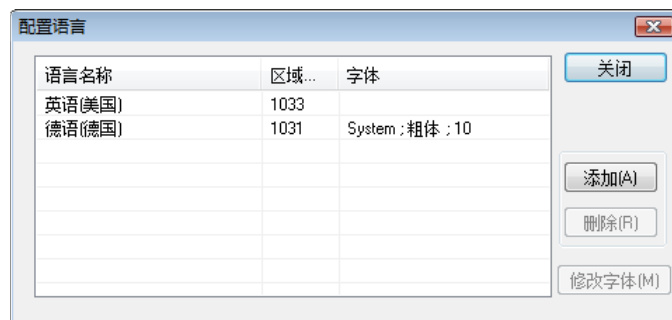
- 要添加更多的语言，重复步骤 3 到 5。
- 完成时，单击**关闭**。

## 为配置的语言更改字体设置

所有语言的缺省字体都是 **Tahoma**。字体和大小取决于基本语言中各个短语的相应设置。您可以为已配置的语言更改字体设置。由于不同语言的文本显示效果之间存在差异，您可以指定适当的字体，以确保翻译好的文本能在按钮及其它对象上正确显示。

### 要为配置的语言更改字体设置

- 在 **WindowMaker** 中，打开要为其配置的语言更改字体设置的应用程序。
- 在**特别菜单**上，指向**语言**，然后单击**配置语言**。此时出现**配置语言**对话框。



- 在语言列表中，选择目标语言，然后单击**修改字体**。此时出现 **Windows** 的标准**字体**对话框。
- 执行更改，然后单击**确定**。
- 单击**确定**以关闭**配置语言**对话框。

## 添加运行时语言切换功能

通过使用 **WindowViewer** **特别菜单**上的**语言**命令，运行时用户可以切换应用程序界面的语言。

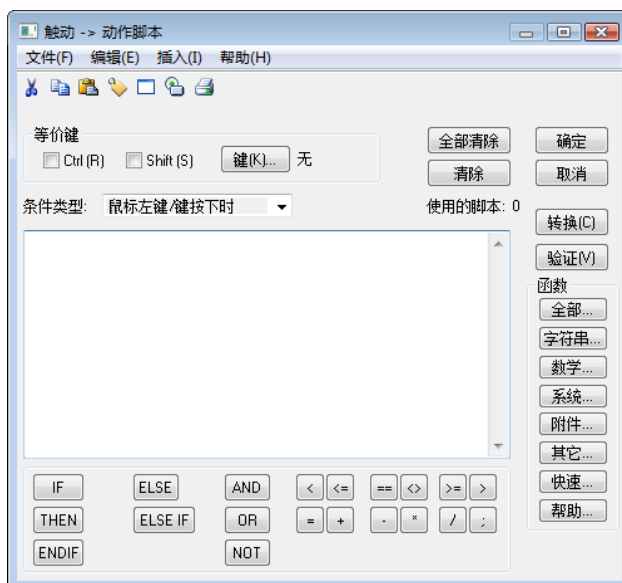
您也可以在应用程序中添加一个按钮，供运行时用户切换语言。在开始之前，确保已经为应用程序配置其它语言，并且您知道该语言的语言环境 ID。如需有关为应用程序配置语言的详细信息，请参阅第 142 页的“为运行时语言切换配置语言”。

#### 要添加用于在运行时切换语言的按钮

- 1 在 WindowMaker 中，打开要添加语言切换按钮的应用程序窗口。
- 2 在窗口中绘制一个按钮。
- 3 给按钮指定文本标签，指明选中时所要切换到的目标语言。
- 4 双击该按钮。此时出现动画选择对话框。

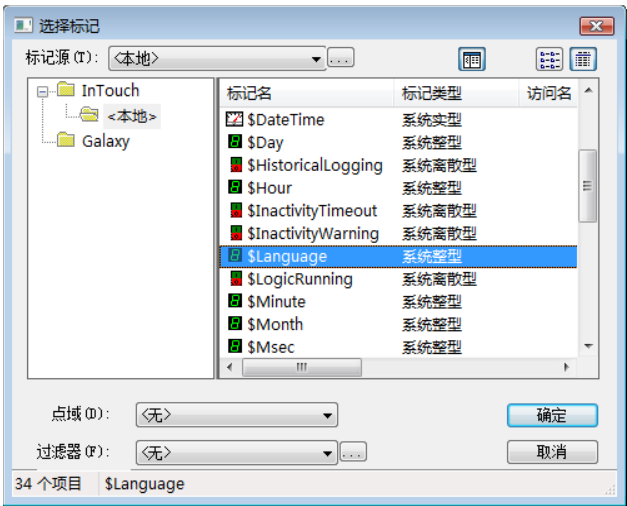


- 5 在**触动按钮**区域，单击**动作**。此时出现**触动 -> 动作脚本**对话框。

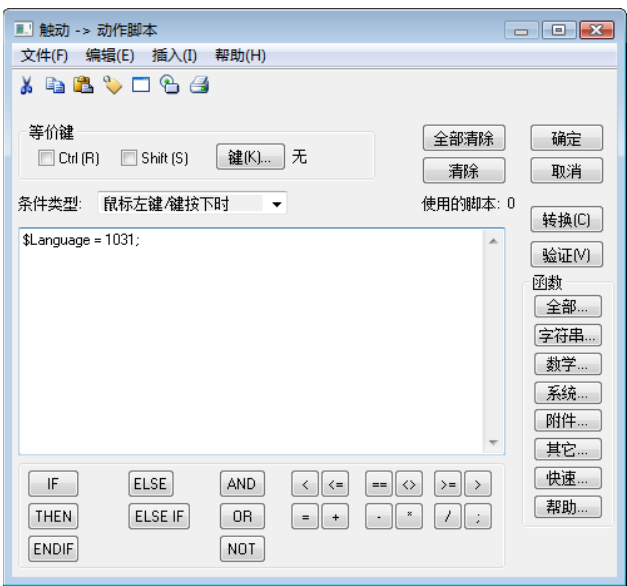




- 6 双击**触动 -> 动作脚本**对话框脚本区域中的任何位置。此时出现**选择标记**对话框。



- 7 单击 **\$Language** 系统标记，然后单击**确定**。
- 8 将 **\$Language** 系统标记设置成要指定给按钮的语言的语言环境 ID，然后单击**确定**。



**备注** 您也可以使用脚本函数 **SwitchDisplayLanguage(LocaleID)** 来取代 **\$Language** 标记。

- 9 单击**确定**以关闭对话框。

## SwitchDisplayLanguage() 函数

将可以看到的静态文本与报警域的显示语言切换成提供了翻译字符串的所需语言。

**类别**

其它

**语法**

```
SwitchDisplayLanguage(LocaleID);
```

**参数**

*LocaleID*

运行时显示静态文本字符串与报警域所用的语言。

**示例**

在本例中，在运行时显示德语。

```
SwitchDisplayLanguage(1031);
```

**另请参阅**

\$Language 系统标记

## \$Language 系统标记

如果为 InTouch 应用程序定义了多种语言，则 \$Language 系统标记反映当前显示的语言的“语言 ID”值。缺省条件下，这是基本 InTouch 系统 (E/F/G/J/SC) 的语言 ID（语言环境 ID）。将它设置为另一个 ID 时，会将字符串与报警域切换为新语言中定义的值。

**类别**

系统

**数据类型**

整型（读取 / 写入）

## 导出应用程序文本进行脱机翻译

如果 InTouch 应用程序包含许多字符串，则通常发送文本字符串进行批量翻译。您可以导出应用程序的字符串进行翻译，并使用文本编辑器、XML 编辑器或 Microsoft Excel 这样的电子表格程序来组织管理它们。

您可以从以下位置导出静态文本：

- 文本对象。
- 按钮文本。
- SmartSymbol 中的文本。
- 工具提示静态文本。

- 用户消息。
- 输入链接中的“打开 / 关闭消息”。
- 输出链接中的“打开 / 关闭消息”。
- 向导上的文本。

您只有在关闭 WindowMaker 中的所有窗口之后才能导出字典。如果导出字典文件之后对应用程序进行了更改，则必须再次导出字典文件。如需有关详细信息，请参阅第 148 页的“将文本导出到现有的字典文件”。

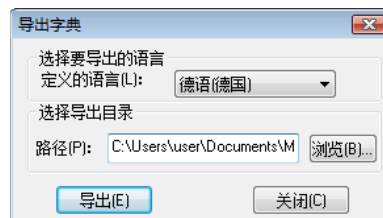
一次只能导出一种语言的文本字符串。缺省条件下，InTouch HMI 打开“我的 InTouch 应用程序”文件夹。如果选择任何其它文件夹，则 InTouch HMI 将该路径设置为缺省值。为每种语言创建一个用于导出短语的新文件夹，可以便于对字典文件的管理。例如，...\我的 InTouch 应用程序\My German Files\。

InTouch HMI 为应用程序创建字典文件，并且为应用程序中的每个 SmartSymbol 创建一个单独的字典文件。应用程序字典名的格式为 application name\_localeID，而 SmartSymbol 字典文件的格式则是 SSD\_Name of the Symbol\_localeID\_GUID。

导出应用程序的字典时，文件是 .xml 文件，您可以使用 Microsoft Excel 2003 或更高的版本来编辑它。

### 要导出应用程序文本进行脱机翻译

- 1 启动 WindowMaker，打开要导出文本字符串进行脱机翻译的应用程序。
- 2 在**特别**菜单上，指向**语言**，然后单击**导出字典**。此时出现**导出字典**对话框。

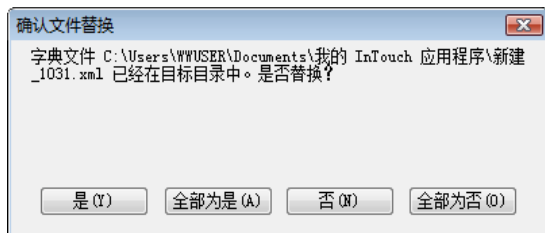


- 3 配置导出设置。
  - 在**定义的语言**列表中，单击要导出的语言字典。
  - 在**路径**框中，输入要导出字典的文件夹。单击**浏览**以选择现有的文件夹或创建一个新文件夹。
- 4 单击**导出**。此时显示导出进度。如果导出成功，则出现**导出成功**对话框。
- 5 单击**关闭**以返回 WindowMaker 窗口，或单击**关闭并启动“资源管理器”**以打开包含字典文件的文件夹。

## 将文本导出到现有的字典文件

导出应用程序文本进行脱机翻译之后，可能需要对应用程序进行更改。如果更改应用程序，您需要再次导出文本。如需有关详细信息，请参阅第 146 页的“导出应用程序文本进行脱机翻译”。

如果多次导出到相同的目录，则出现**确认文件替换**对话框。



如果单击**是**，则使用自上次导出以来所添加的任何新字符串和语言信息来更新现有的 .xml 文件。如果现有的字典文件包含任何短语的翻译，并且先前已导入 InTouch HMI，则那些翻译会保留下来。如果自上次导出以来从应用程序中删除了任何短语，则也会从字典文件中删除它们。

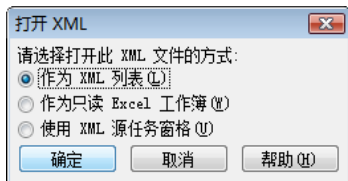
## 翻译导出的字典文件

导出包含应用程序文本的字典文件之后，使用 Microsoft Excel 2003 或更高版本来编辑文本。

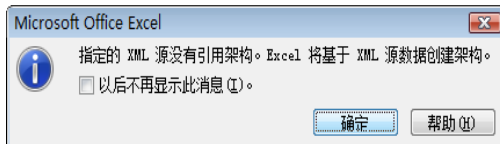
InTouch HMI 为导出的每种语言创建一个单独的字典文件。InTouch HMI 还为应用程序中的每个 SmartSymbol 创建一个单独的字典文件。务必确保翻译所有语言与 SmartSymbol 的所有字典文件。

### 要翻译导出的字典文件

- 1 在 Excel 中打开 XML 文件。此时出现**打开 XML**对话框。



- 2 单击**作为 XML 列表**，然后单击**确定**。此时可能出现一条消息。

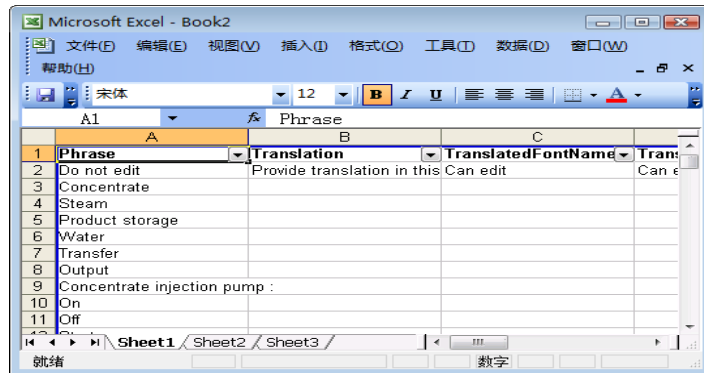


- 3 单击**确定**。

此时 XML 文件在 Excel 中打开，各个列分别是：

- 应用程序中的短语。

- 翻译人员翻译好的短语。
- 翻译的字体名。
- 翻译的字体属性。
- 翻译的字体大小。
- 基本字体属性。
- 基本字体大小。
- 上下文、短语 ID、语言 ID 以及外语 ID。



**重要** 只要修改 **Translation**、**TranslatedFontSize**、**TranslatedFontName** 以及 **TranslatedFontProperty** 列中的数据。请勿更改任何列标题。请勿插入或删除行。

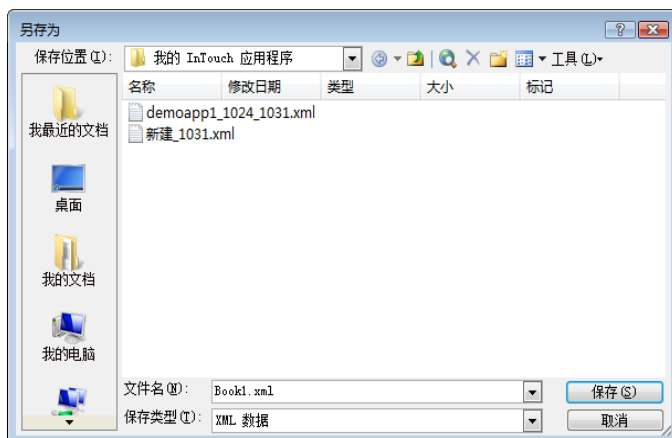
- 4 在与 **Phrase** 列中的基本语言字符串对应的行的 **Translation** 列中，输入特定于语言的文本。
- 5 如果需要，为翻译的字符串更改字体参数，使文本适合 **WindowViewer** 中提供的空间。

- 在 **TranslatedFontName** 列中，输入字体名。
- 在 **TranslatedFontProperty** 列中，输入字体属性的代号：  
B = 粗体  
I = 斜体  
U = 下划线

例如，如果希望文本为粗体，请在

**TranslatedFontProperty** 列中输入 **B**。如果希望文本为粗体并且带下划线，请在 **TranslatedFontProperty** 列中输入 **BU**。

## 6 保存文件，将“XML 数据”用作文件类型。



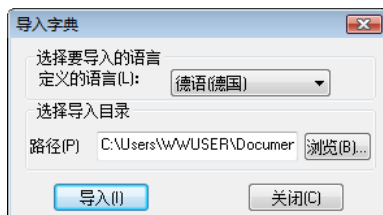
**重要** 如果另存为其它文件类型（如 XML 电子表格），则 Excel 会更改构架，并且会导致 InTouch HMI 无法加载该文件。如果更改 XML 文件的名称，则运行时语言切换功能无法正常工作。

## 导入翻译的字典文件

InTouch HMI 为导出的每种语言创建一个字典文件。InTouch HMI 还为应用程序中的每个 SmartSymbol 创建一个单独的字典文件。翻译之后，必须导入每种语言的字典文件，这样才能进行这些语言的运行时语言切换。给定语言的所有字典文件都应该放在相同的文件夹中。

### 要导入翻译的字典文件

- 1 启动 WindowMaker，打开要导入翻译的字典文件的应用程序。
- 2 在**特别菜单**上，指向**语言**，然后单击**导入字典**。此时打开**导入字典对话框**。



- 3 配置导入设置。
  - 在**定义的语言**列表中，单击要导入的语言字典。
  - 在**路径**框中，输入要导入的字典文件所在的路径。单击**浏览**以浏览并选择文件。
- 4 单击**导入**。

- 5 如果重新导入 SmartSymbol 字典文件，则会提示替换现有的文件。

如果导入成功，则出现**导入成功**对话框。

## 导出报警注释进行翻译

您可以导出报警注释进行翻译。

您可以导出以下标记的“报警状态”、“报警类型”以及“报警类”等字段：

- 包含报警注释的所有标记。
- 包含标记注释的所有标记。
- 系统标记，这样便可以将系统标记激活事件时在客户端中显示的注释进行本地化。

## 理解两个英文字符的应用程序 ID

在导出报警与标记注释进行本地化时，必须指定两个英文字符的应用程序 ID。该 ID 供系统内部使用，以区分同名的应用程序所生成的报警。

由于标记同时可以包含标记注释与报警注释，因此在两个英文字符的应用程序 ID 的后面添加 1 与 2 来区分这两个字段。标记注释在 ID 与标记名之间有一个 1。报警注释在 ID 与标记名之间有一个 2。例如，AA1TankLevel 是标记注释，AA2TankLevel 是报警注释。

如果导出应用程序，将删除应用程序 ID 信息。

如果报警数据库包含不带两个英文字符应用程序 ID 的旧数据，并且新记录以某个 ID 为前缀，则 Alarm DB View 控件中的报警注释查询不处理以下运算符：<、<=、> 及 >=。

## 导出报警注释

您可以导出报警注释进行翻译。

---

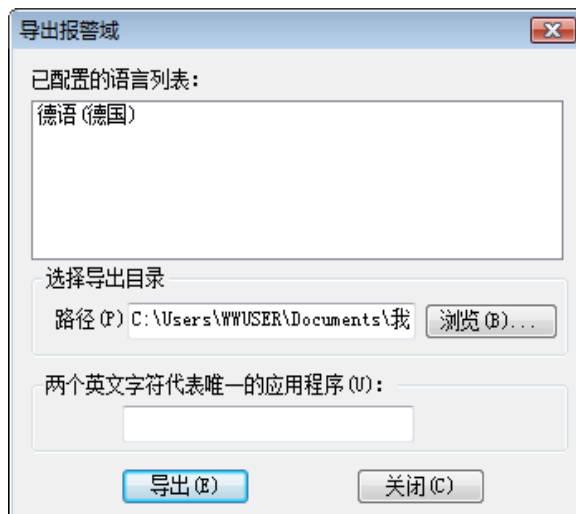
**注意** 在导出报警与标记注释之前，请备份目标目录中的任何文件，以防数据损坏或发生错误。

---

### 要导出报警注释进行脱机翻译

- 1 启动 WindowMaker，打开要导出报警注释进行脱机翻译的应用程序。

- 2 在**特别菜单**上，指向**语言**，然后单击**导出报警域**。此时出现**导出报警域**对话框。



- 3 在**路径**框中，输入要导出字典的文件夹。单击**浏览**以选择现有的文件夹或创建一个新文件夹。
- 4 在**两个英文字符代表唯一的应用程序**框中，输入两个英文字符。ID 只能包含字母数字字符，并且区分大小写。

**注意** 如果先前从这个应用程序中导出了报警或标记注释，则在下次导出它们时必须使用相同的两个英文字符的应用程序 ID。如果输入两个英文字符的新应用程序 ID，InTouch HMI 会为所有的报警与标记重新生成 ID，这样会导致丢失现有的全部翻译。

- 5 单击**导出**将信息导出到 XML 字典文件。

InTouch HMI 为配置的每种语言创建一个单独的导出文件。不同语言的所有字典文件都导出到指定的单个目录。

对于导出的任何语言，如果存在重复的文件，则会提示文件的名称。您可以取消导出或继续执行导出操作。

如果导出成功，则出现**导出成功**对话框。

**备注** 如果标记字典中配置的报警注释的长度超过 127 个字符，或是标记注释的长度超过 46 个字符，则不导出该报警或标记注释。在导出过程结束时，会通知您注释没有导出到字典文件，并在导出目录中创建了 AlarmComment.log 或 TagComment.log 文件。

- 6 单击**关闭**以返回 WindowMaker 窗口，或单击**关闭并启动“资源管理器”**以打开包含字典文件的文件夹。

## 导出到现有的报警注释文件

导出报警与标记注释供脱机翻译之后，可能需要对应用程序进行更改，这可能又会要求再次导出报警与标记注释。如需有关详细信息，请参阅第 151 页的“导出报警注释进行翻译”。



如果多次导出到相同的目录，则出现**确认文件替换**对话框。



单击**是**使用自上次导出以来所添加的任何新字符串和语言信息来更新现有的字典文件。如果现有的字典文件包含任何短语的翻译，并且先前已导入 InTouch，则那些翻译会保留下来。如果自上次导出以来从应用程序中删除了任何短语，则也会从字典文件中删除它们。

单击**全部为是**以更新 InTouch HMI 中配置的所有语言的现有字典文件。

单击**否**或**全部为否**以分别防止覆盖现有的文件或所有语言的现有文件。

如果再次导出，所有报警注释、报警域以及标记注释的现有翻译都会保留下来。

## 编辑字典文件

创建字典文件之后，需要编辑字符串。

字典文件的名称根据两个英文字符的应用程序 ID 与导出的语言进行创建。例如，如果配置的语言是“中文 (PRC)-2052”，两个英文字符的应用程序 ID 是 **AA**，则产生的文件名是 **AA\_2052\_AlarmComment.xml**。编写该文件所使用的 XML 架构与运行时语言切换文件的相同。

字典文件的一般结构如下：

```
<?xml version="1.0" encoding="utf-8" ?>
- <ArrayOfAlarmCommentPhraseItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <AlarmCommentPhraseItem Phrase="Do not edit">
  <Translation>Provide translation in this column</Translation>
  <Context>Do not edit</Context>
  <PhraseID>0</PhraseID>
  <LanguageID>0</LanguageID>
  <ForeignLanguageID>0</ForeignLanguageID>
</AlarmCommentPhraseItem>
- <AlarmCommentPhraseItem Phrase="System">
  <Translation />
  <Context>$System : System Tag Comment</Context>
  <PhraseID>AA1$System</PhraseID>
  <LanguageID>1033</LanguageID>
  <ForeignLanguageID>2052</ForeignLanguageID>
</AlarmCommentPhraseItem>
</ArrayOfAlarmCommentPhraseItem>
```

输入翻译字符串的翻译。请勿更改其它任何信息。

您可以覆盖有些“报警状态”、“报警类型”以及“报警类”的值。这些值的最大允许长度是 50 个字符。

对于 InTouch 生成的报警，以下“报警状态”值可以覆盖：

要覆盖的值	要显示的缺省字符串
UNACK_RTN	UNACK_RTN
ACK_RTN	ACK_RTN
UNACK_ALM	UNACK_ALM
ACK_ALM	ACK_ALM

对于 InTouch 生成的报警，以下“报警类型”值可以覆盖：

要覆盖的值	要显示的缺省字符串
SPC	SPC
HIHI	HIHI
HI	HI
LO	LO
LOLO	LOLO
MINDEV	MINDEV
MAJDEV	MAJDEV
ROC	ROC
DSC	DSC
OPR	OPR
LGC	LGC
DDE	DDE
SYST	SYST
USER	USER
PRO	PRO
LOGON_FAILED	LOGON_FAILED

对于 InTouch 生成的报警，以下“报警类”值可以覆盖：

要覆盖的值	要显示的缺省字符串
DEV	DEV
ROC	ROC

要覆盖的值	要显示的缺省字符串
DSC	DSC
EVENT	EVENT
VALUE	VALUE

## 导入翻译的报警注释

翻译字符串之后，必须导入每种语言的字典文件，这样才能进行这些语言的运行时语言切换。

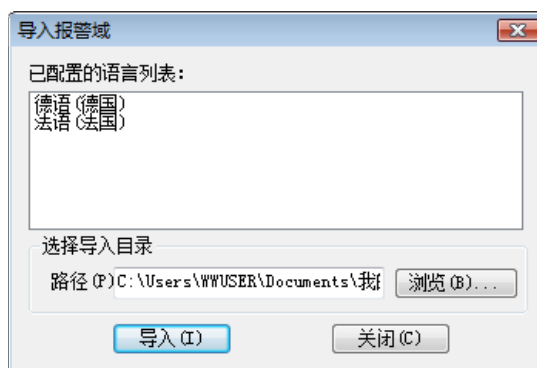
导入翻译的报警注释字典文件之后，它们会复制到应用程序目录内各自语言的文件夹中。

现有应用程序的任何已翻译的报警注释都会被导入的文件的内容覆盖掉，应用程序版本 (\$AppVersion) 按 1 进行递增。

要从其它节点导入多个字典文件以支持多个节点上报警域的本地化，则可以将其它节点上已翻译的字典文件复制到单个目录。选择此目录作为导入路径。多个字典文件可以在单个导入操作中进行导入。InTouch HMI 基于所导入的语言自动创建文件路径。

### 要导入已翻译的报警注释文件

- 1 启动 WindowMaker，打开要导入翻译的字典文件的应用程序。
- 2 在**特别**菜单上，指向**语言**，然后单击**导入报警域**。此时出现**导入报警域**对话框。



- 3 在**路径**框中，输入要导入的字典文件的路径，或者单击**浏览**以浏览并选择文件。
- 4 单击**导入**。如果字典文件中没有完成任何翻译，则出现一个对话框，提示说没有已翻译的字典文件可以导入。
- 5 单击**确定**。如果导入成功，则出现**导入成功**对话框。

## 在运行时测试语言切换功能

在应用程序中启用运行时语言切换之后，需要测试语言切换功能。报警与标记注释以及报警域的语言切换只能在 AlarmViewer 与 AlarmDBView 控件中查看。

在处理本地化版的报警与标记注释时，应注意以下问题：

- 如果报警或标记注释没有翻译成 `$Language` 所指定的语言，则报警客户端显示缺省的注释。
- 如果 Alarm Viewer 控件从多个供应器进行查询，则只要应用程序有远程节点应用程序的已翻译的字典文件，来自远程节点的报警注释、标记注释以及报警域也会显示已翻译的版本。
- 如果确认报警并提供注释，则报警客户端显示此注释，而不是本地化版的报警注释。
- Alarm Viewer 控件处于冻结模式时，即使进行语言切换，也不会显示切换后的语言。取消控件的冻结时，控件使用已翻译的字符串进行更新。
- Alarm Viewer 控件的本地化仅用于控件的显示。即便切换了语言，所有脚本函数仍返回缺省的字符串。
- Alarm DB Logger 只将数据的缺省语言字符串存储在数据库中。本地化版的字符串不存储在数据库中。
- 报警域的唯一 ID（如 EVENT 与 ACK）是预定义的，在不同节点上的多个字典文件中使用的是相同的 ID。报警客户端从第一个加载的字典文件中选择翻译，并忽略其它字典文件中的翻译。理想的情况是，所有字典文件中的报警域在一种语言中都应该有相同的翻译。对于给定语言的相同报警状态，多个报警客户端（Alarm DB View 与 Alarm Viewer 控件）使用相同的翻译。
- 对于报警注释，翻译的文本截断为 131 个字符；对于标记注释，截断为 50 个字符。

### 要测试语言切换功能

- 1 在 WindowViewer 中打开应用程序。
- 2 在**特别菜单**上，指向**语言**，然后单击要切换到的语言的名称。  
此时会加载并显示相应的已翻译的字典文件（如果存在）中的信息。
- 3 如果添加了用于切换语言的按钮，则可以单击此按钮来测试脚本。

## 将本地化版的文件分发到网络应用程序开发客户端

包含本地化版的报警注释、标记注释以及报警域的文件作为 InTouch 应用程序的一部分，分发到各个 “网络应用程序开发” (NAD) 客户端。接收到包含报警注释的更新文件时，必须重新启动 WindowViewer 才能在支持的报警客户端中看到翻译的报警注释。

如果结合使用语言切换与 “网络应用程序开发” (NAD)，请将 NAD 客户端节点的改变模式设置为 “重新启动 WindowViewer” 或 “提示用户重新启动 WindowViewer”。



## 第 7 章

# 在运行时查看应用程序

您使用 WindowViewer 运行 InTouch 应用程序。专为在 ArchestrA Application Server 环境中使用而设计的应用程序称为 InTouchView 应用程序。这些应用程序在 WindowViewer 中运行，但由 Application Server 提供大多数 HMI 功能。

## 关于 WindowViewer

WindowViewer 为 InTouch 应用程序提供运行时环境。基于应用程序的工作要求，可以配置 WindowViewer 支持应用程序的方式。例如，根据应用程序的安全性要求，可以配置 WindowViewer 向操作员提供的菜单与命令。

## 自定义运行时环境

您可以通过设置属性来自定义运行时 WindowViewer 环境。

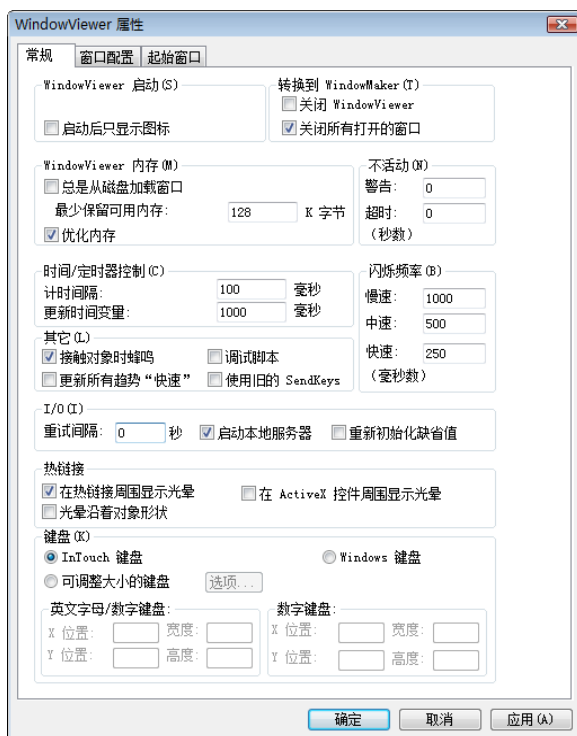
- 常规属性确定运行 InTouch 应用程序的环境条件。
- 窗口配置属性确定用户与 WindowViewer 中运行的 InTouch 应用程序进行交互时可以访问的菜单、命令以及窗口组件。

### 配置常规 WindowViewer 属性

您可以通过配置一组常规属性来确定 WindowViewer 在运行 InTouch 应用程序时的特征。修改这些常规属性之后，必须重新启动 WindowViewer 才能使更改生效。

- 1 打开 WindowMaker。

- 2 在**特别**菜单上，指向**配置**，然后单击 **WindowViewer**。此时出现 **WindowViewer 属性** 对话框。
- 3 单击**常规**选项卡。



- 4 在 **WindowViewer 启动** 区域，如果希望 WindowViewer 在启动时最小化，请选择**启动后只显示图标**复选框。  
仅当使用 WindowViewer 为其它 I/O 互联应用程序收集数据时，才选择**启动后只显示图标**选项。
- 5 在**转换到 WindowMaker** 区域中，执行以下操作：
  - 如果希望在启动 WindowMaker 时自动关闭 WindowViewer，请选择**关闭 WindowViewer** 复选框。  
选择此选项时，同时自动选中 **WindowMaker 属性 / 常规** 选项卡上的**转换到 WindowViewer 时关闭**选项。如果内存足够，并且使用快速切换功能在 WindowViewer 与 WindowMaker 之间切换，则应该清除此选项。
  - 如果希望由 WindowViewer 转换到 WindowMaker 时自动关闭所有打开的窗口，选择**关闭所有打开的窗口**复选框。
- 6 如果应用程序在较旧的硬件上运行，您可能需要配置 **WindowViewer 内存** 区域中的选项。更改这些选项之前，请咨询“Wonderware 技术支持”：
  - 如果希望关闭时应用程序窗口从磁盘加载这些窗口而不是保存在内存中，请选择**总是从磁盘加载窗口**复选框。如果运行 InTouch 应用程序的计算机遇到内存不足的问题时，请选择此选项。



- 在**最少保留可用内存**框中，以 KB 为单位输入需要为其它 Windows 应用程序保留的可用内存量，这些应用程序与 InTouch 应用程序在相同的计算机上运行。
  - 选择**优化内存**复选框，以提高窗口文本字段的更新速率。如果运行 WindowViewer 的计算机遇到内存不足的问题，请清除此选项。
- 7 在**不活动区域**中，设置操作员不活动警告与超时时段。  
如需有关设置警告与超时时段的详细信息，请参阅第 110 页的“配置不活动超时”。
- 8 在**时间 / 定时器控制区域**中，执行以下操作：
- 在**计时间隔**框中，输入 InTouch HMI 用于检查内部定时器的间隔。  
此间隔确定“应用程序运行期间”、“窗口显示期间”、“条件为真时（期间） / 为假时（期间）”以及键与触动物按钮动作“按下期间” QuickScript 可以启动的时间。  
此选项设置 INTOUCH.ini 文件中 TimerTickInterval 参数的值。对于按计划每 100 毫秒运行一次的脚本，应该将“计时间隔”设置为不超过 50 毫秒。在运行 Windows XP 或 Windows 2003 的计算机上，计时间隔下限为 10 毫秒。
  - 在**更新时间变量**框中，输入以毫秒为单位的间隔，系统变量（如 \$Msec、\$Second 或 \$Minute）按此间隔更新时间。  
我们建议使用缺省设置 1000 毫秒。将此选项设置为零可以防止更新时间变量。
- 9 在**其它区域**中，执行以下操作：
- 如果希望 InTouch 应用程序在操作员选择窗口上的触控对象时发出蜂鸣声，请选择**接触对象时蜂鸣**。
  - 如果希望趋势对象的更新速度更快，请选择**更新所有趋势“快速”**。  
仅当应用程序窗口上没有任何对象与运行时趋势重叠时，才能选择此选项。如果选择此选项并且有任何对象与趋势重叠，则可能无法正确绘制对象。
  - 如果希望每次运行 QuickScript 时将一条消息写入 Logger，请选择**调试脚本**。  
如果从**窗口配置**属性页中选择**调试**，则可以从 WindowViewer 的**特别菜单**中打开或关闭 QuickScript 记录功能。
  - 如果使用以 InTouch 3.26 或更早版本开发的国际化应用程序，请选择**使用旧的 SendKeys** 复选框。
- 10 在**闪烁频率区域**中，为**慢速**、**中速**以及**快速**闪烁动画链接输入以毫秒为单位的间隔长度。

**11** 在 **I/O** 区域中，执行以下操作：

- 在**重试间隔**框中，输入连接尝试失败之后 InTouch 应用程序重新尝试连接到 “I/O 服务器” 之前等待的秒数。InTouch 第一次就能成功连接到 I/O 服务器时，I/O **重试间隔**没有任何效果。
- 如果希望在启动 WindowViewer 并且尝试通讯的 “I/O 服务器” 不在运行时出现一个对话框，请选择**启动本地服务器**复选框。
- 如果希望使用缺省设置重新初始化 “访问名”，请选择**重新初始化缺省值**复选框。此时忽略指定给 “访问名” 的当前值，而使用它们的原始设置重新进行初始化。

**12** 在**热链接**区域中，执行以下操作：

- 如果希望在用户将光标移到运行时屏幕上的对象时突出显示该对象，请选择**在热链接周围显示光晕**复选框。
- 如果希望在用户将光标移到对象时沿该对象的轮廓线出现突出显示的光晕，请选择**光晕沿着对象形状**复选框。
- 如果希望在 ActiveX 控件周围显示光晕，请选择在**ActiveX 控件周围显示光晕**复选框。

**13** 在**键盘**区域中，选择希望使用的键盘类型（如果有）。

如需有关设置 WindowViewer 中键盘选项的详细信息，请参阅 *InTouch® HMI 可视化指南* 中的第 4 章 “设置对象动画效果”。

**14** 单击**确定**。

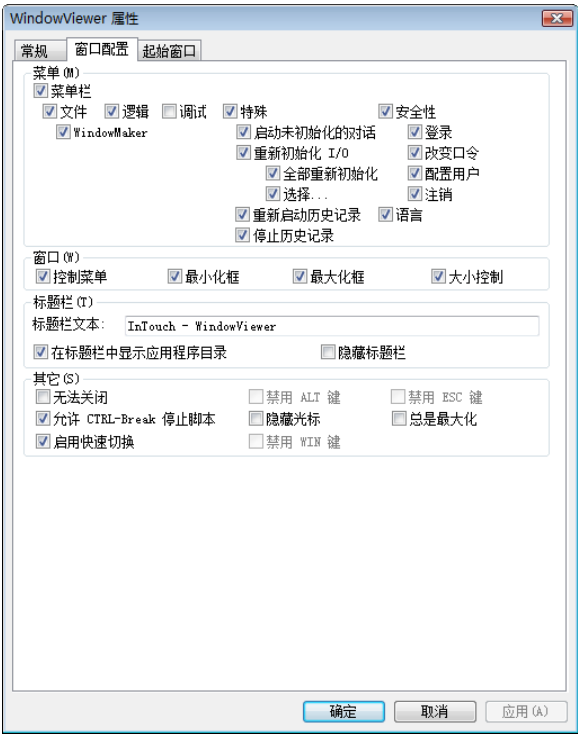
## 配置 WindowViewer 的视觉特征

您可以通过配置属性来确定 WindowViewer 运行 InTouch 应用程序时的视觉特征。这些属性确定 WindowViewer 窗口上出现的菜单、命令以及标准控件。

### 要配置 WindowViewer 的视觉特征

- 1 打开 WindowMaker。
- 2 在**特别菜单**上，指向**配置**，然后单击 **WindowViewer**。此时出现 **WindowViewer 属性**对话框。

3 单击窗口配置选项卡。



4 选择视觉特征对应的复选框。

窗口配置属性	如需有关详细信息，请参阅：
菜单	第 115 页的“在运行时隐藏菜单项”
窗口	第 115 页的“在运行时隐藏菜单项”
标题栏	第 115 页的“在运行时隐藏菜单项”
其它	第 113 页的“锁定系统键”

5 重新启动 WindowViewer。

使用 WindowViewer 窗口

典型的 InTouch 应用程序至少包含多个窗口，操作员可以通过与这些窗口进行交互来管理工业过程。基于从 **WindowViewer 属性对话框窗口配置**选项卡中设置的属性，操作员可以从 WindowViewer 文件菜单中运行标准命令来打开与关闭窗口。

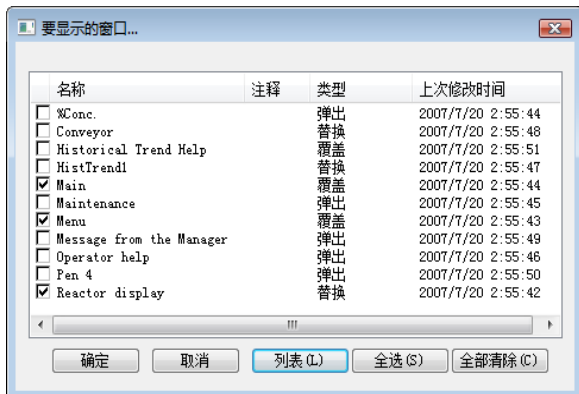
常用对话框功能

如果将 WindowViewer 配置成显示文件菜单，则操作员可以打开或关闭 InTouch 应用程序窗口。操作员从文件菜单中单击**打开窗口**或**关闭窗口**命令时，出现所选命令对应的对话框。

适合所选命令的所有窗口的名称都会出现在列表中。例如，单击 **打开窗口** 命令之后，出现 **要显示的窗口** 对话框。



单击 **详细资料** 从列表视图更改为详细资料视图。详细资料视图显示窗口的类型与最近一次修改窗口的日期与时间。



在详细资料视图中，可以通过单击窗口所在行的任何部分（而不仅是复选框）来选择与取消选择任何未打开的窗口。选中时整行突出显示。

- 要打开所选窗口，请单击 **确定**。
- 要取消选择并关闭对话框，请单击 **取消**。
- 要让对话框返回列表视图，请单击 **列表**。
- 要选择列出的所有窗口，请单击 **全选**。
- 要取消选定的所有窗口，请单击 **全部清除**。
- 要按升序或降序对列表进行排序，请单击列标题。

### 从 WindowViewer 中打开窗口

如果 WindowViewer 配置成显示 **文件** 菜单，则操作员可以打开 InTouch 应用程序窗口。

### 要从 WindowViewer 中打开窗口

- 1 在文件菜单上，单击**打开窗口**。此时出现**要显示的窗口**对话框。
- 2 单击要打开的每个窗口的名称旁边的复选框。
- 3 单击**确定**以关闭对话框并打开所选的窗口。

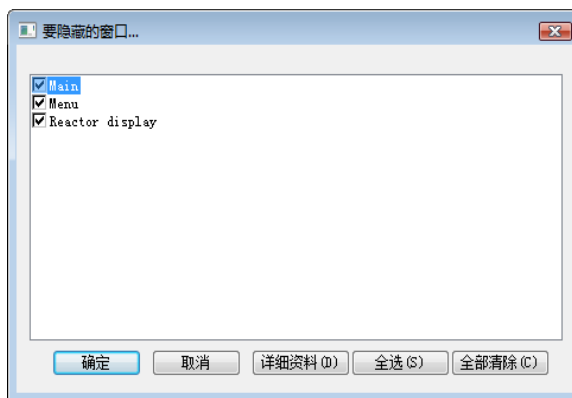
**备注** 如果选择了“替换”型窗口，它会关闭与之相交的任何窗口。

### 从 WindowViewer 中关闭窗口

如果 WindowViewer 配置成显示**文件**菜单，则操作员可以关闭 InTouch 应用程序窗口。

#### 要关闭打开的窗口

- 1 在文件菜单上，单击**关闭窗口**。此时出现**要隐藏的窗口**对话框。



- 2 单击要关闭的一个或多个窗口的名称旁边的复选框。
- 3 单击**确定**以关闭所选窗口。

### 从 WindowViewer 转换到 WindowMaker

开发 InTouch 应用程序时，通过单击**文件**菜单中的 **WindowMaker** 命令或工具栏上的**开发**命令，可以轻松地在 WindowMaker 与 WindowViewer 之间进行转换。这就是所谓的快速切换。

快速切换仅针对快速开发测试。不要在生产环境中使用它。您可以隐藏切换到 WindowMaker 的命令。

### 要从 WindowViewer 转换到 WindowMaker

- 1 在文件菜单上，单击 **WindowMaker**。此时出现**要编辑的窗口**对话框。



- 2 单击转换到 WindowMaker 时要打开的每个窗口的名称旁边的复选框。
- 3 单击**确定**以关闭对话框并转换到 WindowMaker。

**备注** 如果应用程序开发人员在开发期间配置 WindowViewer 的属性时选择**关闭 WindowViewer** 选项，则转换到 WindowMaker 时 WindowViewer 会自动关闭。

## InTouchView

InTouchView 应用程序显示专为在 ArchestrA Application Server 环境中使用而设计的可视化界面。InTouchView 应用程序在 WindowViewer 中运行，但由 Application Server 提供大多数 HMI 功能。

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。
- 无法生成报警。不过，您可以显示与确认来自远程报警供应器（如 ArchestrA 对象）的报警。
- 不记录应用程序数据或事件。InTouchView 应用程序仅生成与 SYS 和 USER 相关的事件。
- 仅可以使用 ArchestrA 安全性。
- 无法充当其它 InTouch 应用程序、InTouchView 应用程序或 WWClient 这类客户端的服务器。

您可以使用 WindowMaker 来开发 InTouchView 应用程序。下表显示创建 InTouchView 应用程序时哪些 WindowMaker 命令与“标记名字典”选项不可用。

- 不可用的特别菜单命令：
  - 访问名
  - 报警组
  - 配置 ... 报警
  - 配置 ... 历史记录
  - 配置 ... 分布式名称管理器
- 不可用的标记名字典选项：
  - 报警
  - 详细和报警
  - 记录数据
  - 记录事件
  - 优先级

您像对待 InTouch 应用程序那样运行 InTouchView 应用程序。只要在 WindowViewer 中启动应用程序即可。

## 创建新的 InTouchView 应用程序

通过在创建应用程序时设置“应用程序管理器”中的某个选项，可以确定 InTouchView 应用程序。

### 要创建新的 InTouchView 应用程序

- 1 启动“应用程序管理器”。
- 2 在文件菜单上，单击**新建**。此时出现**创建新应用程序**向导。
- 3 按**创建新应用程序**向导中的步骤进行操作。向导包含一个对话框，提供用于输入应用程序名、提供描述以及将它指定为 InTouchView 应用程序的选项。



- 4 输入 InTouchView 应用程序的名称与描述。
- 5 选择 **InTouchView 应用程序** 复选框。
- 6 单击**完成**。此时系统创建一个新的 InTouchView 应用程序，并将安全性设置为 ArchestrA 模式。

## 在 InTouch 与 InTouchView 之间转换应用程序

您可以将 InTouch 应用程序转换为 InTouchView 应用程序，反之亦然。运行从 InTouchView 转换到 InTouch 的应用程序时，要求使用完整版的 InTouch 许可证。您可以将 InTouch 应用程序转换为 InTouchView，并仍然使用 InTouch 许可证来运行该应用程序。

您无法转换当前正在 WindowViewer 中运行的应用程序。

### 将 InTouchView 应用程序转换为 InTouch 应用程序

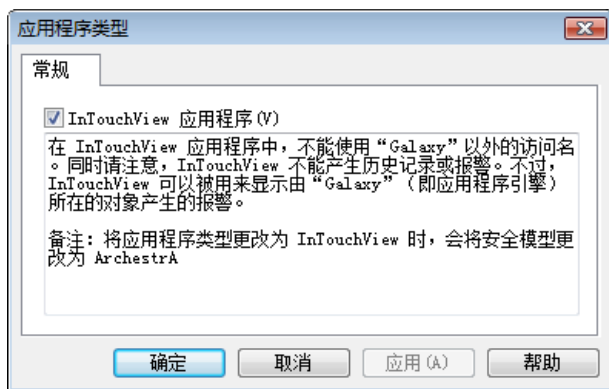
您可以将 InTouchView 应用程序转换为 InTouch 应用程序。例如，如果需要应用程序开始访问 Application Server Galaxy 之外的数据源中的数据，则需要这样做。

在转换应用程序之后，可以访问 InTouchView 应用程序所不能提供的所有 WindowMaker 功能。

转换后的应用程序仍然使用 ArchestrA 安全性，但是您可以更改安全性类型。

### 要将 InTouchView 应用程序转换为 InTouch 应用程序

- 1 在 WindowMaker 中打开 InTouchView 应用程序。
- 2 在**特别菜单**上，单击**应用程序类型**。此时出现**应用程序类型**对话框，其中 **InTouchView 应用程序** 框处于选中状态。



- 3 清除 **InTouchView 应用程序** 复选框，然后单击**确定**。

### 要更改安全类型

- 1 在 WindowMaker 中打开 InTouch 应用程序。



- 2 在**特别菜单**，指向**安全性**，然后指向**选择安全类型**，以显示支持的 InTouch 安全类型列表。
- 3 为 InTouch 应用程序选择安全性的类型。

如需有关安全性的详细信息，请参阅第 5 章“保护 InTouch 安全”。

### 将 InTouch 应用程序转换为 InTouchView 应用程序

如果应用程序仅需要连接到 Application Server，则可以将 InTouch 应用程序转换为 InTouchView。

在将应用程序转换为 InTouchView 之后，有些 WindowMaker 功能变为不可用。

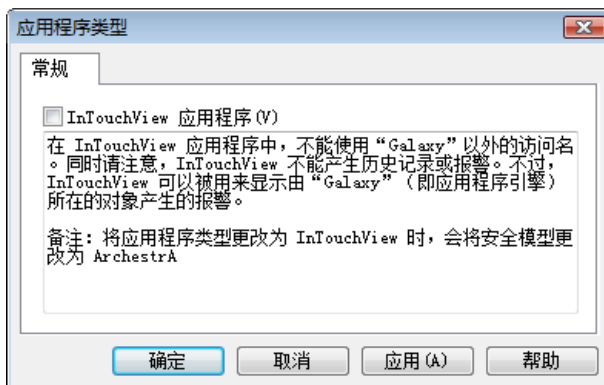
---

**重要** 您必须删除 Galaxy 之外的所有“访问名”，才能将 InTouch 应用程序转换为 InTouchView。如果这些“访问名”没有删除，则会在尝试转换期间显示一条消息。

---

#### 要将 InTouch 应用程序转换为 InTouchView 应用程序

- 1 在 WindowMaker 中打开 InTouch 应用程序。
- 2 在**特别菜单**上，单击**应用程序类型**。此时出现**应用程序类型**对话框。



- 3 选择 **InTouchView 应用程序** 复选框。
- 4 单击**确定**。出现消息时，单击**确定**。

## InTouchView 许可证

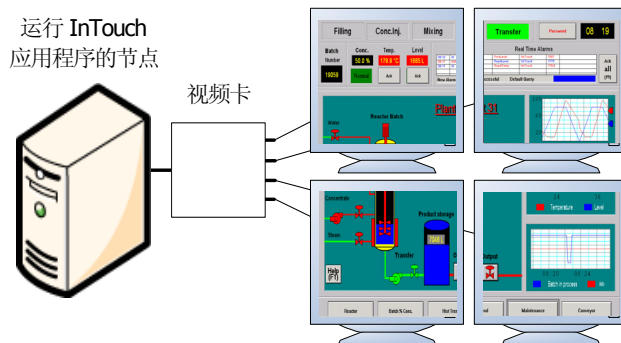
InTouchView 应用程序启动时，WindowViewer 要求从“许可证管理器”提供一个特定的 InTouchView 许可证。如果 InTouchView 许可证不可用，则“许可证管理器”要求提供一个全功能的 InTouch 许可证。如果成功读取 InTouch 许可证，WindowViewer 将启动应用程序。否则 WindowViewer 指出 InTouchView 许可证不可用，并简要说明出现的问题。WindowViewer 提供退出、重试或按演示模式运行 InTouchView 应用程序的选项。

## 第 8 章

# 设置多监视器系统

多监视器系统在多个监视器上同时显示 InTouch 应用程序。多监视器配置合在一起创建一个复合屏幕，由连接到运行 InTouch 应用程序的计算机的所有监视器组成。每个监视器可以显示屏幕的一部分，或是只显示单个窗口组件（如数字小键盘）。

运行 InTouch 应用程序时，可以在监视器之间移动鼠标，以及将窗口从一个监视器拖到另一个监视器。同样，在某些多监视器配置中，可以在所有的监视器上显示整个 InTouch 应用程序窗口，具体如下图所示。



## 多监视器配置

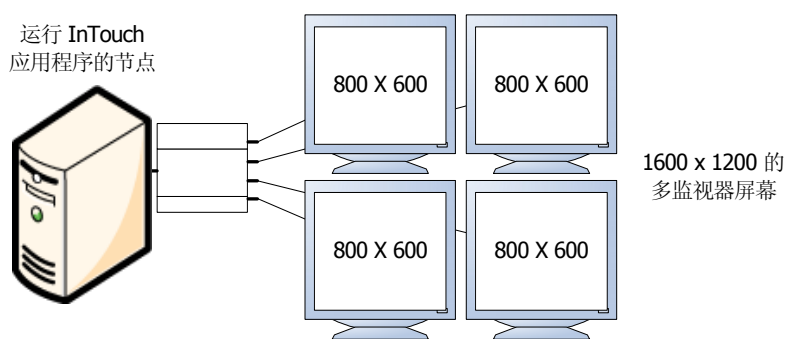
您可以使用两种基本的监视器配置。

- 单显卡
- 多显卡

每种配置都有特有的硬件、软件以及配置要求。同样，每种配置都支持不同的多监视器功能集。

### 单显卡配置

在单显卡配置中，计算机上安装单个显卡，有多个连接到监视器的输出端口。



复合屏幕分辨率是每个监视器的水平与垂直分辨率之和。例如，一个普通显卡连接四个 17 寸监视器，像立方体一样堆起；两个在底部，两个在顶部。在上图中，每个监视器都按 800 x 600 像素的分辨率来运行。此时复合虚拟屏幕分辨率为 1600 x 1200 像素。

#### 单显卡配置的特征

单显卡驱动程序有以下特征：

- 单个显卡同时驱动所有的监视器，以形成一个大屏幕。
- 连接的所有监视器的属性都可以使用一组屏幕值来配置。
- 在配置的尾行，复合屏幕在所有的监视器上显示 Windows 任务栏。
- Windows 应用程序可以最大化，以填满所有的监视器。

### 单显卡驱动程序的特征

下图显示 Windows 显示设置对话框，为连接到单个显卡（带多输出端口）的所有监视器配置驱动程序。

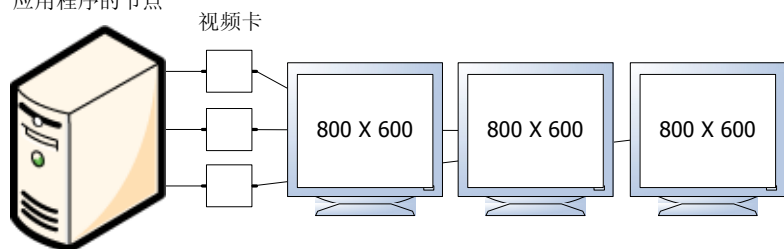


在此图中，分辨率设置是四个监视器并排放置成一行的分辨率。每个监视器的分辨率都是 1024 x 768。这些分辨率加到一起，复合屏幕分辨率为 4096 x 768。您只需配置单个监视器的分辨率、颜色深度以及刷新速率。分辨率设置应用于连接到单个显卡的所有监视器。

## 多显卡配置

在多显卡配置中，计算机上安装多个显卡。每个显卡将一个监视器连接到运行 InTouch 应用程序的计算机。

运行 InTouch  
应用程序的节点



### 多显卡配置的特征

“动态分辨率转换” (DRC) 同发布的其它功能一起，可以消除屏幕分辨率的限制。在 NAD 架构中，您在一个开发节点上创建并维护 InTouch 应用程序，然后将它复制到多个 View 节点。

DRC 使所有 view 节点都可以显示应用程序，即使是以不同的屏幕分辨率运行的节点也是如此。

DRC 使每个 View 节点根据多个用户定义的选项（包括自定义分辨率）来缩放应用程序。这种缩放在 WindowViewer 编译应用程序时进行，而且不要求使用 WindowMaker。由于每个 View 节点都可以使用不同的 DRC 设置，因此必须配置每个单独的 View 节点。

DRC 可以轻松支持多监视器系统。只需从 DRC 分辨率转换选项中进行选择，便可以在整个复合屏幕或其中一部分上显示 InTouch 应用程序。

### 多显卡驱动程序的特征

下图显示 Windows 显示设置对话框，对于连接到计算机（运行 InTouch 应用程序）上安装的各个单独显卡的所有监视器，可以为其配置驱动程序。



您可以单击显示设置对话框中带编号的矩形，以选择要配置的监视器。您可以通过对带编号的矩形进行排列，使之符合监视器的实际位置。屏幕分辨率、颜色深度以及刷新速率仅应用于您选择的监视器。

## 规划多监视器应用程序

要为应用程序设置多个监视器，必须：

- 选择多监视器显卡
- 确定应用程序屏幕分辨率
- 确定用于显示应用程序的监视器数
- 确定应用程序窗口的位置

### 选择多监视器显卡

“Wonderware 技术支持”可以为您提供一份建议的显卡列表，这些显卡支持多监视器 InTouch 应用程序。

在选择显卡之前，可以从“技术支持”获取详细信息以回答以下问题：

- 显卡支持哪些 InTouch 版本？
- 显卡是支持单显卡还是多显卡配置？
- 显卡的建议驱动程序是什么？
- 显卡的建议配置设置是什么？

### 确定应用程序屏幕分辨率

通过确定总屏幕分辨率并了解查看区域的准确尺寸，可以简化为多监视器环境创建应用程序的过程。

创建一个显示总监视器配置的图形。图形应该显示每个监视器的分辨率，以及所有监视器加到一起的组合分辨率。此图形帮助您以图形化方式显示每个监视器的水平与垂直像素范围。

例如，如果有一个由两台水平监视器组成的复合屏幕，每个监视器的屏幕分辨率为 800 x 600，则第二台监视器的左上角像素位置为 800 x 0 像素。第一个监视器的屏幕像素计数从 0 到 799，第二个监视器的像素计数从 800 到 1599。通过将该图形用作基准，可以确定应用程序窗口在复合多监视器屏幕上的位置。

### 确定显示应用程序的监视器数

通过使用类似于生产环境的开发环境，可以简化创建多监视器 InTouch 应用程序的工作。使用多监视器开发环境不是在所有情况下都有可能。仅有一个监视器连接到用于开发 InTouch 应用程序的计算机时，通过开发窗口并根据估计的显示需要来配置窗口的尺寸与位置，仍然可以构建多监视器应用程序。

使用 WindowMaker **窗口属性** 对话框修改窗口的特征。使用鼠标右键单击 WindowMaker **窗口** 窗格中所列窗口的名称，然后单击快捷菜单中的 **属性**，以显示 **窗口属性** 对话框。



**X 位置**与 **Y 位置**值确定窗口在屏幕左上角的水平与垂直像素位置。水平与垂直像素刻度的原点位于屏幕左上角。

**窗口宽度**与**窗口高度**设置确定窗口的总体尺寸。例如，您可以使用以下设置配置窗口：

- X 位置 = 1024
- Y 位置 = 0
- 窗口宽度 = 1024
- 窗口高度 = 768

多监视器配置由水平放置成一排四个监视器组成。每个监视器的分辨率都是 1024 X 768。总的复合屏幕分辨率为 4096 X 768。

通过将窗口的水平原点设置为 1024、垂直原点设置为 0，可以在运行时强制此窗口出现在第二个监视器上。此窗口占据第二个监视器的整个屏幕表面。

## 确定应用程序窗口的位置

为多监视器环境开发 InTouch 窗口时，可以使用多种不同的配置。

### 窗口显示在强制的位置

一种方法是，只需开发窗口并强制它显示在指定的位置。确保 WindowViewer 最大化在所有监视器的总查看区域上。这允许在指定的显示器上显示 InTouch 应用程序窗口。

您可以使用 InTouch 安全性功能来拒绝对 Windows 桌面的访问。



### 可以手工移动窗口

另一个选项是开发一个应用程序，将其中的窗口手工移到所选的监视器，使单个应用程序在不同的监视器配置上运行。这涉及到以下要求：

- 应用程序中的所有窗口必须都是**弹出型**。
- **WindowViewer** 主窗口可以小一点，不占据所有的监视器。不过，由于 **InTouch** 没有最大化，因此不能在此配置中使用 **InTouch** 安全性来拒绝访问 **Windows** 桌面。

在此配置中使用的是弹出窗口，可以轻松地移到任何监视器，而不论 **WindowViewer** 主窗口的位置如何。弹出窗口不必保持在 **WindowViewer** 父窗口中。您可以缩放主窗口的大小并将它移到监视器的一角，使所有弹出窗口都可以自由移到所选的监视器上。

### 窗口根据环境自动放置

最后一种方法包含以上方法之外的一个额外步骤。此步骤使应用程序可以根据使用的环境自动放置窗口。这是配置中最复杂的部分，要求进行大量的脚本编写与规划工作。

在此配置中，**ShowAt()** 与 **ShowTopLeftAt()** 脚本函数根据一组缺省的坐标与算法来动态地放置窗口。这可以使用许多不同的方法进行配置，具体取决于应用程序要求。

## 开发多监视器 InTouch 应用程序

您必须给 **InTouch.ini** 与 **Win.ini** 文件中所选的参数指定数值，才能支持多监视器。这些参数使您可以在复合屏幕中适当的位置上放置 **InTouch** 系统对话框与数字小键盘。

### 配置多监视器参数

要启用多监视器支持，需要将一组 **InTouch** 参数添加到 **Windows** 的 **Win.ini** 文件。这些参数可以为运行 **InTouch** 应用程序的节点与每个监视器的分辨率启用多监视器支持。

#### 要在节点上配置多监视器设置

- 1 在运行 **InTouch HMI** 软件的计算机上，编辑 **Windows** 文件夹中的 **Win.ini** 文件。

2 找到 Win.ini 文件中的 [InTouch] 部分，并添加以下参数：

参数	描述
<b>MultiScreen=1</b>	值为 1 时启用多监视器模式。值为 0 时禁用多监视器模式。
<b>MultiScreenWidth=nnnn</b>	以像素为单位的单屏幕宽度。
<b>MultiScreenHeight=nnnn</b>	以像素为单位的单屏幕高度。

例如，如果希望在两个水平监视器上以 2560 x 1024 的屏幕分辨率显示 InTouch 应用程序，请输入以下内容：

```
[InTouch]
MultiScreen=1
MultiScreenWidth=1280
MultiScreenHeight=1024
```

## 配置屏幕分辨率转换

您可以指定一个参数值，在运行不同屏幕分辨率的节点之间迁移 InTouch 应用程序窗口时可以维持当前的分辨率。

ScaleForResolution 参数值确定运行 WindowViewer 的计算机上的分辨率发生更改之后，WindowMaker 是否自动缩放应用程序窗口 (\*.win)。ScaleForResolution 参数不影响 WindowViewer 对话框的分辨率。

### 要在节点上配置屏幕分辨率转换

- 1 编辑运行 InTouch 的计算机上的 InTouch.ini 文件。
- 2 将 ScaleForResolution 参数添加到文件中。

```
ScaleForResolution=1
```

设置为 0 时，禁用分辨率转换。

设置为 1 时，启用分辨率转换。

**备注** 如果 ScaleForResolution 参数没有添加到 InTouch.ini 文件，缺省值是启用 (ScaleForResolution=1)。禁用参数 (ScaleForResolution=0) 时，程序仍然会提示您是否转换分辨率。但并不会发生分辨率转换。

## 部署应用程序与验证多监视器设置

在单监视器系统开发打算在多监视器系统上运行的应用程序时，**ScaleForResolution** 参数变得尤其重要。指定给 **ScaleForResolution** 参数的值确定将应用程序从一个环境移到另一个环境时，是否对其进行缩放。

---

**重要** 在将应用程序移到不同的环境之前，建议先对它进行备份。

---

例如，如果应用程序在使用单监视器（分辨率为 1024 x 768）的计算机上开发，并且打算在使用四个并排放置的监视器（总分辨率为 4096 x 768）的系统上运行，这要求进行应用程序转换。

在多监视器系统上部署应用程序时，会出现一条消息，提示您是否转换应用程序。

如果配置了 **ScaleForResolution .ini** 设置，仍然会看见这条消息，但应用程序不转换，而且随后可以按设计来运行。只需单击是就可以继续启动。

如果 **.ini** 设置尚未配置，则 **InTouch HMI** 会将应用程序中所有图形与窗口转换并缩放为新的分辨率。执行此操作会拉伸并放大所有窗口与图形的显示画面，因此会造成某些不希望的结果。

---

**重要** 在运行应用程序之前，确保目标计算机上也配置了多监视器 **Win.ini** 参数设置。**Win.ini** 设置不随 **InTouch** 应用程序自动传输。

---

## 在运行时验证多监视器支持

您可以从“Wonderware 技术支持”脚本库中下载一个可选脚本函数，此函数可以验证运行 **InTouch** 应用程序的本地节点是否提供多监视器支持。

**WWMultiMonitorNode()** 函数确定节点是否支持多监视器以及连接到该节点的监视器数。

通常，您可以从 **QuickScript** 中运行 **WWMultiMonitorNode()** 函数，以确定指定给运行 **InTouch** 应用程序的节点的监视器数。

下例显示一个 **QuickScript** 语句示例，它将 **WWMultiMonitorNode()** 函数的值指定给 **InTouch** 整型标记。**QuickScript** 可以设置为在 **WindowViewer** 中启动应用程序时运行。

```
{MultiMonitors defined as an integer tag}
MultiMonitors = WWMultiMonitorNode();
{After executing this function Result = 4}
```

**WWMultiMonitorNode()** 读取节点的 **Win.ini** 文件中指定的 **MultiScreen** 参数。**WWMultiMonitorNode()** 函数返回 0 或一个正整数。

- 返回值 0

如果 MultiScreen=0 或 Win.ini 文件 [InTouch] 部分中 MultiScreenWidth 或 MultiScreenHeight 参数错误地设置为 0，则 WWMultiMonitorNode() 返回 0。

- 正整数返回值

如果 MultiScreen=1 并且给 MultiScreenWidth 与 MultiScreenHeight 参数指定了正确的屏幕分辨率值，则 WWMultiMonitorNode() 返回多监视器配置中的监视器数。

## 第 9 章

# 在 Tablet PC 上使用 InTouch

Wonderware 提供一系列使用 Windows XP Tablet PC Edition 并且预安装了 InTouch 的便携式 Tablet PC。这些牢固的 Tablet PC 防水抗震，能够适应大多数工业环境。Tablet PC 也可以在能运行 InTouch 应用程序的其它计算机制造商处购买。

操作员可以携带着 Wonderware Tablet PC 巡视工厂。Tablet PC 运行代表实际工厂过程的 InTouch 应用程序。通过将笔用作屏幕指针或输入设备，操作员可以在屏幕上选择 InTouch 对象，或用它代替键盘直接在屏幕上书写备注。

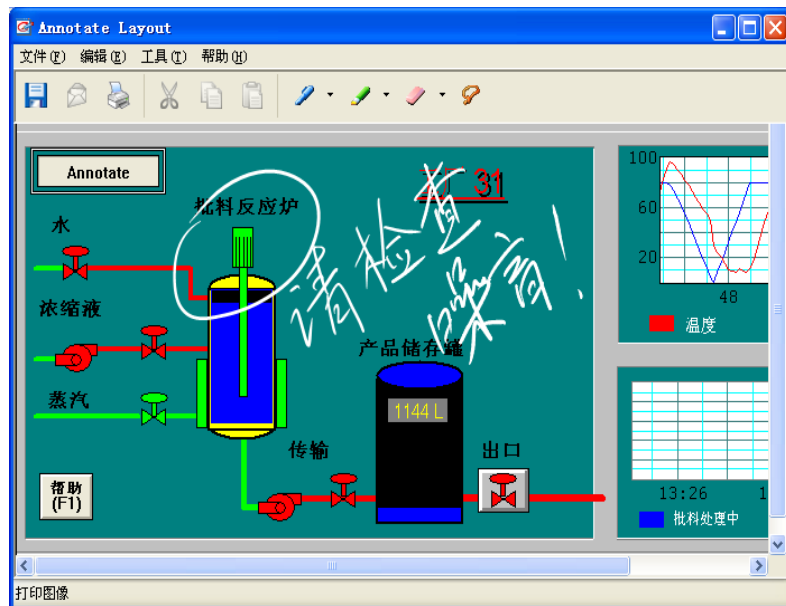


操作员可以书写备注，也可以根据对实际工厂过程的直接观察给正在运行的 InTouch 应用程序作出注解。

## 注解可视化屏幕与将它当作电子邮件消息发送

使用 AnnotateLayout() 脚本函数截取 Tablet PC 上显示的屏幕。仅当 InTouch 运行在使用 Windows XP Tablet PC 操作系统的 Tablet PC 上时，AnnotateLayout() 函数才可以使用。




AnnotateLayout() 函数截取活动 InTouch 窗口可见部分的屏幕截图。捕获的屏幕出现在**注解布局**对话框中。



**注解布局**对话框包含工具栏与菜单选项。该对话框在其客户区显示屏幕截图。您可以使用各种绘图工具作出注解，然后保存、打印，或将屏幕截图作为电子邮件消息发送。

### 给窗口添加注解

要给窗口添加注解，请使用以下工具：

-  • **笔**：绘制和书写注释。
-  • **荧光笔**：使用半透明颜色突出显示窗口中的区域。
-  • **橡皮擦**：删除注解的某些部分。

每种工具都有特定的选项，如大小、颜色或透明度。

- 要设置这些选项，请单击每种工具图标旁的向下箭头，然后单击该选项的命令。
- 要将这些选项恢复成它们的缺省设置，请在**工具**菜单上单击**恢复缺省值**。

## 选择、复制及删除窗口注解

您可以选择、复制及删除希望在窗口中作出的注解。

### 要选择注解



- 1 单击工具栏中的**选择工具**图标。
- 2 按下笔触按钮期间，在您希望选择的注解周围绘制一个区域。

您现在可以剪切、复制或删除所选的注解。

### 要剪切、复制及粘贴对象

- ◆ 使用 Windows 的标准“剪切”、“复制”及“粘贴”命令。

### 要删除注解

- ◆ 执行以下任何操作：
  - 要删除窗口上的所有注解，请在**编辑**菜单上，指向**清除**，然后单击**全部**。
  - 要删除使用选择工具选择的注解，请在**编辑**菜单上，指向**清除**，然后单击**选择**。

## 保存、打印及使用电子邮件发送注解的窗口

给窗口作出注解之后，可以将它保存为图像文件，打印它，或者将它当作电子邮件附件来发送。

您只需要配置电子邮件服务器一次。

### 要保存注解过的窗口

- 1 在**文件**菜单上，单击**保存**。此时出现 Windows 标准的**另存为**对话框。
- 2 输入文件的名称与格式，然后单击**确定**。

### 要打印注解的窗口

- 1 在**文件**菜单上，单击**打印**。此时出现 Windows 的标准“打印”对话框。
- 2 指定任何打印选项，然后单击**确定**。

### 要将注解过的窗口作为电子邮件附件发送

- 1 在**编辑**菜单上，单击**电子邮件配置**。此时出现**电子邮件配置**对话框。
- 2 输入要用于发送电子邮件的 **SMTP** 电子邮件服务器的主机名。如果不确定，请向管理员寻求帮助。单击**确定**。
- 3 在**文件**菜单上，单击**电子邮件**。此时出现**电子邮件**对话框。
- 4 输入发送人与接收人的地址并书写消息。注解的窗口的图像文件会自动添加为附件。
- 5 单击**发送**以发送电子邮件消息。

## AnnotateLayout() 函数

此时出现**注解布局**对话框，您可以在调用此脚本函数的地方对当前查看屏幕作出注解。只有在 Windows XP Tablet PC Edition 操作系统上，此函数才受到支持。

### 类别

系统

### 语法

AnnotateLayout()

### 附注

出现**注解布局**对话框时，捕获 WindowViewer 的屏幕图像。使用此对话框以：

- 使用笔以及工具栏、菜单项目设置给屏幕截图作出注解。
- 将图像与注解保存为 .gif 或 .jpeg 文件。
- 打印图像与注解（如果配置了打印机）。
- 将图像与注解当作电子邮件消息的附件来发送（如果配置了 SMTP）。

## 更改屏幕方向

如果 Tablet PC 按平板配置运行，并且 WindowViewer 配置成将应用程序分辨率动态更改为屏幕分辨率，则在横向模式中开发的 InTouch 应用程序会进行缩放以适合纵向模式。

如果 WindowViewer 未配置成动态更改应用程序分辨率，则横向应用程序不会进行缩放。这种情况下，某些 InTouch 窗口在平板电脑上可能会被截断。



缺省情况下，从一种配置切换到另一种时，屏幕分辨率也会切换。例如，如果按笔记本配置运行的 Tablet PC 切换到平板配置，则屏幕方向也会从横向 (1024 x 768) 模式切换到纵向 (768 x 1024) 模式。



## 附录 A

# 从 INTOUCH.ini 文件中自定义应用程序设置

第一次运行 InTouch 应用程序时，会在应用程序文件夹中创建 INTOUCH.ini 文件。创建 INTOUCH.ini 文件时，将值指定给确定单独 InTouch 应用程序运行特征的一组参数。

从 WindowMaker 或 WindowViewer 中继续配置应用程序时，会创建新的 INTOUCH.ini 参数或修改现有的参数。例如，从 WindowMaker 的历史记录属性对话框中配置记录功能时，记录参数会添加到 INTOUCH.ini 文件。

其它配置参数必须手工添加到 INTOUCH.ini 文件。

在自定义应用程序之后，可以将 INTOUCH.ini 文件复制到另一个应用程序的文件夹。这样，就可以在不重复所有自定义步骤的情况下为应用程序创建一致的运行特征。

## 自定义的 INTOUCH.ini 参数

下表列出一组参数，通过在 INTOUCH.ini 文件中手工输入这些参数，可以为 InTouch 应用程序提供额外的自定义属性。

INTOUCH.ini 参数	用途
16PenTrendDrawMode	确定“16 笔趋势”按平均模式还是最小 / 最大模式显示数据值。
CommentRetentive	确定是否保存运行时对“报警注释”域所作的更改。

INTOUCH.ini 参数	用途
ForceLogCurrentValue	确定是否将所记录的标记的当前值按 ForceLogging 参数设置的间隔写入“历史日志”文件。
ForceLogging	设置将标记值定期写入“历史日志”文件（不论其当前值如何）的间隔长度。
LoopTimeOut	设置 InTouch 脚本中 FOR-NEXT 循环处理的超时时段。
NoKeyboardResize	确定是否按 WindowViewer 屏幕的分辨率来调整数字键盘的大小。
OldRightMouseBehavior	确定 WindowMaker 中是否启用鼠标右键。
PrintScreenWait	设置从 WindowViewer 中打印屏幕之前的等待时段。
PrintWindowWait	设置从 WindowViewer 中打印 InTouch 窗口之前的等待时段。
RemoteTagsLogEvents	确定 InTouch 应用程序是否记录远程引用的标记报警与事件。
RemoteTagsNoIOEvents	确定 InTouch 应用程序是否记录远程引用的标记报警。
ScaleForResolution	确定 InTouch 应用程序窗口在更改到具有不同屏幕分辨率的节点时是否自动调整大小。

## 设置自定义的记录属性

您可以在 INTOUCH.ini 文件中添加一组参数，用于指定标记值如何保存到 InTouch 历史日志文件。指定给这些参数的值确定记录频率以及是否记录远程引用的标记的值。

### 设置记录频率

InTouch HMI 基于以下两个条件将日志项写入历史日志文件：

- 只要标记值的变化超出记录死区值一个工程单位值，则 InTouch HMI 立即写入一个日志项。
- InTouch HMI 按固定间隔写入记录的所有标记的当前值。缺省的固定间隔为 60 分钟。

您可以在 INTOUCH.ini 文件中添加两个参数来更改间隔。

- 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 之间的值。缺省值是 ForceLogging=60。

- ForceLogCurrentValue

ForceLogCurrentValue 强制 InTouch HMI 为记录的所有标记写入日志项，即使它们的当前值小于或等于记录死区范围时也是如此。缺省值是 ForceLogCurrentValue=0。

在下例中，当前标记值以 15 分钟为间隔或在标记值发生变化时写入“历史日志”文件中。

```
ForceLogging=15
ForceLogCurrentValue=1
```

### 记录远程引用的标记

缺省条件下，远程引用的标记不记录到历史日志文件。要记录远程引用的标记，必须启用历史记录功能，然后将

RemoteTagsLogEvents 参数添加到 INTOUCH.ini 文件。

```
RemoteTagsLogEvents=1
```

要将 I/O 标记排除在外不进行记录，请将

RemoteTagsNoIOEvents 参数添加到 INTOUCH.ini 文件。仅当 RemoteTagsLogEvents 参数设置为 1 时，才应用 RemoteTagsNoIOEvents 参数。

```
RemoteTagsNoIOEvents=1
```

## 禁用 WindowMaker 快捷菜单

缺省条件下，使用鼠标右键单击所选的对象时，WindowMaker 会显示一个快捷菜单。如果倾向于使用与较早版本的 InTouch HMI 相同的鼠标行为来开发应用程序，通过在 INTOUCH.ini 文件中将 oldrightmousebehavior 参数设置为 1，可以关闭 WindowMaker 的右击行为。

```
oldrightmousebehavior=1
```

## 设置自定义的 WindowViewer 属性

您可以添加一组 INTOUCH.ini 文件参数来设置 WindowViewer 的行为，以：

- 处理脚本循环。
- 针对不同屏幕分辨率来缩放 InTouch 窗口。
- 设置打印窗口或屏幕的等待时段。
- 将运行时更改记录到报警注释。

- 设置“16 笔趋势”的绘制模式。
- 调整数字小键盘的大小。
- 调整模拟与字符串用户输入链接的输入字段的大小。

### 添加脚本循环计时器

缺省条件下，InTouch 脚本中的 FOR-NEXT 循环必须在五秒钟内完成。如果 FOR-NEXT 循环处理在超时限制内没有完成，则 WindowViewer 自动停止脚本。这个超时限制可以防止由脚本错误而导致无限循环。

有时，可能需要编写 FOR-NEXT 循环代码处理超过五秒超时限制的脚本。通过将 LoopTimeout 参数添加到 INTOUCH.ini 文件，您可以更改超时限制的长度。

在本例中，循环处理持续的最长时间为 20 秒：

```
LoopTimeout=20
```

### 根据不同屏幕分辨率缩放 InTouch 窗口

将应用程序迁移到运行不同屏幕分辨率的其它节点时，通过在 INTOUCH.ini 文件中添加一个参数，可以保持 InTouch 窗口的当前分辨率。

ScaleForResolution 参数值确定运行 WindowViewer 的计算机上的显示分辨率发生更改之后，WindowMaker 是否自动缩放应用程序窗口。ScaleForResolution 参数不影响 WindowViewer 对话框的分辨率。ScaleForResolution 参数设置为 1 时启用分辨率转换。

```
ScaleForResolution=1
```

### 设置打印等待时段的长度

选择要打印的窗口或屏幕时，WindowViewer 将所选窗口或屏幕加载到内存。WindowViewer 随后等待 10 秒，供窗口或屏幕中显示的所有 DDE 变量进行更新。在等待时段结束之后，WindowViewer 将窗口或屏幕发送到打印机。

通过将 PrintWindowWait 或 PrintScreenWait 参数添加到 INTOUCH.ini 文件，可以更改 WindowViewer 打印等待时段。任一参数的等待时段均以毫秒为单位来表示。

```
PrintWindowWait=15000  
PrintScreenWait=20000
```

### 记录报警注释

操作员可以在确认报警时添加注释。要将运行时更改写入标记数据库中的报警注释域，请将下行代码添加到当前应用程序的 INTOUCH.ini 文件中。

```
CommentRetentive=1
```

## 设置 16 笔趋势的绘制模式

您可以根据 16PenTrendDrawMode 参数的值来选择“16 笔趋势”的线条绘制模式。

- 平均模式：16PenTrendDrawMode=0

由于“16 笔趋势”的时间范围与缓冲区大小，趋势上的每个像素都可以代表几秒钟的数据。每个间隔都可能包含多个具有不同值的样本。结果是，趋势的数据点可能显示为间隔内观察到的最大值与最小值之间的一根垂直线。

在绘制最小值到最大值的垂直线之后，趋势笔移到计算的间隔平均值。下一个间隔从平均值开始绘制线条，直到趋势上的下一个间隔结束。此时再次绘制最小值到最大值的垂直线，笔停在计算的间隔平均值上。此过程会为每个采样间隔重复。

如果 INTOUCH.ini 文件中没有指定 16PenTrendDrawMode，则“平均”为“16 笔趋势”的缺省绘制模式。

- 最小 / 最大模式：16PenTrendDrawMode=1

在“最小 / 最大”绘制模式中，趋势线通过直接连接每个数据采集间隔的端点来绘制。

## 调整数字小键盘的大小

您可以将一个参数添加到 INTOUCH.ini 文件，用于确定是否可以调整 InTouch 应用程序的数字小键盘的大小。在屏幕分辨率较高 (1280 x 1024) 的情况下放大数字小键盘，可以使得数字小键盘上出现的文本保持清晰。但是，应用程序可能对屏幕空间有所限制，因此对数字小键盘的大小也有实际的限制。

您可以将 NoKeyboardResize 参数添加到 INTOUCH.ini 文件中。缺省条件下不包含此参数。它的缺省值为：

```
NoKeyboardResize=0
```

缺省值允许数字小键盘根据屏幕分辨率调整大小。

可以指定给此参数的其它值有：

```
NoKeyboardResize=1
```

在这种情况下，数字小键盘不根据屏幕分辨率调整大小，它的大小保持不变。

## 调整模拟与字符串用户输入链接的输入字段的大小

您可以将 ResizableInputLink 参数添加到 INTOUCH.ini 文件，以使用鼠标来调整“模拟”或“字符串”用户输入链接的输入框大小。ResizableInputLink 参数必须设置为非零值。

第一次调整“输入”字段的大小之后，WindowViewer 将 ResizableInputLinkWidth 与 ResizableInputLinkHeight 参数添加到 INTOUCH.ini 文件。这些参数以像素为单位指定输入框的宽度与高度。

**示例：**

```
Resizable InputLink = 1  
Resizable InputLink Width=300  
Resizable InputLink Height=50
```

同样，您也可以通过编辑 INTOUCH.ini 文件来手工修改指定给这些参数的值。



# 索引

## 符号

\$AccessLevel 系统标记 124, 129, 133, 138  
 \$ApplicationChanged 系统标记 44  
 \$ApplicationVersion 系统标记 45  
 \$ChangePassword 系统标记 121, 125, 138  
 \$ConfigureUsers 系统标记 120, 126, 129, 138  
 \$InactivityTimeout 系统标记 111, 112, 138  
 \$InactivityWarning 系统标记 111, 112, 138  
 \$Language 系统标记 145, 146  
 \$LogicRunning 系统标记 116  
 \$Operator 系统标记 51, 129, 133, 137, 138  
 \$OperatorDomain 系统标记 137, 138  
 \$OperatorDomainEntered 系统标记 132, 138  
 \$OperatorEntered 系统标记 128, 131, 138  
 \$OperatorName 系统标记 136, 138  
 \$PasswordEntered 系统标记 128, 132, 138  
 \$VerifiedUserName 系统标记 137, 138

## 数字

16PenTrendDrawMode 参数 191

## A

AddPermission() 函数 119, 123, 138  
 AlarmGroup 关键字 72–74  
 AnnotateLayout() 函数 182, 184  
 ArchestrA  
     从“应用程序管理器”中启动 IDE 16  
     登录到存在安全机制的应用程序 128  
     设置安全性 123  
     通过 IDE 管理 InTouch 应用程序 14  
     主用户帐户 58  
 AttemptInvisibleLogon() 函数 130, 138  
 安全性  
     不活动超时功能 110–111  
     创建自定义的登录窗口 128  
     基于 InTouch 的身份验证 118  
     设置 InTouch 安全性身份验证与授权 120–121  
     设置基于 ArchestrA 的 123  
     设置基于操作系统的授权 122–123  
     身份验证 118  
     失效动画链接 133  
     使用基于 ArchestrA 的安全性登录或注销用户 128

- 使用基于 ArchestrA 的身份验证 119–120
- 使用基于 InTouch 的安全性登录或注销用户 127
- 使用基于操作系统的身份验证 119
- 使用脚本更改用户口令 124
- 使用脚本检索当前登录的用户的有关信息 136
- 使用脚本将用户自动登录到 InTouch 130
- 使用脚本确定用户组成员关系 135
- 使用脚本添加用户权限 123
- 使用脚本显示 InTouch “登录”对话框 129
- 使用脚本验证用户凭证 133
- 授权 118
- 锁定系统键 113–114
- 限制访问 InTouch 功能 133
- 在运行时更改口令 121–122
- 在运行时隐藏菜单项 115–117
- 综述 109

## B

标记

- \$InactivityTimeout 系统标记 111
- DBLoad 关键字 74–75
- 导出“标记名字典”的内容 64
- 关键字属性 75–81
- 记录远程引用的 189
- SuperTag 实例 96
- 使用 DBLoad 导入到“标记名字典” 98

标记名字典

- 创建 DBLoad 输入文件 67
- 设置输入文件格式 66
- 使用 DBDump 导出内容 64
- 使用 DBLoad 导入标记 97–98

不活动超时 110–112

## C

ChangePassword() 函数 124, 138

CommentRetentive 参数 190

参数

- 16PenTrendDrawMode 191
- CommentRetentive 190
- ForceLogCurrentValue 189
- ForceLogging 189

- LoopTimeout 190
- MultiScreen 178
- MultiScreenHeight 178
- MultiScreenWidth 178
- NoKeyboardResize 191
- oldrightmousebehavior 189
- PrintScreenWait 190
- PrintWindow 190
- ResizableInputLink 192
- ScaleForResolution 179, 190

从 WindowViewer 转换到 WindowMaker 166

从磁盘加载窗口 160

## D

DBDump

- 查看导出文件的内容 65
- 导出标记 64
- 介绍 64

DBLoad

- AlarmGroup 关键字 72–74
- 创建 SuperTag 实例 96
- 创建输入文件 67
- IOAccess 关键字 71, 71–72
- 介绍 64
- mode 关键字 68–70
- 输入文件 66
- 指定报警组 72–74
- 指定“访问名” 71–72

打开 / 关闭窗口 165

DRC, 请参阅“动态分辨率转换”

登录 127

调试脚本 161

动态分辨率转换 48

介绍 48

配置应用程序 48–49

对象

闪烁速度 161

对象闪烁速度 161

多监视器系统

部署应用程序 179

单显卡配置 172–173

多显卡配置 173–174

规划 175

介绍 171

- 配置屏幕分辨率转换 178
- 确定屏幕分辨率 175
- 确定显示应用程序的监视器数 175
- 确定应用程序窗口的位置 176
- 使用参数进行配置 177
- 选择显卡 175
- 在运行时验证支持 179

## E

EnableDisableKeys() 函数 113, 115, 138

## F

ForceLogCurrentValue 参数 189

ForceLogging 参数 189

访问名

- InTouchEvent 应用程序的限制 33
- 选择将占位符标记转换为远程引用 105
- 选择转换占位符标记 101
- 在 DBLoad 文件中指定 71
- 只提示激活项 60

分布式历史系统 37

服务

- 查看错误消息 59
- InTouch 组件注册表项 61
- 将 WindowViewer 配置成作为服务启动 57
- 描述 55
- 配置用户帐户 58
- 启动 WindowViewer 56
- 使用命令停止 58
- 使用“控制面板”停止 58
- 手工启动 57
- 疑难排解 59
- 用户帐户问题疑难排解 60

## G

Galaxy

- 定义 ArchestrA 安全性 123
- InTouchEvent 应用程序的限制 33, 166

GetAccountStatus() 函数 134, 139

GetNodeName() 函数 51

GroupVar 关键字 93

关闭 / 打开窗口 165

关键字

GroupVar 关键字 93

HistoryTrend 关键字 93

IndirectAnalog 关键字 94

IndirectDisc 关键字 94

IndirectMsg 关键字 95

IODisc 关键字 82

IOInt 关键字 85–87

IOMsg 关键字 92–93

IOReal 关键字 89–91

MemoryDisc 关键字 81

MemoryInt 关键字 82–84

MemoryMsg 关键字 91

MemoryReal 关键字 87–89

使用缺省值 96

TagID 关键字 93

## H

HistoryTrend 关键字 93

HTSetPenName() 函数 40

HTUpdateToCurrentTime() 函数 38

函数

AddPermission() 函数 119, 123

AnnotateLayout() 函数 182, 184

AttemptInvisibleLogon() 函数 130

ChangePassword() 124

EnableDisableKeys() 函数 113, 115

GetAccountStatus() 函数 134

GetNodeName() 函数 51

HTSetPenName() 函数 40

HTUpdateToCurrentTime() 函数 38

InvisibleVerifyCredentials() 函数 133

IOReinitialize() 函数 51

IOSetAccessName() 函数 34

IsAssignedRole() 函数 135

Logoff() 函数 130

LogonCurrentUser() 函数 129

PostLogonDialog() 函数 127, 129

QueryGroupMembership() 函数 136

ReloadWindowViewer() 函数 46

RestartWindowViewer() 函数 46

SwitchDisplayLanguage() 函数 145

TseGetClientId() 函数 51, 53

TseGetClientNodeName() 函数 53

TseQueryRunningOnClient() 函数 54  
TseQueryRunningOnConsole() 函数 53  
WWMultiMonitorNode() 函数 179

## I

IndirectAnalog 关键字 94  
IndirectDisc 关键字 94  
IndirectMsg 关键字 95  
INTOUCH.ini 文件 17  
    创建 17  
    复制到应用程序文件夹 17  
    自定义的参数 187–192  
InTouchEvent  
    创建应用程序 167–168  
    从 InTouchEvent 应用程序转换 169  
    介绍 159  
    WindowMaker 限制 167  
    许可证 170  
    转换为 InTouchEvent 应用程序 168–169  
InvisibleVerifyCredentials() 函数 133, 139  
IOAccess 关键字 71, 71–72  
IODisc 关键字 82  
IOInt 关键字 85–87  
IOMsg 关键字 92–93  
IOReal 关键字 89–91  
IOReinitialize() 函数 51  
IOSetAccessName() 函数 34  
IsAssignedRole() 函数 135, 139

## J

计时间隔 161  
基于服务器的架构 31  
基于客户端的架构 30  
架构  
    基于服务器 31  
    基于客户端 30  
    网络应用程序开发 32  
脚本  
    更改用户口令 124  
    管理用户帐户 123–126  
    检索当前登录的用户的有关信息 136  
    将用户自动登录到 InTouchEvent 130  
    确定用户组成员关系 135

设置循环超时限制 190  
添加用户权限 123  
显示 InTouchEvent “登录” 对话框。129  
验证用户凭证 133  
在运行时停止 116, 117  
接触对象时蜂鸣 161

## K

快速切换 160

## L

Logoff() 函数 130, 139  
LogonCurrentUser() 函数 129, 139  
LoopTimeout 参数 190  
历史供应器动态配置 40  
历史记录  
    记录远程引用的标记 189  
    设置记录频率 188  
历史记录功能  
    InTouchEvent 应用程序的限制 166  
联系技术支持 12

## M

MemoryDisc 关键字 81  
MemoryInt 关键字 82–84  
MemoryMsg 关键字 91  
MemoryReal 关键字 87–89  
Mode 关键字 68–70  
MultiScreen 参数 178  
MultiScreenHeight 参数 178  
MultiScreenWidth 参数 178  
命令  
    InTouchEvent 中不可用的 167  
    Net Start view 57  
    Net Stop view 58  
    RESET 96  
    通知客户端 32, 46  
    在 WindowViewer 中隐藏 116  
    终端服务客户端 50

## N

NAD, 请参阅 “网络应用程序开发”  
Net Start view 命令 57  
Net Stop view 命令 58

NoKeyboardResize 参数 191

## O

oldrightmousebehavior 参数 189

## P

PostLogonDialog() 函数 127, 129, 139

PrintScreenWait 参数 190

PrintWindow 参数 190

## Q

QueryGroupMembership() 函数 136, 139

## R

RDP 50

ReloadWindowViewer() 函数 46

RESET 命令 96

ResizableInputLink 参数 192

RestartWindowViewer() 函数 46

## S

ScaleForResolution 参数 179, 190

SwitchDisplayLanguage() 函数 145, 146

SuiteLink 的描述 34

闪烁速度 161

身份验证

使用基于 ArchestrA 的安全性 119–120

使用基于 InTouch 的安全性 118

使用基于操作系统的安全性 119

授权

设置 InTouch 安全性 120–121

设置基于操作系统的安全性 122–123

限制访问 InTouch 功能 133

数值时间质量 (VTQ) 34

## T

Tablet PC

打印窗口 183

给窗口添加注解 182

更改屏幕方向 185

剪切、复制及粘贴注解 183

将注解过的窗口作为电子邮件附件发送 184

介绍 181

删除注解 183

选择注解 183

TagID 关键字 93

TseGetClientId() 函数 51, 53

TseGetClientNodeName() 函数 53

TseQueryRunningOnClient() 函数 54

TseQueryRunningOnConsole() 函数 53

通知客户端命令 32, 46

## W

WindowMaker

编辑锁定 47

打开应用程序 19

禁止从 WindowViewer 切换 116

开发 InTouchView 应用程序时的限制 167

设置鼠标行为 189

WindowViewer

打开应用程序 19

调整数字小键盘的大小 191

复制应用程序 47

禁止切换到 WindowMaker 116

逻辑菜单 116

配置成作为 Windows 服务启动 56

设置打印等待时段 190

设置动态更改屏幕分辨率的选项 48

设置脚本循环超时间隔 190

设置“16 笔趋势”的绘制模式 191

在运行时隐藏菜单项 115–117

转换到 WindowMaker 时关闭 160

转换到 WindowMaker 时关闭所有打开的窗口 160

自定义 159

作为服务启动 56

作为服务运行 56

作为图标启动 160

Wonderware Historian

配置为历史供应器 40

WWMultiMonitorNode() 函数 179

网络架构

单机 30

基于服务器 31

支持的类型 30

网络应用程序开发

分发应用程序 39

- 更新期间对应用程序的更改 47–48
- 规划注意事项 33
- I/O 数据访问 33
- I/O 数据源 33
- I/O 数据源的本地地址 34
- 介绍 32
- 配置文件的 UNC 路径 36
- 配置自动更新 44
- 使用本地 I/O 地址 34–35
- 使用分布式历史系统 38
- 使用全局 I/O 地址 33–34
- 手工更新应用程序 44
- 文档惯例 11

## X

### 系统标记

- \$AccessLevel 系统标记 124, 129
- \$ApplicationChanged 系统标记 44
- \$ApplicationVersion 系统标记 45
- \$ChangePassword 系统标记 121, 125
- \$ConfigureUsers 系统标记 120, 126, 129
- \$InactivityTimeout 系统标记 111, 112
- \$InactivityWarning 系统标记 111, 112
- \$Language 系统标记 145, 146
- \$LogicRunning 系统标记 116
- \$Operator 系统标记 51, 129, 137
- \$OperatorDomainEntered 系统标记 132
- \$OperatorEntered 系统标记 128, 131
- \$OperatorName 系统标记 136
- \$PasswordEntered 系统标记 128, 132
- \$VerifiedUserName 系统标记 137

系统键锁定 113–114

## Y

### 应用程序

- 编辑锁定 47
- 查找 21
- 创建 17–19
- 从“应用程序管理器”中删除 20
- 导出报警注释进行翻译 151–152
- 导出文本进行脱机翻译 146–147
- 导入翻译的报警注释 155

- 登录或注销 127–128
- 发布到远程节点 22–26
- 根据不同的屏幕分辨率来缩放窗口 190
- 记录远程引用的标记 189
- 将报警注释导出到现有的报警注释文件 152–153
- 将文本导出到现有的字典文件 148
- 扩展综述 13–14
- 配置运行时语言切换 142–143
- 迁移到当前版本的 InTouch 26–27
- 确定多监视器系统的屏幕分辨率 175
- 缺省文件夹 16
- 任务管理综述 13
- 设置打印等待时段的长度 190
- 设置记录频率 188
- 设置鼠标行为 189
- 托管的 14
- 为 InTouchView 创建 167–168
- 修改 20
- 在 WindowMaker 中打开 19
- 在 WindowViewer 中打开 19
- 在迁移应用程序之前创建备份副本 27
- 终端服务环境中的安全性 51
- 重命名 20
- 应用程序发布者
  - 发布的文件的内容 23
  - 描述 22
  - 配置软件包详细信息 25
  - 软件包信息 22
- 应用程序管理器
  - 查找应用程序 21
  - 创建 InTouch 应用程序 17–19
  - 描述 14
  - 启动 15
  - 启动 ArchestrA IDE 16
  - 使用 WindowMaker 打开应用程序 19
  - 使用 WindowViewer 打开应用程序 19
  - 修改应用程序 20
  - 自定义窗口 21
- 语言切换
  - 导出报警注释进行翻译 151–152
  - 导出文本进行脱机翻译 146–147
  - 导入翻译的报警注释 155
  - 导入翻译的字典文件 150–151

- 翻译导出的字典文件 148–150
- 更改字体设置 143
- 将文本导出到现有的字典文件 148
- 配置 142–143
- 添加运行时切换功能 144–145
- 远程历史供应器
  - 创建列表 39–40
  - 介绍 39
- 远程桌面协议 50
- 运行时
  - 快速切换 160
  - 自定义 159

## Z

- 终端服务
  - 安全应用程序 51
  - 规划注意事项 50
  - 介绍 49, 50
  - 启动 “I/O 服务器” 51
  - 使用 “分布式报警系统” 50–51
  - 限制 52–53
  - 运行脚本 52
- 终端服务客户端命令 50
- 注销 128
- 转换到 WindowMaker 时关闭 WindowViewer 160
- 自定义运行时 InTouchView 159

