

# Руководство по управлению данными человеко-машинного интерфейса InTouch



## Глава 1. Обзор управления данными . . . . . 10

Работа с тэгами InTouch . . . . .	11
Типы тэгов InTouch . . . . .	12
Тэги памяти . . . . .	13
Тэги ввода/вывода . . . . .	14
Косвенные тэги . . . . .	15
Другие тэги . . . . .	16
Тэги исторических тенденций . . . . .	16
Тэги для идентификации других тэгов . . . . .	16
Супертэги. . . . .	17
Устаревшие тэги . . . . .	17
Свойства тэгов . . . . .	17
Свойства тэгов памяти . . . . .	19
Свойства тэгов ввода/вывода . . . . .	19
Удалённые ссылки на тэги . . . . .	21

## Глава 2. Управление тэгами с использованием словаря имён тэгов . . . . . 22

Планирование использования тэгов . . . . .	23
Создание новых тэгов . . . . .	24
Конфигурирование свойств тэгов. . . . .	25
Общие свойства тэгов. . . . .	25
Условные обозначения имён тэгов . . . . .	26
Автоматическое присвоение тэгам имён. . . . .	27
Комментарии тэгов . . . . .	27
Понимание свойств тэгов. . . . .	27
Диапазоны величин, единицы измерений и первоначальная величина . . . . .	28
Зоны нечувствительности тэгов . . . . .	28
Значение удержания тэга. . . . .	30
Присоединение ввода/вывода . . . . .	31
Регистрация тэгов . . . . .	31
Создание дискретных тэгов . . . . .	32
Создание целочисленных и вещественных тэгов . . . . .	34
Создание тэгов сообщений . . . . .	35
Создание тэгов ввода/вывода . . . . .	36
Изменение тэгов . . . . .	36
Удаление тэгов . . . . .	36

Распечатка списка тэгов и информации об использовании . . . . .	37
<b>Глава 3. Системные тэги . . . . .</b>	<b>38</b>
Системные тэги . . . . .	38
<b>Глава 4. Использование точечных полей для просмотра или изменения свойств тэгов . . . . .</b>	<b>44</b>
Изменение пределов значений тэга . . . . .	51
Изменение предела необработанного значения . . . . .	52
.MinRaw Dotfield . . . . .	52
Точечное поле .MaxRaw . . . . .	53
Просмотр необработанного значения тэга . . . . .	54
Точечное поле .RawValue . . . . .	54
Изменение пределов значений технических единиц . . . . .	55
Точечное поле .MaxEU . . . . .	55
Точечное поле .MinEU . . . . .	56
Замена технических единиц тэга . . . . .	57
Точечное поле .EngUnits . . . . .	57
Просмотр значения тэга в технических единицах . . . . .	58
Точечное поле .Value . . . . .	58
Точечное поле .OnMsg . . . . .	59
Точечное поле .OffMsg . . . . .	59
Просмотр или изменение комментария к тэгу . . . . .	60
Точечное поле .Comment . . . . .	60
<b>Глава 5. Доступ к данным с вводом/выводом . . . . .</b>	<b>62</b>
Поддерживаемые коммуникационные протоколы InTouch . . . . .	64
Динамический обмен данными . . . . .	64
Wonderware SuiteLink . . . . .	64
Troubleshooting SuiteLink Communication Problems . . . . .	65
Установка имён доступа . . . . .	65
Удаление имён доступа . . . . .	68
Доступ к данным ввода/вывода с тэгами ввода/вывода . . . . .	69
Конфигурирование свойств тэгов ввода/вывода . . . . .	69
Задание дискретного тэга ввода/вывода . . . . .	69
Задание целочисленных и вещественных тэгов ввода/вывода . . . . .	71
Задание тэга ввода/вывода сообщения . . . . .	73
Установка параметров доступа ввода/вывода . . . . .	73

Восстановление информации о тэгах ввода/вывода в реальном времени . . . . .	75
Функция IOGetNode() . . . . .	75
Функция GetNodeName() . . . . .	75
Функция IOGetApplication() . . . . .	76
IOGetTopic() Function . . . . .	77
Динамически изменяющиеся ссылки на тэги ввода/вывода во время выполнения . . . . .	77
Точечное поле .Reference . . . . .	78
IOSetItem() Function. . . . .	78
Функция IOSetAccessName() . . . . .	79
Преобразование тэгов в удалённые ссылки . . . . .	81
Доступ к данным ввода/вывода по удалённым ссылкам . . . . .	85
Перенаправление удалённых ссылок во время выполнения программы . . . . .	87
Функция IOSetRemoteReferences() . . . . .	87
Восстановление ссылок . . . . .	89
Оценка данных сервера приложений из InTouch. . . . .	90
Тэги InTouch в Сервере приложений . . . . .	91
Создание источников тэгов в пределах InTouch . . . . .	92
Ограничения браузера сервера применений. . . . .	93
Типы данных источников тэгов . . . . .	94
Специальные расширения в объектах Сервера приложений . . . . .	94
Отображение типа данных Сервера приложений на тип данных InTouch. . . . .	96
Поведение считывания/записи атрибутов сервера приложений . . . . .	98
Конфигурирование человеко-машинного интерфейса InTouch для использования Galaxy в качестве удалённого источника тэга. . . . .	99
Просмотр отметок времени и информации по качеству для тэга ввода/вывода . . . . .	103
Просмотр информации отметок времени для тэга ввода/вывода . . . . .	104
.TimeDate Dotfield . . . . .	104
Точечное поле .TimeDateString. . . . .	104
Поле .TimeDateTime. . . . .	105
Точечное поле .TimeDay . . . . .	105
Точечное поле .TimeHour . . . . .	106

Точечное поле .TimeMinute . . . . .	106
Точечное поле .TimeMonth . . . . .	107
Точечное поле .TimeMsec . . . . .	107
Точечное поле .TimeSecond . . . . .	108
Точечное поле .TimeTime . . . . .	108
Точечное поле .TimeTimeString . . . . .	109
Точечное поле .TimeYear . . . . .	109
Просмотр информации о качестве для тэга ввода/вывода . . . . .	110
Формат данных качества . . . . .	110
О точечных полях качества данных . . . . .	111
Точечное поле .Quality . . . . .	112
Точечное поле .QualityLimit . . . . .	113
Точечное поле .QualityLimitString . . . . .	113
Точечное поле .QualityStatus . . . . .	114
Точечное поле .QualityStatusString . . . . .	114
Точечное поле .QualitySubstatus . . . . .	115
Точечное поле .QualitySubstatusString . . . . .	116
Инициализация и сброс соединений ввода/вывода во время выполнения . . . . .	116
Повторная инициализация соединения ввода/вывода с командами	117
Повторная инициализация соединения ввода/вывода с использованием скриптов . . . . .	119
Функция IOReinitAccessName() . . . . .	119
Функция IOReinitialize() . . . . .	120
Функция IOStartUninitConversations() . . . . .	120
Использование функциональности обхода отказа с Именами доступа .	121
Конфигурирование обхода отказа . . . . .	121
Редактирования параметров Имени доступа пары обхода отказа	124
Удаление обхода отказа для Имени доступа . . . . .	124
Направление обхода отказа к резервному Имени доступа . . . . .	124
Выражение обхода отказа . . . . .	125
Функция IOForceFailover() . . . . .	125
Временное отключение функциональности обхода отказа . . . . .	126
Опция конфигурации Disable Failover . . . . .	126
Функция IODisableFailover() Script . . . . .	127
Функция IOGetAccessNameStatus() . . . . .	128
Функция IOGetActiveSourceName() . . . . .	129

Контроль состояния соединения ввода/вывода . . . . .	130
Использование имени темы IOStatus. . . . .	130
Использование имени темы IOStatus в программе Excel . . . . .	132
Контроль состояния связи Сервера ввода/вывода . . . . .	133
Доступ к данным тэга InTouch из других приложений . . . . .	134
<b>Глава 6. Определение косвенных признаков . . . . .</b>	<b>135</b>
Использование косвенных признаков со скриптами . . . . .	136
Использование косвенных признаков с локальными признаками . . . . .	137
Использование косвенных признаков с дистанционными обращениями. . . . .	138
<b>Глава 7. Определение многократно используемых структур признаков. . . . .</b>	<b>141</b>
Определение шаблона SuperTag . . . . .	143
Редактирование шаблонов SuperTag и признаков элемента . . . . .	146
Создание копий SuperTag. . . . .	147
Использование Словаря Tagname для создания копии SuperTag . . . . .	147
Использование Tagname Dictionary для размножения копий SuperTag . . . . .	149
Использование Tagname Dictionary для добавления признака к копии SuperTag . . . . .	150
Другие способы создания SuperTags . . . . .	151
Обращение к элементам SuperTag . . . . .	152
Заимствование SuperTags утилитой группового заимствования. . . . .	152
<b>Глава 8. Уменьшение использования признака . . . . .</b>	<b>153</b>
Определение использования признака . . . . .	154
Определение подсчета признаков. . . . .	155
Определение максимального числа дистанционных признаков на основе лицензирования . . . . .	156
Определение местоположения использования признаков . . . . .	157
Сохранение и Распечатка перечня с перекрестными ссылками признака . . . . .	162
Удаление неиспользуемых признаков. . . . .	165
<b>Глава 9. Запись значения признака . . . . .</b>	<b>166</b>
Конфигурирование исторической регистрации . . . . .	168
Конфигурирование признаков для исторической регистрации . . . . .	168

Для конфигурирования распределенного или исторического приложения. . . . .	170
Управление частотой исторической регистрации . . . . .	172
Запуск и остановка исторической регистрации во время работы . . . . .	173
Использование InTouch HMI с сервером IndustrialSQL . . . . .	174

## **Глава 10. Анализ тенденций в данных по признаку . . . . . 176**

Типы тенденций InTouch . . . . .	177
Осмысление исторических тенденций . . . . .	177
Осмысление тенденций в реальном времени . . . . .	178
Демонстрация сохраненных значений признака в исторической тенденции . . . . .	178
Использование объектов исторической тенденции . . . . .	179
Создание исторической тенденции . . . . .	179
Конфигурирование того, какие признаки показывать из исторической тенденции . . . . .	180
Конфигурирование продолжительности времени и интенсивности обновления исторической тенденции . . . . .	182
Конфигурирование опций дисплея исторической тенденции . . . . .	183
Изменение конфигурации тенденции во время выполнения . . . . .	184
Управление исторической тенденцией с использованием полей точек . . . . .	186
.DisplayMode Dotfield (поле точек режима дисплея) . . . . .	186
.MinRange Dotfield ( поле точек .MinRange). . . . .	187
.MaxRange Dotfield (поле точек MaxRange) . . . . .	188
.UpdateCount Dotfield . . . . .	189
.UpdateInProgress Dotfield . . . . .	190
.UpdateTrend Dotfield . . . . .	191
.ChartLength Dotfield . . . . .	192
.ChartStart Dotfield . . . . .	193
.Pen1-8 Dotfields . . . . .	194
.TagID Dotfield . . . . .	196
.ScooterLockLeft Dotfield . . . . .	197
.ScooterLockRight Dotfield. . . . .	198
.ScooterPosLeft Dotfield . . . . .	199
.ScooterPosRight Dotfield . . . . .	200
Использование мастера исторической тенденции . . . . .	201
Создание тенденции с помощью мастера исторической тенденции . . . . .	202

Конфигурирование того, какие признаки отображать на графике тенденции . . . . .	203
Конфигурирование временного интервала исторической тенденции	204
Конфигурирование опций дисплея . . . . .	205
Смена конфигурации во время выполнения . . . . .	206
Управление мастером исторической тенденции с использованием скриптов . . . . .	207
Обновление тенденции под текущее время . . . . .	207
Функция обновления под текущее время (UpdateToCurrentTime() ) . . . . .	207
Изменение конфигурации тенденции . . . . .	208
Функция HTSelectTag() . . . . .	208
Функция HTSetPenName() . . . . .	208
Восстановление информации о тенденции и исторических данных .	210
HTGetPenName() Функция . . . . .	210
Функция HTGetTimeAtScooter() . . . . .	211
Функция HTGetTimeStringAtScooter() . . . . .	212
Функция HTGetValue() . . . . .	213
Функция HTGetValueAtScooter() . . . . .	214
Функция HTGetValueAtZone() . . . . .	215
Панорамирование и изменение масштаба тенденции . . . . .	216
Функция HTScrollLeft() . . . . .	216
Функция HTScrollRight() . . . . .	217
Функция HTZoomIn() . . . . .	218
Функция HTZoomOut() . . . . .	219
Распечатка тенденции . . . . .	220
Функция PrintHT() . . . . .	220
Устранение неисправностей в тенденции . . . . .	220
Функция HTGetLastError() . . . . .	220
Отображение значений в реальном времени в тенденции . . . . .	221
Использование объектов тенденции в реальном времени . . . . .	222
Создание тенденции в реальном времени . . . . .	222
Конфигурирование того, какие признаки отображать на дисплее в тенденции реального времени . . . . .	223
Конфигурирование скорость обновления и временной интервал тенденции в реальном времени . . . . .	224

Конфигурирование опций индикации тенденции в реальном времени . . . . .	225
Печать тенденции во время выполнения . . . . .	227
Конфигурирование опций печати тенденций . . . . .	227
Отображение значений исторической тенденции из других узлов InTouch или сервера IndustrialSQL . . . . .	228
Конфигурирование перьев для отображения дистанционных данных тенденции . . . . .	229
Использования программы просмотра признака для присваивания перьев для дистанционных провайдеров истории. . . . .	229
Использование QuickScript для присваивания пера дистанционному провайдеру истории . . . . .	230

## **Глава 11. Доступ к значениям исторического признака из других приложений. . . . . 231**

Использование пунктов DDE для отображения исторических данных . . . . .	232
Доступ к данным регистрации с помощью DDE . . . . .	234
Ручное извлечение данных регистрации с помощью HistData. . . . .	234
Создайте имя доступа HistData Access Name . . . . .	234
Создайте признаки HistData . . . . .	235
Создайте окно HistData . . . . .	236
Выполните HistData. . . . .	238
Использование мастера HistData Wizard для извлечения данных регистрации . . . . .	238
Заимствование данных регистрации в файл CSV . . . . .	240
Функция WWpoke() . . . . .	241
Функция WWRequest() . . . . .	242
Устранение ошибок HistData. . . . .	243

## **Приложение А. Десятичные блоки IEEE . . . . . 245**

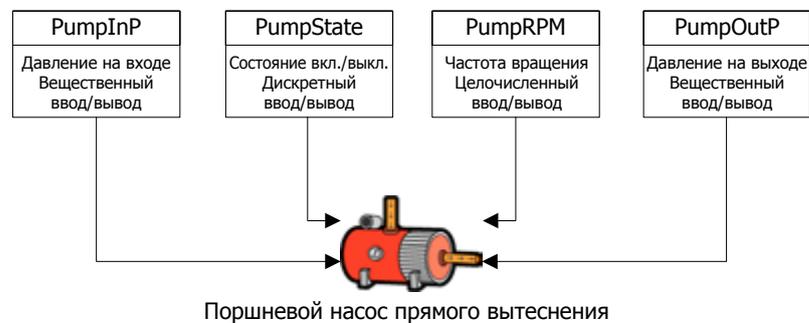
Отображение чисел с плавающей точкой в InTouch HMI . . . . .	246
--	-----

# Глава 1

## Обзор управления данными

Приложение человеко-машинного интерфейса (ЧМИ) Wonderware® InTouch® представляет собой графическое представление компонентов в производственной среде. Операторы производственной установки работают с данным графическим интерфейсом для контроля своих технологических процессов и управления ими.

Рисунок, приведённый ниже, иллюстрирует пример насоса, являющегося компонентом технологического процесса. Данный насос имеет свойства с соответствующими значениями. Давление, частота вращения и состояние являются свойствами насоса, значения которых контролируются посредством человеко-машинного интерфейса.

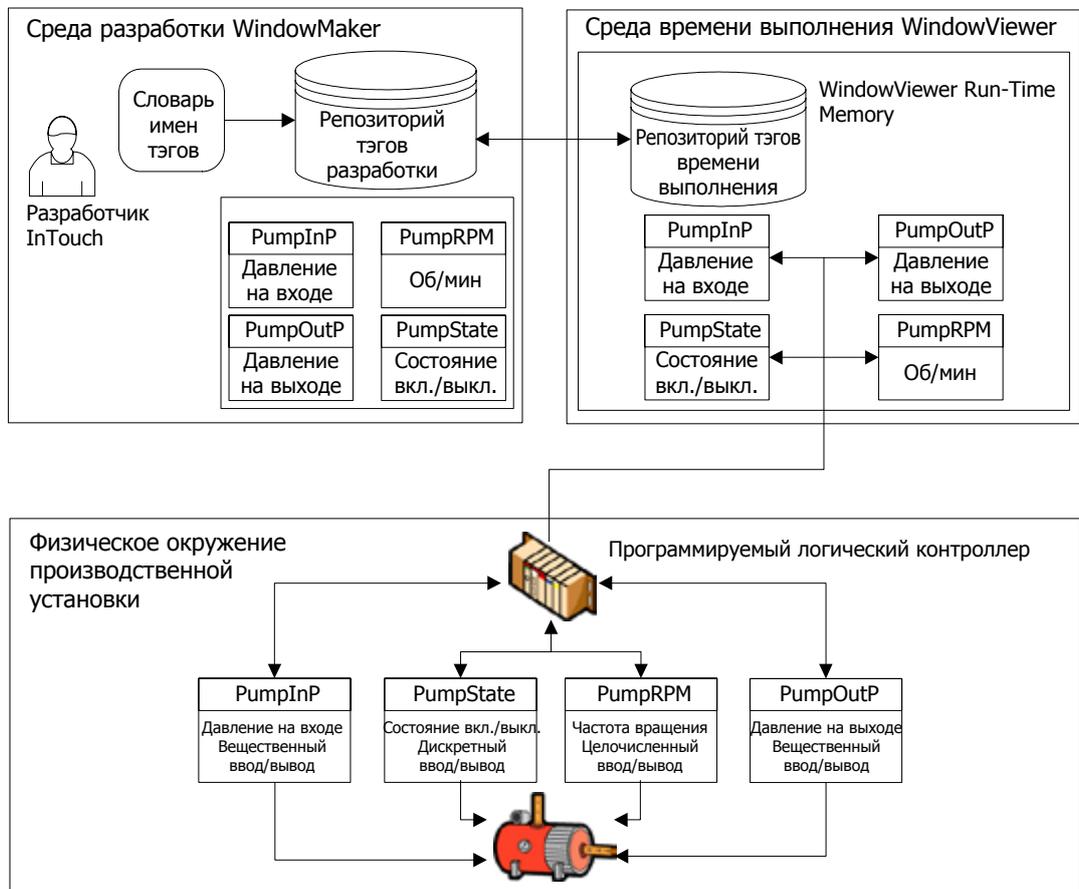


В тэге представлен элемент данных в приложении человеко-машинного интерфейса InTouch. Вы используете тэги для того, чтобы сделать конкретные свойства доступными как элементы данных из производственной среды. Показанный на приведённом выше рисунке тэг PumpState (Состояние насоса) указывает на то, включен ли насос или выключен. Вы создаёте тэги для тех компонентов в своей производственной среде, свойства которых вы желаете контролировать, или свойствами которых желаете управлять в вашем приложении InTouch.

Вы можете использовать различные типы тэгов для различных типов данных, собранных из производственного компонента. Например, тэг PumpState возвращает булеву величину Вкл./Выкл. для указания того, работает ли насос или остановлен. Вы присваиваете соответствующий тип тэга InTouch для того типа данных, который вы желаете сделать частью своего приложения.

## Работа с тэгами InTouch

Вы начинаете с создания приложения InTouch. Вы определяете тэги для приложения, используя Словарь имён тэгов (Tagename Dictionary), который является инструментом программы WindowMaker. На рисунке, приведённом ниже, показана разработка InTouch и среды реального времени.



Вы присваиваете имя и тип тэга с использованием Словаря имён тэгов. Для некоторых типов тэгов вы имеете другие варианты в Словаре имён тэгов для задания дополнительных свойств тэгов. Например, тэги типа ввода/вывода включают в себя дополнительные варианты для задания присоединения к удалённому источнику данных.

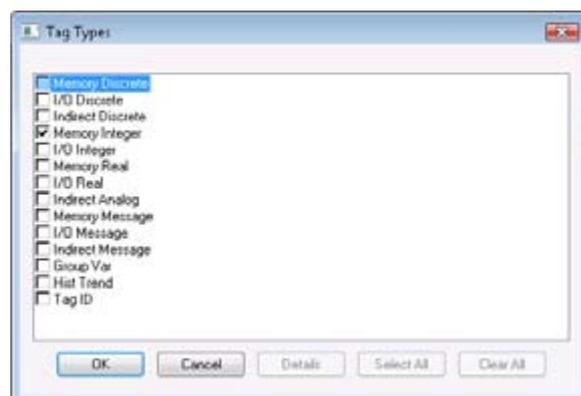
Приложения InTouch работают в среде WindowViewer. Когда среда WindowViewer запускает приложение, она считывает тэги из репозитория разработок и помещает их в память реального времени.

Приложение InTouch связывается с тэгами, размещёнными в памяти реального времени, используя анимационные ссылки или скрипты. Приложение InTouch отслеживает текущие значения и другую информацию о состоянии из свойств компонентов, присвоенных тэгам.

## Типы тэгов InTouch

Когда вы определяете тэг, вы присваиваете ему конкретный тип в соответствии с информацией о процессе, которая будет связана с тэгом. Например, если тэг показывает частоту вращения насоса, то вы присваиваете тэгу целочисленный тип.

В Словаре имён тэгов, вы используете Tag Types (Типы тэгов) диалоговое окно для присвоения типа тэга любому тэгу, который вы создали.



После того, как вы присваиваете тип тэга, в Словаре имён тэгов будут перечислены конкретные варианты для типа тэга, который вы выбрали.

## Тэги памяти

Тэги памяти определяют внутренние постоянные и переменные системы в рамках Приложения InTouch. Например, вы можете определить внутреннюю константу, как вещественное число 3,414. При моделировании технологических процессов, тэги памяти могут управлять действиями фона QuickScripts, действуя в качестве счётчика. На основании счёта, связанного с тэгом, QuickScript может включать различные анимационные эффекты. Тэги памяти могут также действовать в качестве расчётных величин, к которым имеется доступ со стороны других программ.

Выберите из четырёх типов тэгов памяти, на основании информации о процессе, связанной с тэгом.

- **Memory Discrete** (дискретные тэги памяти)

Дискретные тэги памяти связаны со свойствами компонентов процесса. Значения представляются двумя возможными булевыми состояниями, такими, как:

- 0 или 1
- Истина или ложь
- Включено или выключено
- Высокий или низкий

- **Memory Integer (Analog)** (целочисленные тэги памяти – аналоговые)

Вы можете присваивать целочисленным тэгам памяти 32-битные целочисленные значения в диапазоне от –2147483648 до 2147483647. За дополнительной информацией о преобразовании шестнадцатеричных чисел в 32-битные десятичные числа, обращайтесь к разделу «Десятичные единицы IEEE» в приложении А.

- **Memory Real (Analog)** (вещественные тэги памяти – аналоговые)

Вы можете присваивать вещественным тэгам памяти числа с плавающей десятичной точкой в диапазоне от  $-3,4 \times 10^{38}$  и  $3,4 \times 10^{38}$ . Все расчёты с плавающей точкой выполняются с 64-битной точностью, но результаты сохраняются в виде 32-битных целых чисел. За дополнительной информацией о преобразовании шестнадцатеричных чисел в 32-битные десятичные числа одинарной точности, обращайтесь к разделу «Десятичные единицы IEEE» в приложении А.

- **Memory Message** (тэги сообщений памяти)

Вы можете присваивать тэгам сообщений памяти текстовые строки длиной до 131 знаков, каждый из которых состоит из одного байта.

## Тэги ввода/вывода

Тэги ввода/вывода считывают данные приложения InTouch из внешнего источника или записывают их во внешний источник. Внешние данные включают в себя вход и выход с программируемых контроллеров, технологических компьютеров и сетевых узлов. К значениям данных тэгов ввода/вывода имеется удалённый доступ через системы «Microsoft Dynamic Data Exchange (DDE)» (Динамический обмен данными Microsoft) или «Wonderware SuiteLink™».

Когда значение тэга ввода/вывода изменяется в памяти реального времени, человеко-машинный интерфейс InTouch обновляет удалённое приложение. И наоборот, значения тэгов ввода/вывода в интерфейсе InTouch обновляются каждый раз, когда значения соответствующих элементов данных меняются в удалённом приложении.

Человеко-машинный интерфейс InTouch предоставляет четыре типа тэгов ввода/вывода на основании информации о процессе, связанной с тэгом. Эти четыре типа тэгов ввода/вывода аналогичны типам тэгов памяти.

- **I/O Discrete** (дискретные тэги ввода/вывода)

Дискретные тэги ввода/вывода связаны со свойствами компонентов процесса, значения которых представлены двумя возможными состояниями, такими как:

- 0 или 1
- Ложь или истина
- Включено или выключено
- Высокий или низкий

- **I/O Integer** (целочисленные тэги ввода/вывода – аналоговые)

Целочисленным тэгам ввода/вывода могут присваиваться 32-битные целочисленные значения со знаком в диапазоне от -2147483648 до 2147483647. За дополнительной информацией о преобразовании шестнадцатеричных чисел в 32-битные десятичные числа одинарной точности, обращайтесь к разделу «Десятичные единицы IEEE» в приложении А.

- **I/O Real** (тэги вещественного ввода/вывода – аналоговые)

Вещественным тэгам ввода/вывода могут присваиваться числа с плавающей десятичной точкой в диапазоне от  $-3.4 \times 10^{38}$  до  $3.4 \times 10^{38}$ . Все расчёты вещественных тэгов ввода/вывода с плавающей точкой выполняются с 64-битным разрешением, но результаты сохраняются как 32-битные числа. За дополнительной информацией о преобразовании шестнадцатеричных чисел в 32-битные десятичные числа одинарной точности, обращайтесь к разделу «Десятичные единицы IEEE» в приложении А.

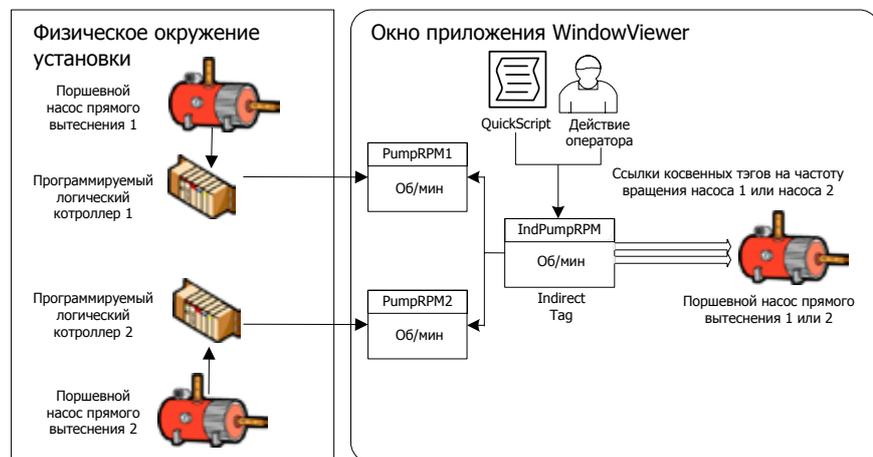
- I/O Message (тэги сообщений ввода/вывода)

Тэгам сообщений ввода/вывода могут присваиваться текстовые строки длиной до 131 знаков, каждый из которых состоит из одного байта.

## Косвенные тэги

Используя тэги косвенного типа, вы можете создавать единое окно в рамках приложения InTouch и заново присваивать его тэги множественным источникам.

На рисунке, приведённом ниже, показан пример окна приложения, в котором может показываться несколько насосов. Вместо создания отдельных окон для каждого насоса, вы можете использовать косвенные тэги в одном окне для демонстрации значений тэгов различных источников, связанных с отдельными насосами.



Действие QuickScript или оператора направляет косвенный тэг на тэги источников. Например, нижеследующие объявления скриптов присваивают два тэга PumpRPM (частота вращения насоса) косвенному аналоговому тэгу, называемому IndPumpRPM (частота вращения отдельного насоса) на основании значения тэга PumpNo (номер насоса).

```

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

```

Когда вы приравниваете косвенный тэг тэгу другого источника, косвенный тэг действует так, как если он является тэгом источника. Значения, связанные с оригинальным источником и косвенными тэгами-дубликатами синхронизируются между собой. Если значение тэга источника изменяется, то косвенный тэг отражает это изменение. Если изменяется значение косвенного тэга, то тэг источника меняется соответствующим образом.

Вы можете использовать дискретный и аналоговый тип, а также тип сообщения в качестве типов косвенных тэгов. Эти три типа косвенных тэгов совместимы с аналогичными типами тэгов памяти и ввода/вывода.

За дополнительной информацией о косвенных тэгов, обращайтесь к разделу «Определение Косвенных тэгов» в главе 7.

## Другие тэги

Вы можете использовать другие типы тэгов InTouch, разработанные для конкретных ограниченных целей. Вы можете использовать эти тэги для создания динамических дисплеев сигнализации и исторических тенденций, а также изменять тэги, присвоенные пишущим элементам (перьям) исторических тенденций.

### Тэги исторических тенденций

Тэги исторических тенденций могут использоваться для ссылки на графики исторических тенденций. Все поля точек, связанные с историческими тенденциями, могут применяться к тэгам исторических тенденций.

За дополнительной информацией об определении и использовании тэгов исторических тенденций, обращайтесь к разделу «Анализ тенденций данных тэгов» в главе 10.

### Тэги для идентификации других тэгов

Тэги для идентификации других тэгов восстанавливают информацию из тэгов, значения которых изображены на графике исторических тенденций InTouch. Обычно вы используете тэг для идентификации другого тэга для того, чтобы показать имя тэга, присвоенное конкретному перу тенденции или изменить тэг, присвоенный перу тенденции.

Вы можете включить заявление в QuickScript для присвоения нового тега перу в любой исторической тенденции с использованием тэга для идентификации другого тэга. Например, нижеследующее заявление QuickScript изменяет тэг, связанный с пером исторической тенденции:

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

При работе данного QuickScript Перо 1 исторической тенденции начинает анализ тенденций исторически сохранённых данных для тэга MyLoggedTag.

За дополнительной информацией об определении и использовании тэгов исторических тенденций, обращайтесь к разделу «Использование мастера исторических тенденций» в главе 10.

## Супертэги

Супертэг представляет собой шаблон, который содержит набор родственных тэгов. Например, вы можете создавать шаблон SuperTag, содержащий набор тэгов, присвоенных всем свойствам насоса.

Используйте супертэги SuperTags, когда вы имеете идентичное оборудование в вашем производственном процессе. Вместо создания набора тэгов для каждого компонента оборудования, присвойте пример шаблона SuperTag каждому из идентичных компонентов процесса.

За дополнительной информацией о супертэгах, обращайтесь к разделу «Определение многократно используемых тэговых структур» в главе 7.

## Устаревшие тэги

Тэг Group Var (групповая переменная) создаёт окна динамической сигнализации или журналы для тэгов, присвоенных группе предупредительных сигналов. Тэги Group Var включаются только для обеспечения обратной совместимости с приложениями, разработанными с использованием InTouch версии 7.11 и раньше. Не используйте тэги Group Var с приложениями, разработанными в InTouch версии 7.11 или позже.

## Свойства тэгов

Каждый тип тэга InTouch имеет набор свойств, которые описывают характеристики данных, связанных с тэгом. Четырьмя основными типами данных, связанными с тэгами InTouch, являются:

- дискретные двоичные значения
- целые числа
- вещественные числа
- текстовые сообщения

Всем свойствам тэгов присваиваются первоначальные значения, когда вы создаёте тэг с использованием Словаря имён тэгов. На приведённом ниже рисунке показаны свойства целочисленного тэга ввода/вывода, которые могут быть установлены из Словаря имён тэгов.

The screenshot shows the 'Tagname Dictionary' dialog box with the following configuration:

- Tab: Main
- Buttons: New, Restore, Delete, Save, <<, Select, >>, Cancel, Close
- Tagname: PumpRPM
- Type: I/O Integer
- Group: \$System
- Read/Write: Read only (unchecked), Read Write (checked)
- Comment: AccessLevel
- Log Data (unchecked), Log Events (unchecked), Retentive Value (unchecked), Retentive Parameters (unchecked)
- Initial Value: 0
- Deadband: 0
- Eng Units: (empty)
- Log Deadband: 0
- Conversion: Linear (checked), Square Root (unchecked)
- Access Name: Galaxy
- Item: RPM
- Use Tagname as Item Name (checked)
- ACK Model: Condition (checked), Event Oriented (unchecked), Expanded Summary (unchecked)
- Alarm Comment: (empty)
- Alarm Value Priority Alarm Inhibitor Value Deadband table:

Alarm Value	Priority	Alarm Inhibitor	Value	Deadband
<input type="checkbox"/> LdLo	0	1		
<input type="checkbox"/> High	0	1		0
<input type="checkbox"/> Low	0	1		
<input type="checkbox"/> HHl	0	1		
- % Deviation Target Priority Alarm Inhibitor Deviation Deadband % table:

% Deviation	Target	Priority	Alarm Inhibitor	Deviation Deadband %
<input type="checkbox"/> Minor Deviation	0	1		
<input type="checkbox"/> Major Deviation	0	1		0
- Rate of Change: 0 % per: Sec (checked), Min (unchecked), Hr (unchecked) Priority: 1 Alarm Inhibitor: (empty)

После установки первоначальных значений свойств тэгов из Словаря имён тэгов, вы можете динамически изменять большую часть свойств тэгов, используя точечные поля. Точечное поле идентифицирует свойство тэга, которое может контролироваться или изменяться скриптом при работе приложения InTouch. Вы добавляете точечное поле к имени тэга в скрипте.

За дополнительной информацией об использовании точечных полей для динамического изменения свойств тэгов, обращайтесь к разделу «Использование точечных полей тэгов для обзора или изменения свойств тэгов» в главе 4.

## Свойства тэгов памяти

В нижеприведённой таблице перечислены свойства четырёх типов тэгов памяти. Каждое свойство может быть выбрано или изменено как опция Словаря имён тэгов. За дополнительной информацией о задании свойств тэгов памяти обращайтесь к разделу «Создание новых тэгов» в главе 2.

Свойства тэгов	Дискретные	Целочисленные	Вещественные	Сообщение
Группа сигнализации	•	•	•	•
Данные из журнала	•	•	•	
События из журнала	•	•	•	•
Удерживающее значение	•	•	•	•
Удерживающие параметры		•	•	
Первоначальное значение	•	•	•	•
Сообщение On (включение)	•			
Сообщение Off (выключение)	•			
Технические единицы		•	•	
Минимальное значение		•	•	
Максимальное значение		•	•	
Зона нечувствительности		•	•	
Зона нечувствительности из журнала		•	•	
Максимальная длина				•
Комментарии сигнализации	•	•	•	•

## Свойства тэгов ввода/вывода

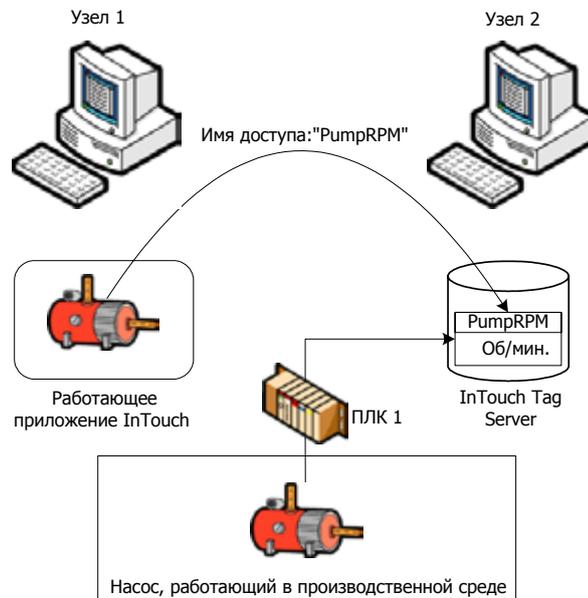
Как и в случае тэгов памяти, свойства тэгов ввода/вывода могут выбираться как варианты в Словаре имён тэгов. За дополнительной информацией о свойствах тэгов памяти обращайтесь к разделу «Создание новых тэгов» в главе 2.

Атрибут тэга	Дискретные	Целочисленные	Вещественные	Сообщение
Группа сигнализации	•	•	•	•
Только чтение		•	•	
Только запись		•	•	•
Данные из журнала	•	•	•	
События из журнала	•	•	•	•
Удерживающее значение	•	•	•	•
Удерживающие параметры		•	•	
Первоначальное значение	•	•	•	•
Максимальная длина				•

Атрибут тэга	Дискретные	Целочисленные	Вещественные	Сообщение
Комментарий к аварийному сигналу				•
Сообщение On (включение)	•			
Сообщение Off (выключение)	•			
Имя доступа	•	•	•	•
Элемент	•	•	•	•
Имя тэга как имя элемента	•	•	•	•
Технические единицы		•	•	
Максимальное значение в технических единицах		•	•	
Минимальное значение в технических единицах		•	•	
Минимальная необработанная величина		•	•	
Минимальная необработанная величина		•	•	
Зона нечувствительности		•	•	
Зона нечувствительности из журнала		•	•	
Линейное преобразование		•	•	
Преобразование типа извлечения квадратного корня		•	•	

## Удалённые ссылки на тэги

Вы можете создавать распределённые приложения InTouch с тэговым сервером, работающим на отдельном узле из работающего на узле приложения InTouch. На приведённом ниже рисунке приведён пример приложения InTouch, которое делает удалённую ссылку на тэг PumpRPM (частота вращения насоса) с тэгового сервера, работающего на другом узле



Вы создаёте приложение InTouch для ссылки на тэги, расположенные на удалённом узле двумя методами:

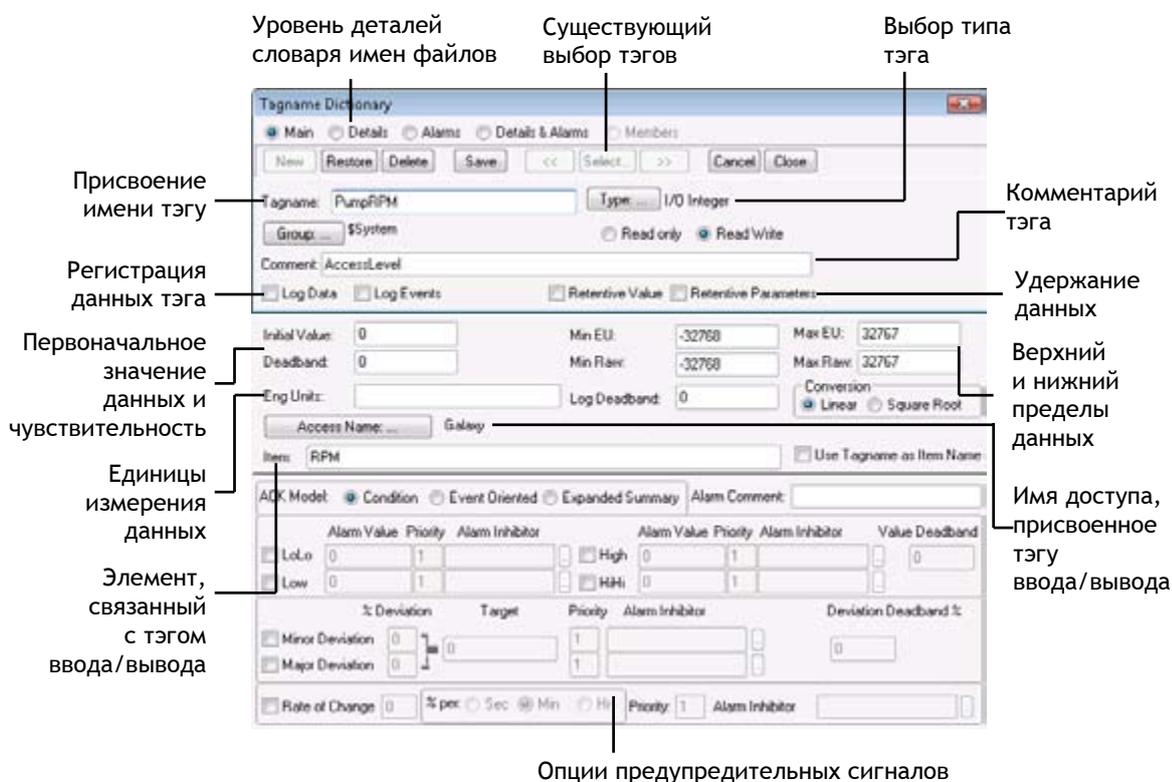
- Свяжите тэги ввода/вывода с Именем доступа, которое идентифицирует удалённый сервер как источник тэга. За дополнительной информацией об определении имени доступа для тэга ввода/вывода обращайтесь к разделу «Установка имён доступа» в главе 5.
- Используйте удалённую ссылку непосредственно по отношению к тэгу. Например, PLC1:PumpRPM.

За дополнительной информацией о конфигурировании и использовании удалённых тэговых ссылок обращайтесь к разделу «Получение доступа к данным ввода/вывода» в главе 5.

# Глава 2

## Управление тэгами с использованием словаря имён тэгов

Используя Словарь имён тэгов, вы создаёте тэги для приложения InTouch. На рисунке, приведённом внизу, показано диалоговое окно Tagname Dictionary (Словарь имён тэгов) со всеми опциями для определения свойств тэга ввода/вывода.



## Планирование использования тэгов

Вы можете сократить время, затрачиваемое на разработки, определяя ключевые требования тэгов приложения в фазе предварительного планирования. Тщательное планирование сокращает время, которое вам требуется для создания приложения InTouch.

Перед тем, как создавать тэги:

- Определите все физические компоненты процесса, которые требуется представить в приложении InTouch.

Создайте список атрибутов компонентов, которые требуется представить как источники данных в приложении.

- Определите тип данных, связанный с атрибутом каждого компонента.

Присвойте каждому тэгу тип тэга на основании данных, связанных с атрибутом компонента. За дополнительной информацией о присвоении типа данных тэгу, обращайтесь к разделу «Создание новых тэгов» в главе 2.

- Определите характеристики данных, которые требуется ввести в ваше приложение InTouch.

Откройте доступ к следующим характеристикам данных для каждого тэга:

- Ожидаемый диапазон значений данных;
- Единицы измерения, приписываемые значениям данных;
- Первоначальное значение данных;
- Значение зоны нечувствительности для установки порога, когда значение тэга признаётся как изменённое;
- Сообщения, которые должны показываться, когда значение тэга изменяет состояние.

За дополнительной информацией об определении характеристик данных тэга, обращайтесь к разделу «Понимание свойств тэгов» в главе 2.

- Разработайте систему условных обозначений и стандарт присвоения имён тэгам.

Обычно сложные применения требуют большого количества тэгов. Разработайте стандартизованную систему условных обозначений, которая предложила бы организацию тэгов в рамках приложения. За дополнительной информацией об условных обозначениях, используемых для присвоения имён тэгам обращайтесь к разделу «Условные обозначения имён тэгов» в главе 2.

- Определите, какую информацию о процессе требуется сохранить.

Избранные данные сохраняются в журнале регистрации. Вы можете использовать зарегистрированные в журнале данные для создания исторических тенденций, которые

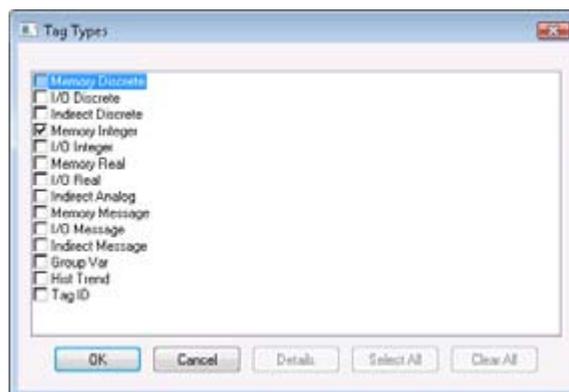
демонстрируют изменения значений тэгов с течением времени. За дополнительной информацией об установке регистрации тэгов обращайтесь к разделу «Регистрация тэгов» в главе 2.

## Создание новых тэгов

Вы создаёте тэги с использованием Словаря имён тэгов (**Tagname Dictionary**) программы WindowMaker. Перед тем, как приступить к созданию тэга, проанализируйте процесс вашей установки для определения тех тэгов, которые вам требуется создать в вашем приложении InTouch.

### Для создания нового тэга

- 1 Откройте приложение InTouch в программе WindowMaker.
- 2 В меню **Special** (Специальные) щёлкните на опции **Tagname Dictionary** (Словарь имён файлов). Когда вы открываете Словарь имён тэгов первый раз, в диалоговом окне **Tagname** (имя тэга) появляется определение для тэга системы `$AccessLevel`. После сохранения нового тэга, **Tagname Dictionary** показывает самое последнее сохранённое определение тэга.
- 3 Выполните следующие операции:
  - a Щёлкните на опции **New** (Новый). Диалоговое окно **Tagname** (Имя тэга) очищается.
  - b Введите имя для нового тэга. За дополнительной информацией о присвоении тэгам имён обращайтесь к разделу «Условные обозначения имён тэгов» в главе 2.
  - c По желанию введите комментарий о новом тэге в диалоговом окне **Comment** (Комментарий).
- 4 Щёлкните на опции **Type** (Тип). Появляется диалоговое окно **Tag Types** (Типы тэгов) со списком поддерживаемых типов тэгов InTouch.

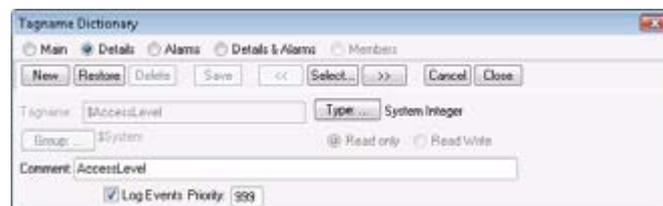


- 5 Выберите тэга из списка и щёлкните на опции **OK**. Снова появляется Словарь имён тэгов, в котором показывается тип тэга, который вы выбрали.
- 6 При необходимости щёлкните на опции **Details** (Детали) для того, чтобы увидеть дополнительные опции Словаря имён тэгов для выбранного типа тэга.

- 7 Задайте дальнейшие опции тэгов в диалоговом окне **Tagname Dictionary** (Словарь имён тэгов).
- 8 За дополнительной информацией о задании свойств тэгов обращайтесь к разделу «Конфигурирование свойств тэгов» в главе 2.
- 9 Щёлкните на опции **Save** (Сохранить). Щёлкните на опции **Close** (Закреть) для закрытия диалогового окна **Tagname Dictionary**.

## Конфигурирование свойств тэгов

Используя диалоговое окно Словарь имён тэгов, вы можете задать общие свойства, которые составляют часть определения каждого тэга. Вы должны присвоить имя каждому тэгу. Вы можете добавить комментарий по желанию. Как имя тэга, так и комментарий являются общими свойствами всех тэгов.



Каждый тип тэга InTouch имеет уникальные свойства данных. После того, как вы выберете тип тэга, диалоговое окно Tagname Dictionary (Словарь имён файлов) расширяется таким образом, чтобы был показан набор опций на основании выбранного типа тэга.

## Общие свойства тэгов

Вы должны присвоить уникальное имя каждому тэгу. Дополнительный комментарий должен составлять часть определения тэга. Хорошей практикой является снабжение каждого тэга, который вы определяете, соответствующими комментариями.

Все тэги принадлежат к группе сигнализации, что является ещё одним общим свойством тэгов. По умолчанию, все тэги принадлежат к группе сигнализации \$System. За дополнительной информацией о присвоении тэгов другим группам сигнализации обращайтесь к Главе 2 «Конфигурирование предупредительных сигналов» в «Руководстве по аварийным сигналам и событиям человеко-машинного интерфейса InTouch®».

## Условные обозначения имён тэгов

Присваивая имена вашим тэгам, используйте последовательную систему условных обозначений имён, если вам требуется большое количество тэгов с аналогичными свойствами.

Следуйте этой системе условных обозначений при присвоении имён для имён тэгов InTouch:

- Используйте в имени тэга 32 знака или меньше.
- Используйте букву или цифру (A-Z, a-z, 0-9) в качестве первого знака имени тэга.

Наилучшей практикой является использование для имён тэгов только букв и цифр.

- Используйте в имени тэга, как минимум, один знак, являющийся буквой.
- По желанию вы можете использовать следующие специальные знаки:

- Дефис
- ! Восклицательный знак
- # Знак номера
- \$ Знак доллара
- % Процент
- & Амперсанд
- ? Вопросительный знак
- @ Знак «при»
- \_ Знак подчёркивания

По возможности избегайте использования специальных знаков в именах тэгов, если в этом нет абсолютной необходимости в соответствии с требованиями вашего применения.

- Избегайте использования дефиса (-) в имени тэга.

Дефис является действующим знаком для имени тэга InTouch. Однако InTouch расценивает дефис как оператор отрицания или вычитания в логических или арифметических выражениях. Например, выражение  $A=B-C$  может быть интерпретировано как  $A=B$  минус  $C$  или присвоение имени тэга  $B-C$  тэгу  $A$ .

- Не используйте пробелов в именах тэгов.
- Не используйте в имени тэга число, которое может быть интерпретировано как экспоненциальное.

Например, вы не можете присвоить тэгу имя  $125E4$ , поскольку оно может быть интерпретировано как основание системы счисления с экспонентой, возведённой в четвёртую степень.

- Не используйте в имени тэга число, которое может быть интерпретировано как шестнадцатеричное.

Например, вы не можете присвоить тэгу имя  $0x123B$ , поскольку оно может быть интерпретировано, как шестнадцатеричное число.

## Автоматическое присвоение тэгам имён

Когда вы присваиваете имена вашим тэгам в Словаре имён тэгов, человеко-машинный интерфейс InTouch прослеживает условные обозначения, которые вы используете при присвоении имён. Если вы даёте своим тэгам имена Pump01, Pump02, то человеко-машинный интерфейс InTouch предлагает следующее имя тэга как Pump03. Вы можете принять или отклонить это имя. Такая помощь в присвоении имён называется индексированием.

Индексирование основывается на последнем порядковом номере в имени тэга. Например, если ваше имя тэга имеет вид PumpInP04LotB99A, человеко-машинный интерфейс InTouch предлагает имя следующего тэга как PumpInP04LotB100A, а не как PumpInP05LotB99A.

## Комментарии тэгов

При создании тэга вы можете ввести дополнительный комментарий до 50 знаков в диалоговое окно Comment (Комментарий) в Словаре имён тэгов.

Когда вы первый раз получаете доступ к Словарю имён тэгов, в диалоговом окне Comment (Комментарий) появляется комментарий, принимаемый по умолчанию для тэга системы \$AccessLevel. Удалите этот комментарий для того, чтобы предотвратить его привязку к любым тэгам, которые вы создаёте.

## Понимание свойств тэгов

После задания общих свойств тэгов вы должны определить другие свойства, которые являются специфическими по отношению к типу тэга, который вы создаёте. В приведённой ниже таблице показываются основные свойства для тэгов памяти по типам тэгов.

Тип тэга	Уникальные свойства
Дискретный	Первоначальное значение, Сообщение On (включение), Сообщение Off (выключение)
Целочисленный	Первоначальное значение, Минимальное значение, Зона нечувствительности, Технические единицы, Максимальное значение, Зона нечувствительности из журнала
Вещественный	Первоначальное значение, Минимальное значение, Зона нечувствительности, Технические единицы, Максимальное значение, Зона нечувствительности из журнала
Сообщение	Максимальная длина, Первоначальное значение, Комментарий к предупредительным сигналам

Тэги ввода/вывода имеют дополнительные свойства для установления сетевых связей и преобразования необработанных данных от сетевых устройств к нормированным значениям, используемым приложением InTouch. За дополнительной информацией об определении тэгов ввода/вывода обращайтесь к разделу «Конфигурирование свойств тэгов ввода/вывода» в главе 5.

### Диапазоны величин, единицы измерений и первоначальная величина

Дискретным, целочисленным и вещественным тэгам присваивается первоначальное значение, когда приложение InTouch запускается в программе WindowViewer. В случае дискретного тэга первоначальное значение является одним из возможных двоичных состояний. Для целочисленных и вещественных тэгов первоначальное значение представляет собой число, связанное с тэгом при запуске приложения.

Вы можете задать первоначальное значение тэга как последнее значение тэга при остановке работы приложения в программе WindowViewer. При выборе опции Retentive Value (удерживающее значение) из Словаря имён тэгов, тэгу присваивается последнее активное значение в качестве первоначального значения, когда приложение запускается снова.

Целочисленные и вещественные тэги включают в себя свойства, которые устанавливают нижние и верхние границы диапазона возможных числовых величин, присваиваемых тэгу. Как целочисленные, так и вещественные тэги включают в себя свойства Min Value (Минимальное значение) и Max Value (Максимальное значение), которые определяют нижний и верхний пределы диапазона.

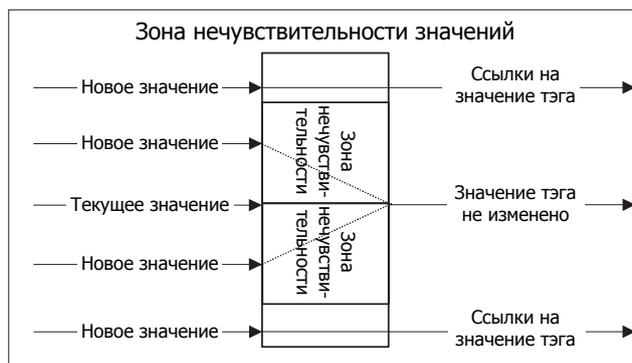
Целочисленные и вещественные тэги также включают в себя свойство Eng Units (Технические единицы) для присвоения метки технических единиц, которая описывает единицу измерения для значения тэга. Например, Вы можете присваивать PSI (фунты на квадратный дюйм) в качестве свойства Eng Units для целочисленного тэга, связанного с давлением насоса.

### Зоны нечувствительности тэгов

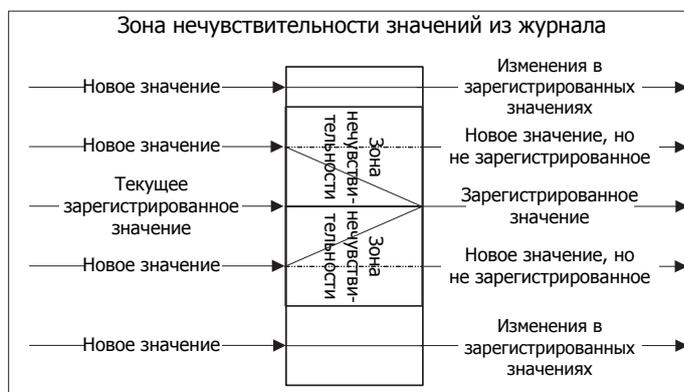
Зона нечувствительности представляет собой установку чувствительности для значений тэгов. Зона нечувствительности обычно связывается с тэгами ввода/вывода, значения которых меняются постоянно. Зона нечувствительности отфильтровывает мелкие мгновенные изменения в значении тэга для уменьшения объёма обработки данных InTouch.

Словарь имён тэгов включает в себя два свойства зоны нечувствительности для тэгов, связанных с целочисленными и вещественными данными.

- **Зона нечувствительности значений:** Свойство зоны нечувствительности значений устанавливает порог, который должен быть превзойдён до того, как программа WindowViewer освежит значение тэга в памяти реального времени. На рисунке, приведённом ниже, показан диапазон абсолютной зоны нечувствительности вокруг текущего значения тэга.



- **Зона нечувствительности значений из журнала:** Зона нечувствительности из журнала устанавливает порог, который должен быть превзойдён до того, как значение тэга будет записано в журнале регистрации. На рисунке, приведённом ниже, показана зона нечувствительности вокруг текущего значения тэга из журнала.



Только величины, лежащие за пределами зоны нечувствительности, записываются в журнал регистрации. Мелкие изменения значений в пределах диапазона зоны нечувствительности игнорируются и не регистрируются в журнале.

**Для установки тэга зоны нечувствительности:**

- 1 Откройте диалоговое окно **Tagname Dictionary** (Словарь имён тэгов).
- 2 Щёлкните на опции **Select** (Выбрать). Появляется диалоговое окно **Select Tag** (Выбрать тэг). Перечисляются тэги, определённые в текущий момент для приложения.
- 3 Выберите из списка целочисленный или вещественный тип тэга.
- 4 Щёлкните на **ОК**. Диалоговое окно части деталей Словаря имён тэгов показывает дополнительные опции, когда вы выбираете вещественный или целочисленный тэг.

Initial Value:	<input type="text" value="0"/>	Min Value:	<input type="text" value="0"/>	Deadband:	<input type="text" value="15"/>
Eng Units:	<input type="text" value="RPM"/>	Max Value:	<input type="text" value="2500"/>	Log Deadband:	<input type="text" value="25"/>

- 5 Установите значение зоны нечувствительности, вводя целое или вещественное число в поле **Deadband** (зона нечувствительности) на основании **выбранного** типа тэга.  
Зона нечувствительности значений устанавливает абсолютный пороговый уровень в технических единицах.
- 6 Устанавливайте зону нечувствительности из журнала, вводя целое или вещественное число в поле **Log Deadband** (зона нечувствительности из журнала) на основании выбранного типа тэга.  
Как и зона нечувствительности значений, зона нечувствительности из журнала устанавливает абсолютный порог в технических единицах.
- 7 Щёлкните на опции **Save** (Сохранить) для сохранения изменений вашей зоны нечувствительности.
- 8 Щёлкните на опции **Close** (Заккрыть) для закрытия диалогового окна Словаря имён тэгов.

**Значение удержания тэга**

Часть деталей Словаря имён тэгов включает два свойства для сохранения значений тэгов и изменений, вносимых оператором в пределы срабатывания сигнализации.

Все типы тэгов включают в себя свойство **Retentive Value** (Удерживающее значение). Выберите свойство **Retentive Value** (Удерживающее значение) для сохранения текущего значения тэга при остановке приложения. Когда вы снова запускаете приложение, функция **WindowViewer** использует сохранённое значение в качестве первоначального значения тэга.

**WindowViewer** не записывает сохранённые значения в устройства ввода/вывода, когда **WindowViewer** снова запускает приложение. Значения ввода/вывода обновляются после того, как Сервер ввода/вывода произведёт первоначальное сканирование устройства, предоставляющего данные.

Опция Retentive Value (Удерживающее значение) не может быть выбрана или сброшена или для новых или существующих тэгов, если действует функция WindowViewer. Когда вы выбираете данную опцию, первоначальное значение тэга непрерывно обновляется для отражения его текущего значения. Когда WindowViewer останавливается, первоначальное значение устанавливается на место последнего сохранённого значения. Если эта опция впоследствии очищается, первоначальное значение тэга устанавливается на место последнего сохранённого значения.

Целочисленные и вещественные тэги включают в себя свойство Retentive Parameters (Удерживающие параметры). Выбирайте Retentive Parameters для сохранения любых изменений, внесённых оператором в пределы предупредительных сигналов тэга при работе приложения. WindowViewer использует изменённые пределы предупредительных сигналов в качестве первоначального значения для пределов предупредительных сигналов при перезапуске приложения.

### Присоединение ввода/вывода

Все типы тэгов ввода/вывода должны идентифицировать Имя доступа (Access Name) и Имя элемента (Item Name) внешнего источника данных. За дополнительной информацией по заданию Имени доступа и Имени элемента для тэгов ввода/вывода обращайтесь к разделу «Установка параметров доступа ввода/вывода» в главе 5.

### Регистрация тэгов

Во время выполнения программы WindowViewer может записывать данные в файл хронологического журнала регистрации каждый раз, когда значение тэга изменяется в большей степени, чем заданная зона нечувствительности из журнала. WindowViewer также записывает данные в журнал с фиксированными интервалами, независимо от текущих значений тэгов. По умолчанию этот фиксированный интервал равен одному часу.

---

**Примечание:** Для лучшей управляемости и универсальности регистрации, рассмотрите возможность использования Сервера IndustrialSQL для сохранения хронологических данных InTouch.

---

Диалоговое окно Tagname Dictionary (Словарь имён тэгов) включает в себя отдельные опции для внесения данных и событий в журнал регистрации. Вы можете устанавливать опции регистрации величин. За информацией об установке регистрации событий обращайтесь к Главе 2 «Конфигурирование предупредительных сигналов» в «Руководстве по аварийным сигналам и событиям человеко-машинного интерфейса InTouch®».

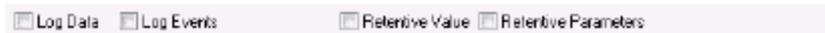
Для того чтобы значение тэга было записано в журнал хронологической регистрации, функция хронологической регистрации должна быть включена. За дополнительной информацией об установке общих свойств регистрации

обращайтесь к разделу «Конфигурирование хронологической регистрации» в главе 5.

В случае целочисленных и вещественных тэгов, вы можете установить зону нечувствительности из журнала в их соответствующих диалоговых окнах. Опция **LogDeadband** (Зона нечувствительности из журнала) указывает, на сколько технических единиц должно измениться значение тэга, чтобы была сделана запись в журнале регистрации.

#### Для конфигурирования регистрации тэга:

- 1 Откройте **Tagname Dictionary** (Словарь имён тэгов).
- 2 Выберите тэг, данные которого вы хотите сохранить в журнале регистрации.
- 3 Выберите **Log Data** (Данные из журнала).



- 4 Выберите **Log Events** (События из журнала) если вы хотите сохранить изменения в значениях тэга, внесённые оператором, системой вводом/вывода, функцией QuickScript или операционной системой. Поле **Priority** (Приоритет) появляется после того, как вы выберете **Log Events**.



Значение **Priority** определяет приоритет событий для тэга. Действующими величинами являются от 1 до 999, где 1 соответствует высшему приоритету, а 999 – низшему.

- 5 Щёлкните на опции **Save** (Сохранить) и затем закройте Словарь имён тэгов.

## Создание дискретных тэгов

Вы можете задавать дискретные тэги для иллюстрации двоичного состояния хода внутренних процессов в приложении InTouch. Дискретному тэгу должно быть присвоено первоначальное значение on (включено) или off (выключено). Также вы можете задать сообщения, которые будут появляться в окне предупредительных сигналов при наступлении аварийной ситуации в технологическом процессе, связанном с данным тэгом, или при выходе из указанной ситуации.

В нижеописанных операциях показано, как определять дискретный тэг памяти. Дискретные тэги ввода/вывода указывают на двоичное состояние всех входов и выходов с программируемых контроллеров, технологических компьютеров и данных из сетевых узлов.

За дополнительной информацией об установке свойств дискретного тэга ввода/вывода обращайтесь к разделу «Задание дискретного тэга ввода/вывода» в главе 5.

**Для определения первоначального значения и сообщений для дискретного тэга памяти:**

- 1 Выберите **Memory Discrete** (Дискретный тэг памяти) в качестве типа тэга в диалоговом окне **Tag Types** (Типы тэгов).
- 2 При необходимости выберите опцию **Details** (Детали) в верхней части диалогового окна **Tagname Dictionary** (Словарь имён тэгов) для того, чтобы показать опции деталей. Появляется часть диалогового окна **Tagname Dictionary**, содержащая детали.



- 3 Выберите **On** (Включение) или **Off** (Выключение) в качестве первоначального значения, связанного с тэгом. Тэгу присваивается данное первоначальное значение при запуске приложения.
- 4 Введите сообщение, которое должно появляться, когда тэг переходит в аварийное состояние и выходит из этого состояния.

Данное сообщение доступно для использования в любой анимационной ссылке или скрипте, независимо от того, имеет ли тэг конфигурированные предупредительные сигналы или нет.

- Если вы определяете дискретный предупредительный сигнал, который активен, когда значение тэга равно 1 (Включение, истина), сообщение, введённое в диалоговое окно Сообщение Off (выключение), появляется в поле **Value/Limit** (Значение/предел) вашего дисплея предупредительных сигналов. Когда предупредительный сигнал неактивен, появляется сообщение, введённое в диалоговое окно **Сообщение On** (включение).
  - Если вы определяете дискретный предупредительный сигнал, который является активным, когда значение тэга равно 0 (Выключено, ложь), то сообщение, введённое в диалоговое окно Сообщение On (включение), появляется в поле **Value/Limit** (Значение/предел) вашего дисплея предупредительных сигналов. Когда предупредительный сигнал неактивен, появляется сообщение, введённое в диалоговое окно Сообщение Off (выключение).
- 5 Сохраните ваши изменения, внесённые в тэг.

## Создание целочисленных и вещественных тэгов

Вы можете задавать целочисленные и вещественные тэги для того, чтобы демонстрировать числовые значения процессов, выполняемых в приложении InTouch.

В нижеописанных процедурах показано, как определять целочисленные и вещественные тэги памяти и ввода/вывода. Целочисленным и вещественным тэгам памяти должно присваиваться первоначальное значение. Вы также должны установить минимальных и максимальных значений данных.

За информацией об установке свойств ввода/вывода целочисленных и вещественных тэгов ввода/вывода обращайтесь к разделу «Задание целочисленных и вещественных тэгов ввода/вывода» в главе 5.

**Важно:** Вещественные числа InTouch ограничены восьмизначной точностью. Во избежание возможных ошибок округления не допускайте превышения восьмизначной точности при задании свойств ваших вещественных тэгов. За дополнительной информацией о том, как человеко-машинный интерфейс InTouch преобразует двоичные величины в вещественные, обращайтесь к разделу «Десятичные величины IEEE».

### Для определения целочисленных и вещественных значений тэгов

- 1 Присвойте тэгу тип целочисленного или вещественного тэга памяти в диалоговом окне **Tag Types** (Типы тэгов). Появляется часть диалогового окна Словаря имён тэгов, содержащая детали.

Initial Value:	0	Min Value:	0	Deadband:	15
Eng Units:	PSI	Max Value:	1500	Log Deadband:	25

- 2 Установите свойства для целочисленных и вещественных тэгов. Выполните следующие действия:
  - В диалоговом окне **Initial Value** (Первоначальное значение) введите целое или вещественное число, связанное с тэгом при запуске приложения.
  - В диалоговом окне **Min Value** (Минимальное значение) ведите минимальное целое или вещественное число для тэга.
  - Параметр **Min Value** устанавливает минимальное возможное значение для чисел, связанных с целочисленными и вещественными тэгами памяти.
  - В диалоговом окне **Max Value** (Максимальное значение) введите максимальное целое или вещественное число для тэга.
  - Параметр **Max Value** устанавливает максимальное возможное значение для чисел, связанных с целочисленными и вещественными тэгами памяти.

- В диалоговом окне **Eng Units** (Технические единицы) введите метку, которую вы хотите использовать для технических единиц тэга.

### 3 Сохраните изменения, внесённые вами в тэг

За дополнительной информацией об установке свойств зоны нечувствительности тэга и зона нечувствительности из журнала обращайтесь к разделу «Зоны нечувствительности тэгов» в главе 2.

## Создание тэгов сообщений

Вы можете задавать тэги сообщений как для внутренних, так и для внешних процессов. Эти тэги включают в себя свойства, которые должны быть заданы при первоначальном сообщении, которое появляется, когда программа WindowViewer запускает приложение, и комментарий, который можно прочитать в программе Alarm Viewer.

За дополнительной информацией об определении тэга сообщения ввода/вывода обращайтесь к разделу «Задание тэга ввода/вывода» в главе 5.

### Для определения тэга сообщения памяти

- 1 Присвойте тэгу тип тэга сообщения памяти в диалоговом окне **Tag Types** (Типы тэгов).
- 2 При необходимости выберите опцию **Details** (Детали) для перехода в часть диалогового окна Словаря имён тэгов, содержащую детали.

The image shows a dialog box with three input fields. The first field is labeled 'Maximum Length' and contains the number '131'. The second field is labeled 'Initial Value' and is empty. The third field is labeled 'Alarm Comment' and is also empty. The dialog box has a light purple background and a white border.

- 3 Установите свойства тэга сообщения памяти. Выполните следующие действия:
  - В диалоговом окне **Maximum Length** (Максимальная длина) введите целое число, которое представляет собой максимальное число знаков, которое может появиться в сообщении тэга. Или примите длину в 131 знак, действующую по умолчанию, которая представляет собой максимальную длину сообщения.
  - В диалоговом окне **Initial Value** (Первоначальное значение) введите текст сообщения, которое вы хотите видеть присвоенным тэгу, когда WindowViewer запускает приложение.
  - В диалоговом окне **Alarm Comments** (Комментарий к предупредительным сигналам) введите сообщение, которое может быть считано в блоке управления AlarmViewer, если тэг сообщения имеет выбранную опцию **Log Events** (События из журнала).
- 4 Сохраните изменения, которые вы внесли в тэг.

## Создание тэгов ввода/вывода

Тэги ввода/вывода имеют набор общих свойств, которые задают сетевую соединимость между приложением InTouch и внешними процессами. Часть диалогового окна Tagname Dictionary (Словарь имён тэгов), содержащая детали, включает в себя опции для установки внешних свойств тэгов ввода/вывода.

За дополнительной информацией об установке свойств ввода/вывода обращайтесь к разделу «Доступ к данным через ввод/вывод» в главе 5.

## Изменение тэгов

Изменение тэга аналогично созданию тэга. Вы выбираете тэг, который должен быть изменён, из Словаря имён тэгов. Затем вы изменяете свойства тэга, выполняя те же самые шаги, что и при создании тэга.

---

**Важно:** Изменить тип тэга после того, как он будет использован в приложении InTouch, непросто. Выбор типа тэга может быть ограничен аналогичными типами данных. Вам следует тщательно выбирать правильный тип данных при определении ваших тэгов.

---

### Для изменения тэга:

- 1 Откройте диалоговое окно **Tagname Dictionary** (Словарь имён тэгов).
- 2 Щёлкните на опции **Select** (Выбрать). Появляется диалоговое окно **Select Tag** (Выбрать тэг). Показывается список тэгов, определённых в текущий момент для приложения.
- 3 Выберите из списка тэг, который должен быть изменён.
- 4 Щёлкните на **ОК**. В диалоговом окне **Tagname Dictionary** (Словарь имён тэгов) показываются значения, заданные для выбранного тэга.
- 5 Внесите изменения в свойства тэга.
- 6 Щёлкните на опции **Save** (Сохранить) для обновления тэга с учётом ваших изменений.
- 7 Щёлкните на опции **Close** (Закрыть) для закрытия Словаря имён тэгов.

## Удаление тэгов

Для всех тэгов, определённых для приложения InTouch, ведётся счёт. Счёт тэгов не уменьшается автоматически при удалении окна, содержащего анимационные ссылки или скрипты. Тэги, связанные с удалённым окном, всё ещё считаются используемыми и не могут быть удалены.

Для того чтобы получить возможность удалить тэг, который больше не используется, вы должны закрыть WindowViewer и обновить счёт местного и удалённого использования тэгов. Вы

можете определить, где тэг используется, при помощи утилиты перекрёстной ссылки (Cross-Reference) интерфейса InTouch. За дополнительной информацией об использовании утилиты перекрёстной ссылки и обновлении счетов тэгов обращайтесь к разделу «Сокращение использования тэгов» в главе 8.

Тэги могут быть удалены после обновления счёта использования. За дополнительной информацией об удалении тэгов обращайтесь к разделу «Удаление неиспользуемых тэгов» в главе 8.

## Распечатка списка тэгов и информации об использовании

Вы можете распечатать содержимое приложения баз данных человеко-машинного интерфейса InTouch, окон и скриптов в программе WindowMaker. Утилита WindowMaker Printout (Распечатка WindowMaker) включает в себя опции для распечатки или сохранения в файл:

- Записи в базу данных;
- Окна
- Скрипты

За дополнительной информацией о распечатке списка тэгов обращайтесь к разделу «Список перекрёстных ссылок» в главе 8.

# Глава 3

## Системные тэги

Использование системных тэгов предоставляет информацию, относящуюся к системе, и стандартные функции, такие, как дата и время, для составления скриптов InTouch. Системные тэги составляют часть всех приложений.

Системные тэги идентифицируются знаком доллара (\$), используемым в качестве первого знака имени тэга в Словаре имён тэгов. Системные тэги не могут быть удалены. Вы можете только изменять комментарий, связанный с системным тэгом.

### Системные тэги

Системные тэги человеко-машинного интерфейса InTouch в следующей таблице:

Системный тэг	Назначение	Источник дополнительной информации
\$AccessLevel (Уровень доступа)	Целочисленный тэг, предназначенный только для чтения, который задаёт уровень доступа, связанный с оператором, в который в настоящий момент осуществлён вход. Эта информация может использоваться в анимационных ссылках или сценариях для контроля доступа оператора к конкретным функциям InTouch.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению».
\$ApplicationChanged (Приложение изменено)	Дискретный тэг, предназначенный только для чтения, который указывает, изменилось ли основное приложение в среде разработки сетевых приложений (NAD).	См. Главу 2, «Распределение приложений» в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению».

Системный тэг	Назначение	Источник дополнительной информации
\$ApplicationVersion (Версия приложения)	Вещественный тэг, предназначенный только для чтения, который задаёт текущий номер версии приложения, запущенного в программе WindowViewer.	См. Главу 2, «Распределённые приложения», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$ChangePassword (Сменить пароль)	Дискретный тэг, предназначенный только для записи, который показывает диалоговое окно «Change Password» (Сменить пароль) при установке на 1.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$ConfigureUsers (Конфигурировать пользователей)	Дискретный тэг, предназначенный только для записи, который показывает общее диалоговое окно «Configure Users» (Конфигурировать пользователей) для редактирования списка имён пользователей защитой).	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$Date (Дата)	Целочисленный тэг, предназначенный только для чтения, который показывает полное число дней, прошедших с 1 января 1970 года.	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® HMI».
\$DateString (Строка даты)	Тэг сообщений, предназначенный только для чтения, который показывает дату в том же самом формате, который задан из диалогового окна операционной системы «Windows» Regional and Language Options (Язык и региональные стандарты).	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® HMI».
\$DateTime (Дата и время)	Вещественный тэг, предназначенный только для чтения, который показывает дробное число дней, прошедших с 1 января 1970 года.	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® HMI».
\$Day (День)	Целочисленный тэг, предназначенный только для чтения, который показывает текущий день месяца (1-31).	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® HMI».

Системный тэг	Назначение	Источник дополнительной информации
\$HistoricalLogging (Регистрация предыстории)	Дискретный тэг, предназначенный для чтения/ записи, используемый для пуска и останова регистрации предыстории во время работы приложения «InTouch».	См. Главу 9, «Запись значений тэгов.
\$Hour (Час)	Целочисленный тэг, предназначенный только для чтения, который показывает текущий час как величины от 0 до 23.	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® HMI».
\$InactivityTimeout (Тайм-аут режима бездействия)	Дискретный тэг, предназначенный только для чтения, который указывает на то, что период бездействия пользователя истёк. Значение тэга \$InactivityTimeout может использоваться для автоматического выхода оператора из системы.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$InactivityWarning (Предупреждение о бездействии)	Дискретный тэг, предназначенный только для чтения, который указывает на истечение периода предупреждения о бездействии. Значение \$InactivityWarning может использоваться для выдачи оператору предупреждения о бездействии.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$Language (Язык)	Целочисленный тэг, предназначенный для чтения/ записи that specifies the language ID of the language shown in an InTouch application.	См. Главу 5, «Переключение языка во время выполнения программы», в «Руководстве по управлению приложениями человеко- машинного интерфейса InTouch® и их расширению»
\$LogicRunning (Запуск логики)	Дискретный тэг, предназначенный для чтения/ записи, который может использоваться для пуска и останова скрипта приложения.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$Minute (Минута)	Целочисленный тэг, предназначенный только для чтения, который показывает текущую минуту (0-59).	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® HMI».

Системный тэг	Назначение	Источник дополнительной информации
\$Month (Месяц)	Целочисленный тэг, предназначенный только для чтения, который показывает порядковый номер текущего месяца (1-12).	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® НМІ».
\$Msec (Миллисекунда)	Целочисленный тэг, предназначенный только для чтения, который показывает текущую миллисекунду (0-999).	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® НМІ».
\$NewAlarm (Новый аларм)	Дискретный тэг, предназначенный для чтения/записи, который указывает, когда был подан новый местный аларм.	См. Главу 6, «Управление свойствами алармов тэгов и групп во время выполнения программы», в «Руководстве по обработке алармов и событий InTouch® НМІ».
\$ObjHor	Целочисленный тэг, предназначенный только для чтения, который показывает горизонтальное расположение центра выделенного объекта на экране по пикселям.	См. Главу 4, Анимационные объекты, in the Руководство по визуализации InTouch® НМІ.
\$ObjVer	Целочисленный тэг, предназначенный только для чтения, который показывает вертикальное расположение центра выделенного объекта на экране по пикселям.	См. Главу 4, «Анимационные объекты», в «Руководство по визуализации InTouch® НМІ».
\$Operator (Оператор)	Тэг сообщений, предназначенный только для чтения, который показывает имя оператора, который вошёл в приложение InTouch.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$OperatorDomain (Домен оператора)	Тэг сообщений, предназначенный только для чтения, который содержит имя домена или машины, указанное в журнале, когда приложение защищено работающим средством защиты на основе операционной системы.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»

Системный тэг	Назначение	Источник дополнительной информации
\$OperatorDomainEntered (Вход в домен оператора осуществлён)	Тэг, предназначенный только для записи, используемый для входа оператора с текущим мандатом, присвоенным системному тэгу, который оканчивается словом «Entered» (Введено), используемым в качестве суффикса его имени. Значение, присвоенное тэгу \$OperatorDomainEntered, используется в качестве домена пользователя при попытке входа в систему.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$OperatorEntered (Вход оператора осуществлён)	Тэг сообщения, предназначенный для чтения/записи, используемый для входа оператора с текущим мандатом, присвоенным системному тэгу, который оканчивается словом «Entered» (Введено), используемым в качестве суффикса его имени. Значение, присвоенное тэгу \$OperatorEntered, используется в качестве имени пользователя оператора при попытке входа в систему.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$OperatorName (Имя оператора)	Тэг сообщений, предназначенный только для чтения, который показывает полное имя оператора при использовании идентификации на основании операционной системы или идентификации ArchestrA®.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$PasswordEntered (Пароль введён)	Тэг сообщения, предназначенный только для записи, используется для входа оператора с мандатом, присвоенным системному тэгу, который оканчивается словом «Entered» (Введено), используемым в качестве суффикса его имени. Значение, присвоенное тэгу \$PasswordEntered используется в качестве пароля оператора при попытке входа в систему.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»

Системный тэг	Назначение	Источник дополнительной информации
\$Second (Секунда)	Целочисленный тэг, предназначенный только для чтения, который показывает текущую секунду (0-59).	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® НМІ».
\$StartDdeConversations	Дискретный тэг, предназначенный для чтения/записи, используемый для начала неиницированных диалогов во время выполнения программы.	См. Главу 5, Доступ к данным с вводом/выводом.
\$System (Система)	Тэг, предназначенный только для чтения, который идентифицирует корневую группу алармов.	См. Главу 1, Обзор алармов и событий, в «Руководстве по обработке алармов и событий InTouch® НМІ».
\$Time (Время)	Целочисленный тэг, предназначенный только для чтения, который показывает истекшее время в миллисекундах с полуночи текущего дня.	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® НМІ».
\$TimeString (Строка времени)	Тэг сообщений, предназначенный только для чтения, который показывает текущее время в том же самом формате, который задан из диалогового окна операционной системы «Windows» «Regional and Language Options» (Язык и региональные стандарты).	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® НМІ».
\$VerifiedUserName (Подтвержденное имя пользователя)	Тэг сообщений, предназначенный только для чтения, который содержит либо подтвержденное полное имя пользователя, либо нуль.	См. Главу 4, «Защита InTouch», в «Руководстве по управлению приложениями человеко-машинного интерфейса InTouch® и их расширению»
\$Year (Год)	Целочисленный тэг, предназначенный только для чтения, который показывает текущий год в виде четырёхзначного числа.	См. Главу 6, «Встроенные функции», в «Руководстве по скриптам и логике работы в InTouch® НМІ».

## Глава 4

# Использование точечных полей для просмотра или изменения свойств тэгов

Каждый тип тэга InTouch имеет уникальный набор свойств, который описывает данные или возможные условия, связанные с тэгом. Точечное поле идентифицирует свойство тэга. Имеется точечное поле почти для каждого свойства тэга, указанного в Словаре имён тэгов. Некоторые точечные поля являются общими для каждого типа тэга InTouch. Например, точечное поле `.Name` всегда связывается с именем тэга. Другие точечные поля применяются только к уникальным свойствам конкретного типа тэга.

Используя точечные поля в скрипте, выражении или входе пользователя, вы можете контролировать и изменять свойства тэга в процессе работы приложения. В нижеследующем примере показан синтаксис точечного поля для получения доступа к свойствам тэга в скрипте или выражении.

**`tag_name.property_dotfield`**

Например, для того, чтобы дать возможность оператору изменять предел подачи аларма HiHi в процессе работы приложения, вы можете создать аналоговую сенсорную ссылку входа пользователя. Затем примените ссылку к кнопке, которая определяется с использованием выражения точечного поля `Analog_Tag.HiHiLimit`. Во время выполнения программы оператор просто щёлкает на кнопке и набирает новое значение предела подачи аларма HiHi для тэга.

Вы можете использовать точечные поля для обеспечения возможности ввода и вывода данных, связанных с тэгом, и можете использовать исторические точечные поля для изменения тенденции за истекший период, демонстрируемой в текущий момент из запущенного приложения. Например, вы можете использовать точечные поля в скриптах, которые позволяют операторам изменять прокрутку тенденции за истекший период, фиксировать или перемещать скутеры на тенденции или

переназначать перья для новых тэгов.

Каждый тип тэга InTouch имеет набор точечных полей, связанных с его уникальными свойствами. В приведённой ниже таблице показан алфавитный список точечных полей для всех типов тэгов.

Точечное поле	Типы тэгов:														
	Память				Ввод/вывод				Косвенный			Другие			
	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Аналоговый	Message	Группа алармов	Тенденция за истекшее время	Распределённый объект аларма/управление	Идентификатор тэга
.Ack	•	•	•		•	•	•		•	•		•			
.AckDev		•	•			•	•			•		•			
.AckDsc	•				•				•			•			
.AckROC		•	•			•	•			•		•			
.AckValue		•	•			•	•			•		•			
.Alarm	•	•	•		•	•	•		•	•		•			
.AlarmAccess														•	
.AlarmAckModel	•	•	•		•	•	•		•	•					
.AlarmClass														•	
.AlarmComment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
.AlarmDate														•	
.AlarmDev		•	•			•	•			•		•			
.AlarmDevCount		•	•			•	•			•		•			
.AlarmDevDeadband		•	•			•	•			•					
.AlarmDevUnAckCount		•	•			•	•			•		•			
.AlarmDisabled	•	•	•		•	•	•		•	•		•			
.AlarmDsc	•				•				•			•			
.AlarmDscCount	•				•				•			•			
.AlarmDscDisabled	•				•				•			•			
.AlarmDscEnabled	•				•				•			•			
.AlarmDscInhibitor	•				•				•			•			
.AlarmDscUnAckCount	•				•				•			•			
.AlarmEnabled	•	•	•		•	•	•		•	•		•			
.AlarmGroup														•	
.AlarmGroupSel														•	
.AlarmHiDisabled		•	•			•	•			•					
.AlarmHiEnabled		•	•			•	•			•					
.AlarmHiHiDisabled		•	•			•	•			•					

Точечное поле	Типы тэгов:													
	Память				Ввод/вывод				Косвенный			Другие		
	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Аналоговый	Message	Группа алармов	Тенденция за истекшее время	Распределённый объект аларма/управление
.AlarmHiHiEnabled		•	•			•	•			•				
.AlarmHiHiInhibitor		•	•			•	•			•				
.AlarmHiInhibitor		•	•			•	•			•				
.AlarmLimit													•	
.AlarmLoDisabled		•	•			•	•			•				
.AlarmLoEnabled		•	•			•	•			•				
.AlarmLoInhibitor		•	•			•	•			•				
.AlarmLoLoDisabled		•	•			•	•			•				
.AlarmLoLoEnabled		•	•			•	•			•				
.AlarmLoLoInhibitor		•	•			•	•			•				
.AlarmMajDevDisabled		•	•			•	•			•				
.AlarmMajDevEnabled		•	•			•	•			•				
.AlarmMajDevInhibitor		•	•			•	•			•				
.AlarmMinDevDisabled		•	•			•	•			•				
.AlarmMinDevEnabled		•	•			•	•			•				
.AlarmMinDevInhibitor		•	•			•	•			•				
.AlarmName													•	
.AlarmOprName													•	
.AlarmOprNode													•	
.AlarmPri													•	
.AlarmProv													•	
.AlarmROC		•	•			•	•			•				
.AlarmROCCount		•	•			•	•			•				
.AlarmROCDisabled		•	•			•	•			•				
.AlarmROCEEnabled		•	•			•	•			•				
.AlarmROCInhibitor		•	•			•	•			•				
.AlarmROCUAckCount		•	•			•	•			•				
.AlarmState													•	
.AlarmTime													•	
.AlarmTotalCount	•	•	•		•	•	•		•	•	•			
.AlarmType													•	
.AlarmUnAckCount	•	•	•		•	•	•		•	•	•			

Точечное поле	Типы тэгов:														
	Память				Ввод/вывод				Косвенный			Другие			
	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Аналоговый	Message	Группа алармов	Тенденция за истекшее время	Распределённый объект аларма/управление	Идентификатор тэга
.AlarmUserDefNum1	•	•	•		•	•	•		•	•		•			
.AlarmUserDefNum1Set	•	•	•		•	•	•		•	•		•			
.AlarmUserDefNum2	•	•	•		•	•	•		•	•		•			
.AlarmUserDefNum2Set	•	•	•		•	•	•		•	•		•			
.AlarmUserDefStr	•	•	•		•	•	•		•	•		•			
.AlarmUserDefStrSet	•	•	•		•	•	•		•	•		•			
.AlarmValDeadband		•	•		•	•			•						
.AlarmValue			•			•			•		•		•		
.AlarmValueCount		•	•		•	•			•		•				
.AlarmValueUnAckCount		•	•		•	•			•		•				
.Caption														•	
.ChartLength												•			
.ChartStart												•			
.Comment	•	•	•	•	•	•	•	•	•	•	•	•			•
.DevTarget		•	•		•	•			•						
.DisplayMode												•			
.Enabled														•	
.EngUnits		•	•		•	•			•						
.Freeze														•	
.HiHiLimit		•	•		•	•			•						
.HiHiSet		•	•		•	•			•						
.HiHiStatus		•	•		•	•			•						
.HiLimit		•	•		•	•			•						
.HiSet		•	•		•	•			•						
.HiStatus		•	•		•	•			•						
.ListChanged														•	
.ListCount														•	
.ListIndex														•	
.LoLimit		•	•		•	•			•						
.LoLoLimit		•	•		•	•			•						
.LoLoSet		•	•		•	•			•						
.LoLoStatus		•	•		•	•			•						

Точечное поле	Типы тэгов:														
	Память				Ввод/вывод				Косвенный			Другие			
	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Аналоговый	Message	Группа алармов	Тенденция за истекшее время	Распределённый объект аларма/управление	Идентификатор тэга
.LoSet		•	•			•	•			•					
.LoStatus		•	•			•	•			•					
.MajorDevPct		•	•			•	•			•					
.MajorDevSet		•	•			•	•			•					
.MajorDevStatus		•	•			•	•			•					
.MaxEU		•	•			•	•			•					
.MaxRange												•			
.MaxRaw		•	•			•	•			•					
.MinEU		•	•			•	•			•					
.MinorDevPct		•	•			•	•			•					
.MinorDevSet		•	•			•	•			•					
.MinorDevStatus		•	•			•	•			•					
.MinRange												•			
.MinRaw		•	•			•	•			•					
.Name	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
.NewIndex														•	
.NextPage														•	
.Normal	•	•	•		•	•	•		•	•	•				
.NumAlarms														•	
.OffMsg	•				•				•						
.OnMsg	•				•				•						
.PageNum														•	
.Pen1 до .Pen8												•			
.PendingUpdates														•	
.PrevPage														•	
.PriFrom														•	
.PriTo														•	
.Quality	•	•	•	•	•	•	•	•	•	•	•				
.QualityLimit	•	•	•	•	•	•	•	•	•	•	•				
.QualityLimitString	•	•	•	•	•	•	•	•	•	•	•				
.QualityStatus	•	•	•	•	•	•	•	•	•	•	•				
.QualityStatusString	•	•	•	•	•	•	•	•	•	•	•				

Точечное поле	Типы тэгов:													
	Память				Ввод/вывод				Косвенный			Другие		
	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Аналоговый	Message	Группа алармов	Тенденция за истекшее время	Распределённый объект аларма/управление
.QualitySubstatus	•	•	•	•	•	•	•	•	•	•				
.QualitySubstatusString	•	•	•	•	•	•	•	•	•	•				
.QueryState													•	
.QueryType													•	
.RawValue	•	•	•		•	•	•		•	•	•			
.ReadOnly														•
.Reference	•	•	•	•	•	•	•	•	•	•	•			
.ReferenceComplete	•	•	•	•	•	•	•	•	•	•	•			
.ROCPct		•	•			•	•			•				
.ROCSet		•	•			•	•			•				
.ROCStatus		•	•			•	•			•				
.ScooterLockLeft												•		
.ScooterLockRight												•		
.ScooterPosLeft												•		
.ScooterPosRight												•		
.Successful													•	
.SuppressRetain													•	
.TagID	•	•	•		•	•	•		•	•				
.TimeDate	•	•	•	•	•	•	•	•	•	•	•			
.TimeDateString	•	•	•	•	•	•	•	•	•	•	•			
.TimeDateTime	•	•	•	•	•	•	•	•	•	•	•			
.TimeDay	•	•	•	•	•	•	•	•	•	•	•			
.TimeHour	•	•	•	•	•	•	•	•	•	•	•			
.TimeMinute	•	•	•	•	•	•	•	•	•	•	•			
.TimeMonth	•	•	•	•	•	•	•	•	•	•	•			
.TimeMsec	•	•	•	•	•	•	•	•	•	•	•			
.TimeSecond	•	•	•	•	•	•	•	•	•	•	•			
.TimeTime	•	•	•	•	•	•	•	•	•	•	•			
.TimeTimeString	•	•	•	•	•	•	•	•	•	•	•			
.TimeYear	•	•	•	•	•	•	•	•	•	•	•			
.TopIndex													•	
.TotalPages													•	

Точечное поле	Типы тэгов:														
	Память				Ввод/вывод				Косвенный			Другие			
	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Целочисленный	Вещественный	Message	Дискретный	Аналоговый	Message	Группа алармов	Тенденция за истекшее время	Распределённый объект аларма/управление	Идентификатор тэга
.UnAck	•	•	•		•	•	•		•	•		•			
.UpdateCount													•		
.UpdateInProgress													•		
.UpdateTrend													•		
.Value(Tagname)	•	•	•	•	•	•	•	•	•	•	•	•			•
.Value(Windows Control)														•	
.Visible														•	

Точечные поля могут быть подразделены по категориям в соответствии с функциями, для выполнения которых они предназначены. За дополнительной информацией о функциональных категориях точечных полей обращайтесь к следующим разделам:

Категория	Обращайтесь к разделу
Значения и пределы	Изменение пределов значений тэга в главе 4.
Параметры алармов	Глава 6, Управление свойствами алармов тэгов и групп, в «Руководстве по обработке алармов и событий InTouch® HMI»
Ввод/вывод	Доступ к данным с вводом/выводом в главе 5.
Объект распределённых алармов	Приложение А, «Работа с объектом распределённых алармов», в «Руководстве по обработке алармов и событий InTouch®».
Дисплей тенденций	Анализ тенденций данных тэгов в главе 10.
Управление окнами	Глава 5, «Мастера», в «Руководстве по визуализации InTouch® HMI»

## Изменение пределов значений тэга

Необработанные входные данные с сервера ввода/вывода преобразуются в диапазон значений, приемлемых для приложения InTouch. Значения тэга фиксируются в диапазоне, ограниченном минимальной и максимальной величинами, заданными свойствами Min Raw и Max Raw Словаря имён тэгов.

Затем указанные необработанные значения преобразуются в диапазон технических единиц, установленный опциями Min EU и Max EU. Вы используете набор точечных полей для контроля и изменения необработанных величин тэга и диапазона технических единиц.

В нижеприведённой таблице приведены точечные поля, которые контролируют или изменяют значения тэгов в процессе работы приложения.

Точечное поле	Показывает
.MinRaw	Нижняя установка фиксации необработанного значения, принятого с сервера ввода/вывода.
.MaxRaw	Верхняя установка фиксации необработанного значения, принятого с сервера ввода/вывода.
.MinEU	Тэгу присваивается минимальное значение в технических единицах.
.MaxEU	Тэгу присваивается максимальное значение в технических единицах.
.EngUnits	Значение текста, присвоенное аналоговому тэгу из опции Eng Units Словаря имён тэгов.
.RawValue	Фактическое дискретное или аналоговое значение, принятое тэгом с сервера ввода/вывода до применения масштабирования.
.Value	Текущее значение тэга.
.OnMsg	Сообщение, присваиваемое дискретному тэгу, когда его значение оценивается как True (истинное), On (включённое) или 1.
.OffMsg	Сообщение, присваиваемое дискретному тэгу, когда его значение оценивается как False (ложное), Off (выключенное) или 0.
.Comment	Комментарий тэга, заданный из Словаря имён тэгов.

## Изменение предела необработанного значения

Необработанные входные значения тэгов с сервера ввода/вывода могут требовать привязки до того, как их можно будет использовать в приложении InTouch. Привязка ограничивает необработанные значения до диапазона с определёнными нижними и верхними пределами. Точечные поля `.MinRaw` и `.MaxRaw` показывают нижние и верхние границы диапазона необработанных величин на входе.

### **.MinRaw Dotfield**

Точечное поле `.MinRaw` показывает нижнюю установку привязки `Min Raw`, присвоенную тэгу. Значение для точечного поля `.MinRaw` приходит из значения `Min Raw`, присвоенного тэгу ввода/вывода в Словаре имён тэгов. Любое необработанное значение, меньшее, чем данная установка, привязывается к указанной минимальной величине.

#### **Категория**

Тэги

#### **Использование**

`tag_name.MinRaw;`

#### **Параметр**

*Tag\_name*

Имя любого дискретного тэга ввода/вывода, косвенного дискретного тэга, целочисленного тэга ввода/вывода, целочисленного тэга памяти, косвенного аналогового тэга, сообщения ввода/вывода или косвенного тэга сообщения.

#### **Примечания**

Это точечное поле, предназначенное только для чтения, показывает значение, присвоенное установке нижней привязки `Min Raw`.

#### **Тип данных**

Вещественный или целочисленный (только для чтения).

#### **Действующие значения**

Любое аналоговое значение.

#### **Пример**

Нижеследующий скрипт показывает окно ошибки, если необработанное значение, связанное с давлением на входе насоса, выходит за пределы нижней и верхней границ, установленных свойствами тэга `Min Raw` и `Max Raw`.

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

**Смотрите также**

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

**Точечное поле .MaxRaw**

Точечное поле .MaxRaw показывает верхнюю установку привязки Max Raw, присвоенную тэгу ввода/вывода из свойства Max Raw в Словаре имён тэгов. Любое необработанное значение данных, которое превышает данную установку, привязывается к этому максимальному необработанному значению.

**Категория**

Тэги

**Использование**

*Tag\_name*.MaxRaw

**Параметр**

*Tag\_name*

Имя любого целочисленного тэга ввода/вывода, вещественного тэга памяти, косвенного аналогового тэга, тэга сообщения ввода/вывода.

**Примечания**

Это точечное поле, предназначенное только для чтения, показывает значение, присвоенное верхней установке привязки Max Raw.

**Тип данных**

Вещественные или целочисленные (только для чтения).

**Действующие значения**

Любое аналоговое значение.

**Пример**

Нижеследующий скрипт определяет, не выходит ли значение тэга за пределы нормального рабочего диапазона, и, если это так, показывается соответствующее окно.

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

**Смотрите также**

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

## Просмотр необработанного значения тэга

Точечное поле `.RawValue` показывает фактическое дискретное или аналоговое значение контролируемого свойства, полученное с сервера ввода/вывода. Необработанное значение представляет собой фактическое входное значение перед тем, как будут применены привязка и масштабирование для нормирования значения в соответствии с техническими единицами тэга.

### Точечное поле `.RawValue`

Точечное поле `.RawValue` показывает фактическое значение, полученное с сервера ввода/вывода посредством `WindowViewer`. Точечное поле `.RawValue` позволяет получить доступ к значению тэга ввода/вывода до того, как интерфейс `InTouch` применит масштабирование.

#### Категория

Тэги

#### Использование

`Tag_name.RawValue`

#### Параметр

*Tag\_name*

Любой дискретный тэг ввода/вывода, косвенный дискретный тэг, целочисленный тэг ввода/вывода, косвенный аналоговый тэг, тэг сообщения ввода/вывода или косвенный тэг сообщения.

#### Примечания

Это точечное поле, предназначенное только для чтения, используется для показа фактического дискретного или аналогового значения ввода/вывода до того, как интерфейс `InTouch` применит масштабирование.

#### Тип данных

Любой тип данных. Например, вещественные для вещественных тэгов и дискретные для дискретных тэгов (только для чтения).

#### Действующие величины

В тех случаях, когда могут быть заданы дискретные или аналоговые величины.

#### Пример

Нижеследующий скрипт выдаёт предупредительное сообщений, когда необработанная величина давления оказывается ниже или выше минимальной и максимальной границы привязки тэга.

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

```
ENDIF;
```

**Смотрите также**

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

## Изменение пределов значений технических единиц

Значение с сервера ввода/вывода считается необработанным, когда оно впервые поступает в WindowViewer. Необработанные значения могут требовать масштабирования. Человеко-машинный интерфейс InTouch выполняет арифметическое преобразование на необработанных привязанных входных величинах для масштабирования их в соответствии с диапазоном технических единиц тэга. Целочисленный и вещественный типы тэгов ввода/вывода включают в себя свойства Min EU и Max EU, которые показывают нижнюю и верхнюю границы диапазона технических величин.

### Точечное поле .MaxEU

Точечное поле .MaxEU показывает максимальное значение технической единицы, присвоенное заданному тэгу из Словаря имён тэгов.

#### Категория

Тэги

#### Использование

*Tag\_name*.MaxEU

#### Параметр

*Tag\_name*

Любой целочисленный, вещественный или косвенный аналоговый тэг.

#### Примечания

Точечное поле .MaxEU используется для масштабирования необработанных значений данных в соответствии с диапазоном технических единиц, определённых для тэга. Оно определяет верхний предел диапазона технических единиц.

#### Тип данных

Вещественный для вещественных тэгов и целочисленный для целочисленных тэгов (только для чтения).

#### Действующие величины

Зависят от заданного типа тэга.

#### Пример

Показание уровня считывается программируемым логическим контроллером на участке эксплуатации. Датчик уровня посылает сигнал в диапазоне между 4 и 20 мА. Программируемый логический контроллер преобразует данный сигнал в целочисленное значение в диапазоне от 0 до 4095. Данное значение присваивается тэгу TankTwoLevel.

Вывод на дисплей необработанного значения (между 0 и 4095) не предоставляет оператору никакой полезной информации. Данную величину необходимо масштабировать для приведения её в соответствующий диапазон технических величин.

Для осуществления этого процесса поля минимальной технической единицы и максимальной технической единицы должны быть установлены правильно. В нашем примере, если необработанное значение 0 (4 мА с поля) переводится в «0 галлонов», а значение 4095 (20 мА с поля) переводится в «100 галлонов», потребуется следующая установка для отображения правильной величины на экране:

```
TankTwoLevel.MinRaw = 0;  
TankTwoLevel.MaxRaw = 4095;  
TankTwoLevel.MinEU = 0;  
TankTwoLevel.MaxEU = 100;
```

При данных установках, когда необработанное значение в поле равно 4095, значение, показываемое оператору, равно 100.

#### Смотрите также

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

### Точечное поле .MinEU

Точечное поле .MinEU показывает минимальное значение технической единицы, присвоенное заданному тэгу из Словаря имён тэгов.

#### Категория

Тэги

#### Использование

*Tag\_name*.MinEU

#### Параметр

*Tag\_name*

Любой целочисленный, вещественный или косвенный аналоговый тэг.

#### Примечания

Точечное поле .MinEU используется для масштабирования необработанных значений данных для приведения их в диапазон технических единиц, определённый для тэга. Он определяет нижний предел диапазона технических величин.

#### Тип данных

Вещественные для вещественных тэгов и целочисленные для целочисленных тэгов (только для чтения).

#### Действующие величины

Зависит от данного типа тэга.

**Пример**

В данном примере диапазон технических единиц, определённый для тэга 1, присваивается тэгу `AbsoluteTagRange`.

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

**Смотрите также**

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

## Замена технических единиц тэга

Вы можете присвоить точечное поле тэгу для определения текстового значения технической единицы, присвоенной тэгу.

### Точечное поле `.EngUnits`

Точечное поле `.EngUnits` показывает текстовое значение, присваиваемое аналоговому тэгу с использованием свойства `Eng Units` (Технические единицы). Точечное поле `.EngUnits` показывает техническую единицу как текстовую величину.

---

**Примечание:** Величины, записанные в данное поле, не являются удерживающими.

---

**Категория**

Тэги

**Использование**

***Tag\_name*.EngUnits**

**Параметр**

*Tag\_name*

Любой целочисленный, вещественный или косвенный аналоговый тэг.

**Тип данных**

Сообщение (предназначенное для чтения/записи).

**Примечания**

Точечное поле `.EngUnits` не оказывает влияния на масштаб, преобразование или формат фактических данных, связанных с тэгом.

**Действующие величины**

Любая строка, содержащая от 0 до 31 знака.

**Пример**

Нижеследующий скрипт вызывает функцию преобразования температуры в градусах Фаренгейта, если технической единицей тэга является градус Цельсия.

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

**Смотрите также**

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

## Просмотр значения тэга в технических единицах

Тэги InTouch присваиваются точечному полю, действующему по умолчанию, когда никакое точечное поле не задано явно в рамках приложения.

### Точечное поле .Value

Точечное поле .Value показывает текущее значение заданного тэга в технических единицах. .Value представляет собой точечное поле InTouch, которое подразумевается применимым ко всем тэгам. Если тэг не имеет присвоенного точечного поля, подразумевается точечное поле .Value.

**Категория**

Тэги

**Использование**

*tag\_name*. Значение

**Параметр**

*tag\_name*

Любой тип тэга, за исключением тэга тенденции за истекший период (Hist Trend).

**Примечания**

Вам редко требуется использовать точечное поле .Value. Однако в некоторых случаях, оно делает расчёт или параметр более чётким.

**Тип данных**

Тот же, что и заданный тип тэга (предназначенный для чтения/записи).

**Действующие величины**

Зависит от данного типа тэга.

**Пример**

Нижеследующее предложение устанавливает значение целочисленного тэга памяти PumpRPM равным 100:

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

Что функционально эквивалентно:

```
PumpRPM=100 ;
```

Обзор или изменение дискретных тэговых сообщений  
Точечные поля .OnMsg и .OffMsg показывают сообщения, присвоенные состоянию включения или выключения дискретного тэга из Словаря имён тэгов. Сообщения о состоянии включения и выключения тэга представляют собой короткие строки с максимальной длиной в 15 знаков.

## Точечное поле `.OnMsg`

Точечное поле `.OnMsg` позволяет получить доступ к сообщению `On` (включённое состояние), присвоенному дискретному тэгу из Словаря имён тэгов.

### Категория

Тэги

### Использование

*Tag\_name*.OnMsg

### Параметр

*Tag\_name*

Любой дискретный тэг.

### Тип данных

Сообщение (предназначенное для чтения/записи).  
Значения, записанные в данное точечное поле, не являются удерживающими.

### Действующие величины

Любая строка, содержащая от 0 до 15 знаков.

### Пример

Следующее предложение выдаёт сообщение, если сообщению о включении (`On`) косвенного тэга `IndPumpState` присваивается строковое значение “Pump1 running”.

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

### Смотрите также

`.OffMsg`

## Точечное поле `.OffMsg`

Точечное поле `.OffMsg` позволяет получить доступ к сообщению `Off` (выключенное состояние), присвоенному дискретному тэгу из Словаря имён тэгов.

### Категория

Тэги

### Использование

*Tag\_name*.OffMsg

### Параметр

*Tag\_name*

Любой дискретный тэг.

### Тип данных

Сообщение (предназначенное для чтения/записи).  
Значения, записанные в данное точечное поле, не являются удерживаемыми.

### Действующие величины

Любая строка, содержащая от 0 до 15 знаков.

### Пример

Следующее предложение присваивает соответствующую строку тэгу StateMessage (Сообщение о состоянии) в соответствии с состоянием тэга MyDiscrete.

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

Смотрите также

.OnMsg

## Просмотр или изменение комментария к тэгу

Комментарий к тэгу может быть перманентно изменён только из Словаря имён тэгов. Вы можете присвоить другой комментарий с точечным полем .Comment при работе приложения. Это изменяет комментарий только на то время, в течение которого выполняется программа. Это не изменяет перманентно комментарий к тэгу в Словаре имён тэгов. После завершения работы и последующего перезапуска программы WindowViewer, тэгу присваивается первоначальный комментарий.

### Точечное поле .Comment

Точечное поле .Comment показывает комментарий, присвоенный тэгу из Словаря имён тэгов. Комментарий к тэгу может представлять собой строку длиной до 50 знаков.

### Категория

Тэги

### Использование

*Tag\_name*. Комментарий

### Параметр

*Tag\_name*

Любое имя тэга.

### Примечания

Комментарий, связанный с тэгом, может быть изменён с использованием точечного поля .Comment во время работы приложения InTouch. После остановки приложения оригинальный комментарий, заданный в Словаре имён тэгов, остается присвоенным тэгу.

**Тип данных**

Сообщение.

**Действующие величины**

Любая строка от 1 до 50 знаков.

**Пример**

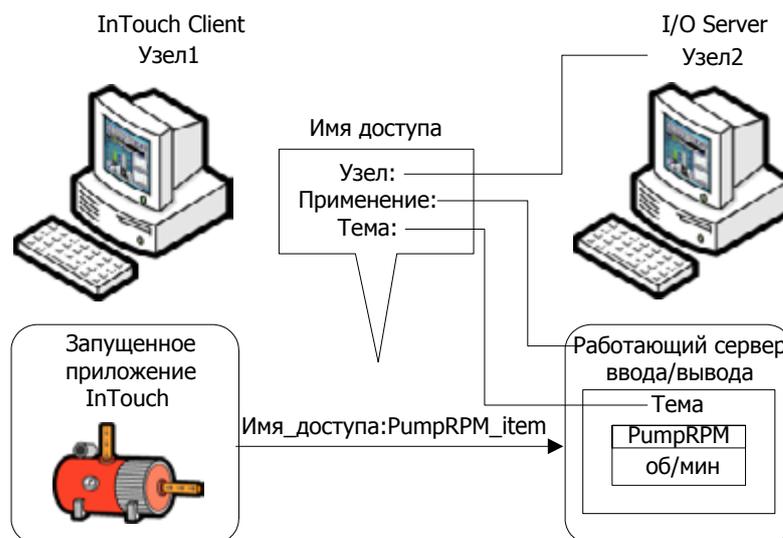
Следующее сообщение создаёт сообщение оператора, объединяя комментарий, присвоенный тэгу, с именем тэга:

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

# Глава 5

## Доступ к данным с вводом/выводом

Вы можете разработать распределённые приложения, в которых функциональные компоненты системы InTouch расположены на различных узлах. На рисунке, приведённом ниже, показано, каким образом вы конфигурируете запрос на ввод/вывод для данных, хранящихся на другом узле.



Вы можете установить приложение InTouch для идентификации элемента данных, хранящегося на другом узле, с использованием условных обозначений адресации и трёх частей. Эти условные обозначения включают в себя название приложения, название темы и название элемента. Для получения данных с удалённого узла вам потребуется конфигурировать Имя доступа для вашего приложения InTouch, в котором заданы эти три элемента.

Например, если вы хотите получить доступ к данным от удалённого сервера ввода/вывода, работающего на другом узле, ваше Имя доступа будет состоять из следующих компонентов:

Опция Access Name (Имя доступа)	Описание
<b>Node Name</b> (Имя узла)	Имя узла компьютера, на котором запущена программа Сервера ввода/вывода.
<b>Application Name</b> (Имя приложения)	Имя программы Сервера ввода/вывода, работающей на данном узле. Например, DASMTEthernet идентифицирует имя приложения Wonderware Mitsubishi Ethernet DAServer. За дополнительной информацией об имени приложения, связанного с Wonderware DAServers, обращайтесь к документации Wonderware DAServer.
<b>Topic Name</b> (Имя темы)	Метка, присвоенная группе устройств Сервера ввода/вывода.
<b>Item Name</b> (Имя элемента)	Конкретный элемент данных, определённый именем элемента устройства Сервера ввода/вывода или ссылкой на элемент.

Если вы используете крупноформатную таблицу Excel в качестве вашего источника данных InTouch, вы можете определить ваше Имя доступа следующим образом:

Опция Access Name (Имя доступа)	Описание
<b>Node Name</b> (Имя узла)	Имя узла компьютера, на котором запущена программа Excel.
<b>Application Name</b> (Имя приложения)	Excel является Именем приложения.
<b>Topic Name</b> (Имя темы)	Имя книги и крупноформатной таблицы Excel, содержащей запрашиваемые данные. Например, [Book1]Sheet1.
<b>Item Name</b> (Имя элемента)	Имя ячейки крупноформатной таблицы Excel, содержащей запрашиваемые данные. Например, R1C1 для ячейки A1.

Помимо приложения, темы и элемента, вам требуется задать тип данных, расположенных на удалённом узле. Эта информация определяет тип ввода/вывода для тэга, когда он определён в Словаре имён тэгов.

## Поддерживаемые коммуникационные протоколы InTouch

Вы можете конфигурировать человеко-машинный интерфейс InTouch для использования DDE или SuiteLink. Человеко-машинный интерфейс InTouch также поддерживает коммуникационный протокол Сервера сообщений Wonderware Archestr.

За дополнительной информацией об использовании Сервера сообщений в рамках приложения InTouch, обращайтесь к разделу «Доступ к данным Сервера приложений из InTouch» в главе 5.

### Динамический обмен данными

Коммуникационный протокол динамического обмена данными (DDE) даёт возможность связи приложений Windows друг с другом. DDE реализует отношения клиент-сервер между двумя одновременно работающими приложениями. Приложение сервера предоставляет данные и принимает запросы от любого другого приложения, заинтересованного в его данных. Запрашивающие приложения называются клиентами. Приложение InTouch может быть одновременно клиентом и сервером.

### Wonderware SuiteLink

Wonderware SuiteLink представляет собой протокол на основе TCP/IP, разработанный конкретно для промышленных применений. SuiteLink обеспечивает целостность данных, высокую пропускную способность и простые диагностические процедуры. Протокол SuiteLink поддерживается операционными системами Microsoft Windows NT 4.0 и выше.

SuiteLink не является заменой для DDE или NetDDE. Каждое соединение между клиентом и сервером зависит от требований вашей сети.

SuiteLink предоставляет следующие возможности:

- Качество и время величины (VTQ) устанавливает временной штамп и индикатор качества на все значения данных, переданные клиентам, осведомлённым о VTQ.
- Большая пропускная способность по данным диагностики, загрузке сервера, потреблению ресурсов компьютера, и сетевому транспорту делаются доступными посредством монитора производительности операционной системы Microsoft Windows.
- Постоянно высокие объёмы данных могут поддерживаться между приложениями, даже если приложения находятся на одном узле или распределяются по большому набору узлов.
- Протоколом сетевого транспорта является TCP/IP, в котором используется стандартный интерфейс компании «Microsoft» Winsock.

## Troubleshooting SuiteLink Communication Problems

Если вы сталкиваетесь с проблемами связи SuiteLink, выполняйте следующие операции:

- Подтвердите, что протокол Microsoft TCP/IP работает на компьютере, на котором установлен человеко-машинный интерфейс InTouch.
- Проверьте, чтобы имя узла компьютера состояло из 15 или меньше знаков.
- Подтвердите, что программа Wonderware SuiteLink действует в качестве службы на компьютере, на котором установлен человеко-машинный интерфейс InTouch.
- SuiteLink автоматически устанавливается в процессе установки InTouch. Служба SuiteLink запускается автоматически. Если услуга SuiteLink останавливается, вы должны запустить её снова.

## Установка имён доступа

Вы должны связывать ссылки на тэги ввода/вывода InTouch или удалённый тэг с именем доступа (Access Name). Имя доступа определяет звено связи с другим источником данных ввода/вывода. Каждое имя доступа задаёт адрес ввода/вывода, состоящий из имени узла, имени приложения и темы.

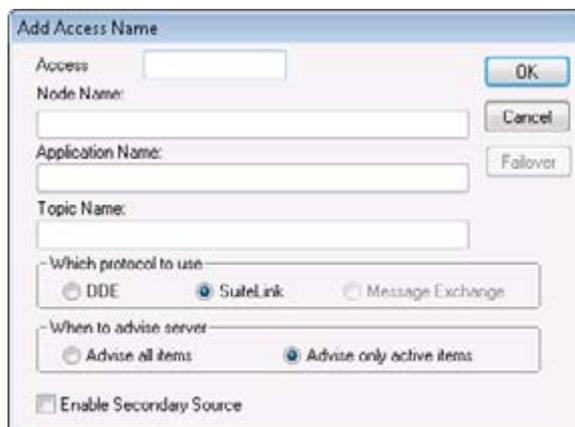
В распределённом приложении ссылки на ввод/вывод могут устанавливаться как глобальные адреса на сетевой Сервер ввода/вывода или локальные адреса на локальный Сервер ввода/вывода.

Приложение InTouchView показывает визуальный интерфейс приложений человеко-машинного интерфейса, разработанный конкретно для использования в среде Сервера приложений ArchestrA. Приложения InTouchView действуют как клиент с Сервером приложений, действующим как сервер, который обеспечивает большую часть функциональности человеко-машинного интерфейса.

Приложения InTouchView предлагают только некоторые из стандартных функций, доступных из полнофункционального Приложения InTouch. Приложения InTouchView не могут присоединяться к источникам ввода/вывода, иным, чем Сервер приложений ArchestrA Galaxy. Вы можете использовать только имя доступа Galaxy, действующее по умолчанию с вашим приложением InTouchView, и не можете создавать имена доступа.

### Для создания имени доступа

- 1 В меню **Special**, щёлкните на **Access Names** (Имена доступа). Появляется диалоговое окно **Access Names**.
- 2 Щёлкните на **Add**. Появляется диалоговое окно **Add Access Name** (Добавить диалоговое окно).



- 3 Установите свойства диалогового окна **Add Access Name**. Выполните следующее:
  - В диалоговом окне **Access** (Доступ), введите с клавиатуры имя, которое идентифицирует данное Имя доступа.
  - Если данные пребывают на сетевом Сервере ввода/вывода, введите имя узла удалённого сервера в диалоговом окне **Node Name** (Имя узла).
  - В диалоговом окне **Application Name** введите фактическое имя программы Сервера ввода/вывода, из которой будут получены данные.

Если источником данных ввода/вывода является Wonderware DAServer, введите имя программы DAServer; не включайте расширение имени файла программы .exe.

- В диалоговом окне **Topic Name** (Имя темы), введите имя темы, к которой вы хотите получить доступ.

Имя темы представляет собой конкретную для приложения подгруппу элементов данных. В случае данных, поступающих из программы Wonderware DAServer, имя темы представляет собой то же самое имя, конфигурированное для темы в программе сервера DAServer. При осуществлении связи с Microsoft Excel, именем темы должно быть имя, присвоенное книге и крупноформатной таблице при её сохранении. Например, [Book1]Sheet1.

- Выберите коммуникационный протокол для осуществления связи с Сервером ввода/вывода.

- Выберите опцию для опроса информации, хранимой на сервере.

Опция	Определение
<b>Advise all items</b> (Сообщить обо всех элементах)	Опрашивает о наличии всех данных, независимо от того, находятся ли они в видимых окнах, алармах, и являются ли регистрируемыми, с прослеживаемой тенденцией или используемыми в пределах скрипта. Выбор данной опции оказывает воздействие на производительность, поэтому использовать её не рекомендуется.
<b>Advise only active items</b> (Сообщить только об активных элементах)	Опрашивает о данных, показанных в видимых окнах, алармах, являющихся регистрируемыми, с прослеживаемой тенденцией или используемыми в пределах скрипта.

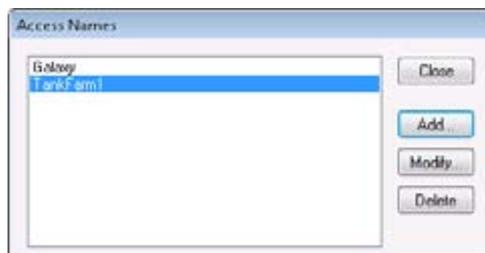
**Примечание:** Скрипт действия кнопки не опрашивается, если он не появляется в видимом окне.

- 4 Выберите **Enable Secondary Source** (Включить вторичный источник), если вы хотите выбрать вторичный резервный сервер. В противном случае перейдите к шагу 5. Если вы выбираете опцию **Enable Secondary Source**, то диалоговое окно **Add Access Name** (Добавить имя доступа) расширяется для того, чтобы показать поля конфигурации для второго источника.



- Завершите опции.
- Щёлкните на **ОК** после присвоения всех значений серверу резервирования вторичного источника.

- Когда вы выполните задание Имени доступа, щёлкните на **ОК**. Диалоговое окно **Access Names** (Имена доступа) появляется снова и показывает, что новое Имя доступа добавляется в список.



- 5 Щёлкните на **Close** (Закреть).

За дополнительной информацией об установке вашего вторичного сервера для переключения обхода отказа, обращайтесь к разделу «Использование функциональных возможностей обхода отказа с именами доступа» в главе 5.

## Удаление имён доступа

Вы можете удалить Имя доступа, которое вы больше не используете. До того, как вы удалите Имя доступа, убедитесь в следующем:

- Никакие тэги не связаны с именем доступа.
- Программа WindowViewer остановлена.

### Для удаления имени доступа

- 1 В меню **Special** (Специальные) щёлкните на **Access Names** (Имена доступа). Появляется диалоговое окно **Access Names**. Перечисляются текущие имена доступа.
- 2 Для удаления Имени доступа выберите его из списка и щёлкните на **Delete** (Удалить). Появляется сообщение, запрашивающее подтверждение того, что Имя доступа следует удалить.
- 3 Щёлкните на **Yes** (Да).
- 4 Щёлкните на **Close** (Закреть) или повторите данную процедуру, если вам потребуется удалить другие определённые Имена доступа.

## Доступ к данным ввода/вывода с тэгами ввода/вывода

Человеко-машинный интерфейс InTouch может посылать и принимать данные из местных или удалённых приложений Windows с тэгами ввода/вывода. Тэг каждого типа ввода/вывода относится к действующему элементу в программе Сервера ввода/вывода. Вы можете определить различные типы тэгов ввода/вывода в Словаре имён тэгов.

### Конфигурирование свойств тэгов ввода/вывода

Вы определяете различные типы тэгов ввода/вывода в Словаре имён тэгов.

#### Задание дискретного тэга ввода/вывода

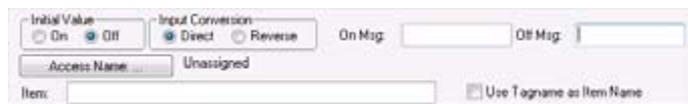
Дискретные тэги ввода/вывода указывают двоичное состояние всех входов и выходов с программируемых контроллеров, компьютеров процессов и данных из сетевых узлов.

Дискретному тэгу ввода/вывода должно быть присвоено первоначальное значение On (Вкл.) или Off (Выкл.). Вы можете также конфигурировать дискретный тэг ввода/вывода для переключения на противоположную величину его двоичного источника. Вы можете задавать сообщения, которые появляются в окне событий алармов, когда процесс, связанный с тэгом, переходит в состояние аларма.

За дополнительной информацией об общей процедуре создания тэга из Словаря имён тэгов, обращайтесь к разделу «Создание новых тэгов» в главе 2.

#### Для определения дискретного тэга ввода/вывода

- 1 Откройте Словарь имён тэгов и присвойте имя новому тэгу.
- 2 Присвойте тэгу тип дискретного ввода/вывода из диалогового окна **Tag Types** (Типы тэгов). Появляется часть диалогового окна **Tagname Dictionary** (Словарь имён тэгов) с указанием подробностей.



- 3 Выполните следующую процедуру:

- Выберите **On** (Вкл.) или **Off** (Выкл.) в качестве первоначального значения, связанного с тэгом.

Человеко-машинный интерфейс InTouch присваивает данную величину тэгу при запуске приложения, но не записывает это первоначальное значение в устройстве ввода/вывода.

- Выберите **Direct** (Вперёд) или **Reverse** (Назад) в качестве входного преобразования, применимого к значению, полученному из тэга удалённого ввода/вывода.

Входное преобразование	Описание
<b>Direct</b> (Прямое)	Входное значение считывается неизменным непосредственно из Сервера ввода/вывода.
<b>Reverse</b> (Обратное)	Переключение дискретного значения ввода/вывода происходит при считывании из программы сервера. Например, если вход ввода/вывода равен 0, то значение автоматически устанавливается на 1.

- Задайте сообщения о состояниях, которые доступны для использования в любой анимационной ссылке или скрипте, независимо от того, имеет ли тэг конфигурированные алармы или нет.

Если вы определяете дискретный предупредительный сигнал, который активен, когда значение тэга равно 1 (Включение, Верно), в поле Value/Limit (Значение/предел) вашего дисплея предупредительных сигналов появляется сообщение, введённое в диалоговое окно On Msg (сообщение о включении). Когда аларм не активен, появляется сообщение, введённое в диалоговое окно Off Msg (сообщение в выключении).

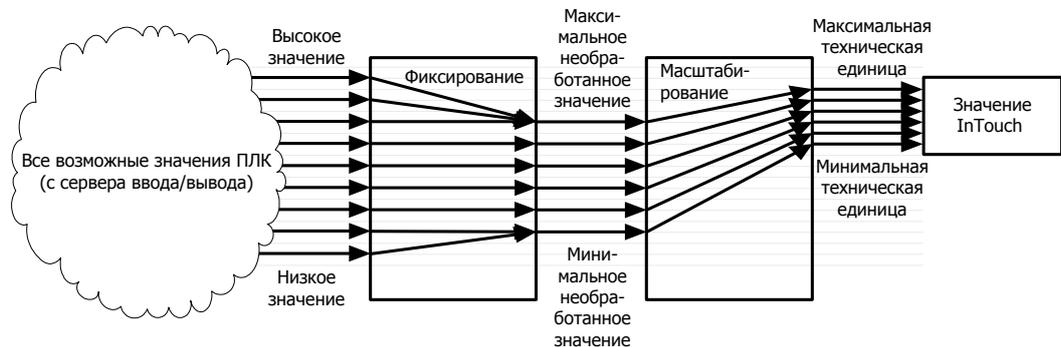
Если вы определяете дискретный предупредительный сигнал, который активен, когда значение тэга равно 0 (Выключение, Ложно), в поле Value/Limit (Значение/предел) вашего дисплея предупредительных сигналов появляется сообщение, введённое в диалоговое окно On Msg (сообщение о включении). Когда аларм не активен, появляется сообщение, введённое в диалоговое окно Off Msg (сообщение в выключении).

- 4 Сохраните изменения, которые вы внесли в тэг.

## Задание целочисленных и вещественных тэгов ввода/вывода

Вы должны присвоить целочисленным и вещественным тэгам набор атрибутов, которые характеризуют числовые данные, посылаемые между приложением InTouch и внешними процессами.

Человеко-машинный интерфейс InTouch нормализует необработанные входные данные с программируемого логического контроллера. На рисунке, приведённом ниже, показан процесс фиксирования необработанных значений ввода/вывода с последующим пересчётом их в технические единицы, которые могут быть показаны из приложения InTouch.



Целочисленные и вещественные тэги ввода/вывода включают в себя атрибуты, которые устанавливают минимальный и максимальный пределы для необработанных входных данных, отправляемых программируемым логическим контроллером. Человеко-машинный интерфейс InTouch фиксирует значения ввода/вывода, которые располагаются ниже или выше диапазона необработанных величин. Фиксирование присваивает заново значения, находящиеся за пределами диапазона, минимальным и максимальным необработанным значениям.

Целочисленные и вещественные тэги ввода/вывода включают в себя атрибуты, которые масштабируют зафиксированные необработанные значения в пределах диапазона технических единиц. Атрибуты минимальной и максимальной технической единицы устанавливают верхнюю и нижнюю границы масштабированных величин.

Когда вы определяете целочисленные и вещественные тэги ввода/вывода, вы задаёте тип преобразования для масштабирования необработанных величин при расчёте технических единиц. Вы можете выбрать линейное преобразование или преобразование методом квадратного корня.

Для линейного масштабирования результат рассчитывается с использованием линейной интерполяции между минимальной и максимальной конечными точками. Алгоритм для линейного масштабирования входа имеет следующий вид:

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

Алгоритм для линейного масштабирования выхода имеет следующий вид:

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

Для масштабирования методом квадратного корня минимальная и максимальная необработанные величины используются для интерполяции. Это полезно для масштабирования входов от нелинейных устройств, таких, как датчики давления. Алгоритм для преобразования входов имеет следующий вид:

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

Алгоритм для преобразования выходов имеет следующий вид:

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

### Определение целочисленных и вещественных тэгов ввода/вывода

- 1 Откройте Словарь имён тэгов и присвойте имя новому тэгу.
- 2 Присвойте целочисленное или вещественное значение ввода/вывода в качестве типа тэга из диалогового окна **Tag Types** (Типы тэгов). Появляется часть диалогового окна **TagName Dictionary** (Словарь имён тэгов) с указанием подробностей.

- 3 Выполните следующие действия:
  - В диалоговом окне **Initial Value** (Первоначальное значение), введите целое или вещественное число, связанное с тэгом при запуске приложения.
  - Приложение не записывает это первоначальное значение во внешний процесс.
  - В диалоговом окне **Min EU** (Минимальная техническая единица) введите минимальную техническую единицу для тэга.
  - В диалоговом окне **Max EU** (Максимальная техническая единица) введите максимальную техническую единицу для тэга.
  - В диалоговом окне **Min Raw** (Минимальное необработанное значение) введите минимальную величину низкой фиксации для необработанных целых или вещественных числовых значений ввода/вывода.
  - В диалоговом окне **Max Raw** (Максимальное необработанное значение) введите максимальную величину высокой фиксации для необработанных целых или вещественных числовых значений ввода/вывода.

- В диалоговом окне Технические единицы введите метку, которая будет использоваться для технических единиц тэга.
- Выберите **Linear** (Линейное) или **Square Root** (Методом квадратного корня) в качестве типа преобразования для масштабирования необработанных величин при расчёте инженерных единиц.

4 Сохраните изменения, которые вы внесли в тэг.

### Задание тэга ввода/вывода сообщения

Вы можете задать варианты тэга сообщения ввода/вывода, которые задают сетевой адрес удалённых процессов. Свойства его сообщения являются такими же, как и у тэга сообщений памяти.

#### Определение значений тэгов памяти и сообщений ввода/вывода

- 1 Откройте Словарь имён тэгов и присвойте имя новому тэгу.
- 2 Выберите **I/O Message** (Сообщение ввода/вывода) в качестве типа тэга из диалогового окна **Tag Types** (Типы тэгов). Появляется часть диалогового окна **Tagname Dictionary** (Словарь имён тэгов) с указанием подробностей.



- 3 В диалоговом окне **Maximum Length** (Максимальная длина) введите максимальное число знаков, допустимое в сообщении тэга. Вы можете вводить сообщения с максимальной длиной в 131 знак.
- 4 В диалоговом окне **Initial Value** (Первоначальное значение) введите текстовую строку, вывод которой на экран вы хотите видеть, когда программа WindowViewer запускает приложение.
- 5 Сохраните изменения, которые вы внесли в тэг.

### Установка параметров доступа ввода/вывода

Вы можете установить атрибуты ввода/вывода тэгов в Словаре имён тэгов. Эти атрибуты идентифицируют внешний источник данных, связанный с тэгом.

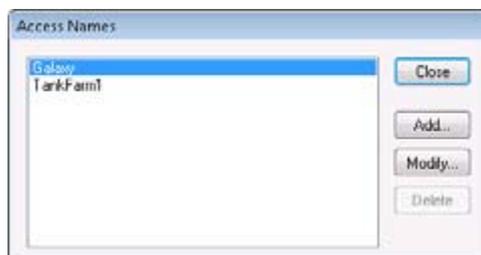
Эти шаги только разъясняют, каким образом задавать атрибуты ввода/вывода из Словаря имён тэгов. За дополнительной информацией о конфигурировании Galaxy и Имен доступа обращайтесь к разделу «Доступ к данным с вводом/выводом» в главе 5.

### Установка атрибутов ввода/вывода

- 1 Присвойте тип тэга ввода/вывода из диалогового окна **Tag Types** (Типы тэгов). Появляется часть диалогового окна **Tagname Dictionary** (Словарь имён тэгов) с указанием подробностей.



- 2 Щёлкните на **Access Name** (Имя доступа) для определения или выбора имени доступа, присвоенного тэгу. Появляется диалоговое окно **Access Names** (Имена доступа), показывающее список текущих имён доступа, признанных человеко-машинным интерфейсом InTouch (Galaxy – имя доступа для соединения ArchestrA, действующее по умолчанию).



- 3 Добавьте имя доступа или примите значение, действующее по умолчанию.
- 4 Выберите информационную точку в программе сервера, где тэг ввода/вывода будет считывать и записывать данные.
  - Для считывания и записи данных в точку данных в программе сервера введите Название элемента в диалоговом окне **Item** (Элемент). Например, для считывания значения из регистра в ПЛК введите идентификатор регистра в качестве Имени элемента. Например:  
 Для использования регистра 1 программируемого логического контроллера Алена-Брэдли введите в качестве Имени элемента R1 в диалоговом окне **Item** (Элемент).  
 Для использования самого младшего бита регистра 1 программируемого логического контроллера Аллена-Брэдли в качестве Имени элемента введите R1:0 в диалоговом окне **Item** (Элемент).
  - Для использования тэга в качестве элемента, выберите в качестве **Use Tagname** (Использовать имя тэга).

## Восстановление информации о тэгах ввода/вывода в реальном времени

Функции скриптов возвращают имена узла, приложения и темы, заданных в определении Имени доступа. На рисунок, приведённом ниже, показано диалоговое окно Add Access Name (Добавить имя доступа) с опциями для задания узла, приложения и темы для Имени доступа.

### Функция IOGetNode()

Функция IOGetNode() возвращает адрес узла, определённый для конкретного тэга Имени доступа, связанного с функцией в пределах скрипта.

#### Категория

Различные

#### Синтаксис

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

#### Аргумент

*AccessName (Имя доступа)*

Существующее Имя доступа, для которого должна быть возвращена информация об узле.

#### Примечания

Вы можете задавать Имя доступа в качестве буквенной строки или строковой величины, предусмотренной другими тэгами InTouch или функциями.

#### Пример

Нижеследующий пример возвращает информацию об узле для Имени доступа ModbusPLC1 для тэга NodeName.

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

### Функция GetNodeName()

Функция GetNodeName() возвращает имя местного узла как строку.

#### Категория

Система

#### Синтаксис

```
GetNodeName ("TagName" , NodeNum) ;
```

### Аргументы

#### *TagName (Имя тэга)*

Тэг сообщения InTouch, которому присвоено имя узла в качестве его значения.

#### *NodeNum (Номер узла)*

Целочисленная величина, которая задаёт максимальную длину строки, присвоенную тэгу сообщения.

### Примечания

Функция `GetNodeName()` возвращает имя местного узла и присваивает его аргументу `TagName`. Аргумент `NodeNum` устанавливает максимальную длину тэга сообщения.

### Пример

В данном примере имя местного узла присваивается тэгу `MyNode`. Если имя узла равно `Master`, то скрипт генерирует сообщение, который указывает, что узлом является первичный компьютер.

```
GetNodeName ("MyNode" , 131) ;  
  If MyNode == "Master" THEN  
    MessageTag = "This is the Primary Computer!" ;  
  ENDIF ;
```

## Функция `IOGetApplication()`

Функция скрипта `IOGetApplication()` возвращает имя приложения, определённое для конкретного Имени доступа, тэгу, присвоенному в качестве аргумента функции.

### Категория

Разнообразная

### Синтаксис

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

### Аргумент

#### *AccessName (Имя доступа)*

Существующее Имя доступа, в котором приложение определено.

### Примечания

Вы можете задавать Имя доступа как буквенную строку или как строковую величину, предусмотренную другими тэгами InTouch или функциями.

### Пример

Пример возвращает имя приложения, заданное для Имени доступа `ModbusPLC1` тэгу `AppName`.

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

## IOGetTopic() Function

Функция скрипта IOGetTopic() возвращает имя темы, определённое для конкретного Имени доступа, тэгу, связанному с функцией в пределах скрипта.

### Категория

Разнообразная

### Синтаксис

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

### Аргумент

*AccessName* (Имя доступа)

Имя доступа, имя темы которого возвращается.

### Примечания

Имя доступа может быть задано как буквенная строка, или оно может представлять собой строковую величину, предоставленную другими тэгами сообщений или функциями InTouch.

### Пример

Данный пример возвращает информацию по темам для Имени доступа ModbusPLC1 тэгу TopicName.

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

## Динамически изменяющиеся ссылки на тэги ввода/вывода во время выполнения

В человеко-машинном интерфейсе InTouch используются динамические ссылки на измерительные точки просмотра, значения которых требуются только временно, такие, как диагностические приложения. Использование динамической справочной адресации позволяет вам обращаться к многочисленным источникам данных с единым тэгом.

Вы можете использовать несколько методов для динамических ссылок на несколько источников данных с единым тэгом:

- Присвойте различные характеристики Имени доступа с точечным полем .Reference тэга ввода/вывода.
- Используйте функцию скрипта IOSetItem() для установки точечного поля .Reference тэга ввода/вывода.
- Используйте функцию скрипта IOSetAccessName() для изменения характеристики Имени доступа во время выполнения программы

## Точечное поле .Reference

Вы можете осуществить Динамическую ссылочную адресацию, присваивая действующую ссылку точечному полю .Reference тэга ввода/вывода. Вы можете использовать точечное поле .Reference для динамической смены источника данных путём изменения характеристик Имени доступа, присвоенного тэгу ввода/вывода.

Синтаксис точечного поля .Reference:

*tag.Reference*="accessname.item"   Изменяет имя доступа и элемент, присвоенный тэгу

*tag.Reference*="[.Jitem]"           Изменяет элемент, присвоенный тэгу ввода/вывода.

*tag.Reference*="accessname."       Изменяет имя доступа тэга ввода/вывода.

*tag.Reference*=""                   Отключает тэг ввода/вывода.

Тэг ввода/вывода каждого типа имеет точечное поле .ReferenceComplete. Значение указанного дискретного точечного поля указывает, отражается ли элемент, запрашиваемый в точечном поле .Reference, в точечном поле .Value.

Поле .ReferenceComplete устанавливается на ложное значение (0) при запуске приложения в программе WindowViewer. При подтверждении того, что точечное поле .Value обновляется источником, заданным в точечном поле .Reference, значение .ReferenceComplete устанавливается на истинное значение (1). Если точечное поле .Reference меняется, точечное поле .ReferenceComplete автоматически устанавливается на ложное значение (0), и затем меняется на истинное значение (1) при обновлении нового значения.

## IOSetItem() Function

Вы можете осуществить Динамическую ссылочную адресацию, используя функцию IOSetItem() в пределах скрипта. IOSetItem() включает в себя аргументы для изменения значения, присвоенного точечному полю .Reference тэга ввода/вывода во время выполнения программы.

### Категория

Разнообразная

### Синтаксис

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

### Аргументы

*Tag* (Тэг)

Любой тэг ввода/вывода InTouch, заключённый в кавычки.

*AccessName* (Имя доступа)

Имя доступа, присвоенное тэгу ввода/вывода.

*Item (Элемент)*

Элемент, присвоенный тэгу ввода/вывода.

Аргументы «Tag», «AccessName» и «Item» могут быть заданы в виде буквенных строк, или они могут представлять собой строковые величины, предоставленные другими тэгами или функциями InTouch.

**Примеры**

В нижеследующем примере точечное поле .Reference тэг PumpInP1 изменяется на точку для Имени доступа к программе Excel и элемент R1C1.

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

или

```
Number = 1;
```

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

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

Если пустая строка ("" ) задаётся как для Имени доступа, так и для значений элементов, то тэг отключается. Например, тэг PumpInP2 отключается строкой:

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

Если нуль задаётся только для элемента, то текущее значение элемента тэга сохраняется, и значение его Имени доступа обновляется. Например, нижеследующая функция изменяет Имя доступа для тэга PumpInP3 на excel2, не оказывая влияния на текущее значение его Элемента (Item):

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

Аналогичным образом, если нулевая строка задаётся только для Имени доступа, то текущее значение Имени доступа тэга сохраняется, и значение его элемента обновляется. Нижеследующий пример меняет Элемент (Item) для тэга PumpInP4 на R1C2 не оказывая влияния на текущее значение Имени доступа:

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

**Функция IOSetAccessName()**

Вы можете осуществить Динамическую ссылочную адресацию, используя функцию IOSetAccessName() в пределах скрипта. IOSetAccessName() изменяет приложение или характеристики Имени доступа темы тэга ввода/вывода во время выполнения программы.

Примечание: При обработке функции IOSetAccessName() имеет место задержка, когда существующее преобразование заканчивается, а новое преобразование начинается. В этот период, любые попытки записи в память (РОКЕ) или в новую тему теряются.

**Категория**

Разнообразная

**Синтаксис:**

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

**Аргументы***AccessName (Имя доступа)*

Существующее Имя доступа, которому должны быть присвоены новые значения AppName и Topic Name. Фактическая строка или тэг сообщения.

*NodeName (Имя узла)*

Новое Имя узла, которое должно быть присвоено. Фактическая строка или тэг сообщения.

*AppName (Имя приложения)*

Новое имя приложения, которое должно быть присвоено. Фактическая строка или тэг сообщения.

*TopicName (Имя темы)*

Новое имя темы, которое должно быть присвоено. Фактическая строка или тэг сообщения.

**Примечания**

Значения, присвоенные аргументам «AccessName», «AppName» и «TopicName» могут быть присвоены как буквенные строки или строковые значения, предоставленные другими тэгами или функциями InTouch.

---

**Примечание:** Если, при создании имён доступа в программе WindowMaker, Имя доступа относится к типу SuiteLink, то человеко-машинный интерфейс InTouch предотвращает доступ Имен доступа к тому же самому узлу, приложению и теме. Не позволяйте функции IOSetAccessName() перенаправлять Имена доступа к дубликатам во время выполнения программы. Использование функции IOSetAccessName() в реальное время даёт возможность перенаправления Имен доступа типа SuiteLink на дубликаты тем. Перенаправленное имя доступа работать не будет

---

**Примеры**

Имя доступа MyAccess1 может быть изменено для указания на приложение Excel и тему [Book1]Sheet1 без воздействия на текущее Имя узла (NodeName) с использованием следующей функции скрипта:

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

Если для Темы задаётся пустая строка, текущее значение Приложения Имени доступа обновляется, а значение его Темы сохраняется.

Например, нижеследующий скрипт заменяет Имя приложения для Имени доступа MyAccess2 на EXCEL без воздействия на текущее значение темы (Topic):

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

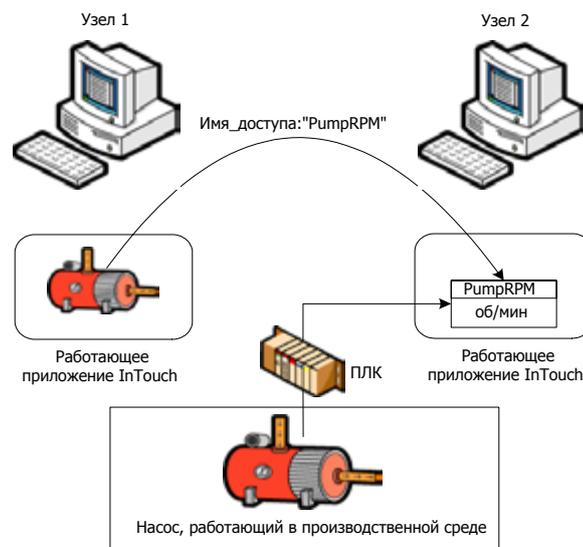
Если пустая строка задана только для Имени приложения, текущее значение Темы обновляется, а значение его Приложения сохраняется. Например, нижеследующий скрипт заменяет Тему для тэга MyAccess3 на [Book2]Sheet1, не оказывая влияния на текущее значение его Имени приложения:

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

Данный пример может использоваться, когда требуется резервирование программируемого логического контроллера.

## Преобразование тэгов в удалённые ссылки

Вы можете создавать распределённые Приложения InTouch на основании архитектуры клиента-сервера. Приложения клиента могут работать на одном сетевом узле, на котором используются тэги, определённые на других удалённых узлах. На рисунке, приведённом ниже, показано приложение InTouch, запущенное на Узле 1, делающее удалённую ссылку на тэг PumpRPM на Узле 2.



В данном примере вы можете восстановить значение тэга PumpRPM с Узла 2 двумя путями:

- Создать тэг типа ввода/вывода в Словаре имён тэгов Узла 2, в котором используется Узел 2 в качестве имени узла в Имени доступа, связанном с тэгом ввода/вывода.

- Используйте прямую удалённую ссылку на тэг PumpRPM. Например, ПЛК1: «PumpRPM».

В окне или скрипте QuickScript вы можете сделать ссылку на удалённый тэг, добавляя Имя доступа в качестве префикса к имени удалённого тэга в следующей форме:

```
access_name:"tag_name"
```

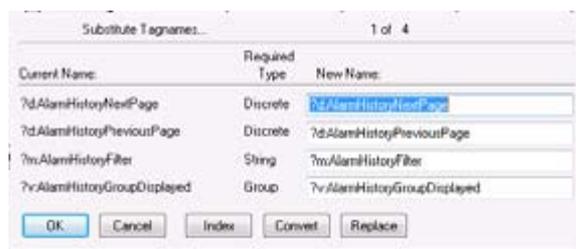
При использовании ссылок на удалённый тэг и импорте окна или QuickScript, преобразуйте тэги указателя места заполнения в ссылки на удалённые тэги. Нет необходимости определения тэгов в вашем местном Словаре имён тэгов.

Вы можете использовать несколько методов для преобразования местных тэгов в ссылке на удалённые тэги:

- Добавьте ссылку на удалённый тэг.
- Преобразуйте тэги указателя места заполнения, связанные с импортируемым окном.
- Запустите браузер тэгов и откройте Словарь имён тэгов для выбора ссылки на удалённый тэг.

#### Ручное преобразование тэгов в ссылки на удалённые тэги

- 1 Откройте окно приложения в программе WindowMaker.
- 2 Выберите объект, связанный с местным тэгом, который вы хотите заменить ссылкой на удалённый тэг.
- 3 В меню **Special** (Специальное), щёлкните на **Substitute Tags** (Заменить тэги). Появляется диалоговое окно **Substitute Tagnames**. Показывается список тэгов, связанный с объектом.
- 4 В диалоговом окне **New Name** (Новое имя) выберите каждый из тэгов, который вы хотите преобразовать в удалённую ссылку.

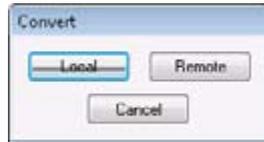


- 5 Добавьте Имя доступа в качестве префикса к имени тэга в форме: access\_name:tag\_name
- 6 Щёлкните на ОК.

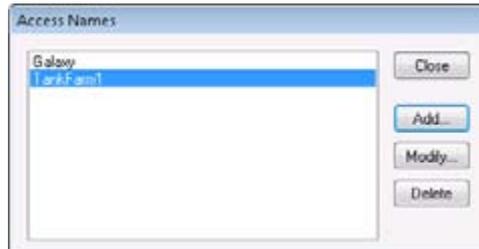
#### Преобразование импортируемых тэгов окна в удалённые ссылки

- 1 Откройте импортируемое окно и выберите все объекты.
- 2 В меню **Special** (Специальное), щёлкните на **Substitute Tags** (Заменить тэги). Появляется диалоговое окно **Substitute Tagnames** (Изменить имена тэгов).

- 3 Щёлкните на **Convert** (Преобразовать). Появляется диалоговое окно **Convert**.

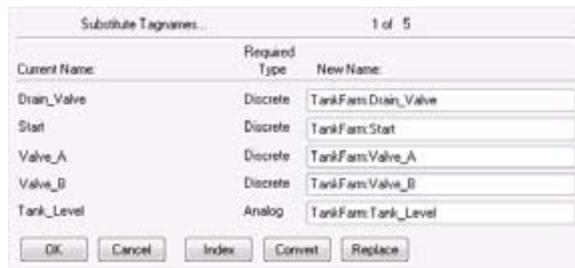


- 4 Щёлкните на **Remote** (Удалённый). Появляется диалоговое окно **Access Names** (Имена доступа). Показываются все Имена доступа, определённые в вашем местном приложении InTouch.



- 5 Выберите Имя доступа из списка. Для проверки правильности конфигурирования Имени доступа щёлкните на **Modify** (Изменить).
- 6 Если вы не имеете Имя доступа, определённое в текущий момент, которое указывает на источник тэга, щёлкните на **Add** (Добавить) и определите его. Имя доступа должно включать в себя имя удалённого узла, на котором располагается приложение.

Щёлкните на **Close** (Закрыть). Все тэги, перечисленные в диалоговом окне **Substitute Tags** (Заменить тэги), автоматически преобразуются в ссылки на удалённые тэги с Именем доступа, добавленным к имени тэга.



- 7 Щёлкните на **OK**.

#### Выбор ссылки на удалённый тэг в браузере тэгов

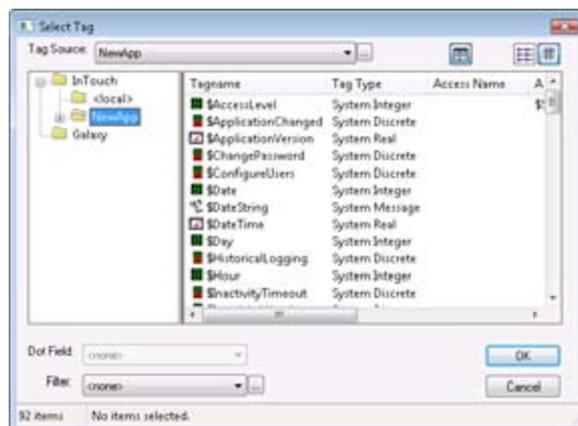
- 1 Выберите объекты, связанные с местным тэгом, который вы хотите преобразовать в ссылку на удалённый тэг.
- 2 В меню **Special** (Специальное) щёлкните на **Substitute Tags** (Заменить тэги). Появляется диалоговое окно **Substitute Tagnames** (Изменить имена тэгов), показывающее выбранные тэги.
- 3 Удалите имя тэга в диалоговом окне **New Name** (Новое имя), которое вы хотите заменить ссылкой на удалённый тэг.

4 Дважды щёлкните на диалоговом окне **New Name** (Новое имя). Появляется диалоговое окно **Select Tag** (Выбрать тэг) со списком тэгов, связанных с приложением.

5 Выберите удалённый тэг, используя Вид дерева.



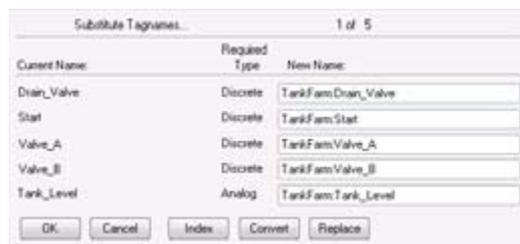
a Щёлкните на значке **Tree** (Дерево) для показания иерархического списка всех местных и удалённых Имен доступа на левой панели.



b Выберите папку Имен доступа для вывода присвоенных в ней тэгов в правой панели.

c Выберите удалённый тэг, который вы хотите использовать в качестве удалённой ссылки.

d Щёлкните на **OK**. Появляется диалоговое окно **Substitute Tagnames** (Изменить имена тэгов) с выбранным именем удалённого тэга в диалоговом окне **New Name** (Новое имя).



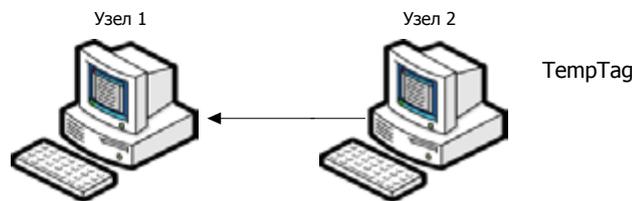
6 Щёлкните на **OK** для закрытия диалогового окна и свяжите удалённый тэг с выбранными объектами.

7 Повторите эти шаги для каждого тэга, который вы хотите связать с удалёнными ссылками.

## Доступ к данным ввода/вывода по удалённым ссылкам

Человеко-машинный интерфейс InTouch даёт возможность получения истинной архитектуры клиент-сервер для заводских приложений, связанных с автоматизацией производства. Вы можете разрабатывать приложения клиента без использования каких-либо тэгов из местного Словаря имён тэгов, расположенного на том же самом узле, на котором расположено работающее приложение InTouch. Вы можете запустить на одном узле приложение, которое будет использовать тэги с удалённого узла путём использования ссылок на удалённые тэги.

На рисунке, приведённом ниже, показан простой пример, в котором временный тэг TempTag определён локально на Узле2:



В данном примере приложение InTouch, работающее на Узле1, может извлекать значение временного тэга на (TempTag) Узле2 двумя методами:

- Создайте тэг типа ввода/вывода в Словаре имён тэгов Узла1, в котором используется Узел 2 в качестве имени узла в Имени доступа, связанном с тэгом ввода/вывода.
- Используйте удалённую ссылку непосредственно на TempTag. Например, Node2: «TempTag».

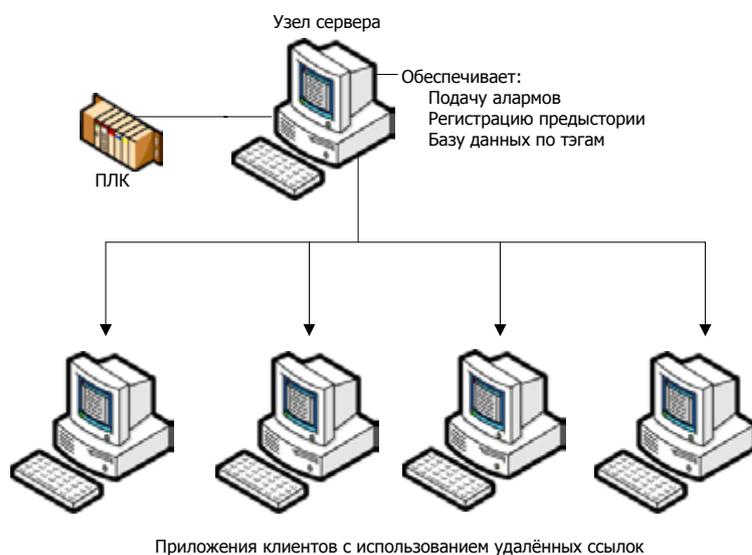
Когда вы хотите сослаться непосредственно на удалённый тэг в любом другом приложении, требуется только AccessName:item. Вам не требуется определять удалённый тэг в вашем местном Словаре имён тэгов. Удалённые ссылки могут также иметь доступ к данным из любого источника данных ввода/вывода, такого, как Wonderware DAServer или Microsoft Excel.

Вы можете также сделать удалённую ссылку на супертэги (SuperTags). Действующий синтаксис для ссылки удалённого тэга на SuperTag является следующим:

**Access\_name:Parent\_Instance\ChildMember\SubMember.**

За дополнительной информацией об удалённых ссылках на супертэги обращайтесь к разделу «Ссылки на компоненты супертэгов» в главе 7.

Когда вы используете ссылку на удалённый тэг для импорта окна или «QuickScript», вы просто преобразуете тэги указатель места заполнения в удалённые ссылки на тэги. Вам не приходится определять ссылки в вашем местном Словаре имён тэгов. Доступ к удалённым ссылкам имеется из любого приложения Wonderware в сети.



## Перенаправление удалённых ссылок во время выполнения программы

Вы можете перенаправить ссылки на объект `ArchestrA` или удалённые ссылки на тэги `InTouch` во время выполнения программы со скриптом. Вы можете переключать экземпляры объектов для графического символа на основании определённых выполняемых условий или непосредственно действием оператора.

### Функция `IOSetRemoteReferences()`

Вы можете использовать функцию скрипта `IOSetRemoteReferences()` для перенаправления ссылок объектов `ArchestrA` или удалённые ссылки на тэги во время работы приложения `InTouch`. Функция `IOSetRemoteReferences()` обнаруживает все удалённые ссылки, которые соответствуют заданным строкам и изменяет эти ссылки в соответствии с заданными величинами аргументов. Вы можете создать скрипт, который запускает функцию для перенаправления ссылок на основании выполняемых условий, или действием оператора.

#### Категория

Изменение данных

#### Синтаксис

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

#### Аргументы

*BaseAccess (Доступ к базе)*

Данный аргумент строки, заданный оригинальным конфигурированным Именем доступа, которому должны соответствовать ссылки.

*NewAccess (Новый доступ)*

Новое имя доступа. Новое имя доступа применяется ко всем ссылкам, в которых оригинальное имя доступа соответствует строке, обеспеченной базовым доступом (`BaseAccess`), и для которых оригинальное Имя элемента соответствует значению `MatchString`, если оно задано.

*MatchString (Соответствие строке)*

Строка для соответствия в изначально конфигурированном Имени элемента в ссылках. Если значение `MatchString` представляет собой пустую строку, она рассматривается как согласование для любого Имени элемента.

*SubstituteString (Замещающая строка)*

Строка для замещения применительно к оригинальному Имени элемента. Строка заменяет значение `MatchString` для создания нового активного Имени элемента для ссылок. Если `SubstituteString` представляет собой пустую строку, то никакой замены не делается.

### *Mode (Режим)*

Определяет способ, используемый для сравнения изначально конфигурированного Имени элемента со значением `MatchString`. Согласование всегда начинается с начала Имени элемента. Значение режима (`Mode`), равное 0, устанавливает, что согласование должно быть для полного Имени элемента или до точки (.) в пределах имени. Значение режима, равное 1, устанавливает, что частичное согласование допускается, если следующий знак не является точкой.

### **Примечания**

Функция `IOSetRemoteReferences()` не проверяет достоверность нового тэга Имени доступа перед изменением ссылки на объект.

- Функция `IOSetRemoteReferences()` изменяет только удалённые ссылки. Функция перенаправляет те ссылки, в которых оригинальное конфигурированное Имя доступа согласуется с заданным значением аргумента `BaseAccess`, и оригинальное Имя элемента согласуется со значением `MatchString`.
- Единичное обращение к функции `IOSetRemoteReferences()` оказывает воздействие на все удалённые ссылки в активных окнах, которые находятся в памяти, в которой оригинальные конфигурированные строки имён соответствуют значениям, присвоенным аргументам `BaseAccess` и `MatchString`.
- Если вы не присваиваете значение аргументу `BaseAccess`, то `IOSetRemoteReferences()` не перенаправляет никакие удалённые ссылки.
- Если аргумент `MatchString` является пустым, то `IOSetRemoteReferences()` перенаправляет все удалённые ссылки, в которых оригинальное Имя доступа согласовывается значение, присвоенное аргументу `BaseAccess`.
- Когда аргумент `Mode` устанавливается на 0, замена в Имени элемента осуществляется только для полных имён объектов (или тэгов), или полных имён свойств (или точечных полей). Значение аргумента `MatchString` должно соответствовать полному оригинальному Имени элемента или до знака, за которым идёт точка.
- Когда аргумент `Mode` устанавливается на 1, частичная замена элемента допускается в том случае, когда строка элементов начинается со строки элементов согласования. То есть, `MatchString` должно соответствовать некоторой части первоначальной строки элементов, но указанная часть должна начинаться с начала строки элементов. За последним знаком в согласующей строке не обязательно должна идти точка.
- Оригинальные конфигурированные имена для удалённой ссылки остаются неизменными. Последующие обращения к `IOSetRemoteReferences()` не обязательно должны признавать текущее активное имя. Обращения к `IOSetRemoteReferences()` могут делаться в любом порядке.
- Если вы хотите, чтобы два или более окон обращались к одной

удалённой ссылке, та удалённая ссылка действует аналогично тэгу ввода/вывода. Когда вы перенаправляете её, все окна выглядят одинаково. Не используйте единичное имя для ссылки на двух отдельных адресатов в одно и то же время.

---

**Примечание:** Одновременная замена многих ссылок, например в скрипте Window OnShow, может потребовать некоторого времени до того, как все ссылки будут разрешены.

---

### Примеры

Нижеследующий пример перенаправляет все удалённые ссылки на имя элемента pump001 Имени доступа Galaxy, если оригинальное имя элемента точно согласуется с pumpX.

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

Нижеследующий пример согласует изменения Имени доступа Galaxy TagServer1 если имя элемента в точности согласуется с pumpX. Кроме того, имя элемента меняется на p2.

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

Нижеследующий пример изменяет Имя доступа TagServer1 на TagServer2, когда имя элемента – pumpX. Имя элемента также меняется на backpump3.

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

Нижеследующий пример меняет имя элемента Tank (Резервуар) Имени доступа TagServer1 на Plant (Установка).

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

Нижеследующий пример не перенаправляет никаких удалённых ссылок , поскольку аргументу BaseAccess не присвоено никакого значения.

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

### Восстановление ссылок

Если аргумент SubstituteString является пустым и не имеет присвоенного значения, то IOSetRemoteReferences() восстанавливает активное Имя доступа на оригинальное базовое Имя доступа.

Если аргумент MatchString является пустым и не имеет присвоенного значения, то IOSetRemoteReferences() восстанавливает активное Имя элемента на оригинальное имя элемента.

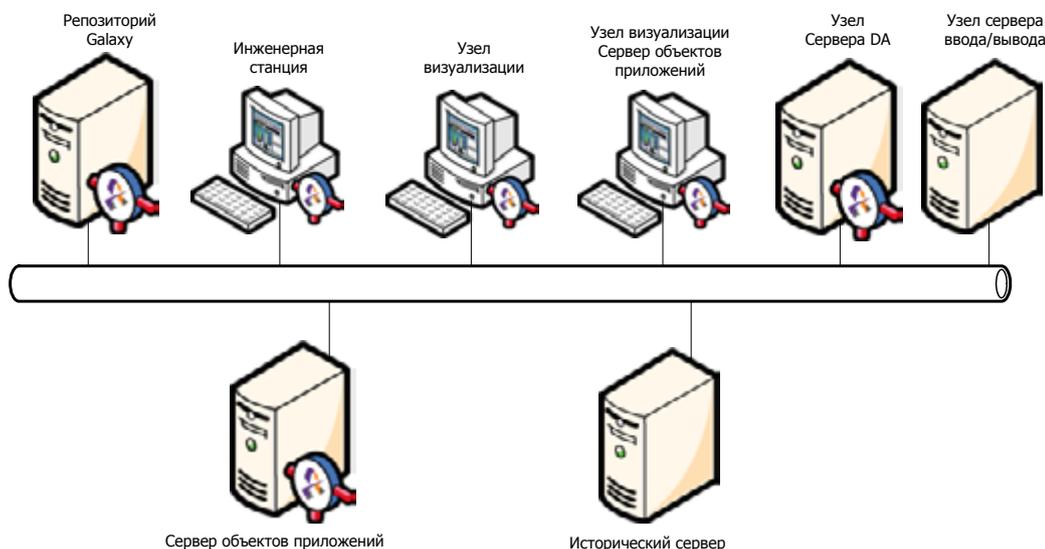
**Примечание:** Даже если `SubstituteString` не является пустым, если `MatchString` является пустым, то имя элемента восстанавливается на оригинальное имя элемента. Вставка текста в начало имени не допускается. Например, запуск скрипта `IOSetRemoteReferences("Access 1", "", "", "Valve", 0)`; не добавляет строку Valve (Клапан) в начало или в конец всех оригинальных Имен элементов.

Если `SubstituteString` является пустым и не имеет присвоенного значения, то `IOSetRemoteReferences()` восстанавливает активное Имя элемента на Оригинальное имя элемента. Использование непустого аргумента `MatchString` с пустым аргументом `SubstituteString` позволяет вам выбрать подмножество удалённых ссылок на доступе к базе и восстановить их на их оригинальные Имена элементов.

## Оценка данных сервера приложений из InTouch

Функция «ArchestrA» предоставляет множество общих услуг и основополагающую архитектуру для комплекта продуктов Wonderware. Вы можете выбрать из массива продуктов построение систем автоматизации установки и информации с использованием модульных компонентов «ArchestrA».

Сервер приложений предоставляет множество услуг для построения приложений автоматизации производственной установки. Услуги сервера приложений распределены по множеству узлов в пределах системы.



Обычно человеко-машинный интерфейс InTouch работает с Сервером приложений для предоставления визуального интерфейса для приложения, с которым операторы взаимодействуют для управления процессом установки.

## Тэги InTouch в Сервере приложений

Вы можете использовать тэги InTouch для взаимодействия с атрибутами объектов Сервера приложений с для передачи данных между приложением InTouch и репозиторием данных Сервера приложений.

Поддержка протокола связи человеко-машинного интерфейса InTouch включает в себя Обмен сообщениями (Message Exchange). Когда WindowViewer запускает приложение InTouch, Message Exchange рассматривает WindowViewer как анонимный механизм.

Данная анонимность означает, что приложение InTouch не имеет атрибутов, доступ к которым могут иметь другие клиенты Обмен сообщениями. WindowViewer не конфигурируется, не управляется и не рассматривается как AutomationObject (Объект автоматизации) в рамках Сервера приложения Galaxy. Человеко-машинный интерфейс InTouch использует только Обмен сообщениями для подписи под теми активными элементами, которые доступны со стороны Сервера приложений.

Вы можете использовать браузер тэгов InTouch для выбора объекта Сервера приложений в качестве источника тэгов для удалённых тэгов и осуществлять просмотр через пространство имён Galaxy. Атрибут объекта Сервера приложений или свойство атрибута могут быть отображены на удалённый тэг InTouch.

За дополнительной информацией об использовании объектов Серверов приложений в качестве источников удалённых тэгов, обращайтесь к разделу «Конфигурирование человеко-машинного интерфейса InTouch для использования Galaxy в качестве источника удалённого тэга» в главе 5.

## Создание источников тэгов в пределах InTouch

Предварительно сконфигурированное Имя доступа, называемое Galaxy, доступно в приложении WindowMaker для доступа к Обмену сообщениями (Message Exchange). Имя доступа Galaxy уместно только для человеко-машинного интерфейса InTouch в приложении ArchestrA. Для Имени доступа Galaxy не существует никаких установок, конфигурируемых пользователем.

Для создания источника тэга Сервера приложений, доступного вашему приложению InTouch, выберите предварительно определённое Имя доступа Galaxy. Затем введите информацию присоединения для словаря:

- Имя пользователя
- Пароль
- Имя Galaxy
- Расположение Galaxy, которое представляет собой имя узла сети, где расположен репозиторий Galaxy.

В диалоговом окне InTouch Attribute Browser перечислены все объекты в пределах мишени Galaxy. Вы можете расширить объект для того, чтобы видеть содержащиеся в нём объекты или атрибуты, доступные во время выполнения. В диалоговом окне Attribute Browser не показываются те атрибуты, которые начинаются с “\_” (отмеченные как скрытые) или любой атрибут типа QualifiedStruct.

Отображения столбцов имеют следующий вид:

- Tagname (Имя тэга) = ObjectTagname.Attributename
- Datatype (Тип данных) = Обратитесь к разделу «Отображение типов данных Сервера приложений в типах данных InTouch» в главе 5 за информацией об отображении данных между человеко-машинный интерфейс InTouch и Сервером приложений
- Access Name (Имя доступа) = “”
- AlarmGroup (Группа алармов) = объект автоматизации участка
- Comment (Комментарий) = Имя шаблона, из которого выводится пример

## Ограничения браузера сервера применений

Нижеследующие ограничения применяются при рассмотрении объектов Сервера приложений с использованием диалогового окна Attribute Browser (Браузер атрибутов) InTouch:

- Можно рассмотреть только атрибуты, видимые во время выполнения в единичном пространстве имён Galaxy. Это включает в себя возможность переключения между именем тэга объекта и его иерархическим именем (HierarchicalName).

Атрибут объекта может быть выбран из диалогового окна Attribute Browser, если он удовлетворяет следующим требованиям:

- Является видимым во время выполнения программы;
- Из проверяемого в настоящее время объекта автоматизации;
- Имя атрибута не имеет знака “\_”, следующего за “.”
- Диалоговое окно Attribute Browser показывает только типы данных атрибута объекта Сервера приложений, поддерживаемые человеко-машинным интерфейсом InTouch. За дополнительной информацией о поддерживаемых типах данных обращайтесь к Разделу «Отображение типов данных Сервера приложений в типах данных InTouch» в главе 5.
- Диалоговое окно Attribute Browser InTouch не показывает никаких атрибутов, которые привели бы к появлению Имени элемента InTouch, превышающего максимальную длину в 95 знаков.
- Имя, выводимое на дисплей из Galaxy, представляет собой Объект «TagName.AttributeName». Длина имени должна включать в себя их все и максимальную длину имени точечного поля. Максимально допустимая длина имени атрибута для выводимого на дисплей имени атрибута составляет 80 знаков.
- Элементы массивов объекта автоматизации могут быть выведены на дисплей или извлечены в человеко-машинном интерфейсе InTouch с использованием “TagName.AttributeName[index]” в качестве ссылки. Используйте показатель -1 для показа или извлечения всех значений элементов массива.
- Вы можете выбрать свойство атрибута объекта с тэгом Browser (Браузер). По умолчанию, свойство Value выбирается при выборе атрибута.

## Типы данных источников тэгов

Нижеприведённый список представляет собой типы данных источников тэгов, которые могут быть показаны в пределах браузера тэгов InTouch.

- Булев (соответствует дискретному типу данных InTouch)
- Целочисленный
- С плавающей точкой (соответствует вещественному типу данных InTouch)
- Двойной (преобразуется в вещественный тип данных InTouch)
- Строка (соответствует типу данных сообщений InTouch)
- Время (преобразуется в тип данных сообщений InTouch)
- Истекшее время (преобразуется в вещественный тип данных InTouch в секундном выражении)
- Ссылка (Преобразуется в тип данных сообщений InTouch)
- Нумерация (Преобразуется в тип данных сообщений InTouch)

## Специальные расширения в объектах Сервера приложений

Браузер тэгов WindowMaker и клиент Обмена сообщениями в программе WindowViewer добавляют и признают специальные расширения для каждого атрибута объекта Сервера приложений. Эти расширения обеспечивают доступ к информации, которая в противном случае была бы недоступной для человеко-машинного интерфейса InTouch.

Эти специальные расширения являются необязательными, и использование их приложением InTouch не требуется. Однако приложениям, которые обрабатывают информацию о статусе и качестве, часто требуется использовать указанные расширения.

Эти элементы расширяют пространство имён атрибутов для включения дополнительных свойств, которые WindowViewer может открыть для скриптов и окон приложений. Например, ссылка на "TIC101.PV.#ReadSts" обеспечивает доступ к информации MxStatus для подписки на TIC101.PV. Эта информация является очень полезной для отображения расширенной информации, которая раскрывается обменом сообщениями.

Эти свойства не существуют как атрибуты объектов в Сервере приложений, как поименованные элементы. Свойства представляют собой расширения со стороны клиента, предоставляемые в слое абстракций клиента, которые делают атрибуты объекта видимыми для человеко-машинного интерфейса InTouch. В нижеследующей таблице описываются расширения атрибутов для человеко-машинного интерфейса InTouch:

Расширение атрибута	Тип данных	Назначение
Нет	Приведённые	Расширение, действующее по умолчанию. Означает, что никакого расширения не предусмотрено. Данный элемент считывается/записывается с типом данных значения, приведённых как применимый к InTouch. Информация, которую не удаётся считать/записать, может быть получена, если клиент подписывается на элементы #ReadSts или #WriteSts, описанные ниже. Пример: "Pump1.PV".
.#VString только для данных с плавающей точкой/ двойных атрибутов: .#VString1 .#VString2 .#VString3 .#VString4	Строка (считывание/ запись)	Устанавливает подписку на ссылку, которая имеет в качестве суффикса ".#VString". Это – базовая ссылка. Возвращает текущее значение базовой ссылки как строку, когда считывания и записи работают правильно. Если UserGetAttribute возвращает недействующий статус, данный элемент возвращает сокращённую строку описания статуса, основанную на MxStatus вместо величины. Сокращённые строки описания статуса имеют вид: «?Pending» – ожидание «?Warning» – предупреждение «?Comms» – ошибка связи «?Config» – ошибка конфигурации «?Oper» – операционная ошибка «?Security» – ошибка защиты «?Software» – ошибка программного обеспечения «?Other» – другая ошибка

Для строки .#VString, если статус является действующим, но количество – плохим, данный элемент возвращает строку описания качества, доступную из Обмена сообщениями вместо значения. Значение возвращается как строка только в том случае, когда качество для UserGetAttribute является хорошим, а статус – действующим, или когда качество является хорошим, а статус – неопределённым. Это может потребовать приведения, если тип данных, возвращаемых Обменом сообщений, не является строкой. Когда качество или статус являются неопределёнными, значение показывает «?» в качестве суффикса. Например, «3.27?» или «True?».

## Отображение типа данных Сервера приложений на тип данных InTouch

Сервер приложений включает в себя некоторые атрибуты и типы данных, которые не отображаются непосредственно в четыре первичных типа данных, поддерживаемых тэгами InTouch.

В нижеследующей таблице показано, каким образом слой абстракций клиента отображает типы данных для операций считывания и записи. В ней также показываються типы данных, которые словарь Galaxy представляет человеку-машинному интерфейсу InTouch.

Тип данных свойства атрибута	Тип данных InTouch	Примечания
Float (с плавающей точкой)	Вещественный – 32 бита	Пройти.
Double (с двойной точностью)	Вещественный – 32 бита	Если данные являются данными с двойной точностью IEEE NAN, то преобразовать в данные с плавающей точкой IEEE NAN. Если при этом происходит переполнение, то установить Quality (качество) на Bad (плохое) и пропустить данные IEEE NAN. Если дробная часть с двойной точностью является меньшей дробью, чем самая малая дробь с плавающей точкой 1.17549E-38, интерпретировать её как величину с плавающей точкой 0,0 и установить Quality на Good (Хорошее).
Boolean (Булев)	Дискретный	Ложный = 0, Истинный = 1.
Integer (Целочисленный)	Целочисленный – 32 бита	Пройти
String (Строка) (всегда Unicode)	Сообщение – MBCS (кодировка набором символов, каждый из которых представлен однобайтовым или двухбайтовым значением)	Обрезать строку, если она слишком для InTouch, и установить качество на uncertain (неопределённое). Сохранить оба байта каждого из знаков Unicode.
Time (Время)	Сообщение – MBCS	Форматировать, как соответствующую строку для местной специфики. Используйте MxValue для преобразования строки.
ElapsedTime (Истекшее время)	Вещественный	Пропустить как число секунд с плавающей точкой. MxValue поддерживает приведение к данному типу.
MxDataType	Сообщение – MBCS	Передать строку.

Тип данных свойства атрибута	Тип данных InTouch	Примечания
MxSecurityClassification	Сообщение – MBCS	Передать строку.
MxQuality	Сообщение – MBCS	Передать строку.
MxReference	Сообщение – MBCS	Передать эталонную строку только как Unicode.
MxCategorizedStatus	Сообщение – MBCS	Передать строку.
MxQualifiedStruct	Не поддерживается	Не поддерживается.
MxQualifiedEnum	Сообщение – MBCS	Передать строку Enum. Доступ к целочисленной порядковой величине может быть получен приложениями со ссылкой на #EnumOrdinal. Например, «Pump1.PV.#EnumOrdinal».
Array of Strings (Массив строк)	Сообщение – MBCS (только для чтения)	Поместить каждый элемент массива строку, разделённую запятыми, следующим образом: «String1, String2, String3» до максимального предела значения строки InTouch. Если это урезано, то соответствующее качество, посылаемое на человеко-машинный интерфейс InTouch, является неопределённым. Вы не можете записывать полный массив строк, но вы можете записывать отдельные элементы массива.
All arrays (Все массивы)	Целочисленный, вещественный, сообщение или дискретный	Поддерживает только подписку на единичный элемент массива. В этом случае применимыми являются преобразования, описанные выше. В противном случае возврат представляет собой пустую строку с плохим качеством (Bad).
MxInternationalizedText	Сообщение	Это оценивается как строковый тип во время выполнения.

## Поведение считывания/записи атрибутов сервера приложений

Когда система осуществляет запись в Атрибут объекта автоматизации, он записывает состояние, изначально установленное на «?Pending».

Когда запись будет завершена, строка `#WriteSts` обновляется с результатом записи. Если запись осуществляется успешно, то величина `#WriteSts` устанавливается на пустую строку. Если запись возвращает ошибку и является незаконченной, элемент `#WriteSts` продолжает показывать самое последнее состояние записи, даже если обновления подписки продолжают появляться в считываниях.

Вы можете также использовать строки с `#VString1` по `#VString4` для преобразования величии с плавающей точкой или величин двойной точности в строковый формат. Число `N` указывает число десятичных разрядов, которые должны быть возвращены. Например, «3.1234» представляет собой строку для `#VString4`. Вы можете использовать элемент `#VString` без номера для округления величины с плавающей точкой или с двойной точностью до целочисленной величины и вернуть его как строковую величину.

Расширение атрибута	Тип данных	Назначение
<code>#EnumOrdinal</code> (порядковая нумерация)	Целочисленный (считывание/ запись)	Содержит считываемую в текущий момент порядковую величину для атрибутов типа <code>Qualified Enum</code> (квалифицированная нумерация). Это – способ возврата для нумераций целочисленных величин вместо возврата строки.
<code>#ReadSts</code> (считать строку)	Строка (только для чтения)	Содержит текущее состояние считывания элемента, на который была произведена подписка и с которым связана строка. Она имеет следующий вид: «TIC101.PV.#ReadSts». Это предусмотрено Обменом сообщений как строка. Она может принимать одно из следующих значений: «?Config» – ошибка конфигурации «?Comms» – ошибка связи «?Oper» – операционная ошибка «?Pending» – ожидание «?Warning» – предупреждение «?Security» – ошибка защиты «?Software» – ошибка программного обеспечения «?Other» – другая ошибка

**Примечание:** При отсутствии подписки на связанный элемент (например, TIC101.PV) возвращаемая строка будет пустой.

Расширение атрибута	Тип данных	Назначение
.#WriteSts	Строка (только для чтения)	Содержит последний статус записи элемента, с которым связана строка, например, Pump1.Cmd.#WriteSts. Это предусмотрено Обменом сообщений как строка. Если строка является пустой, то последняя запись в элемент является успешной. В противном случае #WriteSts может принимать одно из следующих значений: «?Config» – ошибка конфигурации «?Comms» – ошибка связи «?Oper» – операционная ошибка «?Pending» – ожидание «?Warning» – предупреждение «?Security» – ошибка защиты «?Software» – ошибка программного обеспечения «?Other» – другая ошибка

**Примечание:** При отсутствии подписки на связанный элемент (например, TIC101.PV) возвращаемая строка будет пустой.

## Конфигурирование человеко-машинного интерфейса InTouch для использования Galaxy в качестве удалённого источника тэга

Вы можете использовать браузер тэгов InTouch для выбора объекта Сервер приложений в качестве источника тэгов и осуществлять выбор базы данных Galaxy. Атрибуты Сервера приложений или свойства атрибутов могут быть отображены на удалённых тэгах InTouch.

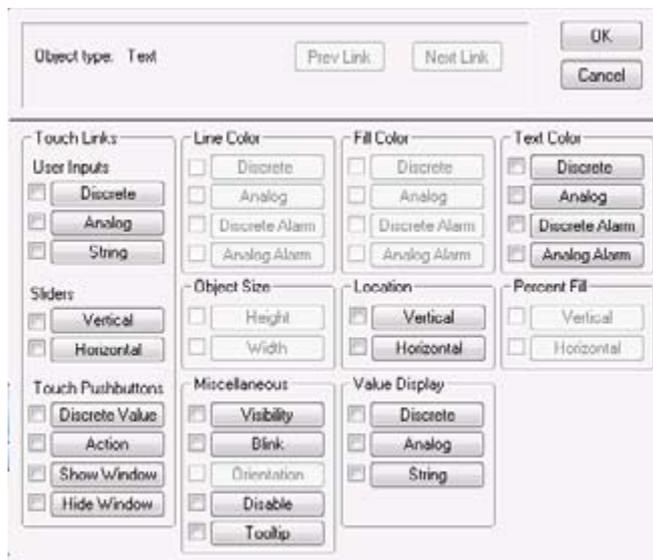
Когда приложение InTouch предоставляет визуальный интерфейс для Сервера приложений, вы должны установить программу самозагрузки Сервера приложений в той же узел, что и приложение InTouch, а также развернуть объект WinPlatform на узле InTouch.

Для прокрутки пространства имён InTouch, вам может также потребоваться установка интегрированной инструментальной системы проектирования и разработки (ИСП) ArchestrA.

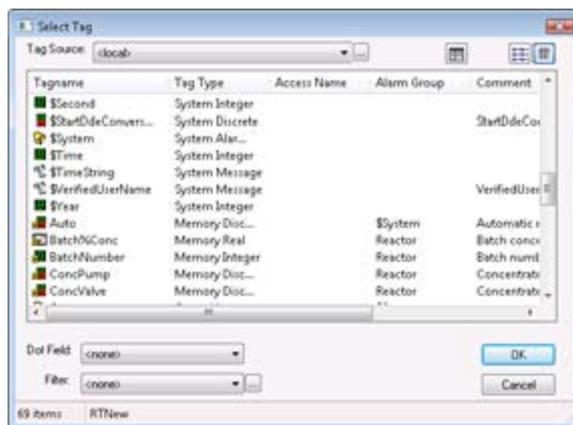
Человеко-машинный интерфейс InTouch использует функциональность обмена сообщениями Платформы для рассмотрения пространства имён Galaxy и обеспечения лучшей передачи данных.

### Конфигурирование In Touch для использования Galaxy в качестве удалённого источника тэгов

- 1 Откройте окно приложения в WindowMaker.
- 2 Дважды щёлкните на текстовом объекте. Появляется диалоговое окно **Animation Links** (Анимационные ссылки).

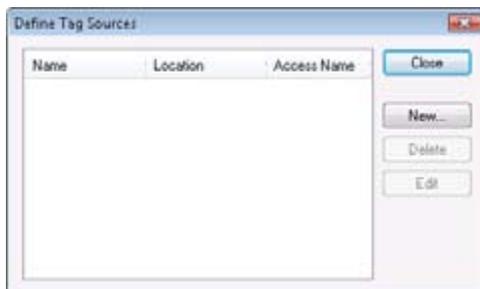


- 3 На участке значений **Display** (Дисплей), щёлкните на **Analog** (Аналоговый). Появляется диалоговое окно для ввода выражения.
- 4 Удалите всякое выражение, расположенное в пределах окна выражений.
- 5 Дважды щёлкните в окне **Expression** (Выражение). Появляется диалоговое окно **Select Tag** (Выбрать тэг).

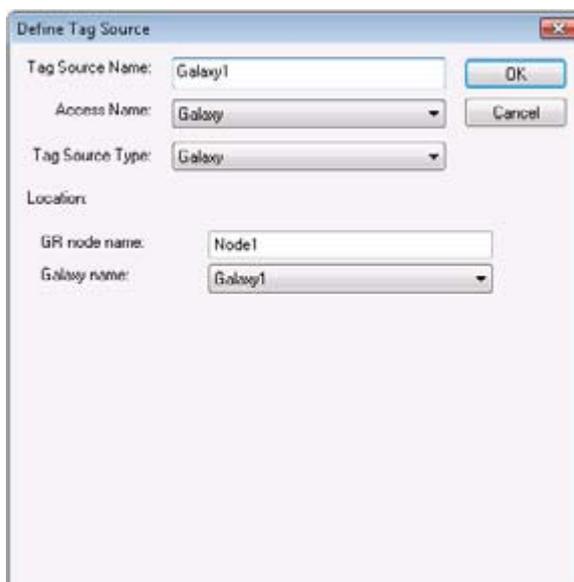




- 6 Щёлкните на кнопке справа от окна **Tag Source** (Источник тэга). Появляется диалоговое окно **Define Tag Sources**.

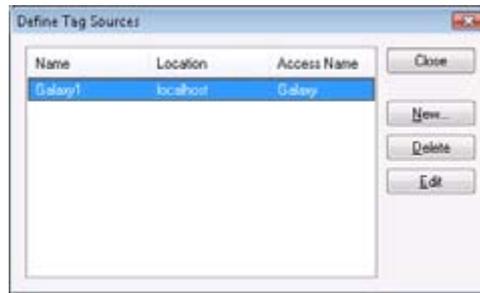


- 7 Щёлкните на **New** (Новый) для вывода диалогового окна **Define Tag Source** (Определить источник тэга).

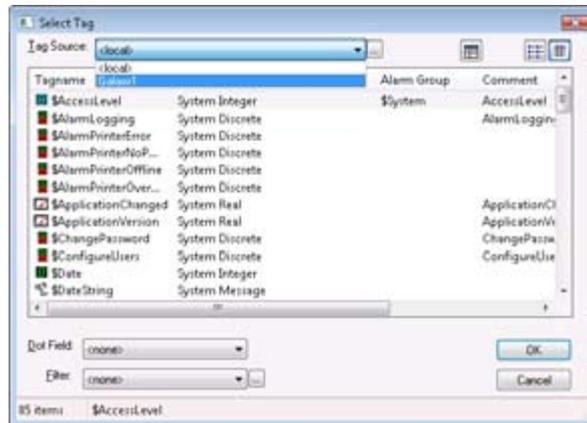


- 8 Введите значения для малых окон диалогового окна **Define Tag Source** (Определить источник тэга). Действуйте следующим образом:
- В окне **Tag Source Name** (Имя источника тэга), введите имя вашего удалённого источника тэгов Galaxy.
  - В окне **Access Name** (Имя доступа) выберите из списка Galaxy.
  - В окне **Tag Source Type** выберите из списка Galaxy.
  - На участке **Location** (Расположение) наберите название Узла репозитория Galaxy и выберите из списка Galaxy.

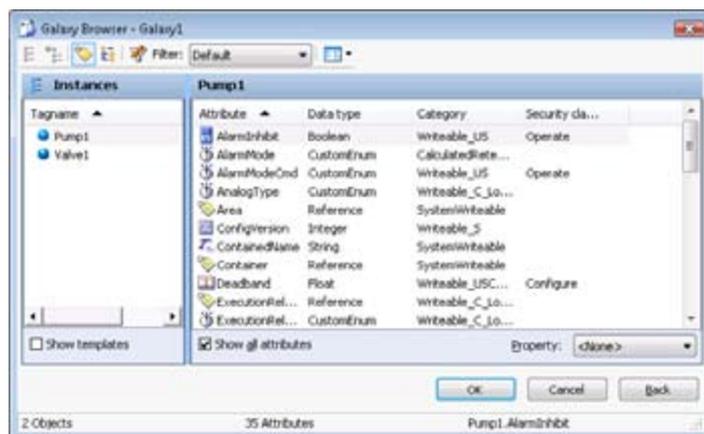
- е Щёлкните на **ОК**. В диалоговом окне **Define Tag Sources** (Определить источники тэгов) показывается удалённый источник тэга, который вы определили в списке.



- 9 Щёлкните на **Close** (Закреть) для закрытия диалогового окна **Define Tag Sources** (Определить источники тэгов). В диалоговом окне **Select Tag** (Выбрать тэг) показывается новый источник тэга из списка окна **Tag Source** (Источник тэга).

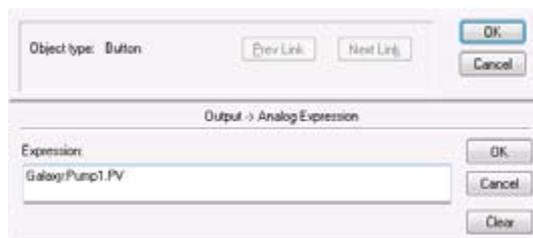


- 10 Выберите новый источник тэга, который вы создали, из окна **Tag Source** (Источник тэгов). Открывается диалоговое окно **Attribute Browser** (Браузер атрибутов) со списком тэгов в левой панели.



- 11 Выберите тэг из левой панели диалогового окна **Attribute Browser** (Браузер атрибутов). Правая панель диалогового окна **Attribute Browser** обновляется с включением атрибутов выбранного тэга.

- 12 Щёлкните на атрибуте, который вы хотите использовать, после чего щёлкните на **ОК**. Появляется диалоговое окно **Output -> Analog Expression** (Выход -> Аналоговое выражение) с выражением в окне **Expression** (Выражение).



- 13 Подтвердите правильность строкового выражения. В выражении используется следующая форма:
- Galaxy:tag\_name.attribute\_name**
- Пример:
- Galaxy:PumpRPM.PV**
- 14 Щёлкните на **ОК** для закрытия **Output > Analog Expression** (Выход -> Аналоговое выражение) диалоговое окно.
- 15 Щёлкните на **ОК** в диалоговом окне **Properties Configuration** (Конфигурация свойств).
- 16 Настройте конфигурацию остальных связей объектов по мере необходимости.
- 17 Щёлкните на **Runtime!** (Время выполнения). Текстовый объект показывает значение для конфигурированного атрибута тэга.

## Просмотр отметок времени и информации по качеству для тэга ввода/вывода

Человеко-машинный интерфейс InTouch устанавливает индикаторы значения, времени и качества (VTQ) на всех значениях данных, поставляемых клиентам для информирования их о значении, времени и качестве. В человеко-машинном интерфейсе InTouch используется набор точечных полей в качестве индикаторов качества данных, которые являются полезными для целей поиска и устранения неисправностей.

- Точечное поле `.Value` содержит значение заданного тэга. Оно также является точечным полем, используемым по умолчанию в каждом тэге InTouch. Если не задано никакого другого точечного поля, подразумевается точечное поле `.Value`.
- Набор точечных полей `Time` (Время) представляет собой отметки времени, указывающие последнее время, когда тэг был обновлён.
- Точечные поля `Quality` (Качество) показывают надёжность данных значений, присвоенных тэгу ввода/вывода.

## Присмотр информации отметок времени для тэга ввода/вывода

Набор точечных полей Time (Время) присваивается тэгам в следующем формате:

*Tag\_name.Time\_Dotfield*

### **.TimeDate Dotfield**

Точечное поле .TimeDate показывает полное число дней, которые прошли между 1 января 1970 года и датой последнего обновления тэга с использованием значения с сервера ввода/вывода.

#### **Категория**

Тэг

#### **Использование**

**Tag\_name.TimeDate**

#### **Параметр**

*Tag\_name*

Любой дискретный, целочисленный, косвенный аналоговый тэг или косвенный тэг сообщения.

#### **Тип данных**

Целочисленный (только для чтения).

#### **Смотрите также**

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

### **Точечное поле .TimeDateString**

Точечное поле .TimeDateString показывает дату в том же самом формате, который установлен в настройке «Язык и региональные стандарты» операционной системы «Windows».

#### **Категория**

Тэг

Любой дискретный, целочисленный, косвенный аналоговый тэг или косвенный тэг сообщения.

#### **Использование**

**Tag.TimeDateString**

#### **Параметр**

*Tag.*

Любой дискретный, целочисленный, косвенный аналоговый тэг или косвенный тэг сообщения.

**Тип данных**

Сообщение (только для чтения).

**Смотрите также**

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

**Поле `.TimeDateTime`**

Точечное поле `.TimeDateTime` показывает дробное число дней, которые прошли в период между 1 января 1970 года и временем последнего обновления тэга с использованием значения с сервера ввода/вывода.

**Категория**

Тэг

**Использование**

**Тэг** `.TimeDateTime`

**Параметр**

*Tag*

Любой дискретный, целочисленный, косвенный аналоговый тэг или косвенный тэг сообщения.

**Тип данных**

Вещественный (только для чтения).

**Смотрите также**

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

**Точечное поле `.TimeDay`**

Точечное поле `.TimeDay` показывает число дней в пределах месяца, которое прошло с момента последнего обновления тэга с использованием значения с сервера ввода/вывода.

**Категория**

Тэг

**Использование**

**Тэг** `.TimeDay`

**Параметр**

*Tag*

Любой дискретный, целочисленный, косвенный аналоговый тэг или косвенный тэг сообщения.

**Тип данных**

Целочисленный (только для чтения).

**Действующие значения**

Значения могут находиться в диапазоне от 1 до 31.

**Смотрите также**

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

**Точечное поле `.TimeHour`**

Точечное поле `.TimeHour` показывает число часов в пределах дня, которые прошли с момента последнего обновления тэга с использованием значения с сервера ввода/вывода.

**Категория**

Тэг

**Использование**

**Тэг** `.TimeHour`

**Параметр**

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

**Тип данных**

Целочисленный (только для чтения).

**Действующие значения**

Значения могут находиться в диапазоне от 0 до 23.

**Смотрите также**

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

**Точечное поле `.TimeMinute`**

Точечное поле `.TimeMinute` показывает число истекших минут, которые прошли с момента последнего обновления тэга с использованием значения с сервера ввода/вывода.

**Категория**

Тэг

**Использование**

**Тэг** `.TimeMinute`

**Параметр**

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

**Тип данных**

Целочисленный (только для чтения).

**Действующие значения**

Значения могут находиться в диапазоне от 0 до 59.

**Смотрите также**

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

**Точечное поле `.TimeMonth`**

Точечное поле `.TimeMonth` показывает порядковый номер месяца (1-12), когда тэг был последний раз обновлён с использованием значения с сервера ввода/вывода.

**Категория**

Тэг

**Использование**

**Тэг** `.TimeMonth`

**Параметр**

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

**Тип данных**

Целочисленный (только для чтения).

**Действующие значения**

Значения могут находиться в диапазоне от 1-12.

**Смотрите также**

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

**Точечное поле `.TimeMsec`**

Точечное поле `.TimeMsec` показывает число миллисекунд, которые прошли с того момента, как тэг был последний раз обновлён с использованием значения с сервера ввода/вывода.

**Категория**

Тэг

**Использование**

**Тэг** `.TimeMsec`

**Параметр**

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

**Тип данных**

Целочисленный (только для чтения).

**Действующие значения**

Значения могут находиться в диапазоне от 0 до 999.

**Смотрите также**

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

**Точечное поле `.TimeSecond`**

Точечное поле `.TimeSecond` показывает число секунд, которые прошли с тех пор, как тэг был последний раз обновлён с использованием значения с сервера ввода/вывода.

**Категория**

Тэг

**Использование**

**Tag**.`TimeSecond`

**Параметр**

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

**Тип данных**

Целочисленный (только для чтения).

**Действующие значения**

Значения могут находиться в диапазоне от 0 до 59.

**Смотрите также**

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

**Точечное поле `.TimeTime`**

Точечное поле `.TimeTime` показывает число миллисекунд, которые прошли с полуночи, и когда тэг был последний раз обновлён с использованием значения с сервера ввода/вывода.

**Категория**

Тэг

**Использование**

**Tag**.`TimeTime`

**Параметр**

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

**Тип данных**

Целочисленный (только для чтения).

**Действующие значения**

Значения могут находиться в диапазоне от 0 до 86399999.

**Смотрите также**

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

**Точечное поле `.TimeTimeString`**

Точечное поле `.TimeTimeString` показывает день и время, когда тэг был обновлён с использованием значения с сервера ввода/вывода.

**Категория**

Тэг

**Использование**

**Tag** `.TimeTimeString`

**Параметр**

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

**Тип данных**

Сообщение (только для чтения).

**Смотрите также**

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

**Точечное поле `.TimeYear`**

Точечное поле `.TimeYear` показывает в четырёхзначном формате год, когда тэг был обновлён с использованием значения с сервера ввода/вывода.

**Категория**

Тэг

**Использование**

**Tag** `.TimeYear`

**Параметр**

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

**Тип данных**

Целочисленный (только для чтения).

**Действующие значения**

Любой год, записанный в виде четырёхзначного числа.

**Смотрите также**

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

## Просмотр информации о качестве для тэга ввода/вывода

Вы можете использовать набор точечных полей качества для обеспечения целостности данных, посылаемых между Сервером ввода/вывода и вашими приложениями InTouch. Точечные поля качества представляют состояние качества значения данных элемента. Этот атрибут качества позволяет совершенно легко контролировать целостность данных InTouch, посылаемых между узлами сети.

Стандарт качества данных Wonderware основывается на стандарте OLE (связывание и встраивание объектов), предложенном для управления технологическими процессами (ОПС), который, в свою очередь, основывается на требованиях к качеству данных Fieldbus.

**Формат данных качества**

Сервер ввода/вывода может сообщать о шести взаимно исключаяющих состояниях качества данных, посылаемых клиенту, путём присвоения значений набору точечных полей .Quality (Качество) человеко-машинного интерфейса InTouch. Восемь младших битов (Самый младший байт) точечных полей качества в настоящее время определяются в форме трёх битовых полей; Качество (Q), Подстатус (S) и Предел (L) в следующем формате: QQSSSSL. Когда приложение клиента не может осуществлять связь с сервером, значение точечного поля .Quality (Качество) равно 0.

Состояния качества данных, сообщаемые Сервером ввода/вывода с точечными полями .Quality показаны в следующей таблице:

Состояния качества	Десятичное значение	Шестнадцатеричное значение	Старший байт XXXXXXXX	Младший байт QSSSSLL	Качество	Подстатус качества	Предел
Хорошее	192	0x00C0	00000000	11000000	Q=3	S=0	L=0
Фиксированное высокое (за пределами диапазона)	86	0x0056	00000000	01010110	Q=1	S=5	L=2
Фиксированное низкое (за пределами диапазона)	85	0x0055	00000000	01010101	Q=1	S=5	L=1
Невозможно преобразовать	64	0x0040	00000000	01000000	Q=1	S=0	L=0
Отказ связи	24	0x0018	00000000	00011000	Q=0	S=6	L=0
Невозможно получить доступ к точке	4	0x0004	00000000	00000100	Q=0	S=1	L=0

### О точечных полях качества данных

Точечные поля .Quality (Качество) указывают качество значений данных по состоянию на последний момент, когда данные были приняты. Протоколы SuiteLink и DDE посылают клиентам (например WindowViewer) обновлённое качество, только когда изменение данных обеспечивается Серверами ввода/вывода. Следовательно, вы наблюдаете изменение качества только тогда, когда принимается новое значение данных. Некоторые Серверы ввода/вывода могут посылать текущие значения данных с обновлённым качеством, когда качество, связанное с данными, изменяется.

Было бы невозможно непосредственно обратиться к качеству значения элемента сервера, используя протоколы SuiteLink и DDE. Для этого Сервер ввода/вывода должен непосредственно поддерживать Item.Quality (Качество элемента). Без этой поддержки, элемент не продолжает рассмотрения, и значение точечного поля .Quality никогда не меняется от 0.

Имитатор Сервера ввода/вывода TestProt не поддерживает непосредственно Item.Quality. Имитатор не высылает новые значения данных, когда вы изменяете качество, используя команду меню Quality (Качество).

Если вы хотите наблюдать качество данных для элемента ввода/вывода, а Сервер ввода/вывода не поддерживает непосредственно адресацию Item.Quality, определите тэг ввода/вывода InTouch для рассмотрения элемента сервера, а затем контролируйте

точечное поле `.Quality` тэга `InTouch`. Если вы превышаете ваш предел тэга, рассмотрите возможность использования функции `IOSetRemoteReferences()` в скрипте для динамической регулировки точек ввода/вывода.

Протоколы `SuiteLink` и `DDE` не интерпретируют состояние присоединения или другие изменения в состоянии Сервера ввода/вывода по мере того, как элементы качества будут посылаться клиенту. В результате качество элемента не обязательно указывает состояние текущего сервера данных или текущее состояние соединения между сервером и клиентом. Процесс Сервера ввода/вывода может остановиться, и значение поля `.Quality` не изменится. Если звено связи потеряно, то значение поля `.Quality` может не изменяться.

Используйте элементы внутреннего состояния `DDE` или `SuiteLink` для контроля присоединения Сервера ввода/вывода.

### Точечное поле `.Quality`

Точечное поле `.Quality` показывает числовую оценку качества данных, предоставляемых Сервером ввода/вывода.

#### Категория

Тэг

#### Использование

**Тэг** `.Quality`

#### Параметр

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

#### Тип данных

Целочисленный (только для чтения).

#### Действующие значения

Значения могут находиться в диапазоне от 0 до 255.

#### Смотрите также

`.QualityLimit`, `.QualityStatus`, `.QualitySubstatus`

#### Пример

```
IF IOTag.Quality <> 192 THEN
    LogMessage ("Качество этих данных не является
    хорошим.");
ENDIF;
```

## Точечное поле `.QualityLimit`

Точечное поле `.QualityLimit` показывает предел качества значения данных, предоставляемых присоединённым Сервером ввода/вывода.

### Категория

Тэг

### Использование

**Тэг** `.QualityLimit`

### Параметр

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

### Тип данных

Целочисленный (только для чтения).

### Действующие значения

0 = Не ограничено

1 = Низкое ограничение

2 = Высокое ограничение

3 = Постоянно

### Смотрите также

`.Quality`

## Точечное поле `.QualityLimitString`

Точечное поле `.QualityLimitString` показывает строку предела качества значения данных, предоставленных присоединённым Сервером ввода/вывода.

### Категория

Тэг

### Использование

**Тэг** `.QualityLimitString`

### Параметр

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

### Тип данных

Сообщение (только для чтения).

### Смотрите также

`.Quality`, `.QualityLimit`

### Точечное поле `.QualityStatus`

Точечное поле `.QualityStatus` показывает состояние качества значения данных, предоставляемых Сервером ввода/вывода, в целочисленной форме.

#### Категория

Тэг

#### Использование

**Tag** `.QualityStatus`

#### Параметр

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

#### Тип данных

Целочисленный (только для чтения).

#### Действующие значения (SSSS)

0 = Плохое

1 = Неопределённое

2 = Хорошее

#### Смотрите также

`.Quality`, `.QualitySubStatus`

### Точечное поле `.QualityStatusString`

Точечное поле `.QualityStatusString` показывает строку состояния качества значения данных, предоставляемых Сервером ввода/вывода.

#### Категория

Тэг

#### Использование

**Tag** `.QualityStatusString`

#### Параметр

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

#### Тип данных

Сообщение (только для чтения).

#### Смотрите также

`.QualityStatus`, `.QualitySubStatus`, `.Quality`

## Точечное поле `.QualitySubstatus`

Точечное поле `.QualitySubstatus` показывает подстатус качества значения данных, предоставляемых Сервером ввода/вывода.

### Категория

Тэг

### Использование

**Tag** `.QualitySubstatus`

### Параметр

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

### Тип данных

Целочисленный (только для чтения).

### Действующие значения (SSSS) и (QQ)

Подстатус (SSSS) для ПЛОХОГО качества (QQ=0).

- 0 = Неконкретно
- 1 = Ошибка конфигурации
- 2 = Не присоединено
- 3 = Отказ устройства
- 4 = Отказ датчика
- 5 = Последнее известное значение
- 6 = Отказ связи
- 7 = За пределами обслуживания

Подстатус (SSSS) для НЕОПРЕДЕЛЁННОГО качества (QQ=1).

- 0 = Неконкретно
- 1 = Последнее значение, пригодное для использования
- 4 = Датчик не точен
- 5 = Технческие единицы превышены
- 6 = Ниже нормального

Подстатус (SSSS) для ХОРОШЕГО качества (QQ=2).

- 0 = Неконкретно
- 6 = Местная отмена

### Смотрите также

`.QualityStatus`, `.QualitySubStatus`, `.Quality`

### Точечное поле `.QualitySubstatusString`

Точечное поле `.QualitySubstatusString` показывает строку подстатуса качества значения данных, предоставляемых Сервером ввода/вывода.

#### Категория

Тэг

#### Использование

**Tag** `.QualitySubstatusString`

#### Параметр

*Tag*

Любой дискретный, целочисленный, вещественный, косвенный аналоговый тэг или тэг типа сообщения.

#### Тип данных

Сообщение (только для чтения).

#### Смотрите также

`.QualityStatus`, `.QualitySubstatus`, `.Quality`

## Инициализация и сброс соединений ввода/вывода во время выполнения

Программа `WindowViewer` инициализирует все диалоги ввода/вывода, когда она запускает приложение `InTouch`. Вы можете также перезапустить вручную диалоги ввода/вывода во время работы вашего приложения `InTouch`. Вы можете инициализировать и сбрасывать соединения ввода/вывода во время выполнения программы с командами или скриптами `WindowViewer`.

Вы можете также произвести повторную инициализацию на основании величин, действующих по умолчанию, с использованием диалогового окна `Properties` (Свойства) программы `WindowViewer`. Если вы выбираете данную опцию, используются установки, действующие по умолчанию, и текущий установки игнорируются при повторной инициализации Имен доступа. Для повторной инициализации диалогов ввода-вывода Именем доступа, вы должны определить приложение `InTouch` с Именами доступа.

## Повторная инициализация соединения ввода/вывода с командами

Меню **Special** (Специальное) в программе WindowViewer включает в себя набор команд для повторной инициализации всех диалогов ввода/вывода или выбора конкретного диалога ввода/вывода.

Вы можете произвести повторную инициализацию Имен доступа, используя действующие по умолчанию установки InTouch. Используя повторную инициализацию, действующую по умолчанию, Имя доступа игнорирует текущие значения, присвоенные имени узла, имени приложения и теме. Имя доступа инициализируется заново с оригинальными установками Имен доступа.

### Для повторной инициализации всех Имен доступа по время выполнения программы

- 1 В меню **Special** (Специальное) щёлкните на **Reinitialize I/O** (Произвести повторную инициализацию ввода/вывода).
- 2 Щёлкните на **Reinitialize All** (Произвести повторную инициализацию всего). Будет произведена повторная инициализация всех Имен доступа.

### Для повторной инициализации выбранных Имен доступа во время выполнения

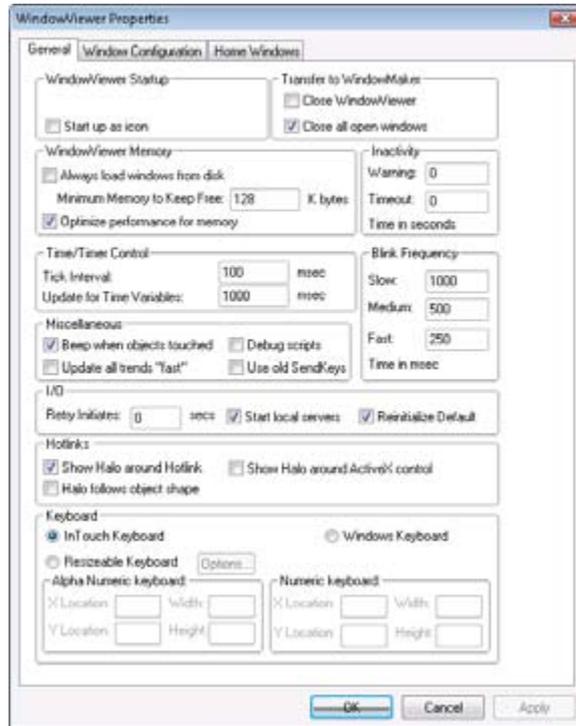
- 1 В меню **Special** (Специальное) щёлкните на **Reinitialize I/O** (Произвести повторную инициализацию ввода/вывода), затем щёлкните на **Select** (Выбрать). В диалоговом окне **Reinitialize I/O** (Произвести повторную инициализацию ввода/вывода) показывается список Имен доступа.



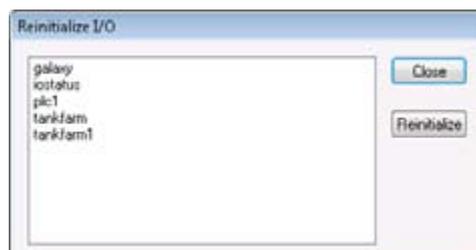
- 2 Щёлкните на одном или нескольких Именах доступа для повторной инициализации, затем щёлкните **Reinitialize** (Произвести повторную инициализацию). Производится повторная инициализация выбранных имен доступа.

Для повторной инициализации Имен доступа с использованием установок, действующих по умолчанию

- 1 Откройте приложение в пределах WindowMaker.
- 2 В меню **Special** (Специальное) щёлкните на **Configure** (Конфигурировать), затем – на WindowViewer. Появится диалоговое окно **Properties** (Свойства) программы WindowViewer со списком опций.



- 3 На закладке **General** (Общая), выберите кнопку-флажок **Reinitialize Default** (Произвести повторную инициализацию установок, действующих по умолчанию) на участке ввода/выводе.
- 4 Щёлкните на **OK**.
- 5 В программе WindowViewer откройте приложение с определёнными Именами доступа.
- 6 В меню **Special** (Специальное) щёлкните на **Reinitialize I/O** (Произвести повторную инициализацию ввода/вывода), после чего щёлкните на **Select** (Выбрать). Появится диалоговое окно (Произвести повторную инициализацию ввода/вывода).



- 7 Выберите одно или несколько Имен доступа для повторной инициализации, затем щёлкните на **Reinitialize** (Произвести повторную инициализацию). Текущие установки игнорируются для имени узла, имени приложения и темы. Имена доступа инициализируются повторно с оригинальными установками Имен доступа.

## Повторная инициализация соединения ввода/вывода с использованием скриптов

Вы можете произвести повторную инициализацию присоединения ввода/вывода к одному или нескольким Именам доступа путём создания скрипта, который включает в себя следующие функции:

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

### Функция IOReinitAccessName()

Функция IOReinitAccessName() повторно инициализирует присоединение ввода/вывода к заданному Имени доступа.

#### Категория

Связь ввода/вывода

#### Синтаксис

```
IOReinitAccessName ("Имя доступа", Default);
```

#### Аргументы

*AccessName (Имя доступа)*

Имя доступа, подлежащее повторной инициализации.

*Default (Установка по умолчанию)*

Установка по умолчанию = 1. В повторной инициализации ввода/вывода используются оригинальные значения Имен доступа, действующие по умолчанию, присвоенные из WindowMaker.

Установка по умолчанию = 0. В повторной инициализации ввода/вывода используются текущие значения узла, приложения и темы, присвоенные Имени доступа.

#### Примечания

Установки по умолчанию определяются установками в панели конфигурации Имени доступа, а также в конфигурации WindowViewer (Retry Initiates, Start Local Servers, Reinitialize Default).

### Примеры

В следующем примере осуществляется повторная инициализация присоединения ввода/вывода к Имени доступа `AccessName1` с использованием значений, действующих по умолчанию, присвоенных узлу, приложению и теме.

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

В следующем примере осуществляется повторная инициализация присоединения ввода/вывода к Имени доступа `AccessName2` с использованием значений, действующих по умолчанию, присвоенных узлу, приложению и теме.

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

### Функция `IOReinitialize()`

Функция `IOReinitialize()` сначала закрывает, а потом перезапускает все активные Соединения ввода/вывода, определённые для приложения `InTouch`.

#### Категория

Разнообразная

#### Синтаксис

```
IOReinitialize () ;
```

#### Аргументы

Нет.

#### Примечания

Функция `IOReinitialize()` выполняет ту же самую операцию, что и команда `Reinitialize I/O` (Произвести повторную инициализацию ввода/вывода) в меню `Special` (Специальное) программы `WindowViewer`.

#### Пример

Данный пример закрывает все активные соединения ввода/вывода и перезапускает соединения ввода/вывода, определённые для приложения `InTouch`.

```
IOReinitialize () ;
```

### Функция `IOStartUninitConversations()`

Когда программа `WindowViewer` начинает запускать приложение `InTouch`, она автоматически обрабатывает первоначальный запрос на запуск всех диалогов ввода/вывода. Если программа Сервера ввода/вывода не отвечает на первоначальный запрос программы `WindowViewer`, вы можете использовать функцию скрипта `IOStartUninitConversations()` для побуждения программы `WindowViewer` на повторную попытку запуска диалога ввода/вывода.

#### Категория

Разнообразная

**Синтаксис**

```
IOStartUninitConversations ( ) ;
```

**Аргументы**

Нет.

**Примечания**

Функция `IOStartUninitConversations()` выполняет ту же самую операцию, что и команда `Start Uninitiated Conversations` (Начать неинициированные диалоги) в меню `Special` (Специальное) программы `WindowViewer`.

**Пример**

Следующий пример побуждает программу `WindowViewer` на подачу другого первоначального запроса для запуска всех соединений ввода/вывода, определённых для приложения `InTouch`.

```
IOStartUninitConversations ( ) ;
```

## Использование функциональности обхода отказа с Именами доступа

Вы можете задать, чтобы человеко-машинный интерфейс `InTouch` автоматически переключался на вторичный Сервер ввода/вывода если в первичном Сервере ввода/вывода возникнут проблемы связи. Это называется обходом отказа ввода/вывода.

### Конфигурирование обхода отказа

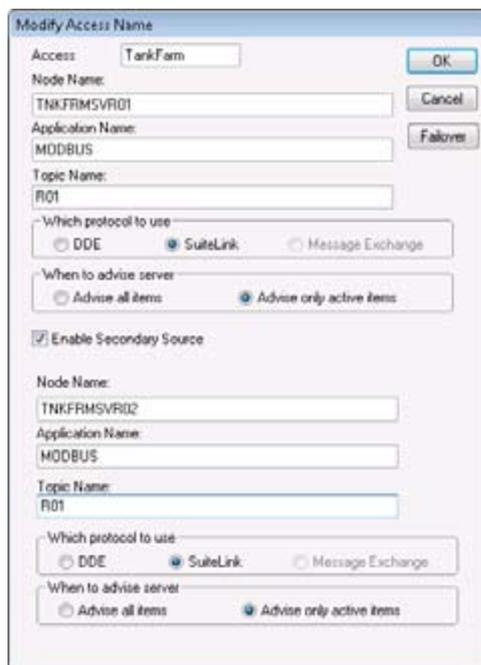
Вы можете задать, чтобы ваше приложение `InTouch` переключалось на вторичный Сервер ввода/вывода в режиме обхода отказа, если оно больше не сможет поддерживать связь с первичным Сервером ввода/вывода.

Когда вы устанавливаете обход отказа, вы задаёте мёртвую зону обхода отказа. Мёртвая зона обхода отказа представляет собой задержку в секундах до переключения с первичного Имени доступа на вторичное Имя доступа. Человеко-машинный интерфейс `InTouch` инициирует обход отказа, когда выражение или отказ связи ввода/вывода являются верными для продолжительности периода мёртвой зоны. Когда мёртвая зона обхода отказа устанавливается на 0 или является пустой, обход отказа инициируется сразу же после обнаружения отказа связи ввода/вывода.

**Конфигурирование обхода отказа для Имени доступа**

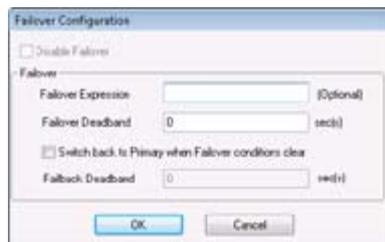
- 1 При необходимости, остановите `WindowViewer`.
- 2 В меню **Special** (Специальное) щёлкните на **Access Names** (Имена доступа). Появляется диалоговое окно **Access Names** (Имена доступа) со списком всех определённых Имен доступа.

- 3 Выберите Имя доступа из списка для добавления сервера обхода отказа.
- 4 Щёлкните на надписи **Modify** (Изменить). Появляется диалоговое окно **Modify Access Names** (Изменить имена доступа).
- 5 Щёлкните на надписи **Enable Secondary Source** (Включить вторичный источник). Диалоговое окно **Modify Access Name** (Изменить имя доступа) расширяется.



- 6 Выполните следующие действия:
  - В окне **Node Name** (Имя узла), введите имя узла вторичного Сервера ввода/вывода.
  - В окне **Application Name** (Имя приложения) введите название программы вторичного Сервера ввода/вывода, из которой будет приниматься данные.
  - В окне **Topic Name** (Имя темы) введите имя темы, к которому вы хотите получить доступ из вторичного источника ввода/вывода.
  - На участке **Which protocol to use** (Какой протокол использовать) выберите либо DDE, либо SuiteLink в качестве протокола связи вторичного Сервера ввода/вывода.
  - На участке **When to advise server** (Когда известить сервер) выберите **Advise all items** (Известить обо всех элементах) или **Advise only active items** (Известить только об активных элементах) для вторичного источника ввода/вывода.

- 7 Щёлкните на **Failover** (Обход отказа). Появляется диалоговое окно **Failover Configuration** (Конфигурация обхода отказа).



- 8 Введите необязательное выражение обхода отказа или дважды щёлкните в окне **Failover expression** (Выражение отказа) для выбора тэга. За дополнительной информацией по выражениям отказа обращайтесь к разделу «Направление обхода отказа к резервному Имени доступа» в главе 5.
- 9 В окне мёртвой зоны обхода отказа введите продолжительность мёртвой зоны обхода отказа в секундах.
- 10 Выберите «**Switch back to primary when Failover conditions clear**» (Выбрать возврат к первичному после устранения состояния обхода отказа), если вы хотите разрешить переключение с вторичного Имени доступа на первичное после устранения состояния обхода отказа.

Установкой, действующей по умолчанию, является отсутствие переключения обратно к первичному Имени доступа. Если вы выбираете «**Switch back to primary when Failover conditions clear**» (Выбрать возврат к первичному после устранения состояния обхода отказа), то опцию **Fail-back Deadband** (Мёртвая зона восстановления) можно будет выбрать из диалогового окна **Failover Configuration** (Конфигурация обход отказа).

- 11 Из опции **Fail-back Deadband** (Мёртвая зона восстановления) введите продолжительность мёртвой зоны восстановления в секундах.

Человеко-машинный интерфейс InTouch инициирует восстановление к первичному Имени доступа после того, как выражение и любой связанный с ним отказ связи ввода вывода будут устранены за период, соответствующий мёртвой зоне. Когда выражение остаётся пустым или устанавливается на 0, восстановление происходит сразу же после устранения состояния отказа связи ввода/вывода.

- 12 Щёлкните на **OK** для закрытия диалогового окна **Failover Configuration** (Конфигурация обхода отказа).
- 13 Щёлкните на **OK** для закрытия диалогового окна **Modify Access Name** (Изменить имя доступа).

## Редактирования параметров Имени доступа пары обхода отказа

Для редактирования параметров Имени доступа, которые составляют часть обхода отказа, вы должны конфигурировать Имя доступа для обхода отказа с вторичным источником ввода/вывода.

### Для редактирования параметров Имени доступа пары обхода отказа

- 1 При необходимости остановите WindowViewer.
- 2 В меню **Special** (Специальное), щёлкните на **Access Names** (Имена доступа). Появляется диалоговое окно **Access Names** (Имена доступа).
- 3 Выберите пару Имён доступа и щёлкните на **Modify** (Изменить). Диалоговое окно **Modify Access Name** (Изменить имя доступа) показывает параметры для первичного и вторичного Имён доступа.
- 4 Измените параметры Имён доступа для первичного и вторичного Имени доступа.
- 5 Щёлкните на **ОК** для закрытия диалогового окна **Modify Access Name** (Изменить имя доступа).
- 6 Щёлкните на **Close** (Закреть) для закрытия диалогового окна **Access Names** (Имена доступа).

## Удаление обхода отказа для Имени доступа

Для удаления обхода отказа для Имени доступа вы должны иметь имя доступа, конфигурированное для обхода отказа с вторичным источником ввода/вывода.

### Удаление обхода отказа для пары Имён доступа

- 1 В меню **Special** (Специальное) щёлкните на **Access Names** (Имена доступа).
- 2 Выберите пару Имён доступа и щёлкните на **Modify** (Изменить). Появляется диалоговое окно **Modify Access Name** (Изменить имя доступа).
- 3 Очистите кнопку-флажок **Enable Secondary Source** (Включить вторичный источник).
- 4 Щёлкните на **ОК**. Обход отказа для пары имён доступа отключается.

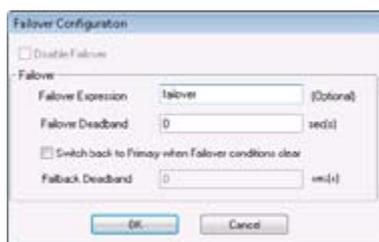
## Направление обхода отказа к резервному Имени доступа

Вы можете вручную осуществлять переключение между первичным и вторичным источниками Имени доступа без прохождения через ситуацию обхода отказа. Это называется принудительным обходом отказа. Для осуществления принудительного обхода отказа вы должны иметь Имя доступа, конфигурированное на обход отказа с вторичным источником ввода/вывода.

Вы можете использовать выражение обхода отказа или функцию `IOForceFailover()` script для осуществления принудительного обхода отказа.

### Выражение обхода отказа

В диалоговом окне Failover Configuration (Конфигурация обхода отказа) показывается опция Failover Expression (Выражение обхода отказа), которая должна включать в себя тэг или выражение, запускающие обход отказа. На рисунке, приведённом ниже, показано диалоговое окно с дискретным тэгом памяти обхода отказа, введённым как значение Failover Expression.



Установка выражения обхода отказа на True (верное), например, путём установки тэга обхода отказа на True (верное), переключает Имя доступа с первичного (ложного) на вторичный (верный) источник данных.

### Функция IOForceFailover()

Функция скрипта `IOForceFailover()` осуществляет переключение между первичным и вторичным источниками данных Имени доступа. Активный узел ввода/вывода осуществляет переключение между первичным и вторичным узлом с каждым вызовом функции скрипта.

Обычно функция `IOForceFailover()` составляет часть скрипта, связанного с кнопкой или другим объектом окна. Операторы выбирают объект из окна приложения для принудительного осуществления обхода отказа. После того, как операторы щёлкнут на объекте повторно, функция `IOForceFailover` приведёт соединение ввода/вывода обратно к ранее активному узлу ввода/вывода.

#### Категория

Связь ввода/вывода

#### Синтаксис

`IOForceFailover ("Имя доступа") ;`

#### Аргумент

*AccessName (Имя доступа)*

Имя доступа, для которого конфигурирован обход отказа.

### Пример

Имя доступа Acc1 имеет Первичный и Вторичный источники данных, и Первичный является активным. При обходе отказа Acc1 переключается на Вторичный источник данных при работе скрипта.

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

## Временное отключение функциональности обхода отказа

Вы можете отключить ручную переключение между первичным и вторичным узлами ввода/вывода Имени доступа в режиме обхода отказа. Типичным случаем для временного отключения обхода отказа является кратковременный период, когда компоненты системы InTouch запущены, но ещё не готовы. После того, как компоненты будут стабилизированы, вы можете восстановить переключение в режиме обхода отказа.

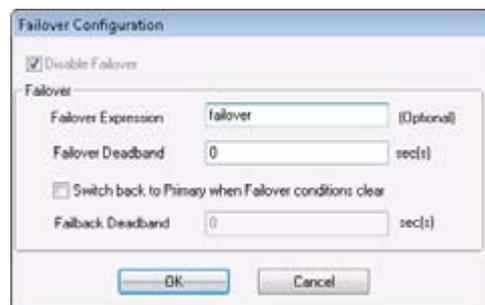
Для отключения обхода отказа для Имени доступа вы должны иметь Имя доступа, конфигурированное для обхода отказа с вторичным источником ввода/вывода.

Вы можете вручную отключить переключение в режиме обхода отказа двумя методами:

- Выберите опцию **Disable Failover** (Отключить обход отказа) из диалогового окна **Failover Configuration** (Конфигурация обхода отказа).
- Запустите скрипт, который включает в себя функцию `IODisableFailover()`.

### Опция конфигурации **Disable Failover**

Диалоговое окно **Failover Configuration** (Конфигурация обхода отказа) включает в себя опцию **Disable Failover** (Отключить обход отказа) для предотвращения переключения Имени доступа между первичным и вторичным узлами ввода/вывода. На рисунке, приведённом ниже, показано диалоговое окно с опцией **Disable Failover** (Отключить обход отказа), установленной в активное состояние.



Вы должны редактировать определение Имени доступа для установки опции **Disable Failover** в активное состояние. Пока

данная опция будет активна для Имени доступа, обход отказа будет отключен.

### Функция `IODisableFailover()` Script

Вы можете использовать функцию `IODisableFailover()` в скрипте для отключения обхода отказа для заданного Имени доступа. Функция `IODisableFailover()` отключает переключение для всех методов обхода отказа, за исключением метода функции скрипта `IOForceFailover()`.

#### Категория

Связь ввода/вывода

#### Синтаксис

```
IODisableFailover ("Имя доступа", Опция);
```

#### Аргументы

*AccessName* (Имя доступа)

Имя доступа, для которого конфигурирован обход отказа.

#### Опция

1 = Отключает обход отказа

0 = Включает обход отказа

#### Примечания

Имя доступа может быть задано как буквенная строка или может представлять собой строковое значение, предоставленное другими тэгами или функциями `InTouch`.

#### Примеры

В следующем примере обход отказа отключен для Имени доступа `ModbusPLC1`.

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

В следующем примере обход отказа включен для Имени доступа `ModbusPLC1`.

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

Извлечение информации о парах обхода отказа с использованием составления скриптов

Вы можете писать скрипты, которые включают в себя функции, которые возвращают состояние первичного, вторичного и активного источника ввода/вывода Имени доступа. Обычно операторы запускают скрипт для определения состояния вторичного источника ввода/вывода перед осуществлением принудительного обхода отказа.

Для создания скриптов, предназначенных для возврата информации об Имени доступа, вы должны иметь Имя доступа, конфигурированное для обхода отказа с вторичным источником ввода/вывода.

## Функция IOGetAccessNameStatus()

Функция скрипта IOGetAccessNameStatus() возвращает целочисленную величину, указывающую состояние присоединения первичного, вторичного или активного источника ввода/вывода Имени доступа.

Обычно значение возврата IOGetAccessNameStatus() связано с целочисленным тэгом. Значение тэга может приводить в действие анимационную ссылку дисплея целочисленных величин, которая показывает оператору состояние активного, первичного и вторичного источника данных Имен доступа.

### Категория

Разнообразная

### Синтаксис

```
Result=IOGetAccessNameStatus("Имя доступа", Режим);
```

### Аргументы

*AccessName (Имя доступа)*

Существующее Имя доступа, для которого следует вернуть имя источника.

*Mode (Режим)*

Значение, присвоенное данному аргументу, определяет, что спрашивается у Имени доступа обхода отказа о его текущем состоянии.

0 – Состояние активного источника ввода/вывода Имени доступа

1 – Состояние первичного источника ввода/вывода Имени доступа

2 – Состояние вторичного источника ввода/вывода Имени доступа

### Результаты

Возвращённая величина	Описание
-1	Имеет место ошибка конфигурации в Имени доступа. Имя доступа либо не существует, либо не имеет определённого вторичного источника ввода/вывода.
0	Присоединение к запрашиваемому источнику ввода/вывода безуспешно.
1	Присоединение к запрашиваемому источнику ввода/вывода успешно.

**Примечания**

Функция `IOGetAccessNameStatus()` обычно используется в скрипте, который определяет состояние вторичного источника ввода/вывода, который в текущий момент является неактивным. Оператор запускает скрипт для проверки состояния вторичного присоединения перед тем, как осуществлять принудительный обход отказа.

**Пример**

Следующий пример возвращает состояние вторичного источника ввода/вывода Имени доступа `ModbusPLC1`. Возвращаемое значение связано с тэгом `ANStatus`.

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

**Функция `IOGetActiveSourceName()`**

Функция скрипта `IOGetActiveSourceName()` возвращает имя текущего узла используемого в качестве источника данных для заданного Имени доступа.

Обычно функция `IOGetActiveSourceName()` включается в скрипт, связанный с кнопкой или другим объектом окна. Затем операторы выбирают объект из окна приложения для запроса состояния Серверов ввода/вывода приложения.

**Категория**

Разнообразная

**Синтаксис**

```
Result=IOGetActiveSourceName ("Имя доступа") ;
```

**Аргумент**

*AccessName (Имя доступа)*

Существующее Имя доступа, для которого должно быть возвращено состояние источника.

**Примечания**

`IOGetActiveSourceName()` возвращает строку, которая указывает, первичные или вторичные узлы Имени доступа в настоящее время активно опрашиваются. Возможными возвращаемыми значениями функции `IOGetActiveSourceName()`:

Первичный	Первичный узел Имени доступа активно опрашивается.
Вторичный	Вторичный узел или узел обхода отказа Имени доступа активно опрашивается.
Нуль	Ни первичный, ни вторичный узел Имени доступа не является активным.

### Пример

В следующем примере, тэг сообщения ActiveServer присвоен возвращаемому значению (первичному, вторичному или нулю), которое идентифицирует текущий активный узел Имени доступа ModbusPLC1.

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

## Контроль состояния соединения ввода/вывода

WindowViewer включает в себя встроенную тему, называемую IOStatus для контроля состояния конкретного диалога ввода/вывода между приложениями InTouch и Сервера ввода/вывода, поддерживающими связь с программируемым логическим контроллером.

---

**Примечание:** В версиях InTouch до 7.0 имя темы было DDEStatus.

---

Вы можете установить тему IOStatus для контроля диалогов ввода/вывода.

## Использование имени темы IOStatus

Вы можете подготовить тему IOStatus для контроля Связи ввода/вывода между Сервером ввода/вывода и программируемым логическим контроллером. В данном примере WindowViewer поддерживает связь с использованием Сервера ввода/вывода Wonderware Simulate с программируемым логическим контроллером, определённым в Сервере ввода/вывода с PLC1 как имя его темы.

---

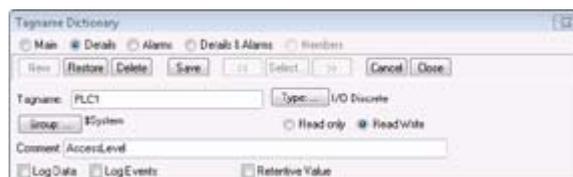
**Примечание:** Simulate представляет собой общий сервер Wonderware DAServer, используемый в качестве тренировочного инструмента. Сервер Simulate расположен в папке c:\program files\common files\Archestra.

---

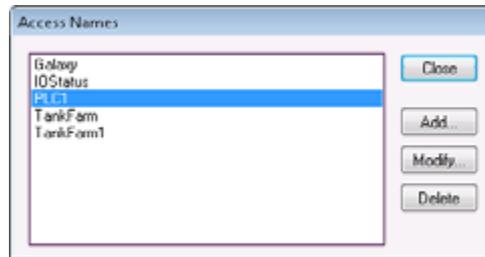
### Для контроля состояния Связи ввода/вывода

- 1 Открыть приложение в WindowMaker.
- 2 Открыть Словарь имён тэгов.
- 3 Создать дискретный тэг ввода/вывода.

Когда вы осуществляете контроль за диалогом ввода/вывода с использованием IOStatus, вы должны определить, как минимум, один тэг типа ввода/вывода для контролируемого Имени доступа.

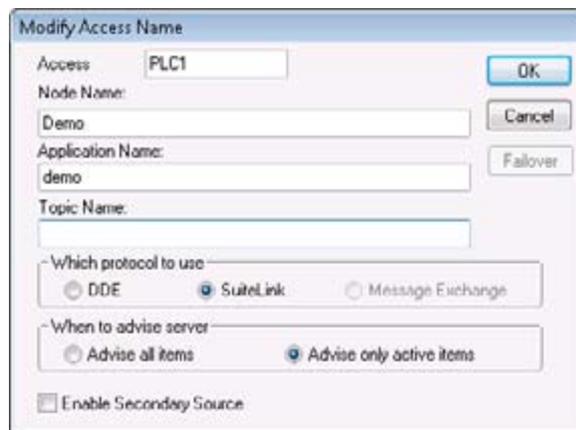


- 4 Щёлкните на **Access Name** (Имя доступа) для присвоения тэга определению Имени доступа, которое определяет IOStatus как имя его темы.



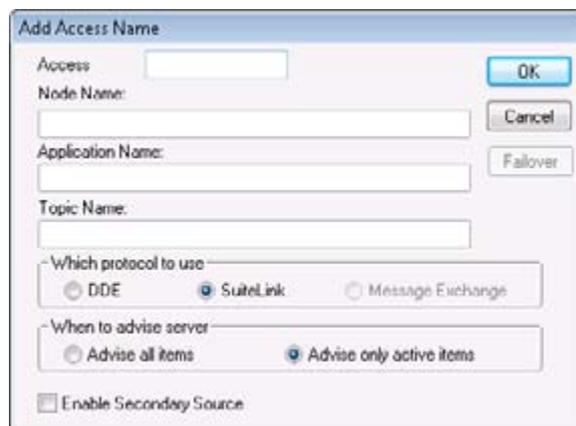
Отметьте, что определение Имени доступа **PLC1** в настоящее время существует.

- 5 Выберите **PLC1** и щёлкните на **Modify** (Изменить).



Обнаружение Имени доступа, содержащего правильное имя темы в следующем примере, является простым, поскольку тэг и **Topic Name** (Имя темы) являются одним и тем же.

- 6 Щёлкните на надписи **Cancel** (Отмена) для закрытия диалогового окна и возврата в первоначальное диалоговое окно **Access Name Definition** (Определение имени доступа).
- 7 Щёлкните на **Add** (Добавить). Появляется диалоговое окно **Add Access Name** (Добавить имя доступа).



- 8 Выполните следующие действия:
  - a В окне **Access** (Доступ) введите IOStatus.
  - b В окне **Application Name** (Имя приложения) введите **View** (Вид), поскольку вы собираетесь контролировать состояние из программы WindowViewer.
  - c В окне **Topic Name** (Имя темы) введите IOStatus в качестве внутренней темы InTouch.
  - d Выберите **Advise only active items** (Сообщить только об активных элементах).
- 9 Щёлкните на **OK** для закрытия диалогового окна. Первоначальное диалоговое окно **Access Name** (Имя доступа) появляется снова, показывая ваше новое Имя доступа, IOStatus, в списке:



- 10 Щёлкните на **Close** (Закреть) для закрытия диалогового окна и связи нового Имени доступа с вашим тэгом **I/O Discrete** (Дискретный ввод/вывод).
- 11 В окне **Item** (Элемент) введите Имя доступа для фактического имени темы, которую вы хотите контролировать. В данном случае – **PLC1**.



- 12 Поскольку ваш тэг является таким же самым, как и **Topic Name** (Имя темы), вы можете выбрать **Use Tagname as Item Name** (Использовать имя тэга в качестве имени элемента) и автоматически ввести его в качестве элемента (Item).

**Примечание:** При использовании встроенной темы IOStatus (DDEStatus до версии InTouch 7.0) для контроля диалога ввода/вывода имя, которое вы вводите в окне Access Name (Имя доступа) всегда используется также для Item.

### Использование имени темы IOStatus в программе Excel

Вы можете использовать программу Excel для контроля состояния ввода/вывода путём ввода той же самой информации, что и в формуле ячейки крупноформатной таблицы. Например, для контроля той же самой темы, которая была описана в предыдущей процедуре, введите в ячейку следующую формулу:

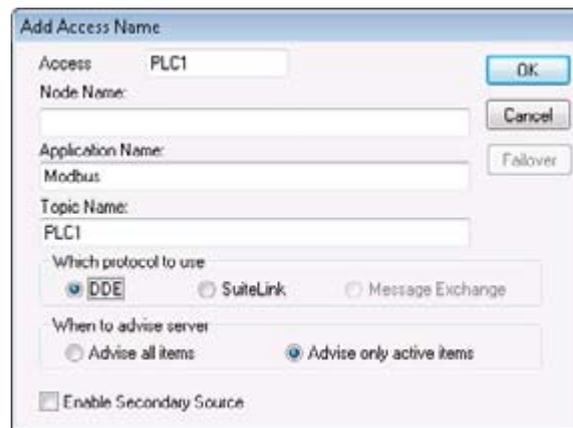
```
=view|IOStatus!' PLC1'
```

## Контроль состояния связи Сервера ввода/вывода

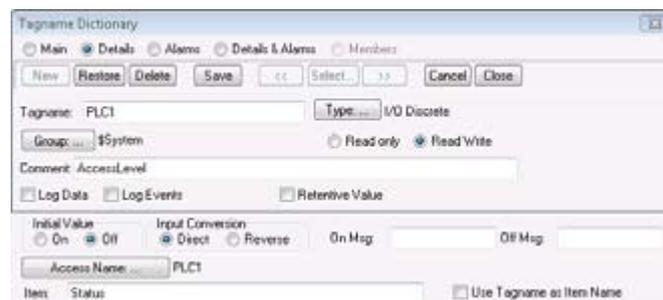
Для каждого используемого имени темы вы можете использовать встроенный дискретный элемент, Status (Состояние), для контроля состояния связи с программой Сервера ввода/вывода. Элемент Состояние устанавливается на 0, когда имеет место отказ связи. Элемент Состояние устанавливается на 1, при нормальной связи с программой Сервера ввода/вывода.

**Примечание:** Когда вы контролируете состояние темы, используя элемент IOStatus, как минимум, одна точка ввода/вывода должна быть активна для контролируемой темы.

Из человеко-машинного интерфейса InTouch вы можете прочитать состояния связи сервера путём определения тэга и связывания его с конфигурируемой темой для устройства путём использования слова Status (Состояние) в качестве имени элемента. Например, если WindowViewer поддерживает связь с программируемым логическим контроллером, используя сервер Wonderware Simulate DAServer, определением Имени доступа является IOStatus:



Для контроля состояния всей связи с темой PLC1 создайте следующее определение тэга:



В программе Excel вы можете прочитать состояние связи программируемого логического контроллера путём ввода в ячейку следующей формулы:

**=SIMULATE | PLC1 ! ' STATUS '**

## Доступ к данным тэга InTouch из других приложений

Когда другое приложение требует значения данных от человеко-машинного интерфейса InTouch, он должен также знать три элемента адреса ввода/вывода. Примите к сведению следующие условные обозначения адресов ввода/вывода InTouch.

VIEW (ВИД) (имя приложения) идентифицирует программу реального времени InTouch, которая содержит элемент данных.

TAGNAME (ИМЯ ТЭГА) (имя темы) – слово, всегда используемое при считывании из тэга/записи в тэг.

ActualTagname (Фактическое имя тэга) (Имя элемента) – фактический тэг, определённый для элемента в Словаре имён тэгов InTouch.

Например, для получения доступа к значению данных в человеко-машинном интерфейсе InTouch из программы Excel, формула удалённой ссылки DDE задаётся для ячейки, в которую записываются данные:

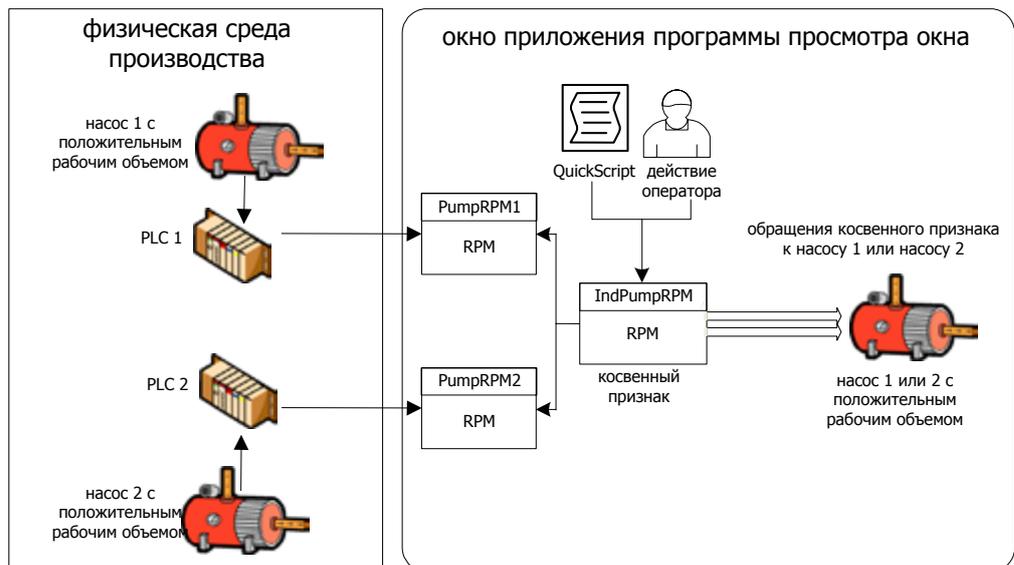
```
=VIEW | TAGNAME!'ActualTag_name'
```

# Глава 6

## Определение косвенных признаков

Используя косвенные признаки вы можете создавать приложения с объектами окна, которые показывают значения от различных признаков.

На рисунке ниже показан объект насоса в окне приложения. Объект насоса представляет два возможных технологических насоса на основе значений, установленных из косвенного признака. QuickScript или действие оператора выбирает признак источника, связанный с косвенным признаком.



Косвенные метки минимизируют время вашей разработки. Вы создаете меньше окон приложения, так как один объект окна может представлять несколько процессов, выполняющихся в среде производства.

## Использование косвенных признаков со скриптами

Вы можете использовать скрипты для присваивания признаков входного источника для косвенных меток. Вы присваиваете признаки входного источника косвенному признаку путем присваивания имени признака источнику полю точек .Name косвенного источника.

Например, если вы создаете косвенный аналоговый признак, называемый IndPumpRPM, то два признака источника PumpRPM присваиваются ему с помощью операторов скрипта аналогично следующему примеру:

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

Скрипт присваивания косвенного признака может запускаться событием в приложении или действиями оператора, вроде щелчка кнопкой в окне.

Когда вы приравниваете косвенную метку другой метке источника, косвенная метка ведет себя так, как если бы это был признак источника. Если значение признака источника меняется, косвенный признак отражает эти изменения. Если меняется величина косвенного признака, признак источника соответственно меняется.

Так как поле точек .Name является простой строкой, вы можете динамически присваивать получателя косвенного признака во время исполнения. Например, если вы создаете Data Change QuickScript, который выполняется каждый раз, как меняется значение признака Number, признак источника, присвоенный косвенному признаку IndPumpRPM меняется соответственно:

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

Когда выполняется скрипт, значение аналогового признака Number преобразуется в текст и прибавляется к строке PumpRPM. Если Number равняется 1, это устанавливает имя косвенного признака Number в PumpRPM1.

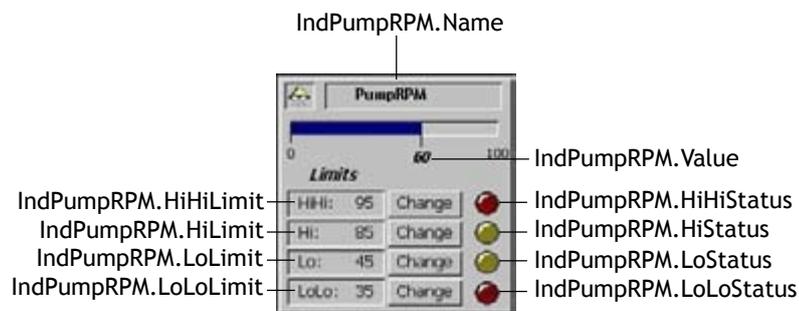
Косвенные признаки аналогового типа используются как для целочисленных, так и для действительных признаков. Косвенные признаки могут преобразовываться в какие либо другие признаки, оставаясь при этом признаком того же типа.

Вы можете также присвоить сохраняемые атрибуты косвенным признакам. При сохранении косвенная метка сохраняет самое последнее присваивание для признака, при повторном запуске приложения.

## Использование косвенных признаков с локальными признаками

Косвенные признаки, как правило, используются с признаками, определенными в локальном словаре Tagname Dictionary. Косвенный локальный признак позволяет вам создавать визуальные объекты, которые показывают несколько атрибутов локального признака. Например, вы можете создавать модуль заглушку в окне приложения. Модуль заглушка содержит выбираемые пункты со связями различным полям точек, назначенным косвенным локальным признаком. В следующем примере оператор изменяет пределы сигнала поля точек, связанные с косвенным локальным признаком PumpRPM.

На следующем рисунке показан модуль заглушка с анимационными связями на пределы сигнализации RPM (оборотов в минуту) насоса. Косвенный признак присваивает атрибуты различных локальных признаков модулю заглушке пределов сигнализации.



Для перенаправления модуля заглушки на соответствующий признак включите оператор в пределах QuickScript. Оператор дополняет оператор поля точек до имени косвенного признака:

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

В этом примере скрипта tag\_name – это имя фактического признака, определенного с локальным словаре Tagname Dictionary. Когда скрипт выполняется, все значения поля точек, связанные с этим локальным признаком, становятся доступными для объекта приложения посредством косвенного признака.

## Использование косвенных признаков с дистанционными обращениями

Дистанционные обращения к косвенным признакам отличаются от локальных обращений к признакам. Синтаксис для дистанционного обращения следующий:

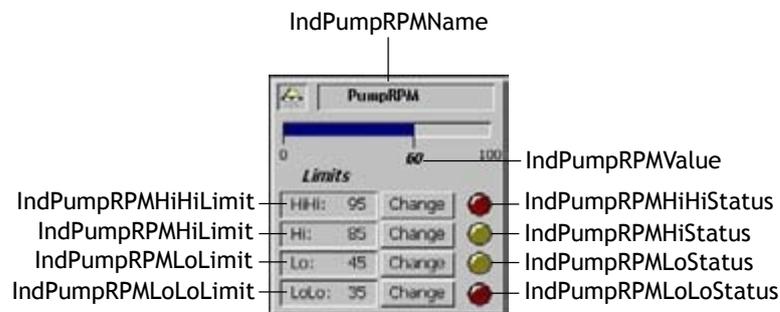
*AccessName:Item*

где

- *AccessName* – это любое действующее имя доступа InTouch.
- *Item* – это любое действующее имя, которое поддерживается сервером ввода/вывода, указанным в определении имени доступа.

Когда вы используете дистанционные обращения, сервер возвращает значение клиенту, а не структуре признака. Значение включает отметку времени и отметку качества. Таким образом косвенная метка, присвоенная дистанционному обращению не может получить доступа к каким либо полям точек признака, иным нежели те, которые связаны со значением, временем и качеством. Например, косвенный признак не может получить доступ к атрибутам признака через дистанционное обращение для указания пределов сигнализации.

Одно из возможных решений состоит в создании модуля заглушки с набором косвенных признаков. На следующем рисунке показан модуль заглушка для изменения пределов сигнализации для насоса.



В этом примере модуль заглушка использует 10 косвенных признаков, которые связываются с подразумеваемым полем точек Value. Модуль заглушка сигнализации перенаправляется на признак дистанционного обращения, IndPumpRPM, на удаленном узле InTouch с именем TagServer1. Имя доступа InTouch конфигурируется следующим образом:

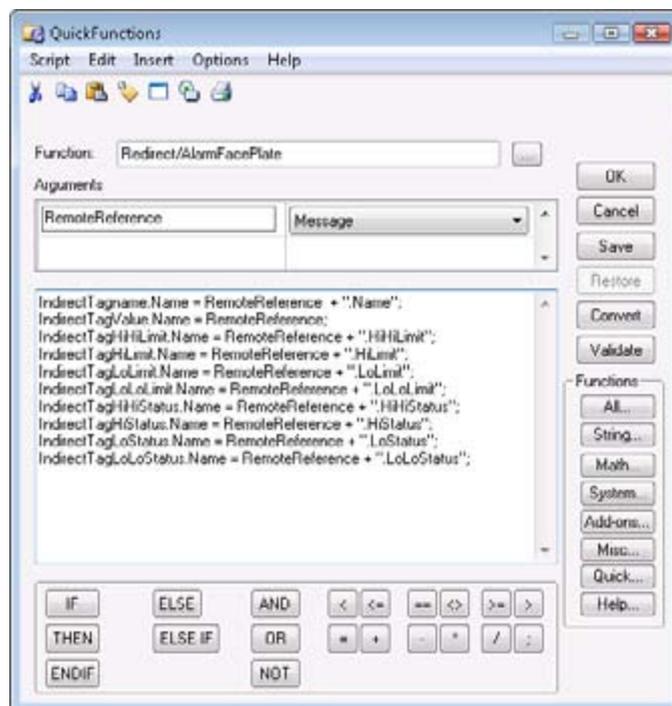
<b>Access Name</b> (имя доступа):	TagSource1
<b>Node Name</b> (имя узла):	TagServer1
<b>Application Name</b> (имя приложения):	View
<b>Topic Name</b> (имя темы):	TagName

Для перенаправления модуля заглушки на признак дистанционного обращения PumpRPM, выполните следующий QuickScript:

```
IndirectTagname.Name = "TagSource1:PumpRPM.Name";
IndirectTagValue.Name = "TagSource1:PumpRPM";
IndirectTagHiHiLimit.Name =
    "TagSource1:PumpRPM.HiHiLimit";
IndirectTagHiLimit.Name =
    "TagSource1:PumpRPM.HiLimit";
IndirectTagLoLimit.Name =
    "TagSource1:PumpRPM.LoLimit";
IndirectTagLoLoLimit.Name =
    "TagSource1:PumpRPM.LoLoLimit";
IndirectTagHiHiStatus.Name =
    "TagSource1:PumpRPM.HiHiStatus";
IndirectTagHiStatus.Name =
    "TagSource1:PumpRPM.HiStatus";
IndirectTagLoStatus.Name =
    "TagSource1:PumpRPM.LoStatus";
IndirectTagLoLoStatus.Name =
    "TagSource1:PumpRPM.LoLoStatus";
```

Скрипт должен выполняться каждый раз, когда перенаправляется модуль заглушка. Другое решение состоит в создании InTouchQuickFunction, которая позволяет вам записывать одиночный скрипт и передавать ему имя дистанционного обращения. Вы можете уменьшить объем программы скрипта путем использования нескольких модулей заглушек, которые вызывают ту же QuickFunction.

Например, используя аналогичный набор команд скрипта, вы можете определить QuickFunction, называемую RedirectAlarmFacePlate.



Вы можете вызвать функцию RedirectAlarmFacePlate для обработки всего перенаправления. Чтобы сделать это, функция должна вызываться другим InTouchQuickScript. Например:

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

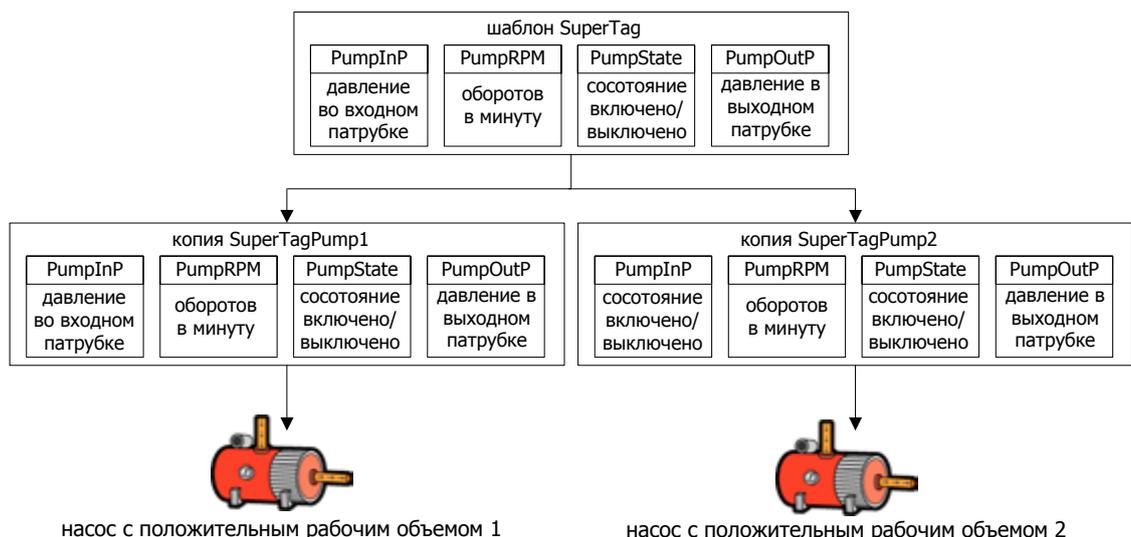
# Глава 7

## Определение многократно используемых структур признаков

SuperTag – это шаблон для группы родственных признаков. Признаки, которые принадлежат шаблону SuperTag, связываются с общими свойствами компонента в производственном процессе.

SuperTag экономит время разработки. Вместо создания набора признаков для каждого компонента в производственном процессе, вы можете размножить один шаблон SuperTag и создавать отдельные копии для все компонентов процесса, которые имеют одинаковые свойства.

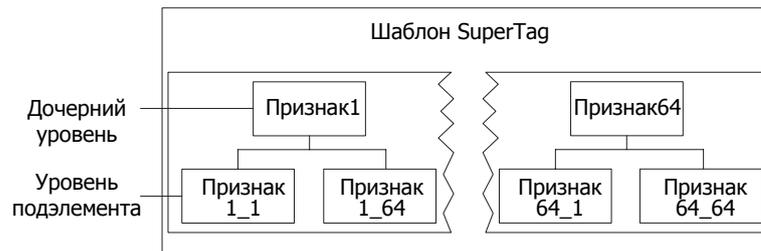
На рисунке ниже показан один шаблон SuperTag, состоящий из набора родственных признаков, связанных с данными о насосе. Шаблон может размножаться для создания копий SuperTag для всех одинаковых насосов в процессе.



Все признаки, которые принадлежат копии SuperTag, ведут себя точно, как обычные признаки. Признаки элемента могут присваиваться, как типы данных дискретный, целый, действительный и сообщение InTouch. Признаки поддерживают анализ тенденций, сигнализацию, и все поля точек признаков.

Шаблон SuperTag может организовать свои признаки элемента по двум уровням вложения. Шаблон SuperTag может содержать до 64 встроенных дочерних признаков. Каждый дочерний признак может содержать до 64 признаков элемента. Это дает вам в сумме 4095 признаков в шаблоне SuperTag.

На следующей схеме показано, как признаки организуются в шаблоне SuperTag.



Когда один исходный элемент шаблона SuperTag встраивается в другой шаблон SuperTag, встроенный признак становится дочерним элементом.

После того, как вы создаете исходный шаблон SuperTag, Tagname Dictionary (словарь имен признаков) вносит его в список, как тип признака в диалоговом блоке Select Tag. Шаблон может выбираться сразу же, как тип признака, когда вы создаете новый признак. Вам не нужно повторно запускать WindowMaker для определения признаков, которые использует заново создаваемый тип SuperTag.

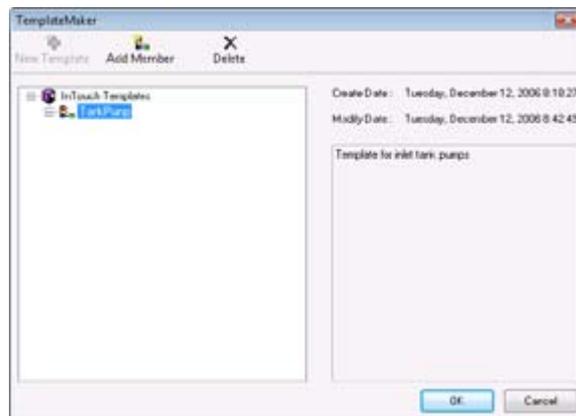
Все шаблоны SuperTag сохраняются в файле supertag.dat в папке C:\Documents and Settings\All Users\Application Data\Wonderware\InTouch.

## Определение шаблона SuperTag

Вы используете утилиту TemplateMaker для создания, редактирования и удаления шаблонов SuperTag и их признаков элементов. Диалоговый блок Tag Types Tagnfme Dictionary добавляет шаблон SuperTag и косвенную форму шаблона, как типы признака.

### Для создания шаблона SuperTag

- 1 С меню **Special** щелкните на **TemplateMaker**. Появляется диалоговый блок **TemplateMaker**.
- 2 Щелкните на **New Template**. Появляется диалоговый блок **New Template**.
- 3 Введите уникальное имя для шаблона **SuperTag** в блоке **Name**.  
Имя шаблона может иметь максимальную длину до 10 цифробуквенных символов. Первый символ в имени шаблона должен быть буквой алфавита.
- 4 Дополнительно введите комментарий, которые описывает шаблон в блоке **Description**.
- 5 Щелкните **OK**. Появляется диалоговый блок **TemplateMake** снова с новым шаблоном, внесенным в список. После добавления шаблона становятся активными кнопки **Add Member** и **Delete**.



**Для добавления признаков к шаблону SuperTag**

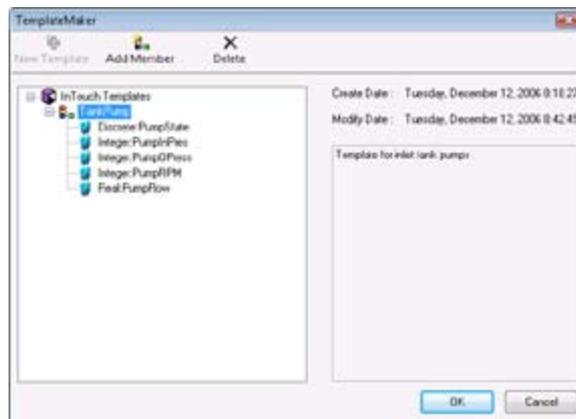
- 1 Откройте диалоговый блок **TemplateMaker**.
- 2 Выберите шаблон **SuperTag** из перечня.
- 3 Щелкните на **Add Member** для отображения диалогового блока **New Member Tag**.
- 4 Прделайте следующее:
  - В блоке Name введите имя признака элемента, добавляемого к шаблону  
Имя признака элемента может иметь до 10 цифробуквенных символов. Первым символом должна быть буква.

---

**Примечание.** Внимательно относитесь к длине имен ваших признаков, если вы будете использовать дистанционное обращение. Дополнительную информацию касательно ограничений для обращений SuperTag смотрите в разделе на странице 162 Обращение к элементам SuperTag.

---

- В блоке Type введите тип признака для элемента или щелкните Type и выберите тип признака из перечня.  
Тип признака может быть дискретным, целым, действительным, сообщением или другим шаблоном SuperTag. По умолчанию, все признаки элемента устанавливаются в типы запоминающего устройства, когда вы определяете их в TemplateMaker. Однако, когда вы определяете копию шаблона в Tagname Dictionary, вы должны указать, остаются ли элементы шаблона как признаки запоминающего устройства, либо они меняются в признаки ввода/вывода.
  - Дополнительно введите дополнительный комментарий, который описывает шаблон в блоке Comment.
- 5 Щелкните **OK**. Появляется признак нового элемента под исходным шаблоном **SuperTag** в диалоговом блоке **TemplateMaker**.



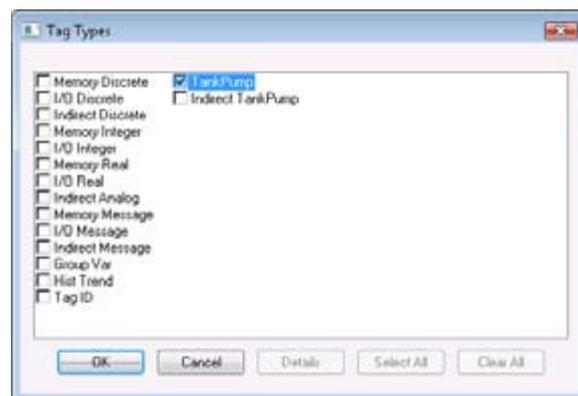
- 6 Имя признака включает выбранный тип признака, присвоенный признаку, в виде ограниченного столбца.

Пример:

Dioscrete:PumpState

- 7 Повторите эти этапы для добавления признаков к шаблону **SuperTag**. Диалоговый блок **TemplateMaker** перечисляет все признаки, которые вы добавили ниже имени шаблона **SuperTag**.

После создания шаблона SuperTag диалоговый блок Tag Types показывает как шаблон SuperTag, так и его косвенную форму в виде типов признаков.



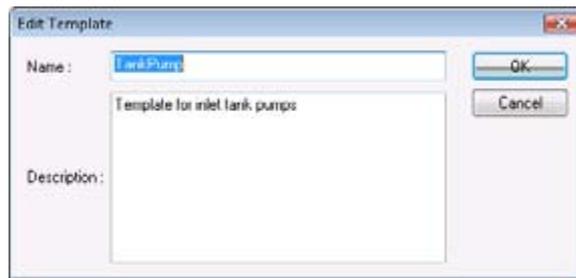
## Редактирование шаблонов SuperTag и признаков элемента

Вы можете изменить шаблоны SuperTag или признаки элементов в любое время. Существующие копии SuperTag, образовавшиеся из шаблона, не воспринимают эти изменения шаблона. Однако, все новые копии приобретают изменения модифицированного шаблона.

Вы можете удалить весь шаблон SuperTag и выбрать признаки элемента, которые принадлежат шаблону. Удаленные шаблоны больше не перечисляются в диалоговом блоке Tag Types в Tagname Dictionary.

### Для редактирования существующего шаблона SuperTag или признака элементов

- 1 Откройте диалоговый блок **TemplateMaker**.
- 2 дважды щелкните на имени шаблона **SuperTag** или признаке элемента, который вы хотите редактировать. Появляются диалоговые блоки **Edit Template** или **Edit Member Tag**. Диалоговый блок показывает текущее определение либо шаблона **SuperTag**, либо признак элемента.



- 3 Сделайте свой выбор.
- 4 Щелкните **OK**.
- 5 Диалоговый блок **TemplateMaker** обновляется и показывает дату и время проведенных изменений при редактировании.

### Для удаления шаблона SuperTag или признака элемента

- 1 Откройте диалоговый блок **TemplateMaker**.
- 2 Выберите имя шаблона **SuperTag** или признак элементов, который вы хотите удалить.
- 3 Щелкните **Delete**. Сообщение требует подтверждения удалить выбранный пункт.
- 4 Щелкните **Yes** для удаления выбранного имени.

Перечень **TemplateMaker** обновляется и удаляет удаляемый пункт.

---

Важные признаки элемента копии шаблона SuperTag не могут удаляться. Например, если PumpRPM является признаком элемента шаблона TankPumpSuperTag, он не может быть удален из любой копии TankPump. Вы можете только удалить признаки из шаблона SuperTag.

---

## Создание копий SuperTag

Шаблоны SuperTag и копия шаблона различны. Копия – это фактическое воплощение шаблона SuperTag в приложении InTouch.

Наиболее важное различие между шаблоном и копией состоит в том, что имя исходного шаблона заменяется именем признака копии. Имена дочернего шаблона и признаки подэлемента не меняются.

## Использование Словаря Tagname для создания копии SuperTag

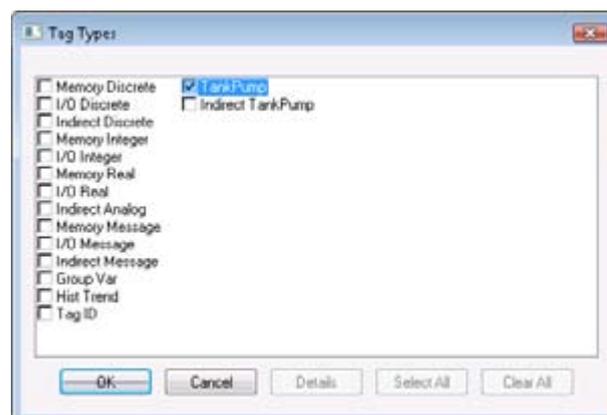
Вы можете использовать Словарь Tagname для создания копии SuperTag. Словарь Tagname автоматически создает все признаки элемента и признаки подэлемента, когда вы определяете новую копию SuperTag.

### Для создания копии SuperTag из шаблона

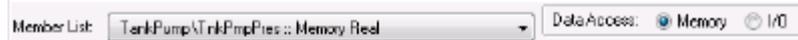
- 1 Откройте Словарь Tagname.
- 2 Щелкните **New**.
- 3 В блоке **Tagname** введите имя, которое вы хотите для новой копии SuperTag.

Имя копии SuperTag может быть до 10 символов. Имя копии следует тем же правилам присваивания имен, что и для признаков InTouch.

- 4 Щелкните **Type** для отображения диалогового блока **Tagname Types**
- 5 Выберите имя шаблона **SuperTag** из перечня.



- 6 Щелкните **ОК**. Диалоговый блок **Tagname Dictionary** расширяется для отображения дополнительных опций.



Новый признак, который вы ввели в блок **Tagname**, становится исходным для всех признаков элементов, которые принадлежат к выбранному шаблону **SuperTag**.

- 7 Установите свойства признака. Сделайте следующее:
- a В блоке **Member List** выберите признак из перечня шаблона **SuperTag**.



- b Из **Data Access** выберите **Memory** или **I/O** для отображения соответствующего подробного диалогового блока **Memory** или **I/O**.
  - c Введите подробности, как вы выполняете для стандартного признака **InTouch**.
  - d Выберите оставшиеся признаки элемента из перечня и сконфигурируйте их.
- 8 Щелкните **Close** после того, как вы укажете все подробности для признаков элементов, которые принадлежат к шаблону **SuperTag**.

## Использование Tagname Dictionary для размножения копий SuperTag

Вы можете использовать Tagname Dictionary для размножения существующей копии. После того, как вы выполните копию, признаки становятся сразу же доступными для использования в анимационных связях и в InTouch QuickScripts.

### Для размножения копии SuperTag из Tagname Dictionary

- 1 Откройте **Tagname Dictionary**.
- 2 Щелкните **Select** для отображения диалогового блока **Select Tag** с перечнем признаков, определенных для приложения.
- 3 Выберите копию **SuperTag** из перечня для использования в качестве шаблона для вашей новой копии.
- 4 Щелкните **OK**. Имя выбранного шаблона появляется в блоке **Tagname**.
- 5 Щелкните **New**. Сообщение требует подтверждения размножения копии **SuperTag**.
- 6 Щелкните **Yes**. Появляется диалоговый блок **Enter Name** и вам дается подсказка для имени новой **SuperTag**.
- 7 Введите имя до 10 символов с использованием стандартных соглашений о присваивании имени признаку.
- 8 Щелкните **OK**. Новый шаблон **SuperTag** появляется в **Tagname Dictionary**.
- 9 При необходимости, отредактируйте признаки элемента, как это вы делаете для нормального признака InTouch.
- 10 Щелкните **Close**.

## Использование Tagname Dictionary для добавления признака к копии SuperTag

Вы можете добавлять признак, как элемент существующей копии SuperTag с использованием Tagname Dictionary. Когда вы добавляете признак, вы вводите точное имя вашей копии SuperTag с последующим ограничителем из обратной наклонной черты (\) и имя нового признака элемента. Например:

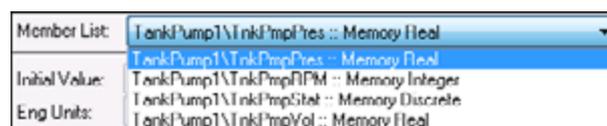
`Pump_8\PumpSTS`

**Примечание** если вы планируете использовать Утилиту заимствования ArchestrA Bulk Import Utility для перемещения признаков из вашего приложения InTouch в Сервер приложения, смотрите раздел Заимствование SuperTags с помощью утилиты заимствования Bulk на странице 162 для получения больше информации о замещении стандартного ограничителя в виде обратной наклонной черты.

### Для добавления признака к копии SuperTag

- 1 Откройте **Tagname Dictionary**.
- 2 Для добавления признака к копии **SuperTag**, выполните следующее:
  - a Щелкните **New**.
  - b В блоке **Tagname**, введите точное имя вашей копии **SuperTag** с последующим ограничителем обратной наклонной чертой (\) и имя нового признака элемента.
  - c Щелкните **Type**.
  - d Выберите тип признака для нового признака элемента, который вы добавляете к копии.
  - e Щелкните **OK**. Появляется диалоговый блок с подробностями для типа признака элемента.
  - f Введите требуемые подробности, как это вы делаете для обычного признака InTouch.
- 3 Щелкните **Save**.
- 4 Щелкните **Select**.
- 5 Выберите копию SuperTag, к которой вы добавляете признак элемент.
- 6 Щелкните **OK**.

В блоке **Member List** перечисляются все признаки элемента, которые принадлежат к шаблону **SuperTag**.



В перечне появляется новый признак элемента.

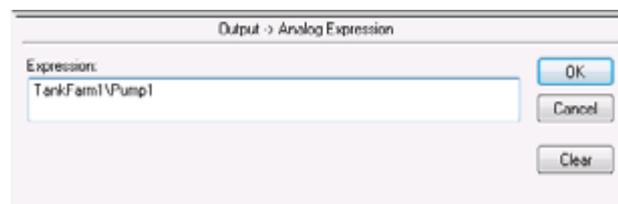
## Другие способы создания SuperTags

В дополнение к TemplateMaker, вы можете также создать SuperTags с использованием следующих методов:

- Во входном блоке выражения связи для анимации
- в пределах InTouch QuickScripts.
- Во внешнем файле, который вы загружаете в приложение с утилитой InTouch DBLoad.

**Примечание** Когда вы используете альтернативный метод для создания элемента, TemplateMaker не показывает элемент в определении шаблона SuperTag.

Когда вы создаете SuperTag посредством выражения для анимации или InTouch QuickScript, вы должны использовать действующий формат SuperTag. Например:



**Примечание** Если копия SuperTag и признак элемента, который вы указываете в выражении для анимации или QuickScript, в настоящее время не определяются, вам дается подсказка определить признак. Щелкните OK. Появляется словарь Tagname и отображает копию SuperTag и метку элемента, который вы создаете.

Правильны следующие примеры синтаксиса:

**ParentInstance\ChildMember**

**ParentInstance\ChildMember\Submember**

Следующие примеры синтаксиса неправильны:

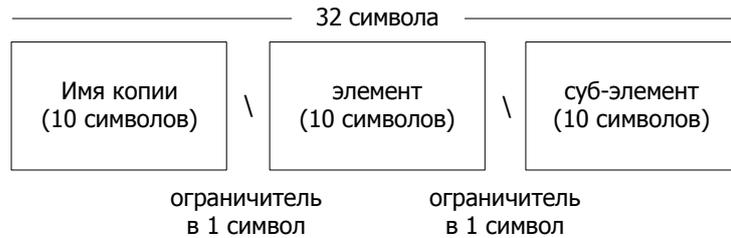
**ParentInstance\**

**ParentInstance\ChildMember\**

Если используется неправильный формат, блок сообщения об ошибке показывает, что синтаксис SuperTag содержит ошибку.

## Обращение к элементам SuperTag

Имена признака InTouch могут иметь максимум 32 символа. Каждый элемент SuperTag ParentInstance\ChildMember\Sub-member может иметь в длину до 32 символа. Максимальная длина имени признака устанавливает ограничение на обращения к SuperTags. Обращение к SuperTag может иметь максимум два шаблона (ParentInstance\ChildMember) и глубину в один элемент, как показано на следующем рисунке.



Каждый элемент в шаблоне SuperTag является доступным в стандартном формате, в настоящее время используемом для доступа к полям точек стандартных типов признака InTouch. Синтаксис обращение к SuperTag поддерживается по всему InTouch, где могут использоваться стандартные признаки. Например, правильное обращение к полю точек SuperTag - это:

**TankFarm\Tank1\Pump1RPM.RawValue**

Дистанционные обращения к признаку также поддерживают SuperTags. Например, правильное дистанционное обращение к SuperTag - это:

PLC1:"TankFarm\Tank1\Pump1RPM.RawValue"

## Заимствование SuperTags утилитой группового заимствования

Вы можете использовать утилиту группового заимствования ArchestrA для преобразования определений элемента из InTouch в объектную структуру Application Server. Утилита группового заимствования ArchestrA позволяет вам более эффективно перемещать ваши признаки InTouch в сервер приложения. В дополнение к стандартным признакам InTouch, утилите группового заимствования ArchestrA может также перемещать SuperTags. Если вы планируете перемещать SuperTags из приложения InTouch в сервер приложения, замените символ обратной наклонной черты (\) в рамках вашего обращения SuperTag на поддерживаемый символ ArchestrA типа подчеркивание (\_).

Пример:

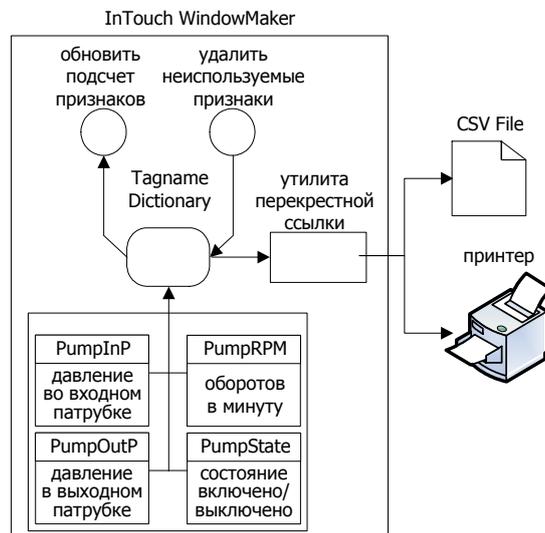
**TankFarm\_Tank1\_Pump1RPM.RawValue**

# Глава 8

## Уменьшение использования признака

Максимальное число признаков, которое вы можете использовать в приложении InTouch основывается на вашей лицензии Wonderware. Ваша лицензия Wonderware

предотвращает подсчет совпадающего признака выполняющего приложения от превышения максимального предела. Вы можете отслеживать использование признака в приложении InTouch. На рисунке ниже показано то, как Cross Reference Utility (утилита перекрестной ссылки) создает отчет об использовании признака путем анализа содержимого Tagname Dictionary (словарь имен признаков).



Вам следует поддерживать минимальное число признаков, необходимых для выполнения вашего приложения InTouch. Один способ минимизации числа ваших меток состоит в удалении неиспользуемых признаков. Вы должны обновить число признаков перед удалением неиспользуемых признаков.

## Определение использования признака

InTouch НМІ поддерживает подсчет использования для всех признаков, определенных в локальном Tagname Dictionary. Подсчет признаков не включает внутренние системные признаки. Удаленные признаки не определяются в Tagname Dictionary. InTouch инкрементирует подсчет признаков для каждой ссылки в рамках приложения для удаленного признака. Дополнительную информацию касательно того, как InTouch подсчитывает обращение к удаленным признакам смотрите в на странице 166 в разделе Максимальное число удаленных признаков на основе лицензирования. Прежде, чем вы удалите неиспользуемые признаки, убедитесь, что вы сделали следующее:

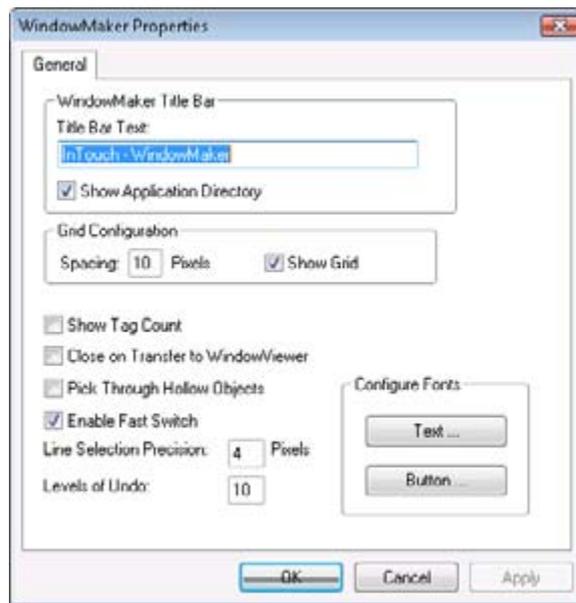
- Закройте WindowViewer.
- Обновите подсчет локальных признаков и обращения к удаленным признакам.
- Создайте отчет о признаках с перекрестными ссылками.
- Определите место использования признака в пределах приложения с помощью утилиты перекрестной ссылки.
- Сохраните выше приложение в WindowMaker.

## Определение подсчета признаков

Вы можете обновить подсчет локальных признаков, который в настоящее время определяется в Tagname Dictionary приложения. Вы можете также обносить подсчет ссылок приложения для дистанционных признаков, расположенных в другом узле.

### Для обновления отсчета локального признака

- 1 Откройте приложение InTouch application в WindowMaker.
- 2 В экране **Tools**, расширьте перечень **Configure** и выберите **WindowMaker**. Появляется диалоговый блок **WindowMaker Properties**.

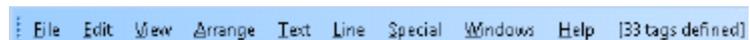


- 3 Выберите **Show Tag Count**.

Когда вы выбираете эту опцию, должен считываться весь **Tagname Dictionary** для обновления отображаемого отсчета признака. Обновления **Tagname Dictionary** могут занять много времени.

- 4 Щелкните **OK**. Появляется сообщение, что WindowMaker должен перезапускаться перед тем, как изменения конфигурации станут действовать.
- 5 Закройте WindowMaker.
- 6 Перезапустите приложение в WindowMaker.

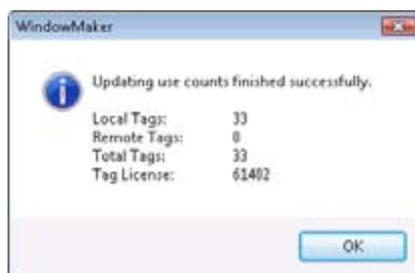
Число локальных признаков, определенных в Tagname Dictionary приложения, показаны справа от линейки меню WindowMaker.



**Для обновления суммарного отчета метки, включая дистанционные ссылки**

- 1 Откройте приложение InTouch в WindowMaker.
- 2 Закройте все открытые окна.
- 3 В меню **Special** щелкните на **Update Use Counts**. Появляется сообщение, требующее подтверждения обновления отчета использования.
- 4 Щелкните на **Yes**.

После того, как расчеты завершены, в диалоговом блоке показывается подсчет локальных признаков и ссылок на дистанционные признаки приложения.



Диалоговый блок также включает суммарное число признаков и максимальное использование признака на основе лицензии InTouch.

## Определение максимального числа дистанционных признаков на основе лицензирования

Wonderware предлагает множество опций лицензирования InTouch на основе максимального предела использования признака. В следующей таблице показаны некоторые типовые опции лицензии Wonderware для InTouch. Wonderware также предлагает больше опций лицензии, менее, чем на 60К, чем показано в таблице.

Число лицензируемых признаков	Доступные признаки	Пределы ссылки на дистанционные признаки
256	256	256 приклеивающихся признаков
500	500	500 приклеивающихся признаков
1000	1000	1000 приклеивающихся признаков
60000 (60К)	61405	60000 приклеивающихся признаков

**Примечание** Число доступных признаков меняется на основе числа имеющихся признаков в Системе для каждой версии InTouch и числа определенных групп сигнала в рамках приложения.

Число дистанционных признаков ограничивается пределом лицензии дистанционного признака, который приводится в действие только тогда, когда приложение InTouch выполняется. Например, лицензия на 256 признаков позволяет сделать 256 дистанционных ссылок в дополнение к 256 признакам,

разрешаемым лицензией. Приклеивающийся признак – это дистанционная ссылка признака, которая привязывается, когда WindowViewer компилирует приложение непосредственно перед временем выполнения. WindowViewer привязывает дистанционные ссылки на признаки во время компиляции вплоть до лицензированного предела, показанного в правом столбце вышеприведенной таблицы. WindowViewer не привязывает какие-либо дополнительные дистанционные ссылки на признаки сверх лицензионного предела. WindowViewer не уменьшает число дистанционных ссылок на признаки при закрытии окна. Число признаков привязывается для всего продолжения сеанса приложения. После того, как достигается максимальный предел приклеивающегося признака, вы должны остановить выполнение приложения, прежде, чем вы сможете открыть другие окна, которые включают одну или более дистанционных ссылок на признаки. Иначе появляется сообщение для каждой дистанционной ссылки признака, что она превышает предел лицензии для дистанционного признака. Лицензия InTouch 60К не использует числа приклеивающихся признаков, которые налагают пределы на число дистанционных ссылок на признаки. Это позволяет приложению иметь динамический доступ к более, чем 60К признаков за время работы сеанса приложения. Число использования признака для дистанционных ссылок меняется вверх и вниз при открытии и закрытии окна с дистанционными ссылками. В настоящее время лицензия InTouch 60К поддерживает 60000 признаков. Окончательное число дистанционных ссылок признака может определяться для приложения, но число активных дистанционных ссылок на признак не может превышать максимального предела. Пределы признака для лицензии 60К могут меняться в будущих версиях InTouch.

## Определение местоположения использования признаков

Используя Утилиту с перекрестными ссылками вы можете составлять сетевой отчет, который показывает использование признаков в рамках приложения InTouch. Отчет Утилиты с перекрестными ссылками показывает локальные признаки, дистанционные признаки и SuperTags в:

- связях анимации
- мастерах
- всех типах скриптов InTouch
- QuickFunctions
- активных элементах управления
- дополнительных компонентах InTouch, таких, как SPC Pro, SQL, Access Manager, и Recipe Manager.

Отчет перечисляет имя окна, в котором определяется признак, и координаты окна всех объектов, связанных с признаком. Отчет также позволяет вам просматривать QuickScript или QuickFunction, которые включают перечисленный признак.

Вы можете перебирать отчет с целью поиска определенных элементов. Групповые символы поиска позволяют вам указать более обобщенный поиск для имен признака, которые соответствуют только части строки поиска.

- Используйте звездочку (\*) в строке для поиска всех признаков, чьи имена соответствуют этим символам перед звездочкой. Например, строка поиска Prod\* находит ProdPump1 и ProdTank2.
- Используйте знак вопроса (?) в строке поиска для замены одного символа. Например, строка поиска Pump? находит Pump1 и Pump2.

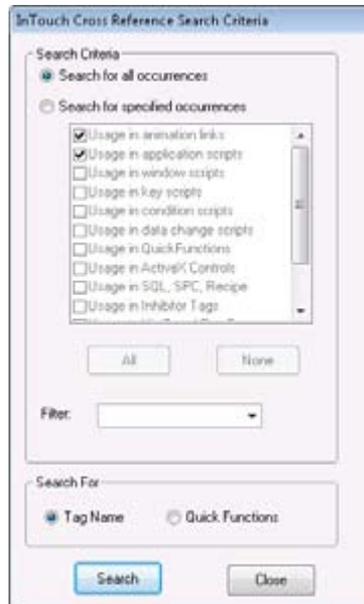
В отчете Утилиты перекрестной ссылки InTouch появляются иконки для отображения состояния и использования.

Иконка	Описание
	Признак или SuperTag определяется в словаре Tagname приложения, но не присваивается объекту
	Признак или SuperTag используется либо в связи анимации либо в InTouch QuickScript.
	Признак или SuperTag присваивается связи анимации.
	Признак или SuperTag используется в скрипте приложения.
	Показано для всех выбранных скриптов. Щелкните на имени скрипта для просмотра признака.
	Признак или SuperTag используется в скрипте Window. Щелкните на имени скрипта для просмотра признака..
	Признак или SuperTag is used in a Data Change script. Щелкните на имени скрипта для просмотра признака.
	Признак или SuperTag используется в скрипте Condition (условие). Выберите расширение вида для отображения условия скрипта и его типа. Например, \$Hour==12 On True. Щелкните на имени скрипта для просмотра признака.
	Признак или SuperTag используется в скрипте Key. Щелкните на иконке для расширения вида и отображения клавиши, присвоенной скрипту и типу скрипта. Например, F2 On Key Down.
	Признак или SuperTag используется в QuickFunction. Щелкните на иконке для расширения вида и отображения имен QuickFunctions в которых используется признак или SuperTag. Щелкните на имени скрипта для просмотра признака.

Иконка	Описание
	Признак или SuperTag используется в скрипте ActiveX Event. . Щелкните на иконке для расширения вида и отображения the ActiveX Event script.
	При перекрестной ссылке посредством Window, эта иконка предшествует имени окна, в котором используется отображаемый признак или SuperTag. Щелкните на иконке для просмотра всех признаков, используемых в окне.
	Признак или SuperTag используются в приложении SPC Pro. Щелкните на иконке для отображения имени набора данных SPC, в котором используются признак или SuperTag.
	Отображаемый признак или SuperTag используется в приложении SQL. Щелкните на иконке для отображения имени перечня связи SQL Bind List, в котором используется признак или SuperTag.
	Отображаемый признак или SuperTag используется в приложении Recipe Manager (менеджер рецепта).
	Признак используется в качестве схемы запрета сигнализации. Щелкните на иконке для отображения имен всех признаков, используемых в качестве схемы запрета сигнализации.

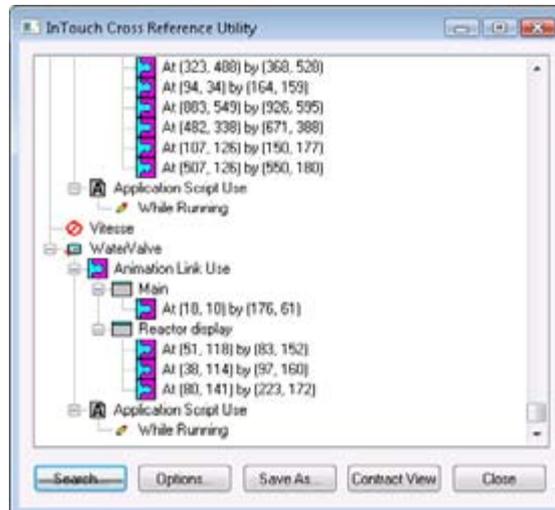
Для создания отчета с помощью Утилиты перекрестной ссылки:

- 1 Откройте приложение InTouch в WindowMaker.
- 2 В меню **Special**, щелкните на **Cross Reference**. Появляется диалоговый блок **Cross Reference Search Criteria InTouch**.



- 3 В области **Search Criteria** выберите для поиска для всех признаков или для ограничения диапазона поиска для указанных компонентов в приложении InTouch.
  - Щелкните на **Search for all occurrences** для поиска для всех появлений признаков или SuperTags.
  - Щелкните на **Search for specified occurrences** для ограничений поиска признака для конкретных компонентов приложения InTouch. Выберите блоки проверки, следующие для тех компонентов, которые следует исследовать на предмет признаков.
- 4 Введите строку в блоке **Filter** для дальнейшего ограничения поиска на предмет признаков с именами, которые соответствуют строке. Оставьте блок **Filter** пустым, если вы не хотите настроить фильтр поиска с помощью имени признака.
- 5 Щелкните правой клавишей в блоке **Filter** для открытия меню с перечнем команд, которые могут применяться к выбранному тексту.

- Щелкните в **Search** для запуска поиска с перекрестной ссылкой для признаков. После того, как поиск завершен, в отчете перечисляются все признаки, которые соответствуют критериям поиска. Если фильтр не используется, отчет показывает, все признаки, определенные в текущем словаре Tagname приложения.



## Сохранение и Распечатка перечня с перекрестными ссылками признака

Вы можете сохранить вашу распечатку признака с перекрестной ссылкой в файле и просматривать ее с помощью любого приложения, которое поддерживает формат файла .csv.

В файле листинга признака перекрестной ссылки все признаки перечисляются по имени, с которым они используются в приложении и по имени окна, в которых они располагаются. Вы можете открыть файл листинга признака с перекрестной ссылкой с помощью программы Excel или другой программы, которая поддерживает файлы .csv.

Tagname	Usage	Location
Application: C:\PROGRAMDATA\INTOUCHDEMOS\DEMOAPP1 1024		
Report generated on: 12/12/05 12:34:57		
Report is organized by tag name		
\$AccessLevel	none	
\$AlarmPrinterNoPaper	none	
\$AlarmPrinterOffline	none	
\$ApplicationChanged	none	
\$ApplicationVersion	none	
\$ChangePassword	none	
\$ConfigureUsers	none	
\$Date	none	

Вы можете также распечатать содержимое словаря Tagname. Распечатка содержимого словаря Tagname показывает вам вводы в базу данных, окна и скрипты, используемые в приложении. Вы можете указать уровень подробности, который вы хотите видеть для каждого из элементов. Вы можете отправить этот отчет на принтер или сохранить его в файле.

### Для сохранения файла перекрестной ссылки

- 1 В диалоговом блоке утилиты перекрестной ссылки InTouch (Cross Reference) щелкните **Save As**. Появляется диалоговое окно **Save As**.



- 2 Укажите имя и положение файла.
- 3 Щелкните на **Save**. Файл с употреблением признака сохраняется в указанной директории.

### Для распечатки содержимого соваля Tagname

- 1 Откройте приложение InTouch в WindowMaker.
- 2 В меню WindowMaker File щелкните Print. The появляется диалоговый блок WindowMaker Printout.



- 3 Выберите **Database Entries**, если вы хотите распечатать информацию из словаря Tagname.

Если вы выбираете Database Entries, следующие опции становятся активными:

- Выберите **Details** для включения подробностей признаков в ваш отчет.
- Выберите **Alarm Information** для включения информации о сигнализации признака в ваш отчет.
- Выберите **With Window Cross-Reference** для печати всех вводов в словарь Tagname с помощью перекрестных ссылок окна. Если вы выбираете эту опцию, укажите уровень подробности печати.

**Link Details** печатает расположение и подробности связей анимации, там где используются признак.

**Window Names Only** распечатывает только имя окон с перекрестными ссылками.

- 4 Выберите Windows для печати листинга вводов базы данных, используемых для окон приложения. Если вы выбираете Windows, укажите опции окна для печати:

- **All** печатает вводы базы данных для всех окон в приложении.
- **Selected** печатает только вводы базы данных для конкретных окон. Появляется **Windows to Print** диалоговый блок.

Выберите окна в вашем приложении, которое вы хотите распечатать и щелкните **OK**.

- **With Link Details** печатает подробности связи для окон.
- **Window Scripts** печатает скрипты, связанные с окном.
- **Database entries used in window** печатает признаки, используемые в окне.

- 5 Выберите **All Scripts** для печати всех скриптов, используемых в приложении. Вы можете ограничить печать только выбранными типами скриптов, путем очистки блока проверки **All Scripts**. Затем выберите блок проверки для каждого типа скрипта, который вы хотите распечатать.

- 6 Щелкните **Next**. Появляется диалоговый блок **Select Output Destination**.

- 7 Выберите опцию для печати содержимого словаря Tagname или отправьте вывод в текстовый файл.

- 8 Щелкните **Print**.

## Удаление неиспользуемых признаков

Вы можете удалить неиспользуемые признаки из приложения InTouch, после обновления счета использования. Вы можете удалять признаки по одному из словаря Tagname, или вы можете удалять несколько признаков одновременно. Счет признака автоматически не обновляется, когда удаляется окно, содержащее признаки, из приложения, или когда меняются признаки в скриптах связи. InTouch рассматривает признаки, как используемые, и поэтому препятствует их удалению. Счет признаков должен обновляться для удаления неиспользуемых признаков из общего количества, прежде, чем вы сможете удалить неиспользованные признаки. Дополнительную информацию касательно обновления чисел признаков смотрите в разделе Определение чисел признаков на странице 164.

**Предостережение** Признаки, которые только подаются как аварийный сигнал, не имеют подсчета использования и могут случайно удаляться. Для обеспечения того, что только признаки с аварийной сигнализацией включаются в подсчет использования, вам необходимо использовать их в окне или в QuickScript.

### Для удаления нескольких неиспользуемых признаков

- 1 Закройте WindowViewer, если она выполняется.
- 2 Откройте приложение InTouch в WindowMaker.
- 3 В меню **Special** щелкните на **Delete Unused Tags**. Появляется диалоговый блок **Choose Names to Delete** с перечнем неиспользуемых признаков.



- 4 Щелкните на одном или более имени признаков для выбора признаков для удаления. Вы можете также выбрать все перечисленные признаки, щелкая на **All**.
- 5 Щелкните **Delete**. Появляется сообщение, требующее подтверждения того, что вы хотите удалить выбранные признаки.
- 6 Щелкните **OK** для подтверждения.

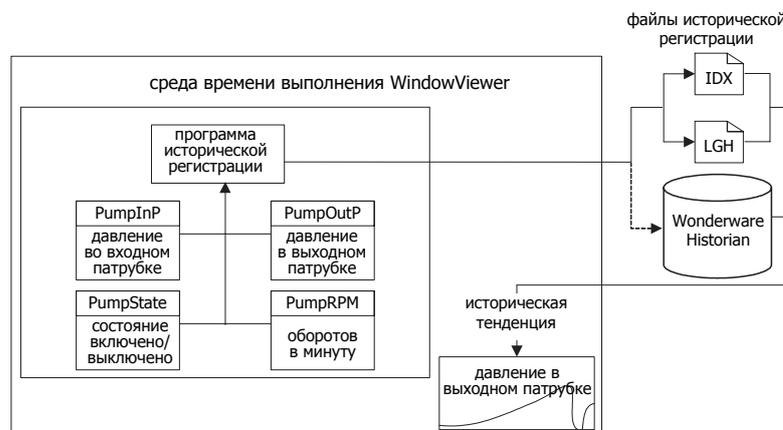
# Глава 9

## Запись значения признака

Во время выполнения приложения InTouch, значение его признаков могут регистрироваться и постоянно сохраняться. Вы можете использовать эти данные регистрации для создания графов исторической тенденции, которые показывают некоторый аспект процессов на вашем заводе со временем.

**Примечание** Вам следует использовать сервер IndustrialSQL для сохранения исторических данных InTouch для больших приложений, или когда вы хотите создать подробные отчеты. Обращайтесь к документации по серверу IndustrialSQL в части подробностей по конфигурированию исторической регистрации.

На фигуре ниже показаны данные признака, полученные с насоса, сохраненные в файлы исторической регистрации или в базу данных сервера IndustrialSQL. Программа исторической регистрации InTouch Historical Logger записывает ввод регистрации каждый раз, как значение признака меняется более, чем на его заданный диапазон чувствительности при регистрации. Программа исторической регистрации не записывает вводы с периодическим интервалом выборки.



InTouch HMI создает два файла регистрации. Один файл содержит зарегистрированные данные, сохраненные в формате собственника. Другой файл представляет собой индекс к данным.

Имена присваиваются двум файлам регистрации в следующем формате:

**ГГММДД00.LGN and ГГММДД00.IDX**

где:

- ГГ** две последние цифры года когда создаются файлы регистрации
- ММ** число из двух цифр месяца, когда создаются файлы (01-12)
- ДД** две цифры дня месяца, , когда создаются файлы (01-31)
- 00** постоянная величина 00 в именах файла регистрации

Ежедневный цикл регистрации начинается и заканчивается в полночь. Программа исторической регистрации записывает последние вводы в активные файлы регистрации в полночь и архивирует их. Для следующего дня создаются два новых файла и данные регистрируются в них. Файлы регистрации сохраняются на определенное количество дней. Файлы регистрации, которые старше, чем период сохранения, удаляются. Дополнительную информацию об установке числа дней удержания файлов регистрации смотрите раздел *Configuring General Logging Properties* (Конфигурирование общих свойств регистрации) на странице 181.

## Конфигурирование исторической регистрации

Вам необходимо выполнить три основных задачи для конфигурирования исторической регистрации для приложения InTouch:

- Сконфигурировать признаки для исторической регистрации
- Сконфигурировать общие свойства регистрации для приложения InTouch
- Дополнительно установить частоту исторической регистрации

## Конфигурирование признаков для исторической регистрации

Вы выбираете признаки для исторической регистрации из словаря Tagname. Выбранные признаки считаются активными до тех пор, пока приложение выполняется в WindowViewer. Когда значение признака меняется, программа исторической регистрации определяет, следует ли записывать ввод в регистрацию на основе зоны невосприимчивости для каждой регистрации признака и его текущего значения. Если вы меняете признак с зарегистрированного на незарегистрированный, данные, связанные с признаком больше не сохраняются в файле регистрации. Регистрация восстанавливается, когда регистрация снова разрешается. Однако, историческая тенденция показывает разрыв на период, когда регистрация была запрещена. Изменения в регистрации признака не воспринимаются, когда WindowViewer выполняет приложение. Изменения регистрации признаков не выполняются до тех пор, пока выполняющееся приложение не остановится и не будет повторно запущено. Вы конфигурируете регистрацию для каждого отдельного признака из словаря Tagname.

### Разрешение исторической регистрации для признаков

- 1 При необходимости, остановите WindowViewer из выполнения приложения.
- 2 Откройте приложение с помощью WindowMaker.
- 3 Откройте словарь Tagname.
- 4 Выберите признак из перечня словаря Tagname, чьи данные вы хотите зарегистрировать.

5 Выберите блок проверки **Log Data**.

Диалоговый блок **Tagname Dictionary** включает другие атрибуты признака, близко связанные с регистрацией:

- **Log Deadband** устанавливает порог инженерных блоков, который должен превышаться перед записью значения признака в файл регистрации. Только новые значения вне зоны невосприимчивости записываются в файл регистрации. Небольшие изменения значения в рамках диапазона невосприимчивости игнорируются.
- Свойства **Min EU** и **Max EU** масштабируют фиксированные чувствительные значения в рамках диапазона инженерных блоков. Минимальные и максимальные EU свойства устанавливают верхнюю и нижнюю границы масштабированных значений. Минимальные /максимальные инженерные блоки определяют границы диапазона для значений регистрации, показанных в тенденции. По умолчанию, историческая тенденция InTouch показывает данные регистрации из 0-100 процентов диапазона EU.

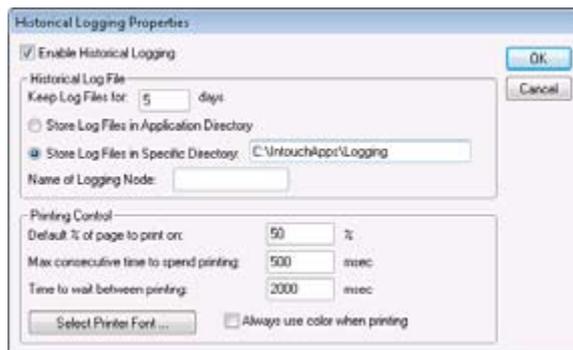
6 Щелкните на **Save**.

## 7 Повторите эти этапы для разрешения регистрации для каждого признака, данные которого вы хотите зарегистрировать.

8 Щелкните на **Close** для закрытия словаря **Tagname**, когда вы выполнили **Configuring General Logging Properties**. Вы можете установить свойства глобальной регистрации, которая применяется для выбранного приложения.

## Для конфигурирования распределенного или исторического приложения.

- 1 При необходимости, закройте приложение InTouch, работающее в WindowViewer.
- 2 Откройте WindowMaker.
- 3 При отображении **Tools**, расширьте **Configure** и выберите **Historical Logging**. Появляется диалоговый блок **Historical Logging Properties**.



Дополнительную информацию по настройке печати тенденции смотрите Печать тенденции во время выполнения на странице 239.

- 4 Выберите блок проверки **Enable Historical Logging**.
- 5 В блоке **Keep Log Files for** введите число дней перед текущим днем для сохранения файлов регистрации. Файлы регистрации сохраняются в течение текущего дня и на продолжении указанного периода сохранения. Файлы регистрации, которые старше, чем период сохранения, удаляются. Установка значения в 0 сохраняет все файлы регистрации неопределенными.

### Пример:

Установите период сохранения пять дней и начните регистрацию в первый день месяца. На седьмой день месяца файлы регистрации сохраняются за предыдущие пять дней и в текущий день (02-07). Файлы регистрации, созданные в первый день месяца, удаляются. Учитывайте использование пространства диска при установке числа дней для сохранения данных регистрации. Историческая регистрация останавливается, если на вашем жестком диске кончается свободное пространство. Вы должны освобождать пространство на диске для продолжения регистрации.

- 6 Выберите положение папки для хранения файлов регистрации. Диалоговый блок **Historical Logging Properties** включает две опции для настройки положения папки для хранения файлов регистрации.

---

**Примечание** адрес папки и имя файла для хранения данных регистрации может иметь максимум 55.

---

Выберите **Store Log Files in Application Directory** для хранения файлов регистрации в той же папке, что и приложение InTouch, формирующее данные регистрации.

Выберите **Store Log Files in specific Directory** для указания другой папки для хранения файлов регистрации. Вы можете указать папку для хранения файлов регистрации, как:

- Адрес папки Windows, такой как C:\History Log Files (файлы исторической регистрации)
- Адрес согласно универсальному соглашению по присваиванию имени (UNC), такой, как \\node\share\directory.

Если вы сохраняете файлы регистрации в распределенном узле, вы должны указать директорий, как адрес UNC. При конфигурировании на запись исторических данных в директорий приложения узла главного приложения, все узлы NAD пытаются записать исторические данные в главное приложение. Для исключения этого конфигурируйте исторические данные на каждом узле NAD для записи в логический директорий, а не в узел главного приложения.

- 7 В блоке **Name of Logging Node**, введите имя узла компьютера, выполняющего приложение InTouch, создающее данные регистрации.
- 8 Щелкните **OK** для сохранения ваших установок.

Изменения конфигурации регистрации начинают действовать сразу же. Регистрация выполняется следующий раз, как вы выполняете приложение.

## Управление частотой исторической регистрации

Вы можете дополнительно указать, что вводы регистрируются на основе двух условий:

- Прямой ввод регистрации записывается при любом изменении значения признака на величину инженерного блока, больше, чем значение зоны нечувствительности регистрации.
- Текущие значения всех зарегистрированных признаков записываются с фиксированным интервалом. Вводы для всех зарегистрированных признаков записываются независимо от их текущих значений. Фиксированные интервалы по умолчанию составляют 60 минут. Вы можете согласиться с фиксированным интервалом по умолчанию или добавить два параметра в файл `intouch.ini` для изменения интервала.

- *ForceLogging*

`ForceLogging` указывает длительность фиксированных интервалов регистрации в минутах. `ForceLogging` может быть установлен в значение между 5 и 120 минутами. Значение по умолчанию для `ForceLogging` – 60.

- *ForceLogCurrentValue*

`ForceLogCurrentValue` воздействует на вводы регистрации, которые должны записываться для всех зарегистрированных признаков, даже если текущие значения меньше, чем их диапазоны зоны нечувствительности регистрации `Log Deadband`. Значение по умолчанию `ForceLogCurrentValue=0`.

В следующем примере показан пример файла `intouch.ini`, который включает два параметра регистрации.

```
WinFullScreen=1
```

```
WinWidth=808
```

```
AlarmBufferSize=5000
```

```
ForceLogging=5
```

```
ForceLogCurrentValue=1
```

В этом примере значения признака записываются в файл исторической регистрации с интервалами в пять минут.

#### Для изменения частоты исторической регистрации

- 1 Закройте WindowMaker и WindowViewer.
- 2 Разместите файл intouch.ini в той же самой папке, что и приложение InTouch.
- 3 Отредактируйте файл intouch.ini.
- 4 Вставьте оператор ForceLogging со значением между 5 и 120.
- 5 Вставьте оператор ForceLogCurrentValue=1.
- 6 Сохраните ваши изменения и закройте файл intouch.ini.
- 7 Перезапустите WindowViewer.

## Запуск и остановка исторической регистрации во время работы

Когда приложение работает, вы можете вручную остановить и перезапустить историческую регистрацию с помощью команд из меню WindowViewer **Special**.

- Команда **Stop Historical Logging** останавливает регистрацию во время текущего сеанса приложения. Регистрация остается остановленной для текущего сеанса до тех пор, пока она не будет повторно запущена вручную.
- Команда **Restart Historical Logging** перезапускает регистрацию после ее ручной остановки с помощью опции **Stop Historical Logging**.

Вы также можете добавить кнопку к вашему приложению и написать QuickScript, который включает признак системы \$HistoricalLogging для запуска и остановки исторической регистрации. Регистрация запускается, когда для HistoricalLogging присваивается значение 1. Регистрация останавливается, когда \$HistoricalLogging присваивается значение 0. Для дополнительной информации о признаке системы \$HistoricalLogging смотрите раздел Признаки системы на странице 41.

## Использование InTouch HMI с сервером IndustrialSQL

Сервер IndustrialSQL (InSQL) – это относительная база данных в реальном времени, созданная специально для промышленных применений. Вы можете дополнительно конфигурировать Historical Logger для хранения исторических данных InTouch в базе данных сервера IndustrialSQL.

---

**Примечание** Дополнительную информацию о регистрации исторических данных InTouch в базе данных смотрите документацию по серверу IndustrialSQL. Дополнительную информацию по настройке InTouch HMI с помощью дистанционного исторического провайдера смотрите в Главе 2, распределительных приложениях (Distributing Applications) в Руководстве по управлению и расширению приложения InTouch® HMI ( InTouch® HMI Application Management and Extension Guide).

---

Если вы используете сервер IndustrialSQL для хранения исторических данных, вы должны использовать Программу управления распределенным именем (Distributed Name Manager) из WindowMaker для указания связи на базу данных.

### Для конфигурирования связи с базой данных сервера IndustrialSQL

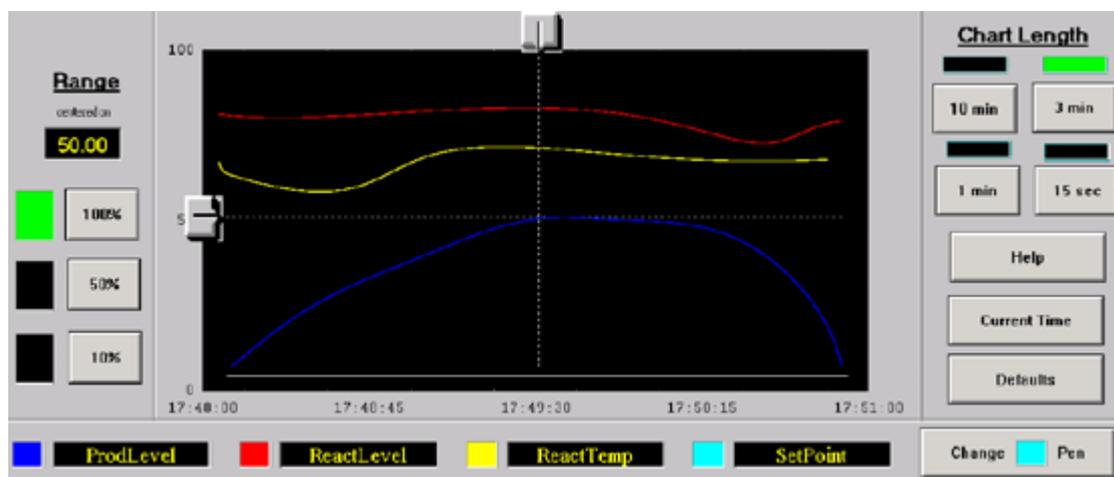
- 1 Откройте WindowMaker.
- 2 На экране **Tools** расширьте перечень конфигурации (**Configure list**).
- 3 Выберите **Distributed Name Manager**. Появляется диалоговый блок **Distributed Name Manager**.
- 4 В таблице **Distributed History** введите InSQL в качестве нового провайдера в блоке **Provider Name**.
- 5 Выберите **InSQL Provider**.
- 6 Щелкните на **Configure InSQL Provider**. Появляется диалоговый блок **SQL History Provider Properties**.
  - a В блоке **Data Source** введите имя узла сервера, где устанавливается сервер IndustrialSQL.
  - b Введите имя абонента пользователя базы данных сервера IndustrialSQL.
  - c Введите пароль для абонента пользователя как в блоке **Password**, так и в блоке **Re-enter password**.
  - d Щелкните на **Test** для проверки соединения с базой данных сервера IndustrialSQL. Появляется сообщение, показывающее успешно ли образуется связь с базой данных или нет.
- 7 Щелкните на **OK** для закрытия диалогового блока.
- 8 Щелкните на **OK** для закрытия диалогового блока **InSQL History Providers**. В перечне **History Providers** появляется узел сервера IndustrialSQL.

- Щелкните на **ОК** для закрытия диалогового блока Distributed Name Manager.

# Глава 10

## Анализ тенденций в данных по признаку

Вы можете построить тенденции, которые графически показывают данные полученные с приложения InTouch. WindowMaker включает набор утилит и мастеров, которые позволяют вам создавать исторические тенденции и тенденции в реальном времени. На следующем рисунке показан пример тенденции InTouch Average/Scatter (усреднения/ рассеивания).



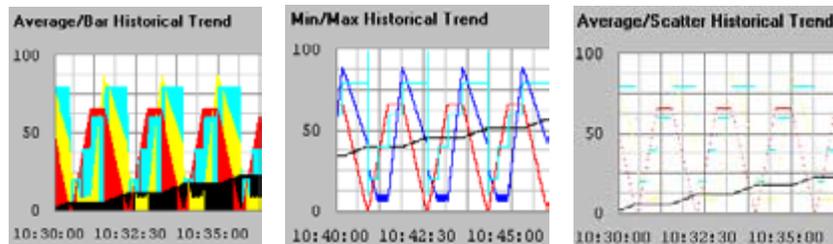
Вы можете также использовать набор органов управления тенденцией. Используя эти органы управления, вы можете выбрать данные, показанные в тенденции и то, как эти данные появляются в тенденции. Вы можете конфигурировать тенденцию в реальном времени и историческую тенденцию. Оба типа тенденции включают опции конфигурирования для настройки интервалов сбора данных для тенденции и визуального представления.

## Типы тенденций InTouch

Историческая тенденция отображает данные регистрации, собранные в прошлом и сохраненные в архивах данных InTouch. Используя распределенную историческую систему, вы можете восстановить исторические данные из файла исторической регистрации InTouch, расположенного на доступном сетевом узле. Историческая распределенная система расширяет восстановительные возможности исторических тенденций по включению отдаленных баз данных регистрации. Тенденции в реальном времени постоянно обновляются для отображения данных в том виде, как они появляются за относительно короткие промежутки времени. Вы можете использовать средство WindowMaker's Real-time Trend для создания объекта тенденции в окне. Если устанавливается дополнительное средство 16-Pen Trend, вы можете также создавать тенденции в реальном времени, которые отображают данные с 16 признаков.

## Осмысление исторических тенденций

Исторические тенденции показывают непрерывный сегмент данных в прошлом. В отличие от тенденций в реальном времени, исторические тенденции обновляются только посредством скрипта или действиями оператора. Исторические тенденции показывают графическое представление данных максимум от восьми признаков. Вы присваиваете данные, которые появляются в исторической тенденции путем присваивания признака перу тенденции. На рисунке ниже показаны три типа исторических тенденций InTouch.



- Историческая тенденция Average/Bar показывает среднее значение точки данных за временные интервалы в виде прямоугольников.
- Историческая тенденция Min/Max показывает изменения в процентной доле шкалы инженерных блоков в виде вертикальной линии за охваченное время. Упор делается на ход времени и на скорость изменения, нежели на величину изменения.
- Историческая тенденция Average/Scatter показывает среднее значение точек данных за каждый интервал времени тенденции.

Вы можете создать графические полосковые элементы интерфейса, называемые скутерами для доступа к подробностям данных по тенденции на основе текущего положения скутера в пределах тенденции. Например, когда оператор размещает скутер над зоной на тенденции, которая имеет видимые данные, отображаются время и значения в этом положении для всех значений базы данных, для которых определяется тенденция.

Вы можете также создавать клавиши или полосковые элементы интерфейса для увеличения или уменьшения масштаба между скутерами или для данных, таких, как от максимальной до минимальной величины. Среднее и стандартное отклонение может быть показано для полного графика или для области между скутерами. Исторические тенденции также можно просмотреть за любое время. Вы можете создать специализированные шкалы и связать их с полями точек MinEU и MaxEU для создания тенденции, которая показывает полный диапазон набора данных посредством их инженерного блока.

## Осмысление тенденций в реальном времени

Тенденция в реальном времени показывает данные из приложения InTouch, которое в настоящее время выполняется. Тенденции в реальном времени постоянно обновляются. Тенденции в реальном времени графически отображаются в виде текущих значений данных, связанных максимально с четырьмя локальными признаками или выражениями.

Вы можете:

- Создать тенденцию в реальном времени
- Выбрать признаки для тенденции
- Указать интервал времени и интервал обновления тенденции
- Сконфигурировать опции дисплея для тенденции

## Демонстрация сохраненных значений признака в исторической тенденции

Вы можете создать исторические тенденции путем использования любого из следующего Утилит WindowMaker:

- средства Historical Trend (историческая тенденция)
- Historical Trend Wizard (мастер исторической тенденции)
- 16-Pen Trend Wizard (Optional) (мастер тенденции на 16 перьев) (дополнительно)

Дополнительно, вы можете включить тенденцию ActiveFactory для отображения данных исторической тенденции InTouch, сохраненной в базе данных сервера IndustrialSQL.

## Использование объектов исторической тенденции

Вы можете создать и сконфигурировать тенденцию с помощью средства WindowMaker Historical Trend. Вы можете

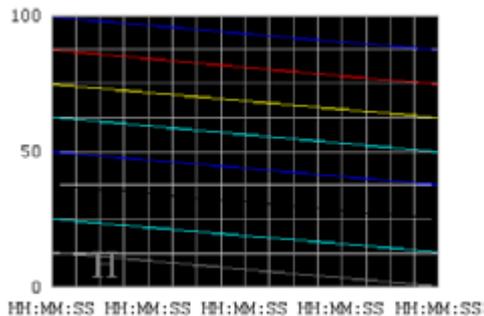
- Создать историческую тенденцию
- Выбрать признаки для тенденции
- Указать промежуток времени и интервал обновления тенденции
- Сконфигурировать опции дисплея для тенденции

### Создание исторической тенденции

Вы можете использовать средство Historical Trend для создания объекта тенденции в окне. Первый раз, когда вы создаете объект исторической тенденции, используются установки по умолчанию для InTouch. После того, как вы сконфигурировали историческую тенденцию, WindowMaker использует последние значения конфигурации в качестве исходных значений для нового объекта тенденции. Вы можете нарисовать график тенденции любого размера в пределах границ окна.

#### Для создания исторической тенденции

- 1 Откройте окно в WindowMaker, в котором вы хотите разместить историческую тенденцию.
- 2 Щелкните кнопку **Historical Trend** в линейке инструментов рисования **Drawing Toolbar**.
- 3 Переместите мышь по области окна, в которой вы хотите разместить историческую тенденцию. Протащите мышь по диагонали для создания прямоугольника нужного размера для тенденции. В окне появляется объект **Historical Trend**.



- 4 При необходимости, настройте высоту и ширину тенденции с помощью манипуляторов объекта.

## Конфигурирование того, какие признаки показывать из исторической тенденции

Перо исторической тенденции создает графическое представление зарегистрированных данных за указанный период. Вы присваиваете перо тенденции для признаков, которые собирали исторические данные. Историческая тенденция поддерживает до восьми перьев.

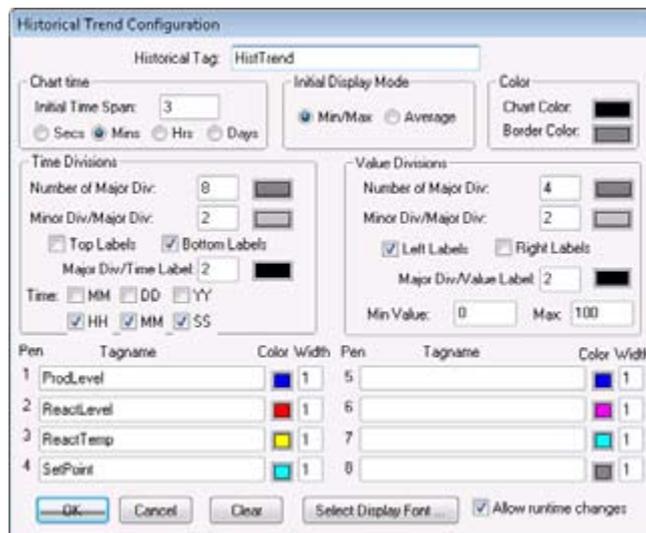
**Примечание** WindowViewer должен быть закрыт. Иначе, нельзя выбрать блоки Pen.

Вы можете выбрать признаки из дистанционных провайдеров истории, если провайдеры сконфигурированы. Информацию о настройке дистанционного провайдера истории смотрите в главе 2, Распределяющие приложения (Distributing Applications), в Руководстве управления и расширения приложения InTouch® HMI

**Примечание** Вы можете также конфигурировать провайдер истории сервера IndustrialSQL для визуализации данных исторической тенденции. Для большей универсальности и других опций графопостроения, используйте средства тенденции ActiveFactory для создания тенденций с помощью исторических данных InTouch, сохраненных в базе данных сервера IndustrialSQL.

Для конфигурирования того, какие признаки отображать из исторической тенденции

- 1 Дважды щелкните на объекте тенденции в окне. Появляется диалоговый блок **Historical Trend Configuration**.

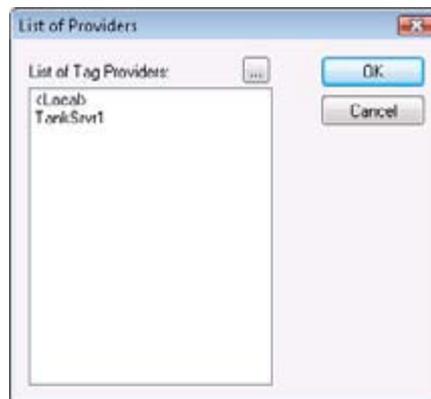


- 2 В блоке **Historical Tag** введите признак, который вы хотите использовать для тенденции.

Признак должен определяться как тип Hist Trend. Вы должны присвоить разный признак Hist Trend tag для каждой исторической тенденции в приложении InTouch. Если признак, который вы вводите, в настоящее время не определен в словаре

Tagname, появляется диалоговый блок и задается вопрос, хотите ли вы создать признак. Если вы выбираете ОК для определения признака, появляется диалоговый блок Tagname Dictionary.

- 3 В области **Tagname** укажите имя существующего локального или дистанционного признака в одной или более блоков **Pen**.
- 4 Для присваивания непосредственно существующего локального или дистанционного признака, щелкните в блоке **Pen** и введите имя признака.
- 5 Для просмотра признака для присваивания:
  - a Дважды щелкните на блоке **Pen**. Появляется диалоговый блок **List of Providers**.



- b Выберите тот провайдер признака, который вы хотите использовать для пера.
  - c Щелкните **OK** для отображения диалогового блока с листингом признаков для выбранного провайдера.
  - d Дважды щелкните на признаке из перечня для выбора его.
- 6 Дважды щелкните на цветном блоке, следующем за каждым пером, присвоенным признаку, для отображения палитры цветов. Щелкните на цвете для пера.
- 7 В блоке **Width**, введите ширину линии в пикселях для каждого пера, показанного в тенденции.
- 8 Повторите этапы от 3 до 6 для каждого признака или выражения, которое вы хотите присвоить перу исторической тенденции.
- 9 Если нужно, выберите блок проверки **Allow runtime changes** для разрешения оператору конфигурировать историческую тенденцию при работе приложения. Дополнительную информацию об обновлении исторической тенденции во время выполнения смотрите в разделе Изменение конфигурации во время выполнения на странице 196.

## Конфигурирование продолжительности времени и интенсивности обновления исторической тенденции

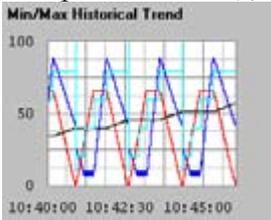
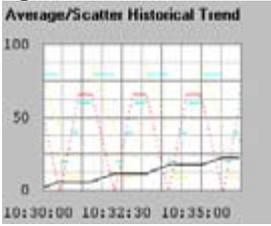
Вы можете конфигурировать продолжительность времени и интенсивность обновления исторической тенденции.

### Для конфигурирования продолжительности времени и интенсивности обновления исторической тенденции

- 1 если необходимо, дважды щелкните на объекте тенденции для отображения диалогового блока **Historical Trend Configuration**.
- 2 В области **Chart Time** введите продолжительность времени в **Initial Time Span**, которое вы хотите, чтобы появлялось на горизонтальной оси тенденции.
- 3 Выберите единицу измерения времени: секунды (Secs), минуты (Mins), часы (Hrs), или дни (Days).

Например, если вы вводите 8 в **Initial Time Span** и, затем, выбираете **Hrs**, продолжительность времени тенденции отображается на протяжении 8 часов.

- 4 В области **Initial Display Mode** выберите тип исторической тенденции, который появляется, когда WindowViewer изначально показывает окно, содержащее тенденцию.

Исходный режим индикации	Описание
Min/Max	<p>Каждый пиксель на графике показывает текущее значение инженерного блока связанное с признаком в это время в рамках исторической тенденции</p> 
Среднее	<p>Каждый пиксель сегмента времени тенденции показывает среднее значение признака за период времени в пределах сегмента</p> 

- 5 Перейдите к Конфигурированию опций дисплея исторической тенденции на странице 194 для конфигурирования визуального проявления исторической тенденции.

## Конфигурирование опций дисплея исторической тенденции

Вы можете конфигурировать визуальное проявление исторической тенденции.

### Для конфигурирования опций дисплея исторической тенденции

- 1 При необходимости дважды щелкните на объекте тенденции. Появляется диалоговый блок **Historical Trend Configuration**.
- 2 Установите опции цвета. Сделайте следующее:
  - В области **Color** щелкните в блоке **Chart Color** для открытия палитры цветов.
  - Щелкните на цвете в палитре в качестве фонового для тенденции. По умолчанию цвет фона белый. Любой другой цвет увеличивает время, необходимое для печати тенденции.
  - Выберите **Border Color** для открытия палитры цветов.
  - Щелкните на цвете в палитре в качестве цвета границы для тенденции.
- 3 Установите опции делений времени. Выполните следующее:
  - В области Time Divisions введите число главных делений времени тенденции в Number of Major Div.  
Появляются главные деления времени на горизонтальной оси времени тенденции. Максимальное время между главными делениями времени составляет 65536 секунд или 18 часов, 12 минут, 16 секунд.
  - Выберите цвет для линий основного деления.
  - Из Minor Div/Major Div введите число малых делений времени в пределах главного деления времени. Число малых делений времени равно четному кратному числу от главного деления. Например, если главное деление устанавливается в 60 секунд, ввод значения 2 в Minor Div/Major Div установит малое деление времени в 30 секунд.
  - Выберите цвет для линий малого деления.
  - Выберите Top Labels или Bottom Labels для указания размещения временных меток на тенденции.
  - Если вы используете временные метки, введите число временных меток на одну линию главного деления в Major Div/Time Label.
  - Выберите цвет временных меток деления.
  - Выберите временные модули, показанные, как метка основного временного деления.

Месяцы (MM)	Часы (HH)
Дни (DD)	Минуты (MM)
Годы (YY)	Секунды (CC)

- 4 В области **Value Divisions** сконфигурируйте вид вертикальной оси тенденции.

Опции **Value Divisions** конфигурируются таким же образом, что и опции **Time Divisions**. Вертикальная ось указывает диапазон значений данных, которые появляются в тенденции, на основе инженерных блоков для всех признаков.

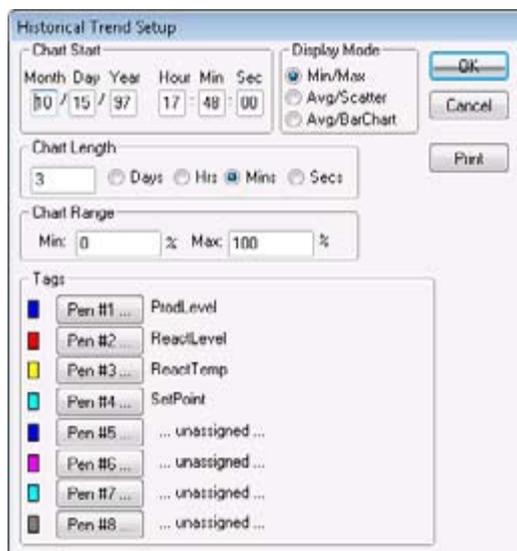
- 5 Щелкните на ОК для сохранения ваших изменений конфигурации и закрытия диалогового блока **Historical Trend Configuration**.

## Изменение конфигурации тенденции во время выполнения

Если вы выбираете опцию *Allow runtime changes*, когда вы конфигурируете историческую тенденцию, операторы могут сконфигурировать историческую тенденцию во время ее выполнения. Операторы конфигурируют тенденцию в диалоговом блоке, который появляется после выбора тенденции в отображаемом окне.

### Для конфигурирования исторической тенденции во время выполнения

- 1 Щелкните на исторической тенденции при ее выполнении. Появляется диалоговый блок **Historical Trend Setup**.



- 2 В области **Chart Start** введите дату и время запуска интервала сбора данных исторической тенденции.

- 3 В области **Display Mode** выберите тип графика исторической тенденции.

Режим отображения тенденции оказывает влияние на работу. Первичным фактором, который определяет работу тенденции, является длина линий, показанных в тенденции. Чем длиннее линии, тем больше времени требуется для формирования тенденции. Ширина линии также влияет на работу. Широкие линии требуют для прорисовки значительно больше времени. Тенденции Min/Max или Average/Scatter могут создаваться более быстро, чем график Average/Bar Chart.

- 4 В области **Chart Length** введите продолжительность времени, которая должна отображаться на тенденции и, затем, выберите единицу измерения. Например, если вы вводите 2 и выбираете Hrs, продолжительность тенденции составит 2 часа.
- 5 В области **Chart Range** введите процентную долю диапазона инженерных блоков, показанную по вертикальной оси исторической тенденции. Шкала тенденции представляет собой сегмент диапазона инженерного блока признака при анализе тенденции, определенный диапазоном процентной доли. Диапазон значений от 0 до 100. Например, если вы хотите провести анализ тенденции для дисперсии выбранных признаков от 40 до 60 процентов от их диапазона для инженерного блока, введите 40 и 60 в блоках диапазона Min и Max, соответственно.
- 6 В области **Tags** щелкните на номере пера для присваивания признака. Появляется диалоговый блок **Select Tags** с перечнем признаков, которые могут присваиваться перу исторической тенденции.
- 7 Используйте следующий оператор в QuickScript или кнопку для разрешения оператору обновления графика:

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

- 8 Используйте любую из следующих функций в QuickScript или по нажатию клавиши:

```
HTUpdateToCurrentTime(Hist_Tag);
```

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

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

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

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

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

For more information about using scripts containing trend functions, see Controlling a Historical Trend Wizard Using Scripts on page 219.

- 9 Измените любое из следующих полей точек признака в тенденции:

`.ChartStart`

`.ChartLength`

`.MaxRange`

`.MinRange`

`.Pen1-.Pen8`

Дополнительную информацию об использовании полей точек с историческими тенденциями смотрите в разделе Controlling a Historical Trend Using Dotfields (управление исторической тенденцией с использованием полей точек) на странице 198.

## Управление исторической тенденцией с использованием полей точек

Вы можете использовать поля точек для управления исторической тенденцией во время выполнения.

### **.DisplayMode Dotfield (поле точек режима дисплея)**

Поле точек `.DisplayMode` указывает формат тенденции, используемый для отображения значений признака.

#### **Category (категория)**

Историческая

#### **Usage (использование)**

`tag_name.DisplayMode`

#### **Parameter (параметр)**

*tag\_name*

Any Hist Trend tag (любой признак Hist Trend).

#### **Data Type (тип данных)**

аналоговый (чтение/запись) .

#### **Valid Values (действующие значения)**

1 = отображает мин/макс значение, которые проявляются в каждом периоде выборки (по умолчанию).

2 = отображает среднее значение каждого периода выборки в исторической тенденции разброса.

3 = отображает среднее для каждого периода выборки в исторической тенденции в виде гистограммы.

#### **Example (пример)**

Этот оператор поля точек указывает значения признака PumpPress и форматируется в виде исторической тенденции, как гистограмма.

`PumpPress.DisplayMode=3;`

**See Also (Смотрите также)**

.ChartLength, .ChartStart

**.MinRange Dotfield ( поле точек .MinRange)**

Поле точек .MinRange указывает минимальную процентную долю диапазона инженерного блока признака для отображения для каждого признака в тенденции Historical.

**Category (категория)**

Historical (историческая).

**Usage (использование)**

**tag\_name.MinRange**

**Parameter (параметр)**

*tag\_name*

Любой признак, присвоенный перу исторической тенденции.

**Remarks (замечания)**

Историческая тенденция может отобразить различные типы признаков в одно и то же время. Указание минимальной и максимальной границ диапазона значений в инженерных блоках сложно, так как различные типы признаков могут иметь различные инженерные диапазоны. Поэтому минимальная и максимальная значения диапазона выражаются в виде процентной доли от инженерного диапазона для каждой метки. Таким образом, независимо от истинного инженерного диапазона признака, историческая тенденция показывает означенную процентную долю этого конкретного инженерного диапазона признака.

**Data Type (тип данных)**

Real (read/write) (действительное (чтение/запись)).

**Valid Values (действующие значения)**

Пределы для полей точек .MaxRange и .MinRange имеют значения от 0 до 100. .MinRange всегда меньше, чем .MaxRange. Если вы присваиваете значение меньше 0 или больше 100 для одного из этих полей точек, величина фиксируется на значении 0 или 100. Если .MinRange больше, чем или равно .MaxRange, тенденция не показывает каких-либо данных.

**Example (пример)**

В этом примере оператор поля точек устанавливает минимальный процентный диапазон исторической тенденции в 25 процентов возможного диапазона для инженерного блока для признака PumpPress. PumpPress.MaxRange=25

**See Also (смотрите также)**

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

**.MaxRange Dotfield (поле точек MaxRange)**

Поле точек `.MaxRange` указывает максимальную процентную долю диапазона инженерного блока признака для отображения каждого признака в исторической тенденции.

**Category (категория)**

Историческая.

**Usage (использование)**

`tag_name.MaxRange`

**Parameter (параметр)**

*tag\_name*

Любой признак, присвоенный перу исторической тенденции.

**Remarks (замечания)**

Историческая тенденция может отображать многие типы признаков одновременно. Указание минимальной и максимальной границ диапазона в инженерных блоках может оказаться сложным из-за того, что признаки могут иметь различные инженерные диапазоны. Поэтому максимальные и минимальные величины диапазона выражаются, как процентная доля инженерного диапазона для каждого признака. Так, независимо от истинного инженерного диапазона признака, историческая тенденция показывает обозначенную процентную долю этого инженерного диапазона признака.

**Data Type (тип данных)**

Real (read/write) (действительное (чтение/запись))

**Valid Values (Действующие значения)**

Пределы для полей точек `.MaxRange` и `.MinRange` имеют значения от 0 до 100. `.MinRange` всегда меньше, чем `.MaxRange`. Если вы присваиваете значение меньше 0 или больше 100 для одного из этих полей точек, величина фиксируется на значении 0 или 100. Если `.MinRange` больше, чем или равно `.MaxRange`, тенденция не показывает каких либо данных.

**Example (пример)**

В этом примере оператор поля точек устанавливает минимальный процентный диапазон исторической тенденции в 75 процентов возможного диапазона для инженерного блока для признака `PumpPress`. `PumpPress.MaxRange=75`

**See Also (смотрите также)**

`.ChartStart`, `.ChartLength`, `.DisplayMode`, `.EngUnits`, `.MinEU`, `.MaxEU`, `.MinRange`, `.MinRaw`, `.MaxRaw`, `.RawValue`

## .UpdateCount Dotfield

Поле точек .UpdateCount инкрементирует отсчет каждый раз, как обновляется историческая тенденция. Поле точек .UpdateCount может использоваться, как устройство запуска для последующих функций.

### Category (категория)

Историческая.

### Usage (использование)

HistTrendTag.UpdateCount

### Parameter (параметр)

*HistTrendTag*

Признак HistTrend присвоенный имени тенденции.

### Data Type (тип данных)

Целое (только для чтения).

### Valid Values (действующие значения)

Любое положительное целое.

### Example (пример)

В этом примере используется функция HTGetValueAtScooter() для восстановления значения Pen1 в правом текущем положении скутера. Изменение любого аргумента функции вызывает повторную оценку функции. Когда обновление завершается и значение .UpdateCount инкрементируется, этот оператор повторно оценивается. MyRealTag=HTGetValueAtScooter

```
MyHistTrendTag, MyHistTrendTag.UpdateCount, 2,
MyHistTrendTag.ScooterPosRight, 1, "PenValue");
```

### See Also (смотрите также)

.UpdateInProgress, .UpdateTrend

## **.UpdateInProgress Dotfield**

Поле точек `.UpdateInProgress` показывает текущее состояние операции обновления исторической тенденции. Значение поля точек устанавливается в 1, если историческое восстановление выполняется в данный момент; в противном случае поле точек устанавливается в 0.

### **Category (категория)**

Историческая.

### **Usage (использование)**

#### **HistTrendTag.UpdateInProgress**

### **Parameter (параметр)**

*HistTrendTag*

Признак `HistTrend` присвоенный имени тенденции.

### **Remarks (замечания)**

В любом случае когда запрашиваются новые данные из исторической тенденции, значение поля точек устанавливается в 1. После того, как процесс завершается, `.UpdateInProgress` устанавливается в 0. `.UpdateInProgress` может использоваться в функциях, связанных с историческими тенденциями. Если оператор прокручивает тенденцию за период, выходящий за пределы текущего показанного периода, ему может понадобиться некоторое время для восстановления исторических данных. Поле точек `.UpdateInProgress` дает способ предупредить оператора, что требуемые данные восстанавливаются. Без обратной связи оператор не может знать, что тенденция обновляется.

### **Data Type (тип данных)**

Дискретный (только чтение).

### **Value Values (действующие значения)**

0 = Не проводится обновления

1 = Обновление проводится

### **Example (пример)**

Поле точек `.UpdateInProgress` как правило, используется, как выражение в связи видимости на текстовом объекте вблизи или на кнопках прокрутки исторической тенденции. Вы можете использовать поле точек `.UpdateInProgress` для отображения "Busy" (занят) в окне, когда данные восстанавливаются со следующей связью анимации дисплея значения сообщения:

**DText (HistTrend1.UpdateInProgress, "Busy", "Ready")**

### **See Also (смотрите также)**

`.UpdateCount`, `.UpdateTrend`

## **.UpdateTrend Dotfield**

Поле точек `.UpdateTrend` запускает обновление исторической тенденции. Используя поле точек `.UpdateTrend` в скрипте действия кнопки, оператор может вручную обновить тенденцию во время выполнения.

### **Category (категория)**

Историческая.

### **Usage (использование)**

`HistTrendTag.UpdateTrend`

### **Parameter (параметр)**

*HistTrendTag*

Признак `HistTrend` присвоенный имени тенденции.

### **Remarks (замечания)**

Исторические тенденции не обновляются автоматически. Изменение должно выполняться в полях точек либо `.ChartStart`, либо `.ChartLength` для обновления графика и демонстрации текущего значения для указанных признаков. Используя это поле точек в скрипте действия кнопки, оператор может обновит график Вовремя выполнения. Вы можете также использовать это поле точек в QuickScript, если предполагается изменять поля точек, связанные с исторической тенденцией. Вам следует только установить `.UpdateTrend dotfield` в значение 1.

### **Data Type (тип данных)**

Дискретный (только запись).

### **Valid Values (действующие значения)**

1

### **Example (пример)**

В этом примере запускается историческая тенденция, связанная с признаком `MyHistTrendTag`, для обновления текущими значениями всех параметров.

```
MyHistTrendTag.UpdateTrend=1 ;
```

## .ChartLength Dotfield

.Поле точек ChartLength dotfield указывает продолжительность времени, показанную в исторической тенденции.

### Category (категория)

Историческая.

### Usage (использование)

**HistTrendTag.ChartLength**

### Parameter (параметр)

*HistTrendTag*

Признак HistTrend присвоенный имени тенденции.

### Remarks

Значение, присвоенное .ChartLength, указывает продолжительность графика в секундах. Продолжительность определяется, как продолжительность времени, показанная в настоящее время на графике исторической тенденции. Более конкретно, вычисление, возвращаемое, как продолжительность графика из графика исторической тенденции – это:

$\text{ChartLength} = (\text{штамп Дата/Время на правой стороне графика}) - (\text{штамп Дата/Время на левой стороне графика});$

Так как штампы Дата/время выражаются в секундах от полночи 1 января 1970, результат вычислений представляется в секундах, отображаемый между левой и правой сторонами графика.

Всегда при добавлении или вычитании из .ChartLength, время выражается в секундах. Поэтому, для вычитания двух часов из текущего .ChartLength, преобразуйте часы в секунды перед выполнением вычислений. Например: (2 часа) \* (60 минут/час) \* (60 секунд/минута) = 7200 секунд.

### Data Type (тип данных)

Целое (чтение/запись).

### Valid Values (действующие значения)

Любое положительной целое.

### Examples (примеры)

В этом примере устанавливается продолжительность исторической тенденции в один час.

```
HtTag.ChartLength=3600 {60 минут * 60 секунд/минута};
```

This example scrolls the trend left by 50 percent.

```
HtTag.ChartStart=HtTag.ChartStart -
```

```
HtTag.ChartLength / 2;
```

This example scrolls the chart left by 10 percent.

```
HtTag.ChartStart=HtTag.ChartStart - (.10 *
```

```
HtTag.ChartLength);
```

**See Also (смотрите также)**

.ChartStart

**.ChartStart Dotfield**

Поле точек .ChartStart может использоваться для настройки или проверки значения штампа даты/времени запуска (левая сторона) исторической тенденции.

**Category (категория)**

Историческая.

**Usage (использование)**

**HistTrendTag.ChartStart**

**Parameter (параметр)**

*HistTrendTag*

Признак HistTrend присвоенный имени тенденции.

**Remarks (замечания)**

Это поле точек чтение/запись используется для настройки или проверки значения штампа даты/времени запуска исторической тенденции. Поле точек .ChartStart выражается, как число прошедших секунд с полуночи 1 января 1970. Точка отсчета определяется, как первый штамп на исторической тенденции.

**Data Type (тип данных)**

Целое (чтение/запись).

**Valid Values (действующие значения)**

Любое положительное целое.

**Example (пример)**

Следующий оператор выполняет прокрутку графика направо на одну минуту.

```
HtTagname.ChartStart=HtTagname.ChartStart + 60;
```

**See Also (смотрите также)**

.ChartLength

## .Pen1-8 Dotfields

Поле точек .Pen1-8 присваивает зарегистрированный признак перу исторической тенденции.

### Category (категория)

Историческая

### Usage (использование)

```
HistTrendTag{ .Pen1 | .Pen2 | .Pen3 | .Pen4 | .Pen5 |
    .Pen6 | .Pen7 | .Pen8};
```

### Parameter

*HistTrendTag*

Признак HistTrend присвоенный имени тенденции.

### Remarks (замечания)

Вы присваиваете признаки перьям тенденции используя поля точек .Pen1-8 со следующим форматом.:

```
HistTrend.PenX = Tag_Name.TagID
```

Где X -это целое от 1 до 8.

Рекомендуется, чтобы вы использовали, при возможности, функции HTSetPenName() и HTGetPenName().

---

**Примечание** Только локальные признаки могут присваиваться для поля точек .PenX. Нельзя использовать запись provider.tag. Запись provider.tag может использоваться только с функцией HTSetPenName(). Хорошая ссылка по использованию при изучении того, как это поле точек работает, представляет собой мастер исторической тенденции, размещенный на экране и раздвинутый в сторону.

---

### Data Type (тип данных)

Идентификатор признака (чтение/запись).

### Valid Values (действующие значения)

Этот тип данных поля точек является типом TagID (идентификатор признака). Это значит, что только идентификатор признака может присваиваться полям точек .Pen1-8. Вы не можете непосредственно присвоить имя признака полям точек .Pen1-8. Вы должны связывать связанное поле точек .TagID признака с полем точек .Pen1-8 с использованием следующего синтаксиса:

```
HistTrendTag.Pen1=LoggedTag.TagID;
```

В общем случае, признак типа TagID может приравниваться только другому признаку

TagID. Он не может использоваться с каким либо другим типом признака, кроме как если расширение поля точек .TagID добавляется к другому признаку. Хотя поля точек .Pen1-8 рассматриваются, как типа чтение/запись, их значения не могут непосредственно на экране.

### Examples (примеры)

В следующем примере присваивается новый признак полю точек `.Pen5` исторической тенденции, связанной с признаком `HistTrend`. Вы должны прилагать поле точек `.TagID` к имени зарегистрированного признака с целью присваивания его полю точек `.Pen5`.

```
HistTrendTag . Pen5= PumpPress . TagID ;
```

Работая с предыдущим примером, вы можете показать имя признака, присвоенное для `HistTrendTag.Pen5`. Создание легенды, которая показывает признак, присвоенный каждому перу тенденции, является полезной информацией для оператора. Вы не можете показать признак, присвоенный `HistTrendTag.Pen5` в связи дисплея сообщения (`Message Display`). Фактическое значение поля точек `.Pen1` является целым, которое представляет местоположение запоминающего устройства в `WindowViewer`, которые не используются для целей дисплея. Вам необходимо создать новый признак типа `TagID`, называемый `Pen05`. Поместите следующий оператор под оператором из предыдущего примера:

```
Pen05=HistTrendTag . Pen5 ;
```

В первом примере признак `PumpPress` присваивается перу `5` в `HistTrendTag`. В этом примере, значение `Pen05` присваивается значению `Pen5` в `HistTrendTag`, которое сейчас равно `TagID` из признака `PumpPress`. Поля точек `.Pen1-8` являются указателями на признаки, которые присваиваются в помощью перьев, выбранных для проявления в тенденции. Поля точек `.Pen1-8` имеют специальный тип данных, а именно `.TagID`. Вам нужен признак-поддержка для сокращения разрыва между системой дисплея и исторической системой. Путем присваивания значения `HistTrendTag.Pen5` признаку `Pen05` типа `TagID`, вы присвоили `MyLoggedTag` признаку `Pen05`. Фактически, вы могли бы выполнить ту же задачу со следующей строкой программы:

```
Pen05=MyLoggedTag . TagID ;
```

### **.TagID Dotfield**

Поле точек `.TagID` может использоваться в совокупности с полями точек `.Pen1` - `.Pen8` для присваивания признака перу исторической тенденции.

#### **Category (категория)**

Исторический признак.

#### **Usage (использование)**

`tag_name.TagID`

#### **Parameter (параметр)**

*tag\_name*

Любой дискретный, целый, действительный, косвенный дискретный, или косвенный аналоговый признак.

#### **Remarks (замечания)**

Поле точек `.TagID` дает идентификатор признака и используется, главным образом, для присваивания признака перьям в исторической тенденции.

#### **Data Type (Тип данных)**

TagID (только чтение).

#### **Example (пример)**

В этом примере используется поле точек `.TagID` для присваивания признака `PumpRPM` перу 6 из исторической тенденции.

**HistTrendTag.Pen6=PumpRPM.TagID ;**

#### **See Also (смотрите также)**

`.Pen1-.Pen8`

## **.ScooterLockLeft Dotfield**

Поле точек `.ScooterLockLeft` указывает, может ли оператор перемещать правый скутер дальше влево, чем текущее положение левого скутера на исторической тенденции.

### **Category (категория)**

Историческая

### **Usage (использование)**

`HistTrendTag.ScooterLockLeft`

### **Parameter (параметр)**

*HistTrendTag*

Признак `HistTrend` присвоенный имени тенденции.

### **Remarks (замечания)**

В общем случае, вам следует предотвращать оператора от перемещения правого скутера дальше влево, чем текущее положение левого скутера. Когда левый скутер фиксируется, это заставляет положение правого скутера равняться положению левого скутера, в любом случае, когда правый скутер доходит до левого скутера.

### **Data Type (тип данных)**

Discrete (read/write) (дискретный (чтение/запись)).

### **Valid Values (действующие значения)**

0 = False (ложное). Правый скутер может двигаться дальше влево, за текущее положение левого скутера на исторической тенденции.

1 = True (истинное) Правый скутер не может двигаться дальше влево, за текущее положение левого скутера на исторической тенденции

### **Example (пример)**

В следующем примере предотвращается перемещение правого скутера дальше влево, за текущее положение левого скутера на исторической тенденции.

```
HistTrendTag.ScooterLockLeft=1;
```

### **See Also (смотрите также)**

`.ScooterPosRight`, `.ScooterPosLeft`, `.ScooterLockRight`

## **.ScooterLockRight Dotfield**

Поле точек `.ScooterLockRight` указывает, может ли оператор перемещать левый скутер дальше вправо, чем текущее положение правого скутера на исторической тенденции.

### **Category (категория)**

Историческая

### **Usage (использование)**

`HistTrendTag.ScooterLockRight`

### **Parameter (параметр)**

*HistTrendTag*

Признак `HistTrend` присвоенный имени тенденции.

### **Remarks (замечания)**

В общем случае, вам следует предотвращать оператора от перемещения левого скутера дальше вправо, чем текущее положение правого скутера. Когда правый скутер фиксируется, это заставляет положение левого скутера равняться положению правого скутера, в любом случае, когда левый скутер доходит до правого скутера.

### **Data Type (тип данных)**

дискретный (чтение/запись).

### **Valid Values (действующие значения)**

0 = False (ложное). Левый скутер может двигаться дальше вправо, за текущее положение правого скутера на исторической тенденции.

1 = True (истинное) Левый скутер не может двигаться дальше вправо, за текущее положение правого скутера на исторической тенденции

### **Example (пример)**

В следующем примере предотвращается перемещение правого скутера дальше вправо, за текущее положение правого скутера на исторической тенденции.

```
HistTrendTag.ScooterLockRight=1 ;
```

### **See Also (смотрите также)**

`.ScooterPosRight`, `.ScooterPosLeft`, `.ScooterLockLeft`

## .ScooterPosLeft Dotfield

Поле точек .ScooterPosLeft динамически управляет положением левого скутера на исторической тенденции.

### Category (категория)

Историческая

### Usage (использование)

**HistTrendTag.ScooterPosLeft**

### Parameter (параметр)

*HistTrendTag*

Признак HistTrend присвоенный имени тенденции.

### Remarks (замечания)

Это поле точек с чтением/записью динамически управляет положением левого скутера. Вы можете использовать это поле точек в функции QuickScript для восстановления текущего положения левого скутера или вы можете присваивать значение этому полю точек для настройки положения левого скутера для другого положения в тенденции. Это поле точек чаще всего используется в совокупности с набором функций HTGetValue(). Эти функции должны указывать, какая историческая тенденция запрашивается, а также текущее положение скутеров тенденции.

### Data Type (тип данных)

Действительное (чтение/запись)

### Valid Values (действующие значения)

От 0.0 до 1.0; где 0.0 - это предельно левый график исторической тенденции и 1.0 – предельно правый график исторической тенденции.

### Examples (примеры)

В следующем примере выполняется перестановка левого скутера. Левый скутер перемещается в положение 34 процента от суммарной длины графика от левой стороны графика исторической тенденции, связываемого с признаком MyHistTrendTag tag.

```
MyHistTrendTag.ScooterPosLeft=.34;
```

В следующем операторе функция QuickScriptHTGetValueAtScooter() восстанавливает значение пера 1 в текущем положении левого скутера. Изменение любого значения в рамках перечня аргумента функции вызывает повторное вычисление функции. Каждый раз, как положение левого скутера меняется, этот оператор повторно вычисляется.

```
MyRealTag=HTGetValueAtScooter  
    (MyHistTrendTag,MyHistTrendTag.UpdateCount, 1,  
MyHistTrendTag.ScooterPosLeft, 1, "PenValue");
```

### See Also (смотрите также)

.ScooterPosRight, .ScooterLockLeft, .ScooterLockRight

## **.ScooterPosRight Dotfield**

Поле точек `. ScooterPosRight` динамически управляет положением правого скутера на исторической тенденции.

### **Category (категория)**

Историческая

### **Usage (использование)**

`HistTrendTag.ScooterPosRight`

### **Parameter (параметр)**

*HistTrendTag*

Признак `HistTrend` присвоенный имени тенденции.

### **Remarks (замечания)**

Это поле точек с чтением/записью динамически управляет положением правого скутера. Вы можете использовать это поле точек в функции `QuickScript` для восстановления текущего положения правого скутера или вы можете присваивать значение этому полю точек для настройки положения правого скутера для другого положения в тенденции. Это поле точек чаще всего используется в совокупности с набором функций `HTGetValue()`. Эти функции должны указывать, какая историческая тенденция запрашивается, а также текущее положение скутеров тенденции.

### **Data Type (тип данных)**

Действительное (чтение/запись)

### **Valid Values (действующие значения)**

От 0.0 до 1.0; где 0.0 - это предельно левый график исторической тенденции и 1.0 – предельно правый график исторической тенденции.

### **Examples (примеры)**

Следующий оператор указывает новое положение левого скутера. Правый скутер перемещается в положение 34 процента от суммарной длины графика от левой стороны графика исторической тенденции, связываемого с признаком `MyHistTrendTag tag`.

```
MyHistTrendTag.ScooterPosRight=.34;
```

В следующем операторе используется функция `HTGetValueAtScooter()``QuickScript` для восстановления значения пера 1 в новом положении правого скутера. Каждый раз, как меняется положение правого скутера, этот оператор повторно вычисляется.

```
MyRealTag=HTGetValueAtScooter (MyHistTrendTag,
```

```
MyHistTrendTag.UpdateCount, 2,
```

```
MyHistTrendTag.ScooterPosRight, 1, "PenValue");
```

### **See Also (смотрите также)**

`.ScooterPosLeft`, `.ScooterLockLeft`, `.ScooterLockRight`

## Использование мастера исторической тенденции

Мастер исторической тенденции автоматически создает историческую тенденцию. В отличие от ручного присваивания признаков перьям исторической тенденции, мастер автоматически конфигурирует историческую тенденцию, используя стандартные значения. На нижеприведенном рисунке показана стандартная тенденция, созданная с помощью мастера исторической тенденции. Тенденция включает ползунки, называемые скутерами, для отображения данных в конкретном месте на графике тенденции или для увеличения масштаба в выбранном диапазоне данных тенденции.



Для добавления функций масштаба и перемещения или органов управления пером к исторической тенденции, используйте мастера Zoom/Pan Panel и Trend Pen Legend тенденции.

Вы можете создавать и конфигурировать историческую тенденцию. Вы можете:

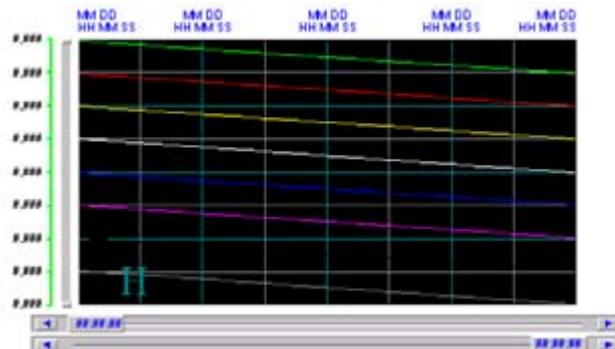
- Создавать историческую тенденцию с помощью мастеров.
- Выбирать признаки для тенденции
- Сконфигурировать временной интервал исторической тенденции
- Управлять тенденцией с помощью QuickScripts

## Создание тенденции с помощью мастера исторической тенденции

Вы можете создать стандартную историческую тенденцию с использованием автоматических функций мастера исторической тенденции. В других разделах описывается то, как вручную сконфигурировать историческую тенденцию путем использования опций мастера.

### Для создания исторической тенденции с помощью мастеров

- 1 откройте окно из WindowMaker для размещения исторической тенденции.
- 2 на линейке меню WindowMaker щелкните на иконке Wizard. Появляется диалоговый блок Wizard Selection.
- 3 Выберите Trends из перечня мастеров. Правая панель диалогового блока Wizard Selection отображает набор иконок мастеров тенденции.
- 4 Выберите мастер Hist Trend with Scooters и щелкните на OK. Диалоговый блок Wizard Selection закрывается и повторно появляется ваше окно.
- 5 Переместите курсор в место окна, куда вы хотите поместить верхний левый угол исторической тенденции. Щелкните для помещения тенденции в окно.



- 6 Дважды щелкните на тенденции. Появляется диалоговый блок Historical Trend Chart Wizard.

- Щелкните на **Suggest**. Мастер графика исторической тенденции автоматически присваивает значения конфигурации по умолчанию тенденции. Единственной оставшейся задачей по конфигурации является присваивание признаков перьям тенденции.

## Конфигурирование того, какие признаки отображать на графике тенденции

Присваивание признаков перьям тенденции в мастере графика исторической тенденции аналогично процедуре для средств в реальном времени и историческим средствам.

### Для присваивания признаков из мастера графика исторической тенденции

- Если нужно, дважды щелкните на исторической тенденции. Появляется диалоговый блок **HistoricalTagSource1Trend Chart Wizard**.



- Введите имя существующего локального признака или выражения в блок **Pen**.

**Примечание** WindowViewer должен закрываться. Иначе, блоки Pen не будут выбираться.

Если вы дважды щелкните в блоке Pen, появляется диалоговый блок Select Tag с перечнем признаков, присвоенным опции Log Data для приложения. Вы можете присвоить признак перу путем выбора его из диалогового блока Select Tag.

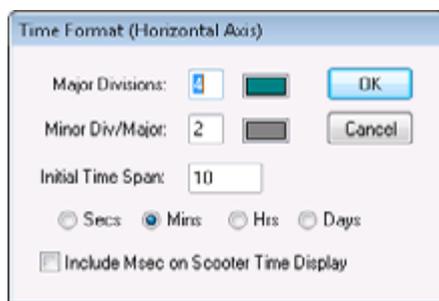
- Щелкните на цветном блоке, следующем за каждым пером и выберите другой цвет, если вы хотите изменить цвет пера по умолчанию. Иначе, пропустите этот этап и примите цвет по умолчанию.
- Щелкните **OK** для закрытия диалогового блока **Historical Trend Chart Wizard**.

## Конфигурирование временного интервала исторической тенденции

Диалоговый блок Historical Trend Chart Wizard включает опцию для ручного конфигурирования продолжительности времени, отображаемой из тенденции, созданной с помощью мастера исторической тенденции. Вы можете вручную конфигурировать время тенденции, вместо принятия конфигурации по умолчанию мастера исторической тенденции.

### Для конфигурирования продолжительности времени исторической тенденции

- 1 Дважды щелкните на исторической тенденции. Появляется диалоговый блок **Historical TrendChart Wizard**.
- 2 Щелкните на **Times**. Появляется диалоговый блок **Time Format**.



- 3 Сконфигурируйте формат времени. Выполните следующее:
  - a В блоке **Major Divisions box** введите число главных делений времени, показанных на горизонтальной оси времени тенденции.
  - b В блоке **Minor Div/Major**, введите число малых делений времени в пределах каждого главного деления.
  - c В блоке **Initial Time Span** введите длину периода времени, показанного на горизонтальной оси тенденции. Тенденции, созданные с помощью мастера исторической тенденции, могут обновляться при выполнении приложения в WindowViewer. Операторы могут менять длину периода времени тенденции. Но историческая тенденция всегда запускается с временным периодом, установленным в диалоговом блоке **Time Format**.
  - d Выберите единицу измерения для периода времени тенденции в секундах, минутах, часах и днях. Движки скутера показывают начальный и конечный период времени на основе значений, установленных в опциях **Initial Time Span** и **Time**.
  - e Дополнительно, включите миллисекунды в дисплей времени скутера. В следующем примере показан движок скутера с миллисекундами, добавленными к текущему времени.

11:30:13.000

- 4 Щелкните **OK** для закрытия диалогового блока **Time Format**.

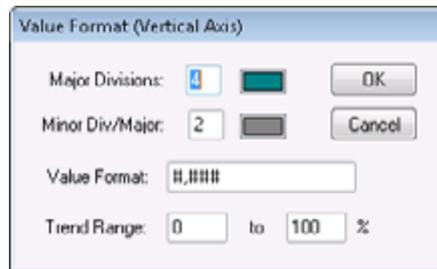
- 5 Щелкните **OK** для закрытия диалогового блока **Historical Trend Chart Wizard**.

## Конфигурирование опций дисплея

Диалоговый блок Historical Trend Chart Wizard включает опцию по конфигурированию вертикальных модулей исторической тенденции. Вы можете вручную сконфигурировать величину главного и малого делений, показанных на вертикальной оси тенденции.

**Для конфигурирования опций дисплея с помощью мастера графика исторической тенденции**

- 1 Если необходимо, дважды щелкните на исторической тенденции. Появляется диалоговый блок **Historical Trend Chart Wizard**.
- 2 Щелкните **Values**. Появляется диалоговый блок **Value Format** с опциями для конфигурирования вертикальной оси тенденции.



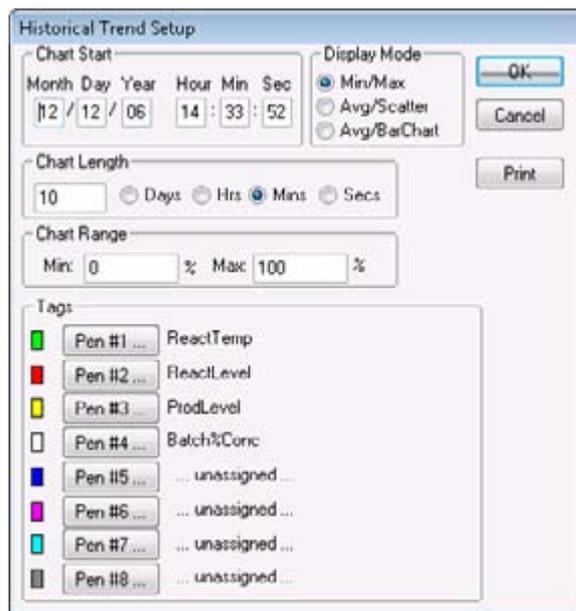
- 3 Сконфигурируйте формат значения. Выполните следующее:
  - a В блоке **Major Divisions** введите число делений с главным значением, показанных на вертикальной оси тенденции. Щелкните на цветном блоке для доступа к палитре цветов и, затем, щелкните на цвете, который вы присваиваете линиям делений оси с главной величиной.
  - b В блоке **Minor Div/Major** введите число малых делений, которые вы хотите, чтобы были видимыми в каждом главном делении оси значений. Щелкните на блоке цвета для доступа к палитре цветов, и, затем, щелкните на цвете, который вы присваиваете линиям главного оси значений.
  - c В блоке **Value Format** введите формат чисел, которые появляются в вертикальной оси значений тенденции. Формат чисел по умолчанию - #,###.
  - d В блоках **Trend Range** введите верхнюю и нижнюю границы в процентах инженерных блоков признака, которые появляются в тенденции.
- 4 Щелкните **OK** для закрытия диалогового блока **Value Format**.
- 5 Щелкните **OK** для закрытия диалогового блока **Historical Trend Chart Wizard**.

## Смена конфигурации во время выполнения

Если вы выбираете опцию *Allow runtime changes*, когда вы конфигурируете вашу историческую тенденцию, оператор может сменить некоторые аспекты исторической тенденции во время выполнения.

### Для конфигурирования исторической тенденции во время выполнения

- 1 Щелкните на тенденции в WindowViewer. Появляется диалоговый блок **Historical Trend Setup**.



- 2 В зоне **Chart Start** введите дату начала и время графика.
- 3 В области **Display Mode** выберите тип исторической тенденции.
- 4 В области **Chart Length** введите продолжительность времени, отображаемого на тенденции, и, затем, выберите инкремент времени для продолжительности.
- 5 В области **Chart Range** введите процентную долю масштаба в виде вертикального диапазона тенденции.
- 6 В области **Tags** щелкните на каждом **Pen#** для присваивания признака перу тенденции. Появляется диалоговый блок **Select Tag** и отображает те признаки, для которых разрешена регистрация.
- 7 Дважды щелкните на имени признака для присваивания его перу тенденции.
- 8 Щелкните на **OK** для сохранения изменений во время выполнения в тенденции.

## Управление мастером исторической тенденции с использованием скриптов

Вы можете использовать функции QuickScript с объектами тенденции или выражениями связи анимации для управления исторической тенденцией во время выполнения. Например, вы можете использовать QuickScripts для всязи перьев графиком, добавления новых событий к графику, удаления или повторного построения сетки или удаления или повторного построения скутеров.

### Обновление тенденции под текущее время

Функция UpdateToCurrentTime() (обновление под текущее время) восстанавливает и отображает данные с конечным временем, равным текущему времени. Время начала равно времени конца минус ширину графика.

#### Функция обновления под текущее время (UpdateToCurrentTime() )

Функция UpdateToCurrentTime() восстанавливает и отображает данные с конечным временем, равным текущему времени. Время начала равно времени конца минус ширину графика.

#### Категория

Историческая

#### Синтаксис

**HTUpdateToCurrentTime (Hist\_Tag) ;**

#### Аргумент

*Hist\_Tag*

Признак HistTrend, присвоенный имени исторической тенденции.

#### Пример

Следующие операторы восстанавливают и отображают данные для исторического признака Trend1 в текущее время:

**HTUpdateToCurrentTime ("Trend1") ;**

Если текущее время равно 3:04 PM и ширина тенденции равна 60 секунд, новое конечное время равно 3:04 PM. Новое время начала тенденции равно 3:03 PM.

## Изменение конфигурации тенденции

Вы можете использовать эти функции скрипта для изменения признаков, присвоенных перьям исторической тенденции:

- функция HTSelectTag()
- функция HTSetPenName()

### Функция HTSelectTag()

Функция HTSelectTag() открывает диалоговый блок Select Tag оператора для присваивания другого признака перу тенденции.

---

**Примечание.** В диалоговом блоке Select Tag перечисляются только признаки, которые определяются для исторической регистрации с опцией Log Data, выбранной в словаре Tagname.

---

#### Категория

Историческая

#### Синтаксис

**HTSelectTag ( ) ;**

#### Remarks

Функция HTSelectTag() отображает только признаки, в которых опция Log Data была выбрана из словаря Tagname. Однако, возможно использовать фильтр просмотра признака для отображения более малого набора признаков. Например, все признаки, которые начинаются с "A". Функция возвращает выбранный признак и может использоваться в качестве параметра признака для присваивания признака перу.

#### Пример

Следующий QuickScript выводит диалоговый блок Select Tag в WindowViewer. Пользователь может затем выбрать признак из перечня. Этот признак присваивается перу 1 историческим объектом (Historical Object) с именем HistTrend.

```
HTSetPenName ("HistTrend" , 1 , HTSelectTag ( ) ) ;
```

#### Смотрите также

HTSetPenName()

### Функция HTSetPenName()

Функция HTSetPenName() присваивает другой признак перу тенденции.

#### Категория

Историческая

#### Синтаксис

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

**Аргументы***Hist\_Tag*

Признак HistTrend, присваивающий имя тенденции.

*PenNum*

Признак Целое или значение, представляющее номер пера (1-8) тенденции.

*TagName*

Имя нового признака для присваивания перу.

**Замечания**

Эта функция QuickScript является только методом добавления признаков из распределенного провайдера истории во время выполнения. Вы можете увидеть следующее сообщение об ошибке, когда вы пытаетесь снять присвоение с пера тенденции:

```
VIEW /UpdateData: Invalid DBS.TAGNAME handle - 0 (VIEW /UpdateData: Неправильный идентификатор DBS.TAGNAME - 0) .
```

Эта ошибка происходит, если вы пытаетесь снять присвоение с пера, которое было ранее присвоено дистанционному признаку в виде histprovider.tag\_name. Для разрешения этой ошибки, создайте локальный признак с помощью выбранной опции Log Data. Затем используйте следующий скрипт для отмены присваивания пера:

```
HTSetPenName( "HistTrend", 1, "localtag" );  
{assigns the pen to a locally logged tag---localtag  
(присваивает перо локально зарегистрированному  
признаку---localtag) }  
HistTrend.Pen1=None;  
{unassigns the pen (снимает присваивание с пера) }
```

Где None – это признак типа TagID.

**Примеры**

Следующий оператор присваивает признак OutletPressure перу 3 тенденции 1.

```
HTSetPenName("Trend1", 3, "OutletPressure (давление в  
выходном патрубке)");
```

Следующий оператор присваивает признак HistPrv1.Tag1 перу TrendPen4 тенденции Trend1.

```
HTSetPenName("Trend1", TrendPen4, "HistPrv1.Tag1");
```

**Смотрите также**

HTSelectTag()

## Восстановление информации о тенденции и исторических данных

Вы можете создавать скрипты, которые восстанавливают информацию из исторической тенденции при ее выполнении. Используйте следующие функции:

- HTGetPenName() Функция
- HTGetTimeAtScooter() Функция
- HTGetTimeStringAtScooter() Функция
- HTGetValue() Функция
- HTGetValueAtScooter() Функция
- HTGetValueAtZone() Функция
- HTScrollLeft() Функция
- HTScrollRight() Функция
- HTZoomIn() Функция
- HTZoomOut() Функция

### HTGetPenName() Функция

Функция HTGetPenName() возвращает имя признака, присваиваемого в настоящее время, указанному номеру пера исторической тенденции.

#### Категория

Исторический

#### Синтаксис

```
MessageResult=HTGetPenName (Hist_Tag, UpdateCount, PenNum) ;
```

#### Arguments

*Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*UpdateCount*

Целое, представляющее поле точек .UpdateCount тенденции. Значение аргумента действует, как пусковой фактор смены данных для повторного расчета функции

*PenNum*

Целый признак или значение, представляющее номер пера (1-8) тенденции.

**Пример**

Следующий оператор восстанавливает имя признака, присвоенного перу Pen 2 тенденции Trend1 и помещает имя в признак сообщения TrendPen:

```
TrendPen=HTGetPenName ("Trend1", Trend1.UpdateCount,2) ;
```

**Функция HTGetTimeAtScooter()**

Функция HTGetTimeAtScooter() возвращает время в секундах после 00:00:00 часов времени по Гринвичу, 1 января, 1970 года для выборки в положении скутера, указанного с помощью аргументов ScootNum и ScootLoc.

**Категория**

Историческая

**Синтаксис**

```
IntegerResult=HTGetTimeAtScooter (Hist_Tag, UpdateCount, ScootNum, ScootLoc) ;
```

**Аргументы**

*Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*UpdateCount*

Целое, представляющее поле точек .UpdateCount тенденции.

*ScootNum*

Целое, представляющее левый или правый скутер:

1=Левый скутер

2=Правый скутер

*ScootLoc*

Действительное число, представляющий значение в положениях .ScooterPosRight или .ScooterPosLeft на тенденции.

**Замечания**

Любые изменения значений, присвоенные аргументам UpdateCount, ScootNum и ScootLoc вызывают выражения, которые должны рассчитываться. Это обеспечивает то, что выражения рассчитываются после того, как восстанавливаются новые данные или после того, как перемещается скутер.

**Пример**

Следующий оператор восстанавливает время в секундах для значения в текущем положении левого скутера тенденции Trend1:

```
HTGetTimeAtScooter ("Trend1", Trend1.UpdateCount, 1, Trend1.ScooterPosLeft) ;
```

## Функция HTGetTimeStringAtScooter()

Функция HTGetTimeStringAtScooter() возвращает строку, содержащую время/дату для выборки в указанном положении скутера.

### Категория

Историческая

### Синтаксис

```
MessageResult=HTGetTimeStringAtScooter(Hist_Tag,
    UpdateCount, ScootNum, ScootLoc, Format_Text);
```

### Аргументы

*Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*UpdateCount*

Целое, представляющее поле точек .UpdateCount тенденции.

*ScootNum*

Целое, представляющее левый и правый скутер.

1=Левый скутер

2=Правый скутер

*ScootLoc*

Действительное число, представляющий значение в положениях .ScooterPosRight или .ScooterPosLeft на тенденции.

*Format\_Text*

Строка, указывающая формат времени/данных для использования. Приемлемы следующие строки Format\_Text:

```
"Date", "Time", "DateTime", "DOWShort"
(Wed, например), и "DOWLong" (Например
Wednesday, (среда)).
```

### Замечания

Любые изменения значений, присвоенные аргументам UpdateCount, ScootNum и ScootLoc вызывают выражения, которые должны рассчитываться. Это обеспечивает то, что выражение рассчитывается после того, как восстанавливаются новые данные или после того, как перемещается скутер.

### Пример

Следующий оператор восстанавливает время и дату для значения в текущем положении правого скутера тенденции Trend1. Значение сохраняется в признаке сообщения NewRightTimeString и имеет формат "Time":

```
NewRightTimeString=HTGetTimeStringAtScooter
("Trend1", Trend1.UpdateCount, 2,
Trend1.ScooterPosRight, "Time");
```

## Функция HTGetValue()

Функция HTGetValue() возвращает значение запрашиваемого типа указанного типа тенденции.

### Категория

Исторический

### Синтаксис

```
RealResult=HTGetValue (Hist_Tag, UpdateCount,
    PenNum, ValType_Text) ;
```

### Аргументы

*Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*UpdateCount*

Целое, представляющее поле точек .UpdateCount тенденции.

*PenNum*

Признак или значение целое, представляющее номер пера (1-8) тенденции.

*ValType\_Text*

Строка, показывающая тип возвращаемого значения:

PenAverageValue = среднее для полной тенденции.

PenMaxValue = максимальное значение пера для полной тенденции.

PenMinValue = минимальное значение пера для полной тенденции.

PenMaxEU = Максимальное значение инженерных блоков для полной тенденции.

PenMinEU = минимальное значение инженерных блоков для полной тенденции.

PenStdDev = стандартное отклонение для полной тенденции.

### Замечания

Действительный признак запоминающего устройства возвращается, представляя рассчитанное значение данного типа.

### Пример

Следующий оператор получает стандартное отклонение для данных пера 2, восстановленных из тенденции PumpPress. Значение сохраняется в действительном признаке запоминающего устройства LeftHemisphereSD:

```
LeftHemisphereSD=HTGetValue ("PumpPress" ,
    PumpPress.UpdateCount, 2, "PenStdDev") ;
```

## Функция HTGetValueAtScooter()

Функция HTGetValueAtScooter() возвращает значение запрашиваемого типа для выборки в указанном положении скутера, тенденции и номера пера. Аргумент UpdateCount вызывает выражение, которое должно рассчитываться после завершения обработки функции.

### Категория

Историческая

### Синтаксис

```
RealResult=HTGetValueAtScooter(Hist_Tag, UpdateCount,
    ScootNum, ScootLoc, PenNum, ValType_Text);
```

### Аргументы

*Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*UpdateCount*

Целое, представляющее поле точек .UpdateCount тенденции.

*ScootNum*

Целое, представляющее левый и правый скутер:

1=Левый скутер

2=Правый скутер

*ScootLoc*

Действительное число, представляющий значение в полях точек .ScooterPosRight или .ScooterPosLeft на тенденции.

*PenNum*

Признак или значение целое, представляющее номер пера (1-8) тенденции.

*ValType\_Text*

Строка, показывающая тип возвращаемого значения:

PenValue = Значение в положении скутера.

PenValid = 0, если значение неправильно, 1 если правильно.

Когда аргумент ValType\_Text используется с функцией HTGetValueAtScooter(), используйте один из перечисленных действующих типов.

### Замечания

Действительный признак запоминающего устройства возвращается, представляя вычисленное значение для “PenValue.” Дискретный признак запоминающего устройства возвращается, представляя значение для “PenValid.”

**Пример**

Следующая функция возвращает 1, если значение является реальной выборкой, или 0, если значение является неправильным для пера 3 тенденции Trend1 для текущего положения правого скутера:

```
HTGetValueAtScooter ("Trend1", Trend1.UpdateCount,
    2, Trend1.ScooterPosRight, 3, "PenValid");
```

**Функция HTGetValueAtZone()**

функция HTGetValueAtZone() возвращает значение запрашиваемого времени для данных, расположенных между правым и левым положениями скутера для указанного пера тенденции.

**Категория**

Историческая

**Синтаксис**

```
RealResult=HTGetValueAtZone (Hist_Tag, UpdateCount,
    Scoot1Loc, Scoot2Loc, PenNum, ValType_Text);
```

**Аргументы**

*Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*UpdateCount*

Целое, представляющее поле точек .UpdateCount тенденции. Оно используется только, как запускающий механизм для расчета функции.

*Scoot1Loc*

Действительное число, представляющий значение в поле точек .ScooterPosLeft на тенденции. Оно используется только, как запускающий механизм для расчета функции.

*Scoot2Loc*

Действительное число, представляющий значение в поле точек .ScooterPosRight на тенденции. Оно используется только, как запускающий механизм для расчета функции.

*PenNum*

Признак или значение целое, представляющее номер пера (1-8) тенденции.

*ValType\_Text*

Строка, показывающая тип возвращаемого значения:

PenAverageValue = Среднее для зоны меду скутерами.

PenMaxValue = максимальное значение для зоны меду скутерами

PenMinValue = минимальное значение для зоны меду скутерами

PenMaxEU = максимальное значение инженерного блока для зоны между скутерами.

PenMinEU = минимальное значение инженерного блока для зоны между скутерами

PenStdDev = стандартное отклонение для зоны между скутерами.

#### Замечание

Величина действительного признака запоминающего устройства возвращается, представляя вычисленное значение данного типа. Указанные постоянные значения для аргументов Scoot1Loc и Scoot2Loc не оказывают воздействия и используются только для запуска функции расчета. Функция использует непосредственно значение точки поля .ScooterPosLeft и .ScooterPosRight признака тенденции, независимо от значения, которое вы указываете для аргументов Scoot1Loc и Scoot2Loc.

#### Пример

Следующий оператор вычисляет среднее значение для даты между правым и левым скутерами тенденции Trend1 для пера 1. Значение сохраняется в действительном признаке запоминающего устройства AvgValue:

```
AvgValue=HTGetValueAtZone ("Trend1" ,
    Trend1.UpdateCount, Trend1.ScooterPosLeft,
    Trend1.ScooterPosRight, 1, "PenAverageValue") ;
```

## Панорамирование и изменение масштаба тенденции

Вы можете создать QuickScripts, содержащий функции, которые выбирают конкретные данные из исторической тенденции во время выполнения.

### Функция HTScrollLeft()

Функция HTScrollLeft() устанавливает время начала тенденции в более раннее время, нежели текущее время начала посредством процентной доли времени продолжительности всей тенденции. Результат состоит в прокрутке графика налево в более раннее время путем указания процентной доли времени продолжительности всей тенденции.

#### Категория

Историческая

#### Синтаксис

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

**Аргументы***Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*Percent*

Действительное число, представляющее процентную долю от продолжительности времени графика для прокрутки влево (0.0 to 100.0).

**Пример**

Следующий оператор прокручивает время/дату влево на 10процентов от суммарной ширины тенденции PumpPress:

```
HTScrollLeft ("PumpPress" , 10 . 0 ) ;
```

Если текущий дисплей начинается при 12:00:00 PM и ширина дисплея составляет 60 секунд, затем начинается новая тенденция при 11:59:54 AM после того, как обрабатывается функция.

**Функция HTScrollRight()**

Функция HTScrollLeft() устанавливает время начала тенденции в более позднее время, нежели текущее время начала посредством процентной доли времени продолжительности всей тенденции. Результат состоит в прокрутке графика вправо в более позднее время путем указания процентной доли времени продолжительности всей тенденции.

**Категория**

Историческая

**Синтаксис**

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

**Аргументы***Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*Percent*

Действительное число, представляющее процентную долю от продолжительности времени графика для прокрутки вправо (0.0 to 100.0).

**Пример**

Следующий оператор прокручивает время/дату вправо на 20 процентов от суммарной ширины тенденции PumpPress:

```
HTScrollRight ("PumpPress" , 20 . 0 ) ;
```

Если текущий дисплей начинается при 12:00:00 PM и ширина дисплея составляет 60 секунд, затем начинается новая тенденция при 12:00:12 PM после того, как обрабатывается функция.

## Функция NTZoomIn()

Функция NTZoomIn() вычисляет новую ширину графика и время начала. Если скутеры тенденции находятся в правой и в левой сторонах тенденции, то ширина нового графика равна старой ширине графика, деленной на два. Новое время запуска вычисляется на основе значения аргумента LockString. Если скутеры находятся не правой и левой частях тенденции, функция NTZoomIn() меняет масштаб изображения тенденции в зону, определенную скутерами и игнорирует аргумент LockString.

### Категория

Историческая

### Синтаксис

```
NTZoomIn (Hist_Tag, LockString) ;
```

### Аргументы

*Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*LockString*

Строка, представляющая тип масштаба:

### Замечания

Если положения скутера не находятся в правой и левой сторонах тенденции, новая ширина графика находится между положениями .ScooterPosLeft и .ScooterPosRight. В этом случае значение LockString не используется. Минимальная ширина графика составляет 1 секунду. Положения скутера устанавливаются в .ScooterPosLeft=0.0 и .ScooterPosRight=1.0 после установки масштаба изображения.

### Пример

Следующий оператор меняет масштаб изображения в два раза и сохраняет то же время начала для тенденции Trend1.Trend1.ScooterPosRight равен 1.0 и Trend1.ScooterPosLeft равен to 0.0. Если время начала перед изменением масштаба изображения было равно 1:25:00 PM и ширина графика составляла 30 секунд, новое время начала составляет по-прежнему 1:25:00. Новая ширина графика составляет 15 секунд.

```
NTZoomIn ("Trend1", "StartTime") ;
```

StartTime сохраняет время начала, что и для предыдущего масштаба

Center сохраняет время центра, что и для предыдущего масштаба.

EndTime сохраняет время окончания, что и для предыдущего масштаба.

## Функция NTZoomOut()

Функция NTZoomOut() вычисляет новую ширину графика и время начала. Новая ширина графика равна старой ширине графика, умноженной на два. Новое время начала вычисляется на основе значения аргумента LockString.

### Категория

Историческая

### Синтаксис

**NTZoomOut(Hist\_Tag, LockString) ;**

### Аргументы

*Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*LockString*

Строка, представляющая тип масштаба:

StartTime = сохраняет время начала, что и для предыдущего масштаба

Center = сохраняет время центра, что и для предыдущего масштаба.

EndTime = сохраняет время окончания, что и для предыдущего масштаба.

### Remarks

Текущие положения скутера не влияют на NTZoomOut(). После завершения изменения масштаба изображения новые положения скутера устанавливаются в .ScooterPosLeft=0.0 и .ScooterPosRight=1.0.

### Пример

Следующий оператор уменьшает масштаб времени тенденции в два раза и сохраняет то же время центра для тенденции Volume. Если время начала перед изменением масштаба изображения было равно 2:15:00 PM и ширина графика составляла 30 секунд, время начала после изменения масштаба составляет 2:14:45. Ширина графика составляет 60 секунд и центр тенденции остается на 2:15:15.

**NTZoomOut("Volume", "Center")**

## Распечатка тенденции

Вы можете распечатать историческую тенденцию, которая отображается в настоящее время на экране WindowViewer с использованием функции PrintHT() в скрипте.

### Функция PrintHT()

Функция PrintHT() распечатывает историческую тенденцию, видимую в настоящее время на экране. Как правило, функция PrintHT() связывается с кнопкой экрана, включенной в окно исторической тенденции. Операторы щелкают на кнопке для распечатки видимой исторической тенденции с ее текущими значениями.

#### Категория.

Историческая

#### Синтаксис

```
PrintHT ("Trend_Tag") ;
```

#### Аргументы

*Trend\_Tag*

Признак HistTrend. Реальный признак строки или сообщения.

#### Пример

В этом примере распечатывается историческая тенденция PumpPress, видимая в настоящее время на экране.

```
PrintHT ("PumpPress") ;
```

## Устранение неисправностей в тенденции

Вы можете создать QuickScripts для проверки, были ли успешно восстановлены, данные, которые появляются в исторической тенденции. Используйте функцию HTGetLastError() для устранения неисправностей в тенденции.

### Функция HTGetLastError()

Функция HTGetLastError() может использоваться в скрипте для определения, произошла ли ошибка во время последнего восстановления данных для указанного пера исторической тенденции.

#### Категория

историческая

#### Синтаксис

```
[Result=]HTGetLastError (Hist_Tag, UpdateCount, PenNum) ;
```

#### Аргументы

*Hist\_Tag*

Признак HistTrend, присвоенный имени тенденции.

*UpdateCount*

Целое, представляющее поле точек .UpdateCount тенденции.

*PenNum*

Признак или значение целого, представляющее номер пера (1-8) тенденции.

*Result*

Целое, которое представляет результат последнего обращения к функции скрипта для указанного пера.

0 = нет ошибки

1 = общая ошибка сервера

2 = старый запрос

3 = ошибка файла

4 = сервер не загружается

5 = Тенденция/пера, отправленное на функцию, не существует

6 = признак тенденции не существует в базе данных

7 = Номер пера, отправленный на функцию, не правильный (не в диапазоне 1 до 8).

8 = нет признака или незарегистрированный признак, присвоенный номеру пера.

**Примеры**

Следующий оператор восстанавливает ошибку из последнего восстановления данных для пера 3 тенденции PumpPress и присваивает результат признаку целого ResultCode.

```
[ResultCode=]HTGetLastError ("PumpPress", Trend1 .
UpdateCount, 3) ;
```

При анимации QuickScript Дисплея аналоговой величины следует использовать следующий оператор:

```
HTGetLastError ("Trend1", Trend1 . UpdateCount, 3) ;
```

## Отображение значений в реальном времени в тенденции

Вы можете создать тенденции в реальном времени двумя способами. Объект тенденции в реальном времени дает стандартный набор органов управления для выбора данных, установки диапазона времени и указывает физическое проявление графика. InTouch также включает мастер тенденции на 16-перьев, который представляет собой дополнительное управление для создания тенденций исторической и в реальном времени. Дополнительную информацию о создании тенденций в реальном времени с помощью мастера на 16 перьев, смотрите Главу 4, Использование мастера тенденции на 16 перьев, в Руководстве по дополнительным Компонентам InTouch® HMI.

## Использование объектов тенденции в реальном времени

Вы можете создавать тенденцию в реальном времени для отображения текущих величин в вашем приложении.

### Создание тенденции в реальном времени

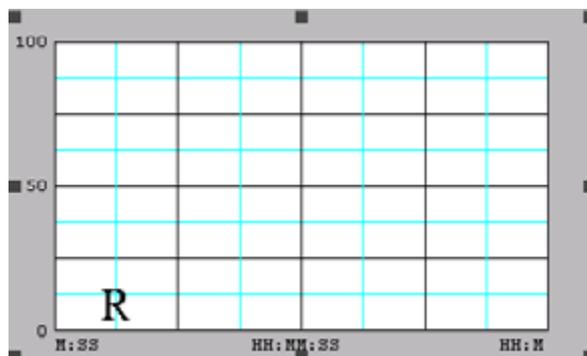
Вы можете использовать средство тенденции в реальном времени Real-time Trend для создания объектов тенденции в окне. Когда вы в первый раз вставляете объект тенденции в реальном времени, WindowMaker использует установки по умолчанию. После конфигурирования тенденции в реальном времени, WindowMaker использует последние значения конфигурации в качестве исходных установок для любого нового объекта тенденции в реальном времени.

Вы можете нарисовать график тенденции любого размера в пределах границ окна.



#### Для создания тенденции в реальном времени

- 1 выберите средство Real-time Trend из Drawing Toolbar (линейка инструментов рисования) .
- 2 Переместите мышь по области окна, где вы хотите разместить тенденцию в реальном времени. Проведите мышью по диагонали для создания прямоугольника нужного размера для тенденции. В окне появляется объект тенденции в реальном времени.



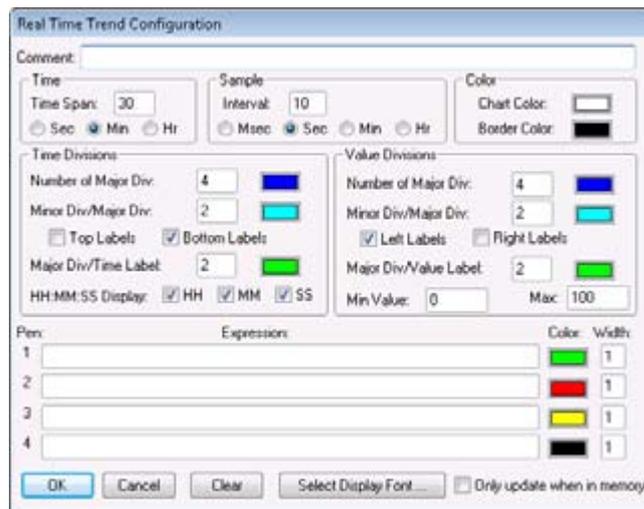
- 3 При необходимости, настройте ширину и высоту тенденции с помощью программы обработки объекта.

## Конфигурирование того, какие признаки отображать на дисплее в тенденции реального времени

Перо тенденции в реальном времени создает графическое представление текущих данных из любого локального признака или выражения, которое содержит один или более локальных признаков. Вы можете конфигурировать перья, которые отображают данные признака в тенденции реального времени.

**Для конфигурирования признаков тенденции в реальном времени**

- 1 Дважды щелкните на объекте тенденции в окне. Появляется диалоговый блок **Real Time Trend Configuration**.



- 2 В области **Expression** введите имя локального признака или выражения, которое содержит один или более локальных признаков. Если вы дважды щелкните на блоке **Pen**, появляется диалоговый блок **Select Tag**, показывающий перечень признаков, определенные в приложении. Вы можете присвоить признак для пера путем выбора его из диалогового блока **Select Tag**.
- 3 Дважды щелкните на цветном блоке, следующим за каждым пером, присвоенным признаку для отображения цветовой палитры.
- 4 Щелкните на цвете, который вы хотите присвоить перу.
- 5 В блоке **Width** введите ширину линии в пикселях для каждого пера, показанного в тенденции. Выбор ширины линии, больше, чем 1, увеличивает время, требуемое для обновления или печати тенденции.
- 6 Выберите блок проверки **Only update when in memory**, если вы хотите, чтобы тенденция обновлялась только тогда, когда отображалась в активном окне. Если вы не выбираете эту опцию, тенденция обновляется непрерывно, даже если окно закрыто. Непрерывное обновление тенденции замедляет работу системы.

- 7 Сохраняйте диалоговый блок открытым **Real Time Trend Configuration** и перейдите к новой процедуре, описанной в разделе **Configuring the Real-Time Trend Time Span and Update Rate** (конфигурирование скорости обновления и временной интервал тенденции в реальном времени).

## Конфигурирование скорости обновления и временной интервал тенденции в реальном времени

Вы можете конфигурировать промежуток времени и скорость обновления для тенденции в реальном времени.

### Для установки промежутка времени и скорости обновления для тенденции в реальном времени

- 1 При необходимости дважды щелкните на объекте тенденции. Появляется диалоговый блок **Real Time Trend Configuration**.
- 2 В области **Time** введите продолжительность времени в блок **Time Span**, который вы хотите, чтобы появился на горизонтальной оси X тенденции.
- 3 Выберите единицу измерения для времени тенденции.
  - Секунды (Sec)
  - Минуты (Min)
  - Часы (Hr)

Например, если вы введете 30 в блок **Time Span** и затем выберите **Min**, промежуток времени, который отображается в графике, составит 30 минут.

- 4 В области **Sample** введите число в блок **Interval**, которое рассчитывает выражение для тенденции и в соответствии с которым обновляется график.
- 5 Вы берите единицу измерения интервала.
  - Миллисекунды (Msec)
  - Секунды (Sec)
  - Минуты (Min)
  - Часы (Hr)

Например, если вы водите 10 в **Interval** и затем выбираете **Sec**, тенденция в реальном времени обновляется каждые 10 секунд.

- 6 Сохраняйте диалоговый блок **Real Time Trend Configuration** открытым и перейдите к следующей процедуре, описанной в **Configuring Real-time Trend Display Options** (конфигурирование опций индикации тенденции в реальном времени)

## Конфигурирование опций индикации тенденции в реальном времени

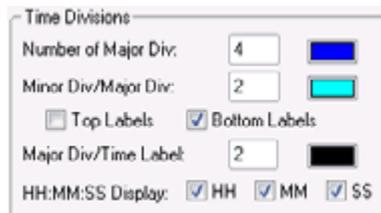
Вы можете сконфигурировать визуальное представление тенденции в реальном времени.

**Для конфигурирования опций отображения тенденции в реальном времени**

- 1 Дважды щелкните на объекте тенденции. Появляется диалоговый блок **Real Time Trend Configuration**.
- 2 В области **Color** сконфигурируйте цвет. Выполните любое из следующего:
  - Дважды щелкните на блоке **Chart Color** для открытия палитры цветов. Выберите фоновый цвет для тенденции. По умолчанию белый цвет фона. Любой другой цвет значительно увеличивает время, необходимое для печати тенденции.
  - Дважды щелкните на блоке **Border Color** для открытия палитры цветов. Выберите цвет границы тенденции.
- 3 В области **Time Divisions** сконфигурируйте временные деления.

Выполните следующее:

- В блоке **Number of Major Div** введите число главных временных делений тенденции. Главные временные деления появляются на горизонтальной оси времени тенденции. Число главных временных делений должно быть кратным величине **Major Div/Time Label**. Например, число делений 20 кратно для величины **Major Div/Time Label** в 4.



- Выберите цвет для линий главных делений.
- В блоке **Minor Div/Major Div** введите число главных временных делений, видимых в пределах каждого главного деления времени. Число малых временных делений должно быть кратным периоду главных делений. Например, если период главного деления установлен в 60 секунд, ввод значения 2 в **Minor Div/Major Div** Создает два периода малого временного деления в 30 секунд.
- Выберите цвет для линий малых делений.
- Выберите блок проверки либо **Top Labels** либо **Bottom Labels** для указания положения временных меток на тенденции. Вы можете выбрать обе опции для размещения временных меток вверху или внизу тенденции. Оставление обеих опций пустыми удаляет временные метки с горизонтальной оси тенденции.

- Если вы используете временные метки, введите число линий главных временных делений на одну временную метку в **Major Div/Time Label**. Число главных делений должно быть кратным величине **Major Div/Time Label**.

Выберите цвет меток временных делений.

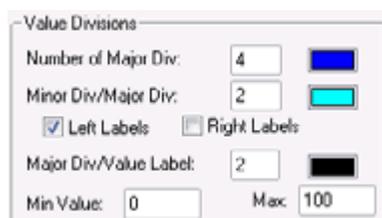
- Выберите единицы времени, показанные, как часть метки главного временного деления.

Часы (HH)

Минуты (MM)

Секунды (SS)

- 4 В области **Value Divisions** сконфигурируйте вид вертикальной оси в тенденции.



Опции **Value Divisions** конфигурируются аналогично опциям **Time Divisions**. Главные и малые деления по оси у показывают величину значения данных, а не времени. Вертикальная ось указывает диапазон значений данных, которые появляются в тенденции на основе инженерных блоков для всех признаков. Для указания десятичных точек для малых и главных делений значения введите действительные числа для опций **Min Value** и **Max**. Например, от 0.00 до 100.00.

- 5 Щелкните на **Select Display Font**. Появляется диалоговый блок **Font** с опциями для установки шрифта, стиля и размера текста, которое появляется в тенденции.
- 6 Щелкните **OK**.

## Печать тенденции во время выполнения

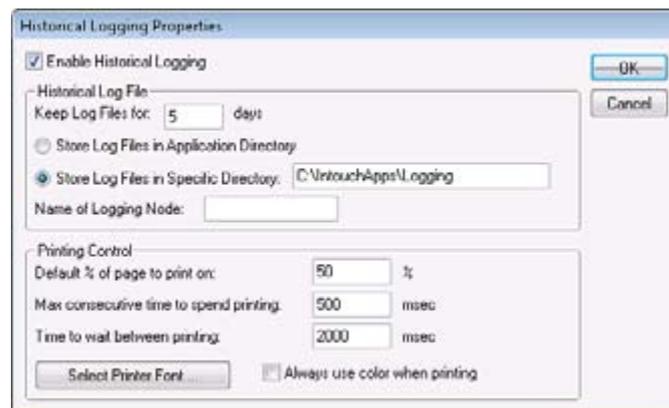
Ряд факторов определяют то, как быстро можно распечатать тенденцию. Первичным фактором является размер тенденции на странице печати. Режим отображения тенденции также влияет на работу при печати. Тенденции Min/Max или Average/Scatter могут печататься более быстро, чем тенденции Average/Bar Chart. Также, чем длиннее или чем толще линии на тенденции, тем больше времени на печать. Другой фактор, который влияет на работу при печати – это цвет фона тенденции. В большинстве случаев печать в белом фоне более быстра, чем цветного фона.

## Конфигурирование опций печати тенденций

Вы можете конфигурировать опции, которые определяют то, как тенденция распечатывается.

Для конфигурирования печати исторической тенденции

- 1 В меню **Special** щелкните на **Configure** и затем на **Historical Logging**. Появляется диалоговый блок **Historical Logging Properties**.



- 2 В области **Printing Control** укажите процентную долю страницы для печати тенденции в блоке **Default % of page to print on**. Если вы введете 50, тенденция печатается на половине странице по вертикали и горизонтали. Тенденция, напечатанная при 50 процентах, занимает значительно меньше времени, чем при печати полной страницы. В качестве альтернативы при печати вы можете использовать функцию `PrintWindow() QuickScript`.
- 3 В блоке **Max consecutive time to spend printing** введите часть времени процесса в миллисекундах. Часть времени представляет период, локализованный в процессор компьютера для выполнения процесса модуля печати на переднем плане и печати тенденции. Более продолжительная часть времени позволяет напечатать тенденцию более быстро ценой других процессов, выполняемых в компьютере.

- 4 В блоке **Time to wait between printing** введите время в миллисекундах, которое модуль печати ожидает между секционированиями времени процессора. Более короткий период времени между секционированием времени процессора позволяет печатать тенденцию более быстро.
- 5 Щелкните на **Select Printer Font**. Появляется диалоговый блок **Font**. Выберите характеристики текста, который появляется в тенденции.
- 6 Щелкните на **OK** для сохранения вашей конфигурации печати и закройте диалоговый блок **Historical Logging Properties**.

## Отображение значений исторической тенденции из других узлов InTouch или сервера IndustrialSQL

Если вы хотите использовать данные, сохраненные дистанционно для создания исторических тенденций, дистанционный провайдер должен зарегистрироваться в перечне истории провайдера InTouch. Этот перечень указывает положение имени и сети каждого провайдера истории. К этим именам обращаются всегда, когда перья исторической тенденции указывают на признаки в дистанционном провайдере истории. Вы можете определить дистанционного исторического провайдера и присвоить перья исторической тенденции данным признака, сохраненным в дистанционном положении. Вы можете:

- Сконфигурируйте дистанционных провайдеров истории.
- Сконфигурируйте перья для отображения данных из дистанционного провайдера истории.
- Присвойте перья данным признака, сохраненным в дистанционном провайдере истории используя **Tag Browser**.
- Присвойте перо дистанционному признаку используя **QuickScript**.

Дополнительную информацию касательно использования данных из дистанционного провайдера истории смотрите Главу 2, Распределяющие приложения в Руководстве Управления и расширения приложения InTouch® HMI

## Конфигурирование перьев для отображения дистанционных данных тенденции

Исторические тенденции могут отображать данные признака как из локального, так и из дистанционного провайдеров истории. Вы можете присваивать перья тенденции для отображения данных из дистанционного провайдера истории.

**Для отображения признака из дистанционного провайдера истории**

- 1 Дважды щелкните на исторической тенденции для появления диалогового блока **Historical Trend Configuration**.
- 2 В каждом блоке пера Tagname введите ссылку на дистанционный провайдер истории. Формат ссылки на дистанционный провайдер истории следующий:

*history\_provider\_name.tag\_name*

Пример:

**TankFarm1 . Pump1RPM**

Каждое перо исторической тенденции может отсылать к иному дистанционному провайдеру истории.

- 3 Щелкните на ОК для сохранения изменений конфигурации.

---

**Примечание** Поле точек .TagID не может использоваться при ссылках признака на дистанционный провайдер истории.

---

## Использования программы просмотра признака для присваивания перьев для дистанционных провайдеров истории.

В следующей процедуре поясняется, как использовать Программу просмотра признака (Tag Browser) для присваивания пера тенденции данным признака из дистанционного провайдера истории. Использование программы просмотра признака для выбора признаков исключает потребность в ручном вводе имени каждого признака и снижает вероятность ошибок. Имя дистанционного узла, которое вы указываете в Access Name, не должно быть фактическим именем узла, где сохраняется признак. Но вы должны создавать Access Name (имя доступа) для определения дистанционного провайдера истории, как источника признака. Дополнительную информацию о создании Access Name смотрите в Setting Up Access Names (настройка имен доступа) на странице 70.

**Для определения дистанционного провайдера истории в качестве источника признака**

- 1 Создайте Access Name, которое указывает имя узла, в котором размещается провайдер истории.
- 2 Дважды щелкните на исторической тенденции для открытия диалогового блока **Historical Trend Configuration**.

- 3 Дважды щелкните на входном блоке пера **Tagname** для отображения диалогового блока **Select Tag**.
- 4 Щелкните на **Define Tag Sources** для определения дистанционного провайдера истории в качестве источника признака.
- 5 Щелкните на стрелке **Tag Source** и выберите новый источник признака в дистанционном провайдере истории в перечне или щелкните на кнопке **Tree View** и выберите источник признака на панораме вида дерева. Обновляется диалоговый блок **Select Tag** и отображает признаки из выбранного дистанционного провайдера истории.
- 6 Выберите признак, который вы хотите присвоить историческому перу и щелкните на **ОК**. Повторно появляется диалоговый блок **Historical Trend Configuration** с выбранным признаком, приведенным в блоке **Tagname** перьев, как: `AccessName:Item`.
- 7 Замените `AccessName:` часть с именем провайдера истории, которое вы определили в **Distributed Name Manager**. Например, `HistPrv1.tag_name`

Этот процесс может показаться громоздким, но после того, как вы определили провайдера истории, как источник признака в программе просмотра признака (Tag Browser), каждый раз, как вы щелкаете дважды на другом входном блоке признака, вы просто дважды щелкаете на имени признака в Tag Browser, и затем заменяете `AccessName:` часть с именем провайдера истории.

---

**Примечание** В WindowViewer, если допускаются изменения во время исполнения для исторической тенденции, когда пользователь щелкает на кнопке пера для изменения имени признака tagname, появляется Tag Browser (просмотр признака), но доступными являются только локальные признаки приложения.

---

## Использование QuickScript для присваивания пера дистанционному провайдеру истории

Когда выполняется приложение InTouch, вы можете конфигурировать перо тенденции для отображения данных признака от дистанционного провайдера истории. Создайте QuickScript, который указывает ссылку признака дистанционного провайдера истории в функции `HTSetPenName()`. Например:

```
HTSetPenName ("HistTrendTag", 1, "HistPrv1.Boiler1");
```

В этом примере номер 1 указывает номер пера в исторической тенденции, которая строит значения дистанционного признака Boiler1 из дистанционного провайдера истории HistPrv1.

---

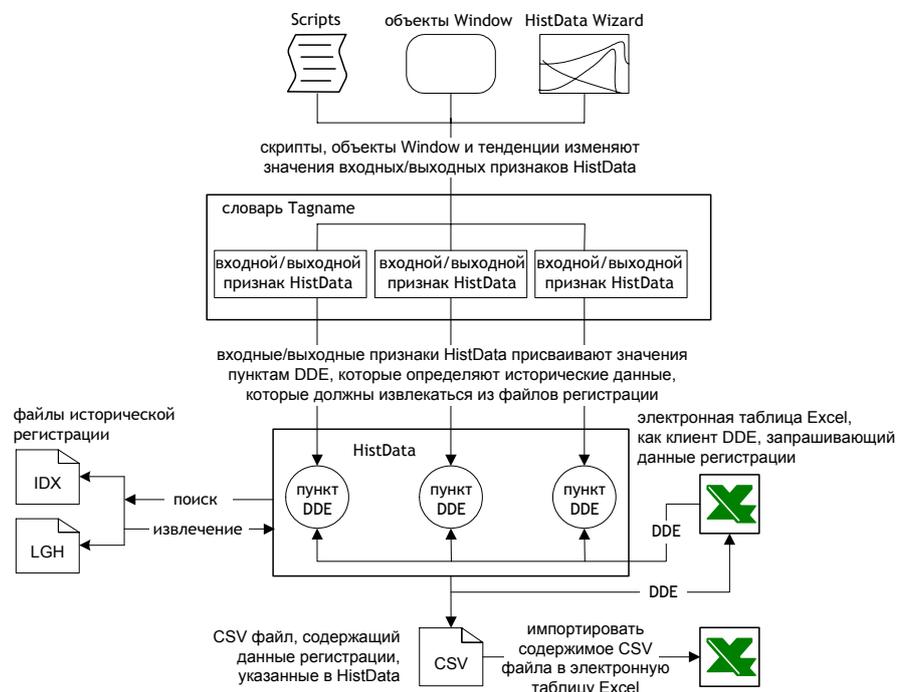
**Примечание** Диалоговый блок при выполнении Historical Trend Setup и поля точек Pen не поддерживаются для дистанционных провайдеров истории.

---

# Глава 11

## Доступ к значениям исторического признака из других приложений

Вы можете использовать утилиту InTouch HistData для извлечения данных из файлов исторической регистрации в файлы значений с разделением запятой (.csv). Приложения типа Excel могут извлекать непосредственно данные регистрации InTouch из HistData, как данные клиента DDE, или регистрации импорта из выходного файла, созданного утилитой HistData. На рисунке ниже показан процесс сохранения запомненных данных исторической регистрации в файл или приложение клиента DDE.



## Использование пунктов DDE для отображения исторических данных

Программа HistData включает набор пунктов DDE, которые указывают то, как исторические данные извлекаются из исторических файлов. Эти пункты являются частью внутренней базы данных HistData. Вы присваиваете значение каждому пункту. В следующей таблице сведены пункты HistData, определенные в программе HistData.

Пункт	Тип данных	Описание
DATADIR	Сообщение	Путь к папке, содержащей файлы исторической регистрации
DBDIR	Сообщение	Путь к папке, содержащей содержимое словаря InTouch Tagname
STARTDATE	Сообщение	Дата запуска для извлечения данных из файла регистрации. Формат даты запуска - ММ/ДД/YY
STARTTIME	Сообщение	Время запуска для извлечения данных из файла регистрации. Формат времени запуска - НН:ММ:СС с использованием 24-х часовой шкалы.
DURATION	Сообщение	<p>Длительность интервала сбора данных из файлов регистрации. Продолжительность может выражаться следующим образом:</p> <ul style="list-style-type: none"> <li>• недели (w)</li> <li>• дни (d)</li> <li>• часы (h)</li> <li>• минуты (m)</li> <li>• секунды (s)</li> </ul> <p>Могут указываться частичные периоды DURATION. Например, DURATION=0,5 m равно 30 секундам. Для запроса простого примера установите DURATION в 0.</p>
INTERVAL	Сообщение	<p>Длительность времени между интервалами сбора данных. INTERVAL может выражаться в неделях, днях, часах, минутах и секундах. Эти единицы времени периода INTERVAL те же, что и для периода DURATION</p> <p>Могут указываться частичные интервалы. Например, INTERVAL=0.25d представляет 6 часов.</p> <p>Максимальный период для DURATION или INTERVAL составляет шесть недель. Максимальный период в шесть недель применяется для любых значений времени, присвоенных для DURATION или INTERVAL. Например, 42 – это максимальное число дней в периоде DURATION или INTERVAL.</p>
FILENAME	Сообщение	Имя и положение папки файла, содержащего данные, извлеченные из файла исторической регистрации.

Пункт	Тип данных	Описание
WRITEFILE	Целое	Флаг, который показывает состояние операции записи HistData в выходной файл. Когда устанавливается в 1, HistData записывает требуемые данные в файл, указанный с помощью имени пункта FILENAME. Когда обновление файла завершается, WRITEFILE автоматически сбрасывается в 0.
Пункт	Тип данных	Описание
ERROR	Сообщение	Строка, содержащая описание последней ошибки, произошедшей при извлечении данных из файлов регистрации. Когда STATUS устанавливается в 1, строка ERROR устанавливается в None. Когда STATUS устанавливается в 0, строка ERROR содержит сообщение об ошибке.
TAGS1, TAGS2	Сообщение	Строка TAGS может содержать 131 символ в WindowViewer и 255 символов в Excel. Строка может присоединяться для более длинных запросов путем добавления пунктов признаков, именуемых Tagsn, где n представляет инкрементирующее целое. Если признак требует дополнительный текст признака, введите + в конце строки. Например: TAGS="\$Date,ProdLevel,ProdTemp,+" TAGS1="ReactLevel,Temp,GasLevel,+" TAGS2="MotorStatus" Двойные признаки не разрешаются и максимальная длина каждой строки признака равна 512 байтов.
PRINTTAGNAMES	Дискретный	Флаг, который показывает, помещаются ли имена признаков выше связанных столбцов значений. Когда устанавливается в 1, печатаются имена признаков. Когда устанавливаются в 0, имена не печатаются.
DATA	Сообщение	Этот пункт содержит требуемые данные в программе HistData в формате значений, отделенном запятой. Он используется другими приложениями для данных ADVISE или REQUEST в DDE.
STATUS	Дискретный	Состояние самой последней операции HistData. Значение 1 означает, что HistData успешно извлекла исторические данные из файла регистрации. Значение 0 показывает, что имеет место ошибка.
SENDDATA	Целое	Флаг, который показывает состояние операции обновления HistData. При установке в 1 HistData обновляет пункт DATA с требуемыми данными. После завершения обновления, автоматически устанавливается в 0. Если вы принимаете сообщение об ошибке, показывающее что слишком много данных было запрошено с использованием SENDDATA, укоротите период DURATION или уменьшите число запрашиваемых признаков. Двойные признаки не разрешаются и максимальная длина каждой строки признака равна 512 байтов.

## Доступ к данным регистрации с помощью DDE

Вы можете использовать два метода для извлечения данных регистрации в выходной файл.

- используйте ручной метод, если вы хотите сохранить данные исторической регистрации от восьми и более признаков, в выходной файл.
- используйте вместо этого мастер HistData, если вы только хотите сохранить данные по регистрации, преобразованных для перьев, присвоенных в настоящее время исторической тенденции.

## Ручное извлечение данных регистрации с помощью HistData

Вы можете вручную извлекать данные регистрации в выходной файл.

Выполните этапы в следующем порядке.

- Создайте имя доступа HistData Access Name
- Создайте входные/выходные признаки для HistData
- Создайте окно HistData
- Запустите HistData

### Создайте имя доступа HistData Access Name

Для InTouch для запроса данных из программы HistData, вы должны определить имя доступа Access Name:

- 1 В блоке **Access Name** введите имя длиной до 32 цифробуквенных символов. Значения, присваиваемые для **Access Name** и **Topic Name** должны быть одинаковыми.
- 2 В блоке **Node Name** введите имя узла, в котором файлы регистрации в настоящее время размещаются.
- 3 В блоке **Application Name** введите **HistData** без расширения файла .exe.

- 4 В блоке **Topic Name** введите имя, которое вы указали из блока **Access Name**. **Access Name** и **Topic Name** должны быть одинаковыми.
- 5 выберите DDE в качестве протокола связи.
- 6 В области **When to advise server** выберите **Advise all items**, как только используется HistData.
- 7 щелкните на **OK**.

### Создайте признаки HistData

После определения Access Name, создайте следующие признаки типа ввода/вывода для формирования одного выходного файла, который содержит данные из файлов регистрации. Присвойте Access Name, созданные на предыдущем этапе, признакам.

Признак	Тип признака ввода/ вывода	Пункт
HDWDATADIR	Сообщение	DataDir
HDWDBDIR	Сообщение	DbDir
HDWDURATION	Сообщение	Duration
HDWERROR	Сообщение	Error
HDWFILENAME	Сообщение	FileName
HDWINTERVAL	Сообщение	Interval
HDWSTARTDATE	Сообщение	StartDate
HDWSTARTTIME	Сообщение	StartTime
HDWSTATUS	Сообщение	Status
HDWTAGS HDWTAGS1 HDWTAGS2	Сообщение	Tags
PRINTTAGNAMES	Дискретный	PrintTagNames
HDWWRITEFILE	Целый	WriteFile

**Примечание** Мастер HistData Wizard создает эти признаки автоматически, за исключением признака PRINTTAGNAMES. Создайте два дополнительных признака, если вы хотите отправить данные регистрации на пункт Data, так, чтобы он был доступен с других приложений. Также мастер HistData Wizard автоматически не создает признаки HDWSendDate и HDWData.

Признак	Тип признака ввода/ вывода	Пункт
HDWSendDate	Дискретный	SendData
HDWData	Сообщение	Data

## Создайте окно HistData

После того, как вы создали тип признаков ввода/вывода, создайте новое окно, называемое HistData, аналогично следующему примеру:

The screenshot shows a configuration window titled "HistData". It has two tabs: "Status" and "Write File". There are two buttons: "Initialize Data" and "Write File". The fields include:

- Path to the historical data file: # (Drive:\Path)
- Path to the InTouch Database: # (Drive:\Path)
- Start date for requested historical data: # (MM/DD/YY)
- Start time for requested historical data using the 24 hour clock: # (HH:MM:SS)
- Duration for the requested data to be returned: # (m=week, d=day, h=hr, m=min, s=sec, 0 = one sample)
- Length of time between samples: # (1d = 1 day, 1w = 1 week, 25d = 6 hour)
- Complete Pathname of the file to write data to: # (Drive:\Path)
- List of tagnames to return historical data for: #

There are also notes: "(Note: maximum length of time allowed for Duration & Interval is 6 weeks)" and "(Note: Date & Time for a sample can be requested via the System tags, \$Date and \$Time)". An "Error Message" field is at the bottom.

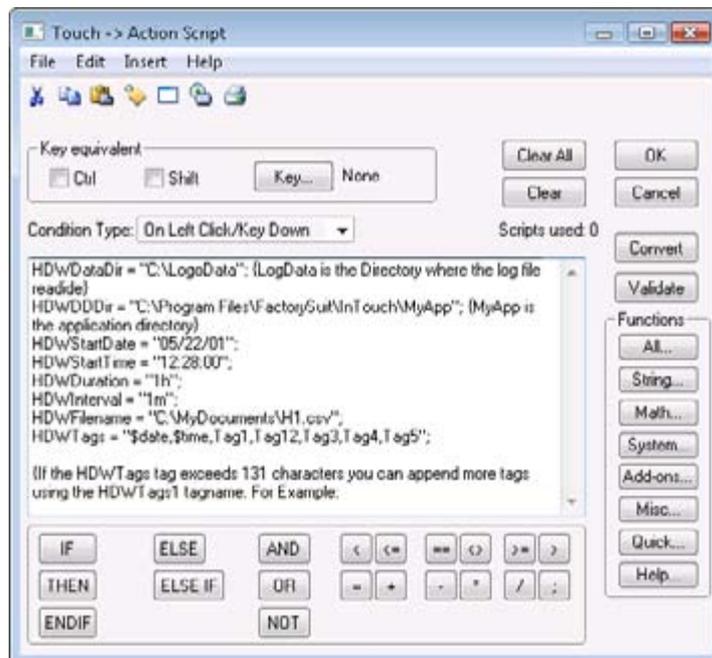
Символы # связываются с входной связью пользователя. Например, символ # связь User Inputs/String на признак HDWDataDir. Входная связь пользователя позволяет вам менять значение признака во время выполнения.

Кнопка **Status** связывается с цветом заполнения – дискретным выражением, основанным на признаке HDWStatus.

The dialog box shows the configuration for a fill color based on a discrete expression. The expression is "HDWStatus==1". The colors are defined as:

- 1,TRUE,On: Green
- 0,FALSE,Off: Black

Кнопка **Initialize Data** связывается с Touch Pushbutton (нажимная кнопка)—скрипт Action.



Когда щелкается на кнопке **Initialize Data**, пункты HistData инициализируются дискретными значениями. При необходимости, эти значения могут также меняться во время выполнения путем использования **User Inputs Links**.

Кнопка **Write File** связывается с Touch Pushbutton (нажимная кнопка)—скрипт Action.



При щелчке кнопка WriteFile выработивает выходной файл.

## Выполните HistData

После создания окна HistData, выполните следующие этапы по управлением WindowViewer:

- 1 запустите **HistData** и минимизируйте его.
- 2 Запустите WindowViewer и откройте окно **HistData**.
- 3 Щелкните кнопку **Initialize** и проведите изменения в пунктах **HistData**, при необходимости.
- 4 Щелкните кнопку **WriteFile**.

Если операция успешна, значение **Status - ON** и появляется цвет, связываемый с состоянием **ON**. Если операция не успешна, состояние **Status - OFF** и сообщение **Error Message** показывает причину сбоя.

## Использование мастера HistData Wizard для извлечения данных регистрации

Вы можете создавать выходной файл, содержащий данные регистрации, которые появляются в исторической тенденции. InTouch включает мастер HistData Wizard для автоматизации этапов по извлечению данных из файла регистрации. Так как выходной файл HistData записывает данные регистрации, показанные в исторической тенденции, файл может содержать только данные из признаков, в настоящее время присвоенные перьям исторической тенденции.

**Для использования мастера HistData Wizard для извлечения данных регистрации**

- 1 запустите WindowMaker.
- 2 Откройте окно, которое включает историческую тенденцию.
- 3 Щелкните на средстве **Wizards** в линейке меню. Появляется диалоговый блок **Wizard Selection**.
- 4 Выберите группу **Trends** из левой панорамы.
- 5 Выберите иконку **HistData Wizard** из правой панорамы и щелкните **OK**.
- 6 Переместите указатель мыши через область окна тенденции там, где вы хотите разместить объект **HistData**.
- 7 Щелкните для установки объекта **HistData** в окне тенденции.



Мастер HistData Wizard создает объект окна, содержащий кнопку и блок **Filename**, который показывает путь, по которому будет создаваться выходной файл.



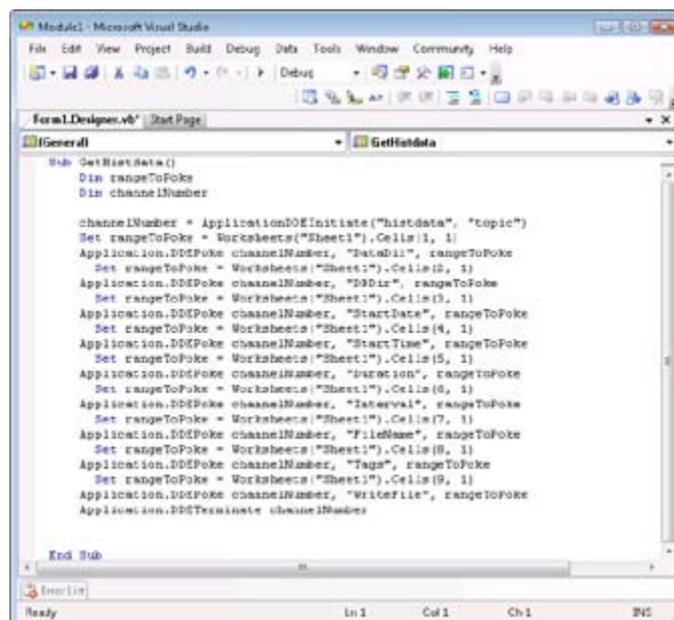
- 8 Дважды щелкните на объекте **HistData Wizard**, размещенном в окне тенденции. Появляется диалоговый блок **HistData Panel Wizard**.



- 9 В блоке **Hist Trend** введите имя для признака **HistTrend**.
- 10 В блоке **Number of Records to Write per CSV File** введите число записей в для выполнения записи в выходной файл.
- 11 Щелкните на **OK**. Мастер HistData Wizard создает набор признаков, которые идентифицируются с помощью HDW префикса T. Мастер HistData Wizard создает признаки, перечисленные в Create HistData Tags (создать признаки HistData) на странице 248. Мастер HistData Wizard присваивает признаки имени доступа HistDataViewSt Access Name.
- 12 Запустите окно исторической тенденции с помощью WindowViewer.
- 13 Дайте приложению поработать в течение времени, пока вы хотите собрать исторические данные из файла регистрации.
- 14 Щелкните на **Save to File**, которая является частью объекта окна HistData. HistData создает выходной файл в положении папки, показанном в объекте окна.

## Заимствование данных регистрации в файл CSV

Вы можете написать макрос в Excel для интерфейса в HistData. Программа HistData отвечает на функции INITIATE, ROKE и TERMINATE в пределах макроса. Функция ROKE с зарезервированным словом (пункт внутренней базы данных) устанавливает параметры, которые определяют запрос. После того, как запрос правильно указан, запустите макрос для запроса выбранных исторических данных из файла HistData. Вы можете записать макрос на Excel для сопряжения с HistData. В следующих примерах показан макрос, написанный с помощью VBA:



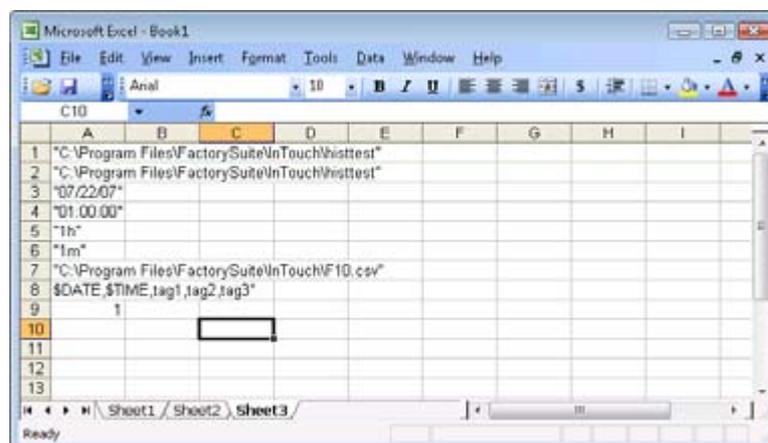
```

Sub GetHistData()
    Dim rangeToPoke
    Dim channelNumber

    channelNumber = Application.DDEInitiate("histdata", "topic")
    Set rangeToPoke = Worksheets("Sheet1").Cells(1, 1)
    Application.DDEPoke channelNumber, "DataDir", rangeToPoke
    Set rangeToPoke = Worksheets("Sheet1").Cells(2, 1)
    Application.DDEPoke channelNumber, "DDEDir", rangeToPoke
    Set rangeToPoke = Worksheets("Sheet1").Cells(3, 1)
    Application.DDEPoke channelNumber, "StartDate", rangeToPoke
    Set rangeToPoke = Worksheets("Sheet1").Cells(4, 1)
    Application.DDEPoke channelNumber, "StartTime", rangeToPoke
    Set rangeToPoke = Worksheets("Sheet1").Cells(5, 1)
    Application.DDEPoke channelNumber, "Duration", rangeToPoke
    Set rangeToPoke = Worksheets("Sheet1").Cells(6, 1)
    Application.DDEPoke channelNumber, "Interval", rangeToPoke
    Set rangeToPoke = Worksheets("Sheet1").Cells(7, 1)
    Application.DDEPoke channelNumber, "FindName", rangeToPoke
    Set rangeToPoke = Worksheets("Sheet1").Cells(8, 1)
    Application.DDEPoke channelNumber, "Tags", rangeToPoke
    Set rangeToPoke = Worksheets("Sheet1").Cells(9, 1)
    Application.DDEPoke channelNumber, "WriteFile", rangeToPoke
    Application.DDETerminate channelNumber
End Sub

```

В вышеприведенном примере данные, которые должны записываться в запоминающее устройство, находятся в Sheet1. В следующем примере показаны данные, которые должны записываться в запоминающее устройство:



	A	B	C	D	E	F	G	H	I
1	"C:\Program Files\FacilitySuite\InTouch\histtest"								
2	"C:\Program Files\FacilitySuite\InTouch\histtest"								
3	"07/22/07"								
4	"01:00:00"								
5	"1h"								
6	"1m"								
7	"C:\Program Files\FacilitySuite\InTouch\F10.csv"								
8	\$DATE,\$TIME,tag1,tag2,tag3"								
9	1								
10									
11									
12									
13									

## Функция WWPoke()

Функция WWPoke() записывает в запоминающее устройство значение, указанное как Application, Topic и Item с использованием протокола DDE.

### Катгория

WWDDE

### Синтаксис

```
[Status=]WWPoke ("Application", "Topic", "Item",
  "TextValue");
```

### Аргументы

#### *Application*

Приложению послать команду Poke. Фактическая строка или признак сообщения.

#### *Topic*

Теме в пределах приложения послать команду Poke. Фактическая строка или признак сообщения.

#### *Item*

Пункту в пределах темы на Poke. Фактическая строка или признак сообщения.

#### *TextValue*

Переменная сообщения или строка. Если значение, которое вы хотите послать, является числом, вы можете преобразовать его используя функцию Text(), StringFromIntg() или StringFromReal(). Фактическая строка или признак сообщения.

### Замечание

The value assigned to the TextValue argument is sent to the specified Application, Topic, and Item.

### Examples

Следующий оператор преобразует значение в текст и заносит его в запоминающее устройство в ячейку цифровой таблицы Excel:

```
String=Text(Value,"0"); WWPoke("excel", "[Book1.
  xls]sheet1", "r1c1", String);
```

Работа WWPoke() из приложения "View" to "View" является неопределенной и не поддерживается. Не гарантируется, что функция WWPoke() достигнет цели в этом случае и вероятно произойдет перерыв без возврата необходимых данных. Функция WWPoke() возвращает 1, если приложение выполняется, тема и пункт существуют и значение успешно пересылается. WWPoke() возвращает 0, когда приложение занято и -1, когда происходит ошибка. Поэтому результат обращения к функции WWPoke() может отслеживаться.

```
Status=WWPoke ("excel", "[Book1.xls]sheet1", "r1c1",
  String);
```

В этом примере, `Status` – это признак целое, присвоенный возвращенному значению 1, -1 или 0 из функции `WWPoke()`.

#### Смотрите также

`Text()`, `StringFromIntg()`, `StringFromReal()`

### Функция `WWRequest()`

Функция `WWRequest()` выполняет одnorазовый запрос значения из конкретного Приложения, Темы и Пункта с использованием протокола DDE.

#### Категория

WWDDE

#### Синтаксис.

```
WWRequest (Application , Topic , Item , ValueMsg_Tag) ;
```

#### Arguments

##### *Application*

Запросить из приложения данные. Фактическая строка или признак сообщения.

##### *Topic*

Тема в рамках приложения, из которой запрашиваются данные. Фактическая строка или признак сообщения.

##### *Item*

Пункт в теме, из которого запрашиваются данные. Фактическая строка или признак сообщения.

##### *ValueMsg\_Tag*

Признак сообщения, заключенный в кавычки, который содержит запрашиваемые данные из приложения, темы и пункта.

Фактическая строка или признак сообщения.

#### Замечания

В этом операторе значение DDE в конкретном приложении, теме или пункте присваивается в качестве значения для `ValueMsg_Tag`.

Значение возвращается в качестве строки, присвоенной признаку сообщения. Если значение – это число, можете преобразовать его, используя функции `StringToIntg()` или `StringToReal()`.

#### Примеры

Следующий оператор запрашивает значение из цифровой таблицы Excel и преобразует результирующую строку в действительное число:

```
WWRequest ("excel" , " [Book1.xls] sheet1" , "r1c1" , Result) ;  
Value=StringToReal (Result) ;
```

Функция `WWRequest()` возвращает 1, если приложение выполняется, тема и пункт существуют и значение успешно возвращается. возвращает 0, когда приложение занято и -1, когда

происходит ошибка. Поэтому результат обращения к функции `WWReques ()` может отслеживаться. В этом примере `Status` – это целое признака, присвоенного возвращенному значению 1, -1, или 0 из функции `WWRequest()`.

```
Status=WWRequest("excel", "[Book1.xls]sheet1",
    "r1c1", Result);
```

Смотрите также

`StringToIntg()`, `StringToReal()`

## Устранение ошибок HistData

Вы можете видеть ошибки, которые могут происходить при извлечении данных регистрации с помощью `HistData`. В следующей таблице в левом столбце перечисляются типовые сообщения при проблемах или ошибках `HistData`. В правом столбце описываются возможные причины и решения проблем.

Сообщения об ошибке или состоянии	Причина и/или решение
Сообщение об ошибке: <i>Too much data requested – shorten the duration or reduce the number of tagnames.</i> (запрошено слишком много данных- уменьшите продолжительность или уменьшите число имен файлов)	Эта ошибка происходит, когда запрашивается слишком много данных с помощью пункта <code>SendData</code> . Если единственной целью является создание выходного файла, содержащего данные из файлов регистрации, не используйте пункт <code>SendData</code> .
Сообщение об ошибке: <i>Could not open file C:\FILES1\HISTDATA.CSV</i> (нельзя открыть файл C:\FILES1\HISTDATA.CSV)	Этот путь к папке не существует или написание пути к папке не правильное
Сообщение об ошибке: <i>Could not open file C:\FILES\</i> (нельзя открыть файл C:\FILES\)	Не определено выходного файла
Сообщение об ошибке: <i>DATADIR item invalid</i> (неправильный пункт DATADIR)	Не существует заданного с помощью пункта <code>DataDir</code> пути папки назначения. Проверьте написание пути папки.
Сообщение об ошибке: <i>STARTDATE item invalid</i> (неправильный пункт STARTDATE)	Пункт <code>StartDate</code> содержит неправильный формат для данных начала. Из Windows измените формат данных компьютера для <code>mm/dd/yy</code>
Сообщение об ошибке: <i>No log files found</i> (не обнаружено файлов регистрации)	Отсутствуют файлы регистрации для запрашиваемых данных в пути, указанном с помощью пункта <code>DataDir</code> .
Сообщение об ошибке <i>Could not find tagname TAG• in database</i>	Запрашиваемый признак не существует в словаре <code>Tagname</code> приложения. Проверьте то, что имя признака правильно написано.

Сообщения об ошибке или состоянии	Причина и/или решение
<p>Сообщение об ошибке:  <i>Could not find tagname.x in: C:\IT6.0B\HISTEST</i>            (не может найти tagname.x в: C:\IT6.0B\HISTEST)</p>	<p>Файл tagname.x не существует или он нарушен</p>
<p>Не создается выходной файл .csv и не появляется ошибок</p>	<ul style="list-style-type: none"> <li>• HistData не находится.</li> <li>• Пункт Tags не перечисляет какие либо признаки, которые указываются ля регистрации.</li> <li>• HDWritefile не правильно определен. Убедитесь, что признак является целым признаком, DDE Access Name правильный и пункт – это WriteFile. Также убедитесь, что нем масштабирования там, где MinEU=MinRaw И MaxEU=MaxRaw.</li> </ul>
<p>Выходной файл .csv содержит штампы даты и времени, но не содержит какие либо данные по регистрации для запрашиваемых признаков.</p>	<p>Отсутствуют вводы в исторической регистрации для признаков во время периода времени запроса. Отобразите историческую тенденцию для проверки, содержит ли файл регистрации данные за период запроса.</p>
<p>WWLogger содержит следующее сообщение:  <i>Error for DDE HistData   Viewstream1! WriteFile: Poke was rejected by the server.</i>            ( Ошибка для DDE HistData   Viewstream1! WriteFile: запись в запоминающее устройство запрещена сервером).</p>	<p>Сообщение об ошибке записывается в WWLogger каждый раз, как не происходит создания файла .csv из-за ошибки присваивания значения для пунктов HistData. Эта ошибка может также произойти, если вы пытаетесь установить пункт WriteFile в 0, или если вы пытаетесь провести запись в пункт с ошибкой.</p>
<p>Файл .csv содержит только одну запись, когда должно быть много записей из файла регистрации.</p>	<p>Пункту Interval может присваиваться неправильная величина, которая формирует очень малый интервал сбора. Также пункт Duration может иметь неправильный формат, такой, как HDWDuration=1- (не указан инкремент).</p>

# Приложение А

## Десятичные блоки IEEE

InTouch HMI использует 754 стандарт Института инженеров по электротехнике и электронике (IEEE) 754 для преобразования 32-х разрядных двоичных величин в числа с плавающей десятичной точкой.

32-х разрядные числа IEEE 754 сохраняются в 16-ти разрядных программируемых логических контроллерах (PLCs), как 2 16-ти разрядного слова. Регистры десятичной точки в PLC обычно являются последовательными в их схеме нумерации для старшего и младшего шестнадцатиричных слов. Современное поколение 32-х разрядных персональных компьютеров использует один 32-х разрядный регистр. Схема нумерации битов регистра имеет тот же формат, что и для двух последовательных 16-ти разрядных регистров. Для использования чисел с плавающей точкой в приложении InTouch сервер ввода/вывода должен быть способным преобразовать значения. Сохраненные в двух 16-ти разрядных регистрах PLC. Преобразование разрядов должно выполняться, потому что InTouch всегда рассматривает неисправленные значения регистра PLC, как отдельные целые. Невозможно выполнить булеву операцию AND для двух целых из регистра и преобразовать их в действительное число. InTouch не может выполнить преобразование типа для регистров с двойным целым.

## Отображение чисел с плавающей точкой в InTouch НМІ

InTouch НМІ использует 32-разрядный формат с плавающей точкой IEEE для отображения действительных чисел в приложении. Формат с плавающей точкой IEEE является только приближением фактического действительного числа. За исключением случая действительного числа равного степени двух, оно не может быть представлено точно с использованием 32-разрядного формата с плавающей точкой IEEE. Точность 32-разрядного числа с плавающей точкой IEEE составляет приблизительно восемь десятичных мест. Когда вы хотите показать действительное число в приложении InTouch, убедитесь, что число не превышает восемь цифр. Форматы числа с плавающей точкой показывают правильные действительные числа для приложения InTouch:

`#####`

`### #####`

`#####.###`

Любые числа с плавающей точкой с числом цифр более восьми имеют ошибку округления.

### Пример

Приложение InTouch будет отображать действительное число 2.3.

Но число 2.3 не является степенью двух и не может быть точно представлено 32-разрядным форматом с плавающей точкой IEEE свыше 8 десятичных цифр. Для обеспечения того, что величина 2.3 показывается из приложения, как символы 2.3 ASCII, число не должно превышать восемь цифр. Если число превышает максимум восемь цифр, результирующее число может быть показано, как 2.29999999 или, вместо него, 2.30000001.

# Приложение В

## Лицензирование InTouch

Система Wonderware предлагает несколько лицензий InTouch. Эти лицензии основаны на различном количестве тегов, которые можно использовать при выполнении приложений. Необходимо понимать, как подсчитываются теги в схеме лицензий InTouch. Имеется несколько функций, которые можно использовать для подсчета предполагаемого количества удаленных ссылочных тегов в приложении.

---

**Указание** - Номера лицензий могут быть изменены в любой момент.

---

### Счетчики тегов в лицензиях

Во время работы приложений InTouch описатели тегов (tag handle) хранятся в базе данных, располагающейся в памяти, объемом 64 К. Каждому тегу должен быть назначен описатель. Описатели тегов инициализируются и используются Табло Окно (WindowViewer), но не сохраняются на диске по окончании работы приложения.

В вышеупомянутой рабочей базе данных теоретически может храниться 65535 описателей тегов, включая как локальные теги, так и теги, ссылающиеся на удаленные источники тегов. Приложение InTouch ни в какой момент не может иметь больше тегов, активных одновременно, чем имеется описателей в памяти в вышеуказанной рабочей базе данных. Лицензия InTouch устанавливает, скольким локальным и удаленным тегам можно присвоить описатели в рабочей базе данных.

Кроме того, максимальное количество активных тегов в приложении ограничивается функциональными пределами рабочей базы данных. Фактическое максимальное значение счетчика тегов меньше, чем теоретически возможная максимальная величина – 65535 описателей тегов. На рисунке, приведенном ниже, показано, как определяется фактическое максимальное значение счетчика тегов InTouch. Из максимально возможного количества тегов вычитается несколько постоянных величин.

$$61404 = (65535 - (1 + 34 + 4096))$$

Индекс  
неправильного  
описателя тегов

Распределение  
памяти для  
тегов-заполнителей

Теоретическое  
максимальное  
значение  
счетчика тегов

Количество  
системных  
тегов

- Разряд неправильного описателя тегов зарезервирован, чтобы в случае необходимости указать на возникновение неправильного значения описателя тега в рабочей базе данных Табло Окн.
- Система InTouch версии 10 включает 34 системных тега, которые не могут быть заменены тегами, заданными пользователем. Если выполнен перевод (миграция) приложения версии 7.11 (или более ранней) в текущую версию InTouch, то значение счетчика системных тегов будет составлять 37.
- При конфигурировании резервируется 4096 описателей базы данных для хранения тегов-заполнителей. Когда во время конфигурирования выполняется импорт окон, скриптов или символов, теги-заполнители назначаются этому сегменту памяти. Во время работы все 4096 описателей заполнителей могут назначаться удаленным ссылочным тегам.

Wonderware предлагает ряд возможностей лицензии InTouch, основанных на максимальном количестве локальных и удаленных ссылочных тегов, которые можно использовать в приложении.

Некоторые типичные возможности лицензий Wonderware для InTouch приведены в таблице ниже. Следует обратить внимание, что в таблице указаны не все возможности лицензий, предлагаемые Wonderware.

Максимальное значение счетчика локальных тегов	Максимальное значение счетчика удаленных ссылочных тегов	Максимальное значение общего счетчика тегов
512	256 фиксированных тегов	768
8192	4096 фиксированных тегов	12288
61404	60000 активных тегов	60000 (60 К)

В первом примере, показанном в таблице, можно создать приложение, в котором можно будет использовать до 512 локальных тегов, где 512 – предельное максимальное количество тегов в лицензии. Можно также добавить до 256 удаленных ссылочных тегов. Лицензия ограничивает значение общего счетчика тегов числом 768 (сумма величин, задающих максимальные значения счетчиков локальных и удаленных ссылочных тегов).

В первых двух примерах, приведенных в таблице, лицензии устанавливают значения счетчиков удаленных ссылочных тегов фиксированным образом. Фиксированный тег (sticky tag) – это удаленный тег-ссылка, который связывается во время работы, когда Табло Окон принимает уведомление об изменении данных для удаленной ссылки. Табло Окон во время работы обновляет удаленные теги-ссылки до предельного максимального количества, установленного лицензией InTouch. Табло Окон не обновляет дополнительные удаленные теги-ссылки сверх предельного количества, установленного лицензией. При закрытии окна Табло Окон не уменьшает счетчик удаленных ссылочных тегов. Таким образом, счетчик удаленных ссылочных тегов фиксируется на время работы приложения.

При превышении максимального значения счетчика удаленных ссылочных тегов, заданного лицензией InTouch, выводится одиночное сообщение. После того, как достигается предельное максимальное количество, заданное лицензией, величины, связанные с неправильными удаленными ссылочными тегами, больше не обновляются в приложении. Чтобы иметь возможность открыть новые окна, включающие один или несколько удаленных тегов-ссылок, уже не связанных с тегами, подсчитанными относительно заданного лицензией предела, необходимо закрыть приложение и запустить его заново.

Обозначение лицензии 60 К, приведенное в таблице, в некоторой степени неправильное. Общее максимальное количество тегов не ограничивается числом 60000. Максимально возможное количество удаленных ссылочных и локальных тегов определяется на основе предела реализации, определяемого рабочей базой данных InTouch; этот предел несколько больше, чем 60000.

При использовании лицензии 60 К пределы реализации для локальных и удаленных ссылочных тегов следующие:

- общее возможное количество локальных тегов:  
 $61404 = 65535 - (4096 + 1 + 34)$
- общее возможное количество удаленных ссылочных тегов:  
 Максимум =  $65535 - (4096 + 1 + \text{количество\_локальных\_тегов})$

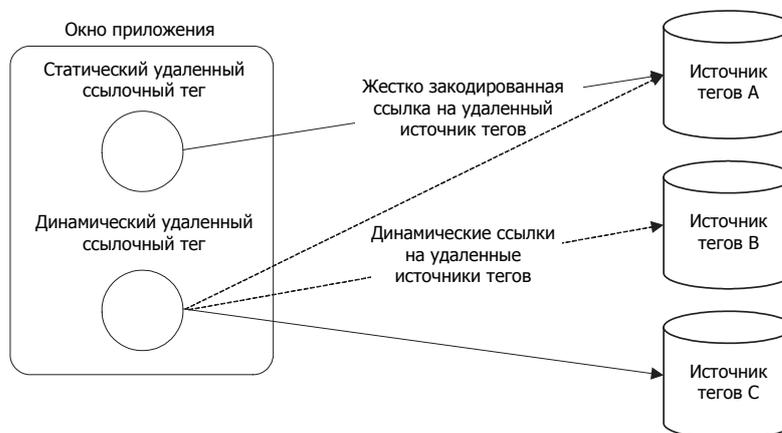
Таким образом, при запуске приложения с лицензией 60 К фактически выполняется обмен потенциальных удаленных ссылок на пользовательские теги в базе данных локальных тегов. В конфигурации для работы приложения количество возможных удаленных ссылочных тегов никогда не оказывается меньше 4096.

## Ограничения на количество удаленных ссылок InTouch

Имеются два типа удаленных ссылочных тегов. Статические удаленные ссылки жестко программируются на фиксированный удаленный адрес при задании тега из Словаря Имен Тегов. Статической удаленной ссылке назначается описатель тега в базе данных тегов при запуске приложения на выполнение. На время работы приложения счетчик статических удаленных ссылочных тегов фиксируется.

Динамические удаленные ссылки определяют адрес получателя во время работы приложения. Если динамический удаленный ссылочный тег назначается описателю базы данных, то адрес получателя можно изменять во время работы, используя точечное поле .Reference или функцию IOSetRemoteReference() в скрипте.

На рисунке, приведенном ниже, показан пример работы приложения InTouch с лицензией 60 К без фиксированных удаленных ссылочных тегов. Счетчик статических удаленных ссылочных тегов фиксируется на время работы приложения. Однако счетчик динамических удаленных ссылочных тегов относится только к активному источнику тега. Предыдущие соединения с источниками удаленных тегов не фиксируются и не учитываются при определении значений счетчиков удаленных ссылочных тегов или общего количества тегов.



В лицензии InTouch на 60 К не используются фиксированные счетчики тегов, налагающие ограничения на количество динамических удаленных тегов-ссылок. Это позволяет приложению за время его выполнения иметь динамический доступ к более чем 60 К тегов. Счетчик использования тегов для динамических удаленных ссылок увеличивается и уменьшается по мере того, как окна с удаленными ссылками открываются или закрываются. В то же время количество тегов, одновременно активных в приложении, не может превышать предела реализации, определяемого рабочей базой данных тегов.

Удаленные ссылки от тегов ввода-вывода не включаются в фиксированный счетчик удаленных ссылок, определяемый согласно лицензии InTouch. Удаленная ссылка тега ввода-вывода может изменяться неограниченное количество раз, не учитываясь в значении счетчика, сравниваемого с предельным количеством фиксированных удаленных ссылок.

Имеется возможность задавать имена доступа (Access Names) InTouch с первичным и вторичным адресом получателя. Вторичный адрес во время выполнения приложения обычно служит в качестве резервного адреса или в качестве тега-источника, используемого для обработки сбоев. При переключении приложения с первичного на вторичный (или наоборот) тег-источник в Имени Доступа связанный с ним фиксированный счетчик удаленного ссылочного тега не увеличивается, несмотря на изменение адреса Имени Доступа.

## Функции счетчика удаленных тегов

Человеко-машинный интерфейс InTouch HMI включает ряд функций, позволяющих проверять, соответствуют ли используемые приложения требованиям удаленного тега лицензии, имеющейся у пользователя. Можно написать временные скрипты, включающие эти функции, чтобы проверять приложения на предмет выявления в них возможных проблем, связанных с лицензиями, прежде чем вводить эти приложения в производственную эксплуатацию. После того, как приложение будет проверено на наличие таких проблем, и будет установлено их отсутствие, скрипты следует удалить.

### Функция IORRGetSystemInfo()

Возвращает счетчик тегов для выполняемого приложения InTouch. В зависимости от заданного аргумента числовое значение, возвращаемое функцией IORRGetSystemInfo(), может представлять собой следующее:

- максимальное количество адресов удаленных тегов, заданное лицензией InTouch;
- количество адресов удаленных тегов, подсчитанное согласно лицензии за время работы приложения InTouch;
- количество удаленных тегов, активизированных в текущий момент в приложении InTouch;
- количество удаленных тегов, имеющих в работающем приложении InTouch;
- количество удаленных ссылочных тегов, заблокированных (disabled) в текущий момент;
- количество локальных тегов в работающем приложении InTouch;
- -1, если при вызове функции произошла ошибка, или в качестве аргумента функции (Опция) задано недопустимое значение.

#### Категория

Разное

#### Синтаксис

**IORRGetSystemInfo (Опция) ;**

#### Аргумент

##### *Опция*

Целочисленный тег или целочисленная константа, задающая тип счетчика удаленных ссылочных тегов, который должен быть возвращен функцией. Допустимые значения этого аргумента следующие:

- 
- 1 Функция возвращает максимальное количество разрешенных адресов удаленных тегов на основе лицензии InTouch. При этом локальные теги ввода-вывода не учитываются в счетчике удаленных тегов. Это число является постоянным во время работы приложения InTouch.

---

  - 2 Функция возвращает количество уникальных адресов удаленных тегов, подсчитанных относительно заданных лицензией пределов. Подсчитываются теги, активизированные во время работы приложения InTouch. Локальные теги ввода-вывода не учитываются в счетчике удаленных тегов.  
Если лицензия разрешает более 60000 удаленных ссылочных тегов, то значение, возвращаемое функцией, может быть равно нулю, независимо от того, сколько адресов удаленных тегов активизировано. Работая на условиях неограниченной лицензии, Табло Алармов не подсчитывает активизированные адреса удаленных тегов. Если приложение работает на условиях лицензии, имеющей ограничение на количество удаленных тегов, то данный счетчик может увеличиваться до предела, установленного лицензией для счетчика удаленных тегов. После того, как этот предел достигается, активизация каких-либо дополнительных адресов удаленных тегов невозможна. Только уже активизированные адреса могут активизироваться заново. Чтобы получить перечень удаленных ссылочных тегов, подсчитанных относительно заданного лицензией предельного количества, следует использовать функцию IORRWriteState с аргументом, равным 3.

---

  - 3 Функция возвращает количество удаленных ссылочных тегов, активизированных в текущий момент в приложении InTouch.

---

  - 4 Функция возвращает количество удаленных ссылочных тегов, которые могут быть активизированы в приложении InTouch без превышения количества описателей удаленных тегов. Этот счетчик обычно изменяется в ходе работы приложения. На значение этого счетчика влияет завершение и запуск скриптов, открытие и закрытие окон, содержащих удаленные ссылочные теги.  
Это число может быть меньше числа, еще остающегося доступным согласно лицензии, особенно если лицензия неограниченная. Это происходит потому, что имеется внутреннее ограничение на количество удаленных ссылочных тегов, которые могут быть активизированы одновременно.

---

  - 5 Функция возвращает количество удаленных тегов в приложении InTouch, находящихся в текущий момент в заблокированном состоянии (disabled).

---

  - 6 Функция возвращает текущее количество локальных тегов в приложении InTouch.
-

### Пример

В приведенном примере возвращается количество удаленных ссылочных тегов, подсчитанных относительно заданного лицензией предела во время работы приложения In-Touch. Возвращаемое значение счетчика ссылочных тегов присваивается целочисленному тегу `RRTagCount`.

```
RRTagCount = IORRGetSystemInfo(2);
```

## Функция `IORRWriteState()`

Сохраняет в текстовом файле информацию о текущем состоянии удаленных тегов приложения. Если указанный файл отсутствует, он создается. Каждый раз при выполнении скрипта, содержащего эту функцию, новая информация добавляется в файл.

Можно указать, какая именно информация об удаленных тегах должна сохраняться в файле. Кроме того, функцией возвращается величина, указывающая, успешно ли выполнено сохранение информации в файле.

### Категория

Разное

### Синтаксис

```
IORRWriteState(Путевое_имя_файла, Опция, " ");
```

### Аргументы

*Путевое\_имя\_файла*

Полное путевое имя текстового файла, содержащего информацию об удаленных тегах приложения. Этот аргумент может представлять собой строковую константу или тег-сообщение.

### *Опция*

Целочисленный тег или целочисленная константа, задающая содержимое информации об удаленных тегах, которая должна быть записана в файл. Допустимые значения этого аргумента следующие:

- 
- |       |   |
|-------|---|
| 1     | Перечень текущих адресов удаленных тегов. Сохраняемая информация включает также состояние, время активизации и деактивизации каждого удаленного тега. |
| <hr/> |   |
| 2     | Перечень всех адресов, активных в текущий момент, с указанием времени активизации.  |
-

- 
- 3 Перечень всех адресов удаленных ссылочных тегов, которые были активизированы и подсчитаны относительно предела, заданного лицензией. По мере того, как активизируются новые адреса удаленных ссылочных тегов, в список добавляются новые элементы. При достижении предела, заданного лицензией, новые элементы перестают добавляться в список. Если лицензия не имеет ограничения на количество удаленных тегов, адреса в этот список не добавляются.
- 
- 4 Перечень текущих адресов, не активизированных из-за того, что счетчик удаленных ссылочных тегов превысил предел, установленный лицензией, или из-за того, что достигнут внутренний предел по количеству описателей тегов. Если лицензия не имеет ограничения на количество удаленных тегов, то адреса, связанные с лицензированием, не возвращаются. Для неограниченной лицензии список содержит элементы, не активизированные из-за ограничений реализации. Если какой-либо из этих элементов активизируется, он удаляется из данного списка. Если элемент деактивизируется из-за ограничений, связанных с лицензированием, он удаляется из списка. Список обновляется во время работы приложения InTouch.
- 

*Пустая строка “ “*

Этот аргумент зарезервирован для будущего использования, однако должен указываться в функции IORRWriteState() при ее указании в скрипте.

#### **Результаты**

Приведем несколько примеров интерпретации информации, сохраненной в файле вызовом функции IORRWriteState().

#### **Перечень текущих адресов**

Приведенная ниже строка из выходного файла – пример полностью активизированного удаленного ссылочного тега, который может обновляться. В этой строке указано, что адрес 65535 назначен удаленному ссылочному тегу TestProt:di000.

```
65535 <TestProt:di000> (RAA) {C:5/23/2007 9:58:35 AM}
      {A:5/23/2007 9:58:35 AM}
```

После имени удаленного ссылочного тега здесь указаны (в скобках) три флага:

- Первый флаг, R, указывает, что удаленная ссылка тега была успешно разрешена. Если удаленная ссылка тега еще находится в состоянии ожидания, то первому флагу присваивается значение X.
- Второй флаг указывает текущее состояние удаленного тега: активен (A) или неактивен (D).
- Третий флаг указывает состояние адреса: разрешен (A) или запрещен (D) из-за лицензионных ограничений.

Дата, указанная за буквой С, представляет собой дату создания удаленного тега. Дата, указанная за буквой А – дата последней активизации тега. Если удаленный ссылочный тег был деактивизирован во время работы приложения InTouch, то в конце строки указывается время деактивизации.

Следующая строка из выходного файла содержит пример активного удаленного ссылочного тега, выходящего за предел счетчика тегов, установленный лицензией InTouch. Удаленная ссылка тега успешно разрешена, и тег в текущий момент времени активен:

```
65414 <TestProt:di121> (RAD) {C:5/23/2007 9:58:35 AM}
      {A:5/23/2007 9:58:35 AM}
```

В то же время значение тега в приложении InTouch не обновляется, так как адрес, назначенный удаленному ссылочному тегу, выходит за предел счетчика тегов, установленный лицензией InTouch.

#### Перечень активных адресов

Приведенная ниже строка из выходного файла – пример полностью активизированного удаленного тега, которому назначены величины, обновляющие приложение InTouch:

```
65429 <TestProt:di106> (A) {C:5/23/2007 9:58:35 AM}
      {A:5/23/2007 9:58:35 AM}
```

Первое число указывает описатель удаленного тега. За ним указывается адрес, затем – флаги (А – разрешен, D – запрещен). Затем указывается время создания, последней активизации и последней деактивизации. Время деактивизации не указывается, если удаленный тег за время работы приложения не был деактивизирован.

Следующая строка из выходного файла содержит пример активного удаленного тега, выходящего за пределы, установленные лицензией:

```
65342 <TestProt:di193> (D) {C:5/23/2007 9:58:35 AM}
      {A:5/23/2007 9:58:35 AM}
```

#### Лицензированные адреса

Приведенная ниже строка из перечня содержит адрес, назначенный удаленному ссылочному тегу, и время его включения в перечень:

```
<testprot:di000> {C:5/23/2007 9:58:36 AM}
```

#### Запрещенные адреса

Запрещенные адреса указываются в перечне из-за ограничений реализации или из-за превышения счетчиком тега максимальной величины, установленной лицензией.

В следующей строке приведен пример адреса удаленного тега, выходящего за предел, установленный лицензией:

```
testprot:di125 [1] (L) {F:5/23/2007 9:58:39 AM}
      {R:5/23/2007 9:58:39 AM}
```

Адрес указывается вместе со счетчиком, чтобы показать, сколько раз делалась попытка ссылки на элемент. Флаг указывает причину внесения адреса в перечень: превышение предела, установленного лицензией (L), или внутренний предел реализации (I). Указываются также две отметки времени: первая из них указывает время первого добавления элемента в перечень, вторая – время последнего доступа.

#### Пример

В данном примере адреса удаленных тегов, активные в текущий момент, записываются в файл, расположенный в папке c:\intouch\data. Tегу ReturnValue присваивается целое число, которое указывает, удалось ли успешно выполнить запись информации об удаленных тегах в файл.

```
ReturnValue = IORRWriteState("c:\intouch\data", 2,
    ""); Функция IORRGetItemActiveState()
```

Возвращает состояние заданного адреса удаленного тега.

## Функция IORRGetItemActiveState()

#### Категория

Разное

#### Синтаксис

```
IORRGetItemActiveState (Путевое_имя_элемента, Опция);
```

#### Аргументы

*Путевое\_имя\_элемента*

Строка, представляющая собой адрес, о котором требуется получить информацию. Этот аргумент может представлять собой строковую константу или тег-сообщение.

#### *Опция*

Целочисленный тег или целочисленная константа, задающая тип счетчика удаленных ссылочных тегов, который требуется вернуть. Допустимые значения этого аргумента следующие:

- |   |   |
|---|---|
| 1 | Определяется, активен ли в данный момент текущий адрес удаленного тега.<br>Возвращается одно из следующих значений: 1, если адрес является текущим и активным; -1, если адрес не текущий; 0, если адрес текущий, но не активный.  |
| 2 | Определяется, был ли текущий адрес удаленного тега активизирован когда-либо во время работы приложения.<br>Возвращается одно из следующих значений: 1, если адрес является текущим и был активизирован хотя бы один раз; -1, если адрес не текущий; 0, если адрес текущий, но ни разу не был активизирован. |

- 
- 3 Определяется, был ли текущий адрес удаленного тега когда-либо деактивизирован.  
Возвращается значение -1, если адрес не текущий; -1, если адрес текущий, но ни разу не был деактивизирован.
- 
- 4 Определяется, заблокирован ли текущий адрес удаленного тега.  
Возвращается значение 1, если адрес является текущим и деактивизирован хотя бы один раз. Возвращается значение -1, если адрес не текущий; 0, если адрес не заблокирован; 1, если адрес текущий и заблокирован (disabled).
- 
- 5 Определяется, находится ли адрес в перечне разрешенных адресов (allowed list).  
Возвращается значение 0, если адрес не находится в перечне разрешенных адресов, 1 – если адрес находится в этом перечне.
- 
- 6 Определяется, находится ли адрес в перечне запрещенных адресов (disallowed list).  
Возвращается значение 0, если адрес не находится в перечне запрещенных адресов, 1 – если адрес находится в этом перечне.
- 

#### Примеры

В следующем примере определяется, является ли в данный момент активным адрес удаленного тега TestProt:di000:

```
ReturnValue = IORRGetItemActiveState("TestProt:di000",  
1);
```

В следующем примере определяется, заблокирован ли в данный момент адрес удаленно-го тега TestProt:di121:

```
ReturnValue = IORRGetItemActiveState("TestProt:di121",  
4);
```

В следующем примере определяется, подсчитан ли в данный момент адрес удаленного тега TestProt:di001 относительно предела, установленного лицензией:

```
ReturnValue = IORRGetItemActiveState("TestProt:di001",  
5);
```

Руководство по управлению данными  
человеко-машинного интерфейса InTouch  
Статистика изменений

Апрель 2008

Версия 1.0

Первая версия

**KLINKMANN**

[www.klinkmann.com](http://www.klinkmann.com)

**Helsinki**

ph. +358 9 540 4940  
[automation@klinkmann.fi](mailto:automation@klinkmann.fi)

**Санкт-Петербург**

тел. +7 812 327 3752  
[klinkmann@klinkmann.spb.ru](mailto:klinkmann@klinkmann.spb.ru)

**Москва**

тел. +7 495 641 16 16  
[moscow@klinkmann.spb.ru](mailto:moscow@klinkmann.spb.ru)

**Екатеринбург**

тел. +7 343 376 53 93  
[yekaterinburg@klinkmann.spb.ru](mailto:yekaterinburg@klinkmann.spb.ru)

**Самара**

тел. +7 846 993 49 33  
[samara@klinkmann.spb.ru](mailto:samara@klinkmann.spb.ru)

**Київ**

тел. +38044 495-33-40  
[klinkmann@klinkmann.kiev.ua](mailto:klinkmann@klinkmann.kiev.ua)

**Мінск**

тел. +375 17 2000876  
[minsk@klinkmann.com](mailto:minsk@klinkmann.com)

**Rīga**

tel. +371 738 1617  
[klinkmann@klinkmann.lv](mailto:klinkmann@klinkmann.lv)

**Tallinn**

tel. + 372 6 684 500  
[klinkmann.est@klinkmann.ee](mailto:klinkmann.est@klinkmann.ee)

**Vilnius**

tel. +370 5 215 1646  
[post@klinkmann.lt](mailto:post@klinkmann.lt)