



www.klinkmann.com

Wonderware® FactorySuite™

Справочное руководство по InTouch™

Редакция С
Дата пересмотра: август 2005

Invensys Systems, Inc.

© 2006 Klinkmann. Все права защищены.

www.klinkmann.com



Finland
P.O. Box 38, FI-00371 Helsinki
Ph. +358 9 540 4940
Fax +358 9 5413 541
automation@klinkmann.fi

Санкт-Петербург	тел. +7 812 327 3752; klinkmann@klinkmann.spb.ru
Москва	тел. +7 495 461 3623; moscow@klinkmann.spb.ru
Tallinn	tel. + 372 668 4500; klinkmann.est@klinkmann.ee
Rīga	tel. +371 738 1617; klinkmann@klinkmann.lv
Vilnius	tel. +370 5 215 1646; post@klinkmann.lt
Київ	тел. +38 044 239 1250; klinkmann@klinkmann.kiev.ua

Все права защищены. Дублирование, хранение в справочной системе, а также передача настоящего руководства, как целиком, так и частями, в любом виде (электронном, печатном, фотографическом и ином другом) без предварительного письменного согласия со стороны Invensys Systems, Inc. запрещается. Никакая ответственность по авторским правам и патентам в результате использования информации, содержащейся в настоящем документе, не возникает. Несмотря на то, что при подготовке настоящего руководства и соблюдались все требуемые меры контроля, ни издатель, ни авторы не несут никакой ответственности за возможные ошибки или опечатки. Кроме того, не предполагается возникновения никакой ответственности за ущерб, причинённый использованием информации, которая содержится в настоящем руководстве.

Информация, приведённая в настоящем руководстве, может модифицироваться и корректироваться без какого-либо предварительного уведомления и ни в какой мере не представляет собой какие-либо обязательства со стороны Invensys Systems, Inc. Описываемое в данном документе программное обеспечение поставляется в соответствии с условиями лицензии или соглашения о нераспространении. Указанное программное обеспечение может использоваться и копироваться только в соответствии с положениями данных документов.

© 2002-2005 Invensys Systems, Inc. Все права защищены.

26561 Rancho Parkway South
Lake Forest, CA 92630 USA
(949) 727-3200
<http://www.wonderware.com>

Торговые марки

Все используемые в настоящем руководстве термины, известные как торговые марки или служебные обозначения, выделены соответствующим образом. Invensys Systems, Inc. не имеет возможности проверить достоверность этой информации. Вызов какого-либо термина в настоящем руководстве не должно рассматриваться как подтверждение достоверности указанной торговой марки или служебного обозначения.

Alarm Logger, ActiveFactory, ArchestrA, Avantis, DBDump, DBLoad, DTAnalyst, FactoryFocus, FactoryOffice, FactorySuite, FactorySuite A2, InBatch, InControl, IndustrialRAD, IndustrialSQL Server, InTouch, InTrack, MaintenanceSuite, MuniSuite, QI Analyst, SCADAAlarm, SCADASuite, SuiteLink, SuiteVoyager, WindowMaker, WindowViewer, Wonderware и Wonderware Logger являются торговыми знаками компании Invensys plc, её дочерних компаний и подразделений. Все остальные наименования могут представлять собой торговые марки соответствующих владельцев.

Содержание

Введение	16
Условные обозначения	16
О руководстве	17
Техническая поддержка	17
Просмотр лицензии FactorySuite	18
Системные теги	19
\$AccessLevel	20
\$AlarmPrinterError	21
\$AlarmPrinterNoPaper	22
\$AlarmPrinterOffline	22
\$AlarmPrinterOverflow	23
\$ApplicationChanged	23
\$ApplicationVersion	24
\$ChangePassword	24
\$ConfigureUsers	25
\$Date	25
\$DateString	26
\$DateTime	26
\$Day	26
\$False	27
\$HistoricalLogging	27
\$Hour	28
\$InactivityTimeout	28
\$InactivityWarning	29
\$Language	29
\$LogicRunning	30
\$Minute	30
\$Month	30
\$Msec	31
\$NewAlarm	31
\$ObjHor	31
\$ObjVer	32
\$Operator	32
\$OperatorDomain	33
\$OperatorDomainEntered	33
\$OperatorEntered	34
\$OperatorName	34
\$PasswordEntered	35

\$Second	35
\$StartDdeConversations	35
\$System	36
\$Time	36
\$TimeString	36
\$VerifiedUserName	37
\$Year	37
Поля "через точку" и свойства	38
Типы тегов	38
Внутренние теги (типа Memory)	38
Теги ввода/вывода	39
Косвенные теги (логические, аналоговые, сообщения)	40
Прочие типы тегов	40
Таблица соответствия типов тегов и полей	40
.Ack	47
.AckDev	48
.AckDsc	49
.AckROC	50
.AckValue	51
.Alarm	52
.AlarmAccess	53
.AlarmAckModel	54
.AlarmClass	55
.AlarmComment	56
.AlarmDate	57
.AlarmDev	58
.AlarmDevCount	59
.AlarmDevDeadband	60
.AlarmDevUnAckCount	61
.AlarmDisabled	62
.AlarmDsc	63
.AlarmDscCount	64
.AlarmDscDisabled	65
.AlarmDscEnabled	66
.AlarmDscInhibitor	67
.AlarmDscUnAckCount	68
.AlarmEnabled	69
.AlarmGroup	70
.AlarmGroupSel	71
.AlarmHiDisabled	72

.AlarmHiEnabled	73
.AlarmHiHiDisabled	74
.AlarmHiHiEnabled	75
.AlarmHiHiInhibitor	76
.AlarmHiInhibitor	77
.AlarmLimit.....	78
.AlarmLoDisabled	79
.AlarmLoEnabled	80
.AlarmLoInhibitor	81
.AlarmLoLoDisabled	82
.AlarmLoLoEnabled.....	83
.AlarmLoLoInhibitor.....	84
.AlarmMajDevDisabled	85
.AlarmMajDevEnabled	86
.AlarmMajDevInhibitor	87
.AlarmMinDevDisabled	88
.AlarmMinDevEnabled	89
.AlarmMinDevInhibitor	90
.AlarmName	91
.AlarmOprName.....	92
.AlarmOprNode.....	93
.AlarmPri.....	94
.AlarmProv	95
.AlarmROC	96
.AlarmROCCount	97
.AlarmROCDisabled	98
.AlarmROCEnabled	99
.AlarmROCInhibitor	100
.AlarmROCUAckCount	101
.AlarmState	102
.AlarmTime	103
.AlarmTotalCount	104
.AlarmType	105
.AlarmUnAckCount	106
.AlarmUserDefNum1	107
.AlarmUserDefNum2	109
.AlarmUserDefStr	111
.AlarmValDeadband.....	113
.AlarmValue.....	114
.AlarmValueCount	115
.AlarmDevUnAckCount.....	116
.Caption	117

.ChartLength.....	118
.ChartStart.....	119
.Comment.....	120
.DevTarget.....	121
.DisplayMode.....	122
.Enabled.....	123
.EngUnits.....	124
.Freeze.....	125
.HiHiLimit.....	126
.HiHiSet.....	127
.HiHiStatus.....	128
.HiLimit.....	129
.HiSet.....	130
.HiStatus.....	131
.ListChanged.....	132
.ListCount.....	133
.ListIndex.....	134
.LoLimit.....	135
.LoLoLimit.....	136
.LoSet.....	137
.LoLoSet.....	138
.LoLoStatus.....	139
.LoStatus.....	140
.MajorDevPct.....	141
.MajorDevSet.....	142
.MajorDevStatus.....	143
.MaxEU.....	144
.MaxRange.....	145
.MaxRaw.....	146
.MinEU.....	147
.MinorDevPct.....	148
.MinorDevSet.....	149
.MinorDevStatus.....	150
.MinRange.....	151
.MinRaw.....	152
.Name.....	153
.NewIndex.....	155
.NextPage.....	156
.Normal.....	157
.NumAlarms.....	158
.OffMsg.....	159
.OnMsg.....	159

.PageNum	160
.Pen1 – .Pen8.....	161
.PendingUpdates	164
.PrevPage.....	165
.PriFrom	166
.PriTo.....	167
.ProviderReq	168
.ProviderRet	169
.Quality.....	170
.QualityLimit.....	173
.QualityLimitString	173
.QualityStatus	174
.QualityStatusString	175
.QualitySubstatus	176
.QualitySubstatusString.....	177
.QueryState.....	177
.QueryType	178
.RawValue.....	179
.ReadOnly	180
.Reference	181
.ReferenceComplete	182
.ROCPet	183
.ROCSet	184
.ROCStatus.....	185
.ScooterLockLeft.....	186
.ScooterLockRight	187
.ScooterPosLeft	188
.ScooterPosRight.....	189
.Successful.....	190
.SuppressRetain.....	191
.TagID	192
.TimeDate.....	192
.TimeDateString.....	193
.TimeDateTime	193
.TimeDay.....	194
.TimeHour	194
.TimeMinute.....	195
.TimeMonth.....	195
.TimeMsec.....	196
.TimeSecond	196
.TimeTime.....	197
.TimeTimeString	197

.TimeYear.....	198
.TopIndex	199
.TotalPages	200
.UnAck	201
.UpdateCount.....	202
.UpdateInProgress	203
.UpdateTrend.....	204
.Value	205
.Value	206
.Visible	207

QuickScript-функции InTouch 208

Abs().....	208
Ack().....	209
ActivateApp().....	210
AddPermission().....	211
almAckAll().....	212
almAckDisplay()	212
almAckGroup()	213
almAckPriority()	214
almAckRecent().....	215
almAckSelect().....	216
almAckSelectedGroup().....	217
almAckSelectedPriority().....	218
almAckSelectedTag().....	219
almAckTag()	220
almDefQuery()	221
almMoveWindow()	222
almQuery()	223
almSelectAll()	224
almSelectGroup()	225
almSelectionCount().....	225
almSelectItem()	226
almSelectPriority()	227
almSelectTag()	228
almSetQueryByName().....	229
almShowStats()	229
almSupressAll().....	230
almSuppressDisplay()	230
almSuppressGroup().....	231
almSuppressDisplay()	231

almSuppressPriority()	232
almSuppressRetain()	233
almSuppressSelected()	234
almSuppressSelectedGroup()	234
almSuppressTag()	235
almSuppressSelectedGroup()	236
almSuppressSelectedPriority()	237
almSuppressSelectedTag()	238
almUnselectAll()	238
almUnSuppressAll()	239
AnnotateLayout()	239
О функциях APU	240
APUFindAlarmGroupInstance()	241
APUFindFileInstance()	242
APUFindPrinterInstance()	243
APUGetAlarmGroupText()	244
APUGetConfigurationFilePath()	245
APUGetInstanceCount()	246
APUGetPrinterJobCount()	247
APUGetPrinterName()	248
APUGetPrinterStatus()	249
APUGetQueryAlarmState()	250
APUGetQueryFromPriority()	250
APUGetQueryProcessingState()	251
APUGetQueryToPriority()	252
APUInstanceUsed()	253
APUSetAlarmGroupText()	254
APUSetQueryAlarmstate()	255
APUSetQueryFromPriority()	256
APUSetQueryToPriority()	257
APUSetTimeoutValues()	258
APUStartInstance()	259
APUStartQuery()	260
APUStopInstance()	261
APUStopQuery()	262
APUTranslateErrorCode()	263
ArcCos()	263
ArcSin()	264
ArcTan()	264
AttemptInvisibleLogon()	265
ChangePassword()	266
ChangeWindowColor()	267

Cos()	268
DateTimeGMT()	268
DialogStringEntry()	269
DialogValueEntry()	271
DText()	273
EnableDisableKeys()	274
Exp()	274
FileCopy()	275
FileDelete()	276
FileMove()	277
FileReadFields()	279
FileReadMessage()	280
FileWriteFields()	281
FileWriteMessage()	282
GetAccountStatus()	283
GetNodeName()	284
GetPropertyD()	284
GetPropertyI()	285
GetPropertyM()	286
Hide	287
HideSelf	287
HTGetLastError()	288
HTGetPenName()	289
HTGetTimeAtScooter()	290
HTGetTimeStringAtScooter()	291
HTGetValue()	292
HTGetValueAtScooter()	292
HTGetValueAtZone()	294
HTScrollLeft()	295
HTScrollRight()	296
HTSelectTag()	296
HTSetPenName()	297
HTUpdateToCurrentTime()	297
HTZoomIn()	298
HTZoomOut()	299
InfoAppActive()	300
InfoAppTitle()	300
InfoDisk()	301
InfoFile()	302
InfoInTouchAppDir()	303
InfoResources()	303
Int()	304

InTouchVersion()	304
InvisibleVerifyCredentials()	305
IODisableFailover()	305
IOForceFailover()	306
IOGetAccessNameStatus()	306
IOGetApplication()	307
IOGetNode()	307
IOGetTopic()	308
IOReinitAccessName()	308
IOReinitialize()	309
IOSetAccessName()	309
IOSetItem()	311
IOSetRemoteReferences()	312
IOStartUninitConversations()	313
IsAnyAsynchFunctionBusy()	314
IsAssignedRole()	315
Log()	315
LogMessage()	316
LogN()	317
Logoff()	317
LogonCurrentUser()	318
OpenWindowsList()	318
Pi()	318
PlaySound()	319
PostLogonDialog()	320
PrintHT()	320
PrintScreen()	320
PrintWindow()	322
ptGetTrendType()	323
ptLoadTrendCfg()	324
ptPanCurrentPen()	325
ptPanTime()	326
ptPauseTrend()	326
ptSaveTrendCfg()	327
ptSetCurrentPen()	327
ptSetPen()	327
ptSetPenEx()	328
ptSetTimeAxis()	328
ptSetTimeAxisToCurrent()	329
ptSetTrend()	329
ptSetTrendType()	329
ptZoomCurrentPen()	330

ptZoomTime()	330
QueryGroupMembership()	331
RecipeDelete()	332
RecipeGetMessage()	333
RecipeLoad()	334
RecipeSave()	335
RecipeSelectNextRecipe()	336
RecipeSelectPreviousRecipe()	337
RecipeSelectRecipe()	338
RecipeSelectUnit()	339
ReloadWindowViewer()	340
RestartWindowViewer()	340
Round()	340
SendKeys	342
SetPropertyD()	344
SetPropertyI()	345
SetPropertyM()	345
SetWindowPrinter()	346
Sgn()	347
Show	348
ShowAt()	349
ShowHome	350
ShowTopLeftAt()	350
Sin()	351
SPCConnect()	351
SPCDataSetDlg()	352
SPCDisconnect()	352
SPCDisplayData()	353
SPCLocateScooter()	353
SPCMoveScooter()	354
SPCSaveSample()	354
SPCSelectDataset()	355
SPCSelectProduct()	355
SPCSetControlLimits()	355
SPCSetMeasurement()	356
SPCSetProductCollected()	356
SPCSetProductDisplayed()	357
SPCSetRangeLimits()	358
SPCSetSpecLimits()	358
SQLAppendStatement()	359
SQLClearParam()	359
SQLClearStatement()	360

SQLClearTable()	360
SQLCommit()	361
SQLConnect()	362
SQLCreateTable()	364
SQLDelete()	365
SQLDisconnect()	366
SQLDropTable()	366
SQLEnd()	366
SQLErrorMsg()	367
SQLExecute()	367
SQLFirst()	368
SQLGetRecord()	368
SQLInsert()	369
SQLInsertEnd()	370
SQLInsertExecute()	370
SQLInsertPrepare()	371
SQLLast()	371
SQLLoadStatement()	372
SQLManageDSN()	372
SQLNext()	373
SQLNumRows()	373
SQLPrepareStatement()	374
SQLPrev()	374
SQLRollback()	375
SQLSelect()	376
SQLSetParamChar()	379
SQLSetParamDate()	380
SQLSetParamDateTime()	380
SQLSetParamDecimal()	381
SQLSetParamFloat()	381
SQLSetParamInt()	381
SQLSetParamLong()	382
SQLSetParamNull()	383
SQLSetParamTime()	384
SQLSetStatement()	385
SQLTransact()	386
SQLUpdate()	387
SQLUpdateCurrent()	388
Sqrt()	388
StartApp	389
StringASCII()	390
StringChar()	391

StringCompare()	392
StringCompareEncrypted()	392
StringCompareNoCase()	393
StringFromGMTTimeToLocal()	394
StringFromIntg()	396
StringFromReal()	397
StringFromTime()	398
StringFromTimeLocal()	399
StringInString()	400
StringLeft()	401
StringLen()	402
StringLower()	402
StringMid()	403
StringReplace()	404
StringRight()	405
StringSpace()	406
StringTest()	407
StringToIntg()	408
StringToReal()	409
StringTrim()	410
StringUpper()	411
Tan()	411
Text()	412
Trunc()	413
TseGetClientId()	413
TseGetClientNodeName()	414
TseQueryRunningOnClient()	414
TseQueryRunningOnConsole()	415
TseQueryRunningOnClient()	415
UTCDateTime()	416
wcAddItem()	417
wcClear()	417
wcDeleteItem()	418
wcDeleteSelection()	418
wcErrorMessage()	419
wcFindItem()	420
wcGetItem()	421
wcGetItemData()	422
wcInsertItem()	423
wcLoadList()	424
wcLoadText()	425
wcSaveList()	426

wcSaveText()	427
wcSetItemData()	428
WindowState()	429
WWControl()	430
WWExecute().....	431
WWIsDayLightSaving()	432
WWMoveWindow()	432
WWPoke()	433
WWRequest().....	434
WWStringFromTime().....	435

OLE-автоматизация и система InTouch437

Элементы OLE-автоматизации	437
Клиентские расширения OLE-автоматизации в InTouch	438
Ссылки на объекты OLE-автоматизации в InTouch	438
Функция создания объекта OLE_CreateObject()	441
Функция управления объектами OLE_ReleaseObject()	442
Функция проверки действительности объекта OLE_IsObjectValid().....	443
Определение значения свойства	443
Установка значения свойства	444
Вызов методов	446
Обработка ошибок в InTouch	448
Частые ошибки объектов OLE-автоматизации	451

Ошибки выполнения функций InTouch452

Сообщения об ошибках функций управления окнами и обработки распределённых алармов	452
Ошибки исполнения функций рецептурных скриптов	453
Сообщения об ошибках для скриптовых функций утилиты принтера алармов	455
Названия DDE-элементов SPC.....	456
DDE-элементы управления и вывода на экран	457
DDE-элементы текущей выборки SPC	459
DDE-элементы ручного ввода SPC	464
DDE-элементы выделения	466
Ошибки исполнения SQL-функций	469
Коды ошибок взаимодействия с отдельными базами данных.....	471
Oracle	471
Sybase или Microsoft SQL Server	472
dBASE	472

Введение

Настоящее руководство содержит упорядоченную в алфавитном порядке справочную информацию по всем полям тегов, свойствам элементов управления окнами, свойствам объектов алармов, системным тегам и скриптовым функциям QuickScript, входящих в состав программного продукта InTouch. Кроме того, в нём описаны функции дополнительных программ Recipe Manager, SPC Pro и SQL Access Manager. Справочное руководство имеет следующую структуру:

Содержание

- Глава 1, "Системные теги". Содержит описание всех предопределённых системных тегов InTouch.

Примечание. Имена всех системных тегов InTouch начинаются со знака доллара "\$".

- Глава 2, "Поля тегов". Содержит перечень всех полей тегов, простых и распределённых алармов, элементов управления окнами, архивных трендов и других объектов системы.
- Глава 3, "Функции QuickScript InTouch". Содержит описание всех встроенных в систему InTouch функций (символьных, математических, системных, управления окнами и распределёнными алармами, вспомогательных и т.д.).
- Глава 4, "OLE-автоматизация и система InTouch". Содержит краткое описание архитектуры OLE-автоматизации.

Условные обозначения

Пример обозначения	Описание
.ChartStart	Как правило, полужирным шрифтом выделены название полей, системные теги или функции.
<i>TagName</i>	В примерах синтаксиса <i>курсивом</i> обозначены фиктивные аргументы (места, где требуется указание конкретной информации).
{ .HiLimit .HiHiLimit }	Фигурные скобки с вертикальной чертой между ними обозначают возможные варианты параметров.
[ErrorMessage =]	В квадратные скобки заключены необязательные элементы.

Для более лёгкого восприятия каждый системный тэг, свойство, поле "через точку" и поле таблицы скрипта функции сопровождается категорией применения (безопасность, аларм, приложение и т.д.) в соответствии с этой таблицей.

О руководстве

Если при просмотре электронной версии данного руководства вам встретилась ссылка, выделенная зелёным цветом, щёлкнув на ней кнопкой мыши, вы перейдёте к другому разделу или главе. Для возврата в исходное место на экране будет отображена кнопка "назад".

Подсказка. Такие "подсказки" указывают более легкий или быстрый путь выполнения требуемого действия.

Руководство пользователя InTouch поможет ознакомиться со средой разработки и инструментальными средствами WindowMaker (см. Главу 1 "Программные компоненты WindowMaker"). О том, как работать с окнами, графическими объектами, мастер-средствами, элементами ActiveX и т.п., рассказывается в Главе 2 "Работа с WindowMaker". Дополнительные сведения о скриптах InTouch можно найти в Главе 6 "Создание скриптов в системе InTouch".

Подробное описание среды исполнения InTouch (WindowViewer) содержится в *Руководстве пользователя исполнительной среды InTouch*.

Дополнительную информацию о программе SPC Pro можно найти в *Руководстве пользователя SPC Pro*.

Дополнительную информацию о программе Recipe Manager можно найти в *Руководстве пользователя Recipe Manager*.

Дополнительную информацию о программе SQL Access Manager можно найти в *Руководстве пользователя SQL Access Manager*.

Руководство администратора FactorySuite содержит исчерпывающую информацию об общих компонентах FactorySuite и требованиях к системе, советы по организации сети и интеграции продуктов, сведения о технической поддержке и многое другое.

В состав программного пакета FactorySuite входит электронная версия справочной документации по всем компонентам пакета.

Примечание. Для просмотра электронных справочных документов необходимо наличие установленной программы Acrobat Reader версии 4.0 или старше.

Требуемая подготовка пользователя

Настоящее руководство рассчитано на пользователя, который:

- знаком с операционными системами Windows 2000 и/или Windows NT;
- умеет работать с мышью, меню Windows, устанавливать различные параметры и пользоваться справочной системой;
- имеет опыт программирования или знаком с одним из макроязыков. Для достижения наилучших результатов необходимо также иметь представление об основных понятиях программирования, таких как переменные, выражения, функции и методы.

Техническая поддержка

Служба технической поддержки корпорации Wonderware оказывает различные виды помощи, связанные с любыми аспектами использования продуктов Wonderware.

Прежде чем обращаться в службу технической поддержки, попытайтесь найти в *Руководстве пользователя InTouch* возможные решения тех проблем, которые возникли у вас при работе с системой. Если же обращение в службу технической поддержки окажется необходимым, будьте готовы предоставить ей следующую информацию:

1. Серийный номер программного обеспечения.
2. Номер версии используемой системы InTouch.
3. Тип и номер версии используемой операционной системы. Например Microsoft Windows NT workstation версии 4.0 с пакетом обновления SP4.
4. Точный текст полученного сообщения об ошибке.
5. Любые важные на ваш взгляд листинги Wonderware Logger, утилиты Microsoft Diagnostic (MSD) и других диагностических программ.
6. Описание предпринятых вами попыток по устранению проблемы и полученных результатов.
7. Инструкции по воспроизведению проблемы.
8. Номер, присвоенный вашему запросу службой технической поддержки Wonderware (если вы обращаетесь с этой проблемой не первый раз).

Дополнительные сведения о службе технической поддержки можно найти в электронном *Руководстве администратора системы FactorySuite*.

Просмотр лицензии FactorySuite

Информация о лицензии на систему FactorySuite может быть получена при помощи утилиты просмотра лицензии, которая запускается из окна **About (о программе)** справочной системы InTouch.

Для запуска утилиты просмотра лицензии:

1. Выполните команду **About (о программе)** меню **Help (справка)** приложения WindowMaker.
2. Щёлкните кнопкой мыши на пункте **View License (просмотр лицензии)**. Откроется окно утилиты License Utility – **LicView**.

Дополнительные сведения об утилите просмотра лицензии можно найти в *Руководстве администратора системы FactorySuite*.

Г Л А В А 1

Системные теги

В системе InTouch существует набор предопределённых системных тегов. Эти теги являются встроенными и отличаются от других тегов знаком доллара (\$) в начале названия. Во время исполнения прикладной программы система InTouch устанавливает значения этих тегов в ответ на определённые системные события. Системные теги используются таким же образом, как и обычные теги InTouch, например в анимационных функциях и скриптах.

Назначение каждого системного тега InTouch описывается в соответствующем параграфе руководства.

\$AccessLevel

	Определяет уровень доступа вошедшего в систему пользователя.
Категория	контроль доступа
Использование	\$AccessLevel
Примечание	<p>Это тег только для чтения, чьё значение определяется уровнем доступа (AccessLevel), определённым для профиля текущего зарегистрированного пользователя системы InTouch. профиль пользователя можно изменить командой Configure Users (Конфигурирование списка пользователей) в программе WindowViewer.</p> <p>Фактическое числовое значение тега \$AccessLevel неважно для WindowViewer. Вся предполагаемая "защита доступа" должна быть создана во время проектирования системы. Используя значение \$AccessLevel, можно требуемым образом варьировать доступ к различным возможностям системы.</p>
Тип данных	Целый (только для чтения)
Допустимые значения	0–9999
Примеры	<p>Следующее выражение используется для управления видимостью объекта (например кнопки) для его показа пользователю, имеющего необходимый уровень доступа:</p> <pre> \$AccessLevel >= 2000; {С объектами может быть связана функция "запрета", определяемого значением тега \$AccessLevel} \$AccessLevel < 5411; IF \$AccessLevel <= 500 THEN Show "Access Denied"; {всплывающее окно, запрещающее доступ} ELSE Show "Access Granted"; {всплывающее окно, разрешающее доступ} ENDIF; </pre>
См. также	\$Operator, \$OperatorEntered, \$PasswordEntered; \$ConfigureUsers

\$AlarmPrinterError

	Ошибка вывода сведений об аларме на печать.
Категория	алармы
Использование	\$AlarmPrinterError
Примечание	Значение тега равно 1, если есть ошибка при выводе алармов на печать.
Тип данных	Логический (только чтение)
Допустимые значения	0 = Нет ошибки в принтере алармов 1 = Есть ошибка в принтере алармов
Примеры	Следующее выражение, используемое в качестве описания анимационной связи типа "Analog Display Link", выдаст значение 1 при наличии ошибок при выводе сведений об алармах на печать и 0 в противном случае. IF \$AlarmPrinterError == 1 THEN DisplayMessageTag = "Error on alarm printer"; ELSE DisplayMessageTag = "No error on alarm printer"; ENDIF;
См. также	\$AlarmPrinterNoPaper, \$AlarmPrinterOffline, \$AlarmPrinterOverflow

\$AlarmPrinterNoPaper

	Извещение об отсутствии бумаги в принтере
Категория	алармы
Использование	\$AlarmPrinterNoPaper
Примечание	Значение тега равно 1, если в принтере алармов нет бумаги.
Тип данных	Логический (только чтение)
Допустимые значения	0 = В принтере алармов есть бумага 1 = В принтере алармов нет бумаги
Примеры	Следующее выражение, используемое при описании анимационной связи типа "Analog Display Link", выдаст значение 1, если в принтере для вывода алармов на печать нет бумаги, и 0 в противном случае. \$AlarmPrinterNoPaper Использование в скрипте типа OnTrue: Условие: \$AlarmPrinterOffline == 1 Скрипт: show "window"; playsound("c:\winnt\system32\alarm.wav", 1);
См. также	\$AlarmPrinterOffline, \$AlarmPrinterError, \$AlarmPrinterOverflow

\$AlarmPrinterOffline

	Индикация того, что принтер алармов отключен.
Категория	алармы
Использование	\$AlarmPrinterOffline
Примечание	Значение тега равно 1, если принтер отключен.
Тип данных	Логический (только чтение)
Допустимые значения	0 = Принтер алармов включен 1 = Принтер алармов отключен
Примеры	Следующее выражение, используемое при описании анимационной связи типа "Analog Display Link", выдаст значение 1, если принтер отключен, и 0 в противном случае. IF \$AlarmPrinterOffline == 1 THEN Call PLCHorn(); {Функция Quick-скрипта, включающая звуковой сигнал в цехе предприятия} ENDIF;
См. также	\$AlarmPrinterNoPaper, \$AlarmPrinterError, \$AlarmPrinterOverflow

\$AlarmPrinterOverflow

	Индикация того, что буфер принтера алармов переполнен.
Категория	алармы
Использование	\$AlarmPrinterOverflow
Примечание	Значение тега равно 1 при возникновении ошибки принтера алармов (На принтер послано слишком много символов).
Тип данных	Логический (только чтение)
Допустимые значения	0 = Буфер принтера алармов не переполнен 1 = Буфер принтера алармов заполнен и переполняется
Примеры	Следующее выражение, используемое при описании анимационной связи типа "Analog Display Link", выдаст значение 1, если принтер переполнен, и 0 в противном случае. \$AlarmPrinterOverflow
См. также	\$AlarmPrinterNoPaper, \$AlarmPrinterOffline, \$AlarmPrinterError

\$ApplicationChanged

	Сигнализирует об изменении главного приложения в архитектуре NAD (Network Application Development – Сетевая разработка приложений).
Категория	приложение
Использование	\$ApplicationChanged
Примечание	Значение этого тега устанавливается в 1, когда с помощью Notify Clients (Уведомить клиентов) в меню Special (Специальные) приложения WindowMaker генерируется сигнал обновления. Применяется для распределённого обслуживания приложений.
Тип данных	Логический (только чтение)
Примеры	Если следующее выражение встречается в поле тега скрипта по изменению данных, то оно вызывает этот скрипт на выполнение. При этом на экране может быть открыто окно, информирующее пользователя о необходимости перезапустить WindowViewer для того, чтобы новые изменения в программе возымели эффект. \$ApplicationChanged
См. также	\$ApplicationVersion

\$ApplicationVersion

	Содержит номер текущей версии приложения. Этот номер меняется при каждом редактировании тега или Quick-скрипта в приложении.
Категория	приложение
Использование	\$ApplicationVersion
Тип данных	Вещественный (только чтение)
Примеры	При использовании этого выражения в качестве аналоговой анимационной связи типа "Analog Display Link", данный системный тег хранит текущую версию приложения, запущенного из WindowViewer. \$ApplicationVersion
См. также	\$ApplicationChanged

\$ChangePassword

	Выводит на экран диалоговое окно для изменения пароля. Эквивалентно команде Change Password (Изменить пароль) при выборе пункта меню Security (Контроль доступа) меню Special (Специальные) приложения WindowViewer.
Категория	контроль доступа
Использование	\$ChangePassword
Примечание	Установка значения тега в 1 вызывает вывод на дисплей диалогового окна изменения пароля. Когда это окно закрывается, система InTouch автоматически сбрасывает значение тега в 0. Установка тега в значение, отличное от 1, не имеет смысла, и результат в этом случае будет не определен.
Тип данных	Логический (только запись)
Допустимые значения	1
Примеры	Чтобы разрешить пользователю выводить на дисплей окно изменения пароля Change Password , можно создать экранную кнопку с анимационной функцией типа Логический Pushbutton. Для данной кнопки должно быть определено единственное действие "Установить". При нажатии на кнопку системный тег \$ChangePassword будет установлен в "1", что вызовет появление на экране диалогового окна изменения пароля.
См. также	\$AccessLevel, \$OperatorEntered, \$PasswordEntered, \$Operator, \$ConfigureUsers

\$ConfigureUsers

	Выводит на экран общее диалоговое окно Configure Users (Конфигурирование списка пользователей). Равнозначно команде Change Password (Изменить пароль) после выбора пункта Security (Контроль доступа) в меню Special (Специальные) приложения WindowViewer.
Категория	контроль доступа
Использование	\$ConfigureUsers
Примечание	Установка значения тега в 1 вызывает открытие на экране окна конфигурирования пользователей Configure Users. Чтобы окно было открыто, пользователь должен иметь уровень доступа \$AccessLevel > 9000. Когда это окно закрывается, система InTouch автоматически сбрасывает значение тега в 0. Установка тега в значение, отличное от 1, не имеет смысла и результат в этом случае будет не определен.
Тип данных	Логический (только запись)
Допустимые значения	1
Примеры	Чтобы разрешить пользователю выводить на дисплей окно изменения пароля Change Password, можно создать экранную кнопку с анимационной функцией типа Логический Pushbutton. Для данной кнопки должно быть определено единственное действие "Установить". При нажатии на кнопку системный тег \$ChangePassword будет установлен в "1", что вызовет появление на экране диалогового окна изменения пароля.
См. также	\$Operator, \$OperatorEntered, \$ChangePassword, \$PasswordEntered, \$AccessLevel

\$Date

	Содержит целое число дней, прошедших с 1 января 1970 года
Категория	система
Использование	\$Date
Тип данных	Целый (только чтение)
Примеры	<code>StringFromTime((\$Date*86400)+(\$Time/1000), 3);</code> {возвращает текущее время}
См. также	\$DateTime, \$DateString, \$Day, \$Month, \$Year

\$DateString

	Содержит дату в формате, определённом в файле WIN.INI.
Категория	система
Использование	\$DateString
Примечание	Формат даты устанавливается в Панели управления Windows или путем двойного нажатия на поле времени в правом углу панели задач.
Тип данных	Сообщение (Memory message, только чтение)
Примеры	<pre>If StringRight(\$DateString, 2) == "04" THEN LogMessage("The Year 2004!"); ENDIF;</pre>
См. также	\$Date, \$DateTime, \$Day, \$Month, \$Year, \$Time, \$TimeString

\$DateTime

	Содержит дробное число дней, прошедших с 1 января 1970 года
Категория	система
Использование	\$DateTime
Тип данных	Вещественный (только чтение)
Примеры	<pre>dayvalue = 86400 * \$DateTime; If StringFromTime(dayvalue, 4) == "Fri" THEN DisplayMessageTag = "It's Friday!"; ENDIF;</pre>
См. также	\$Date, \$DateString, \$Day, \$Month, \$Year, \$Time, \$TimeString

\$Day

	Содержит текущий день месяца
Категория	система
Использование	\$Day
Примечание	Содержит значения от 1 до 31.
Тип данных	Целый (только чтение)
Примеры	<pre>If \$Day == 15 THEN Show "Mid-Month Washdown Window"; ENDIF;</pre>
См. также	\$DateTime, \$DateString, \$Date, \$Month, \$Year

\$False

	\$False используется, чтобы вернуть значение FALSE (ЛОЖЬ) внутри выражения. Также используется при переносе приложений из более ранних версий InTouch, чтобы заменить вызовы устаревших системных тэгов – \$AlarmPrinterError, \$AlarmPrinterOffline, \$AlarmPrinterOverflow и \$AlarmPrinterNoPaper.
Категория	системный
Вызов	<code>\$False</code>
Комментарии	\$False устанавливает в False (ложь). Этот тэг применим только в случае переноса приложений из более ранних версий InTouch.
Тип данных	Логический (логический), только для чтения

\$HistoricalLogging

	Контролирует и/или управляет запуском и остановкой архивной регистрации данных. Эквивалентно командам Restart Historical Logging (Возобновить архивную регистрацию) или Stop Historical Logging (Остановить архивную регистрацию) меню Special (Специальные) приложения WindowViewer.
Категория	архивация
Использование	<code>\$HistoricalLogging</code>
Примечание	Установка значения тега в 0 останавливает регистрацию данных в архиве. Установка в 1 возобновляет архивную регистрацию. "Запустить" архивную регистрацию с помощью тега \$HistoricalLogging невозможно, если предварительно она не была разрешена опцией Enable Historical Logging (Разрешить архивную регистрацию) окна Historical Logging Properties (параметры архивной регистрации).
Тип данных	Логический (чтение/запись)
Примеры	<pre>IF (InfoDisk("C", 4, \$Second)/1024)<50 THEN {остановить регистрацию архива, если на диске осталось меньше 50 Мбайт} \$HistoricalLogging = 0; ENDIF;</pre>

\$Hour

	Содержит значение часа текущего времени суток
Категория	система
Использование	\$Hour
Примечание	Содержит значения от 0 до 23
Тип данных	Целый (только чтение)
Примеры	Условный скрипт типа On True: \$Hour == 20 AND \$Second == 30 PrintWindow("Day Batch Summary", 1, 1, 0, 0, 0);
См. также	\$DateTime, \$Minute, \$Msec, \$Second, \$Time, \$TimeString

\$InactivityTimeout

	Указывает на то, что интервал времени, после которого будет осуществлено автоматическое завершение сеанса пользователя в результате его бездействия, истёк.
Категория	контроль доступа
Использование	\$InactivityTimeout
Примечание	Значение тега становится равным 1 в момент истечения периода ожидания перед автоматическим завершением сеанса пользователя. Для установки этого времени выполните команду WindowViewer пункта Configure (Конфигурировать) меню Special (Специальные) либо дважды щёлкните кнопкой мыши на строке WindowViewer в группе Configure (Конфигурировать) списка Проводника WindowMaker. Откроется окно WindowViewer Properties (Свойства WindowMaker) с открытой страницей свойств General (Общие). Интервал ожидания указывается в секундах в группе параметров Inactivity (Бездеятельность). Примечание. Таймер бездеятельности не инициализируется для следующих объектов: объекты ActiveX, OLE Automation и мастер-средства SPC.
Тип данных	Логический (только чтение)
Примеры	Условный скрипт типа On True: Условие: \$InactivityTimeout == 1 Скрипт: Show "You have been logged off window";
См. также	\$InactivityWarning

\$InactivityWarning

	Указывает на то, что интервал времени перед выводом предупреждения об автоматическом завершении сеанса пользователя истек.
Категория	контроль доступа
Использование	\$InactivityWarning
Примечание	<p>Значение тега становится равным 1, когда наступает момент автоматического завершения сеанса пользователя в результате его бездействия. Таймер бездействия сбрасывается <u>только</u> щелчком кнопки мыши либо нажатием на любую клавишу. Для установки этого времени выполните команду WindowViewer пункта Configure (Конфигурировать) меню Special (Специальные) либо дважды щёлкните кнопкой мыши на строке WindowViewer в группе Configure (Конфигурировать) списка Проводника WindowMaker. Откроется окно WindowViewer Properties (Свойства WindowMaker) с открытой страницей свойств General (Общие). Интервал ожидания указывается в секундах в группе параметров Inactivity (Бездеятельность).</p> <p>Примечание. Таймер бездействия не инициализируется для следующих объектов: объекты ActiveX, OLE Automation и мастер-средства SPC.</p>
Тип данных	Логический (только чтение)
Примеры	<p>Условный скрипт типа On True:</p> <pre>If \$InactivityWarning == 1 THEN Show "You are about to be logged off-window"; ENDIF;</pre>
См. также	\$InactivityTimeOut

\$Language

	Если для приложения InTouch задано несколько языков, \$Language показывает значение идентификатора языка (Language ID) текущего показываемого языка. По умолчанию это идентификатор языка (локальный идентификатор) базовой системы InTouch (E/F/G/J/SC). Установка этого тэга на другой идентификатор заставляет строки с определёнными значениями переключаться на новый язык.
Категория	системный
Тип данных	Целый (чтение / запись)

\$LogicRunning

	Следит за и/или управляет выполнением скриптов. Эквивалентно командам Start Logic (Запустить скрипт) либо Halt Logic (Остановить скрипт) меню Logic (Логика) в WindowViewer. Примечание. Выполняемые асинхронные скрипты не могут быть остановлены, однако можно запретить выполнение новых скриптов.
Категория	система
Использование	\$LogicRunning
Примечание	Установка значения тега в 1 запускает скрипт QuickScript. Установка значения тега в 0 останавливает выполнение скрипта.
Тип данных	Логический (чтение/запись)

\$Minute

	Содержит значение минут текущего часа.
Категория	система
Использование	\$Minute
Примечание	Имеет значения от 0 до 59.
Тип данных	Целый (только чтение)
Примеры	В анимационной функции вывода символьной строки: IF InfoFile("C:\InTouch.32\WIZ.INI", 1, \$Minute) == 1 THEN LogMessage("The File Exists!"); ENDIF;
См. также	\$DateTime, \$Hour, \$Msec, \$Second, \$Time, \$TimeString

\$Month

	Содержит текущий месяц года.
Категория	система
Использование	\$Month
Примечание	Имеет значения от 1 до 12.
Тип данных	Целый (только чтение)
Примеры	IF \$Month == 10 THEN CurrentMonthName = "October"; ENDIF;
См. также	\$Date, \$Day, \$Year

\$Msec

	Содержит текущее число миллисекунд. Частота обновления этого тега устанавливается параметрами Tick Interval (Длительность тика) и Update Time (Интервал обновления). Для установки этого времени выполните команду WindowViewer пункта Configure (Конфигурировать) меню Special (Специальные) либо дважды щёлкните кнопкой мыши на строке WindowViewer в группе Configure (Конфигурировать) списка Проводника WindowMaker. Откроется окно WindowViewer Properties (Свойства WindowMaker) с открытой страницей свойств General (Общие).
Категория	система
Использование	\$Msec
Примечание	Имеет значения от 0 до 999.
Тип данных	Целый (только чтение)
См. также	\$DateTime, \$Minute, \$Second, \$Hour, \$Time, \$TimeString

\$NewAlarm

	Указывает на то, что возник новый аларм.
Категория	алармы
Использование	\$NewAlarm
Примечание	Тег принимает значение 1 в момент возникновения нового аларма. Изменение значение выполняется только для локальных, но не удалённых алармов.
Тип данных	Логический (чтение/запись)
Примеры	Данный тег можно связать с логической функцией PlaySound, которая выдает звуковой сигнал тревоги. Для сброса значения тега в 0 и подтверждения аларма можно создать экранную кнопку подтверждения.

\$ObjHor

	Содержит координату центра выбранного объекта по горизонтали в пикселях.
Категория	система
Использование	\$ObjHor
Тип данных	Целый (только чтение)
См. также	\$ObjVer

\$ObjVer

	Содержит координату центра выбранного объекта по вертикали в пикселях.
Категория	система
Использование	\$ObjVer
Тип данных	Целый (только чтение)
См. также	\$ObjHor

\$Operator

	Управляет способностью пользователя выполнять те или иные функции.
Категория	система
Использование	\$Operator
Примечание	Содержит имя пользователя, зарегистрированного в текущий момент в системе.
Тип данных	Сообщение (только чтение)
Примеры	Следующий скрипт позволяет управлять доступом к конкретному окну: IF \$Operator == "DayShift" THEN Show "Control Panel Window"; ELSE Show "Wrong Operator"; ENDIF;
См. также	\$OperatorEntered, \$AccessLevel, \$PasswordEntered, \$ChangePassword, \$ConfigureUsers

\$OperatorDomain

Категория	контроль доступа
Использование	При установке режима контроля доступа средствами операционной системы после успешной регистрации оператора в этом теге будет храниться название домена либо компьютера, которое было указано в момент регистрации. При установке режима контроля доступа средствами ArchestrA после успешной регистрации оператора в этом теге будет храниться строка "ArchestrA". При установке режима контроля доступа средствами InTouch в этом теге будет храниться строка "InTouch". При выборе "None" в качестве режима контроля доступа значением этого тега будет пустая строка "".
Тип данных	Символьный
Примеры	<code>\$Operator = "john";</code> <code>\$OperatorDomain = "CORPORATE_HQ";</code>
См. также	<code>\$Operator</code>

\$OperatorDomainEntered

Категория	контроль доступа
Использование	При каждом изменении значения тега \$PasswordEntered будет выполняться внутренняя процедура регистрации без открытия каких-либо графических окон. В этой процедуре будут использованы значения всех тегов с названиями типа \$*Entered в качестве имени пользователя и значение тега \$OperatorDomainEntered в качестве названия домена (используется, если только режим контроля доступа – средствами операционной системы. Если режим контроля доступа иной, то этот тег игнорируется).
Тип данных	Символьный
Примеры	<code>\$OperatorEntered = "john";</code> <code>\$OperatorDomainEntered = "Corporate_hq";</code> <code>\$PasswordEntered = "password";</code>
См. также	<code>\$Operator</code>

\$OperatorEntered

	Используется для ввода допустимого имени пользователя.
Категория	контроль доступа
Использование	\$OperatorEntered
Примечание	Создает окно входа пользователя в систему. С этим тегом можно связывать сенсорные объекты ввода и/или скрипты, используемые для ввода имени пользователя. Примечание. Если тег \$OperatorEntered имеет допустимое значение, теги \$AccessLevel и \$Operator получают соответствующие предопределённые значения.
Тип данных	Сообщение (чтение/запись)
См. также	\$AccessLevel, \$Operator, \$PasswordEntered, \$ChangePassword, \$ConfigureUsers

\$OperatorName

Категория	контроль доступа
Использование	Данный тег будет содержать полное имя оператора, если проверка полномочий осуществляется средствами операционной системы или ArcestrA и в системе присутствует зарегистрированный пользователь, ещё не завершивший сеанс работы. В противном случае в теге будет содержаться имя зарегистрированного пользователя (такое же, как и в теге \$Operator).
Тип данных	Символьный
Примеры	\$Operator = "john"; \$OperatorName = "John Smith";
См. также	\$Operator

\$PasswordEntered

	Используется для ввода допустимого пароля
Категория	контроль доступа
Использование	\$PasswordEntered
Примечание	Значение тега всегда считывается как пустая строка. Поле отображения значения тега \$PasswordEntered всегда пустое. Поскольку значение тега всегда возвращается как пустая строка, связанный с ним скрипт по изменению данных никогда будет запускаться. Тег может использоваться для создания пользовательского окна входа в систему. Для установки пароля пользователя с этим тегом могут быть связаны сенсорные объекты ввода и скрипты QuickScripts. Примечание. Если тег \$PasswordEntered имеет допустимое значение, теги \$AccessLevel и \$Operator получают соответствующие предопределённые значения.
Тип данных	Сообщение (только запись)
См. также	\$AccessLevel, \$Operator, \$OperatorEntered, \$ChangePassword, \$ConfigureUsers

\$Second

	Содержит значение секунд текущего времени
Категория	система
Использование	\$Second
Примечание	Имеет значения от 0 до 59.
Тип данных	Целый (только чтение)
Примеры	Wave = 100*Sin(6*\$Second);
См. также	\$DateTime, \$Minute, \$Msec, \$Hour, \$Time

\$StartDdeConversations

	Начинает неинициализированный диалог DDE во время выполнения программы, когда меню Special (Специальные) отключено. Равносильно команде Start Uninitiated DDE Conversations (Начать DDE-обмен) меню Special приложения WindowViewer.
Категория	система
Использование	\$StartDdeConversations
Примечание	Установка тега в 1 начинает неинициализированный диалог DDE.
Тип данных	Логический (чтение/запись)

\$System

	Группа алармов по умолчанию
Категория	система
Использование	\$System
Примечание	Если тег не связан с конкретной группой алармов, то ему автоматически сопоставляется данная корневая группа по умолчанию. Все определённые пользователем группы алармов являются подгруппами корневой группы \$System.
Тип данных	System Alarm Group (только чтение)
Примеры	\$System.Ack = 1; {подтверждение всех алармов}

\$Time

	Содержит время в миллисекундах от полуночи
Категория	система
Использование	\$Time
Тип данных	Целый (только чтение)
Примеры	Sec_Midnight = \$Time/1000; {число секунд после полуночи}
См. также	\$DateTime, \$Minute, \$Msec, \$Second, \$Hour, \$TimeString

\$TimeString

	Содержит время в формате, указанном в файле WIN.INI.
Категория	система
Использование	\$TimeString
Примечание	Формат времени устанавливается в Панели управления Windows или путем двойного щелчка кнопкой мыши на показаниях времени в правом углу панели задач.
Тип данных	Символьный (только чтение)
Примеры	BatchStartString = \$TimeString;
См. также	\$DateTime, \$Minute, \$Msec, \$Second, \$Hour, \$Time

\$VerifiedUserName

	Содержит полное имя прошедшего проверку пользователя, если выполнение функции InvisibleVerifyCredentials() завершилось успешно и если контроль доступа осуществляется средствами операционной системы или Archestra. При неуспешном завершении указанной функции данный системный тег будет содержать пустую строку.
Категория	контроль доступа
Использование	\$VerifiedUserName
Примечание	При каждом изменении значения данного тега (т.е. при каждом выполнении функции InvisibleVerifyCredentials()) будет сгенерировано событие, а в столбце "Value" при успешном завершении функции будет содержаться полное имя прошедшего проверку пользователя. При неуспешном завершении функции в этом столбце будет находиться пустая строка. В столбце "Name" будет записано значение тега \$VerifiedUserName.
Тип данных	Символьный
Допустимые значения	Полное имя пользователя
Примеры	<p>Tag = InvisibleVerifyCredentials("john", "password", "Plant_Floor");</p> <p>Если этот вызов функции завершится успешно, то тег \$VerifiedUserName получит значение "John Smith", при этом будет сгенерировано событие Operator Event. В столбце названия будет отображено значение тега \$VerifiedUserName, а в столбце Value будет строка "John Smith". Если функция завершится с ошибкой, то в тег \$VerifiedUserName и столбец Value будет записана пустая строка "". При каждом вызове указанной скриптовой функции в тег \$VerifiedUserName будет записано либо полное имя соответствующего пользователя, либо пустая строка.</p>
См. также	InvisibleVerifyCredentials(); \$OperatorName, \$Operator

\$Year

	Содержит текущий год в четырёхзначном формате.
Категория	система
Использование	\$Year
Примечание	Содержит текущий год в следующем формате: 2004
Тип данных	Целый (только чтение)
Примеры	<p>CurrentYear = \$Year;</p> <p>NoYrsTill2010 = 2010 - CurrentYear;</p>
См. также	\$Day, \$Month

Поля "через точку" и свойства

Поля тегов в системе InTouch используются для доступа к различным свойствам таких объектов, как теги, архивные тренды или объекты управления окнами. Поля тегов как раз и предназначены для того, чтобы можно было посмотреть и изменить эти свойства. В зависимости от типа объекта доступ к этим свойствам может осуществляться одним из двух способов: непосредственно с помощью синтаксиса *Название_тега.поле* либо косвенно с помощью скриптов.

Все объекты, за исключением объектов управления окнами и распределённых алармов, используют синтаксис *Название_тега.поле*. Чтобы использовать то или иное свойство в скрипте либо выражении анимационной функции, надо просто указать название объекта и за ним – название свойства (после точки). Например, чтобы разрешить изменение второго предельного значения HiHi тега Analog_Tag во время исполнения программы, следует создать анимационную связь аналогового ввода типа **Analog – User Input Touch** с экранной кнопкой и указать **Analog_Tag.HiHiLimit** в качестве выражения. Во время работы системы оператор просто нажмёт на эту кнопку и введёт новое значение для предельного значения HiHi тега Analog_Tag.

Для обращения к свойствам объектов управления окнами и распределённым алармам используются скриптовые функции *GetPropertyX()* и *SetPropertyX()*, где X – тип данных этого свойства (D = логический, I = целый, M = сообщение). Эти функции также можно использовать в скриптах и анимационных связях для доступа к требуемым свойствам.

Дополнительные сведения о скриптовых функциях *GetProperty()* и *SetProperty()* можно найти в главе 3 данного руководства.

Типы тегов

При определении тега в Словаре приложения InTouch ему необходимо присвоить тип в соответствии с его назначением. Если тег будет использоваться для получения или передачи значений из или в другое приложение Windows типа сервера ввода/вывода, то это должен быть тег типа ввода/вывода. Ниже описаны все типы тегов InTouch и их применение.

Внутренние теги (типа Memory)

Внутренние теги существуют внутри приложения InTouch. Их используют для хранения различных констант и промежуточных результатов вычислений. С их помощью можно также создавать вычисляемые переменные, значения которых могут передаваться в другие программы Windows. Например, внутренние теги можно использовать для хранения

различных констант (типа 3,14159) либо рецептов (в группах внутренних тегов). В расчётах внутренние теги можно использовать для управления фоновым исполнением Quick-скрипта. В частности, можно определить внутренний тег COUNT, значение которого будет изменяться в скрипте для создания различных анимационных эффектов для текущей стадии технологического процесса. Существует четыре типа внутренних тегов:

Логический (Memory Discrete)

Внутренние логические теги могут иметь значения либо 0 (ложно, выключено), либо 1 (истинно, включено).

Целый (Memory Integer)

32-разрядная целая величина со знаком, которая может иметь значения от –2147483648 до 2147483647.

Вещественный (Memory Real)

Теги с плавающей десятичной точкой. Допустимые значения: от –3.4e38 до 3.4e38. Все вычисления с плавающей точкой производятся с 64-разрядным разрешением, однако результат округляется до 32-разрядной величины.

Сообщение (Memory Message)

Теги, значения которых представляют собой строки символов длиной до 131 символа.

Теги ввода/вывода

Все теги, используемые для считывания или записи значений из или в другое приложение Windows, относятся к типу тегов ввода/вывода. К этому типу относятся все данные, считываемые из или передаваемые в программируемые контроллеры, компьютеры управления процессами, а также сетевые узлы. Доступ к тегам ввода/вывода осуществляется при помощи коммуникационных протоколов Microsoft Dynamic Data Exchange (DDE) и Wonderware SuiteLink.

При изменении значения тега ввода/вывода для чтения/записи оно немедленно записывается в удалённое приложение. Значение тега можно также обновить из удалённого приложения путем изменения в нём элемента, связанного с данным тегом. По умолчанию все теги ввода/вывода определяются как теги для чтения/записи, хотя их можно сделать тегами только для чтения, установив опцию **Read Only** в диалоговом окне Словаря приложения Tagname. Существует четыре типа тегов ввода/вывода:

Логический (I/O Discrete)

Логические теги ввода/вывода могут иметь значения либо 0 (ложно, выключено), либо 1 (истинно, включено).

Целый (I/O Integer)

32-разрядная целая величина со знаком, которая может иметь значения от –2147483648 до 2147483647.

Вещественный (I/O Real)

Теги с плавающей десятичной точкой. Допустимые значения: от –3.4e38 до 3.4e38. Все вычисления с плавающей точкой производятся с 64-разрядным разрешением, однако результат округляется до 32-разрядной величины.

Сообщение (I/O Message)

Теги, значения которых представляют собой строки символов длиной до 131 символа.

Косвенные теги (логические, аналоговые, сообщения)

Косвенные типы тегов позволяют создавать единственное окно и связывать теги этого окна с несколькими различными источниками. В частности, можно написать Quick-скрипт по изменению данных, который будет менять источник всех тегов в окне в момент обновления соответствующей величины.

Прочие типы тегов

Существует несколько специальных типов тегов для выполнения сложных действий типа создания динамических окон отображения алармов, архивных трендов, контроля или управления тегами каждого пера архивного тренда. Существуют также косвенные типы тегов, которые можно использовать для переназначения источников значений тегов. Специальные типы тегов описаны далее.

Архивный тренд (Hist Trend)

Для создания архивных трендов в системе InTouch используются теги типа Hist Trend. Для тегов этого типа допустимы все поля, связанные с архивными трендами.

Идентификатор тега (Tag ID)

Это специальный тип тегов, предназначенный для работы с архивными трендами. Теги типа Tag ID используются для поиска информации о тегах, значения которых выводятся на архивный тренд. В большинстве случаев теги типа Tag ID используются для отображения названия тега, связанного с указанным пером или для выбора другого тега для данного пера тренда.

Супертеги (SuperTags)

Супертеги InTouch предназначены для создания тегов составного типа. Возможно определение супертегов, состоящих не более чем из 64 тегов-членов и имеющих 2 уровней вложения. Теги-члены используются в приложении точно так же, как обычные теги. Они поддерживают тренды, алармы и все поля тегов.

Более подробную информацию по супертегам можно найти в *Руководстве пользователя InTouch*.

Таблица соответствия типов тегов и полей

В следующей таблице показано соответствие между типами тегов и полями тегов. Указанные примечания относятся к этой таблице.

Если обращение к внутренним тегам осуществляется из *клиентской программы* как к точкам ввода/вывода, то приложение View (*сервер*) предоставляет их вместе с отметкой времени, соответствующей местному времени передачи этих значений клиенту. Следует учесть, что в локальном узле внутренние теги всегда являются достоверными (качества GOOD) и не имеют отметок времени. Это означает, что соответствующая

отметка времени в результате изменения внутренних тегов в скриптах или анимационных функциях НЕ изменяется.

* Поддерживается только в системе InTouch версии 7.0 и старше.

** Поддерживается только в системе InTouch версии 7.1 и старше.

*** Поддерживается только в системе InTouch версии 7.11 и старше.

Типа тега	Внутренний			Ввода/вывода			Косвенный			Прочий		
	Логический	Цель	Вещественный	Логический	Цель	Вещественный	Логический	Аналоговый	Группа алармов	Архивный тренд	Распределённый аларм	Tag ID
Поле тега												
.Ack	X	X	X	X	X	X	X	X	X			
.AckDev***		X	X		X	X		X	X			
.AckDsc	X			X			X		X			
.AckROC***		X	X		X	X		X	X			
.AckValue***		X	X		X	X		X	X			
.Alarm	X	X	X	X	X	X	X	X	X			
.AlarmAccess **											X	
.AlarmAckModel***	X	X	X	X	X	X	X	X	X			
.AlarmClass **											X	
.AlarmComment ** Ver. 1	X	X	X	X	X	X	X	X	X			X
.AlarmComment ** Ver. 2											X	
.AlarmDate**											X	
.AlarmDev		X	X		X	X		X	X			
.AlarmDevCount		X	X		X	X		X	X			
.AlarmDevDeadband		X	X		X	X		X				
.AlarmDevUnAckCount		X	X		X	X		X	X			
.AlarmDisabled***	X	X	X	X	X	X	X	X	X			
.AlarmDsc	X			X			X		X			
.AlarmDscCount	X			X			X		X			
.AlarmDscDisabled	X			X			X		X			
.AlarmDscEnabled	X			X			X		X			
.AlarmDscInhibitor	X			X			X		X			
.AlarmDscUnAckCount	X			X			X		X			
.AlarmEnabled	X	X	X	X	X	X	X	X	X			

Типа тега	Внутренний			Ввода/ вывода			Косвенный			Прочий		
	Логический	Целый	Вещественный	Логический	Целый	Вещественный	Логический	Аналоговый	Группа алармов	Архивный тренд	Распределённый аларм	Tag ID
.AlarmGroup**											X	
.AlarmGroupSel **											X	
.AlarmHiDisabled***		X	X		X	X		X	X			
.AlarmHiEnabled***		X	X		X	X		X	X			
.AlarmHiHiDisabled***		X	X		X	X		X	X			
.AlarmHiHiEnabled***		X	X		X	X		X	X			
.AlarmHiHiInhibitor***		X	X		X	X		X	X			
.AlarmHiInhibitor***		X	X		X	X		X	X			
.AlarmLimit **											X	
.AlarmLoDisabled***		X	X		X	X		X	X			
.AlarmLoEnabled***		X	X		X	X		X	X			
.AlarmLoInhibitor***		X	X		X	X		X	X			
.AlarmLoLoDisabled***		X	X		X	X		X	X			
.AlarmLoLoEnabled***		X	X		X	X		X	X			
.AlarmLoLoInhibitor***		X	X		X	X		X	X			
.AlarmMajDevDisabled***		X	X		X	X		X	X			
.AlarmMajDevEnabled***		X	X		X	X		X	X			
.AlarmMajDevInhibitor***		X	X		X	X		X	X			
.AlarmMinDevDisabled***		X	X		X	X		X	X			
.AlarmMinDevEnabled***		X	X		X	X		X	X			
.AlarmMinDevInhibitor***		X	X		X	X		X	X			
.AlarmName **											X	
.AlarmOprName **											X	
.AlarmOprNode **											X	
.AlarmPri **											X	
.AlarmProv **											X	
.AlarmROC		X	X		X	X		X	X			
.AlarmROCCount		X	X		X	X		X	X			
.AlarmROCDisabled***		X	X		X	X		X	X			
.AlarmROCEnabled***		X	X		X	X		X	X			

Типа тега	Внутренний			Ввода/ вывода			Косвенный			Прочий		
	Логический	Целый	Вещественный	Логический	Целый	Вещественный	Логический	Аналоговый	Группа алармов	Архивный тренд	Распределённый аларм	Tag ID
.AlarmROCIInhibitor***		X	X		X	X		X	X			
.AlarmROCUAckCount		X	X		X	X		X	X			
.AlarmState **											X	
.AlarmTime **											X	
.AlarmTotalCount	X	X	X	X	X	X	X	X	X			
.AlarmType **											X	
.AlarmUnAckCount	X	X	X	X	X	X	X	X	X			
.AlarmUserDefNum1***	X	X	X	X	X	X	X	X	X			
.AlarmUserDefNum2***	X	X	X	X	X	X	X	X	X			
.AlarmUserDefStr***	X	X	X	X	X	X	X	X	X			
.AlarmValDeadband		X	X		X	X		X				
.AlarmValue **											X	
.AlarmValueCount		X	X		X	X		X	X			
.AlarmValueUnAckCount		X	X		X	X		X	X			
.Caption											X	
.ChartLength										X		
.ChartStart										X		
.Comment	X	X	X	X	X	X	X	X	X			
.DevTarget		X	X		X	X		X				
.DisplayMode										X		
.Enabled	X	X	X	X	X	X	X	X				
.EngUnits *		X	X		X	X		X				
.Freeze											X	
.HiHiLimit		X	X		X	X		X				
.HiHiSet **		X	X		X	X		X				
.HiHiStatus		X	X		X	X		X				
.HiLimit		X	X		X	X		X				
.HiSet **		X	X		X	X		X				
.HiStatus		X	X		X	X		X				
.ListChanged											X	

Типа тега	Внутренний			Ввода/ вывода			Косвенный			Прочий		
	Логический	Целый	Вещественный	Логический	Целый	Вещественный	Логический	Аналоговый	Группа алармов	Архивный тренд	Распределённый аларм	Tag ID
.ListCount											X	
.ListIndex											X	
.LoLimit		X	X		X	X		X				
.LoLoLimit **		X	X		X	X		X				
.LoLoSet		X	X		X	X		X				
.LoLoStatus		X	X		X	X		X				
.LoSet **		X	X		X	X		X				
.LoStatus		X	X		X	X		X				
.MajorDevPct		X	X		X	X		X				
.MajorDevSet **		X	X		X	X		X				
.MajorDevStatus		X	X		X	X		X				
.MaxEU		X	X		X	X		X				
.MaxRange										X		
.MaxRaw *					X	X		X				
.MinEU		X	X		X	X		X				
.MinorDevPct		X	X		X	X		X				
.MinorDevSet **		X	X		X	X		X				
.MinorDevStatus		X	X		X	X		X				
.MinRange										X		
.MinRaw *					X	X		X				
.Name	X	X	X	X	X	X	X	X	X	X	X	X
.NewIndex					X						X	
.NextPage											X	
.Normal	X	X	X	X	X	X	X	X	X			
.NumAlarms											X	
.OffMsg *	X			X			X					
.OnMsg *	X			X			X					
.PageNum											X	
.Pen1 – .Pen8										X		
.PendingUpdates											X	

Типа тега	Внутренний			Ввода/вывода			Косвенный			Прочий		
	Логический	Целый	Вещественный	Логический	Целый	Вещественный	Логический	Аналоговый	Группа алармов	Архивный тренд	Распределённый аларм	Tag ID
.PrevPage											X	
.PriFrom											X	
.PriTo											X	
.ProviderReq											X	
.ProviderRet											X	
.Quality	X	X	X	X	X	X	X	X				
.QualityLimit *	X	X	X	X	X	X	X	X				
.QualityLimitstring *	X	X	X	X	X	X	X	X				
.QualityStatus *	X	X	X	X	X	X	X	X				
.QualityStatusString *	X	X	X	X	X	X	X	X				
.QualitySubstatus *	X	X	X	X	X	X	X	X				
.QualitySubstatusString *	X	X	X	X	X	X	X	X				
.QueryState											X	
.QueryType											X	
.RawValue *					X	X		X				
.ReadOnly											X	
.Reference				X	X	X	X	X				
.ReferenceComplete				X	X	X	X	X				
.ROCPct		X	X		X	X		X				
.ROCSets **		X	X		X	X		X				
.ROCStatus		X	X		X	X		X				
.ScooterLockLeft										X		
.ScooterLockRight										X		
.ScooterPosLeft										X		
.ScooterPosRight										X		
.Successful											X	
.SuppressRetain											X	
.TagID										X		
.TimeDate *	X	X	X	X	X	X	X	X				
.TimeDateString	X	X	X	X	X	X	X	X				

Типа тега	Внутренний			Ввода/ вывода			Косвенный			Прочий		
	Логический	Цельный	Вещественный	Логический	Цельный	Вещественный	Логический	Аналоговый	Группа алармов	Архивный тренд	Распределённый аларм	Tag ID
.TimeDateTime	X	X	X	X	X	X	X	X				
.TimeDay *	X	X	X	X	X	X	X	X				
.TimeHour *	X	X	X	X	X	X	X	X				
.TimeMinute *	X	X	X	X	X	X	X	X				
.TimeMonth *	X	X	X	X	X	X	X	X				
.TimeMsec *	X	X	X	X	X	X	X	X				
.TimeSecond *	X	X	X	X	X	X	X	X				
.TimeTime *	X	X	X	X	X	X	X	X				
.TimeTimeString	X	X	X	X	X	X	X	X				
.TimeYear *	X	X	X	X	X	X	X	X				
.TopIndex											X	
.TotalPages											X	
.UnAck	X	X	X	X	X	X	X	X	X			
.UpdateCount										X		
.UpdateInProgress										X		
.UpdateTrend										X		
.Value(Tagname)	X	X	X	X	X	X	X	X	X			X
.Value(Windows Control)											X	
.Visible											X	

.Ack

	Управляет состоянием подтверждения локальных алармов всех типов	
Категория	алармы	
Использование	Tagname.Ack = 1	
	Параметр	Описание
	Tagname	Название любого тега логического, целого или вещественного типа, косвенного логического тега либо группы алармов.
Примечание	<p>Установите это поле в 1, чтобы подтвердить все неподтверждённые алармы, которые могут быть связаны с указанной группой алармов или тегом. Если указанное название является названием группы алармов, все неподтверждённые алармы, связанные с тегами указанной группы, будут подтверждены. Если же указан тег любого другого типа, то подтверждены будут только неподтверждённые алармы, связанные с этим тегом. Установка поля в значение, отличное от 1, не имеет смысла, и результат будет неопределённым.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	1	
Примеры	<p>Следующий оператор подтверждает логический аларм, связанный с тегом Tag1.</p> <p>Tag1.Ack = 1;</p> <p>В следующем примере подтверждаются все неподтверждённые алармы тегов в группе алармов PumpStation:</p> <p>PumpStation.Ack = 1;</p> <p>Примечание. Для поля .Ack существует поле противоположного назначения .UnAck. При наличии неподтверждённого аларма поле .UnAck устанавливается в 1. Поле .UnAck может использоваться в анимационных функциях или условных скриптах для выдачи уведомлений о существовании неподтверждённых алармов.</p>	
См. также	.Alarm, .UnAck, .AckDev, AckROC, .AckDSC, .AckValue, .AlarmAckModel	

.AckDev

	Управляет состоянием подтверждения локальных алармов по отклонению	
Категория	алармы	
Использование	Tagname.AckDev = 1	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, косвенного логического тега либо группы алармов.
Примечание	Установите это поле в 1, чтобы подтвердить все неподтверждённые алармы по отклонению, которые могут быть связаны с указанной группой алармов или тегом. Если указанное название является названием группы алармов, все неподтверждённые алармы по отклонению, связанные с тегами указанной группы, будут подтверждены. Если же указан тег любого другого типа, то подтверждены будут только неподтверждённые алармы по отклонению, связанные с этим тегом. Установка поля в значение, отличное от 1, не имеет смысла, и результат будет неопределённым.	
Тип данных	Логический (чтение/запись)	
Допустимые значения	1	
Примеры	Следующий оператор подтверждает аларм по отклонению, связанный с тегом Tag1 . Tag1.AckDev = 1; В следующем примере подтверждаются все неподтверждённые алармы по отклонению в группе алармов PumpStation : PumpStation.AckDev = 1;	
См. также	.Alarm, .Ack, .UnAck, .AckDSC, .AckROC, .AckValue, .AlarmAckModel	

.AckDsc

	Управляет состоянием подтверждения локальных логических алармов	
Категория	алармы	
Использование	Tagname.AckDsc = 1	
	Параметр	Описание
	Tagname	Название любого косвенного логического тега либо группы алармов.
Примечание	<p>Установите это поле в 1, чтобы подтвердить все неподтверждённые логические алармы, которые могут быть связаны с указанной группой алармов или тегом. Если указанное название является названием группы алармов, все неподтверждённые логические алармы, связанные с тегами указанной группы, будут подтверждены. Если же указан тег любого другого типа, то подтверждены будут только неподтверждённые логические алармы, связанные с этим тегом. Установка поля в значение, отличное от 1, не имеет смысла, и результат будет неопределённым.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	1	
Примеры	<p>Следующий оператор подтверждает логический аларм, связанный с тегом Tag1.</p> <p>Tag1. AckDsc = 1;</p> <p>В следующем примере подтверждаются все неподтверждённые логические алармы в группе алармов PumpStation:</p> <p>PumpStation.AckDsc = 1;</p>	
См. также	.Alarm, .Ack, .UnAck, .AckDev, .AckDSC, .AckROC, .AckValue, .AlarmAckModel	

.AckROC

	Управляет состоянием подтверждения локальных алармов по скорости изменения	
Категория	алармы	
Использование	Tagname.AckROC = 1	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, косвенного логического тега либо группы алармов.
Примечание	Установите это поле в 1, чтобы подтвердить все неподтверждённые алармы по скорости изменения, которые могут быть связаны с указанной группой алармов или тегом. Если указанное название является названием группы алармов, все неподтверждённые алармы по скорости изменения, связанные с тегами указанной группы, будут подтверждены. Если же указан тег любого другого типа, то подтверждены будут только неподтверждённые алармы по скорости изменения, связанные с этим тегом. Установка поля в значение, отличное от 1, не имеет смысла, и результат будет неопределённым.	
Тип данных	Логический (чтение/запись)	
Допустимые значения	1	
Примеры	Следующий оператор подтверждает аларм по скорости изменения, связанный с тегом Tag1 . Tag1.Ack = 1; В следующем примере подтверждаются все неподтверждённые алармы по скорости изменения в группе алармов PumpStation : PumpStation.Ack = 1;	
См. также	.Alarm, .Ack, .UnAck, .AckDev, .AckDSC, .AckValue, .AlarmAckModel	

.AckValue

	Управляет состоянием подтверждения локальных алармов по изменению значения	
Категория	алармы	
Использование	Tagname.AckValue = 1	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, косвенного логического тега либо группы алармов.
Примечание	<p>Установите это поле в 1, чтобы подтвердить все неподтверждённые алармы по изменению значения, которые могут быть связаны с указанной группой алармов или тегом. Если указанное название является названием группы алармов, все неподтверждённые алармы по изменению значения, связанные с тегами указанной группы, будут подтверждены. Если же указан тег любого другого типа, то подтверждены будут только неподтверждённые алармы по изменению значения, связанные с этим тегом. Установка поля в значение, отличное от 1, не имеет смысла, и результат будет неопределённым.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	1	
Примеры	<p>Следующий оператор подтверждает аларм по изменению значения, связанный с тегом "Tag1".</p> <p>Tag1.Ack = 1;</p> <p>В следующем примере подтверждаются все алармы по изменению значения в группе алармов PumpStation:</p> <p>PumpStation.Ack = 1;</p> <p>Для подтверждения алармов по изменению значений можно воспользоваться косвенной переменной AlarmGroup (с помощью групповой переменной). Например:</p> <p>StationAlarms.Name = "PumpStation";</p> <p>где StationAlarms определяется как название группы алармов и затем связывается с PumpStation. Следующее выражение похоже на предыдущее, однако используется для подтверждения всех неподтвержденных алармов по изменению значений из группы алармов PumpStation, которая в текущий момент времени связана с группой алармов с названием StationAlarms.</p> <p>StationAlarms.Ack = 1;</p>	
См. также	.Alarm, .AlarmValue .Ack, .UnAck, .AckDev, .AckDSC, .AckROC, .AlarmAckModel	

.Alarm

	Имеет значение 1, если состояние аларма для указанного тега или группы алармов существует.	
Категория	алармы	
Использование	TagName.Alarm	
	Параметр	Описание
	TagName	Название любого тега логического, целого или вещественного типа, косвенного логического тега либо группы алармов
Примечание	Обычно значение этого поля "только для чтения" равно 0. При возникновении состояния аларма указанного тега система устанавливает его в 1. Это значение сохраняется всё время, пока существует хотя бы одно из состояний аларма. Если название тега представляет собой название группы алармов, то поле устанавливается в 1 при возникновении состояния аларма <u>любого</u> из тегов, входящего в эту группу. Поле с противоположным назначением называется .Normal .	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 или 1	
Примеры	<p>Следующий оператор проверяет, находится ли тег Tag1 в состоянии аларма:</p> <pre>IF (Tag1.Alarm == 1) THEN</pre> <p>Операторы в теле следующего условного оператора будут выполняться, если в состоянии аларма находится хотя бы один тег из группы алармов PumpStation:</p> <pre>IF (PumpStation.Alarm == 1) THEN MyAlarmMessage = "Нештатный режим насосной станции!"; ENDIF;</pre> <p>Данное поле никак не связано с полями .Ack и .UnAck. Это означает, что даже если состояние аларма и будет подтверждено, значение этого поля всё равно будет равно 1 (пока существует состояние аларма).</p>	
См. также	.Ack, .Normal, Ack()	

.AlarmAccess

	Хранит название канала доступа для тега, связанного с выбранным алармом. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	GetPropertyM("ObjectName.AlarmAccess", <i>Tagname</i>) ;	
	Параметр	Описание
	ObjectName	Название объекта вывода распределённых алармов.
	<i>Tagname</i>	Любой внутренний тег типа "сообщение".
Тип данных	Внутренний, сообщение (только для чтения)	
Примеры	<p>Следующий оператор возвращает название тега, связанного с указанным алармом:</p> <p>GetPropertyM("AlmObj_1.AlarmAccess", almAccess);</p> <p>где AlmObj_1 – это название объекта вывода распределённых алармов, а almAccess – название внутреннего тега типа "сообщение" (Memory Message), в который будет записано название канала доступа тега, связанного с указанным алармом.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almAccess, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almAccess будет содержать название канала доступа для тега, связанного с выделенным алармом.</p>	
См. также	GetPropertyM(), .AlarmClass, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmTime, .AlarmType	

.AlarmAckModel

	Выбор модели подтверждения алармов, связанной с тегом: 0 – условия (значение по умолчанию), 1 – события, 2 – расширенная сводка	
Категория	алармы	
Использование	Tagname.AlarmAckModel	
	Параметр	Описание
	Tagname	Название любого тега логического, целого или вещественного типа, косвенного логического тега либо группы алармов
Примечание	Данное поле по умолчанию имеет значение 0 (модель подтверждения условий).	
Тип данных	Аналоговый (только для чтения)	
Допустимые значения	0, 1 или 2	
Примеры	Операторы в теле следующего условного оператора будут выполняться, если PumpStation – аларм типа "event": IF (PumpStation.AlarmAckModel == 1) THEN MyAlarmMessage = "PumpStation is an Event alarm"; ENDIF;	
См. также	.Alarm, .Ack, .UnAck, .AckDev, .AckDSC, .AckROC	

.AlarmClass

	Хранит класс тега, связанного с выбранным алармом. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmClass", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого внутреннего тега типа "сообщение"
Тип данных	Внутренний, сообщение (только для чтения)	
Примеры	<p>Следующий оператор возвращает класс аларма, в который входит выделенный аларм.</p> <pre>GetPropertyM("AlmObj_1.AlarmClass", almClass);</pre> <p>где AlmObj_1 – название объекта вывода распределённых алармов, а almClass – название внутреннего тега типа "сообщение", в который будет записан класс тегов, связанный с выделенным алармом.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almClass, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almClass будет содержать название класса алармов, к которому принадлежит выделенный аларм.</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmTime, .AlarmType, .AlarmValue	

.AlarmComment

	<p>Текстовое поле для чтения/записи, в котором хранится описание природы аларма (не тега)</p> <p>По умолчанию, в новом приложении в этом поле хранится пустая строка.</p> <p>Следует учитывать, что при преобразовании приложений InTouch предыдущих версий в приложение InTouch версии 7.11 в это поле копируется комментарий к тегу (для обеспечения обратной совместимости).</p>	
Категория	алармы	
Вариант использования 1	TagName.AlarmComment	
	Параметр	Описание
	<i>TagName</i>	Название любого тега или группы алармов InTouch.
Вариант использования 2	[ErrorNumber =]GetPropertyM("ObjectName.AlarmComment", Tagname);	
	Параметр	Описание
	<i>TagName</i>	Название внутреннего тега типа "сообщение".
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
Тип данных	Сообщение (только для чтения)	
Примеры	<p>Вариант использования 1: следующий оператор возвращает в тег mTag1 комментарий к определению аларма для тега Tag1. Тег mTag1 – внутренний типа "сообщение" (Memory Message):</p> <p>mTag1 = Tag1.AlarmComment;</p> <p>Вариант использования 2: следующий оператор возвращает во внутренний тег almComment типа "сообщение" указанный в определении тега комментарий к аларму, выделенному в окне распределённых алармов AlmObj_1:</p> <p>GetPropertyM(AlmObj_1.AlarmComment", almComment);</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmTime, .AlarmType, .AlarmValue	

.AlarmDate

	Возвращает дату для тега, связанного с выделенным алармом. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmDate", Tagname);	
	Параметр	Описание
	Tagname	Название внутреннего тега типа "сообщение".
	ObjectName	Название объекта вывода распределённых алармов, например AlmObj_1.
Тип данных	Внутренний, сообщение (только для чтения)	
Примеры	<p>GetPropertyM("AlmObj_1.AlarmDate", almDate); где AlmObj_1 – название объекта вывода распределённых алармов, а almDate – название внутреннего тега типа "сообщение", в который будет записана дата для тега, связанного с выделенным алармом.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almDate, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almDate будет содержать дату возникновения выделенного аларма.</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmComment, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmTime, .AlarmType, .AlarmValue, .AlarmValue	

.AlarmDev

	Значение поля равно 1, если состояние аларма по отклонению для указанного тега или группы алармов существует.	
Категория	алармы	
Использование	Tagname.AlarmDev	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, косвенного аналогового тега или группы алармов
Примечание	Обычно значение данного поля "только для чтения" равно 0. При возникновении состояния аларма по отклонению для указанного тега система устанавливает поле .AlarmDev в 1. Это значение сохраняется всё время существования аларма по отклонению. Если название тега представляет собой название группы алармов, то поле устанавливается в 1 при возникновении состояния аларма по отклонению для <u>любого</u> из тегов, входящего в эту группу.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 или 1	
Примеры	<p>Следующий оператор проверяет, находится ли тег Tag1 в состоянии аларма по отклонению:</p> <p>IF (Tag1.AlarmDev == 1) THEN</p> <p>Операторы в теле следующего условного оператора будут выполняться, если в состоянии аларма по отклонению находится хотя бы один тег из группы алармов</p> <p>PumpStation:</p> <p>IF (PumpStation.AlarmDev == 1) THEN</p> <p>MyAlarmMessage = "Нештатный режим насосной станции!";</p> <p>ENDIF;</p> <p>Данное поле никак не связано с полями .Ack и .UnAck. Это означает, что даже если состояние аларма по отклонению и будет подтверждено, значение данного поля всё равно будет равно 1 (пока существует состояние аларма по отклонению).</p>	
См. также	.Ack, .UnAck, .Alarm, .AckDev	

.AlarmDevCount

	Хранит общее число активных алармов по отклонению для указанного тега или группы алармов.	
Категория	алармы	
Использование	Tagname.AlarmDevCount	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа или группы алармов.
Примечание	В данном поле хранится число алармов как по значительному, так и по незначительному отклонению, как подтверждённых, так и не подтверждённых. В случае нерасширенной сводки алармов тега в поле всегда будет храниться значение 1. Следует учесть, что для различных групп алармов это значение будет разным.	
Тип данных	Аналоговый (только для чтения)	
Допустимые значения	0 и любое положительное целое число.	
Примеры	ADC = Tag1.AlarmDevCount; где Tag1 – это тег аналогового типа, для которого возможно возникновение алармов по отклонению. ADC тоже является аналоговым тегом, в который возвращается общее число активных алармов (как не подтверждённых, так и подтверждённых) по отклонению, существующих для тега Tag1.	
См. также	.AlarmDSCCount, .AlarmValueCount, .AlarmROCUAckCount, .AlarmTotalCount, .AlarmDSCUnAckCount, .AlarmValueUnAckCount, .AlarmDevUnAckCount, .AlarmROCUAckCount, .AlarmUnAckCount	

.AlarmDevDeadband

	Используется для контроля и изменения величиной зоны нечувствительности (в процентах) значительного и незначительного отклонений.	
Категория	алармы	
Использование	Tagname.AlarmDevDeadband	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа или группы алармов.
Примечание	Установка опции Retentive Parameters (Сохранение параметров) в окне Словаря тегов обеспечивает автоматическое сохранение любых изменений значения этого поля в результате выполнения скриптов или анимационных функций.	
Тип данных	Целый (чтение/запись)	
Допустимые значения	от 0 до 100 (целые числа)	
Примеры	Следующий оператор изменяет размер зоны нечувствительности к отклонениям в 25 %: Tagname.AlarmDevDeadband = 25;	
См. также	.AlarmValDeadband, .AlarmDev	

.AlarmDevUnAckCount

	Хранит общее число активных неподтверждённых алармов по отклонению для данного тега или группы алармов. Учитываются алармы как по значительному, так и по незначительному отклонению.	
Категория	алармы	
Использование	Tagname.AlarmDevUnAckCount	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа или группы алармов.
Тип данных	Аналоговый (только для чтения)	
Допустимые значения	0 или любое положительное число	
Примеры	ADUC = Tag1.AlarmDevUnAckCount; где Tag1 – это тег аналогового типа, для которого возможно возникновение алармов по отклонению. ADUC тоже является аналоговым тегом, в который возвращается общее число неподтверждённых алармов по отклонению, существующих для тега Tag1.	
См. также	.AlarmDevCount, .AlarmDSCCount, .AlarmValueCount, .AlarmROCUAckCount, .AlarmTotalCount, .AlarmDSCUnAckCount, .AlarmValueUnAckCount, .AlarmROCUAckCount, .AlarmUnAckCount	

.AlarmDisabled

	Запрещает и разрешает генерацию алармов для указанных тега или группы алармов	
Категория	алармы	
Использование	Tagname.AlarmDisabled	
	Параметр	Описание
	Tagname	Название любого тега логического, целого или вещественного типа, косвенного логического или аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmDisabled установлено в 1, все события и алармы игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Если указанное название тега представляет собой название группы алармов, то игнорируются все алармы для всех тегов, связанных с этой группой. Поле, имеющее противоположное значение, – .AlarmEnabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = разрешение алармов (значение по умолчанию), 1 = запрет алармов	
Примеры	Следующий оператор разрешает генерацию алармов для тега Tag1 : Tag1.AlarmDisabled = 0;	
См. также	.AlarmEnabled	

.AlarmDsc

	Значение поля равно 1, если состояние логического аларма для указанного тега или группы алармов существует.	
Категория	алармы	
Использование	Tagname.AlarmDsc	
	Параметр	Описание
	Tagname	Название любого тега логического типа, косвенного логического тега или группы алармов
Примечание	Обычно значение данного поля "только для чтения" равно 0. При возникновении состояния логического аларма для указанного тега система устанавливает поле .AlarmDsc в 1. Это значение сохраняется всё время существования логического аларма. Если название тега представляет собой название группы алармов, то поле устанавливается в 1 при возникновении состояния логического аларма для <u>любого</u> из тегов, входящего в эту группу.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 или 1	
Примеры	<p>Следующий оператор проверяет, находится ли тег Tag1 в состоянии логического аларма:</p> <pre>IF (Tag1.AlarmDsc == 1) THEN</pre> <p>Операторы в теле следующего условного оператора будут выполняться, если в состоянии логического аларма находится хотя бы один тег из группы алармов PumpStation:</p> <pre>IF (PumpStation.AlarmDsc == 1) THEN MyAlarmMessage = "Нештатный режим насосной станции!"; ENDIF;</pre> <p>Данное поле никак не связано с полями .Ack и .UnAck. Это означает, что даже если состояние логического аларма и будет подтверждено, значение данного поля всё равно будет равно 1 (пока существует состояние аларма).</p>	
См. также	.Ack, .UnAck, .Alarm, .AlarmDsc, .AckDsc	

.AlarmDscCount

	Хранит общее число активных логических алармов для указанного тега или группы алармов.	
Категория	алармы	
Использование	Tagname.AlarmDscCount	
	Параметр	Описание
	Tagname	Название любого тега логического типа или группы алармов
Примечание	В данном поле хранится число как подтверждённых, так и не подтверждённых алармов. В случае нерасширенной сводки алармов в поле всегда будет храниться значение 1. Следует учесть, что для различных групп алармов это значение будет разным.	
Тип данных	Аналоговый (только для чтения)	
Допустимые значения	0 и любое положительное целое число.	
Примеры	ADC = Tag1.AlarmDscCount; где Tag1 – это тег аналогового типа, для которого возможно возникновение логических алармов. ADC тоже является аналоговым тегом, в который возвращается общее число активных логических алармов (как не подтверждённых, так и подтверждённых), существующих для тега Tag1.	
См. также	.AlarmDevCount, .AlarmDevUnAckCount, .AlarmValueCount, .AlarmROCUAckCount, .AlarmTotalCount, .AlarmDscUnAckCount, .AlarmValueUnAckCount, .AlarmROCUAckCount, .AlarmUnAckCount	

.AlarmDscDisabled

	Запрещает и разрешает генерацию логических алармов для указанного тега	
Категория	алармы	
Использование	Tagname.AlarmDscDisabled	
	Параметр	Описание
	Tagname	Название любого тега логического типа или группы алармов.
Примечание	<p>Когда поле .AlarmDscDisabled установлено в 1, все события и алармы логического типа игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmDscEnabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = разрешение алармов (значение по умолчанию), 1 = запрет алармов	
Примеры	Следующий оператор разрешает генерацию логических алармов для тега Tag1 : Tag1.AlarmDscDisabled = 0;	
См. также	.AlarmDscEnabled	

.AlarmDscEnabled

	Разрешает и запрещает генерацию логических алармов для указанного тега	
Категория	алармы	
Использование	Tagname.AlarmDscEnabled	
	Параметр	Описание
	Tagname	Название любого тега логического типа или группы алармов.
Примечание	<p>Когда поле .AlarmDscEnabled установлено в 0, все события и алармы логического типа игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmDscDisabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = запрет алармов, 1 = разрешение алармов (значение по умолчанию)	
Примеры	Следующий оператор запрещает генерацию логических алармов для тега Tag1 : Tag1.AlarmDscEnabled = 0;	
См. также	.AlarmDscEnabled	

.AlarmDscInhibitor

	Хранит название тега-ингибитора логических алармов (если таковые могут возникать) для данного тега	
Категория	алармы	
Использование	Tagname.DscInhibitor	
	Параметр	Описание
	Tagname	Название любого тега логического типа или группы алармов
Примечание	Устанавливается во время проектирования в WindowMaker и не может изменяться во время исполнения программы	
Тип данных	Сообщение (только для чтения)	
Примеры	<p>Значение поля .AlarmDscInhibitor записывается в поле .Name косвенного тега, меняя тем самым значение косвенного тега. Следующий оператор возвращает название тега-ингибитора логических алармов тега AlarmedTag (SomeIndirectTag является названием некоторого косвенного аналогового тега):</p> <pre>SomeIndirectTag.Name = AlarmedTag.AlarmDscInhibitor;</pre> <p>Подавление алармов определяется значением косвенного тега:</p> <p>SomeIndirectTag = 1;</p> <p>Подавление включено. Логические алармы тега AlarmedTag заблокированы.</p> <p>SomeIndirectTag = 0;</p> <p>Подавление алармов выключено. Генерация логических алармов для тега AlarmedTag разрешена.</p>	

.AlarmDscUnAckCount

	Хранит общее число активных неподтверждённых логических алармов для данного тега или группы алармов.	
Категория	алармы	
Использование	Tagname.AlarmDscUnAckCount	
	Параметр	Описание
	Tagname	Название любого тега логического типа или группы алармов.
Тип данных	Аналоговый (только для чтения)	
Допустимые значения	0 или любое положительное число	
Примеры	ADUC = Tag1.AlarmDevUnAckCount; где Tag1 – это тег аналогового типа, для которого возможно возникновение логических алармов. ADUC тоже является аналоговым тегом, в который возвращается общее число неподтверждённых логических алармов, существующих для тега Tag1.	
См. также	.AlarmDevCount, .AlarmDevUnAckCount, .AlarmDscCount, .AlarmValueCount, .AlarmROCCount, .AlarmTotalCount, .AlarmValueUnAckCount, .AlarmROCUAckCount, .AlarmUnAckCount	

.AlarmEnabled

	Разрешает и запрещает генерацию алармов для указанных тега или группы алармов	
Категория	алармы	
Использование	Tagname.AlarmEnabled	
	Параметр	Описание
	Tagname	Название любого тега логического, целого или вещественного типа, косвенного логического или аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmEnabled установлено в 0, все события и алармы игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Если указанное название тега представляет собой название группы алармов, то разрешение действительно для всех алармов всех тегов, связанных с этой группой. Поле, имеющее противоположное значение, – .AlarmDisabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = запрет алармов, 1 = разрешение алармов (значение по умолчанию)	
Примеры	Следующий оператор запрещает генерацию алармов для тега Tag1 : Tag1.AlarmEnabled = 0;	
См. также	.AlarmDisabled	

.AlarmGroup

	Содержит текущий запрос, с помощью которого из базы данных извлекается информация в объект вывода распределённых алармов	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmGroup", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название тега типа "сообщение", в который будет записано значение свойства при выполнении данной функции.
Примечание	Поле .AlarmGroup содержит запрос на сведения об алармах, которые отображаются в окне объекта вывода распределённых алармов. Этот запрос может представлять собой перечень групп алармов либо прямых ссылок на провайдеров алармов.	
Тип данных	Сообщение (только для чтения)	
Примеры	Следующий оператор возвращает в тег CurrentQuery типа "сообщение" текст последнего запроса на сведения об алармах, показанных в окне объекта вывода распределённых алармов.	
См. также	GetPropertyM() , .AlarmGroupSel , .AlarmName	

.AlarmGroupSel

	Возвращает группу алармов, к которой принадлежит выделенный аларм. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmGroupSel", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого внутреннего тега типа "сообщение".
Примеры	<p>GetPropertyM("AlmObj_1.AlarmGroupSel", almGroup); где AlmObj_1 – это название объекта вывода распределённых алармов, а almGroup – название внутреннего тега типа "сообщение" (Memory Message), в который будет записано название группы алармов для тега, аларм которого был выбран.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almGroup, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almGroup будет содержать название группы алармов, которой принадлежит выделенный аларм.</p>	
См. также	GetPropertyM(), .AlarmGroup, .AlarmName	

.AlarmHiDisabled

	Запрет и разрешение событий и алармов выхода за границу Hi	
Категория	алармы	
Использование	Tagname.AlarmHiDisabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, косвенного аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmHiDisabled установлено в 1, все события и алармы выхода за границу Hi игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmHiEnabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = разрешение алармов (значение по умолчанию), 1 = запрет алармов	
Примеры	Следующий оператор разрешает генерацию алармов выхода за границу Hi для тега Tag1 : Tag1.AlarmHiDisabled = 0;	
См. также	.AlarmHiEnabled, .AlarmDisabled	

.AlarmHiEnabled

	Разрешение и запрет событий и алармов выхода за границу Hi	
Категория	алармы	
Использование	Tagname.AlarmHiEnabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, косвенного аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmHiEnabled установлено в 0, все события и алармы выхода за границу Hi игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmHiDisabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = запрет алармов, 1 = разрешение алармов (значение по умолчанию)	
Примеры	Следующий оператор запрещает генерацию алармов выхода за границу Hi для тега Tag1 : Tag1.AlarmHiEnabled = 0;	
См. также	.AlarmHiDisabled, .AlarmEnabled	

.AlarmHiHiDisabled

	Запрет и разрешение событий и алармов выхода за границу HiHi	
Категория	алармы	
Использование	Tagname.AlarmHiHiDisabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, косвенного аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmHiHiDisabled установлено в 1, все события и алармы выхода за границу HiHi игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmHiHiEnabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = разрешение алармов (значение по умолчанию), 1 = запрет алармов	
Примеры	Следующий оператор разрешает генерацию алармов выхода за границу HiHi для тега Tag1 : Tag1.AlarmHiHiDisabled = 0;	
См. также	.AlarmHiHiEnabled, .AlarmDisabled	

.AlarmHiHiEnabled

	Разрешение и запрет событий и алармов выхода за границу HiHi	
Категория	алармы	
Использование	Tagname.AlarmHiHiEnabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmHiHiEnabled установлено в 0, все события и алармы выхода за границу HiHi игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmHiHiDisabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = запрет алармов, 1 = разрешение алармов (значение по умолчанию)	
Примеры	Следующий оператор запрещает генерацию алармов выхода за границу HiHi для тега Tag1 : Tag1.AlarmHiHiEnabled = 0;	
См. также	.AlarmHiHiDisabled, .AlarmEnabled	

.AlarmHiHiInhibitor

	Возвращает название тега-ингибитора алармов выхода за границу HiHi для данного тега	
Категория	алармы	
Использование	Tagname.HiHiInhibitor	
	Параметр	Описание
	Tagname	Название любого аналогового тега с алармами или группы алармов
Примечание	Устанавливается во время проектирования в WindowMaker и не может изменяться во время исполнения программы	
Тип данных	Сообщение (только для чтения)	
Примеры	<p>Значение поля .AlarmHiHiInhibitor записывается в поле .Name косвенного тега, меняя тем самым значение косвенного тега. Следующий оператор возвращает название тега-ингибитора алармов выхода за границу HiHi значений тега AlarmedTag (SomeIndirectTag является названием некоторого косвенного аналогового тега):</p> <p>SomeIndirectTag.Name = AlarmedTag.AlarmHiHiInhibitor; Подавление алармов определяется путём установки соответствующего значения косвенного тега:</p> <p>SomeIndirectTag = 1; Подавление включено. Алармы выхода за границу HiHi для тега AlarmedTag блокированы.</p> <p>SomeIndirectTag = 0; Подавление алармов выключено. Генерация алармов выхода за границу HiHi для тега AlarmedTag разрешена.</p>	
См. также	.AlarmHiInhibitor, .AlarmLoInhibitor, .AlarmLoLoInhibitor	

.AlarmHiInhibitor

	Возвращает название тега-ингибитора алармов выхода за границу Hi для данного тега	
Категория	алармы	
Использование	Tagname.AlarmHiInhibitor	
	Параметр	Описание
	Tagname	Название любого аналогового тега с алармами или группы алармов
Примечание	Устанавливается во время проектирования в WindowMaker и не может изменяться во время исполнения программы	
Тип данных	Сообщение (только для чтения)	
Примеры	<p>Значение поля .AlarmHiInhibitor записывается в поле .Name косвенного тега, меняя тем самым значение косвенного тега. Следующий оператор возвращает название тега-ингибитора алармов выхода за границу Hi значений тега AlarmedTag (SomeIndirectTag является названием некоторого косвенного аналогового тега): SomeIndirectTag.Name = AlarmedTag.AlarmHiInhibitor; Подавление алармов определяется путём установки соответствующего значения косвенного тега: SomeIndirectTag = 1; Подавление включено. Алармы выхода за границу Hi для тега AlarmedTag блокированы. SomeIndirectTag = 0; Подавление алармов выключено. Генерация алармов выхода за границу Hi для тега AlarmedTag разрешена.</p>	
См. также	.AlarmHiHiInhibitor, .AlarmLoInhibitor, .AlarmLoLoInhibitor	

.AlarmLimit

	Возвращает предельное значение для тега, соответствующего выбранному аларму. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM(<i>ObjectName.AlarmLimit</i> , <i>TagName</i>);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>TagName</i>	Название любого внутреннего тега типа "сообщение".
Примеры	GetPropertyM("AlmObj_1.AlarmLimit", almLimit); где AlmObj_1 – это название объекта вывода распределённых алармов, а almLimit – название внутреннего тега типа "сообщение" (Memory Message), в который будет записано предельное значение, определяющее генерацию выбранного аларма. При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almLimit , которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almLimit будет содержать соответствующее предельное значение для выбранного аларма.	
См. также	GetPropertyM() , .AlarmAccess , .AlarmClass , .AlarmComment , .AlarmDate , .AlarmName , .AlarmOprName , .AlarmOprNode , .AlarmPri , .AlarmProv , .AlarmState , .AlarmTime , .AlarmType , .AlarmValue	

.AlarmLoDisabled

	Запрет и разрешение событий и алармов выхода за границу Lo	
Категория	алармы	
Использование	Tagname.AlarmLoDisabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmLoDisabled установлено в 1, все события и алармы выхода за границу Lo игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmLoEnabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = разрешение алармов (значение по умолчанию), 1 = запрет алармов	
Примеры	Следующий оператор разрешает генерацию алармов выхода за границу Lo для тега Tag1 : Tag1.AlarmLoDisabled = 0;	
См. также	.AlarmDisabled, .AlarmEnabled, .AlarmLoEnabled	

.AlarmLoEnabled

	Разрешение и запрет событий и алармов выхода за границу Lo	
Категория	алармы	
Использование	Tagname.AlarmLoEnabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmLoEnabled установлено в 0, все события и алармы выхода за границу Lo игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmLoDisabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = запрет алармов, 1 = разрешение алармов (значение по умолчанию)	
Примеры	Следующий оператор запрещает генерацию алармов выхода за границу Lo для тега Tag1 : Tag1.AlarmLoEnabled = 0;	
См. также	.AlarmDisabled, .AlarmEnabled, .AlarmLoDisabled	

.AlarmLoInhibitor

	Возвращает название тега-ингибитора алармов выхода за границу Lo для данного тега	
Категория	алармы	
Использование	Tagname.AlarmLoInhibitor	
	Параметр	Описание
	Tagname	Название любого аналогового тега с алармами или группы алармов
Примечание	Устанавливается во время проектирования в WindowMaker и не может изменяться во время исполнения программы	
Тип данных	Сообщение (только для чтения)	
Примеры	<p>Значение поля .AlarmLoInhibitor записывается в поле .Name косвенного тега, меняя тем самым значение косвенного тега. Следующий оператор возвращает название тега-ингибитора алармов выхода за границу Lo значений тега AlarmedTag (SomeIndirectTag является названием некоторого косвенного аналогового тега):</p> <p>SomeIndirectTag.Name = AlarmedTag.AlarmLoInhibitor;</p> <p>Подавление алармов определяется путём установки соответствующего значения косвенного тега:</p> <p>SomeIndirectTag = 1;</p> <p>Подавление включено. Алармы выхода за границу Lo для тега AlarmedTag блокированы.</p> <p>SomeIndirectTag = 0;</p> <p>Подавление алармов выключено. Генерация алармов выхода за границу Lo для тега AlarmedTag разрешена.</p>	
См. также	.AlarmHiInhibitor, .AlarmHiHiInhibitor, .AlarmLoLoInhibitor	

.AlarmLoLoDisabled

	Запрет и разрешение событий и алармов выхода за границу LoLo	
Категория	алармы	
Использование	Tagname.AlarmLoLoDisabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmLoLoDisabled установлено в 1, все события и алармы выхода за границу LoLo игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmLoLoEnabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = разрешение алармов (значение по умолчанию), 1 = запрет алармов	
Примеры	Следующий оператор разрешает генерацию алармов выхода за границу LoLo для тега Tag1 : Tag1.AlarmLoLoDisabled = 0;	
См. также	.AlarmDisabled, .AlarmEnabled, .AlarmLoLoEnabled	

.AlarmLoLoEnabled

	Разрешение и запрет событий и алармов выхода за границу LoLo	
Категория	алармы	
Использование	Tagname.AlarmLoLoEnabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmLoLoEnabled установлено в 0, все события и алармы выхода за границу LoLo игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmLoLoDisabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = запрет алармов, 1 = разрешение алармов (значение по умолчанию)	
Примеры	Следующий оператор запрещает генерацию алармов выхода за границу LoLo для тега Tag1 : Tag1.AlarmLoLoEnabled = 0;	
См. также	.AlarmDisabled, .AlarmEnabled, .AlarmLoLoDisabled	

.AlarmLoLoInhibitor

	Возвращает название тега-ингибитора алармов выхода за границу LoLo для данного тега	
Категория	алармы	
Использование	Tagname.AlarmLoLoInhibitor	
	Параметр	Описание
	Tagname	Название любого аналогового тега с алармами или группы алармов
Примечание	Устанавливается во время проектирования в WindowMaker и не может изменяться во время исполнения программы	
Тип данных	Сообщение (только для чтения)	
Примеры	<p>Значение поля .AlarmLoLoInhibitor записывается в поле .Name косвенного тега, меняя тем самым значение косвенного тега. Следующий оператор возвращает название тега-ингибитора алармов выхода за границу LoLo значений тега AlarmedTag (SomeIndirectTag является названием некоторого косвенного аналогового тега):</p> <p>SomeIndirectTag.Name = AlarmedTag.AlarmLoLoInhibitor; Подавление алармов определяется путём установки соответствующего значения косвенного тега:</p> <p>SomeIndirectTag = 1; Подавление включено. Алармы выхода за границу LoLo для тега AlarmedTag блокированы.</p> <p>SomeIndirectTag = 0; Подавление алармов выключено. Генерация алармов выхода за границу LoLo для тега AlarmedTag разрешена.</p>	
См. также	.AlarmHiInhibitor, .AlarmHiHiInhibitor, .AlarmLoInhibitor	

.AlarmMajDevDisabled

	Запрет и разрешение событий и алармов по значительному отклонению	
Категория	алармы	
Использование	Tagname.AlarmMajDevDisabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmMajDevDisabled установлено в 1, все события и алармы по значительному отклонению игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmMajDevEnabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = разрешение алармов (значение по умолчанию), 1 = запрет алармов	
Примеры	Следующий оператор разрешает генерацию алармов по значительному отклонению для тега Tag1 : Tag1.AlarmMajDevDisabled = 0;	
См. также	.AlarmDisabled, .AlarmEnabled, .AlarmMajDevEnabled	

.AlarmMajDevEnabled

	Разрешение и запрет событий и алармов по значительному отклонению	
Категория	алармы	
Использование	Tagname.AlarmMajDevEnabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmMajDevEnabled установлено в 0, все события и алармы по значительному отклонению игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmMajDevDisabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = запрет алармов, 1 = разрешение алармов (значение по умолчанию)	
Примеры	Следующий оператор запрещает генерацию алармов по значительному отклонению для тега Tag1 : Tag1.AlarmMajDevEnabled = 0;	
См. также	.AlarmDisabled, .AlarmEnabled, .AlarmMajDevDisabled	

.AlarmMajDevInhibitor

	Возвращает название тега-ингибитора алармов по значительному отклонению для данного тега	
Категория	алармы	
Использование	Tagname.AlarmMajDevInhibitor	
	Параметр	Описание
	Tagname	Название любого аналогового тега с алармами или группы алармов
Примечание	Устанавливается во время проектирования в WindowMaker и не может изменяться во время исполнения программы	
Тип данных	Сообщение (только для чтения)	
Примеры	<p>Значение поля .AlarmMajDevInhibitor записывается в поле .Name косвенного тега, меняя тем самым значение косвенного тега. Следующий оператор возвращает название тега-ингибитора алармов по значительному отклонению тега AlarmedTag (SomeIndirectTag является названием некоторого косвенного аналогового тега):</p> <pre>SomeIndirectTag.Name = AlarmedTag.AlarmMajDevInhibitor;</pre> <p>Подавление алармов определяется путём установки соответствующего значения косвенного тега:</p> <p>SomeIndirectTag = 1;</p> <p>Подавление включено. Алармы по значительному отклонению для тега AlarmedTag блокированы.</p> <p>SomeIndirectTag = 0;</p> <p>Подавление алармов выключено. Генерация алармов по значительному отклонению для тега AlarmedTag разрешена.</p>	
См. также	.AlarmMinDevInhibitor	

.AlarmMinDevDisabled

	Запрет и разрешение событий и алармов по незначительному отклонению	
Категория	алармы	
Использование	Tagname.AlarmMinDevDisabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmMinDevDisabled установлено в 1, все события и алармы по незначительному отклонению игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmMinDevEnabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = разрешение алармов (значение по умолчанию), 1 = запрет алармов	
Примеры	Следующий оператор разрешает генерацию алармов по незначительному отклонению для тега Tag1 : Tag1.AlarmMinDevDisabled = 0;	
См. также	.AlarmDisabled, .AlarmEnabled, .AlarmMinDevEnabled	

.AlarmMinDevEnabled

	Разрешение и запрет событий и алармов по незначительному отклонению	
Категория	алармы	
Использование	Tagname.AlarmMinDevEnabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmMinDevEnabled установлено в 0, все события и алармы по незначительному отклонению игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmMinDevDisabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = запрет алармов, 1 = разрешение алармов (значение по умолчанию)	
Примеры	Следующий оператор запрещает генерацию алармов по незначительному отклонению для тега Tag1 : Tag1.AlarmMinDevEnabled = 0;	
См. также	.AlarmDisabled, .AlarmEnabled, .AlarmMinDevDisabled	

.AlarmMinDevInhibitor

	Возвращает название тега-ингибитора алармов по незначительному отклонению для данного тега	
Категория	алармы	
Использование	TagName.AlarmMinDevInhibitor	
	Параметр	Описание
	TagName	Название любого аналогового тега с алармами или группы алармов
Примечание	Устанавливается во время проектирования в WindowMaker и не может изменяться во время исполнения программы	
Тип данных	Сообщение (только для чтения)	
Примеры	<p>Значение поля .AlarmMinDevInhibitor записывается в поле .Name косвенного тега, меняя тем самым значение косвенного тега. Следующий оператор возвращает название тега-ингибитора алармов по значительному отклонению тега AlarmedTag (SomeIndirectTag является названием некоторого косвенного аналогового тега):</p> <p>SomeIndirectTag.Name = AlarmedTag.AlarmMinDevInhibitor;</p> <p>Подавление алармов определяется путём установки соответствующего значения косвенного тега:</p> <p>SomeIndirectTag = 1;</p> <p>Подавление включено. Алармы по незначительному отклонению для тега AlarmedTag заблокированы.</p> <p>SomeIndirectTag = 0;</p> <p>Подавление алармов выключено. Генерация алармов по незначительному отклонению для тега AlarmedTag разрешена.</p>	
См. также	.AlarmMajDevInhibitor	

.AlarmName

	Возвращает название выбранного аларма. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmName", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого внутреннего тега типа "сообщение".
Тип данных	Внутренний, сообщение (только для чтения)	
Примеры	<p>GetPropertyM("AlmObj_1.AlarmName", almName); где AlmObj_1 – это название объекта вывода распределённых алармов, а almName – название внутреннего тега типа "сообщение" (Memory Message), в который будет записано название выбранного аларма.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almName, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almName будет содержать название выбранного аларма.</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmTime, .AlarmType, .AlarmValue	

.AlarmOprName

	Возвращает имя оператора, во время которого возник выбранный аларм. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmOprName", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого внутреннего тега типа "сообщение".
Тип данных	Внутренний, сообщение (только для чтения)	
Примеры	GetPropertyM("AlmObj_1.AlarmOprName", almOprName); где AlmObj_1 – это название объекта вывода распределённых алармов, а almOprName – название внутреннего тега типа "сообщение" (Memory Message), в который будет записано имя оператора. При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almOprName , которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almOprName будет содержать имя оператора, во время работы которого возник выбранный аларм.	
См. также	GetPropertyM() , .AlarmAccess , .AlarmClass , .AlarmComment , .AlarmDate , .AlarmLimit , .AlarmName , .AlarmOprNode , .AlarmPri , .AlarmProv , .AlarmState , .AlarmTime , .AlarmType , .AlarmValue	

.AlarmOprNode

	Возвращает название операторского узла, связанного с выбранным алармом. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmOprNode", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого внутреннего тега типа "сообщение".
Тип данных	Внутренний, сообщение (только для чтения)	
Примеры	<p>GetPropertyM("AlmObj_1.AlarmOprNode", almOprNode); где AlmObj_1 – это название объекта вывода распределённых алармов, а almOprNode – название внутреннего тега типа "сообщение" (Memory Message), в который будет записано название операторского узла.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almOprNode, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almOprNode будет содержать название операторского узла, связанного с выбранным алармом.</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmPri, .AlarmProv, .AlarmState, .AlarmTime, .AlarmType, .AlarmValue	

.AlarmPri

	Возвращает значение приоритета (1–999) выданного аларма. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmPri", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого аналогового тега.
Тип данных	Внутренний, сообщение (только для чтения)	
Допустимые значения	1–999 (целые, только для чтения)	
Примеры	<p>GetPropertyM("AlmObj_1.AlarmPri", almPri); где AlmObj_1 – это название объекта вывода распределённых алармов, а almPri – название целого тега, в который будет записано значение приоритета выбранного аларма.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almPri, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almPri будет содержать значение приоритета выбранного аларма.</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmProv, .AlarmState, .AlarmTime, .AlarmType, .AlarmValue	

.AlarmProv

	Возвращает название провайдера для тега, соответствующего выбранному аларму. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmProv", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого внутреннего тега типа "сообщение".
Тип данных	Внутренний, сообщение (только для чтения)	
Примеры	<p>GetPropertyM("AlmObj_1.AlarmProv", almProv); где AlmObj_1 – это название объекта вывода распределённых алармов, а almProv – название внутреннего тега типа "сообщение" (Memory Message), в который будет записано название провайдера для тега, соответствующего выбранному аларму.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almProv, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almProv будет содержать название провайдера для тега, соответствующего выбранному аларму.</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmState, .AlarmTime, .AlarmType, .AlarmValue	

.AlarmROC

	Значение поля равно 1, если состояние аларма по скорости изменения для указанного тега существует.	
Категория	алармы	
Использование	Tagname.AlarmROC	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, косвенного аналогового тега или группы алармов
Примечание	Обычно значение данного поля "только для чтения" равно 0. При возникновении состояния аларма по скорости изменения для указанного тега система устанавливает поле .AlarmROC в 1. Это значение сохраняется всё время существования аларма по отклонению. Если название тега представляет собой название группы алармов, то поле устанавливается в 1 при возникновении состояния аларма по скорости изменения для <u>любого</u> из тегов, входящего в эту группу.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 или 1	
Примеры	<p>Следующий оператор проверяет, находится ли тег Tag1 в состоянии аларма по скорости изменения:</p> <p>IF (Tag1.AlarmROC == 1) THEN</p> <p>Операторы в теле следующего условного оператора будут выполняться, если в состоянии аларма по скорости изменения находится хотя бы один тег из группы алармов PumpStation:</p> <p>IF (PumpStation.AlarmROC == 1) THEN</p> <p>MyAlarmMessage = "Нештатный режим насосной станции!";</p> <p>ENDIF;</p> <p>Данное поле никак не связано с полями .Ack и .UnAck. Это означает, что даже если состояние аларма по скорости изменения и будет подтверждено, значение данного поля всё равно будет равно 1 (пока существует состояние аларма).</p>	
См. также	.Ack, .AckROC, .Alarm, .AlarmROCEnabled, .AlarmROCDisabled	

.AlarmROCCount

	Хранит общее число активных алармов по скорости изменения для указанного тега или группы алармов. В данном поле хранится число как подтверждённых, так и не подтверждённых алармов. В случае нерасширенной сводки алармов в нём всегда будет храниться значение 1. Следует учесть, что для различных групп алармов это значение будет разным. Данное поле является полем только для чтения.	
Категория	алармы	
Использование	Tagname.AlarmROCCount	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа или группы алармов.
Тип данных	Целый (только для чтения)	
Допустимые значения	0 или любое положительное целое число.	
Примеры	ARC = Tag1.AlarmROCCount; где Tag1 – это тег аналогового типа, для которого возможно возникновение алармов по скорости изменения. ARC тоже является аналоговым тегом, в который возвращается общее число активных алармов (как не подтверждённых, так и подтверждённых) по скорости изменения, существующих для тега Tag1.	
См. также	.AlarmDevCount, .AlarmDevUnAckCount, .AlarmDscCount, .AlarmDscUnAckCount, .AlarmValueCount, .AlarmTotalCount, .AlarmValueUnAckCount, .AlarmROCUAckCount, .AlarmUnAckCount	

.AlarmROCDisabled

	Запрет и разрешение событий и алармов по скорости изменения	
Категория	алармы	
Использование	Tagname.AlarmROCDisabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmROCDisabled установлено в 1, все события и алармы по скорости изменения игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmROCEnabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = разрешение алармов (значение по умолчанию), 1 = запрет алармов	
Примеры	Следующий оператор разрешает генерацию алармов по скорости изменения для тега Tag1 : Tag1.AlarmROCDisabled = 0;	
См. также	.AlarmDisabled, .AlarmEnabled, .AlarmROCEnabled	

.AlarmROCEnabled

	Разрешение и запрет событий и алармов по скорости изменения	
Категория	алармы	
Использование	Tagname.AlarmROCEnabled	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа, косвенного аналогового тега или группы алармов.
Примечание	<p>Когда поле .AlarmROCEnabled установлено в 0, все события и алармы по незначительному отклонению игнорируются. Они не сохраняются в буфере и не записываются на диск. Во избежание потери данных необходимо вновь разрешить события/алармы, как только это станет возможным.</p> <p>Поле, имеющее противоположное значение, – .AlarmROCDisabled.</p>	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = запрет алармов, 1 = разрешение алармов (значение по умолчанию)	
Примеры	Следующий оператор запрещает генерацию алармов по скорости изменения для тега Tag1 : Tag1.AlarmROCEnabled = 0;	
См. также	.AlarmDisabled, .AlarmEnabled, .AlarmROCDisabled	

.AlarmROCIInhibitor

	Возвращает название тега-ингибитора алармов по скорости изменения для данного тега	
Категория	алармы	
Использование	TagName.AlarmROCIInhibitor	
	Параметр	Описание
	TagName	Название любого аналогового тега с алармами или группы алармов
Примечание	Устанавливается во время проектирования в WindowMaker и не может изменяться во время исполнения программы	
Тип данных	Сообщение (только для чтения)	
Примеры	<p>Значение поля .AlarmROCIInhibitor записывается в поле .Name косвенного тега, меняя тем самым значение косвенного тега. Следующий оператор возвращает название тега-ингибитора алармов по скорости изменения значений тега AlarmedTag (SomeIndirectTag является названием некоторого косвенного аналогового тега):</p> <p>SomeIndirectTag.Name = AlarmedTag.AlarmROCIInhibitor;</p> <p>Подавление алармов определяется путём установки соответствующего значения косвенного тега:</p> <p>SomeIndirectTag = 1;</p> <p>Подавление включено. Алармы по скорости изменения значений тега AlarmedTag блокированы.</p> <p>SomeIndirectTag = 0;</p> <p>Подавление алармов выключено. Генерация алармов по скорости изменений значений тега AlarmedTag разрешена.</p>	
См. также	.AlarmMajDevInhibitor	

.AlarmROCUAckCount

	Хранит общее число активных неподтверждённых алармов по скорости изменения для данного тега или группы алармов.	
Категория	алармы	
Использование	Tagname.AlarmROCUAckCount	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа или группы алармов.
Тип данных	Целый (только для чтения)	
Допустимые значения	0 или любое положительное число	
Примеры	ARUC = Tag1.AlarmROCUAckCount; где Tag1 – это тег аналогового типа, для которого возможно возникновение алармов по скорости изменения. ARUC тоже является аналоговым тегом, в который возвращается общее число неподтверждённых алармов по скорости изменения, существующих для тега Tag1.	
См. также	.AlarmDevCount, .AlarmDevUnAckCount, .AlarmDscCount, .AlarmDscUnAckCount, .AlarmValueCount, .AlarmTotalCount, .AlarmValueUnAckCount, .AlarmROCCount, .AlarmUnAckCount	

.AlarmState

	Возвращает состояние тега, соответствующего выбранному аларму. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmState", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого внутреннего тега типа "сообщение".
Тип данных	Внутренний, сообщение (только для чтения)	
Примеры	<p>GetPropertyM("AlmObj_1.AlarmState", almState); где AlmObj_1 – это название объекта вывода распределённых алармов, а almState – название внутреннего тега типа "сообщение" (Memory Message), в который будет записано состояние тега, соответствующего выбранному аларму.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almState, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almState будет содержать состояние тега, соответствующего выбранному аларму.</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmTime, .AlarmType, .AlarmValue	

.AlarmTime

	Возвращает время возникновения выбранного аларма. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmTime", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого внутреннего тега типа "сообщение".
Тип данных	Внутренний, сообщение (только для чтения)	
Примеры	<p>GetPropertyM("AlmObj_1.AlarmTime", almTime); где AlmObj_1 – это название объекта вывода распределённых алармов, а almTime – название внутреннего тега типа "сообщение" (Memory Message), в который будет записано время возникновения выбранного аларма.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almTime, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almTime будет содержать время возникновения выбранного аларма.</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmType, .AlarmValue	

.AlarmTotalCount

	Хранит общее число активных алармов для указанного тега или группы алармов.	
Категория	алармы	
Использование	TagName.AlarmTotalCount	
	Параметр	Описание
	TagName	Название тега любого типа.
Примечание	В данном поле хранится число всех алармов: по изменению значения, по отклонению, по скорости изменения, логические. Учитываются как подтверждённые, так и не подтверждённые алармы.	
Тип данных	Целый (только для чтения)	
Допустимые значения	0 или любое положительное целое число.	
Примеры	ATC = Tag1.AlarmTotalCount; где Tag1 – это тег аналогового типа, для которого возможно возникновение алармов. ATC тоже является аналоговым тегом, в который возвращается общее число всех активных алармов (как не подтверждённых, так и подтверждённых), существующих для тега Tag1.	
См. также	.AlarmDevCount, .AlarmDevUnAckCount, .AlarmDSCCount, .AlarmDSCUnAckCount, .AlarmValueCont, .AlarmUnAckCount, .AlarmValueUnAckCount, .AlarmROCCount, .AlarmROCUACKCount	

.AlarmType

	Возвращает тип тега, соответствующего выбранному аларму. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmType", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого внутреннего тега типа "сообщение".
Тип данных	Внутренний, сообщение (только для чтения)	
Примеры	<p>GetPropertyM("AlmObj_1.AlarmType", almType); где AlmObj_1 – это название объекта вывода распределённых алармов, а almType – название внутреннего тега типа "сообщение" (Memory Message), в который будет записан тип тега, соответствующего выбранному аларму.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almType, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almType будет содержать тип тега, соответствующего выбранному аларму.</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmTime, .AlarmValue	

.AlarmUnAckCount

	Хранит общее число активных неподтверждённых для данного тега или группы алармов.	
Категория	алармы	
Использование	Tagname.AlarmUnAckCount	
	Параметр	Описание
	Tagname	Название тега любого типа.
Примечание	В данном поле хранится число всех алармов: по изменению значения, по отклонению, по скорости изменения и логические.	
Тип данных	Целый (только для чтения)	
Допустимые значения	0 или любое положительное число	
Примеры	AUC = Tag1.AlarmUnAckCount; где Tag1 – это тег аналогового или логического типа, для которого возможно возникновение алармов. AUC тоже является аналоговым тегом, в который возвращается общее число неподтверждённых алармов, существующих для тега Tag1.	
См. также	.AlarmDevCount, .AlarmDevUnAckCount, .AlarmDscCount, .AlarmDscUnAckCount, .AlarmValueCount, .AlarmTotalCount, .AlarmValueUnAckCount, .AlarmROCCount, .AlarmUnAckCount	

.AlarmUserDefNum1

	<p>Поле .AlarmUserDefNum1 позволяет заносить в запись об аларме дополнительные сведения, которые сохраняются в базе данных об алармах программой-регистратором Alarm DB Logger. (Это поле соответствует столбцу User1 таблицы базы данных.) "Определяемые пользователем" столбцы можно указывать в операторе SELECT для уточнения выборки сведений об алармах из базы данных. Например, если в поле \$System.AlarmUserDefNum1 записывать код замеса, меняющийся при производстве другого продукта, то выборку данных об алармах, возникших при производстве определённого вида продукции, можно выполнить по значениям столбца User1.</p>	
Категория	алармы	
Использование	Tagname.AlarmUserDefNum1	
	Параметр	Описание
	Tagname	Название любого тега логического, целого или вещественного типа, косвенного логического или аналогового тега либо группы алармов.
Примечание	<p>Данное поле определено для самых разных тегов, включая логические и аналоговые, а также группы алармов (независимо от того, определены или нет в них отдельные алармы). Эти поля можно оставить неопределёнными, установить их все либо только часть (для отдельных тегов, групп и родительских групп).</p>	
Тип данных	Аналоговый (чтение/запись)	
Допустимые значения	0.0 и "не установлено" (значение по умолчанию)	

Примеры	<p>Значение, указанное в этом поле, добавляется в запись базы данных об аларме, ЕСЛИ ТОЛЬКО это значение было установлено, например с помощью скрипта или команды POKE.</p> <p>\$System.AlarmUserDefNum1 = 4; GroupA.AlarmUserDefNum1 = 27649;</p> <p>В приведённых примерах полю присваиваются константы. Вместе с тем "пользовательские" поля можно определять и в скриптах InTouch, копируя в них значения других тегов. Для установки или проверки значения можно также воспользоваться функцией PtAcc либо серверными функциями системы InTouch.</p> <p>В целом, когда системе распределённых алармов посылается сигнал о возникновении аларма, используется уставка самого низкого уровня. То есть, если полю .AlarmUserDefNum1 какого-либо тега было присвоено некоторое значение, то в запись об аларме будет скопировано именно оно. Если же значение поля тега не было установлено, то WindowViewer проверит, не было ли оно определено для соответствующего поля группы алармов этого тега и т.д. вплоть до корневой группы \$System. Если же ни на одном уровне значение этого поля определено не было, то в запись об алармах ничего не будет записано (т.е. там будет 0 как число и пустая строка как строка символов).</p> <p>Примечание. Операция поиска выполняется отдельно для каждого элемента. Например, если поле .AlarmUserDefNum2 тега было определено, а поле .AlarmUserDefNum1 нет, то, если оно было определено для родительской группы этого тега, значение родительского поля .AlarmUserDefNum1 будет унаследовано соответствующим полем тега.</p>
См. также	.AlarmUserDefNum2, .AlarmUserDefStr

.AlarmUserDefNum2

	<p>Поле .AlarmUserDefNum2 позволяет заносить в запись об аларме дополнительные сведения, которые сохраняются в базе данных об алармах программой-регистратором Alarm DB Logger. (Это поле соответствует столбцу User2 таблицы базы данных.) "Определяемые пользователем" столбцы можно указывать в операторе SELECT для уточнения выборки сведений об алармах из базы данных. Например, если в поле \$System.AlarmUserDefNum2 записывать код замеса, меняющийся при производстве другого продукта, то выборку данных об алармах, возникших при производстве определённого вида продукции, можно выполнить по значениям столбца User2.</p>	
Категория	алармы	
Использование	Tagname.AlarmUserDefNum2	
	Параметр	Описание
	Tagname	Название любого тега логического, целого или вещественного типа, косвенного логического или аналогового тега либо группы алармов.
Примечание	<p>Данное поле определено для самых разных тегов, включая логические и аналоговые, а также группы алармов (независимо от того, определены или нет в них отдельные алармы). Эти поля можно оставить неопределёнными, установить их все либо только часть (для отдельных тегов, групп и родительских групп).</p>	
Тип данных	Аналоговый (чтение/запись)	
Допустимые значения	0.0 и "не установлено" (значение по умолчанию)	

<p>Примеры</p>	<p>Значение, указанное в этом поле, добавляется в запись базы данных об аларме, ЕСЛИ ТОЛЬКО это значение было установлено, например с помощью скрипта или команды POKE.</p> <p>\$System.AlarmUserDefNum2 = 4; GroupA.AlarmUserDefNum2 = 27649;</p> <p>В приведённых примерах полю присваиваются константы. Вместе с тем "пользовательские" поля можно определять и в скриптах InTouch, копируя в них значения других тегов. Для установки или проверки значения можно также воспользоваться функцией PtAcc либо серверными функциями системы InTouch.</p> <p>В целом, когда системе распределённых алармов посылается сигнал о возникновении аларма, используется уставка самого низкого уровня. То есть, если полю .AlarmUserDefNum2 какого-либо тега было присвоено некоторое значение, то в запись об аларме будет скопировано именно оно. Если же значение поля тега не было установлено, то WindowViewer проверит, не было ли оно определено для соответствующего поля группы алармов этого тега и т.д. вплоть до корневой группы \$System. Если же ни на одном уровне значение этого поля определено не было, то в запись об алармах ничего не будет записано (т.е. там будет 0 как число и пустая строка как строка символов).</p> <p>Примечание. Операция поиска выполняется отдельно для каждого элемента. Например, если поле .AlarmUserDefNum1 тега было определено, а поле .AlarmUserDefNum2 нет, то, если оно было определено для родительской группы этого тега, значение родительского поля .AlarmUserDefNum2 будет унаследовано соответствующим полем тега.</p>
<p>См. также</p>	<p>.AlarmUserDefNum1, .AlarmUserDefStr</p>

.AlarmUserDefStr

	<p>Поле .AlarmUserDefStr позволяет заносить в запись об аларме дополнительные сведения, которые сохраняются в базе данных об алармах программой-регистратором Alarm DB Logger. (Это поле соответствует столбцу User3 таблицы базы данных.) "Определяемые пользователем" столбцы можно указывать в операторе SELECT для уточнения выборки сведений об алармах из базы данных. Например, если в поле \$System.AlarmUserDefStr записывать текстовую характеристику замеса, меняющуюся при производстве другого продукта, то выборку данных об алармах, возникавших при производстве определённого вида продукции, можно выполнить по значениям столбца User3.</p>	
Категория	алармы	
Использование	Tagname.AlarmUserDefStr	
	Параметр	Описание
	Tagname	Название любого тега логического, целого или вещественного типа, косвенного логического или аналогового тега либо группы алармов.
Примечание	<p>Данное поле определено для самых разных тегов, включая логические и аналоговые, а также группы алармов (независимо от того, определены или нет в них отдельные алармы). Эти поля можно оставить неопределёнными, установить их все либо только часть (для отдельных тегов, групп и родительских групп).</p>	
Тип данных	Сообщение (чтение/запись)	
Допустимые значения	NULL и любая допустимая строка символов	

<p>Примеры</p>	<p>Значение, указанное в этом поле, добавляется в запись базы данных об аларме, ЕСЛИ ТОЛЬКО это значение было установлено, например с помощью скрипта или команды POKE.</p> <p>\$System.AlarmUserDefStr = "Joe";</p> <p>В приведённом примере полю был присвоен литерал. Вместе с тем "пользовательские" поля можно определять и в скриптах InTouch, копируя в них значения других тегов. Для установки или проверки значения можно также воспользоваться функцией PtAcc либо серверными функциями системы InTouch.</p> <p>В целом, когда системе распределённых алармов посылается сигнал о возникновении аларма, используется уставка самого низкого уровня. То есть, если полю .AlarmUserDefStr какого-либо тега было присвоено некоторое значение, то в запись об аларме будет скопировано именно оно. Если же значение поля тега не было установлено, то WindowViewer проверит, не было ли оно определено для соответствующего поля группы алармов этого тега и т.д. вплоть до корневой группы \$System. Если же ни на одном уровне значение этого поля определено не было, то в запись об алармах ничего не будет записано (т.е. там будет 0 как число и пустая строка как строка символов).</p> <p>Примечание. Операция поиска выполняется отдельно для каждого элемента. Например, если поле .AlarmUserDefNum1 тега было определено, а поле .AlarmUserDefStr нет, то, если оно было определено для родительской группы этого тега, значение родительского поля .AlarmUserDefStr будет унаследовано соответствующим полем тега.</p>
<p>См. также</p>	<p>.AlarmUserDefNum1, .AlarmUserDefNum2</p>

.AlarmValDeadband

	Используется для контроля значения зоны нечувствительности аларма	
Категория	алармы	
Использование	Tagname.AlarmValDeadband	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа либо косвенного аналогового тега.
Примечание	Установка опции Retentive Parameters (Сохранение параметров) в окне Словаря тегов обеспечивает автоматическое сохранение любых изменений значения этого поля в результате выполнения скриптов или анимационных функций.	
Тип данных	Аналоговый (чтение/запись)	
Допустимые значения	Должно быть в диапазоне значений, определённых для указанного тега.	
Примеры	Следующий оператор изменяет размер зоны нечувствительности для тега Tag1 в 25 %: Tag1.AlarmValDeadband = 25;	
См. также	.AlarmDevDeadband	

.AlarmValue

	Возвращает значение выбранного аларма. Выбор аларма должен осуществляться путём щелчка кнопкой мыши на соответствующей строке в окне сводных сведений об распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyM("ObjectName.AlarmValue", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название любого тега типа "сообщение".
Тип данных	Сообщение (только для чтения)	
Примеры	<p>GetPropertyM("AlmObj_1.AlarmValue", almValue); где AlmObj_1 – это название объекта вывода распределённых алармов, а almValue – название внутреннего тега типа "сообщение" (Memory Message), в который будет записано значение выбранного аларма.</p> <p>При использовании в скрипте экранной кнопки, этот оператор после нажатия на кнопку будет возвращать требуемое значение в almName, которое затем может быть использовано для дальнейшей обработки или вывода на экран. Тег almName будет содержать значение выбранного аларма соответствующего тега.</p>	
См. также	GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmTime, .AlarmType	

.AlarmValueCount

	Хранит общее число активных алармов по изменению значения для указанного тега или группы алармов.	
Категория	алармы	
Использование	Tagname.AlarmValueCount	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа или группы алармов.
Примечание	В данном поле хранится общее число алармов по выходу за границы HiHi, Hi, Lo и LoLo, как подтверждённых, так и не подтверждённых. В случае нерасширенной сводки алармов тега в поле всегда будет храниться значение 1. Следует учесть, что для различных групп алармов это значение будет разным.	
Тип данных	Целый (только для чтения)	
Допустимые значения	0 и любое положительное целое число.	
Примеры	AVC = Tag1.AlarmDevCount; где Tag1 – это тег аналогового типа, для которого возможно возникновение алармов по изменению значения. AVC тоже является аналоговым тегом, в который возвращается общее число активных алармов по значению, существующих для тега Tag1.	
См. также	.AlarmDevCount, .AlarmDevUnAckCount, .AlarmDscCount, .AlarmDscUnAckCount, .AlarmROCCount, .AlarmTotalCount, .AlarmValueUnAckCount, .AlarmROCUnAckCount, .AlarmUnAckCount	

.AlarmDevUnAckCount

	Хранит общее число активных неподтверждённых алармов по изменению значения для данного тега или группы алармов. Учитываются алармы выхода за границы HiHi, Hi, Lo и LoLo.	
Категория	алармы	
Использование	Tagname.AlarmValueUnAckCount	
	Параметр	Описание
	Tagname	Название любого тега целого или вещественного типа или группы алармов.
Тип данных	Целый (только для чтения)	
Допустимые значения	0 или любое положительное число	
Примеры	AVUC = Tag1.AlarmDevUnAckCount; где Tag1 – это тег аналогового типа, для которого возможно возникновение алармов по отклонению. AVUC тоже является аналоговым тегом, в который возвращается общее число неподтверждённых алармов по изменению значения, существующих для тега Tag1.	
См. также	.AlarmDevCount, .AlarmDevUnAckCount, .AlarmDscCount, .AlarmDscUnAckCount, .AlarmROCCount, .AlarmTotalCount, .AlarmValueCount, .AlarmROCUnAckCount, .AlarmUnAckCount	

.Caption

	Определяет текст, который должен отображаться рядом с окошком выбора	
Категория	алармы	
Использование	<pre>[ErrorNumber =]GetPropertyM("ControlName.Caption", Tagname); [ErrorNumber =]SetPropertyM("ControlName.Caption", "Message");</pre>	
	Параметр	Описание
	<i>ControlName</i>	Название объекта в окне, например ChkBox_4 .
	<i>Tagname</i>	Название любого тега типа "сообщение", в котором хранится выводимая строка.
	<i>Message</i>	Выводимая строка, заключённая в кавычки.
Примечание	Данное свойство является свойством "для чтения/записи" как во время разработки, так и во время исполнения	
Тип данных	Сообщение (только для чтения)	
Объекты применения	Окошки выбора	
Примеры	<p>Следующий оператор отображает рядом с окошком выбора CheckBox_1 строку "Blue Paint Option":</p> <pre>SetPropertyM("CheckBox_1.Caption", "Blue Paint Option");</pre>	
См. также	GetPropertyM(), SetPropertyM()	

.ChartLength

	Регулирует длину диаграммы архивного тренда (в секундах).	
Категория	архивирование	
Использование	TagName.ChartLength	
	Параметр	Описание
	TagName	Название любого тега типа "архивный тренд" (Hist Trend)
Примечание	<p>Данное поле для чтения/записи используется для установки (или определения) временной длительности графиков архивных трендов. Длительность выражается в секундах и определяется как период времени, данные для которого в текущий момент отображаются на графике архивного тренда. Эта величина определяется следующим образом:</p> <p>Длительность графика = (Временная отметка правого края) - (Временная отметка левого края);</p> <p>Поскольку отметки времени выражаются в виде числа секунд, прошедших с полуночи 1 января 1970 года, результатом вычислений является время в секундах левым и правым краем графика.</p> <p>При сложении с или вычитании из поля .ChartLength важно помнить, что его значение выражено в секундах. Поэтому, если из значения поля .ChartLength требуется вычесть "два часа", то эти два часа следует сначала перевести в секунды, т.е. из значения поля вычитать величину, равную (2 часа) * (60 мин/час) * (60 с/мин) = 7200 секунд.</p>	
Тип данных	Целый (чтение/запись)	
Допустимые значения	Любое целое положительное число	
Примеры	<p>Следующий оператор устанавливает длительность графика по времени, равную одному часу:</p> <p>HfTagName.ChartLength = 3600 {60 минут * 60 с/мин};</p> <p>Следующее выражение прокручивает график на половину влево:</p> <p>HfTagName.ChartStart = HfTagName.ChartStart - HfTagName.ChartLength/2;</p> <p>Следующее выражение прокручивает график влево на 10%:</p> <p>HfTagName.ChartStart = HfTagName.ChartStart - (.10 * HfTagName.ChartLength);</p>	
См. также	.ChartStart	

.ChartStart

	Определяет временную отметку левого края (дату и время суток) графика архивного тренда	
Категория	архивирование	
Использование	Tagname.ChartStart	
	Параметр	Описание
	Tagname	Название любого тега типа "архивный тренд" (Hist Trend)
Примечание	Данное поле для чтения/записи используется для установки (или определения) начального момента времени (соответствующего левому краю) графика архивного тренда. В поле .ChartStart содержится количество секунд, прошедших с 12:00 AM 1 января 1970 года. Начальный момент времени определяется как первая временная отметка данных левого края графика архивного тренда.	
Тип данных	Целый (чтение/запись)	
Допустимые значения	Любое целое положительное число	
Примеры	Следующий оператор прокручивает график на 1 минуту вправо: HtTagname.ChartStart = HtTagname.ChartStart + 60;	
См. также	.ChartLength	

.Comment

	Содержит значение поля комментария .Comment , указанного в окне определения тега в Словаре приложения	
Категория	теги	
Использование	Tagname.Comment	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега.
Примечание	Несмотря на то что это поле является полем "только для чтения" в WindowMaker, в WindowViewer его значение менять можно (только для внутренних тегов). После остановки исполнения и повторном запуске WindowViewer использоваться будет старое значение поля .Comment . Данное поле используется системой распределённых алармов для хранения комментария к аларму.	
Тип данных	Сообщение (только для чтения)	
Допустимые значения	Любая строка символов длиной от 1 до 50 символов. Примечание. Поле комментария тега может быть изменено только в окне определения тега в Словаре тегов приложения. Ввод значения в поле .Comment в среде WindowViewer не приведет к записи нового значения в Словарь тегов.	
Примеры	Следующий оператор формирует строку (содержимое тега типа "сообщение") путём объединения названия тега и поле комментария тега: OperatorMessage = MyTag.Name + " has a comment of: " + MyTag.Comment; Примечание. Это поле записывается только в рабочую базу данных времени выполнения и в Словаре тегов не сохраняется.	

.DevTarget

	Определение эталонного значения для алармов по значительному и незначительному отклонению	
Категория	алармы	
Использование	Tagname.DevTarget	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега целого или вещественного типа или косвенный аналоговый тег.
Примечание	Установка опции Retentive Parameters (Сохранение параметров) в окне Словаря тегов обеспечивает автоматическое сохранение любых изменений значения этого поля в результате выполнения скриптов или анимационных функций.	
Тип данных	Аналоговый (чтение/запись)	
Допустимые значения	Должно быть в диапазоне допустимых значений указанного тега	
Примеры	Следующий оператор устанавливает эталонное значение для тега MyTag равным 500: MyTag.DevTarget = 500;	
См. также	.AckDev, .AlarmDev, .AlarmMajDevDisabled, .AlarmMajDevEnabled, .AlarmMajDevInhibitor, .MajorDevSet, .MajorDevStatus	

.DisplayMode

	Определяет способ вывода значений на график тренда	
Категория	архивирование	
Использование	TagName.DisplayMode	
	Параметр	Описание
	<i>TagName</i>	Название любого тега типа "архивный тренд" (Hist Trend)
Тип данных	Аналоговый (чтение/запись)	
Допустимые значения	<p>1 = Выводит минимаксный график для каждого периода выборки тренда (значение по умолчанию)</p> <p>2 = Выводит арифметическое среднее для каждого периода выборки тренда в виде графика усреднения/разброса.</p> <p>3 = Выводит арифметическое среднее для каждого периода выборки тренда в виде столбиковой диаграммы.</p>	
Примеры	<p>Следующий оператор указывает, что архивный тренд с названием MyHistTrendTag будет выведен в виде столбиковой диаграммы:</p> <p>MyHistTrendTag.DisplayMode = 3;</p>	
См. также	.ChartLength, .ChartStart	

.Enabled

	Определяет, должен ли объект окна реагировать на действия пользователя	
Категория	управление окном	
Использование	<pre>[ErrorNumber =]GetPropertyD("ControlName.Enabled", Tagname); [ErrorNumber =]SetPropertyD("ControlName.Enabled", Discrete);</pre>	
	Параметр	Описание
	<i>ControlName</i>	Название объекта в окне, например ChkBox_4 .
	<i>Tagname</i>	Название логического тега, в который будет записано значение свойства.
	<i>Discrete</i>	Логическое значение: 0 = объект недоступен, 1 = объект доступен. Либо: название логического тега со значением, которое будет передано функции в момент исполнения.
Примечание	Данное свойство является свойством "для чтения/записи" как во время разработки, так и во время исполнения	
Тип данных	Логический (только для чтения)	
Объекты применения	Текстовые поля, списки, развёртывающиеся списки, окошки выбора и переключатели	
Примеры	Следующий оператор делает поле списка с названием ListBox_1 недоступным: SetPropertyD("ListBox_1.Enabled", 0);	
См. также	GetPropertyD(), SetPropertyD()	

.EngUnits

	Определение единиц измерения аналогового тега, заданных в Словаре тегов.	
Категория	теги	
Использование	Tagname.EngUnits	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Данное поле является текстовым, не влияющим ни на масштабирование, ни на преобразование, ни на форматирование значений тега.	
Тип данных	Сообщение (чтение/запись) Примечание. Записываемые в это поле значения после выключения системы не сохраняются.	
Допустимые значения	Любая строка символов длиной от 0 до 31 символа	
Примеры	IF Temperature.EngUnits == "Celsius" THEN CALL TempConvert(Temperature); {Этой Quick-функцией можно воспользоваться для преобразования температуры по шкале Цельсия в шкалу Фаренгейта } ENDIF;	
См. также	.MinEU, .MaxEU, .MaxRange, .MinRange, .MinRaw, .MaxRaw, .RawValue	

.Freeze

	Установка и определение состояния "замораживания" окна вывода сведений о распределённых алармах	
Категория	алармы	
Использование	<pre>[ErrorNumber =]GetPropertyD("ObjectName.Freeze", Tagname); [ErrorNumber =]SetPropertyD("ObjectName.Freeze", Tagname);</pre>	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода распределённых алармов, например AlmObj_1 .
	<i>Tagname</i>	Название логического тега, значение которого будет передано функции или который получит значение свойства во время исполнения.
Примечание	Поле для чтения/записи, с помощью которого производится определение или изменение состояния "замораживания" окна вывода сведений о распределённых алармах.	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = "замораживание" окна выключено, 1 = окно "заморожено"	
Примеры	Следующий оператор устанавливает состояние "замороженности" окна объекта AlmObj_1 в зависимости от значения логического тега AlmFreeze : SetPropertyD("AlmObj_1.Freeze", AlmFreeze);	
См. также	GetPropertyD(), SetPropertyD()	

.HiHiLimit

	Используется для определения или установки предела HiHi для проверки возникновения соответствующих алармов по изменению значения.	
Категория	алармы	
Использование	TagName.HiHiLimit	
	Параметр	Описание
	<i>TagName</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Установка опции Retentive Parameters (Сохранение) в окне Словаря тегов обеспечивает автоматическое сохранение любых изменений значения этого поля в результате выполнения скриптов или анимационных функций.	
Тип данных	Аналоговый (чтение/запись)	
Допустимые значения	Должны быть в диапазоне допустимых значений указанного тега	
Примеры	Следующий оператор увеличивает на 5 значение HiHiLimit (предельное значение) для тега с названием MyTag : MyTag.HiHiLimit = MyTag.HiHiLimit + 5;	
См. также	.Alarm, .AlarmValue, .Ack, .HiHiStatus, .HiHiSet, .AlarmDisabled, .AlarmEnabled, .AlarmHiHiDisabled, .AlarmHiHiEnabled, .AlarmHiHiInhibitor	

.HiHiSet

	Возвращает значение 1 (истина) либо 0 (ложь) в зависимости от того, было ли установлено предельное значение HiHi для указанного аналогового тега или нет.	
Категория	алармы	
Использование	Tagname.HiHiSet	
	Параметр	Описание
	<i>Tagname</i>	Название любого аналогового тега.
Примечание	Данное поле может использоваться только с тегами целого или вещественного типа.	
Тип данных	Логический	
Допустимые значения	1 (истина) или 0 (ложь)	
Примеры	<p>Следующий оператор может использоваться в анимационной функции вывода аналоговых значений для определения того, установлено ли предельное значение HiHi для тега MyTag:</p> <p>MyTag.HiHiSet; { "0" возвращается, если не установлено, "1" если установлено }</p>	
См. также	.Alarm, .AlarmValue, .Ack, .HiHiStatus, .HiHiLimit, .AlarmDisabled, .AlarmEnabled, .AlarmHiHiDisabled, .AlarmHiHiEnabled, .AlarmHiHiInhibitor	

.HiHstatus

	Указывает, существует ли аларм выхода за границу HiHi	
Категория	алармы	
Использование	Tagname.HiHstatus	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Как правило, значение этого поля "только для чтения" равно 0. При возникновении аларма по выходу за границу HiHi система устанавливает значение этого поля в 1. Это значение сохраняется всё время, пока существует данное состояние аларма. Поле .HiHstatus обычно используется в сочетании с полями .Alarm и .Ack для определения точного состояния аларма конкретного тега в системе.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = Указанное состояние аларма не наступило 1 = Указанное состояние аларма наступило	
Примеры	<p>Оператор в теле оператора IF-THEN будет выполняться только тогда, когда значение поля</p> <p>.HiHstatus тега MyTag равно 1 (существует аларм выхода за предельное значение HiHi):</p> <pre>IF (MyTag.HiHstatus == 1) THEN OperatorMessage = "MyTag has gone into High-High Alarm"; ENDIF;</pre>	
См. также	.Alarm, .AlarmValue, .Ack, .HiHiLimit, .HiHiSet, .AlarmDisabled, .AlarmEnabled, .AlarmHiHiDisabled, .AlarmHiHiEnabled, .AlarmHiHiInhibitor	

.HiLimit

	Используется для определения или установки предела Hi для проверки возникновения соответствующих алармов по изменению значения.	
Категория	алармы	
Использование	Tagname.HiLimit	
	Параметр	Описание
	<i>Tagname</i>	Название любого аналогового тега.
Примечание	Установка опции Retentive Parameters (Сохранение) в окне Словаря тегов обеспечивает автоматическое сохранение любых изменений значения этого поля в результате выполнения скриптов или анимационных функций.	
Тип данных	Аналоговый (чтение/запись)	
Допустимые значения	Должны быть в диапазоне допустимых значений указанного тега	
Примеры	Temperature.HiLimit = 212;	
См. также	.Alarm, .AlarmValue, .Ack, .Histatus, .HiSet, .AlarmDisabled, .AlarmEnabled, .AlarmHiDisabled, .AlarmHiEnabled, .AlarmHiInhibitor	

.HiSet

	Возвращает значение 1 (истина) либо 0 (ложь) в зависимости от того, было ли установлено предельное значение Hi для указанного аналогового тега или нет.	
Категория	алармы	
Использование	Tagname.HiSet	
	Параметр	Описание
	<i>Tagname</i>	Название любого аналогового тега.
Примечание	Данное поле может использоваться только с тегами целого или вещественного типа.	
Тип данных	Логический	
Допустимые значения	1 (истина) или 0 (ложь)	
Примеры	Следующий оператор может использоваться в анимационной функции вывода аналоговых значений для определения того, установлено ли предельное значение Hi для тега MyTag : MyTag.HiSet; { "0" возвращается, если не установлено, "1" если установлено }	
См. также	.Alarm, .AlarmValue, .Ack, .Histatus, .HiLimit, .AlarmDisabled, .AlarmEnabled, .AlarmHiDisabled, .AlarmHiEnabled, .AlarmHiInhibitor	

.Histatus

	Указывает, существует ли аларм выхода за границу Hi	
Категория	алармы	
Использование	Tagname.Histatus	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Как правило, значение этого поля "только для чтения" равно 0. При возникновении аларма по выходу за границу Hi система устанавливает значение этого поля в 1. Это значение сохраняется всё время, пока существует данное состояние аларма. Поле .Histatus обычно используется в сочетании с полями .Alarm и .Ack для определения точного состояния аларма конкретного тега в системе.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = Указанное состояние аларма не наступило 1 = Указанное состояние аларма наступило	
Примеры	IF (Motor.Histatus == 1) THEN Call PumpShutdown(); ENDIF; {Quick-функция используется для отключения двигателя насоса в случае нештатных параметров работы}	
См. также	.Alarm, .AlarmValue, .Ack, .HiLimit, .HiSet, .AlarmDisabled, .AlarmEnabled, .AlarmHiDisabled, .AlarmHiEnabled, .AlarmHiInhibitor	

.ListChanged

	Указывает, появились ли сведения о новых алармах либо обновлялись ли существующие сведения в окне распределённых алармов	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyD("ObjectName.ListChanged", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название логического тега, в который будет записано значение свойства в момент исполнения функции.
Примечание	Данное поле указывает, требуется ли обновлять окно вывода сведений о распределённых алармах в результате каких-либо изменений. Значение этого поля после считывания автоматически сбрасывается системой.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = новых алармов либо обновлений нет, 1 = сведения об алармах были обновлены	
Примеры	Следующий оператор устанавливает соответствующим образом значение логического тега AlmDispStat в зависимости от того, появились ли новые сведения для объекта вывода распределённых алармов AlmObj_1 : GetPropertyD("AlmObj_1.ListChanged", AlmDispStat);	
См. также	Get PropertyD()	

.ListCount

	Возвращает число элементов в обычном или развёртываемом списке	
Категория	управление окнами	
Использование	[ErrorNumber =]GetPropertyI("ControlName.ListCount", Tagname);	
	Параметр	Описание
	<i>ControlName</i>	Название оконного объекта, например ListBox_4 .
	<i>Tagname</i>	Название тега, в который будет записано число элементов списка.
Примечание	Данное свойство доступно только во время исполнения программы.	
Тип данных	Целый (только для чтения)	
Объекты применения	Простые и развёртываемые списки	
Примеры	Следующий оператор возвращает число элементов в списке ListBox_1 во внутренний тег целого типа MyListBoxCount : GetPropertyI("ListBox_1.ListCount", MyListBoxCount);	
См. также	GetPropertyI(), .ListIndex	

.ListIndex

	<p>Определяет или устанавливает порядковый номер элемента списка (определяется аргументами Tagname или Number). Значение аргумента "-1" для простых списков означает, что в текущий момент никакой элемент не выбран. Значение аргумента "-1" для развёртываемых списков означает, что пользователь ввёл в окно списка новое значение.</p>	
Категория	управление окнами	
Использование	<pre>[ErrorNumber =]GetPropertyI("ControlName.ListIndex", Tagname); [ErrorNumber =]SetPropertyI("ControlName.ListIndex", Number);</pre>	
	Параметр	Описание
	<i>ControlName</i>	Название оконного объекта, например ListBox_4 .
	<i>Tagname</i>	Допустимое название тега, хранящего количество элементов в списке.
	<i>Number</i>	Целое число, определяющее конкретный элемент списка.
Примечание	Данное свойство доступно только во время исполнения программы.	
Тип данных	Целый (только для чтения)	
Объекты применения	Простые и развёртываемые списки	
Примеры	<p>Следующий оператор возвращает во внутренний тег целого типа MyListBoxIndex порядковый номер элемента, выбранного в текущий момент в списке ListBox_1:</p> <pre>GetPropertyI("ListBox_1.ListIndex", MyListBoxIndex);</pre>	
См. также	GetPropertyI(), SetPropertyI(), .NewIndex, .TopIndex	

.LoLimit

	Используется для определения или установки предела Lo для проверки возникновения соответствующих алармов по изменению значения.	
Категория	алармы	
Использование	Tagname.LoLimit	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега целого или вещественного типа либо косвенного аналогового тега.
Примечание	Установка опции Retentive Parameters (Сохранение) в окне Словаря тегов обеспечивает автоматическое сохранение любых изменений значения этого поля в результате выполнения скриптов или анимационных функций.	
Тип данных	Аналоговый (чтение/запись)	
Допустимые значения	Должны быть в диапазоне допустимых значений указанного тега	
Примеры	Следующий оператор уменьшает на 10 нижнее предельное значение Lo тега MyTag : MyTag.LoLimit = MyTag.LoLimit - 10;	
См. также	.Alarm, .AlarmValue, .Ack, .LoStatus, .LoSet, .AlarmDisabled, .AlarmEnabled, .AlarmLoDisabled, .AlarmLoEnabled, .AlarmLoInhibitor	

.LoLoLimit

	Используется для определения или установки предела LoLo для проверки возникновения соответствующих алармов по изменению значения.	
Категория	алармы	
Использование	TagName.LoLoLimit	
	Параметр	Описание
	<i>TagName</i>	Название любого тега целого или вещественного типа либо косвенного аналогового тега.
Примечание	Установка опции Retentive Parameters (Сохранение) в окне Словаря тегов обеспечивает автоматическое сохранение любых изменений значения этого поля в результате выполнения скриптов или анимационных функций.	
Тип данных	Аналоговый (чтение/запись)	
Допустимые значения	Должны быть в диапазоне допустимых значений указанного тега	
Примеры	Следующий оператор уменьшает на 10 второе нижнее предельное значение LoLo тега MyTag : MyTag.LoLoLimit = MyTag.LoLoLimit - 10;	
См. также	.Alarm, .AlarmValue, .Ack, .LoLoStatus, .LoLoSet, .AlarmDisabled, .AlarmEnabled, .AlarmLoLoDisabled, .AlarmLoLoEnabled, .AlarmLoLoInhibitor	

.LoSet

	Возвращает значение 1 (истина) либо 0 (ложь) в зависимости от того, было ли установлено предельное значение Lo для указанного аналогового тега или нет.	
Категория	алармы	
Использование	Tagname.LoSet	
	Параметр	Описание
	<i>Tagname</i>	Название любого аналогового тега.
Примечание	Данное поле может использоваться только с тегами целого или вещественного типа.	
Тип данных	Логический	
Допустимые значения	1 (истина) или 0 (ложь)	
Примеры	Следующий оператор может использоваться в анимационной функции вывода аналоговых значений для определения того, установлено ли предельное значение Lo для тега MyTag : MyTag.LoSet; { "0" возвращается, если не установлено, "1" если установлено }	
См. также	.Alarm, .AlarmValue, .Ack, .LoStatus, .LoLimit, .AlarmDisabled, .AlarmEnabled, .AlarmLoDisabled, .AlarmLoEnabled, .AlarmLoInhibitor	

.LoLoSet

	Возвращает значение 1 (истина) либо 0 (ложь) в зависимости от того, было ли установлено предельное значение LoLo для указанного аналогового тега или нет.	
Категория	алармы	
Использование	TagName.LoLoSet	
	Параметр	Описание
	<i>TagName</i>	Название любого аналогового тега.
Примечание	Данное поле может использоваться только с тегами целого или вещественного типа.	
Тип данных	Логический	
Допустимые значения	1 (истина) или 0 (ложь)	
Примеры	Следующий оператор может использоваться в анимационной функции вывода аналоговых значений для определения того, установлено ли предельное значение LoLo для тега MyTag : MyTag.LoLoSet; { "0" возвращается, если не установлено, "1" если установлено }	
См. также	.Alarm, .AlarmValue, .Ack, .LoLoStatus, .LoLoLimit, .AlarmDisabled, .AlarmEnabled, .AlarmLoLoDisabled, .AlarmLoLoEnabled, .AlarmLoLoInhibitor	

.LoLoStatus

	Указывает, существует ли аларм выхода за границу LoLo	
Категория	алармы	
Использование	Tagname.LoLoStatus	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Как правило, значение этого поля "только для чтения" равно 0. При возникновении аларма по выходу за границу LoLo система устанавливает значение этого поля в 1. Это значение сохраняется всё время, пока существует данное состояние аларма. Поле .LoLoStatus обычно используется в сочетании с полями .Alarm и .Ack для определения точного состояния аларма конкретного тега в системе.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = Указанное состояние аларма не наступило 1 = Указанное состояние аларма наступило	
Примеры	<p>Оператор в теле оператора IF-THEN будет выполняться только тогда, когда значение поля</p> <p>.LoLoStatus тега MyTag равно 1 (существует аларм выхода за второе предельное значение LoLo):</p> <pre>IF (MyTag.LoLoStatus == 1) THEN OperatorMessage = "MyTag has gone into LoLo Alarm"; ENDIF;</pre>	
См. также	.Alarm, .AlarmValue, .Ack, .LoLoLimit, .LoLoSet, .AlarmDisabled, .AlarmEnabled, .AlarmLoLoDisabled, .AlarmLoLoEnabled, .AlarmLoLoInhibitor	

.LoStatus

	Указывает, существует ли аларм выхода за границу Lo	
Категория	алармы	
Использование	TagName.LoStatus	
	Параметр	Описание
	<i>TagName</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Как правило, значение этого поля "только для чтения" равно 0. При возникновении аларма по выходу за границу Lo система устанавливает значение этого поля в 1. Это значение сохраняется всё время, пока существует данное состояние аларма. Поле .LoStatus обычно используется в сочетании с полями .Alarm и .Ack для определения точного состояния аларма конкретного тега в системе.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = Указанное состояние аларма не наступило 1 = Указанное состояние аларма наступило	
Примеры	<pre>IF (WaterLevelTank1.LoStatus == 1) THEN PumpShutdown = 1; WaterFillValue = 1; ENDIF;</pre>	
См. также	.Alarm, .AlarmValue, .Ack, .LoLimit, .LoSet, .AlarmDisabled, .AlarmEnabled, .AlarmLoDisabled, .AlarmLoEnabled, .AlarmLoInhibitor	

.MajorDevPct

	Используется для определения и установки величины значительного отклонения (в процентах) для проверки возникновения аларма по значительному отклонению	
Категория	алармы	
Использование	TagName.MajorDevPct	
	Параметр	Описание
	<i>TagName</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Установка опции Retentive Parameters (Сохранение) в окне Словаря тегов обеспечивает автоматическое сохранение любых изменений значения этого поля в результате выполнения скриптов или анимационных функций.	
Тип данных	Вещественный (чтение/запись)	
Допустимые значения	от 0 до 100 (%)	
Примеры	Следующий оператор устанавливает для тега MyTag величину предела значительного отклонения в 25%: MyTag.MajorDevPct = 25;	
См. также	.AckDev, .AlarmDev, .AlarmMajDevDisabled, .AlarmMajDevEnabled, .AlarmMajDevInhibitor, .MajorDevSet, .MajorDevStatus	

.MajorDevSet

	Возвращает значение 1 (истина) либо 0 (ложь) в зависимости от того, было ли установлено поле значительного отклонения для указанного аналогового тега или нет.	
Категория	алармы	
Использование	TagName.MajorDevSet	
	Параметр	Описание
	<i>TagName</i>	Название любого аналогового тега.
Примечание	Данное поле может использоваться только с тегами целого или вещественного типа.	
Тип данных	Логический	
Допустимые значения	1 (истина) или 0 (ложь)	
Примеры	Следующий оператор может использоваться в анимационной функции вывода аналоговых значений для определения того, установлено ли поле предельной величины значительного отклонения для тега MyTag : MyTag.MajorDevSet; { "0" возвращается, если не установлено, "1" если установлено }	
См. также	.AckDev, .AlarmDev, .AlarmMajDevDisabled, .AlarmMajDevEnabled, .AlarmMajDevInhibitor, .MajorDevPct, .MajorDevStatus	

.MajorDevStatus

	Указывает, существует ли аларм по значительному отклонению	
Категория	алармы	
Использование	Tagname.MajorDevStatus	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Как правило, значение этого поля "только для чтения" равно 0. При возникновении аларма по значительному отклонению система устанавливает значение этого поля в 1. Это значение сохраняется всё время, пока существует данное состояние аларма. Поле .MajorDevStatus обычно используется в сочетании с полями .Alarm и .Ack для определения точного состояния аларма конкретного тега в системе.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = Указанное состояние аларма не наступило 1 = Указанное состояние аларма наступило	
Примеры	<p>Оператор в теле оператора IF-THEN будет выполняться только тогда, когда значение поля</p> <p>.MajorDevStatus тега MyTag равно 1 (существует аларм по значительному отклонению):</p> <pre> IF (MyTag.MajorDevStatus == 1) THEN OperatorMessage = "MyTag has gone into Major Deviation Alarm"; ENDIF; </pre>	
См. также	.AckDev, .AlarmDev, .AlarmMajDevDisabled, .AlarmMajDevEnabled, .AlarmMajDevInhibitor, .MajorDevPct, .MajorDevSet	

.MaxEU

	Верхняя предельная граница допустимых значений указанного тега в единицах измерения	
Категория	теги	
Использование	TagName.MaxEU	
	Параметр	Описание
	<i>TagName</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	<p>Параметры, поступающие в систему WindowViewer от сервера ввода/вывода, представляют собой исходные значения, которые бывает необходимо масштабировать. Параметры масштабирования определяются значениями полей .MinEU и .MaxEU. В качестве примера допустим, что программируемый контроллер (ПЛК) считал показания некоторого датчика уровня. Датчик уровня посылает сигнал в диапазоне от 4 до 20 мА. ПЛК с помощью аналого-цифрового преобразователя преобразует это значение в целую величину от 0 до 4096, которая записывается в тег TankTwoLevel.</p> <p>Вывод на дисплей такого необработанного значения (от 0 до 4096) не несет оператору никакой полезной информации. Поэтому его необходимо ещё раз преобразовать в некоторую величину, выраженную в единицах измерения (масштабировать). Для этого необходимо правильно задать поля минимального и максимального значения единиц измерения. Если, например, необработанное значение 0 (4 мА) означает "0 литров", а значение 4096 (20 мА) означает "100 литров", то для отображения на экране значимой величины следует сделать следующие установки:</p> <p>Minimum Raw = 0, Maximum Raw = 4096, Minimum Engineering Units = 0, Maximum Engineering Units = 100</p> <p>При таких установках при получении от датчика необработанного значения 4096 на экране будет показано значение 100.</p>	
Тип данных	Целый для тегов целого типа и вещественный для тегов вещественного типа (только для чтения)	
Допустимые значения	Определяется характеристиками указанного тега	
Примеры	<p>Поскольку это поле имеет атрибут "только для чтения", ему нельзя присваивать значения. Отображение значений полей .MinEU и .MaxEU на экране имеет очень большое значение с точки зрения понимания оператором производимых вычислений и получаемых показаний.</p> <p>DialogValueEntry ("IO_Point_717", IO_Point_717.MinEU, IO_Point_717.MaxEU, "Please Enter a New Value:");</p>	
См. также	.EngUnits, .MinEU, .MaxRange, .MinRange, .MinRaw, .MaxRaw, .RawValue	

.MaxRange

	Верхняя граница значений каждого тега, выводимых на график архивного тренда, в процентах от диапазона значений тега, выраженного в единицах измерения	
Категория	архивирование	
Использование	Tagname.MaxRange	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега типа "архивный тренд" (Hist Trend).
Примечание	Поскольку в окне архивного тренда могут строиться кривые для самых разных тегов, указывать верхние и нижние границы (поля .MinRange и .MaxRange) в единицах измерения не имеет смысла, поскольку теги могут выражаться (и, как правило, выражаются) в несопоставимых единицах измерения. Границы диапазона целесообразнее выразить в процентной величине от диапазона допустимых значений каждого тега. Таким образом, независимо от того, в каких единицах измеряется значение того или иного тега, на тренде они будут отображаться унифицированным образом.	
Тип данных	Вещественный (чтение/запись)	
Допустимые значения	Значения полей .MinRange и .MaxRange задаются от 0 до 100 и выражаются в процентах. Значение поля .MinRange должно быть меньше значения поля .MaxRange . Если указанное значение меньше отрицательно либо превышает 100, то оно приводится к соответствующему пределу (0 или 100). Если значение поля .MinRange будет превышать значение поля .MaxRange , то на тренд не будут выводиться никакие данные.	
Примеры	Следующий оператор устанавливает верхнюю границу отображаемых на тренде MyHistTrendTag значений тега в 25%: MyHistTrendTag.MaxRange = 25;	
См. также	.ChartStart, .ChartLength, .DisplayMode, .EngUnits, .MinEU, .MaxEU, .MinRange, .MinRaw, .MaxRaw, .RawValue	

.MaxRaw

	Максимально допустимая граница исходных значений, получаемых от сервера ввода/вывода приложение WindowViewer, выступающим в роли клиента. Значение этого поля определяется значением в поле Max Raw (Максимальное исходное значение) окна определения соответствующего тега ввода/вывода (в Словаре тегов). Если значение считываемого показателя начинает превышать эту величину, то в WindowViewer пересылается не оно, а указанное максимально допустимое значение.	
Категория	теги	
Использование	Tagname.MaxRaw	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега ввода/вывода логического типа, косвенного логического тега, тега ввода/вывода целого типа, внутреннего вещественного, косвенного аналогового, тега ввода/вывода типа "сообщение" либо косвенного тега типа "сообщение".
Примечание	Данное поле только для чтения используется только для индикации максимально допустимой величины исходных значений.	
Тип данных	Целый или вещественный (только для чтения)	
Допустимые значения	Любое аналоговое значение	
Примеры	Следующий оператор используется для определения выхода значений тега из диапазона нормальных значений и фиксации значения на одной из границ диапазона. IF ((Temp01.RawValue > Temp01.MaxRaw) OR (Temp01.RawValue < Temp01.MinRaw)) THEN Show "Instrument Failure Window"; ENDIF;	
См. также	.EngUnits, .MinEU, .MaxEU, .MaxRange, .MinRange, .MinRaw, .RawValue	

.MinEU

	Нижняя предельная граница допустимых значений указанного тега в единицах измерения	
Категория	теги	
Использование	TagName.MinEU	
	Параметр	Описание
	<i>TagName</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	<p>Параметры, поступающие в систему WindowViewer от сервера ввода/вывода, представляют собой исходные значения, которые бывает необходимо масштабировать. Параметры масштабирования определяются значениями полей .MinEU и .MaxEU. В качестве примера допустим, что программируемый контроллер (ПЛК) считал показания некоторого датчика уровня. Датчик уровня посылает сигнал в диапазоне от 4 до 20 мА. ПЛК с помощью аналого-цифрового преобразователя преобразует это значение в целую величину от 0 до 4096, которая записывается в тег TankTwoLevel.</p> <p>Вывод на дисплей такого необработанного значения (от 0 до 4096) не несет оператору никакой полезной информации. Поэтому его необходимо ещё раз преобразовать в некоторую величину, выраженную в единицах измерения (масштабировать). Для этого необходимо правильно задать поля минимального и максимального значения единиц измерения. Если, например, необработанное значение 0 (4 мА) означает "0 литров", а значение 4096 (20 мА) означает "100 литров", то для отображения на экране значимой величины следует сделать следующие установки:</p> <p>Minimum Raw = 0, Maximum Raw = 4096, Minimum Engineering Units = 0, Maximum Engineering Units = 100</p> <p>При таких установках при получении от датчика необработанного значения 4096 на экране будет показано значение 100.</p>	
Тип данных	Целый для тегов целого типа и вещественный для тегов вещественного типа (только для чтения)	
Допустимые значения	Определяется характеристиками указанного тега	
Примеры	AbsoluteTagRange = (Tag.MaxEU - Tag.MinEU);	
См. также	.EngUnits, .MaxEU, .MaxRange, .MinRange, .MinRaw, .MaxRaw, .RawValue	

.MinorDevPct

	Используется для определения и установки величины незначительного отклонения (в процентах) для проверки возникновения аларма по значительному отклонению	
Категория	алармы	
Использование	TagName.MinorDevPct	
	Параметр	Описание
	<i>TagName</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Установка опции Retentive Parameters (Сохранение) в окне Словаря тегов обеспечивает автоматическое сохранение любых изменений значения этого поля в результате выполнения скриптов или анимационных функций.	
Тип данных	Вещественный (чтение/запись)	
Допустимые значения	от 0 до 100 (%)	
Примеры	Следующий оператор устанавливает для тега MyTag величину предела незначительного отклонения в 25%: MyTag.MinorDevPct = 25;	
См. также	.AckDev, .AlarmDev, .AlarmMinDevDisabled, .AlarmMinDevEnabled, .AlarmMinDevInhibitor, .MinorDevSet, .MinorDevStatus	

.MinorDevSet

	Возвращает значение 1 (истина) либо 0 (ложь) в зависимости от того, было ли установлено поле незначительного отклонения для указанного аналогового тега или нет.	
Категория	алармы	
Использование	TagName.MinorDevSet	
	Параметр	Описание
	<i>TagName</i>	Название любого аналогового тега.
Примечание	Данное поле может использоваться только с тегами целого или вещественного типа.	
Тип данных	Логический	
Допустимые значения	1 (истина) или 0 (ложь)	
Примеры	Следующий оператор может использоваться в анимационной функции вывода аналоговых значений для определения того, установлено ли поле предельной величины незначительного отклонения для тега MyTag : MyTag.MinorDevSet; { "0" возвращается, если не установлено, "1" если установлено }	
См. также	.AckDev, .AlarmDev, .AlarmMinDevDisabled, .AlarmMinDevEnabled, .AlarmMinDevInhibitor, .MinorDevPct, .MinorDevStatus	

.MinorDevStatus

	Указывает, существует ли аларм по незначительному отклонению	
Категория	алармы	
Использование	Tagname.MinorDevStatus	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Как правило, значение этого поля "только для чтения" равно 0. При возникновении аларма по значительному отклонению система устанавливает значение этого поля в 1. Это значение сохраняется всё время, пока существует данное состояние аларма. Поле .MinorDevStatus обычно используется в сочетании с полями .Alarm и .Ack для определения точного состояния аларма конкретного тега в системе.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = Указанное состояние аларма не наступило 1 = Указанное состояние аларма наступило	
Примеры	<p>Оператор в теле оператора IF-THEN будет выполняться только тогда, когда значение поля</p> <p>.MinorDevStatus тега MyTag равно 1 (существует аларм по незначительному отклонению):</p> <pre>IF (MyTag.MinorDevStatus == 1) THEN OperatorMessage = "MyTag has gone into Minor Deviation Alarm"; ENDIF;</pre>	
См. также	.AckDev , .AlarmDev , .AlarmMinDevDisabled , .AlarmMinDevEnabled , .AlarmMinDevInhibitor , .MinorDevPct	

.MinRange

	Нижняя граница значений каждого тега, выводимых на график архивного тренда, в процентах от диапазона значений тега, выраженного в единицах измерения	
Категория	архивирование	
Использование	Tagname.MinRange	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега типа "архивный тренд" (Hist Trend).
Примечание	Поскольку в окне архивного тренда могут строиться кривые для самых разных тегов, указывать верхние и нижние границы (поля .MinRange и .MaxRange) в единицах измерения не имеет смысла, поскольку теги могут выражаться (и, как правило, выражаются) в несопоставимых единицах измерения. Границы диапазона целесообразнее выразить в процентной величине от диапазона допустимых значений каждого тега. Таким образом, независимо от того, в каких единицах измеряется значение того или иного тега, на тренде они будут отображаться унифицированным образом.	
Тип данных	Вещественный (чтение/запись)	
Допустимые значения	Значения полей .MinRange и .MaxRange задаются от 0 до 100 и выражаются в процентах. Значение поля .MinRange должно быть меньше значения поля .MaxRange . Если указанное значение меньше отрицательно либо превышает 100, то оно приводится к соответствующему пределу (0 или 100). Если значение поля .MinRange будет превышать значение поля .MaxRange , то на тренд не будут выводиться никакие данные.	
Примеры	TotalChartHeight = MyHistTrendTag.MaxRange - MyHistTrendTag.MinRange;	
См. также	.ChartStart, .ChartLength, .DisplayMode, .EngUnits, .MinEU, .MaxEU, .MaxRange, .MinRaw, .MaxRaw, .RawValue	

.MinRaw

	Минимально допустимая граница исходных значений, получаемых от сервера ввода/вывода приложение WindowViewer, выступающим в роли клиента. Значение этого поля определяется значением в поле Min Raw (Минимальное исходное значение) окна определения соответствующего тега ввода/вывода (в Словаре тегов). Если значение считываемого показателя становится меньше этой величины, то в WindowViewer пересылается не оно, а указанное минимально допустимое значение.	
Категория	теги	
Использование	Tagname.MinRaw	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега ввода/вывода логического типа, косвенного логического тега, тега ввода/вывода целого типа, внутреннего вещественного, косвенного аналогового, тега ввода/вывода типа "сообщение" либо косвенного тега типа "сообщение".
Примечание	Данное поле только для чтения используется только для индикации минимально допустимой величины исходных значений.	
Тип данных	Целый или вещественный (только для чтения)	
Допустимые значения	Любое аналоговое значение	
Примеры	Следующий оператор используется для определения выхода значений тега из диапазона нормальных значений и фиксации значения на одной из границ диапазона. IF ((Temp01.RawValue > Temp01.MaxRaw) OR (Temp01.RawValue < Temp01.MinRaw)) THEN Show "Instrument Failure Window"; ENDIF;	
См. также	.EngUnits, .MinEU, .MaxEU, .MaxRange, .MinRange, .MaxRaw, .RawValue	

.Name

	Содержит название указанного тега в виде строки символов	
Категория	теги	
Использование	Tagname.Name	
	Параметр	Описание
	<i>Tagname</i>	Тег любого типа
Тип данных	Сообщение (чтение/запись)	
Допустимые значения	<p>Присвоение значений этому полю возможно только для косвенных тегов, включая группы алармов и теги идентификаторов (типа Alarm Group и TagID). При этом формат присваиваемой строки должен соответствовать правилам именования тегов, например в ней должны использоваться только допустимые символы, а первым знаком должен быть алфавитно-цифровой символ.</p>	
Примечание	<p>Данное поле имеет атрибут "только для чтения" для следующих типов тегов: архивный тренд (Hist Trend), любой тип внутренних тегов, любой тип тегов для ввода/вывода.</p> <p>Поле имеет атрибут "для чтения/записи" для следующих типов тегов: групповая переменная (Group Var), идентификатор тега (Tag ID), любой тип косвенных тегов.</p> <p>В случае, если поле является полем "только для чтения", название тега в кавычки заключать не нужно (т.е. запись MyTag.Name эквивалента записи MyTag). Например, при выполнении команды обновления счётчиков использования тегов (команда Update Use Counts меню Special приложения WindowMaker) либо построении таблицы кросс-ссылок система не "увидит" тег, если его название будет заключено в кавычки. Таким образом, если название тега будет в кавычках, то после обновления счётчиков использования и последующего выполнения команды Delete unused tags (Удалить неиспользуемые теги) этот тег будет удалён, поскольку будет рассматриваться системой как "неиспользуемый".</p> <p>Поле с атрибутом "для чтения/записи" позволяет выполнять ещё больше функций. С его помощью можно, например изменять "суть" косвенного тега. Когда этому полю присваивается название другого тега, то, если оба тега принадлежат к одному и тому же типу, указанный косвенный тег фактически <u>становится</u> этим другим тегом и его можно использовать в системе InTouch таким же образом, как и тот, на который он ссылается.</p>	

Примеры

В следующем операторе значение этого поля используется для составления сообщения оператору:

```
MyMessageTag = "The tag named " + MyTag.Name + " has a  
comment of " + MyTag.Comment;
```

Другой пример: пусть требуется создать графическое окно для отображения трех важнейших статистических параметров нефтяной скважины, причём на предприятии имеется 100 таких скважин. Следующий скрипт по изменению данных позволяет обновлять данные в графическом окне независимо от того, какую скважину укажет оператор. В скрипте для этой цели используется тег **OilWellNumber**.

```
IndOilWellPump.Name = "OilWellPump" +  
Text(OilWellNumber, "#");
```

```
IndOilWellTEMP.Name = "OilWellTemp" +  
Text(OilWellNumber, "#");
```

```
IndOilWellPressure.Name = "OilWellPressure" +  
Text(OilWellNumber, "#");
```

Для выполнения этого скрипта следует создать экранную кнопку с анимационной функцией ввода аналоговых величин для тега **OilWellNumber**. Во время исполнения программы этот скрипт будет запускаться, когда оператор нажмёт эту кнопку и введёт номер скважины.

.NewIndex

	Возвращает порядковый номер элемента, последним занесённого в простой либо развёртывающийся список функцией wcAddItem() или wcInsertItem()	
Категория	управление окнами	
Использование	[ErrorNumber =]GetPropertyI("ControlName.NewIndex", Tagname);	
	Параметр	Описание
	<i>ControlName</i>	Название оконного элемента, например ListBox_4 .
	<i>Tagname</i>	Название тега целого типа, в который будет записан порядковый номер элемента, добавленного в простой либо развёртывающийся список последним. Если список пуст, возвращается -1.
Примечание	Данное свойство доступно только во время исполнения программы.	
Тип данных	Целый (только для чтения)	
Объекты применения	Простые и развёртывающиеся списки	
Примеры	При выполнении следующего оператора в тег NewItemIndex целого типа записывается порядковый номер элемента, занесённого в список ListBox_1 в последний раз GetPropertyI("ListBox_1.NewIndex", NewItemIndex);	
См. также	GetPropertyI(), wcAddItem(), wcInsertItem(), .ListIndex, .TopIndex	

.NextPage

	При изменении значения этого поля с 1 на 0 выполняется "прокручивание" экрана объекта вывода сведений о распределённых алармов на одну страницу вниз.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyD("ObjectName.NextPage", Tagname); [ErrorNumber =]SetPropertyD("ObjectName.NextPage", Value);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название тега логического типа, в который будет записано значение свойства в момент выполнения функции и который указывается в виде " Название_тега " либо Название_тега.Name .
	<i>Value</i>	Логическое значение либо название логического тега с требуемым логическим значением.
Примечание	При изменении значения поля с 1 на 0 на экран выводится следующая страница сведений о распределённых алармах, после чего логическое значение автоматически устанавливается в 1. Если будет достигнут конец списка страниц, значение поля не изменится и останется равным 0.	
Тип данных	Логический (чтение/запись)	
См. также	GetPropertyD(), SetPropertyD(), .PrevPage, .PageNum, .TotalPages	

.Normal

	Значение поля равно 1, если алармов для указанного тега нет.	
Категория	алармы	
Использование	TagName.Normal	
	Параметр	Описание
	<i>TagName</i>	Название любого тега логического, целого или вещественного типа или косвенного аналогового тега.
Примечание	Данное поле используется в скриптах или анимационных функциях отображения для индикации того, что указанный тег находится в "нормальном" состоянии (т.е. активных алармов нет). При возникновении хотя бы одного аларма значение поля становится равным нулю.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = для указанного тега существуют алармы, 1 = алармы для указанного тега отсутствуют	
Примеры	<p>Операторы в теле выражения IF-THEN будут выполняться только при отсутствии алармов тега Tag1. Если с ним связан хотя бы один активный аларм, будут выполнены операторы в теле конструкции ELSE:</p> <pre> IF (Tag1.Normal == 1) THEN MyOperatorMessage = "Tag1 is OK - No alarms associated with it"; ELSE MyOperatorMessage = "Tag1 has one or more alarms active!"; ENDIF; </pre>	
См. также	.Alarm, .AlarmDev, .AlarmROC, .AlarmValue	

.NumAlarms

	Возвращает число алармов в данном объекте вывода сведений о распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyI("ObjectName.NumAlarms", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано значение этого свойства при выполнении функции.
Примечание	Данное поле содержит общее число всех алармов, зарегистрированных в указанном объекте вывода сведений о распределённых алармах, а не только отображаемых в текущий момент.	
Тип данных	Целый (только для чтения)	
Примеры	При выполнении следующего оператора в тег AlarmCount целого типа записывается текущее число всех алармов, зарегистрированных в объекте вывода распределённых алармов с названием AlmObj_1 : GetPropertyI("AlmObj_1.NumAlarms", AlarmCount);	
См. также	GetPropertyI()	

.OffMsg

	Возвращает текстовое сообщение для состояния Off логического тега, указанное при его определении в Словаре тегов.	
Категория	теги	
Использование	Tagname.OffMsg	
	Параметр	Описание
	<i>Tagname</i>	Название тега логического типа.
Тип данных	Сообщение (чтение/запись). Строки, записываемые в это поле, после завершения работы приложения не сохраняются.	
Допустимые значения	Строки символов длиной от 0 до 15 символов.	
Примеры	Если значение тега MyDiscrete равно 0, то в тег StateMessage записывается строка OffMsg , которая была определена в Словаре приложения для логического тега MyDiscrete . StateMessage = Dtext (MyDiscrete, MyDiscrete.OnMsg, MyDiscrete.OffMsg);	
См. также	.OnMsg	

.OnMsg

	Возвращает текстовое сообщение для состояния On логического тега, указанное при его определении в Словаре тегов.	
Категория	теги	
Использование	Tagname.OnMsg	
	Параметр	Описание
	<i>Tagname</i>	Название тега логического типа.
Тип данных	Сообщение (чтение/запись). Строки, записываемые в это поле, после завершения работы приложения не сохраняются.	
Допустимые значения	Строки символов длиной от 0 до 15 символов.	
Примеры	IF IndAnalog.OnMsg == "Running" THEN TypeOfTag = "IndAnalog was assigned a Motor Starter"; ENDIF;	
См. также	.OffMsg	

.PageNum

	Содержит номер текущей страницы, отображаемой в окне объекта вывода сведений о распределённых алармах	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyI("ObjectName.PageNum", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано значение свойства при выполнении функции.
Примечание	Данное поле "только для чтения" содержит номер страницы, в текущий момент отображаемой в окне объекта вывода сведений о распределённых алармах.	
Тип данных	Целый (только для чтения)	
Примеры	При выполнении следующего оператора в тег AlarmPage целого типа записывается номер текущей страницы данных объекта вывода сведений о распределённых алармах AlmObj_1 : GetPropertyI("AlmObj_1.PageNum", AlarmPage);	
См. также	GetPropertyI(), .NextPage, .PrevPage, .TotalPages	

.Pen1 – .Pen8

	Содержит идентификатор тега, кривая значений которого строится соответствующим пером архивного тренда	
Категория	архивирование	
Использование	<i>TagName</i> {.Pen1 .Pen2 .Pen3 .Pen4 .Pen5 .Pen6 .Pen7 .Pen8};	
	Параметр	Описание
	<i>TagName</i>	Название тега типа "архивный тренд" (Hist Trend).
Примечание	<p>В связи со сложностями использования полей .PenX рекомендуется по возможности использовать функции HTSetPenName() и HTGetPenName().</p> <p>Примечание. Полям .PenX можно присваивать названия только локальных тегов. Нотация provider.tag является недопустимой и может использоваться только в функции HTSetPenName().</p> <p>Для того, чтобы понять, как действуют эти поля тегов, полезно поместить на экран мастер-объект Historical Trend Wizard и "разобрать" его на части. Это пояснит, каким образом используются поля .Pen1–.Pen8.</p>	
Тип данных	TagID (чтение/запись)	
Допустимые значения	<p>Тип этих полей – TagID (идентификатор тега). Это означает, что данному полю тегов может быть присвоено значение идентификатора тега. Записывать в него название тега нельзя. Присваивать ему следует только значение поля .TagID требуемого тега. Дополнительные сведения можно найти в описании поля .TagID. В целом, тегу типа TagID можно присвоить только значение другого тега этого же типа. Использовать их с тегами других типов нельзя, если только к их названию не добавлено расширение .TagID.</p> <p>Несмотря на атрибут "для чтения/записи", значение этого поля не может быть непосредственно выведено на экран.</p>	
Примеры	<p>Следующий оператор может быть использован для связи нового тега с полем .Pen1 архивного тренда MyHistTrendTag. Нового тега. При его выполнении в скрипте перо Pen1 архивного тренда MyHistTrendTag начнет выводить на дисплей архивированные значения тега MyLoggedTag. Для того чтобы связать тег MyLoggedTag с пером Pen1, название тега следует указать с полем .Pen1, поскольку поле .Pen1 имеет тип TagID, а для присваивания значений необходимо совпадение типов тегов (т.е. запись типа DisctereTag1 = DiscretTag допустима, а запись типа DiscreteTag = MessageTag нет).</p> <p>MyHistTrendTag.Pen1 = MyLoggedTag.TagID;</p> <p>Пусть теперь необходимо отобразить на экране название</p>	

тега, которое только что было записано в **MyHistTrendTag.Pen1**. Это довольно полезная информация для оператора. Предположим далее, что название тега должно отображаться в условных обозначениях архивного тренда. Следует отметить, что вывод значения **MyHistTrendTag.Pen1** анимационной функцией вывода символьных строк ничего не даст. Причина в том, что фактическое значение поля **.Pen1** представляет собой некоторое целое число, которое является адресом ячейки памяти внутри приложения WindowViewer и потому не подходит для непосредственного отображения на экране. Для решения этой задачи следует воспользоваться следующим способом. Сначала надо создать новый тег типа TagID (например с названием **Pen01**), а затем записать под приведённым ранее оператором следующий:

Pen01 = MyHistTrendTag.Pen1;

В первом примере с пером **Pen1** архивного тренда **MyHistTrendTag** был связан тег **MyLoggedTag**. Теперь же значение **Pen1** тренда **MyHistTrendTag** (которое представляет собой идентификатор тега **MyLoggedTag**) записано в тег **Pen01**. Это сделано для того, чтобы получить доступ к полю **.Name**. Отобразить непосредственно значение **MyHistTrendTag.Name** на экране нельзя, потому что в этом случае всегда будет выводиться строка **MyHistTrendTag**. Вывести на экран значение **MyHistTrendTag.Pen1** не разрешит редактор связей, потому что тип **Pen1** – TagID (значение которого на экране не отображается). Необходим какой-то вспомогательный тег, который обеспечит взаимодействие подсистемы отображения и подсистемы архивирования. Присвоение значения **MyHistTrendTag.Pen1** тегу **Pen01** типа TagID фактически означает присвоение тегу **Pen01** строки **MyLoggedTag**. На практике эта же самая задача может быть решена с помощью следующего оператора:

Pen01 = MyLoggedTag.TagID;

(функционально оба оператора идентичны).

Почему же предпочтение отдано первому оператору? Потому что он является более универсальным. Кроме того, запись **MyHistTrendTag.Pen1** может быть использована в качестве названия тега в теле скрипта по изменению данных: **Pen01 = MyHistTrendTag.Pen1**. Таким образом, в **Pen01** можно записать название того же самого тега, который только что был связан с пером **Pen1** архивного тренда, без его определения. Подобное решение позволяет постоянно отображать на экране название тега, значения которого отображаются пером **Pen1** архивного тренда, даже если эта связь каким-либо образом будет в системе изменена (либо средствами пользовательского интерфейса после двойного щелчка кнопкой мыши на тренде в окне WindowViewer, либо с помощью Quick-скрипта InTouch).

Предположим, что в определённый момент времени на

	<p>тренде понадобилось отображать минимальное и максимальное значения регистрируемого тега в единицах измерения. Эта задача прекрасно решена в мастер-средстве Historical Trend Wizard. Поместите его на экран и "разберите" на составные части. Это позволит ознакомиться с методами применения данных полей.</p> <p>Так как теги типа TagID типа Pen01 не имеют полей единиц измерения, равно как и других полей, свойственных обычным тегам InTouch, ими нельзя воспользоваться для вывода значений соответствующих тегов в единицах измерения. Например, запись типа Pen01.MaxEU неверна, поскольку редактор связей не допускает наличия поля .MaxEU у тега типа TagID, каковым является тег Pen01.</p> <p>Для решения этой задачи необходим ещё один вспомогательный тег, который позволит выполнить необходимые преобразования между типом TagID и отображаемой величиной. Сначала необходимо создать тег косвенного аналогового типа (например с названием IndirectAnalogPen1). В теле того же самого скрипта по изменению данных укажите оператор следующего вида:</p> <p>IndirectAnalogPen1.Name = Pen01.Name;</p> <p>Название тега Pen01 (которое в текущий момент должно совпадать с MyLoggedTag) записывается в косвенный тег IndirectAnalogPen1. Это даёт возможность использовать все обычные поля аналоговых тегов (в данном случае тега целого типа с названием MyLoggedTag). Теперь в функции вывода налоговых значений на экран можно записать следующее:</p> <p>IndirectAnalogPen1.MaxEU</p> <p>Функция будет выводить значения того тега в единицах измерения, который в текущий момент связан с косвенным тегом IndirectAnalogPen1. Поскольку в скрипте по изменению данных ранее этому тегу было присвоено значение Pen01.Name, можно быть уверенным, что на экране будет показано максимальное значение в единицах измерения того тега, который связан с пером Pen1 архивного тренда MyHistTrendTag.</p>
См. также	.TagID, HTGetPenName(), HTSetPenName()

.PendingUpdates

	Показывает, завершены ли все обновления окна вывода сведений о распределённых алармах	
Категория	алармы	
Использование	[ErrorMessage =]GetPropertyI("ObjectName.PendingUpdates", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано значение свойства при выполнении функции.
Примечание	Данное поле показывает, завершены ли все обновления данных в окне указанного объекта вывода сведений о распределённых алармах. Положительное значение поля свидетельствует о наличии сведений о новых алармах. Поле сбрасывается в 0 каждый раз при перерисовке объекта.	
Тип данных	Целый (только для чтения)	
Примеры	При выполнении следующего оператора в тег AlarmPendingUpdates целого типа записывается количество незавершённых обновлений объекта вывода сведений о распределённых алармах AlmObj_1 : GetPropertyI("AlmObj_1.PendingUpdates", AlarmPendingUpdates);	
См. также	GetProperty()	

.PrevPage

	При изменении значения этого поля с 1 на 0 выполняется "прокручивание" экрана объекта вывода сведений о распределённых алармов на одну страницу вверх.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyD("ObjectName.PrevPage", Tagname); [ErrorNumber =]SetPropertyD("ObjectName.PrevPage", Value);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название тега, в который будет записано значение свойства в момент выполнения функции.
	<i>Value</i>	Логическое значение либо название логического тега с требуемым логическим значением.
Примечание	При изменении значения поля с 1 на 0 на экран выводится предыдущая страница сведений о распределённых алармах, после чего логическое значение автоматически устанавливается в 1. Если будет достигнуто начало списка страниц, значение поля не изменится и останется равным 0.	
Тип данных	Логический (чтение/запись)	
См. также	GetPropertyD(), SetPropertyD(), .NextPage, .PageNum, .TotalPages	

.PriFrom

	Содержит минимальное значение приоритета, указанного для текущего запроса	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyI("ObjectName.PriFrom", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано значение свойства в момент выполнения функции.
Примечание	Поле "только для чтения", содержащее минимальное значение приоритета, указанного для данного объекта вывода сведений о распределённых алармах в запросе на соответствующие данные.	
Тип данных	Целый (только для чтения)	
Примеры	При выполнении следующего оператора в тег MinPri будет записано минимальное значение приоритета запроса, используемого для заполнения объекта вывода сведений о распределённых алармах AlmObj_1 : GetPropertyI("AlmObj_1.PriFrom", MinPri);	
См. также	GetPropertyI(), .PriTo, .AlarmPri	

.PriTo

	Содержит максимальное значение приоритета, указанного для текущего запроса	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyI("ObjectName.PriTo", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано значение свойства в момент выполнения функции.
Примечание	Поле "только для чтения", содержащее максимальное значение приоритета, указанного для данного объекта вывода сведений о распределённых алармах в запросе на соответствующие данные.	
Тип данных	Целый (только для чтения)	
Примеры	При выполнении следующего оператора в тег MaxPri будет записано максимальное значение приоритета запроса, используемого для заполнения объекта вывода сведений о распределённых алармах AlmObj_1 : GetPropertyI("AlmObj_1.PriTo", MaxPri);	
См. также	GetPropertyI(), .PriFrom, .AlarmPri	

.ProviderReq

	Содержит количество провайдеров алармов, указанных в текущем запросе	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyI("ObjectName.ProviderReq", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано значение свойства в момент выполнения функции.
Примечание	В данном поле содержится количество провайдеров алармов, указанных в текущем запросе на данные, которые используются для заполнения указанного объекта вывода сведений о распределённых алармах.	
Тип данных	Целый (только для чтения)	
Примеры	Следующий оператор возвращает в тег TotalProv общее число провайдеров алармов для текущего запроса на данные для объекта вывода сведений о распределённых алармах AlmObj_1 : GetPropertyI("AlmObj_1.ProviderReq", TotalProv);	
См. также	GetPropertyI(), .ProviderRet	

.ProviderRet

	Содержит количество провайдеров алармов, от которых были успешно получены данные для выполнения текущего запроса	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyI("ObjectName.ProviderRet", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано значение свойства в момент выполнения функции.
Примечание	В данном поле содержится количество провайдеров алармов, от которых были получены данные для текущего запроса для заполнения указанного объекта вывода сведений о распределённых алармах.	
Тип данных	Целый (только для чтения)	
Примеры	Следующий оператор возвращает в тег RetProv число провайдеров алармов, от которых были успешно получены требуемые сведения для выполнения текущего запроса на данные для объекта вывода сведений о распределённых алармах AlmObj_1 : GetPropertyI("AlmObj_1.ProviderRet", RetProv);	
См. также	GetPropertyI(), .ProviderReq	

.Quality

	<p>Для более четкого представления о поле качества (Quality) приведем краткое определение стандарта качества, используемого фирмой Wonderware. Этот стандарт основан на нормативах качества OPC (OLE for Process Control – Технология OLE для управления процессами), которые, в свою очередь, опираются на спецификации Fieldbus Data Quality.</p> <p>Флаг качества отражает состояние качества значения элемента данных. Это позволяет разработчикам легко определять требуемый уровень функциональности при проектировании как серверных и клиентских приложений.</p> <p>Младшие 8 разрядов (наименее значащие разряды) байта флага качества в настоящее время делятся на три битовых поля; поле качества (Quality), поле субстатуса (Substatus) и поле предельного значения (Limit):</p> <p>QQSSSSL</p> <p>Качество получаемого от сервера ввода/вывода значения записано в поле Quality.</p> <p>Примечание. При разрыве соединения с сервером ввода/вывода все битовые поля качества автоматически сбрасываются в исходное нулевое состояние. Флаг .ReferenceComplete при этом также сбрасывается в ноль.</p>		
Категория	теги		
Использование	Tagname.Quality		
	Параметр	Описание	
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа или косвенного аналогового или логического тега.	
Тип данных	Целый (только для чтения)		
Допустимые значения	От 0 до 255		
Примеры	<pre>IF IOTag.Quality <> 192 THEN LogMessage("This data is not Good!"); ENDIF;</pre>		
	Сервер ввода/вывода Wonderware передают своим клиентам шесть взаимоисключающих значений качества:		
			Байты качества OPC
		Биты качества OPC	
	Старший байт	Младший байт	.QualitySubStatus

Качество	Шестнадцатеричное значение	xxxxxxx	QQSSSSL	.QualityLimit		
1. Достоверные	0x00C0	00000000	00000000	Q = 3	S = 0	L = 0
2. Макс. предел	0x0056	00000000	00000000	Q = 1	S = 5	L = 2
3. Мин. предел	0x0055	00000000	00000000	Q = 1	S = 5	L = 1
4. Ошибка преобразования	0x0040	00000000	00000000	Q = 1	S = 0	L = 0
5. Недоступность точки в/в	0x0004	00000000	00000000	Q = 0	S = 1	L = 0
6. Сбой связи	0x0018	00000000	00000000	Q = 0	S = 6	L = 0

Если клиентское приложение не может обмениваться данными с сервером, то .QualityStatus = 0.

	0x0000	00000000	00000000	Q = 0	S = 0	L = 0
--	--------	----------	----------	-------	-------	-------

Далее перечислены ситуации, в которых возможно возвращение указанных значений.

1. Достоверные данные

Линия связи в порядке.

ПЛК принял запрос и возвратил допустимый пакет отклика.

Запись в регистр (если выполнялась) завершилась без ошибок.

Преобразование возвращённых контроллером данных было выполнено без ошибок.

Пример

При опросе регистра, содержащего число 10 (десятичное), было возвращено значение 0x0000A.

2. Данные были зафиксированы на верхней границе

Линия связи в порядке.

ПЛК принял запрос и возвратил допустимый пакет отклика. Чтение регистра или запись в него были выполнены без ошибок.

Фиксация на верхней границе была выполнена в связи с тем, что контролируемое значение превысило некоторый установленный максимум.

Если возвращаемым значением является строка символов, то она усекается.

Пример

Целое без знака будет ограничено сверху числом 65535.

3. Данные были зафиксированы на нижней границе

Линия связи в порядке.

ПЛК принял запрос и возвратил допустимый пакет отклика. Чтение регистра или запись в него были выполнены без ошибок.

Фиксация на нижней границе была выполнена в связи с тем, что контролируемое значение было меньше некоторого установленного минимума.

Пример

Целое без знака будет ограничено снизу нулём.

4. Невозможность преобразования

Линия связи в порядке.

ПЛК принял запрос и возвратил допустимый пакет отклика.

Данные от ПЛК не были преобразованы в требуемый формат. Возможные причины (список не полный):

Неверно заданный формат преобразования данных от ПЛК.

Вместо переменных данных сервер возвратил константу либо только информацию о качестве данных.

Данные бессмысленные.

Неизвестно, слишком велико или слишком мало возвращаемое значение.

Тип возвращаемых контроллером данных некорректен.

Возвращено число с плавающей запятой, которое не является ожидаемым параметром (например NaN – Not A Number).

Пример

Из регистра VCD контроллера прочитано значение 0x000A.

5. Недоступность точки ввода/вывода

Линия связи в порядке.

ПЛК принял запрос и возвратил допустимый пакет отклика.

ПЛК сообщил о недоступности требуемой точки.

Возможные причины (список неполный):

Указанный элемент данных в памяти ПЛК не существует.

Указанный элемент данных в текущий момент времени недоступен (каким-либо образом заблокирован вследствие конкуренции за ресурсы).

Формат или тип элемента данных неправильный;

Была выполнена попытка записи в элемент типа "только для чтения".

В большинстве случаев, если один из элементов группы будет некорректным, то недействительной станет вся группа. Это является следствием использования в серверах механизма группового опроса. Например, если какой-либо элемент в группе из 10 элементов станет недействительным, то ПЛК пометит всю группу как недействительную. При этом сервер будет возвращать "недостоверное" качество для всех элементов группы.

Данные не имеют смысла.

Пример

Была выполнена попытка прочитать значение регистра R4001, который в карте памяти ПЛК не определён.

6. Сбой связи

Любые из следующих причин:
 Прекращение процедур обмена данными.
 Режим обмена с группой данных – медленный опрос (slow poll) или эквивалентный;
 Отсутствие сообщений проверки работоспособности линии связи.
 Недостаток ресурсов сервера (например, резидентная задача или драйвер не могут получить блок памяти требуемого объёма).
 Недостаток ресурсов линии связи.
 Линия связи отключена.
 Все каналы связи заняты.
 Сеть не может передать сообщение в ПЛК.

Пример

Попытка прочитать значение регистра выключенного ПЛК.

См. также

.QualityLimit, .QualityStatus, .QualitySubstatus

.QualityLimit

	Тип предельного значения величины ввода/вывода, передаваемой сервером ввода/вывода при исправном соединении.	
Категория	теги	
Использование	Tagname.QualityLimit	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа или косвенного аналогового или логического тега.
Тип данных	Целый (только для чтения)	
Допустимые значения	Часть (LL)	
	0	Не ограничено
	1	Ограничено снизу
	2	Ограничено сверху
	3	Константа
См. также	.Quality	

.QualityLimitString

	Тип предельного значения величины ввода/вывода, передаваемой сервером ввода/вывода при исправном соединении, в символьном виде.
Категория	теги

Использование	Tagname.QualityLimitString	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа или косвенного аналогового или логического тега.
Тип данных	Сообщение (только для чтения)	
См. также	.QualityLimit, .Quality	

.QualityStatus

	Статус значения ввода/вывода, переданного сервером ввода/вывода при исправном соединении.	
Категория	теги	
Использование	Tagname.QualityStatus	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа или косвенного аналогового или логического тега.
Примечание	Поле субстатуса (QQ) определяется значением байта качества (QQSSSSL)	
Тип данных	Целый (только для чтения)	
Допустимые значения	Часть (QQ)	
	0	Недостоверные данные
	1	Неопределённые данные
	3	Достоверные данные
См. также	.QualitySubStatus, .Quality	

.QualityStatusString

	Статус значения ввода/вывода, переданного сервером ввода/вывода при исправном соединении, в символьном виде	
Категория	теги	
Использование	Tagname.QualityStatusString	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа или косвенного аналогового или логического тега.
Тип данных	Сообщение (только для чтения)	
См. также	.QualityStatus, .QualitySubstatus, .Quality	

.QualitySubstatus

	Субстатус значения ввода/вывода, переданного сервером ввода/вывода при исправном соединении.	
Категория	теги	
Использование	Tagname.QualitySubstatus	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа или косвенного аналогового или логического тега.
Тип данных	Целый (только для чтения)	
Допустимые значения	Части (SSSS) и (QQ)	
	Субстатус SSSS при недостоверных данных (QQ = 0)	
	0	Неконкретная
	1	Ошибка конфигурирования
	2	Не подключено
	3	Отказ устройства
	4	Отказ датчика
	5	Последнее известное значение
	6	Сбой связи
	7	Не работает
	Субстатус SSSS при неопределённых данных (QQ = 1)	
	0	Неконкретная
	1	Последнее использованное значение
	4	Неточность датчика
	5	Выход за пределы допустимых значений
	6	Почти нормальное
	Субстатус SSSS при достоверных данных (QQ = 3)	
	0	Неконкретная
	6	Местная перезапись
См. также	.QualityStatus, .Quality	

.QualitySubstatusString

	Субстатус значения ввода/вывода, переданного сервером ввода/вывода при исправном соединении, в символьном виде	
Категория	теги	
Использование	Tagname.QualitySubstatusString	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа или косвенного аналогового или логического тега.
Тип данных	Сообщение (только для чтения)	
См. также	.QualityStatus, .QualitySubstatus, .Quality	

.QueryState

	Тип фильтра для текущего запроса	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyI("ObjectName.QueryState", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, AlmObj_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано соответствующее значение во время исполнения функции.
Примечание	Данное поле содержит тип фильтра текущего запроса для указанного объекта вывода сведений о распределённых алармах	
Тип данных	Целый (только для чтения)	
Допустимые значения	0 = все алармы, 1 = неподтверждённые алармы, 2 – подтверждённые алармы	
Примеры	При выполнении следующего оператора в тег AlmQueryState целого типа будет записан тип фильтра текущего запроса для вывода сведений о распределённых алармах AlmObj_1 : GetPropertyI("AlmObj_1.QueryState", AlmQueryState);	
См. также	GetPropertyI() , .QueryType	

.QueryType

	Тип текущего запроса на сведения об алармах	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyI("ObjectName.QueryType", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, AlmObj_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано соответствующее значение во время исполнения функции.
Примечание	Данное поле содержит тип текущего запроса для указанного объекта вывода сведений о распределённых алармах	
Тип данных	Целый (только для чтения)	
Допустимые значения	1 = хронология 2 = сводка	
Примеры	При выполнении следующего оператора в тег AlmQueryType целого типа будет записан тип текущего запроса для объекта вывода сведений о распределённых алармах AlmObj_1 : GetPropertyI("AlmObj_1.QueryType", AlmQueryType);	
См. также	GetPropertyI(), .QueryState	

.RawValue

	Фактическое значение, полученное от сервера ввода /вывода приложением WindowViewer как клиентом. Данное поле хранит необработанное значение тега I/O, которое преобразуется системой InTouch путём масштабирования	
Категория	теги	
Использование	TagName.RawValue	
	Параметр	Описание
	<i>TagName</i>	Название любого логического тега для ввода/вывода, косвенного логического тега, тега целого типа для ввода/вывода, внутреннего вещественного тега, косвенного аналогового тега, тега типа "сообщение" для ввода/вывода либо косвенного тега типа "сообщение".
Примечание	Данное поле "только для чтения" хранит исходное значение тега для ввода/вывода, которое система InTouch преобразует путём масштабирования.	
Тип данных	Любой соответствующий тип данных. Например, вещественный для вещественных тегов, логический для логических тегов и т.д. (только для чтения)	
Допустимые значения	Логические и аналоговые (соответственно типу)	
Примеры	<p>Следующий оператор позволяет определить, находятся ли значения тега в пределах нормальных рабочих значений.</p> <pre>IF ((IOTag.RawValue > IOTag.MaxRaw) OR (IOTag.RawValue < IOTag.MinRaw)) THEN AlarmMessage = "Датчик требует калибровки или замены."; ENDIF;</pre>	
См. также	.EngUnits, .MinEU, .MaxEU, .MinRaw, .MaxRaw	

.ReadOnly

	Указывает, является ли содержимое текстового поля значением "только для чтения" или значением "для чтения/записи"	
Категория	управление окнами	
Использование	[ErrorNumber =]GetPropertyD("ControlName.ReadOnly", Tagname);	
	Параметр	Описание
	<i>ControlName</i>	Название оконного объекта, например TextBox_1 .
	<i>Tagname</i>	Название логического тега, которому будет присвоено значение свойства при выполнении функции.
Примечание	Данное свойство доступно как во время разработки, так и во время исполнения программы.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = содержимое текстового поля можно изменять, 1 = тип текстового поля – "только для чтения"	
Объекты применения	Текстовые поля	
Примеры	Следующий оператор проверяет, является ли поле TextBox_1 полем "только для чтения": GetPropertyD("TextBox_1.ReadOnly", A_Tagname);	
См. также	GetPropertyD(), SetPropertyD()	

.Reference

	Предназначено для оперативного изменения канала доступа и названия элемента данных во время исполнения программы	
Категория	теги	
Использование	Tagname.Reference	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега для ввода/вывода или косвенного тега, связанного с тегом ввода/вывода.
Примечание	Данное поле предназначено для оперативного изменения канала доступа и названия элемента данных во время исполнения программы	
Тип данных	Сообщение (чтение/запись)	
Допустимые значения	Любая строка символов с названием канала доступа и/или элемента данных	
Примеры	<p>Следующий оператор записывает в поле Item тега для ввода/вывода целого типа название элемента данных, сформированное текстовой функцией и определяемое значением внутреннего тега целого типа. Этот оператор мог бы использоваться в условном скрипте для последовательного опроса набора тегов в зависимости от значения поля .ReferenceComplete:</p> <pre> MyTag.Reference = "R" + Text(MyIndex, "#"); {Если MyIndex = 40001, то значением ИТЕМ будет R40001} ExcelTag.Reference = "R"+Text(RowNum, "#")+Text(ColNum, "#"); </pre>	

.ReferenceComplete

	Возвращает подтверждение, когда изменённое, получившее новое значение и обновлённое поле .Reference имеет такое же значение, как и поле .Value этого же тега.	
Категория	теги	
Использование	Tagname.ReferenceComplete	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега для ввода/вывода или косвенного тега, связанного с тегом ввода/вывода.
Примечание	<p>Данное поле обычно используется в сочетании с полем .Reference. Оно показывает, что канал доступа и/или элемент данных были изменены, и, что более важно, что из нового источника данных было получено новое значение. Даже если величина из нового источника данных будет совпадать со значением из предшествующего источника, после первой же операции обновления система установит поле .ReferenceComplete в 1.</p> <p>В течение обновления, от момента изменения поля .Reference до поступления первой величины из нового источника данных поле .ReferenceComplete будет иметь значение 0, установленное системой.</p>	
Тип данных	Логическое (только для чтения)	

.ROCPct

	Используется для определения и установки предельного значения скорости изменения для проверки возникновения соответствующего аларма	
Категория	алармы	
Использование	Tagname.ROCPct	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Значение данного поля выражается в процентах и совпадает с величиной, указанной в панели алармов окна определения тега в Словаре тегов приложения. Это значение можно отображать на экране и использовать в скриптах. Кроме того, его можно менять в ходе исполнения программы в зависимости от прикладных требований контролируемого технологического процесса.	
Тип данных	Целый (чтение/запись)	
Допустимые значения	от 0 до 100 (%)	
Примеры	Следующий оператор устанавливает свойство Rate-of-Change (предельное значение скорости изменения) тега MyTag в 25%: MyTag.ROCPct = 25;	
См. также	.ROCStatus, .ROCSet	

.ROCSet

	Возвращает значение 1 (истина) либо 0 (ложь) в зависимости от того, было ли установлено предельное значение скорости изменения значений указанного аналогового тега или нет.	
Категория	алармы	
Использование	TagName. ROCSet	
	Параметр	Описание
	<i>TagName</i>	Название любого аналогового тега.
Примечание	Данное поле может использоваться только с тегами целого или вещественного типа.	
Тип данных	Логический	
Допустимые значения	1 (истина) или 0 (ложь)	
Примеры	Следующий оператор может использоваться в анимационной функции вывода аналоговых значений для определения того, было установлено ли предельное значение скорости изменения для тега MyTag : MyTag.ROCSet; { "0" возвращается, если не установлено, "1" если установлено }	
См. также	.Alarm, .Ack, .LoLimit, .LoLoLimit, .HiHiLimit, .HiLimit, .HiSet, .LoSet, .LoLoSet, .Histatus, .HiHistatus, .ROCPct, .ROCStatus	

.ROCStatus

	Указывает, существует ли аларм превышения предельной скорости изменения значений для указанного тега	
Категория	алармы	
Использование	Tagname. ROCStatus	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега целого или вещественного типа или косвенного аналогового тега.
Примечание	Как правило, значение этого поля "только для чтения" равно 0. При возникновении аларма по превышению предельной скорости изменения система устанавливает значение этого поля в 1. Оно сохраняется всё время, пока существует данное состояние аларма. Поле .ROCStatus обычно используется в сочетании с полями .Alarm и .Ack для определения точного состояния аларма конкретного тега в системе.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = Указанное состояние аларма не наступило 1 = Указанное состояние аларма наступило	
Примеры	<p>Оператор в теле оператора IF-THEN будет выполняться только тогда, когда значение поля</p> <p>.ROCStatus тега MyTag равно 1 (существует аларм превышения предельной скорости изменения значений тега):</p> <pre>IF (MyTag. ROCStatus == 1) THEN OperatorMessage = "MyTag has gone into Rate-of-Change Alarm"; ENDIF;</pre>	
См. также	.ROCPct , .ROCSet	

.ScooterLockLeft

	Установка значения этого поля в 1 (Истина) не разрешает правому визиру графика перемещаться левее позиции левого визира.	
Категория	архивирование	
Использование	Tagname.ScooterLockLeft	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега типа "архивный тренд" (Hist Trend).
Примечание	Как правило, перемещение правого визира графика левее позиции левого визира нежелательно. Если левый визир не зафиксирован, то при попытке установить правый визир левее левого их позиции будут совмещены и передвинуты в указанное положение (левый визир будет "захвачен" правым).	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = правый визир <i>может</i> быть установлен левее позиции левого визира, 1 = правый визир <i>не может</i> быть установлен левее позиции левого визира	
Примеры	После выполнения следующего оператора правый визир невозможно будет передвинуть левее позиции левого визира графика тренда MyHistTrendTag. MyHistTrendTag.ScooterLockLeft = 1;	
См. также	.ScooterPosRight, .ScooterPosLeft, .ScooterLockRight	

.ScooterLockRight

	Установка значения этого поля в 1 (Истина) не разрешает левому визирю графика перемещаться правее позиции правого визира.	
Категория	архивирование	
Использование	Tagname.ScooterLockRight	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега типа "архивный тренд" (Hist Trend).
Примечание	Как правило, перемещение левого визира графика правее позиции правого визира нежелательно. Если правый визир не зафиксирован, то при попытке установить левый визир правее правого их позиции будут совмещены и передвинуты в указанное положение (правый визир будет "захвачен" левым).	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = левый визир <i>может</i> быть установлен правее позиции правого визира, 1 = левый визир <i>не может</i> быть установлен правее позиции правого визира	
Примеры	После выполнения следующего оператора левый визир невозможно будет передвинуть правее позиции правого визира графика тренда MyHistTrendTag. MyHistTrendTag.ScooterLockRight = 1;	
См. также	.ScooterPosRight, .ScooterPosLeft, .ScooterLockLeft	

.ScooterPosLeft

	Считывание и установка позиции левого визира	
Категория	архивы	
Использование	Tagname.ScooterPosLeft	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега типа "архивный тренд" (Hist Trend).
Примечание	<p>Данное поле с атрибутом "для чтения/записи" позволяет оперативно управлять позицией левого визира графика. Поле можно указывать в качестве аргумента функций в скриптах, чтобы получить текущую позицию левого визира, а также присваивать ему новое значение для установки левого визира в другое положение.</p> <p>Поле .ScooterPosLeft чаще всего используется в сочетании с функцией HTGetValue(). В аргументах функции указывается требуемый тренд и текущая позиция визиров.</p>	
Тип данных	Вещественный (чтение/запись)	
Допустимые значения	От 0.0 до 1.0, 0 соответствует крайней левой позиции, а 1 – крайней правой позиции графика.	
Примеры	<p>Следующий оператор перемещает левый визир в новое положение. Левый визир будет установлен в точку, которая находится правее левой границы тренда на расстоянии, равном 34 % от общей протяжённости графика тренда MyHistTrendTag:</p> <p>MyHistTrendTag.ScooterPosLeft = .34;</p> <p>В следующем выражении функция HTGetValueAtScooter() используется для извлечения значения Pen1 в текущей позиции левого визира. Поскольку изменение любого значения в списке аргументов функции вызывает её повторное вычисление, значение этого выражения будет меняться каждый раз при изменении позиции левого визира.</p> <pre>MyRealTag = HTGetValueAtScooter(MyHistTrendTag, MyHistTrendTag.UpdateCount, 1, MyHistTrendTag.ScooterPosLeft, 1, "PenValue");</pre>	
См. также	.ScooterPosRight, .ScooterLockLeft, .ScooterLockRight	

.ScooterPosRight

	Считывание и установка позиции правого визира	
Категория	архивы	
Использование	Tagname.ScooterPosLeft	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега типа "архивный тренд" (Hist Trend).
Примечание	<p>Данное поле с атрибутом "для чтения/записи" позволяет оперативно управлять позицией правого визира графика. Поле можно указывать в качестве аргумента функций в скриптах, чтобы получить текущую позицию правого визира, а также присваивать ему новое значение для установки правого визира в другое положение.</p> <p>Поле .ScooterPosRight чаще всего используется в сочетании с функцией HTGetValue(). В аргументах функции указывается требуемый тренд и текущая позиция визиров.</p>	
Тип данных	Вещественный (чтение/запись)	
Допустимые значения	От 0.0 до 1.0, 0 соответствует крайней левой позиции, а 1 – крайней правой позиции графика.	
Примеры	<p>Следующий оператор перемещает правый визир в новое положение. Правый визир будет установлен в точку, которая находится левее правой границы тренда на расстоянии, равном 34 % от общей протяжённости графика тренда MyHistTrendTag:</p> <p>MyHistTrendTag.ScooterPosRight = .34;</p> <p>В следующем выражении функция HTGetValueAtScooter() используется для извлечения значения Pen1 в текущей позиции правого визира. Поскольку изменение любого значения в списке аргументов функции вызывает её повторное вычисление, значение этого выражения будет меняться каждый раз при изменении позиции правого визира.</p> <p>MyRealTag = HTGetValueAtScooter(MyHistTrendTag, MyHistTrendTag.UpdateCount, 2, MyHistTrendTag.ScooterPosRight, 1, "PenValue");</p>	
См. также	.ScooterPosLeft, .ScooterLockLeft, .ScooterLockRight	

.Successful

	Указывает, успешно ли был выполнен последний запрос	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyD("ObjectName.Successful", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название любого объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название логического тега, в который будет записано значение данного свойства во время исполнения функции.
Примечание	Данное свойство индицирует успех выполнения последнего запроса на данные для указанного объекта вывода сведений о распределённых алармах.	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = ошибка при выполнении запроса, 1 = успешное выполнение запроса	
Примеры	При выполнении следующего оператора значение логического тега AlmFlag будет установлено в зависимости от успеха выполнения запроса на данные для объекта вывода сведений о распределённых алармах AlmObj_1 : GetPropertyD("AlmObj_1.Successful", AlmFlag);	
См. также	GetPropertyD()	

.SuppressRetain

	Определение и установка статуса сохранения окна вывода сведений о распределённых алармах.	
Категория	алармы	
Использование	<pre>[ErrorNumber =]GetPropertyD("ObjectName.SuppressRetain", Tagname); [ErrorNumber =]SetPropertyD("ObjectName.SuppressRetain", Tagname);</pre>	
	Параметр	Описание
	<i>ObjectName</i>	Название любого объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	Tagname	Название логического тега, в который будет записано значение данного свойства во время исполнения функции.
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = сохранение отменено 1 = сохранение разрешено	
Примеры	Следующий оператор возвращает в логический тег SupRtn статус сохранения объекта вывода сведений о распределённых алармах AlmObj_1 : SetPropertyD("AlmObj_1.SuppressRetain", SupRtn);	
См. также	GetPropertyD(), SetProperty()	

.TagID

	Используется совместно с полями .Pen1–.Pen8 архивного тренда для управления тегами, значения которых выводятся соответствующим пером.	
Категория	теги	
Использование	TagName.TagID	
	Параметр	Описание
	<i>TagName</i>	Название любого тега логического, целого или вещественного типа или косвенного логического или аналогового тега.
Примечание	Поле .TagID предоставляет собой указатель на тег и используется главным образом для связи тегов с перьями архивного тренда.	
Тип данных	TagID (только чтение)	
Примеры	MyHistTrendTag.Pen6 = SomeAnalogTag.TagID;	
См. также	.Pen1–.Pen8	

.TimeDate

	Целочисленное поле, содержащее целую часть количества дней, прошедших с момента получения значения от сервера ввода/вывода при исправном соединении.	
Категория	теги	
Использование	TagName.TimeDate	
	Параметр	Описание
	<i>TagName</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Целый (только для чтения)	
См. также	.TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear	

.TimeDateString

	Строка с датой в формате, указанном в Реестре операционной системы Windows.	
Категория	теги	
Использование	Tagname.TimeDateString	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Сообщение (только чтение)	
См. также	.TimeDate, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear	

.TimeDateTime

	Вещественное поле, содержащее дробную часть количества дней, прошедших с момента получения значения от сервера ввода/вывода при исправном соединении.	
Категория	теги	
Использование	Tagname.TimeDateTime	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Вещественный (только для чтения)	
См. также	.TimeDate, .TimeDateString, .TimeDay, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear	

.TimeDay

	Целочисленное поле, хранящее день месяца, в который от сервера ввода/вывода было получено значение ввода/вывода при исправном соединении.	
Категория	теги	
Использование	TagName.TimeDay	
	Параметр	Описание
	<i>TagName</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Целый (только для чтения)	
Допустимые значения	от 1 до 31	
См. также	.TimeDate, .TimeDateString, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear	

.TimeHour

	Целочисленное поле, хранящее показания часа дня, в который от сервера ввода/вывода было получено значение ввода/вывода при исправном соединении.	
Категория	теги	
Использование	TagName.TimeHour	
	Параметр	Описание
	<i>TagName</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Целый (только для чтения)	
Допустимые значения	от 0 до 23	
См. также	.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear	

.TimeMinute

	Целочисленное поле, хранящее показания минуты часа дня, в который от сервера ввода/вывода было получено значение ввода/вывода при исправном соединении.	
Категория	теги	
Использование	Tagname.TimeMinute	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Целый (только для чтения)	
Допустимые значения	от 0 до 59	
См. также	.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear	

.TimeMonth

	Целочисленное поле, хранящее номер месяца, в который от сервера ввода/вывода было получено значение ввода/вывода при исправном соединении.	
Категория	теги	
Использование	Tagname.TimeMonth	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Целый (только для чтения)	
Допустимые значения	от 1 до 12	
См. также	.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear	

.TimeMsec

	Целочисленное поле, хранящее номер миллисекунды, когда от сервера ввода/вывода было получено значение ввода/вывода при исправном соединении.	
Категория	теги	
Использование	TagName.TimeMsec	
	Параметр	Описание
	<i>TagName</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Целый (только для чтения)	
Допустимые значения	от 0 до 999	
См. также	.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear	

.TimeSecond

	Целочисленное поле, хранящее номер секунды, когда от сервера ввода/вывода было получено значение ввода/вывода при исправном соединении.	
Категория	теги	
Использование	TagName.TimeSecond	
	Параметр	Описание
	<i>TagName</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Целый (только для чтения)	
Допустимые значения	от 0 до 59	
См. также	.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeTime, .TimeTimeString, .TimeYear	

.TimeTime

	Целочисленное поле, хранящее число миллисекунд, прошедших с полуночи того дня, когда от сервера ввода/вывода было получено значение ввода/вывода при исправном соединении.	
Категория	теги	
Использование	Tagname.TimeTime	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Целый (только для чтения)	
Допустимые значения	от 0 до 86399999	
См. также	.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTimeString, .TimeYear	

.TimeTimeString

	Поле, хранящее в символьном виде показания даты и времени суток, когда от сервера ввода/вывода было получено значение ввода/вывода при исправном соединении.	
Категория	теги	
Использование	Tagname.TimeTimeString	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Сообщение (только для чтения)	
См. также	.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeYear	

.TimeYear

	Целочисленное поле, хранящее год (четыре цифры), в который от сервера ввода/вывода было получено значение ввода/вывода при исправном соединении.	
Категория	теги	
Использование	Tagname.TimeYear	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега логического, целого или вещественного типа, или косвенного аналогового тега, или косвенного теги типа "сообщение".
Тип данных	Целый (только для чтения)	
Допустимые значения	Любой год в формате ####, например 2004.	
См. также	.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString	

.TopIndex

	Возвращает начальный порядковый номер самого первого элемента в обычном списке.	
Категория	управление окнами	
Использование	<pre>[ErrorNumber =]GetPropertyI("ControlName.TopIndex", Tagname); [ErrorNumber =]SetPropertyI("ControlName.Caption", Number);</pre>	
	Параметр	Описание
	<i>ControlName</i>	Название объекта в окне, например ListBox_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано значение данного свойства во время исполнения функции.
	<i>Number</i>	Целое число, определяющее порядковый номер самого первого элемента списка.
Примечание	Данное свойство доступно только во время исполнения программы.	
Тип данных	Целый (чтение/запись)	
Объекты применения	Списки	
Примеры	При выполнении следующего оператора нумерация элементов списка ListBox_1 начинается с 14: SetPropertyI("ListBox_1.TopIndex", 14);	
См. также	GetPropertyM(), SetPropertyM(), .ListIndex, .NewIndex	

.TotalPages

	Возвращает число страниц (заполненных данными об алармах) в объекте вывода сведений о распределённых алармах.	
Категория	алармы	
Использование	[ErrorNumber =]GetPropertyI("ObjectName.TotalPages", Tagname);	
	Параметр	Описание
	<i>ObjectName</i>	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	<i>Tagname</i>	Название тега целого типа, в который будет записано значение данного свойства во время исполнения функции.
Примечание	Данное свойство только для чтения содержит число страниц с данными об алармах в указанном объекте вывода сведений о распределённых алармах.	
Тип данных	Целый (чтение/запись)	
Примеры	Следующий оператор возвращает в тег целого типа AlmTotalPage число страниц, содержащихся в объекте вывода сведений о распределённых алармах AlmObj_1 : GetPropertyI("AlmObj_1.TotalPages", AlmTotalPages);	
См. также	GetPropertyI(), .NextPage, .PrevPage, .PageNum	

.UnAck

	Управляет состоянием подтверждения локальных алармов всех типов	
Категория	алармы	
Использование	Tagname.UnAck = 0	
	Параметр	Описание
	Tagname	Название любого тега логического, целого или вещественного типа, косвенного логического тега либо группы алармов.
Примечание	Установите это поле в 0, чтобы подтвердить все неподтверждённые алармы, связанные с указанной группой алармов или тегом. Если указанное название является названием группы алармов, будут подтверждены все неподтверждённые алармы, связанные с тегами указанной группы. Если же указан тег любого другого типа, то подтверждены будут только неподтверждённые алармы, связанные с этим тегом. Установка поля в значение, отличное от 0, не имеет смысла, и результат будет неопределённым.	
Тип данных	Логический (только чтение/сброс)	
Допустимые значения	0	
Примеры	<p>Следующий оператор подтверждает все логические алармы, связанные с тегом Tag1.</p> <p>Tag1.UnAck = 0;</p> <p>В следующем примере подтверждаются все неподтверждённые алармы тегов в группе алармов PumpStation:</p> <p>PumpStation.UnAck = 0;</p> <p>Примечание. Для поля .UnAck существует поле противоположного назначения .Ack. После подтверждения аларма поле .Ack устанавливается в 1.</p>	
См. также	.Ack, Ack(), .Alarm, .AlarmAckModel	

.UpdateCount

	Увеличивается на единицу при каждом обновлении связанного с этим полем архивного тренда.	
Категория	архивирование	
Использование	Tagname.UpdateCount	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега типа "архивный тренд" (Hist Trend).
Примечание	Данное поле напрямую связано с подсистемой поиска данных для архивных трендов. Как только для указанного архивного тренда запрашиваются новые данные, подсистема поиска обращается к диску и считывает требуемые сведения. По окончании процесса считывания данных значение поля .UpdateCount увеличивается на 1. Поле .UpdateCount используется во многих функциях, связанных с архивными трендами, как способ инициализации пересчета значений этих функций.	
Тип данных	Целый (только для чтения)	
Допустимые значения	Любое положительное число	
Примеры	Функция HTGetValueAtScooter() в следующем операторе используется для считывания значения Pen1 в текущей позиции правого визира. Поскольку изменение любого аргумента функции вызывает повторное вычисление её значения, каждое считывание (обновление) данных и увеличение значения поля .UpdateCount приведёт к повторному выполнению указанного оператора. MyRealTag = HTGetValueAtScooter(MyHistTrendTag, MyHistTrendTag.UpdateCount, 2, MyHistTrendTag.ScooterPosRight, 1, "PenValue");	
См. также	.UpdateInProgress, .UpdateTrend	

.UpdateInProgress

	Равно 1, если процесс поиска данных ещё не завершился, в противном случае равно 0.	
Категория	архивирование	
Использование	Tagname.UpdateInProgress	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега типа "архивный тренд" (Hist Trend).
Примечание	<p>Данное поле напрямую связано с подсистемой поиска данных для архивных трендов. Как только для указанного архивного тренда запрашиваются новые данные, подсистема поиска обращается к диску и считывает требуемые сведения. В время считывания данных значение этого поля равно 1, по завершении этого процесса оно сбрасывается в 0. Поле .UpdateInProgress используется во многих функциях, связанных с архивными трендами.</p> <p>Большинство экранов архивных трендов имеют элементы управления, с помощью которых может "прокручивать" выводимое изображение. Во время манипуляций с этими элементами подсистема архивирования должна гарантировать появление на экране соответствующих актуальных данных. Если на экран должны быть показаны сведения для интервала времени, который ещё не отображается или не находится в резидентной памяти, то подсистема обращается к диску и считывает требуемую информацию. Поскольку этот процесс может быть довольно длительным, система предоставляет проектировщику экрана архивных трендов средство уведомления оператора о выполнении операции поиска данных. Без такого рода обратной связи оператор не может быть уверенным, что система выполняет затребованное действие.</p>	
Тип данных	Логический (только для чтения)	
Допустимые значения	0 = процесс обновления не выполняется, 1 = процесс обновления ещё не завершён	
Примеры	<p>Следующее выражение часто используется для отображения надписи в текстовом объекте около или на кнопке прокручивания архивного тренда. Когда архивная подсистема считывает запрошенные данные, результатом этого выражения будет значение 1. Если тренд не требует обновления, то выражение будет иметь значение 0.</p> <p>MyHistTrendTag.UpdateInProgress</p>	
См. также	.UpdateCount, .UpdateTrend	

.UpdateTrend

	Используется для обновления архивного тренда актуальными данными	
Категория	архивирование	
Использование	Tagname.UpdateTrend	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега типа "архивный тренд" (Hist Trend).
Примечание	<p>Архивные тренды автоматически не обновляются. Для этого необходимо изменить значение либо поля .ChartStart, либо поля .ChartLength, чтобы на экран были выведены текущие значения указанного тега. Данное поле можно использовать в скрипте, запускаемом при нажатии на экранную кнопку, для того чтобы оператор в любой момент времени мог обновлять тренд текущими значениями тегов.</p> <p>Поле .UpdateTrend может также использоваться в случаях, когда меняться будут другие поля архивных трендов. Это гарантирует, что на экране будут отображены самые свежие данные.</p> <p>Установка данного поля в значение, отличное от 1, не имеет смысла и может привести к неопределённым результатам.</p>	
Тип данных	Логический (только запись)	
Допустимые значения	1	
Примеры	<p>Выполнение следующего оператора приводит к обновлению архивного тренда MyHistTrendTag актуальными данными согласно текущим параметрам:</p> <p>MyHistTrendTag.UpdateTrend = 1;</p>	

.Value

	Содержит текущее значение указанного тега. Является полем по умолчанию каждого тега в системе InTouch. Если после названия тега никакое поле не указано, подразумевается поле .Value.	
Категория	теги	
Использование	Tagname.Value	
	Параметр	Описание
	<i>Tagname</i>	Название любого тега за исключением тегов типа "архивный тренд" (Hist Trend).
Примечание	Данное поле является полем по умолчанию каждого тега в системе InTouch. Если после названия тега никакое поле не указано, подразумевается поле .Value. Указание этого поля не обязательно, однако его использование делает запись вычислений более понятной и потому рекомендуется в целях документирования.	
Тип данных	Зависит (т.е. является таким же) от типа соответствующего тега. Поле для чтения/записи.	
Примеры	Следующий оператор устанавливает значение внутреннего тега целого типа MyTag равным 100: Tagname.Value = 100; Функционально данный оператор идентичен следующему: Tagname = 100;	
См. также		

.Value

	Свойство по умолчанию всех управляющих мастер-объектов в окнах приложений InTouch. Изменения этого свойства отражаются в свойствах соответствующих тегов и мастер-объектов.	
Категория	управление окнами	
Использование	<pre>[ErrorNumber =]GetPropertyM("ControlName[.Value]", Tagname);</pre> <pre>[ErrorNumber =]SetPropertyM("ControlName[.Value]", Value);</pre> <pre>[ErrorNumber =]GetPropertyI("ControlName[.Value]", Tagname);</pre> <pre>[ErrorNumber =]SetPropertyI("ControlName[.Value]", Value);</pre> <pre>[ErrorNumber =]GetPropertyD("ControlName[.Value]", Tagname);</pre> <pre>[ErrorNumber =]SetPropertyD("ControlName[.Value]", Value);</pre> <p>Примечание. Начальное значение тегов, связанных с простыми или развёртываемыми списками, не может использоваться для определения начального содержания этих списков.</p>	
	Параметр	Описание
	<i>ControlName</i>	Название объекта в окне, например CheckBox_4 .
	<i>Tagname</i>	Название тега, содержащего (целое) число элементов списка.
	<i>[.Value]</i>	Данный аргумент является необязательным. Если он не указан, то всегда предполагается поле .Value .
	<i>Value</i>	Фактическая величина либо название тега InTouch (совпадающего по типу со свойством, которому присваивается значение), значение которых будет записано в свойство во время выполнения функции.
Примечание	Данное свойство для чтения/записи доступно как во время разработки, так и во время исполнения прикладной программы. Если доступ к этому свойству простых и развёртываемых списков выполняется с указанием тега, то оно является полем только для чтения. Для объектов типа "окошко выбора", переключатель или текстовое поле оно является полем для чтения/записи. Указанная во время проектирования величина является начальным значением этого свойства во время исполнения программы.	

Тип данных	Сообщение (чтение/запись) для текстовых полей, простых и развёртываемых списков. Целый (чтение/запись) для переключателей. Логический (чтение/запись) для окошек выбора.
Примеры	Следующий оператор присваивает свойству Value переключателя с названием RadioButton_1 значение 4: SetPropertyI("RadioButton_1.Value", 4);
См. также	GetPropertyM(), SetPropertyM(), GetPropertyI(), SetPropertyI(), GetPropertyD(), SetPropertyD()

.Visible

	Определяет, является ли соответствующий объект окна видимым или нет	
Категория	управление окнами	
Использование	[ErrorNumber =]GetPropertyD("ControlName.Visible", Tagname); [ErrorNumber =]SetPropertyD("ControlName.Visible", Number);	
	Параметр	Описание
	<i>ControlName</i>	Название объекта в окне, например ListBox_4 .
	<i>Tagname</i>	Название тега (того же типа), в который будет записано значение свойства при выполнении функции.
	<i>Number</i>	Логическое значение либо название логического тега, значение которого будет присвоено свойству во время выполнения функции.
Примечание	Данное свойство для чтения/записи доступно как во время разработки, так и во время исполнения прикладной программы.	
Тип данных	Логический (чтение/запись)	
Допустимые значения	0 = объект невидим, 1 = объект видим на экране.	
Объекты применения	Текстовые поля, простые и развёртываемые списки, окошки выбора и переключатели.	
Примеры	Следующий оператор делает текстовое поле TextBox_1 невидимым: SetPropertyD("TextBox_1.Visible", 0);	
См. также	GetPropertyD(), SetPropertyD()	

Г Л А В А 3

QuickScript-функции InTouch

Скрипты – это один из наиболее мощных инструментов приложения InTouch. Они представляют собой средство выполнения тех или иных команд и логических операторов в зависимости от выполнения определённых условий типа нажатия на клавишу, открытия окна, изменения значения и т.д.

Quick-функции – это скрипты, которые можно вызывать из других скриптов или выражений анимационных связей. Многократно используемый код хранится в одном скрипте и в одном месте, что позволяет при необходимости обновлять все экземпляры скрипта в течение одного сеанса редактирования.

Примечание. Для обновления анимационной характеристики в число аргументов соответствующей Quick-функции должен входить "запускающий" тег. Например, если в число аргументов функции будет входить системный тег \$Second, то выражение для анимационной характеристики будет вычисляться при каждом изменении его значения, в результате чего эта Quick-функция будет запускаться каждую секунду.

На основе скриптов можно разрабатывать самые разные специализированные и автоматические функции.

Abs()

	Возвращает абсолютное значение (число без знака) указанной величины	
Категория	математическая функция	
Синтаксис	Result=Abs(Number);	
	Аргумент	Описание
	Number	Любое число или название тега целого или вещественного типа.
Примечание	В тег Result записывается значение модуля аргумента Number .	
Примеры	Abs(14) возвращает 14, Abs(-7.5) возвращает 7.5.	

Аск()

	Подтверждение всех неподтверждённых алармов InTouch	
Категория	алармы	
Синтаксис	Ack Tagname;	
	Аргумент	Описание
	Tagname	Название любого тега InTouch, группы алармов или групповой переменной.
Примечание	Эта функция может применяться тэгу или Группе алармов.	
Примеры	<p>Следующие операторы обеспечивают подтверждение всех неподтверждённых алармов:</p> <p>Ack \$\$System; (подтверждение всех алармов в системе)</p> <p>Ack Tagname;</p> <p>Ack GroupName;</p> <p>Ack GroupVariable;</p>	
См. также	almAckAll(), almAckGroup() almAckTag(), almAckDisplay(), almAckRecent(), almAckPriority(). almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()	

ActivateApp()

	Активизация указанного запущенного приложения	
Категория	система	
Синтаксис	ActivateApp TaskName;	
	Аргумент	Описание
	TaskName	Название активизируемой задачи
Примечание	Аргумент TaskName должен содержать название задачи в таком же виде, в каком оно выводится в Панели задач либо окне Диспетчера задач (вызывается в системе Windows NT после щелчка правой кнопкой мыши на Панели задач и запуска Диспетчера либо после нажатия клавиш Ctrl+Alt+Del).	
Примеры	<p>Следующий оператор проверяет, исполняется ли интерпретатор командной строки. Если да, то система переключается в его окно, в противном случае система сначала запускает интерпретатор командной строки, а затем программу DOS edit.com из интерпретатора.</p> <pre>IF InfoAppActive(InfoAppTitle("cmd")) == 1 THEN ActiveApp InfoAppTitle("cmd"); ELSE StartApp "cmd /c edit"; ENDIF;</pre>	
См. также	StartApp(), InfoAppTitle()	

AddPermission()

	<p>Присваивает соответствующий уровень доступа InTouch заданной группе пользователей в логической системе или домене. Когда пользователь, принадлежащий этой группе, регистрируется в InTouch после того, как была вызвана AddPermission, он получит указанный уровень доступа.</p>	
Категория	контроль доступа	
Синтаксис	DiscreteTag=AddPermission("Domain", "Group", AccessLevel);	
	Аргумент	Описание
	Domain	Название домена или локального компьютера, в котором определена указанная группа.
	Group	Название группы пользователей Windows.
	AccessLevel	Уровень доступа в системе InTouch для данной группы.
Примечание	<p>Применима исключительно для режима безопасности операционной системы. Когда эта функция вызывается, она проверяет присутствие в указанном домене заданной Группы. В случае успешного выполнения возвращается TRUE (истина) и the указанный УровеньДоступа связывается с Группой во внутренних записях InTouch для использования во время последующих регистраций пользователей. Во всех других случаях, то есть, если для любого из параметров указано недопустимое значение, возвращается FALSE (ложь). Эта функция функция обычно запускается при запуске системы. Только пользователи, регистрируемые ПОСЛЕ успешного вызова AddPermission, получают уровень доступа доступа, присвоенный его группе с помощью вызова функции AddPermission.</p>	
Примеры	<p>DiscreteTag=AddPermission("wonderware_hq", "InTouchAdmins", 9000); DiscreteTag=AddPermission("ravih01", "InTouchUsers", 5000);</p>	
См. также	<p>PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), AttemptInvisibleLogon(), QueryGroupMembership()</p>	

almAckAll()

	Подтверждает все алармы, найденные при выполнении текущего запроса, включая те, записи о которых не отображаются в текущий момент на экране сводных сведений об алармах.	
Категория	алармы	
Синтаксис	[Result=]almAckAll(ObjectName, Comment);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	Comment	Комментарий, вводимый при подтверждении аларма.
Примечание	Список возвращаемых кодов ошибок см. в Приложении А, "QuickScript-функции подтверждения."	
Примеры	MessageTag = "Acknowledge All by" + \$Operator; almAckAll("AlmObj_1", MessageTag);	
См. также	Ack(), almAckGroup(), almAckTag(), almAckDisplay(), almAckRecent(), almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()	

almAckDisplay()

	Подтверждает только те алармы, записи о которых видимы в текущий момент в окне сводки алармов	
Категория	алармы	
Синтаксис	[Result=]almAckDisplay(ObjectName, Comment);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	Comment	Комментарий, вводимый при подтверждении аларма.
Примечание	Список возвращаемых кодов ошибок можно найти в Приложении А.	
Примеры	almAckDisplay("AlmObj_1", "Display Acknowledgement");	
См. также	Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckRecent(), almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()	

almAckGroup()

	Подтверждает в указанном объекте вывода сведений о распределённых алармах все найденные во время выполнения последнего запроса алармы, названия провайдера и группы алармов которых совпадают с указанными.	
Категория	алармы	
Синтаксис	[Result=]almAckGroup(ObjectName, ApplicationName, GroupName, Comment);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	ApplicationName	Название приложения, например \\Node\InTouch .
	GroupName	Название группы, например \$System .
	Comment	Комментарий, вводимый во время подтверждения аларма.
Примечание	Список возвращаемых кодов ошибок можно найти в Приложении А.	
Примеры	MessageTag = "Acknowledge group, Turbines, by" + \$Operator; almAckGroup("AlmObj_1", "\Intouch", "Turbine", MessageTag);	
См. также	Ack(), almAckAll(), almAckDisplay(), almAckTag(), almAckRecent(), almAckPriority(), almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()	

almAckPriority()

	Подтверждает в указанном объекте вывода сведений о распределённых алармах все найденные во время выполнения последнего запроса алармы, приоритет которых попадает в указанный диапазон приоритетов и названия провайдера и группы алармов которых совпадают с указанными.	
Категория	алармы	
Синтаксис	[Result=]almAckPriority(ObjectName, ApplicationName, GroupName, FromPri, ToPri, Comment);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	ApplicationName	Название приложения, например \\Node\InTouch .
	GroupName	Название группы, например \$System .
	FromPri	Нижняя граница диапазона приоритетов, например 100.
	ToPri	Верхняя граница диапазона приоритетов, например 900.
	Comment	Комментарий, вводимый во время подтверждения аларма.
Примечание	Список возвращаемых кодов ошибок можно найти в Приложении А.	
Примеры	almAckPriority("AlmObj_1", "\\node1\Intouch", "Turbines", 10, 100, "almAckPriorityComment");	
См. также	Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckDisplay(), almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()	

almAckRecent()

	Подтверждает самые последние возникшие алармы	
Категория	алармы	
Синтаксис	[Result=]almAckRecent(ObjectName, Comment);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	Comment	Комментарий, вводимый во время подтверждения аларма.
Примечание	Список возвращаемых кодов ошибок можно найти в Приложении А.	
Примеры	almAckRecent("AlmObj_1", \$DateString);	
См. также	Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckDisplay(), almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()	

almAckSelect()

	Подтверждает только те алармы, которые выбраны (отмечены) в окне сводки алармов.	
Категория	алармы	
Синтаксис	[Result=]almAckSelect(ObjectName, Comment);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	Comment	Комментарий, вводимый во время подтверждения аларма.
Примечание	Список возвращаемых кодов ошибок можно найти в Приложении А.	
Примеры	<pre> IF (\$Hour > 0 and \$Hour < 8) THEN AckTag = "NightShift"; ELSE AckTag = "Day Shift"; ENDIF; almAckSelect("AlmObj_1",AckTag); </pre>	
См. также	Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckDisplay(), almAckRecent(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()	

almAckSelectedGroup()

	Подтверждает алармы, названия провайдеров и групп алармов которых совпадают с соответствующими параметрами алармов, выделенных в текущий момент в окне указанного объекта вывода сведений о распределённых алармах.	
Категория	алармы	
Синтаксис	[Result=]almAckSelectedGroup(ObjectName, Comment);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	Comment	Комментарий, вводимый во время подтверждения аларма.
Примечание		
Примеры	MessageTag = "Подтверждение алармов из указанных групп оператором " + \$Operator; almAckSelectedGroup("AlmObj_1", MessageTag);	
См. также	Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckDisplay(), almAckRecent(), almAckSelect(), almAckSelectedPriority(), almAckSelectedTag()	

almAckSelectedPriority()

	Подтверждает алармы, названия провайдеров и групп алармов которых, а также приоритеты совпадают с соответствующими параметрами алармов, выделенных в текущий момент в окне указанного объекта вывода сведений о распределённых алармах.	
Категория	алармы	
Синтаксис	[Result=]almAckSelectedPriority(ObjectName, Comment);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	Comment	Комментарий, вводимый во время подтверждения аларма.
Примечание	Диапазон приоритетов вычисляется по приоритетам выделенных в текущий момент алармов.	
Примеры	MessageTag = "Подтверждение алармов с указанными приоритетами оператором " + \$Operator; almAckSelectedPriority("AlmObj_1", MessageTag);	
См. также	Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckDisplay(), almAckRecent(), almAckSelect(), almAckSelectedPriority(), almAckSelectedTag()	

almAckSelectedTag()

	Подтверждает алармы, названия провайдеров и групп алармов, приоритеты и теги которых совпадают с соответствующими параметрами алармов, выделенных в текущий момент в окне указанного объекта вывода сведений о распределённых алармах.	
Категория	алармы	
Синтаксис	[Result=]almAckSelectedTag(ObjectName, Comment);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	Comment	Комментарий, вводимый во время подтверждения аларма.
Примечание		
Примеры	MessageTag = "Подтверждение алармов тех же тегов оператором " + \$Operator; almAckSelectedTag("AlmObj_1", MessageTag);	
См. также	Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckDisplay(), almAckRecent(), almAckSelect(), almAckSelectedPriority(), almAckSelectedTag()	

almAckTag()

	Подтверждает в указанном объекте вывода сведений о распределённых алармах все найденные во время выполнения последнего запроса алармы, соответствующий тег которых совпадает с указанным.	
Категория	алармы	
Синтаксис	[Result=]almAckTag(ObjectName, ApplicationName, GroupName, Tagname, FromPri, ToPri, Comment);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	ApplicationName	Название приложения, например \\Node\InTouch .
	GroupName	Название группы, например \$System .
	Tagname	Название тега.
	FromPri	Нижняя граница диапазона, например 100.
	ToPri	Верхняя граница диапазона, например 900.
	Comment	Комментарий, вводимый во время подтверждения аларма.
Примечание		
Примеры	almAckTag("AlmObj_1", "\\node1\Intouch", "Turbines", "Valve1", 10, 100, "Value", "LoLo", "almAckTagComment");	
См. также	Ack(), almAckAll(), almAckGroup(), almAckDisplay(), almAckRecent(), almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag(0	

almDefQuery()

	Выполняет запрос на обновление объекта вывода сведений о распределённых алармах с параметрами по умолчанию	
Категория	алармы	
Синтаксис	[Result=]almDefQuery(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примечание	<p>Параметры запроса по умолчанию указываются во время разработки программы.</p> <p>Перечень кодов возможных ошибок можно найти в Приложении А.</p>	
Примеры	almDefQuery("AlmObj_1");	
См. также	almQuery(), almSetQueryByName()	

almMoveWindow()

	Пролистывает окно объекта вывода сведений о распределённых алармах		
Категория	алармы		
Синтаксис	[Result=]almMoveWindow(ObjectName, Options, Repeat);		
	Аргумент	Описание	
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .	
	Options	Тип смещения:	
		Значение	Описание
		LineDn	На строку вниз
		LineUp	На строку вверх
		PageDn	На страницу вниз
		PageUp	На страницу вверх
		Top	В начало списка
		Bottom	В конец списка
		PageRt	На страницу вправо
		PageLf	На страницу влево
		Right	К концу списка (правая сторона)
		Left	К началу списка (левая сторона)
		Repeat	Количество повторений указанного действия
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.		
Примеры	almMoveWindow("AlmObj_1", "Bottom", 0); almMoveWindow("AlmObj_1", "LineDn", 3); almMoveWindow("AlmObj_1", "PageUp", 0);		

almQuery()

	Выполняет запрос на обновление указанного объекта вывода сведений о распределённых алармах.	
Категория	алармы	
Синтаксис	[Result=]almQuery(ObjectName, AlarmList, FromPri, ToPri, State, Type);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	AlarmList	Название группы алармов или список групп для выполнения запроса, например "\intouch!\$System", либо, название тега типа "сообщение".
	FromPri	Нижняя граница диапазона приоритетов, например 100, либо название тега целого типа.
	ToPri	Верхняя граница диапазона приоритетов, например 900, либо название тега целого типа.
	State	Статус отображаемых алармов, например "UnAck", либо название тега типа "сообщение". Допустимые статусы: "Ack", "UnAck", "All".
	Type	Типа запроса: "Hist" (Хронологический) либо "Summ" (Сводка)
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Примеры	<p>Следующий оператор выбирает все зарегистрированные сведения об алармах из группы MyAlarmListGroup с приоритетами от 500 до 600. Найденная информация будет показана в окне объекта вывода сведений о распределённых алармах AlmObj_1:</p> <pre>AlmQuery("AlmObj_1", "MyAlarmListGroup", 500, 600, "All", "Hist");</pre> <p>Допустимый синтаксис аргумента AlarmList:</p> <p>MyAlarmList где MyAlarmList – название списка алармов, созданного в Менеджере названий:</p> <pre>\intouch!GroupA</pre> <p>где GroupA – это группа алармов, определённая в локальном узле;</p> <pre>\\NodeXInTouch!GroupB</pre> <p>где GroupB – это группа алармов, определённая в узле Node X.</p>	
См. также	almDefQuery(), almSetQueryByName()	

almSelectAll()

	Последовательно выбирает и отменяет выбор всех записей об алармах в окне указанного объекта вывода сведений о распределённых алармах	
Категория	алармы	
Синтаксис	[Result=]almSelectAll(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Примеры	<pre> If \$AccessLevel > 8000 THEN almSelectAll("AlmObj_1"); almAckSelect("AlmObj_1", "Алармы подтверждены начальником"); ENDIF; </pre>	
См. также	almSelectItem(), almSelectGroup(), almSelectPriority(), almSelectTag(), almUnSelectAll()	

almSelectGroup()

	Последовательно выбирает и отменяет в окне указанного объекта вывода сведений о распределённых алармах выбор всех записей о найденных в результате исполнения последнего запроса алармах, название группы алармов которых совпадает с указанным.	
Категория	алармы	
Синтаксис	[Result=]almSelectGroup(ObjectName, ApplicationName, GroupName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	ApplicationName	Название приложения, например \\node1\Intouch .
	GroupName	Название группы алармов, например System .
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Примеры	almSelectGroup("AlmObj_1", "\\Intouch", "Turbine");	
См. также	almSelectAll(), almSelectItem(), almSelectPriority(), almSelectTag(), almUnSelectAll()	

almSelectionCount()

	Возвращает число записей об алармах, выделенных оператором в окне объекта вывода сведений о распределённых алармах	
Синтаксис	[Result=]almSelectionCount(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примеры	Tag1 = almSelectionCount("AlmObj_1");	

almSelectItem()

	Последовательно выбирает и отменяет выбор выбранного элемента в окне указанного объекта вывода сведений о распределённых алармах	
Категория	алармы	
Синтаксис	[Result=]almSelectItem(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Примеры	almSelectItem("AlmObj_1");	
См. также	almSelectAll(), almSelectGroup(), almSelectPriority(), almSelectTag(), almUnSelectAll()	

almSelectPriority()

	Последовательно выбирает и отменяет в окне указанного объекта вывода сведений о распределённых алармах выбор всех записей о найденных в результате исполнения последнего запроса алармах, приоритет которых находится в указанном диапазоне значений.	
Категория	алармы	
Синтаксис	[Result=]almSelectPriority (ObjectName, ApplicationName, GroupName, FromPri, ToPri);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	ApplicationName	Название приложения, например \\node1\Intouch .
	GroupName	Название группы алармов, \$System .
	FromPri	Нижняя граница диапазонов приоритетов, например 100, либо название тега целого типа.
	ToPri	Верхняя граница диапазонов приоритетов, например 900, либо название тега целого типа.
Примеры	almSelectPriority("AlmObj_1", "\\node1\Intouch", "Turbines", 10, 100);	
См. также	almSelectAll(), almSelectItem(), almSelectGroup(), almSelectTag(), almUnSelectAll()	

almSelectTag()

	Последовательно выбирает и отменяет в окне указанного объекта вывода сведений о распределённых алармах выбор всех записей о найденных в результате исполнения последнего запроса алармах указанного тега.	
Категория	алармы	
Синтаксис	[Result=]almSelectPriority (ObjectName, ApplicationName, GroupName, Tagname, FromPri, ToPri);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	ApplicationName	Название приложения, например \\node1\Intouch .
	GroupName	Название группы алармов, \$System .
	Tagname	Название тега аларма.
	FromPri	Нижняя граница диапазонов приоритетов, например 100, либо название тега целого типа.
	ToPri	Верхняя граница диапазонов приоритетов, например 900, либо название тега целого типа.
Примеры	almSelectTag("AlmObj_1", "\\node1\Intouch", "Turbines", "Valve1", 10, 100);	
См. также	almSelectAll(), almSelectItem(), almSelectGroup(), almSelectPriority(), almUnSelectAll()	

almSetQueryByName()

	Определяет новый запрос на алармы для объекта вывода сведений о распределённых алармах с параметрами по умолчанию, установленными для этого запроса	
Категория	алармы	
Синтаксис	[Result=]almSetQueryByName(ObjectName, QueryName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	QueryName	Название запроса, указанное в окне Query Favorites (Основные запросы) .
Примечание	При выполнении запроса полученными данными заполняется окно указанного объекта вывода сведений о распределённых алармах. На экране может быть открыто несколько таких окон, каждое с отдельным запросом.	
Примеры	almSetQueryByName("AlmObj_1", "Turbine Queries");	
См. также	almQuery(), almDefQuery()	

almShowStats()

	Открывает окно статистики объекта вывода сведений о распределённых алармах	
Категория	алармы	
Синтаксис	[Result=]almShowStats(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Примеры	almShowStats("AlmObj_1");	

almSupressAll()

	Запрет вывода сведений о всех текущих и будущих экземплярах алармов, найденных при выполнении текущего запроса, включая не отображаемые в текущий момент.	
Категория	алармы	
Синтаксис	[Result=] almSupressAll(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примечание	Данная функция аналогична функции almAckAll() , запрещающая отображение сведений обо всех алармах, записи о которых можно увидеть в окне объекта вывода сведений о распределённых алармах, прокручивая его вверх и вниз.	
Примеры	almSupressAll("AlmObj_1");	
См. также	almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressPriority(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()	

almSuppressDisplay()

	Подавляет показ текущего и последующих возниуновений тех алармов, которые видны в объекте сводного представления алармов.	
Категория	алармы	
Синтаксис	[Результат=] almSuppressDisplay(ИмяОбъекта) ;	
	Параметр	Описание
	ИмяОбъекта	Имя объекта алармов. Например, AlmObj_1.
Комментарии	Эта функция работает подобно соответствующей функции almAckDisplay() , определяя, какие алармы необходимо подавить, определяя все алармы, которые видны в настоящий момент.	
Пример(ы)	almSuppressDisplay("AlmObj_1") ;	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressPriority(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()	

almSuppressGroup()

	Запрет вывода сведений о всех текущих и будущих экземплярах алармов, входящих в указанную группу.	
Категория	алармы	
Синтаксис	[Result=]almSuppressGroup(ObjectName, ApplicationName, GroupName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	ApplicationName	Название приложения, например \\node1\InTouch
	GroupName	Название группы, например \$System
Примеры	almSuppressGroup("AlmObj_1", "\InTouch", "Turbines");	
См. также	almSuppressAll(), almSuppressTag(), almSuppressDisplay(), almSuppressPriority(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()	

almSuppressDisplay()

	Запрет вывода сведений о всех текущих и будущих экземплярах алармов, записи о которых в текущий момент видны на экране.	
Категория	алармы	
Синтаксис	[Result=]almSuppressDisplay(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примечание	Данная функция аналогична функции almAckDisplay() , за исключением того что она запрещает дальнейшее отображение информации о тех алармах, записи о которых в текущий момент видны в окне объекта вывода сведений о распределённых алармах.	
Примеры	almSuppressDisplay("AlmObj_1");	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressPriority(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()	

almSuppressPriority()

	Запрет вывода сведений о всех текущих и будущих экземплярах алармов, приоритет которых попадает в указанный диапазон значений и названия провайдера и группы алармов которых совпадают с указанными.	
Категория	алармы	
Синтаксис	[Result=]almSuppressPriority(ObjectName, ApplicationName, GroupName, FromPri, ToPri);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	ApplicationName	Название приложения, например \\node1\InTouch
	GroupName	Название группы, например \$\$System
	FromPri	Нижняя граница диапазонов приоритетов, например 100, или название тега целого типа.
	ToPri	Верхняя граница диапазонов приоритетов, например 900, или название тега целого типа.
Примеры	almSuppressPriority("AlmObj_1", "\\node1\Intouch", "Turbines", 10, 100);	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()	

almSuppressRetain()

	Сохраняет состояние фильтрации алармов для всех будущих запросов	
Категория	алармы	
Синтаксис	[Result=]almSuppressRetain (ObjectName, SuppressionRetainFlag);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	SuppressionRetainFlag	Название любого тега логического или аналогового типа, могущего иметь нулевое и ненулевое значение. TRUE означает сохранение статуса фильтрации для всех последующих запросов, FALSE – отмену сохранения.
Примечание	Если значение аргумента SuppressionRetainFlag равно 0, то при выполнении последующих запросов все параметры фильтрации сбрасываются.	
Примеры	almSuppressRetain("AlmObj_1", TRUE);	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressPriority(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()	

almSuppressSelected()

	Запрет вывода сведений о всех текущих и будущих экземплярах алармов, выбранных в текущий момент в окне сводных сведений об алармах	
Категория	алармы	
Синтаксис	[Result=]almSuppressSelected (ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примечание	Данная функция аналогична функции almAckSelect() , алармы для которой определяются путём выбора соответствующих записей в окне вывода сведений об алармах	
Примеры	almSuppressSelected("AlmObj_1");	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressPriority(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()	

almSuppressSelectedGroup()

	Подавляет экран текущих и будущих появлений алармов алармов, которые принадлежат принадлежат к тем группам одного или нескольких выделенных алармов, которые имеют одинаковое имя Провайдера в рамках именованного Экранного элемента управления распределёнными алармами.	
Категория	алармы	
Синтаксис	[Результат=]almSuppressSelectedGroup (ИмяОбъекта);	
	Параметр	Описание
	ИмяОбъекта	Имя объекта алармов. Например, AlmObj_1 .
Комментарии	Эта функция подобно функции almAckSelectedGroup() , выбирая те алармы, что выделены, затем выбирая группы, к которым они принадлежат, и подавляя появление алармов для этих групп.	
Пример(ы)	almSuppressSelectedGroup ("AlmObj_1") ;	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressSelected(), almSuppressSelectedPriority(), almSuppressSelectedTag()	

almSuppressTag()

	Запрет вывода сведений о всех текущих и будущих экземплярах алармов указанного тега, приоритет которых попадает в указанный диапазон значений и названия провайдера и группы алармов которых совпадают с указанными.	
Категория	алармы	
Синтаксис	[Result=]almSuppressTag(ObjectName, ApplicationName, GroupName, TagName, FromPri, ToPri, AlarmClass, AlarmType);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	ApplicationName	Название приложения, например \\node1\InTouch
	GroupName	Название группы, например \$System
	TagName	Название тега аларма.
	FromPri	Нижняя граница диапазонов приоритетов, например 100, или название тега целого типа.
	ToPri	Верхняя граница диапазонов приоритетов, например 900, или название тега целого типа.
	AlarmClass	Класс аларма, например "Value".
	AlarmType	Тип аларма, например "HiHi".
Примеры	almSuppressTag("AlmObj_1", "\\node1\Intouch", "Turbines", "Valve1", 10, 100, "Value", "LoLo");	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressDisplay(), almSuppressPriority(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()	

almSuppressSelectedGroup()

	Запрет вывода сведений о всех текущих и будущих экземплярах алармов, принадлежащих к той же группе алармов и имеющих того же провайдера, что и один или несколько алармов, записи о которых выделены в окне вывода сведений о распределённых алармах.	
Категория	алармы	
Синтаксис	[Result=]almSuppressSelectedGroup(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примечание	Данная функция аналогична функции almAckSelectedGroup() , за исключением того что после определения групп алармов и названий провайдеров после выделения каких-либо записей в окне выполняется блокировка вывода сведений о текущих и будущих экземпляров этих алармов.	
Примеры	almSuppressSelectedGroup("AlmObj_1");	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressSelected(), almSuppressSelectedPriority(), almSuppressSelectedTag()	

almSuppressSelectedPriority()

	Запрет вывода сведений о всех текущих и будущих экземплярах алармов, приоритет которых находится в том же диапазоне значений, что и приоритеты выбранных алармов, и которые принадлежат к тем же группам алармов и имеют тех же провайдеров.	
Категория	алармы	
Синтаксис	[Result=]almSuppressSelectedPriority(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примечание	<p>Диапазон значений приоритетов определяется по минимальному и максимальному приоритету всех выбранных в текущий момент алармов.</p> <p>Данная функция аналогична функции almAckSelectedPriority(), за исключением того что после определения диапазона приоритетов, групп алармов и названий провайдеров после выделения каких-либо записей в окне выполняется блокировка вывода сведений о текущих и будущих экземпляров всех алармов с похожими приоритетами.</p>	
Примеры	almSuppressSelectedPriority("AlmObj_1");	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressTagName(), almSuppressDisplay(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedTag(), almAckSelectedPriority()	

almSuppressSelectedTag()

	Запрет вывода сведений о всех текущих и будущих экземплярах алармов, принадлежащих тем же группам алармов и тегам и имеющих тех же провайдеров, что и один или несколько алармов, записи о которых выделены в окне объекта вывода сведений о распределённых алармах.	
Категория	алармы	
Синтаксис	[Result=]almSuppressSelectedTag(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примеры	almSuppressSelectedTag("AlmObj_1");	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressSelectedAlarm(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almAckSelectedTag(), almUnSuppressAll()	

almUnSelectAll()

	Отменяет выделение всех тегов, выбранных в текущий момент в указанном окне объекта вывода сведений о распределённых алармах	
Категория	алармы	
Синтаксис	[Result=]almUnSelectAll(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Примеры	<pre> If \$AccessLevel == 9999 THEN almAckSelect("AlmObj_1", "Comment"); {Этот аларм может подтверждать только администратор системы} ELSE almUnSelectAll("AlmObj_1"); ENDIF; </pre>	
См. также	almSelectAll(), almSelectItem(), almSelectGroup(), almSelectPriority(), almSelectTag()	

almUnSuppressAll()

	Отменяет фильтрацию вывода сведений об алармах	
Категория	алармы	
Синтаксис	[Result=] almUnSupressAll(ObjectName);	
	Аргумент	Описание
	ObjectName	Название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
Примеры	almUnSupressAll("AlmObj_1");	
См. также	almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressPriority(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag()	

AnnotateLayout()

	AnnotateLayout() позволяет пользователям добавить комментарий к текущему экрану просмотра, из которого вызывается скриптовая функция. Эта функция поддерживается только для операционной системы Windows XP Tablet PC Edition.
Категория	Системная
Синтаксис	AnnotateLayout ()
Комментарии	<p>Когда вызывается AnnotateLayout, появляется диалоговое окно с шаблоном комментария. Это диалоговое окно имеет следующие возможности.</p> <ul style="list-style-type: none"> ▪ Захватывает экранное изображение WindowViewer на момент, когда выводится диалоговое окно. ▪ Позволяет прокомментировать экранное изображение, используя перо в совокупности с панелью инструментов и элементы настроек меню. ▪ Позволяет сохранить изображение и комментарий в GIF- или JPEG-файле. ▪ Позволяет распечатать изображение и комментарий (если принтер сконфигурирован). ▪ Позволяет переслать изображение и комментарий по электронной почте в виде вложения (если сконфигурирован SMTP).
Пример(ы)	Нет
См. также	отсутствует

О функциях APU

Возвращаемыми кодами ошибок для функций APU являются положительные числа, при этом нулевой код означает успешное завершение.

Код ошибки	Сообщение ошибки
0	Success (успешное завершение).
1	Instance not found or not running (экземпляр не найден или не выполняется).
2	Interface not initialized (интерфейс не инициализирован).
3	Failure to access virtual memory (ошибка доступа к виртуальной памяти).
4	Invalid error code (код неизвестной ошибки).
5	Too many instances already running (уже запущено очень много экземпляров).
6	Результат строки would be too long (строка результата была бы очень длинной).
7	Invalid instance index passed to function (в функцию передан неверный экземпляр индекса).
8	Failed to post message to alarm printer program (ошибка при передаче сообщения в программу принтера алармов).
9	Failed wait for response from alarm printer application (ошибка ожидания отклика от приложения принтера алармов).
20	To priority must be greater or equivalent to From priority (приоритет "В" должен быть больше и равен приоритету "ИЗ").
21	Invalid priority value (неверное значение приоритета).
22	Invalid alarm state (неверное состояние аларма).
23	Fail to execute command because query is running (невозможно выполнить команду, поскольку выполняется запрос).
24	Query Строка is not valid (неверная строка запроса)
25	Invalid query processing state (неверное состояние обработки запроса).
26	Invalid printer state selector (неверный селектор состояния принтера).
27	Command received by alarm printer window is not recognized (команда, полученная окном принтера алармов, не распознана).
28	Query could not be started (запрос невозможно было запустить).

APUFindAlarmGroupInstance()

	Ищет первый экземпляр утилиты печати алармов (Утилита печати алармов (Alarm Print Utility)), используя указанную строку группы алармов. Может быть использована для начального получения зелаемого экземпляра утилиты печати алармов. Строка начальной группы алармов читается из файла ALC.	
Категория	Представление	
Синтаксис	[Результат=] APUFindAlarmGroupInstance(sGroup, iInstance);	
	Параметр	Описание
	sGroup	Имя группы
	iInstance	Целое – экземпляр
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUFindAlarmGroupInstance("\$System", 1);	
См. также	APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUFindFileInstance()

	Ищет первый экземпляр утилиты печати алармов (Утилита печати алармов (Alarm Print Utility)), используя указанную строку группы алармов. Может быть использована для начального получения зелаемого экземпляра утилиты печати алармов. Путь к файлу, который использует экземпляр ALC, может контролироваться пользователем вручную, или же он может быть указан в командной строке во время запуска программы.	
Категория	Представление	
Синтаксис	[Результат=] APUFindFileInstance(sFilePath, iInstance);	
	Параметр	Описание
	sFilePath	Строка – путь к файлу.
	iInstance	
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUFindFileInstance(sFilePath, iInstance);	
См. также	APUFindAlarmGroupInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUSTranslateErrorCode().	

APUFindPrinterInstance()

	Ищет первый экземпляр утилиты печати алармов (Утилита печати алармов (Alarm Print Utility)), используя указанную строку группы алармов. Может быть использована для начального получения элаемого экземпляра утилиты печати алармов. Имя принтера записывается и читается из ALC-файла. Строка имени принтера не чувствительна к регистру.	
Категория	Представление	
Синтаксис	[Результат=] APUFindPrinterInstance(sPrinter, iInstance);	
	Параметр	Описание
	sPrinter	Строка – принтер
	iInstance	Целое – экземпляр
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUFindPrinterInstance(sPrinter, iInstance);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUGetAlarmGroupText()

	Получает текст запроса аларма группы алармов. Первоначальный текст запроса аларма группы алармов читается из ALC-файла.	
Категория	Представление	
Синтаксис	[Результат=] APUGetAlarmGroupText(iInstance, sTagGroup) ;	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	sTagGroup	Текст – группа алармов
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUGetAlarmGroupText(iInstance, sTagGroup) ;	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUGetConfigurationFilePath()

	Получает полный путь используемого ALC-файла конфигурации. Путь к файлу, который использует экземпляр ALC, может контролироваться пользователем вручную, или же он может быть указан в командной строке во время запуска программы.	
Категория	Представление	
Синтаксис	[Результат=] APUGetConfigurationFilePath(iInstance, sTagFilePath);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	sTagFilePath	Строка – путь к файлу
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUGetConfigurationFilePath(iInstance, sTagFilePath);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUGetInstanceCount()

	Возвращает количество работающих экземпляров Утилиты печати алармов (Alarm Print Utility), вплоть до и включая шестнадцатый экземпляр. Любые экземпляры вне первых шестнадцати одновременно работающих экземпляров и динамически не контролируются, и для них не может быть получен статус.	
Категория	Представление	
Синтаксис	[Результат=] APUGetInstanceCount(iCount);	
	Параметр	Описание
	iCount	Целое – счётчик
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUGetInstanceCount(iCount);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUGetPrinterJobCount()

	<p>Получает самый последний статусный счётчик работ принтеров Windows для принтера, используемого данным экземпляром. Результаты не являются текущими, если только не выполняется запрос. Результаты не являются текущими, если только не печатается аларм: счётчик работ обычно обновляется, когда принтер открывается первый раз, и затем каждый раз, как печатается строка аларма. Это значение применимо только для принтеров, принадлежащих к принтерам Windows, и не распространяется на принтеры, ассоциированные с параллельным или последовательным портом.</p>	
Категория	Представление	
Синтаксис	[Результат=] APUGetPrinterJobCount (iInstance, iTagCount);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	iTagCount	Целое – счётчик.
	[Возврат]	Код ошибки (целое)
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	<pre>Status = APUGetPrinterJobCount(iInstance, iTagCount);</pre>	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUGetPrinterName()

	Получает имя принтера Windows или имя порта, используемого данным экземпляром. Возвращает 'НЕТ', если не сконфигурирован ни один принтер. Имя принтера записывается и читается из ALC-файла.	
Категория	Представление	
Синтаксис	[Результат=] APUGetPrinterName(iInstance, sTagPrinter);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	sTagPrinter	Текст – принтер
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUGetPrinterName(iInstance, sTagPrinter);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUGetPrinterStatus()

	<p>Получает самый последний статус принтеров Windows для принтера, используемого данным экземпляром. Результаты не являются текущими, если только не выполняется запрос. Результаты не являются текущими, если только не печатается аларм: счётчик работ обычно обновляется, когда принтер открывается первый раз, и затем каждый раз, как печатается строка аларма. Выбрать можно одно из следующего:</p> <ul style="list-style-type: none"> ▪ 0 = AlarmPrinterError (ошибка принтера алармов); ▪ 1 = AlarmPrinterNoPaper (в принтере алармов нет бумаги); ▪ 2 = AlarmPrinterOffline (принтер алармов выключен); ▪ 3 = AlarmPrinterOverFlow (принтер алармов переполнен). 	
Категория	Представление	
Синтаксис	[Результат=] APUGetPrinterStatus(iInstance, iSelector, iTagStatus);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	iSelector	Целое – селектор
	iTagStatus	Целое – статус
Комментарии	<p>Возвращает код ошибки в виде целого числа.</p> <p>ПРИМЕЧАНИЕ. Эта статусная информация запрашивается у принтера на базе стандартов драйверов Microsoft/Windows. Не все производители принтеров следуют этим стандартам, и, следовательно, не все принтеры будут возвращать статусную информацию.</p>	
Пример(ы)	Status = APUGetPrinterStatus(iInstance, iSelector, iTagStatus);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUGetQueryAlarmState()

	Получает состояние аларма для запроса: 0 = All (все), 1 = Ack (подтверждённые), 2 = UnAck (неподтверждённые). Начальное состояние аларма читается из ALC-файла.	
Категория	Представление	
Синтаксис	[Результат=] APUGetQueryAlarmState(iInstance, iTagState);	
	Параметр	Описание
	iInstance	Целевой экземпляр Утилиты печати алармов (Alarm Print Utility) – от 0 до 15.
	iTagState	0 = All (все), 1 = Ack (подтверждённые), 2 = UnAck (неподтверждённые)
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUGetQueryAlarmState(iInstance, iTagState);	

APUGetQueryFromPriority()

	Получает "приоритет из " для запроса. Начальный приоритет читается из ALC-файла.	
Категория	Представление	
Синтаксис	[Результат=] APUGetQueryFromPriority(iInstance, iTagPriority);	
	Параметр	Описание
	iInstance	Целевой экземпляр Утилиты печати алармов (Alarm Print Utility) – от 0 до 15.
	iTagPriority	
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUGetQueryFromPriority(iInstance, iTagPriority);	

APUGetQueryProcessingState()

	Получает состояние обработки запроса аларма.	
Категория	Представление	
Синтаксис	[Результат=] APUGetQueryProcessingState (iInstance, iTagState);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	iTagState	Целое – состояние: 0 = Stop (остановка), 1 = Start (запуск).
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUGetQueryProcessingState (iInstance, iTagState);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUGetQueryToPriority()

	Получает "приоритет в" для запроса. Начальный приоритет читается из ALC-файла.	
Категория	Представление	
Синтаксис	[Результат=] APUGetQueryToPriority(iInstance, iTagPriority);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	iTagPriority	Выходное (целое) состояние.
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUGetQueryToPriority(iInstance, iTagPriority);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUInstanceUsed()

	Успешное завершение означает, что экземпляр используется.	
Категория	Представление	
Синтаксис	[Результат=] APUInstanceUsed(iInstance);	
	Параметр	Описание
	iInstance	Вход (целое).
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUInstanceUsed(iInstance);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUSetAlarmGroupText()

	Устанавливает текст Группы алармов запроса аларма. Чтобы эта функция завершилась успешно, не должен выполняться никакой запрос. (О состоянии запуска, остановки и обработки запроса см. функции ниже.) Начальный текст группы алармов читается из ALC-файла.	
Категория	Представление	
Синтаксис	[Результат=] APUSetAlarmGroupText(iInstance, sGroup);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	sGroup	Текст – группа алармов
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUSetAlarmGroupText(iInstance, sGroup);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUSetQueryAlarmstate()

	Устанавливает состояние аларма для запроса: 0 = All (все), 1 = Ack (подтверждённые), 2 = UnAck (неподтверждённые). Чтобы эта функция завершилась успешно, не должен выполняться никакой запрос.	
Категория	Представление	
Синтаксис	[Результат=] APUSetQueryAlarmState(iInstance, iState);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	iState	0 = All (все), 1 = Ack (подтверждённые), 2 = UnAck (неподтверждённые)
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUSetQueryAlarmState(iInstance, iState);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUSetQueryToPriority()

	Устанавливает "приоритет в" для запроса. Приоритет 'В' должен устанавливаться на такой же или более высокий уровень, что и приоритет 'Из', в противном случае функция успешно не завершится. Чтобы эта функция завершилась успешно, не должен выполняться никакой запрос.	
Категория	Представление	
Синтаксис	[Результат=] APUSetQueryToPriority(iInstance, iPriority);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	iPriority	Входное (целое) состояние.
Remarks	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUSetQueryToPriority(iInstance, iPriority);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUSetTimeoutValues()

	<p>Задаёт интервал таймаута в миллисекундах. Таким способом можно управлять тем, как много ошибок, связанных с доступом к памяти или возникающих из-за невозможности получения правильного результата, будет наблюдаться во время работы программы. Таймаут по умолчанию для доступа к памяти составляет 2 секунды. Время ожидания по умолчанию для короткого отклика составляет 10 секунд, и время ожидания по умолчанию для длинного отклика составляет 20 секунд.</p>	
Категория	Представление	
Синтаксис	<pre>[Результат=] APUSetTimeoutValues(iMemory, iShort, iLong);</pre>	
	Параметр	Описание
	iMemory	Целое – время доступа
	iShort	Время ожидания отклика (целое)
	iLong	Время ожидания отклика (целое)
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	<pre>Status = APUSetTimeoutValues(iMemory, iShort, iLong);</pre>	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUStartInstance()

	<p>Запускает экземпляр Принтера алармов в минимизированном состоянии в соответствии с указанным конфигурационным файлом. Возвращает запущенный экземпляр или код ошибки ErrorCode. Программа Принтера алармов не будет автоматически начинать обработку базы данных алармов. Чтобы заставить экземпляр начать обработку базы данных алармов, используйте команду APUStartQuery().</p>	
Категория	Представление	
Синтаксис	[Результат=] APUStartInstance(sFilePath, iTagInstance);	
	Параметр	Описание
	sFilePath	Полный путь к конфигурационному файлу (входная строка)
	iTagInstance	Возвращаемый номер запущенного экземпляра или сообщение об ошибке (целое).
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUStartInstance(sFilePath, iTagInstance);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartQuery(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUStartQuery()

	<p>Устанавливает пределы даты и времени для записей, которые должны быть обработаны, и затем запускает запрос. Будет ошибкой попытка запустить запрос, если какой-либо запрос уже выполняется. Установка даты в 0 приводит к печати всех алармов. Каждой конкретный 0, подразумевающий "не проверяемое состояние", по умолчанию устанавливает минимальное возможное значение. Полночь 1 января 1900 года является наиболее ранним временем, используемым по умолчанию. Указанные время и дата задаётся в местном времени. Значение '-1' для года установит дату на текущее время, в которое обрабатывается команда.</p>	
Категория	Представление	
Синтаксис	<pre>[Результат=] APUStartQuery(iInstance, iYear, iMonth, iDay, iHour, iMinute);</pre>	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
	iYear	Входной (целое)
	iMonth	Входной (целое)
	iDay	Входной (целое)
	iHour	Входной (целое)
	iMinute	Входной (целое)
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	<pre>Status = APUStartQuery(iInstance, iYear, iMonth, iDay, iHour, iMinute);</pre>	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStopInstance(), APUStopQuery(), APUTranslateErrorCode().	

APUStopInstance()

	<p>Останавливает указанный экземпляр Принтера Алармов. Любое дальнейшее добавление записей, которые должны быть распечатаны, будет остановлено, любой выполняющийся запрос на печать базы данных алармов будет прерван, и экземпляр программы будет завершён.</p>	
Категория	Представление	
Синтаксис	[Результат=] APUStopInstance(iInstance);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUStopInstance(iInstance);	
См. также	<p>APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopQuery(), APUTranslateErrorCode().</p>	

APUStopQuery()

	Посылает запрос указанному экземпляру на прекращение выполнения его запроса. Приложение будет оставаться запущенным, но он не будет обрабатывать никакие запросы базы данных алармов. Заставить продолжить запросы базы данных алармов может вызов APUStartQuery().	
Категория	Представление	
Синтаксис	[Результат=] APUStopQuery(iInstance);	
	Параметр	Описание
	iInstance	Целевой экземпляр утилиты печати алармов (от 0 до 15).
Комментарии	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUStopQuery(iInstance);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUTranslateErrorCode().	

APUTranslateErrorCode()

	Преобразовывает код ошибки, возвращаемой одной из APU-функций, в строку на английском языке, кратко описывающей код ошибки. Может привести к ошибке, если на преобразование передан неизвестный код ошибки.	
Категория	Представление	
Синтаксис	[Результат=] APUTranslateErrorCode (iErrorCode, sTagMessage);	
	Параметр	Описание
	iErrorCode	Целое – код ошибки
	sTagMessage	Строка – сообщение ошибки
Remarks	Возвращает код ошибки в виде целого числа.	
Пример(ы)	Status = APUTranslateErrorCode(iErrorCode, sTagMessage);	
См. также	APUFindAlarmGroupInstance(), APUFindFieldInstance(), APUFindPrinterInstance(), APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetInstanceCount(), APUGetPrinterJobCount(), APUGetPrinterName(), APUGetPrinterStatus(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority(), APUIsInstanceUsed(), APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues(), APUStartInstance(), APUStartQuery(), APUStopInstance(), APUStopQuery().	

ArcCos()

	Для заданного числа от -1 до 1 (включительно) возвращает угол от 0 до 180 градусов, <i>косинус</i> которого равен указанной величине.	
Категория	математическая	
Синтаксис	Result=ArcCos(Number);	
	Аргумент	Описание
	Number	Любое число либо тег целого или вещественного типа.
Примечание	Функция вычисляет арккосинус указанной величины и возвращает полученное значение в тег Result.	
Примеры	ArcCos(1) = 0, ArcCos(-1) = 180	
См. также	Cos(), Sin(), Tan(), ArcSin(), ArcTan()	

ArcSin()

	Для заданного числа от -1 до 1 (включительно) возвращает угол от -90 до 90 градусов, <i>синус</i> которого равен указанной величине.	
Категория	математическая	
Синтаксис	Result=ArcSin(Number);	
	Аргумент	Описание
	Number	Любое число либо тег целого или вещественного типа.
Примечание	Функция вычисляет арксинус указанной величины и возвращает полученное значение в тег Result. Перечень кодов возможных ошибок можно найти в Приложении А.	
Примеры	ArcSin(1) = 90, ArcSin(-1) = -90	
См. также	Cos(), Sin(), Tan(), ArcCos(), ArcTan()	

ArcTan()

	Для заданного числа возвращает угол от -90 до 90 градусов, <i>тангенс</i> которого равен указанной величине.	
Категория	математическая	
Синтаксис	Result=ArcTan(Number);	
	Аргумент	Описание
	Number	Любое число либо тег целого или вещественного типа.
Примечание	Функция вычисляет арктангенс указанной величины и возвращает полученное значение в тег Result. Перечень кодов возможных ошибок можно найти в Приложении А.	
Примеры	ArcTan(1) = 45, ArcTan(-) = 0	
См. также	Cos(), Sin(), Tan(), ArcCos(), ArcSin()	

AttemptInvisibleLogon()

	Попытка регистрации в системе InTouch с использованием указанных параметров	
Категория	контроль доступа	
Синтаксис	DiscreteTag=AttemptInvisibleLogon("UserId", "Password", "Domain");	
	Аргумент	Описание
	UserId	Действительное имя пользователя
	Password	Пароль пользователя
	Domain	Название локальной машины, рабочей группы либо домена, к которому принадлежит указанный пользователь. Данное значение используется, если контроль доступа осуществляется средствами операционной системы.
	DiscreteTag	Логический тег, в который возвращается результат выполнения функции: TRUE, если проверка прошла успешно, в противном случае возвращается FALSE.
Примечание	Функция выполняет попытку регистрации в системе InTouch с указанными параметрами (значение аргумента Domain используется, только если контроль доступа осуществляется средствами операционной системы). При успешной регистрации функция возвращает значение TRUE, значения системных тегов \$OperatorDomain, \$OperatorName, \$AccessLevel и \$Operator изменяются соответствующим образом. При ошибке регистрации функция возвращает значение FALSE, сеанс работы текущего зарегистрированного пользователя (если таковой был) сохраняется.	
Примеры	AnalogTag=InvisibleVerifyCredentials("ravi", "Password", "wonderware_hq"); \\ контроль доступа осуществляется средствами ОС AnalogTag=InvisibleVerifyCredentials("ravi", "Password", ""); \\ контроль доступа осуществляется средствами InTouch либо сервера AppServer	
См. также	PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), QueryGroupMembership(), AddPermission().	

ChangePassword()

	Открывает диалоговое окно Change Password (Изменить пароль) , давая возможность зарегистрированному пользователю изменить свой пароль.	
Категория	контроль доступа	
Синтаксис	[Result=]ChangePassword();	
	Аргумент	Описание
	Result	Возвращаемая функцией величина, которая может принимать одно из следующих значений: 0 = была нажата кнопка Cancel (Отмена) , 1 = была нажата кнопка ОК .
Примечание	При использовании в системах с сенсорными экранами предоставляется возможность вводить данные с помощью алфавитно-цифровой клавиатуры.	
Примеры	Errmgs=ChangePassword(); Если связать эту функцию с экранной кнопкой или запустить её при исполнении скрипта, то она откроет диалоговое окно (с изображением клавиатуры), в котором пользователь должен ввести текущий пароль, новый пароль, а также подтверждение нового пароля.	
См. также		

ChangeWindowColor()

	Возвращает статус – успех или отказ – для изменения цвета фона окна.	
Категория	дополнительная	
Синтаксис	IntegerResult = ChangeWindowColor("Window Name", RValue,GValue,BValue);	
	Параметр	Описание
	WindowName	Имя окна, у которого нужно изменить цвет фона. Допускается строка или тэг сообщения.
	RValue	Задаёт интенсивность красного цвета. Допускается целое число или целый тэг памяти (допустимый диапазон 0 – 255).
	BValue	Задаёт интенсивность голубого цвета. Допускается целое число или целый тэг памяти (допустимый диапазон 0 – 255).
	GValue	Задаёт интенсивность зелёного цвета. Допускается целое число или целый тэг памяти (допустимый диапазон 0 – 255).
Комментарии	Возвращаемым типом является 0, 1 или 2, и он присваивается целому тэгу. 0 – ошибка – окно не определено или RGB-значение вне диапазона, 1 – нормальное завершение, 2 – ошибка – окно существует и в настоящий момент не открыто.	
Пример	<p>ChangeWindowColor("Win1", 255,255,255) возвратит 1, где "Win1" – имя определённого и открытого в настоящий момент окна.</p> <p>ChangeWindowColor("Win2", 255,255,255) возвратит 0, где "Win2" – имя окна, причём окно с указанным именем RGB-..... (..... 0 – 255)</p> <p>ChangeWindowColor("Win3", 255,255,255) возвратит 0, где "Win3" – имя определённого, но не открытого в настоящий момент окна.</p>	

Cos()

	Возвращает <i>косинус</i> угла, выраженного в градусах	
Категория	математическая	
Синтаксис	Result=Cos(Number);	
	Аргумент	Описание
	Number	Любое число либо название тега целого или вещественного типа.
Примечание	Функция вычисляет значение косинуса угла, выраженного в градусах.	
Примеры	Cos(90) = 0 Cos(0) = 1 Wave = 50 * Cos(6 * \$Second);	
См. также	Sin(), Tan(), ArcCos(), ArcSin(), ArcTan()	

DateTimeGMT()

	Возвращает число, выражающее количество и дробную часть дней, прошедших с 1 января 1970 года, в Универсальных координатах времени (Coordinated Universal Time – UTC), независимо от местной временной зоны.	
Категория	дополнительная	
Синтаксис	Результат=DateTimeGMT () ;	
	Параметр	Описание
	Нет	
Комментарии		
Пример(ы)	MessageTag = StringFromTime (DateTimeGMT () * 86400.0, 3) ;	
См. также		

DialogStringEntry()

	Выдает на экран алфавитно-цифровую клавиатуру, давая пользователю возможность изменить текущее значение тега типа "сообщение" в Словаре тегов	
Категория	прочие	
Синтаксис	[Result=]DialogStringEntry(MessageTag_Text, UserPrompt_Text);	
	Аргумент	Описание
	Result	Возвращаемая функцией величина, которая может принимать одно из следующих значений: 0 = была нажата кнопка Cancel (Отмена) , 1 = была нажата кнопка OK , -1 = внутренняя ошибка, -2 = ошибка инициализации, -3 = тег не определен, -4 = тип тега не "сообщение", -5 = ошибка записи.
	MessageTag_Text	Название тега типа "сообщение", значение которого требуется изменить. (Эта функция требует указания строки символов в качестве данного параметра, поэтому название тега должно быть либо заключено в кавычки, либо дополнено полем .Name , но уже без кавычек). Допускается также указание тега типа "сообщение" в указателя на тег. См. приведенные примеры.
	UserPrompt_Text	Задаёт строку символов, которая будет выведена над клавиатурой.
Примечание	Данная функция предназначена в основном для использования в системах с сенсорными экранами.	

<p>Примеры</p>	<pre>Errmsg=DialogStringEntry(MyMessageTag.Name, " Введите новое значение..."); Errmsg=DialogStringEntry("MyMessageTag", " Введите новое значение...");</pre> <p>Указание названий тегов на месте аргумента MessageTag_Text позволяет менять отображаемый текст во время исполнения программы.</p> <pre>Errmsg=DialogStringEntry(MessageTagX, MessageDisplay);</pre> <p>Определив соответствующие анимационные функции ввода значений, значения этих аргументов можно изменять до выполнения данной функции.</p> <p>Например, следующий скрипт отображает на экране алфавитно-цифровую клавиатуру для изменения значения тега MyMessageTag, при этом над клавиатурой будет выведена строка "Введите новое значение...":</p> <pre>MessageTagX="MyMessageTag"; {Запись строки символов "MyMessageTag" (которая фактически является названием изменяемого тега) во внутренний тег MessageTagX типа "сообщение" } MessageDisplay="Введите новое значение..."; {Определение значения внутреннего тега MessageDisplay типа "сообщение" } Errmsg=DialogStringEntry(MessageTagX, MessageDisplay); {Кавычки не требуются, поскольку тег MessageTagX был определён как тег типа "сообщение" }</pre>
<p>См. также</p>	<p>DialogValueEntry()</p>

DialogValueEntry()

	Выдает на экран алфавитно-цифровую клавиатуру, давая пользователю возможность изменить текущее значение тега логического, целого или вещественного типа в Словаре тегов	
Категория	прочие	
Синтаксис	[Result=]DialogValueEntry(ValueTag_Text, LowLimit, HighLimit, UserPrompt_Text);	
	Аргумент	Описание
	Result	Возвращаемая функцией величина, которая может принимать одно из следующих значений: 0 = была нажата кнопка Cancel (Отмена) , 1 = была нажата кнопка OK , -1 = HighLimit <= LowLimit, -2 = ошибка инициализации, -3 = тег не определен, -4 = тип тега не является логическим, целым либо вещественным, -5 = ошибка записи.
	ValueTag_Text	Название тега логического, целого либо вещественного типа, значение которого требуется изменить. (Эта функция требует указания строки символов в качестве данного параметра, поэтому название тега должно быть либо заключено в кавычки, либо дополнено полем .Name , но уже без кавычек). Допускается также указание тега типа "сообщение" в качестве указателя на тег. См. приведенные примеры.
	LowLimit	Минимально допустимое значение для данного тега (не меньше, чем значение полей Min Value, Min Raw либо MinEU, в зависимости от определения).
	HighLimit	Максимально допустимое значение для данного тега (не больше, чем значение полей Max Value, Max Raw либо MaxEU, в зависимости от определения).
	UserPrompt_Text	Задаёт строку символов, которая будет выведена над клавиатурой.
Примечание	Данная функция предназначена в основном для использования в системах с сенсорными экранами.	

<p>Примеры</p>	<pre>Errmsg=DialogValueEntry(MyIntegerTag.Name, MyIntegerTag.MinEU, MyIntegerTag.MaxEU, "Введите новое значение..."); Errmsg=DialogValueEntry("MyIntegerTag", -100, 100, " Введите новое значение...");</pre> <p>Указание названий тегов на месте аргумента MessageTag_Text позволяет менять отображаемый текст во время исполнения программы.</p> <pre>Errmsg=DialogValueEntry(TagnameX, Min, Max, MessageDisplay);</pre> <p>Определив соответствующие анимационные функции ввода значений, значения этих аргументов можно изменять до выполнения данной функции.</p> <p>Например, следующий скрипт отображает на экране цифровую клавиатуру для установки нового значения тега MyIntegerTag из диапазона от -100 до 100, при этом над клавиатурой будет выведена строка "Введите новое значение...":</p> <pre>TagnameX="MyIntegerTag"; { Запись строки символов "MyIntegerTag" (которая фактически является названием изменяемого тега) во внутренний тег TagnameX типа "сообщение" }</pre> <pre>Min=-100; {Запись минимально допустимого нового значения во внутренний тег Min целого либо вещественного типа }</pre> <pre>Max=100; { Запись максимально допустимого нового значения во внутренний тег Max целого либо вещественного типа }</pre> <pre>MessageDisplay="Enter a new value..."; {Определение строки, которая будет выводиться над цифровой клавиатурой }</pre> <pre>Errmsg=DialogValueEntry(TagnameX, Min, Max, MessageDisplay); { Кавычки не требуются, поскольку тег TagnameX был определён как тег типа "сообщение". После записи названия тега логического, целого или вещественного типа в тег TagnameX функция изменит его значение }</pre>
<p>См. также</p>	<p>DialogStringEntry()</p>

DText()

	Изменяет во время исполнения программы значение тега типа "сообщение" в зависимости от текущего значения тега логического типа	
Категория	символьные	
Синтаксис	MsgTag=DText(Discrete_Tag, OnMsg, OffMsg);	
	Аргумент	Описание
	MsgTag	Тег типа "сообщение".
	Discrete_Tag	Тег логического типа.
	OnMsg	Строка, соответствующая значению 1 (истина, TRUE) логического тега Discrete_Tag.
	OffMsg	Строка, соответствующая значению 0 (ложь, FALSE) логического тега Discrete_Tag.
Примеры	MessageTag=Dtext(DiscreteTag, DiscreteTag.OnMsg, DiscreteTag.OffMsg); If Dtext(D_Tag, "True", "False") == "True" THEN Message="Значение тега D_Tag равно 1"; ELSE Message="Значение тега D_Tag равно 0"; ENDIF;	

EnableDisableKeys()

	Разрешает и запрещает распознавание нажатий на клавиш Alt, Escape и Windows	
Категория	View	
Синтаксис	EnableDisableKeys(int AltKey, int EscKey, int WinKey);	
	Аргумент	Описание
	AltKey	Целое, разрешающее или запрещающее клавишные фильтры для клавиши Alt; 1=разрешает фильтр (disable клавишу Alt), 0=запрещает фильтр (разрешает клавишу Alt).
	EscKey	Целое, разрешающее или запрещающее клавишные фильтры для клавиши Escape key, 1=разрешает фильтр (запрещает клавишу Esc), 0=запрещает фильтр (разрешает клавишу Esc).
	WinKey	Целое, разрешающее или запрещающее клавишные фильтры для клавиши Windows key, 1= разрешает фильтр (запрещает клавишу Win), 0= запрещает filter (разрешает клавишу Win).
Примечание	Разрешает/запрещает клавиши Alt,Escape и Windows. Если вы запретили клавиш Alt, это также запрещает комбинацию клавиш Win+L (для блокирования настольной Windows). Win+L является сокращением для другого комбинирования клавиш, которое включает клавишу Alt key, следовательно, запрещение клавиши Alt также запрещает это частное сокращение.	
Примеры	<pre>EnableDisableKeys(0,0,0); // разрешает все три клавиши EnableDisableKeys(1,1,1); // запрещает все три клавиши EnableDisableKeys(0,0,1); // разрешает клавиши Alt и Escape, запрещает клавишу Windows.</pre>	

Exp()

	Возвращает степень числа e	
Категория	математическая	
Синтаксис	Result=Exp(Number);	
	Аргумент	Описание
	Number	Любое число либо тег целого или вещественного типа.
Примечание	В тег Result возвращается значение числа e , возведённого в степень Number .	
Примеры	<pre>Exp(1) = 2.71828...</pre> <p>Диапазон допустимых значений показателя функции – от -88,72 до 88,72.</p>	

FileCopy()

	Копирует исходный файл SourceFile в файл назначения DestFile аналогично команде Copy в DOS или Проводнике Windows	
Категория	системная	
Синтаксис	[Result=]FileCopy(SourceFile, DestFile, DoneTag);	
	Аргумент	Описание
	SourceFile	Ссылка на исходный файл (включая путь доступа).
	DestFile	Название конечного файла (включая путь доступа) либо каталог.
	DoneTag	Название тега, используемого для контроля завершения операции копирования функцией FileCopy(). Данный аргумент должен представлять собой строку с названием тега (а не сам тег). Таким образом, если тег контроля называется Monitor , то на месте этого аргумента необходимо указать " Monitor " либо Monitor.Name , а не просто Monitor.
Примечание	<p>Функция FileCopy() сразу же возвращает 1 при успешном начале процедуры копирования. Функция возвращает 0, если в данный момент выполняется другая процедура копирования (невозможно начать новую процедуру), и -1 в случае ошибки. Таким образом, с помощью возвращаемого значения можно контролировать выполнение функции FileCopy():</p> <pre>Status=FileCopy("C:*.TXT", "C:\BACKUP", "Monitor");</pre> <p>Здесь Status – название тега целого типа, в который записывается 1, -1 или 0. Копирование с помощью функции FileCopy() выполняется в фоновом режиме, поэтому оно не будет препятствовать исполнению приложения InTouch. Тег DoneTag обеспечивает возможность контроля копирования со стороны пользователя или приложения в ходе его выполнения. Это позволяет предупреждать пользователя о возникновении ошибок ПОСЛЕ начала копирования, в то время как тег Status отражает лишь результат ЗАПУСКА указанной процедуры. Значение тегу DoneTag присваивается только после успешного начала выполнения функции. Если копирование ещё не завершено, значение тега равно 0. Оно устанавливается равным 1, если процедура выполнилась успешно, или -1, если до завершения процедуры возникла какая-либо ошибка.</p> <p>Как правило, SourceFile и DestFile – это названия файлов. Однако, если копируется только один файл, назначением может быть каталог:</p> <pre>FileCopy("C:\DATA.TXT", "C:\BACKUP", "Monitor");</pre> <p>В результате выполнения функции файл DATA.TXT будет скопирован в каталог \BACKUP на диске C:. После завершения данной операции тег Monitor будет установлен в 1.</p> <p>Если в названии исходного файла присутствуют знаки подстановки, то местом назначения ДОЛЖЕН быть каталог (а</p>	

	<p>не название файла), в противном случае функция возвратит ошибку:</p> <pre>FileCopy("C:*.TXT", "C:\BACKUP", "Monitor");</pre> <p>В данном примере копируются все файлы с расширением .TXT из корневого каталога диска C:\ в каталог C:\BACKUP. По окончании копирования файлов значение тега Monitor станет равным 1.</p> <p>Функция FileCopy() внутри асинхронных Quick-функций не исполняется, её можно использовать только в синхронных функциях.</p>
См. также	FileDelete(), FileMove(), FileReadFields(), FileReadMessage(), FileWriteFields(), FileWriteMessage()

FileDelete()

	Удаляет указанный файл	
Категория	системная	
Синтаксис	[Result=]FileDelete(Filename);	
	Аргумент	Описание
	Filename	Название удаляемого файла.
Примечание	<p>Если файл указанный существует и может быть удалён, функция возвращает 1, в противном случае возвращается 0.</p> <p>Знаки подстановки (символы "*" и "?") в названии файла не допускаются. При указании символов подстановки функция возвращает ошибку "Could not find file" (Файл не найден).</p> <p>Функция FileDelete() внутри асинхронных Quick-функций не исполняется, её можно использовать только в синхронных функциях.</p>	
Примеры	<pre>Status=FileDelete("C:\DATA.TXT");</pre> <p>Тег Status получает значение 1, если функция находит файл DATA.TXT в каталоге C:\, в противном случае тегу присваивается значение 0.</p>	
См. также	FileCopy(), FileMove(), FileReadFields(), FileReadMessage(), FileWriteFields(), FileWriteMessage()	

FileMove()

	Аналогична функции FileCopy() , за исключением того что она перемещает файл, а не создаёт новую копию.	
Категория	системная	
Синтаксис	[Result=]FileMove(SourceFile, DestFile, DoneTag);	
	Аргумент	Описание
	SourceFile	Ссылка на исходный файл (включая путь доступа).
	DestFile	Название конечного файла (включая путь доступа) либо каталог.
	DoneTag	Название тега, используемого для контроля завершения операции копирования функцией FileCopy(). Данный аргумент должен представлять собой строку с названием тега (а не сам тег). Таким образом, если тег контроля называется Monitor , то на месте этого аргумента необходимо указать " Monitor " либо Monitor.Name , а не просто Monitor.
Примечание	<p>Функция FileMove() сразу же возвращает 1 при успешном начале процедуры копирования. Функция возвращает 0, если в данный момент выполняется другая процедура копирования (невозможно начать новую процедуру), и -1 в случае ошибки. Таким образом, с помощью возвращаемого значения можно контролировать выполнение функции FileMove():</p> <pre>Status=FileMove("C:\DATA.TXT", "D:\DATA.TXT", "Monitor");</pre> <p>Здесь Status – название тега целого типа, в который записывается 1, -1 или 0. Перемещение файла с помощью функции FileMove() выполняется в фоновом режиме, поэтому оно не будет препятствовать исполнению приложения InTouch. Тег DoneTag позволяет контролировать ход выполнения данной операции. Таким образом, пользователь может быть предупреждён о возникновении ошибок ПОСЛЕ начала операции, в то время как тег Status отражает лишь результат ЗАПУСКА указанной процедуры.</p> <p>Функция FileCopy() внутри асинхронных Quick-функций не исполняется, её можно использовать только в синхронных функциях.</p> <p>Значение тегу DoneTag присваивается только после успешного начала выполнения функции. Если перемещение файла ещё не завершено, значение тега равно 0. Оно устанавливается равным 1, если процедура выполнена успешно, или -1, если до завершения процедуры возникла какая-либо ошибка.</p> <p>Если файлы SourceFile и DestFile находятся на одном и том же диске, функция может просто изменить ссылку на каталог файла (в которой указаны название и расположение файла на диске) без фактического переноса данных. В данном случае операция</p>	

	<p>переноса будет выполнена очень быстро, независимо от размера файла. Если же исходный файл и файл назначения находятся на разных дисках, то время, затрачиваемое на перенос, будет зависеть от размера файла. Это связано с тем, что данные должны быть переписаны с одного физического диска на другой.</p>
Примеры	<pre>FileMove("C:\DATA.TXT", "C:\BACKUP\DATA.TXT", "Monitor");</pre> <p>При выполнении данного оператора файл DATA.TXT будет перемещён из корневого каталога диска C:\ в каталог \BACKUP. По завершении операции тег Monitor получит значение 1.</p> <p>С помощью этой же функции можно переименовывать файлы, если исходный файл и файл назначения ссылаются на один и тот же каталог.</p> <pre>FileMove ("C:\DATA.TXT", "C:\DATA.BAK", "Monitor");</pre>
См. также	<pre>FileCopy(), FileDelete(), FileReadFields(), FileReadMessage(), FileWriteMessage(), FileWriteFields()</pre>

Примечание. В функции FileMove() используется флаг готовности, поэтому очередную функцию перемещения файлов до завершения предыдущей запускать нельзя.

FileReadFields()

	Считывает записи данных из указанного файла в формате CSV (Comma Separated Values – значения, разделённые запятыми).	
Категория	системная	
Синтаксис	[Result=]FileReadFields(Filename, FileOffset, StartTag, NumberOfFields);	
	Аргумент	Описание
	Filename	Название файла.
	FileOffset	Начальная позиция в файле, с которой требуется считывать данные.
	StartTag	Название тега InTouch, в который будет записано значение <i>первого</i> элемента данных. Его название должно заканчиваться цифрой (например MyTag1). Данный аргумент должен быть строкой символов, определяющей название тега, а не самим названием. Таким образом, если тег называется MyTag1 , то в качестве аргумента необходимо указывать " MyTag1 " или MyTag1.Name , а не просто MyTag1.
	NumberOfFields	Задаёт число полей для считывания (число разделённых запятыми полей в каждой записи файла).
Примечание	Если начальный тег называется MyTag1 и число полей равно 3, из файла будет считано три поля, значения которых будут записаны в теги MyTag1 , MyTag2 и MyTag3 . Указанные теги с последовательными названиями должны быть предварительно созданы в Словаре InTouch и могут иметь различные типы (целый, сообщение и т.д.).	
Примеры	<p>Пусть первая запись файла C:\DATA\FILE.CSV содержит такие данные: This is text, 3.1416, 5</p> <p>Следующий оператор считает эту запись и сохранит строку "This is text" в теге MyTag1, число 3.1416 в теге MyTag2, а число 5 – в теге MyTag3</p> <pre>BytePosition=FileReadFields("C:\DATA\FILE.CSV", 0, "MyTag1", 3);</pre> <p>Выполнив чтение, функция возвратит новое значение указателя файла, которое можно использовать в последующих вызовах этой функции на месте аргумента FileOffset:</p> <pre>FileReadFields("C:\DATA\FILE.CSV", BytePosition, "MyTag1", 3);</pre> <p>Следует учесть, что теги типа "сообщение" могут содержать не более 131 символа.</p>	
См. также	FileCopy() , FileMove() , FileDelete() , FileReadMessage() , FileWriteMessage() , FileWriteFields()	

FileReadMessage()

	Считывает указанное количество байт (или целую строку) из указанного файла.	
Категория	системная	
Синтаксис	[Result=]FileReadMessage(Filename, FileOffset, Message_Tag, CharsToRead);	
	Аргумент	Описание
	Filename	Название файла.
	FileOffset	Начальная позиция в файле, с которой требуется считывать данные.
	Message_Tag	Тег, в который будут записаны прочитанные из файла данные (не более 131 символа).
	CharsToRead	Число считываемых байтов. Для текстовых файлов этот аргумент можно установить в 0. В этом случае функция будет читать файл либо до символа перевода строки LF, либо пока не прочтёт 131 символ.
Примеры	FileReadMessage ("C:\DATA\FILE.TXT", 0, MsgTag, 0); В тег MsgTag будет прочитана первая строка файла C:\DATA\FILE.TXT . Возвращаемым значением будет номер байта, непосредственно следующего за прочитанной строкой. Это значение можно использовать в качестве аргумента FileOffset в последующих вызовах функции.	
См. также	FileCopy(), FileMove(), FileDelete(), FileReadFields(), FileWriteFields(), FileWriteMessage()	

FileWriteFields()

	Заносит в указанный файл запись данных в формате CSV (Comma Separated Values – значения, разделённые запятыми)	
Категория	системная	
Синтаксис	FileWriteFields(Filename, FileOffset, StartTag, NumberOfFields);	
	Аргумент	Описание
	Filename	Название файла. Если файл с указанным названием не существует, он будет создан.
	FileOffset	Начальная позиция в файле, с которой требуется записывать данные. Если значение этого аргумента равно –1, то запись осуществляется в конец файла.
	StartTag	Название тега InTouch, в котором хранится значение <i>первого</i> элемента данных. Его название должно заканчиваться цифрой (например MyTag1). Данный аргумент должен быть строкой символов, определяющей название тега, а не самим названием. Таким образом, если тег называется MyTag1 , то в качестве аргумента необходимо указывать "MyTag1" или MyTag1.Name , а не просто MyTag1.
	NumberOfFields	Задаёт число полей для считывания (число разделённых запятыми полей в каждой записи файла).
Примечание	Если начальный тег называется MyTag1 и число полей равно 3, то в файл будет записано три поля, значения которых будут определяться тегами MyTag1 , MyTag2 и MyTag3 . Указанные теги с последовательными названиями должны быть предварительно созданы в Словаре InTouch и могут иметь различные типы (целый, сообщение и т.д.).	
Примеры	<p>При выполнении следующего скрипта строка "This is text, 3.1416, 5" будет занесена в файл C:\DATA\FILE.CSV в его первую запись. Строка "This is text" является значением тега MyTag1, число 3.1416 – тега MyTag2, а число 5 – тега MyTag3.</p> <p>FileWriteFields ("C:\DATA\FILE.CSV", 0, "MyTag1", 3);</p> <p>Выполнив запись, функция возвратит новое значение указателя файла, которое можно использовать в последующих вызовах этой функции на месте аргумента FileOffset:</p> <p>FileReadFields("C:\DATA\FILE.CSV", BytePosition, "MyTag1", 3);</p> <p>При выполнении следующего скрипта строка "This is text, 3.1416, 5" будет записана в конец файла C:\DATA\FILE.CSV.</p> <p>Строка "This is text" является значением тега MyTag1, число 3.1416 – тега MyTag2, а число 5 – тега MyTag3.</p> <p>FileWriteFields ("C:\DATA\FILE.CSV", -1, "MyTag1", 3);</p>	
См. также	FileCopy(), FileDelete(), FileMove(), FileReadFields(), FileReadMessage(). FileWriteMessage()	

FileWriteMessage()

	Записывает указанное количество байт (или целую строку) в заданный файл.	
Категория	системная	
Синтаксис	FileWriteMessage(Filename, FileOffset, Message_Tag, LineFeed);	
	Аргумент	Описание
	Filename	Название файла. Если файл с таким названием не существует, он будет создан.
	FileOffset	Начальная позиция в файле, с которой будут записаны данные. Если значение этого аргумента равно -1, то запись осуществляется в конец файла.
	Message_Tag	Строка символов, записываемая в файл.
	LineFeed	Аргумент, определяющий запись в файл символа LF (перевод строки). При записи в текстовый файл рекомендуется устанавливать его в 1.
Примечание	Функция возвращает номер позиции следующего байта в файле, который можно использовать в аргумента FileOffset в последующих вызовах этой функции.	
Примеры	Следующий оператор запишет сообщение, содержащееся в теге MsgTag в конец файла C:\DATA\FILE.TXT : FileWriteMessage ("C:\DATA\FILE.TXT", -1, MsgTag, 1);	
См. также	FileCopy(), FileDelete(), FileMove(), FileReadFields(), FileReadMessage(), FileWriteFields()	

GetAccountStatus()

	Показывает количество дней, по истечению которых перестает действовать пароль пользователя. Положительное возвращаемое функцией число показывает количество дней, в течение которых действует пароль пользователя.		
Категория	безопасность		
Синтаксис	Результат=GetAccountStatus (Domain, UserID) ;		
	Параметр	Описание	
	Domain	Имя домена или локального компьютера, на котором находится пользователь.	
	UserID	Учётное имя пользователя в операционной системе Windows, которое принадлежит компьютеру, рабочей группе или домену.	
	Результат	Могут быть возвращены следующие результаты:	
		Результат	Описание
		-1	Пароль учётной записи истёк
		-2	Пароль учётной записи бессрочный
		-3	Учётная запись заблокирована
		-4	Учётная запись отменена
		-5	Информация об учётной записи неверна
Комментарии	Эта скриптовая функция предназначена для использования в случае безопасности операционной системы. Эта функция не должна использоваться в режиме безопасности ArchestrA. Если GetAccountStatus() используется в режиме безопасности ArchestrA, скрипт будет пытаться получить учётную информацию непосредственно из домена контроллера. Это будет работать до тех пор, пока архив ArchestrA Galaxy Repository использует безопасность операционной системы с тем же доменом.		
Пример(ы)	<pre>Status = GetAccountStatus ("Corporate_HQ", "Operator") ;</pre>		

GetNodeName()

	Возвращает в символьную переменную название узла NetDDE	
Категория	системная	
Синтаксис	GetNodeName(Tagname, NodeNum);	
	Аргумент	Описание
	Tagname	Название тега InTouch типа "сообщение", в который будет записано название узла.
	NodeNum	Аргумент целого типа, значение которого равно длине названия в символах.
Примечание	При выполнении следующего скрипта функция GetNodeName() запишет в тег Tagname название локального узла, причём длина этого названия в символах будет храниться в теге NodeNum .	
Примеры	<pre>GetNodeName("MyNodeTag",131); If MyNodeTag == "Master" THEN MessageTag = "Это головной компьютер!"; ENDIF;</pre>	

GetPropertyD()

	Возвращает логическое значение указанного свойства во время исполнения программы	
Категория	GOT	
Синтаксис	[ErrorNumber=]GetPropertyD("ControlName[.Property]", Tagname);	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ChkBox_1 , либо название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	.Property	Свойство оконного объекта либо объекта вывода сведений о распределённых алармах. Перечень свойств объектов с их описаниями можно найти в Главе 2 "Поля тегов".
	Tagname	Название допустимого тега (того же типа, что и возвращаемое значение), в который будет записано значение указанного свойства в момент выполнения функции.
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
См. также	GetPropertyI(), GetPropertyM(), SetPropertyD(), SetPropertyI(), SetPropertyM()	

GetPropertyI()

	Возвращает целое значение указанного свойства во время исполнения программы	
Категория	GOT	
Синтаксис	[ErrorNumber=]GetPropertyI("ControlName[.Property]", Tagname);	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ChkBox_1 , либо название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	.Property	Свойство оконного объекта либо объекта вывода сведений о распределённых алармах. Перечень свойств объектов с их описаниями можно найти в Главе 2 "Поля тегов".
	Tagname	Название допустимого тега (того же типа, что и возвращаемое значение), в который будет записано значение указанного свойства в момент выполнения функции.
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
См. также	GetPropertyD(), GetPropertyM(), SetPropertyD(), SetPropertyI(), SetPropertyM()	

GetPropertyM()

	Возвращает значение указанного свойства в виде строки символов во время исполнения программы	
Категория	GOT	
Синтаксис	[ErrorNumber=]GetPropertyM("ControlName[.Property]", Tagname);	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ChkBox_1 , либо название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	.Property	Свойство оконного объекта либо объекта вывода сведений о распределённых алармах. Перечень свойств объектов с их описаниями можно найти в Главе 2 "Поля тегов".
	Tagname	Название допустимого тега (того же типа, что и возвращаемое значение), в который будет записано значение указанного свойства в момент выполнения функции.
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
См. также	GetPropertyD(), GetPropertyI(), SetPropertyD(), SetPropertyI(), SetPropertyM()	

Hide

	Скрывает (делает невидимыми) различные окна во время исполнения Quick-скрипта. Название закрываемого окна указывается после названия функции	
Категория	прочие	
Синтаксис	Hide Window	
	Аргумент	Описание
	Window	Название окна либо тег типа "сообщение", в котором оно хранится.
Примечание	<p>Если указанное окно во время исполнения программы не существует во время выполнения, данный оператор игнорируется. Если требуется лишь скрывать и показывать окна, то вместо этой функции рекомендуется пользоваться кнопками типа Show Window ("Показать окно") и Hide Window ("Скрыть окно").</p> <p>Если в качестве аргумента функции будет указан локальный тег, то на экране появится следующее сообщение об ошибке: "Expecting window name – must be string expression" (Отсутствует название окна – ожидается строка символов).</p> <p>Во избежание этой ошибки перед названием тега типа "сообщение" необходимо поставить знак плюса "+", например: DIM Test AS message; Test = "MyWindow1"; Hide ""+Test;</p>	
Примеры	Hide "My Popup Alarm Window";	
См. также	HideseIf, Show, ShowAt(), ShowTopLeftAt(), ShowHome	

HideSelf

	Скрывает текущее активное окно	
Категория	прочие	
Синтаксис	HideSelf;	
Примечание	Эту функцию можно вызывать только в скриптах, запускаемых при нажатии на экранные кнопки.	
См. также	Hide, Show	

HTGetLastError()

	Определяет, возникла ли ошибка во время последнего считывания данных для указанного пера указанного архивного тренда		
Категория	архивирование		
Синтаксис	[Result=]HTGetLastError(Hist_Tag, UpdateCount, PenNum);		
	Аргумент	Описание	
	Hist_Tag	Название тега типа "архивный тренд".	
	UpdateCount	Аргумент целого типа, соответствующего полю тренда .UpdateCount .	
	PenNum	Целое число либо тег целого типа, определяющий номер пера (от 1 до 8)	
	Result	Возвращаемая величина, которая может иметь следующие значения:	
		0	Ошибок не было.
		1	Общая ошибка сервера.
		2	Устаревший запрос.
		3	Ошибка файла.
		4	Сервер не загружен.
		5	Тренд или перо, указанные в аргументах функции, не существует.
		6	Тег тренда с таким названием не существует в Словаре.
	7	Недопустимый номер пера (не в диапазоне от 1 до 8).	
	8	Ошибка указания тренда для данного номера пера.	
Примечание			
Примеры	<p>Следующий оператор возвращает в тег ResultCode целого типа код ошибки последнего считывания данных для пера Pen3 архивного тренда Trend1:</p> <pre>[ResultCode=]HTGetLastError("Trend1", Trend1.UpdateCount, 3);</pre> <p>В анимационной функции вывода аналогового значения может быть использовано следующее выражение:</p> <pre>HTGetLastError("Trend1", Trend1.UpdateCount, 3);</pre>		

HTGetPenName()

	Возвращает название тега, значения которого выводятся указанным пером указанного тренда	
Категория	архивирование	
Синтаксис	MessageResult=HTGetPenName(Hist_Tag, UpdateCount, PenNum);	
	Аргумент	Описание
	Hist_Tag	Название тега типа "архивный тренд".
	UpdateCount	Аргумент целого типа, соответствующего полю тренда .UpdateCount .
	PenNum	Целое число либо тег целого типа, определяющий номер пера (от 1 до 8)
	MessageResult	Название тега типа "сообщение", в который будет записано название тега, значения которого выводятся указанным пером указанного тренда
Примеры	<p>Следующий оператор определяет название тега, связанного с пером Pen2 тренда Trend1 и помещает результат в тег TrendPen типа "сообщение":</p> <pre>TrendPen=HTGetPenName("Trend1", Trend1.UpdateCount, 2);</pre>	

HTGetTimeAtScooter()

	Возвращает число секунд, прошедших с 00:00:00 часов по гринвичскому времени 1 января, 1970 года, соответствующих выборке тренда в позиции визира, определяемом аргументами ScootNum и ScootLoc .	
Категория	архивирование	
Синтаксис	IntegerResult=HTGetTimeAtScooter(Hist_Tag, UpdateCount, ScootNum, ScootLoc);	
	Аргумент	Описание
	Hist_Tag	Название тега типа "архивный тренд".
	UpdateCount	Аргумент целого типа, соответствующего полю тренда .UpdateCount .
	ScootNum	Аргумент целого типа, определяющий визир: 1 = левый, 2 = правый.
	ScootLoc	Аргумент вещественного типа, имеющий значение полей тренда .ScooterPosRight либо .ScooterPosLeft .
Примечание	Любое изменение тегов UpdateCount , ScootNum или ScootLoc приводит к очередному выполнению функции, благодаря чему информация будет обновляться при каждом считывании новых данных либо перемещении какого-либо из визиров.	
Примеры	Следующий оператор определяет время в секундах для выборки тренда, указанной текущей позицией левого визира тренда с названием Trend1 : HTGetTimeAtScooter("Trend1", Trend1.UpdateCount, 1, Trend1.ScooterPosLeft);	

HTGetTimeStringAtScooter()

	Возвращает дату и время выборки тренда в позиции визира. Номер и позиция визира определяются значениями аргументов ScootNum и ScootLoc .	
Категория	архивирование	
Синтаксис	MessageResult=HTGetTimeStringAtScooter(Hist_Tag, UpdateCount, ScootNum, ScootLoc, Format_Text);	
	Аргумент	Описание
	Hist_Tag	Название тега типа "архивный тренд".
	UpdateCount	Аргумент целого типа, соответствующего полю тренда .UpdateCount .
	ScootNum	Аргумент целого типа, определяющий визир: 1 = левый, 2 = правый.
	ScootLoc	Аргумент вещественного типа, имеющий значение полей тренда .ScooterPosRight либо .ScooterPosLeft .
	Format_Text	Строка форматирования даты и времени. Допустимы следующие значения этого аргумента: "Date", "Time", "DateTime", "DOWShort" (например Wed) и "DOWLong" (например Wednesday).
Примечание	Любое изменение тегов UpdateCount , ScootNum или ScootLoc приводит к очередному выполнению функции, благодаря чему информация будет обновляться при каждом считывании новых данных либо перемещении какого-либо из визиров. Вид возвращаемой строки определяется указанной строкой форматирования.	
Примеры	Следующий оператор возвращает дату и время выборки тренда, указанной текущей позицией правого визира тренда с названием Trend1 . Результат записывается в тег NewRightTimeString типа "сообщение" в формате "Time": NewRightTimeString=HTGetTimeStringAtScooter("Trend1", Trend1.UpdateCount, 2, Trend1.ScooterPosRight, "Time");	

HTGetValue()

	Возвращает значение указанного типа, вычисленное для указанного пера для всего временного диапазона тренда. После считывания данных тренда изменение поля UpdateCount приведёт к повторному вычислению выражения	
Категория	архивирование	
Синтаксис	RealResult=HTGetValue(Hist_Tag, UpdateCount, PenNum, ValType_Text);	
	Аргумент	Описание
	Hist_Tag	Название тега типа "архивный тренд".
	UpdateCount	Аргумент целого типа, соответствующего полю тренда .UpdateCount .
	PenNum	Номер пера тренда (от 1 до 8).
	ValType_Text	Строка, определяющая тип возвращаемого значения:
	"PenAverageValue"	Среднее для всего тренда.
	"PenMaxValue"	Максимальное для тренда.
	"PenMinValue"	Минимальное для тренда.
	"PenMaxEU"	Максимальное для тренда в единицах изменения.
	"PenMinEU"	Минимальное для тренда в единицах изменения.
	"PenStdDev"	Величина стандартного отклонения для тренда.
	При использовании аргумента ValType_Text необходимо, чтобы он имел только одно из указанных значений.	
Примечание	Функция возвращает вещественное число в соответствии с указанным типом расчёта для тренда.	
Примеры	Следующий оператор возвращает величину стандартного отклонения данных пера Pen2 тренда Trend1 , которая записывается во внутренний тег LeftHemisphereSD вещественного типа: LeftHemisphereSD=HTGetValue("Trend1", Trend1.UpdateCount, 2, "PenStdDev");	

HTGetValueAtScooter()

	Возвращает значение указанного типа, вычисленное для выборки указанного тренда, определяемой позицией указанного визира на кривой указанного пера. После считывания данных
--	--

	тренда изменение поля UpdateCount приведёт к повторному вычислению выражения		
Категория	архивирование		
Синтаксис	RealResult=HTGetValueAtScooter(Hist_Tag, UpdateCount, ScootNum, ScootLoc, PenNum, ValType_Text);		
	Аргумент	Описание	
	Hist_Tag	Название тега типа "архивный тренд".	
	UpdateCount	Аргумент целого типа, соответствующего полю тренда .UpdateCount .	
	ScootNum	Аргумент целого типа, определяющий визир: 1 = левый, 2 = правый.	
	ScootLoc	Аргумент вещественного типа, имеющий значение полей тренда .ScooterPosRight либо .ScooterPosLeft .	
	PenNum	Номер пера тренда (от 1 до 8).	
	ValType_Text	Строка, определяющая тип возвращаемого значения:	
		"PenValue"	Значение в позиции визира.
		"PenValid"	0, если значение недействительно, 1, если действительно.
	При использовании аргумента ValType_Text необходимо, чтобы он имел только одно из указанных значений.		
Примечание	Функция возвращает вещественное число в соответствии при указании аргумента ValType_Text "PenValue" и логическое значение при указании аргумента ValType_Text "PenValid".		
Примеры	Следующий оператор записывает "1" во внутренний тег ValidFlag логического типа 1, если значение тренда Trend1 в текущей позиции правого визира для пера Pen3 соответствует реальной выборке, и 0, если значение недопустимо: ValidFlag=HTGetValueAtScooter("Trend1", Trend1.UpdateCount, 2, Trend1.ScooterPosRight, 3, "PenValid");		

HTGetValueAtZone()

	Возвращает значение указанного типа для указанного пера, вычисленное для диапазона тренда между левым и правым визирами. После считывания данных тренда изменение поля UpdateCount приведёт к повторному вычислению выражения	
Категория	архивирование	
Синтаксис	RealResult=HTGetValueAtZone(Hist_Tag, UpdateCount, Scoot1Loc, Scoot2Loc, PenNum, ValType_Text);	
	Аргумент	Описание
	Hist_Tag	Название тега типа "архивный тренд".
	UpdateCount	Аргумент целого типа, соответствующего полю тренда .UpdateCount .
	PenNum	Номер пера тренда (от 1 до 8).
	Scoot1Loc	Аргумент вещественного типа, соответствующий полю тренда .ScooterPosLeft .
	Scoot2Loc	Аргумент вещественного типа, соответствующий тренда .ScooterPosRight .
	ValType_Text	Строка, определяющая тип возвращаемого значения:
	"PenAverageValue"	Среднее для диапазона.
	"PenMaxValue"	Максимальное для диапазона.
	"PenMinValue"	Минимальное для диапазона.
	"PenMaxEU"	Максимальное для диапазона в единицах изменения.
	"PenMinEU"	Минимальное для диапазона в единицах изменения.
	"PenStdDev"	Величина стандартного отклонения для диапазона.
		При использовании аргумента ValType_Text необходимо, чтобы он имел только одно из указанных значений.
Примечание	Функция возвращает вещественное число в соответствии с указанным типом расчёта для тренда. Указание конкретных значений аргументов Scoot1Loc и Scoot2Loc ни на что не влияет и используется только для запуска функции с целью обновления линий графика. Для определения границ диапазона функция извлекает значения полей тренда .ScooterPosLeft и .ScooterPosRight непосредственно из базы данных.	

Примеры	<p>Следующий оператор вычисляет среднее значение данных пера Pen1 тренда Trend1, находящихся между левым и правым визирами возвращает величину стандартного отклонения данных пера Pen2 тренда, которая записывается во внутренний тег AvgValue вещественного типа:</p> <pre>AvgValue=HTGetValueAtZone("Trend1", Trend1.UpdateCount, Trend1.ScooterPosLeft, Trend1.ScooterPosRight, 1, "PenAverageValue");</pre>
----------------	---

HTScrollLeft()

	<p>Изменяет начальное времени тренда на более раннее, чем текущее, в процентах от ширины тренда. Результатом выполнения этой функции будет прокручивание графика влево по оси времени на заданную величину.</p>	
Категория	архивирование	
Синтаксис	HTScrollLeft(Hist_Tag, Percent);	
	Аргумент	Описание
	Hist_Tag	Название тега типа "архивный тренд".
	Percent	Величина смещения в процентах от ширины графика (от 0,0 до 100,0).
Примеры	<p>Следующий оператор смещает тренд Trend1 на 10%: HTScrollLeft("Trend1", 10.0);</p> <p>Если начальное время тренда равно 12:00:00 PM и ширина тренда составляет 60 секунд, то после выполнения функции начальное время нового тренда будет 11:59:54 AM.</p>	

HTScrollRight()

	Изменяет начальное времени тренда на более позднее, чем текущее, в процентах от ширины тренда. Результатом выполнения этой функции будет прокручивание графика вправо по оси времени на заданную величину.	
Категория	архивирование	
Синтаксис	HTScrollRight(Hist_Tag, Percent);	
	Аргумент	Описание
	Hist_Tag	Название тега типа "архивный тренд".
	Percent	Величина смещения в процентах от ширины графика (от 0,0 до 100,0).
Примеры	<p>Следующий оператор смещает тренд Trend1 на 20%: HTScrollRight("Trend1", 20.0);</p> <p>Если начальное время тренда равно 12:00:00 PM и ширина тренда составляет 60 секунд, то после выполнения функции начальное время нового тренда будет 12:00:12 PM.</p>	

HTSelectTag()

	Открывает диалоговое окно Select Tag (Выбор тегов) для выбора иного тега для указанного пера. (В списке перечислены только те теги, для которых в Словаре тегов предусмотрена архивная регистрация (опция Log Data разрешена)).
Категория	архивирование
Синтаксис	HTSelectTag()
Примечание	Функция HTSelectTag() показывает ВСЕ теги с разрешённой опцией архивной регистрации (опция Log Data). Вместе с тем, воспользовавшись фильтрами браузера, на экран можно выводить меньший набор (например названия тегов, начинающиеся с буквы "A"). Функция HTSelectTag() НЕ позволяет отображать перечень всех тегов – выводятся только теги, для которых регистрация их значений в архиве разрешена.
Примеры	<p>Следующий оператор может использоваться в кнопочных скриптах. При нажатии на кнопку откроется окно браузера тегов, в котором пользователь сможет выбрать название требуемого тега. После этого перо 1 тренда HistTrend будет выводить на график значения выбранного тега:</p> <p>HTSetPenName("HistTrend", 1, HTSelectTag ());</p>
См. также	HTSetPenName()

HTSetPenName()

	Связывает с указанным пером тренда указанный тег	
Категория	архивирование	
Синтаксис	HTSetPenName(Hist_Tag, PenNum, Tagname);	
	Аргумент	Описание
	Hist_Tag	Название тренда типа "архивный тренд".
	PenNum	Аргумент целого типа, определяющий номер пера (от 1 до 8).
	Tagname	Строка, определяющая название нового тега.
Примечание	Данная функция представляет собой единственный способ извлечения значений новых тегов из провайдера распределённого архива во время исполнения программы.	
Примеры	<p>В следующем операторе перу Pen3 тренда Trend1 назначается новый тег с названием OutletPressure:</p> <pre>HTSetPenName("Trend1", 3, "OutletPressure");</pre> <p>В следующем операторе перу Pen4 тренда Trend1 назначается удалённый тег с названием HistPrv1.Tag1:</p> <pre>HTSetPenName("Trend1", TrendPen4, "HistPrv1.Tag1");</pre>	
См. также	HTSelectTag()	

HTUpdateToCurrentTime()

	Вызывает считывание данных и вывод их на дисплей с конечным временем, равным текущему. Начальное время вычисляется путём вычитания ширины тренда из его конечного времени	
Категория	архивирование	
Синтаксис	HTUpdateToCurrentTime(Hist_Tag);	
	Аргумент	Описание
	Hist_Tag	Название тега типа "архивный тренд"
Примеры	<p>Следующий оператор считывает с диска и выводит на экран данные для архивного тренда Trend1, соответствующие текущему моменту времени:</p> <pre>HTUpdateToCurrentTime("Trend1");</pre> <p>Если выполнение функции пришлось на 3:04 PM и ширина тренда составляет 60 секунд, то новым значением конечного времени будет 3:04 PM. Новым начальным временем будет 3:03 PM.</p>	

HTZoomIn()

	Вычисляет новую ширину и начальное время графика тренда. Если значение .ScooterPosLeft равно 0.0, а .ScooterPosRight – 1.0, то ширина нового графика будет равна половине старой ширины. Новое начальное время вычисляется с учётом значения аргумента LockString	
Категория	архивирование	
Синтаксис	HTZoomIn(Hist_Tag, LockString);	
	Аргумент	Описание
	Hist_Tag	Название тега типа "архивный тренд"
	LockString	Тип изменения масштаба графика:
	"StartTime"	Сохранение исходного начального времени тренда.
	"Center"	Сохранение центральной точки тренда.
	"EndTime"	Сохранение исходного конечного времени тренда.
Примечание	Если визеры находятся не на краях графика, то новая ширина графика станет равной интервалу времени между позициями, определяемыми значениями полей .ScooterPosLeft и .ScooterPosRight . В этом случае значение аргумента LockString не используется. Минимальная ширина графика равна одной секунде. После масштабирования графика позиции визера будут установлены на края графика, а поля .ScooterPosLeft и .ScooterPosRight примут значения 0.0 и 1.0 соответственно.	
Примеры	Следующий оператор растягивает график тренда Trend1 в 2 раза, сохраняя без изменения его начальное время. Поле Trend1.ScooterPosRight равно 1.0, а поле Trend1.ScooterPosLeft равно 0.0. Если начальное время графика перед растяжением было 1:25:00 PM, а его ширина – 30 секунд, то начальное время нового графика будет прежним – 1:25:00, а ширина графа станет равной 15 секундам: HTZoomIn("Trend1", "StartTime");	
См. также		

HTZoomOut()

	Вычисляет новые ширину и начальное время графика тренда. Ширина нового графика будет равна удвоенной ширине предыдущего графика. Новое начальное время вычисляется с учётом значения аргумента LockString		
Категория	архивирование		
Синтаксис	HTZoomOut(Hist_Tag, LockString);		
	Аргумент	Описание	
	Hist_Tag	Название тега типа "архивный тренд"	
	LockString	Тип изменения масштаба графика:	
		"StartTime"	Сохранение исходного начального времени тренда.
		"Center"	Сохранение центральной точки тренда.
		"EndTime"	Сохранение исходного конечного времени тренда.
Примечание	Исходные положения визиров не влияют на изменение масштаба графика, однако после выполнения функции будут установлены на края графика (поле .ScooterPosLeft станет равным 0.0, а поле .ScooterPosRight – 1.0).		
Примеры	Следующий оператор изменяет масштаб графика тренда Volume в два раза, сохраняя без изменения время его центральной точки. Если начальное время исходного графика было 2:15:00 PM, а ширина графика составляла 30 секунд, то после выполнения функции начальное время изменится на 2:14:45. Центральное же время останется прежним – 2:15:15. HTZoomOut("Volume","Center");		

InfoAppActive()

	Проверяет, является ли указанное приложение активным	
Категория	системная	
Синтаксис	DiscreteResult=InfoAppActive(AppTitle);	
	Аргумент	Описание
	AppTitle	Строка с названием приложения
Примеры	<p>InfoAppActive("Calculator") возвращает 0, если программа калькулятора в текущий момент не выполняется.</p> <p>Название приложения можно узнать с помощью функции InfoAppTitle().</p> <p>В следующем примере анализируется список активных задач в системе. Если задача с названием "Notepad" в текущий момент выполняется, то функция InfoAppActive() возвратит 1. Таким образом можно избежать запуска второго экземпляра программы Notepad. Если функция InfoAppActive() возвращает 0 (т.е. приложение Notepad не выполняется), то осуществляется запуск требуемой программы.</p> <pre>IF InfoAppActive(InfoAppTitle("Notepad")) == 1 THEN ActivateApp InfoAppTitle("Notepad"); ELSE StartApp "Notepad"; ENDIF;</pre>	

InfoAppTitle()

	Возвращает из списка задач Windows название выполняющейся в текущий момент программы	
Категория	системная	
Синтаксис	MessageResult=InfoAppTitle("ProgramEXENAME");	
	Аргумент	Описание
	ProgramEXENAME	Название выполняющегося файла программы.
Примеры	<p>Следующий оператор возвращает строку "Calculator" для запущенной программы calc.exe. Внимание: расширение .exe в аргументе ProgramEXENAME указывать не надо.</p> <p>InfoAppTitle("calc") возвратит "Calculator"</p> <p>InfoAppTitle("excel") возвратит "Microsoft Excel"</p>	

InfoDisk()

	Возвращает информацию об указанном локальном или сетевом дисковом устройстве	
Категория	системная	
Синтаксис	IntegerResult=InfoDisk("Drive", InfoType, Trigger);	
	Аргумент	Описание
	Drive	Строка или тег типа "сообщение" с обозначением диска.
	InfoType	Аргумент целого типа, определяющий тип возвращаемой информации:
	1	Общий объём диска в байтах.
	2	Доступное пространство на диске в байтах.
	3	Общий объём диска в килобайтах.
	4	Доступное пространство на диске в килобайтах.
	Trigger	Название тега, при каждом изменении значения которого данная функция будет запускаться на исполнение. Этот тег может быть любым допустимым, а не только системным, тегом. Данный параметр используется <u>только</u> в поле выражения анимационной функции, поэтому в скрипте в вызове функции InfoDisk() может указываться любой фиктивный аргумент, потому что внутри скрипта он никакого значения не имеет.
Примечание	Функция, возвращающая сведения об указанном диске в тег целого типа.	
Примеры	<p>Следующие операторы будут выполняться ежеминутно, возвращая текущее значение объёма диска. Например:</p> <p>InfoDisk("C", 1, \$Minute) возвращает 233869345 {общий объём в байтах}</p> <p>InfoDisk("C", 2, \$Minute) возвращает 3238935 {свободный объём в байтах}</p> <p>InfoDisk("C", 3, \$Minute) возвращает 228388 {общий объём в килобайтах}</p> <p>InfoDisk("C", 4, \$Minute) возвращает 3163 {свободный объём в килобайтах}</p> <p>(1 килобайт = 1024 байт)</p>	
Примечание	Как и в других функциях, в которых в аргументах должны указываться только единичные символы, если тег типа "сообщение", передаваемый функции InfoDisk() в аргументе обозначения диска, будет содержать более одного символа, то использован будет только первый из них. Поскольку эта функция возвращает значения, получаемую от операционной системы, то они могут быть ошибочными в среде Windows 95 с дисками больше 2 гигабайт.	

InfoFile()

	Возвращает информацию об указанном файле или каталоге	
Категория	системная	
Синтаксис	IntegerResult=InfoFile("Filename", "InfoType", Trigger);	
	Аргумент	Описание
	Filename	Строка или тег типа "сообщение" с названием файла.
	InfoType	Целый аргумент, определяющий тип возвращаемой информации:
	1	<i>Существует ли файл?</i> Функция возвращает 1, если указанное название соответствует реальному файлу, 2, если подкаталогу, и 0, если файл с таким названием не найден.
	2	Размер файла в байтах.
	3	Дата создания файла (в виде числа секунд, прошедших с первого января 1970 года).
	4	Число файлов, соответствующих указанному описанию. Значения, большие чем 1, возвращаются, только если указанному шаблону названия файла с символами подстановки соответствуют несколько файлов.
Trigger	Название тега, при каждом изменении значения которого данная функция будет запускаться на исполнение. Этот тег может быть любым допустимым, а не только системным, тегом. Данный параметр используется <u>только</u> в поле выражения анимационной функции, поэтому в скрипте в вызове функции InfoDisk() может указываться любой фиктивный аргумент, потому что внутри скрипта он никакого значения не имеет.	
Примечание	Функция возвращает в указанный тег целого типа сведения о файле с указанным названием. В названии при этом могут использоваться символы подстановки "?" и "*".	
Примеры	Следующие операторы будут выполняться ежеминутно, возвращая текущее значение указанного типа. Например: InfoFile("c:\IT56\view.exe", 1, \$Minute) возвратит 1 {файл найден} InfoFile("c:\InTouch\view.exe", 2, \$Minute) возвратит 634960 {размер файла} InfoFile("c:\InTouch\view.exe", 3, \$Minute) возвратит 736701852 {секунд с 1-1-70} InfoFile("c:\InTouch*.exe", 4, \$Minute) возвратит 17 {найдено 17 EXE-файлов}	

InfoInTouchAppDir()

	Возвращает название каталога текущего приложения InTouch
Категория	системная
Синтаксис	MessageResult=InfoInTouchAppDir();
Примечание	В тег MessageResult возвращается строка, содержащая каталог текущего приложения InTouch.
Примеры	InfoInTouchAppDir() ; возвратит "c:\ProgramFiles\FactorySuite\InTouch\DemoApp1\1024"

InfoResources()

	Возвращает значения различных системных ресурсов	
Категория	системная	
Синтаксис	IntegerResult=InfoResources(ResourceType, Trigger);	
	Аргумент	Описание
	ResourceType	Аргумент целого типа, определяющий требуемый системный ресурс
Примечание	Функция возвращает в тег целого типа значение указанного системного ресурса.	
Примеры	<p>Следующие операторы будут выполняться ежеминутно, возвращая текущее значение ресурса указанного типа. Например:</p> <p>InfoResources(1, \$Minute) возвращает 54 {процентов свободно}</p> <p>InfoResources(2, \$Minute) возвращает 36 {процентов свободно}</p> <p>InfoResources(3, \$Minute) возвращает 11524093 {байт}</p> <p>InfoResources(4, \$Minute) возвращает 14 {задач}</p> <p>В 1 и 2 случаях в системах Windows NT и 2000 для ресурсов GDI и USER функция всегда будет возвращать значения 50 %.</p> <p>В третьем случае в системах Windows NT и 2000 функция возвратит "число свободных байтов в файле подкачки".</p> <p>В четвёртом случае в системах Windows NT и 2000 функция возвращает число всех окон верхнего уровня. Подсчитываются окна, видимые на экране и не имеющие владельцев.</p> <p>Возвращаемое значение не является числом активных задач в системе. В операционной системе Windows NT оно приблизительно равно числу элементов на странице с закладкой Applications (приложения) в окне Менеджера программ, а в системе Windows 2000 – числу элементов в списке программ для закрытия при нажатии на клавиши Ctrl+Alt+Del.</p>	

Int()

	Возвращает ближайшее целое, не превышающее указанное число	
Категория	математическая	
Синтаксис	IntegerResult=Int(Number);	
	Аргумент	Описание
	Number	Любое число или тег целого или вещественного типа.
Примечание	При указании отрицательного числа, функция возвратит наибольшее по модулю целое значение.	
Примеры	Int(4.7) возвращает 4 Int(-4.7) возвращает -5	

InTouchVersion()

	Возвращает строку, зависящую от значения VERSION_TYPE. Пользователи могут ввести одно из 5 значений от 0 до 4 (включая оба крайних). Если пользователь вводит отрицательное значение или целое, большее, чем 4, будет показана полная строка версии по умолчанию.	
Категория	Представление	
Синтаксис	InTouchVersion(VersionType)	
	Параметр	Описание
	VersionType	Входной (целое)
	0	Будет показана полная версия.
	1	Будет показано значение главной версии ("9").
	2	Будет показано значение младшей версии ("0").
	3	Пакет обновления ("1"). Для InTouch SP1 пакет обновления равен 1.
	4	Будет показана версия сборки.
Комментарии	Нет.	
Пример(ы)	Mtag = IntouchVersion (nVersionType); где Mtag является тэгом сообщения памяти и nVersionType является целым значением.	

InvisibleVerifyCredentials()

	Выполняет проверку полномочий пользователя с указанными параметрами без его регистрации в системе InTouch	
Категория	контроль доступа	
Синтаксис	AnalogTag=InvisibleVerifyCredentials("UserId", "Password", "Domain");	
	Аргумент	Описание
	UserId	Имя пользователя операционной системы Windows в локальной машине, рабочей группе либо домене
Примечание	Если указанное сочетание имени пользователя, пароля и домена действительны, то функция возвращает уровень доступа данного пользователя (целое число), во всех остальных случаях возвращается значение -1. Вызов функции не приводит к смене текущего зарегистрированного пользователя.	
Примеры	AnalogTag=InvisibleVerifyCredentials("john", "Password", "corporate_hq");	
См. также	PostLogonDialog(), AttemptInvisibleLogon(), IsAssignedRole(), QueryGroupMembership(), AddPermission()	

IODisableFailover()

	Эта функция разрешает/запрещает переключение сбоя для имени доступа.	
Категория	Связь в/в	
Синтаксис	IODisableFailover("AccessName", Option);	
	Параметр	Описание
	AccessName	Имя доступа, для которого настраивается переключение сбоя.
	Option	1 = запрещение failover 0 = разрешение failover
Комментарии	Имя доступа может быть указано как строка символов, или оно может быть строковым значением, предоставляемым другими тэгами или функциями InTouch. Параметр "1" запрещает переключение сбоя, параметр "0" – разрешает его.	
Пример(ы)	IODisableFailOver("Acc1", 1); запрещает переключение сбоя для имени доступа "Acc1".	

IOForceFailover()

	Воникает переключение сбоя и вступает в работу вторичный источник в/в, определённый для первичного имени доступа.	
Категория	Связь в/в	
Синтаксис	IOForceFailover (AccessName) ;	
	Параметр	Описание
	AccessName	Имя доступа, для которого настраивается переключение сбоя.
Комментарии		
Пример(ы)	Предположим, что "Accl" имеет 2 источника, "Primary" (первичный) и "Secondary" (вторичный), и Primary активен. Тогда IOForceFailOver("Accl") переключает активный источник "Primary" на "Secondary".	

IOGetAccessNameStatus()

	Возвращает статус первичного, вторичного или активного источника в/в для имени доступа. Имя доступа может задаваться в виде строки символов, или оно может строковым значением, задаваемым другими тэгами или функциями InTouch.	
Категория	дополнительная	
Синтаксис	Результат=IOGetAccessNameStatus (Accessname, Mode) ;	
	Параметр	Описание
	AccessName	Существующее имя доступа, для которого возвращается имя источника.
	Mode	0 – статус активного источника в/в для AccessName. 1 – статус первичного источника в/в для AccessName. 2 – статус вторичного источника в/в для AccessName.
	Результат	-1 – Имеется ошибка конфигурации (AccessName не существует или запрошен статус вторичного источника и AccessName не сконфигурирован на вторичный источник). 0 – Соединение к запрашиваемому источнику не в нормальном состоянии. 1 – Соединение к запрашиваемому источнику в нормальном состоянии.
Комментарии	Эта функция опеределяет статус вторичного источника, который в настоящее время не активен. Оператор может таким способом, прежде чем форсировать преоболение сбоя, убедиться, что вторичное соединение в порядке.	
Пример(ы)	Status=IOGetAccessNameStatus ("Accl", 0) ; где "Accl" – имя доступа.	

IOGetApplication()

	Возвращает название приложения в определении указанного канала доступа	
Категория	прочие	
Синтаксис	IOGetApplication("AccessName");	
	Аргумент	Описание
	AccessName	Канал доступа, по которому определяется название приложения.
Примечание	В качестве аргумента AccessName может указываться как литерал, так и тег типа "сообщение" либо вызов функции. В следующем примере название приложения канала доступа MyAccess1 возвращается в тег MyTag1 :	
Примеры	MyTag1 = IOGetApplication("MyAccess1");	

IOGetNode()

	Возвращает сведения об узле (адрес) для указанного канала доступа	
Категория	прочие	
Синтаксис	IOGetNode("AccessName");	
	Аргумент	Описание
	AccessName	Канал доступа, по которому определяется адрес узла.
Примечание	В качестве аргумента AccessName может указываться как литерал, так и тег типа "сообщение" либо вызов функции. В следующем примере сведения об узле для канала доступа MyAccess1 возвращаются в тег MyTag1 :	
Примеры	MyTag1 = IOGetNode("MyAccess1");	

IOGetTopic()

	Возвращает название группы данных для указанного канала доступа	
Категория	прочие	
Синтаксис	IOGetTopic("AccessName");	
	Аргумент	Описание
	AccessName	Канал доступа, по которому определяется название группы данных.
Примечание	В качестве аргумента AccessName может указываться как литерал, так и тег типа "сообщение" либо вызов функции. В следующем примере название группы данных для канала доступа MyAccess1 возвращается в тег MyTag1 :	
Примеры	MyTag1 = IOGetTopic("MyAccess1");	

IOReinitAccessName()

	Повторно инициализирует указанное имя доступа с установками по умолчанию.	
Категория	СоединениеВВ	
Синтаксис	IOReinitAccessName("AccessName", Default);	
	Параметр	Описание
	AccessName	Имя доступа, которое должно повторно инициализироваться.
	Default	Default = 1. Имя доступа будет использовать установки по умолчанию, как они заданы конфигурацией в WindowMaker. Default = 0. Имя доступа будет повторно инициализироваться, но будет продолжать использовать текущие установки узла, приложения и раздела.
Комментарии	Установки по умолчанию определяются установками в панели конфигурирования имени доступа, а также в конфигурации WindowViewer: Retry Initiates (повторить инициализации), Start Local Servers (запустить локальные серверы), Reinitialize Default (повторно инициализировать по умолчанию).	
Пример(ы)	IOReinitAccessName("AccessName1", 1); В этом случае "AccessName1" будет использовать установки по умолчанию. IOReinitAccessName("AccessName2", 0); В этом случае "AccessName2" будет повторно инициализироваться с текущими установками.	

IOReinitialize()

	Закрывает все существующие каналы ввода/вывода и запускает с самого начала процесс установления каналов ввода/вывода. Данная функция воздействует на все точки ввода/вывода
Категория	прочие
Синтаксис	IOReinitialize()
Примечание	Данная функция выполняет те же самые действия, что и команда Reinitialize I/O (Снова открыть все каналы в/в) меню Special (Специальные) приложения WindowViewer.

IOSetAccessName()

	Во время исполнения программы изменяет название приложения и название группы данных в определении канала доступа, позволяя тем самым реализовать политику "горячего" резервирования приложения InTouch	
Категория	прочие	
Синтаксис	IOSetAccessName("AccessName", "NodeName", "AppName", "TopicName");	
	Аргумент	Описание
	AccessName	Название канала доступа, в определении которого требуется изменить название приложения или группы данных. Литерал или название тега типа "сообщение".
	NodeName	Новое название узла. Литерал или название тега типа "сообщение". Для сохранения текущего названия необходимо указать пустую строку "". Для ссылки на локальный узел необходимо ввести символ пробела в кавычках: " ".
	AppName	Новое название приложения. Литерал или название тега типа "сообщение".
	TopicName	Новое название группы данных. Литерал или название тега типа "сообщение".
Примечание	Значения всех аргументов функции могут быть заданы как в виде литералов, так и названий тегов или функций InTouch. Следующий оператор изменяет названия приложения и группы данных канала доступа MyAccess1 соответственно на "EXCEL" и "Sheet1":	

Примеры	<pre>IOSetAccessName("MyAccess1","MyComputer1", "EXCEL", "[Book1.xls]Sheet1");</pre> <p>Этот оператор аналогичен следующим:</p> <pre>Number = 1; AccNameString = "MyAccess" + Text(Number, "#"); IOSetAccessName(AccNameString, "", "EXCEL", "[Book1.xls]Sheet1");</pre> <p>Если в качестве аргумента TopicName будет указана пустая строка, то в канале доступа будет изменено только название приложения, а название группы данных останется без изменений. Например, следующий оператор меняет название приложения в канале доступа MyAccess2 на "excel", не меняя при этом название группы данных:</p> <pre>IOSetAccessName("MyAccess2", "", "excel", "");</pre> <p>Аналогичным образом, если пустая строка указана на месте аргумента AppName, то изменено будет только название группы данных. Следующий оператор меняет название группы данных в канале доступа MyAccess2 на "Sheet2":</p> <pre>IOSetAccessName("MyAccess3", "", "", "[Book1.xls]Sheet2");</pre> <p>Выполнение функции сопровождается небольшой задержкой, связанной с закрытием текущего канала доступа и установлением нового. Во время этой задержки все данные, передаваемые командами РОКЕ или операциями записи, будут потеряны.</p>
----------------	---

IOSetItem()

	Изменяет название канала доступа и/или элемента данных в поле .Reference тега для ввода/вывода	
Категория	прочие	
Синтаксис	IOSetItem("TagName", "AccessName", "Item");	
	Аргумент	Описание
	TagName	Название любого тега InTouch (заключённое в кавычки).
	AccessName	Новое название канала доступа для указанного тега.
	Item	Новое название элемента данных для указанного тега.
Примеры	<p>IOSetItem(TagName, AccessName, Item)</p> <p>Все аргументы функции могут представлять собой как литералы, так и названия других тегов или функций InTouch с символьными значениями. Например, чтобы поле .Reference тега MyTag1 указывало на канал доступа "excel" и элемент "R1C1", необходимо выполнить следующий оператор: IOSetItem("MyTag1", "excel", "R1C1");</p> <p>Он аналогичен следующим операторам: Number = 1; TagNameString = "MyTag" + Text(Number, "#"); IOSetItem(TagNameString, "excel", "R1C1");</p> <p>Если в качестве обоих аргументов указать пустую строку "", то тег будет деактивирован. Например: IOSetItem("MyTag2", "", "");</p> <p>Если пустая строка задана только в качестве аргумента Item, то текущее название элемента данных сохраняется, а название канала доступа изменяется. Например, следующий оператор изменит канал доступа тега MyTag3 на "excel2", не меняя текущее название элемента данных: IOSetItem("MyTag3", "excel2", "");</p> <p>Аналогично, если пустая строка указана в качестве аргумента AccessName, то текущий канал доступа сохраняется, название элемента данных меняется. Например, следующий оператор изменит значение элемента данных тега MyTag4 на "R1C2", сохранив неизменным канал доступа: IOSetItem("MyTag4", "", "R1C2");</p>	
См. также		

IOSetRemoteReferences()

	Изменяет в реальном времени объектные ссылки. Может использоваться для изменения ссылок ArchestrA, удалённых ссылок InTouch и ссылок на тэги в/в.	
Категория	Data Change	
Синтаксис	IOSetRemoteReferences (BaseAccess, NewAccess, MatchString, SubstituteString, Mode);	
	Параметр	Описание
	BaseAccess	Этот строковый параметр задаёт первоначально сконфигурированное имя доступа для поиска в ссылке(ах).
	NewAccess	Новое имя доступа. Оно применимо ко всем ссылкам, для которых первоначально имя доступа удовлетворяет строке, предоставляемой в BaseAccess и для которой первоначально имя элемента удовлетворяет MatchString (если она задана).
	MatchString	Строка для поиска в первоначально сконфигурированном имени элемента (Item Name) в ссылке(ах). Если эта строка пустая, она трактуется как подходящая для любого имени элемента.
	SubstituteString	Строка для подстановки в первоначально сконфигурированное имя элемента, заменяющая MatchString, чтобы дать для ссылок активное имя элемента. Если эта строка пустая, никаких подстановок не делается.
	Mode	Определяет способ, которым MatchString должна удовлетворять первоначально сконфигурированному имени элемента. Проверка всегда начинается с начала имени элемента и никогда – с середины. 0 означает, что совпадение должно быть для всего имени элемента или до ".", 1 означает, что допускается частичное совпадение, даже если следующий символ не "."
Комментарии	Эта функция перед изменением объектных ссылок не проверяет правильность нового тэга или имени доступа.	
Пример(ы)	<pre>IOSetRemoteReferences ("Galaxy", "", "pumpX", "pump001", 0); IOSetRemoteReferences ("Galaxy", "TagServer1", "pumpX", "p2", 0); IOSetRemoteReferences ("TagServer1", "TagServer2", "pumpX", "backpump3", 0) IOSetRemoteReferences ("TagServer1", "", "Tank", "Plant", 1)</pre>	

IOStartUninitConversations()

	Во время запуска приложения WindowViewer оно автоматически выдаёт запрос <i>initiate</i> для установления всех каналов обмена данными. Если сервер ввода/вывода не отвечает на запрос <i>initiate</i> , то, вызвав эту функцию, можно заставить приложение WindowViewer повторно попытаться установить соединение с этим сервером.
Категория	прочие
Синтаксис	IOStartUninitConversations();
Примечание	Данная функция выполняет те же самые действия, что и команда Start Uninitiated Conversations (Установить неоткрытые каналы) меню Special (Специальные) приложения WindowViewer.

IsAnyAsynchFunctionBusy()

	Производит проверку на наличие выполняемых в текущий момент асинхронных Quick-функций. Её можно использовать в скрипте для приостановки исполнения до тех пор, пока все текущие асинхронные функции не завершатся. Таким образом можно обеспечить синхронизацию выполнения асинхронных функций.	
Категория	системная	
Синтаксис	=IsAnyAsynchFunctionBusy();	
	Аргумент	Описание
	DiscreteTag	<p>Название логического тега, в который возвращаются следующие значения.</p> <p>Если период ожидания истекает, а в системе всё ещё исполняются какие-либо асинхронные функции, возвращается 1 (TRUE).</p> <p>Если исполняющихся в текущий момент асинхронных функций нет, то немедленно возвращается 0 (FALSE), в противном случае функция переходит в режим ожидания. Значение 0 (FALSE) возвращается также, если по истечении периода ожидания в системе не будет никаких исполняющихся асинхронных функций.</p>
	timeout	<p>Аргумент целого типа, задающий интервал ожидания в секунда.</p> <p>Значение аргумента, равное 0, означает запрет ожидания.</p>
Примеры	<p>Предположим, что приложению необходимо установить соединения с разными SQL-базами данных с помощью асинхронных Quick-функций, причём известно, что эта операция занимает две минуты. Для этого сначала нужно запустить асинхронную функцию для соединения с SQL-базами. Затем следует запустить функцию IsAnyAsynchFunctionBusy(120), указав достаточный для ожидания период времени, в течение которого все соединения должны быть установлены до завершения этой функции.</p> <p>Если через 2 минуты соединения не будут установлены и асинхронные Quick-функции будут всё ещё пытаться их установить, функция IsAnyAsynchFunctionBusy() возвратит значение 1 (истинно). После этого на экран можно вывести сообщение об ошибке, информирующее оператора о неудачной попытке соединения с SQL-базой.</p> <p>Можно создать следующий оконный скрипт типа On Show (при показе):</p> <pre>IF IsAnyAsynchFunctionBusy(120) == 1 THEN SHOW "Ошибка соединения с SQL-базой данных "; ENDIF;</pre>	

IsAssignedRole()

	Определяет, назначена ли зарегистрированному в текущий момент пользователю указанная роль.	
Категория	контроль доступа	
Синтаксис	DiscreteTag=IsAssignedRole("RoleName");	
	Аргумент	Описание
	RoleName	Роль пользователя сервера AppServer.
Примечание	Функция действительна только для режима контроля доступа средствами сервера и применима к зарегистрированному в текущий момент пользователю. Если пользователь зарегистрирован и ему назначена указанная роль в системе Galaxy IDE, то функция возвращает значение TRUE, во всех остальных случаях возвращается значение FALSE.	
Примеры	DiscreteTag=IsAssignedRole("Administrator");	
См. также	AttemptInvisibleLogon(), PostLogonDialog(), InvisibleVerifyCredentials(), QueryGroupMembership(), AddPermission().	

Log()

	Возвращает значение натурального логарифма указанного числа.	
Категория	математическая	
Синтаксис	RealResult=Log(Number);	
	Аргумент	Описание
	Number	Любое число либо название тега целого или вещественного типа.
Примечание	Функция возвращает значение натурального логарифма аргумента Number . При указании 0 в аргумента результат будет не определён.	
Примеры	Log(100) возвращает 4.605... Log(1) возвращает 0	

LogMessage()

	Записывает определённое пользователем сообщение в журнал Wonderware Logger	
Категория	прочие	
Синтаксис	LogMessage("Message_Tag");	
	Аргумент	Описание
	Message_Tag	Сообщение, которое должно быть записано в журнал Wonderware Logger. Литерал или название тега типа "сообщение".
Примечание	Функция, которой можно пользоваться для отладки скриптов. Распределив вызовы функции по скрипту, можно определить порядок его исполнения, характеристики и значения тегов до и после исполнения скрипта. Каждое передаваемое в журнал Wonderware Logger сообщение сопровождается датой и временем его передачи.	
Примеры	<p>LogMessage("Report Script is Running"); При выполнении этого оператора в журнале Wonderware Logger появится следующая запись: 94/01/14 15:21:14 WWSCRIPT Message:Report Script is Running. Другой пример: LogMessage("Тег MyTag имеет значение " + Text(MyTag, "#")); MyTag+MyTag + 10; LogMessage("Тег MyTag имеет значение " + Text(MyTag, "#"));</p> <p>Дополнительные сведения о программе регистрации Wonderware Logger можно найти в системном <i>Руководстве администратора системы FactorySuite</i>.</p>	

LogN()

	Возвращает значение логарифма числа x по основанию n .	
Категория	математическая	
Синтаксис	Result=LogN(Number, Base);	
	Аргумент	Описание
	Number	Любое число либо название тега целого или вещественного типа.
	Base	Аргумент целого типа, определяющий основание логарифма. Целое число либо название тега целого типа.
Примечание	При указании основания, равного 1, результат будет неопределённым.	
Примеры	LogN(8, 3) возвращает 1.89279 LogN(NumberTag, BaseTag) возвратит 0.564, если тег NumberTag имеет значение 3, а тег BaseTag – значение 7.	

Logoff()

	Автоматически завершает сеанс работы пользователя в системе InTouch	
Категория	контроль доступа	
Синтаксис	DiscreteTag = LogOff();	
	Аргумент	Описание
	нет	
Примечание	Завершение сеанса работы пользователя в системе и установка текущего статуса пользователя в значение по умолчанию none (Отсутствует).	
Примеры	DiscreteTag = LogOff();	
См. также	PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), AttemptInvisibleLogon(), QueryGroupMembership(), AddPermission()	

LogonCurrentUser()

	Регистрирует в InTouch отключённого в текущий момент пользователя операционной системы и возвращает текущему пользователю AccessLevel.	
Категория	безопасность	
Синтаксис	IntegerResult = LogonCurrentUser();	
	Параметр	Описание
	ОТСУТСТВУЕТ	
Комментарии	<p>Используйте эту функцию только в режиме безопасности ОС.</p> <p>Если регистрация выполнена успешно, функция возвращает уровень доступа пользователя и устанавливает системные тэги \$Operator, \$OperatorName, \$OperatorDomain и \$AccessLevel. Если регистрация отвергнута, функция возвращает -1 и никаких изменений системных тэгов \$Operator, \$OperatorName, \$OperatorDomain и \$AccessLevel не производит.</p>	
Пример(ы)	IntegerResult = LogonCurrentUser();	
См. также	PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), AttemptInvisibleLogon(), QueryGroupMembership(), AddPermission().	

OpenWindowsList()

	Возвращает в диалоговом окне список открытых окон.
Категория	Представление
Синтаксис	[Результат=] OpenWindowsList();
Комментарии	В режиме реального времени выводится диалоговое окно "Open Windows List" (список открытых окон) и перечисляются все открытые окна InTouch.
Пример(ы)	OpenWindowsList ();

Pi()

	Возвращает значение числа пи
Категория	математическая
Синтаксис	RealResult=Pi();
Примеры	Pi() возвращает число 3.1415926

PlaySound()

	Выдает звуковой сигнал, определённый файлом .wav	
Категория	прочие	
Синтаксис	PlaySound("SoundName", Flags);	
	Аргумент	Описание
	SoundName	Литерал или тег типа "сообщение" с названием требуемого звукового файла.
	Flags	Аргумент, управляющий воспроизведением звука. Допустимые значения:
		0 Синхронное воспроизведение (значение по умолчанию).
		1 Асинхронное воспроизведение.
		2 Запрет использования звука по умолчанию. Сигнал воспроизводится из указанного файла .wav. Кроме того, функция может воспроизводить файл, определённый в секции [Sounds] Реестра Windows. Например, если в Реестре будет записана строка вида MouseClick=C:\Sounds\Click.wav то в качестве аргумента SoundName функции можно указать MouseClick .
		3–7 Не используется.
		8 Циклическое воспроизведение звука, пока не встретится очередной вызов функции. Только для Windows 2000.
		9 Вызов PlaySound() . Для Windows NT.
		10 Продолжение воспроизведения любого воспроизводимого в текущий момент звукового сигнала.
Примеры	PlaySound ("c:\horns.wav",1);	
	Аудиоданные должны находиться в имеющейся физической памяти, а их формат должен соответствовать установленному драйверу звуковой платы. Поиск аудиофайлов осуществляется в следующем порядке: текущий каталог, каталог Windows, каталог Windows\System, каталоги, определённые в переменной окружения PATH. Если в указанных каталогах такого файла нет, необходимо в Панели управления системе указать его размещение (дважды щёлкните кнопкой мыши на пункте Sounds (Звук) и укажите расположение аудиофайла). При отсутствии аудиофайла по умолчанию никакой звук воспроизводиться не будет.	

PostLogonDialog()

	Открывает окно регистрации InTouch и возвращает значение TRUE
Категория	контроль доступа
Синтаксис	DiscreteTag=PostLogonDialog();
Примечание	Открывает окно регистрации InTouch и возвращает значение TRUE.
Примеры	DiscreteTag=PostLogonDialog();
См. также	InvisibleVerifyCredentials(), AttemptInvisibleLogon(), IsAssignedRole(), QueryGroupMembership(), AddPermission().

PrintHT()

	Функция печати графика указанного архивного тренда. Печать возможна только тогда, когда архивный тренд видим на экране.	
Категория	архивирование	
Синтаксис	PrintHT("Trend_Tag");	
	Аргумент	Описание
	Trend_Tag	Название архивного тренда. Литерал либо название тега типа "сообщение".
Примеры	PrintHT("HistTrend1");	

PrintScreen()

	Вывод указанного экрана на печать	
Категория	прочие	
Синтаксис	PrintScreen(ScreenOption, PrintOption);	
	Аргумент	Описание
	ScreenOption	1 = клиентская область приложения View (без меню), 2 = окно целиком (всё в окне WindowViewer, включая меню), любое другое значение (включая 0) интерпретируется как 1.

	PrintOption	<p>1 = наилучшее соответствие (либо по горизонтали, либо по вертикали с сохранением отношения сторон изображения), 2 = растянуть по вертикали (печать по всей высоте страницы с сохранением пропорций), 3 = растянуть по горизонтали (печать по всей ширине страницы с сохранением пропорций), 4 = растянуть на страницу (печать по всей площади страницы с нарушением пропорций), любое другое значение (включая 0) интерпретируется как 1.</p>
Примечание	<p>Всплывающие окна, выходящие за границы области View, обрезаются в любом режиме печати. Кроме того, если окно другого приложения (например окно отладчика в режиме отображения "всегда сверху") перекрывает окно View, то оно тоже будет распечатано.</p> <p>Данная функция аналогична функции PrintWindow(), отличаясь количеством распечатываемых окон. Длительность периода ожидания задаётся параметром PrintScreenWait в файле INTOUCH.INI, например: PrintScreenWait=10000 где 10000 – число миллисекунд.</p> <p>Шрифты распечатываются обычным образом, графические объекты перед этим преобразуются в растровые изображения. Окна с белым фоном, содержащие только текст, распечатываются быстрее всего. Больше всего времени занимает печать окон с цветным фоном, содержащие много графических объектов.</p> <p>Чтобы напечатанный текст выглядел так же, как и на экране, во всех окнах (которые будут распечатываться) рекомендуется использовать шрифты типа True Type.</p> <p>При печати кнопок надписи на них могут обрезаться, поскольку текст на них выводится системным шрифтом System, который не является шрифтом типа True Type. Кроме того, в напечатанном виде он несколько отличается от изображения на экране. В таких случаях рекомендуется увеличивать размеры экранных кнопок.</p> <p>Если принтер используется для печати алармов, то вывод экранов на печать с помощью функции PrintScreen() рекомендуется осуществлять на другое устройство.</p> <p>Экран будет распечатан с искажениями, если какое-либо из окон перекрывает такие объекты, как окно вывода сведений о распределённых алармах, развёртывающиеся списки, текстовые поля, переключатели, окошки выбора и т.д.</p>	

PrintWindow()

	Печатает указанное окно	
Категория	прочие	
Синтаксис	PrintWindow("Window", Left, Top, Width, Height, Options);	
	Аргумент	Описание
	Window	Название распечатываемого окна. Литерал либо название тега типа "сообщение".
	Left	Левый отступ в дюймах. Вещественное число либо название тега вещественного типа.
	Top	Верхний отступ в дюймах. Вещественное число либо название тега вещественного типа.
	Width	Ширина печатной копии в дюймах. "0" означает печать максимально возможного изображения с сохранением пропорций. Вещественное число либо название тега вещественного типа.
	Height	Высота печатной копии в дюймах. "0" означает печать максимально возможного изображения с сохранением пропорций. Вещественное число либо название тега вещественного типа.
	Options	Аргумент логического типа, используемый, когда нулю равен либо аргумент Width, либо аргумент Height. Единица (1) означает печать максимально возможного изображения с размерами, кратными размеру исходного окна. Ноль (0) – печать изображения на всю страницу (с сохранением пропорций). Логическое значение либо тег логического типа. Если в окне есть какие-либо растровые изображения, рекомендуется устанавливать Options в 1, чтобы при печати они не были искажены.
Примечание	<p>С помощью этой функции можно распечатывать множество окон. Ею рекомендуется пользоваться вместо функции PrintHT(), особенно когда необходима печать всего окна, а не только графика архивного тренда.</p> <p>При выполнении этой функции View "выгружает" окно в отдельную область памяти, затем в течение 10 секунд ожидает обновления всех DDE-тегов. Через 10 секунд окно посылается на печать, при этом оно не обязательно должно быть открыто или видимо на экране.</p> <p>Длительность периода ожидания задаётся параметром PrintWindowWait в файле INTOUCH.INI, например: PrintWindowWait=10000 где 10000 – число миллисекунд.</p> <p>Шрифты распечатываются обычным образом, графические объекты перед этим преобразуются в растровые изображения. Окна с белым</p>	

	<p>фоном, содержащие только текст, распечатываются быстрее всего. Больше всего времени занимает печать окон с цветным фоном, содержащие много графических объектов.</p> <p>Чтобы напечатанный текст выглядел так же, как и на экране, во всех окнах (которые будут распечатываться) рекомендуется использовать шрифты типа True Type.</p> <p>При печати кнопок надписи на них могут обрезаться, поскольку текст на них выводится системным шрифтом System, который не является шрифтом типа True Type. Кроме того, в напечатанном виде он несколько отличается от изображения на экране. В таких случаях рекомендуется увеличивать размеры экранных кнопок.</p> <p>Если принтер используется для печати алармов, то вывод окон на печать с помощью функции PrintWindow() рекомендуется осуществлять на другое устройство.</p> <p>Окно будет распечатано с искажениями, если какое-либо из других окон перекрывает такие объекты, как окно вывода сведений о распределённых алармах, развёртывающиеся списки, текстовые поля, переключатели, окошки выбора и т.д.</p> <p>Если в текущий момент времени окно не является активным, то простые и развёртывающиеся списки и прочие ActiveX-объекты и объекты окна (типа окна браузера тегов) в распечатке будут отсутствовать. Нормальная печатная копия получается, когда окно активно.</p> <p>Другими словами, кнопка печати окна должна находиться в этом же окне, иначе списки и ActiveX-объекты в распечатке будут отсутствовать.</p> <p>Чтобы эта проблема не возникала, перед вызовом функции PrintWindow() можно запустить на выполнение такие функции, как Show(), ShowAt() либо almMoveWindow() (для того чтобы требуемое окно стало активным).</p>
<p>Примеры</p>	<p>Следующий условный скрипт типа On True (Если истина) обеспечивает ежедневную печать трёх страниц сводки в 8:30 утра.</p> <p>Условие: \$Hour == 8 AND \$Minute == 30</p> <p>Скрипт: PrintWindow("Сводка по 1-й смене", 1, 1, 0, 0, 0); PrintWindow("Сводка по 2-й смене", 1, 1, 0, 0, 0); PrintWindow("Сводка по 3-й смене", 1, 1, 0, 0, 0);</p> <p>Функция PrintWindow() возвращает 1, если окно с указанным названием существует и может быть поставлено в очередь на печать. В противном случае она возвращает 0. Таким образом можно контролировать успех выполнения функции.</p> <p>Status=PrintWindow("Сводка по смене", 1, 1, 0, 0, 0);</p> <p>где Status – это название логического тега, принимающего значения 0 или 1.</p>

ptGetTrendType()

Эта функция покажет значение, соответствующее текущему

	режиму указанного 16-перьевого трэнда.	
Категория	перьевого трэнд	
Синтаксис	ptGetTrendType (<i>TrendObjName</i>) ;	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
Комментарии	Возвращает тип трэнда: 0 для исторического, (1 для не прокручиваемого в реальном времени – в текущей версии не используется) и 2 – для реального времени.	
Пример(ы)	ptGetTrendType (GroupTrendObjName); ptGetTrendType ("PenTrend_1");	
См. также		

ptLoadTrendCfg()

	Эта функция загружает конфигурационную информацию трэнда из конфигурационного файла.	
Категория	перьевого трэнд	
Синтаксис	ptLoadTrendCfg(<i>TrendObjName</i> , <i>Filename</i>);	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>Filename</i>	Имя конфигурационного файла (может включать путь).
См. также		

ptPanCurrentPen()

	Эта функция будет прокручивать указанные оси перьев 16-перьевых трэндов, основываясь на предоставленных значениях. Включает имя трэнда и значение для развёртывания либо в промежуточных, либо в основных делениях.	
Категория	перьевой трэнд	
Синтаксис	ptPanCurrentPen (<i>TrendObjName</i> , <i>majorUnits</i> , <i>minorUnits</i>);	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>majorUnits</i>	Множитель для прокрутки в единицах, определённых отметками основных делений.
	<i>minorUnits</i>	Множитель для дополнительной прокрутки в единицах, определённых отметками промежуточных делений.
Комментарии	Параметры Основных (Major) делений и Промежуточных (Minor) делений, задаваемые при разработке в диалоговом окне конфигурирования PenTrend Control (управление перьевыми трэндами) , являются базисом, на котором вычисляется величина прокрутки. Диапазон трэнда от 0 до 200 по оси значений, значение основного деления 10 и значение промежуточного деления 2 – всё это даст трэнд с основными отметками делений через каждые 20 единиц и промежуточными отметками делений через каждые 10 единиц. Функция ptPanCurrentPen("TrendName",0.4,0.5) прокрутит экран текущего пера на $0.4*20 + 0.5*10 = 13$ единиц.	
Пример(ы)	<p>ptPanCurrentPen(GroupTrendObjName, 1, 0); (прокрутит на одно основное деление вниз)</p> <p>ptPanCurrentPen(GroupTrendObjName, 0, 0.5); (прокрутит на половину промежуточного деления вниз)</p> <p>ptPanCurrentPen(GroupTrendObjName, -2, -0.5); (прокрутит на 2 основных деления вверх и на половину промежуточного деления вверх)</p> <p>ptPanCurrentPen(GroupTrendObjName, 1, -2); (прокрутит на 1 основное деление вверх и на два промежуточных деления вниз)</p>	

ptPanTime()

	Эта функция прокрутит ось времени указанного 16-перьевого трэнда, основываясь на предоставленных значениях. Включает имя трэнда и значение для развёртывания либо в промежуточных, либо в основных делениях.	
Категория	перьевого трэнд	
Синтаксис	ptPanTime (<i>TrendObjName</i> , <i>MajorFactor</i> , <i>MinorFactor</i>);	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>MajorFactor</i>	Множитель для прокрутки в основных делениях.
	<i>MinorFactor</i>	Множитель для прокрутки в промежуточных делениях.
Комментарии	Параметры Основных (Major) делений и Промежуточных (Minor) делений, задаваемые при разработке в диалоговом окне конфигурирования PenTrend Control (управление перьевыми трендами) , являются базисом, на котором вычисляется величина прокрутки. Трэнда с временным промежутком 120 секунд, значение основного деления 10 и значение промежуточного деления 2 – всё это даст трэнд с основными отметками делений через каждые 12 секунд и промежуточными отметками делений через каждые 6 секунд. Функция ptPanTime("TrendName",1,0.5) прокрутит ось времени на $0.4*20 + 0.5*10 = 13$ секунд.	
Пример(ы)	ptPanTime(GroupTrendObjName, 1, 0); (прокрутит на одно основное деление вправо) ptPanTime(GroupTrendObjName, 0, 0.5); (прокрутит на половину промежуточного деления вправо) ptPanTime(GroupTrendObjName, -2, -0.5); (прокрутит на 2 основных деления плюс на половину промежуточного деления влево) ptPanTime(GroupTrendObjName, 1, -2); (прокрутит на 1 основное деление вправо и на два промежуточных деления влево)	

ptPauseTrend()

	Эта функция на время приостановит обновление окна вывода 16-перьевого трэнда. Обновление приостановится, если значение будет true (1).	
Категория	перьевого трэнд	
Синтаксис	ptPauseTrend (<i>TrendObjName</i> , <i>Value</i> ,);	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>Value</i>	0 возобновляет обновление и 1 – останавливает.
Пример(ы)	ptPauseTrend (GroupTrendObjName , 1);	

ptSaveTrendCfg()

	Эта функция сохраняет конфигурационную информацию текущего трэнда в файле конфигурации.	
Категория	перьевой трэнд	
Синтаксис	<code>ptSaveTrendCfg(<i>TrendObjName</i>, <i>Filename</i>);</code>	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>Filename</i>	Имя файла конфигурации (может включать путь).
См. также		

ptSetCurrentPen()

	Вызывайте эту функцию, чтобы выбрать перо в качестве текущего пера. Текущее перо является пером, которогe управляет перьевой осью.	
Категория	перьевой трэнд	
Синтаксис	<code>ptSetCurrentPen(<i>TrendObjName</i>, <i>PenNum</i>);</code>	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>PenNum</i>	Задаёт номер пера, которое должно использоваться в качестве текущего пера.
Пример(ы)	<code>ptSetCurrentPen(GroupTrendObjName, 2);</code> Установит текущее перо в качестве пера 2.	

ptSetPen()

	Эта функция приписывает укзанному перу тэг.	
Категория	перьевой трэнд	
Синтаксис	<code>ptSetPen(<i>TrendObjName</i>, <i>penNum</i>, <i>Тэг</i>);</code>	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>PenNum</i>	Задаёт номер пера, которое должно сопоставляться новому тэгу.
	<i>Тэг</i>	Имя тэга.

ptSetPenEx()

	Эта функция приписывает тэг указанному перу и переписывает конфигурацию словаря для тэга.	
Категория	перьевой трэнд	
Синтаксис	<code>ptSetPenEx(TrendObjName, penNum, Тэг, minEU, maxEU, minPercent, maxPercent, Decimal, EU);</code>	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>PenNum</i>	Задаёт номер пера, которое должно сопоставляться новому тэгу.
	<i>Тэг</i>	Имя тэга.
	<i>minEU</i>	Минимальное значение инженерных единиц измерения для указанного тэга.
	<i>maxEU</i>	Максимальное значение инженерных единиц измерения для указанного тэга.
	<i>minPercen</i>	Минимальное процентное значение, на которое тэг может отклоняться от планового значения отклонения для него.
	<i>maxPercen</i>	Максимальное процентное значение, на которое тэг может отклоняться от планового значения отклонения для него.
	<i>Decimal</i>	Десятичная точность.
	<i>EU</i>	Обозначение инженерных единиц тэга.
См. также		

ptSetTimeAxis()

	Эта функция устанавливает дату и время начала и конца оси времени графика.	
Категория	перьевой трэнд	
Синтаксис	<code>ptSetTimeAxis(TrendObjName, StartDateTime, EndDateTime);</code>	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>StartDateTime</i>	Дата и время начала графика. Допустимый формат: mm/dd/yyyy hh:mm:ss AM/PM.
	<i>EndDateTime</i>	Дата и время конца графика. Допустимый формат: mm/dd/yyyy hh:mm:ss AM/PM.
Пример(ы)	<code>ptSetTimeAxis("PenTrend_1", "08/13/2005 12:54:13 AM", "08/15/2005 03:47:54 PM");</code>	

ptSetTimeAxisToCurrent()

	Эта функция используется для обновления тренда до текущего момента.	
Категория	перьевой тренд	
Синтаксис	<code>PtSetTimeAxisToCurrent (TrendObjName) ;</code>	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта тренда. Должно быть либо строковой константой, либо тэгом сообщения.
Пример(ы)	<code>ptSetTimeAxisToCurrent ("PenTrend_1") ;</code>	
См. также		

ptSetTrend()

	Эта функция будет останавливать и запускать обновление указанного окна вывода 16-перьевого тренда.	
Категория	перьевой тренд	
Синтаксис	<code>ptSetTrend(TrendObjName, EnableUpdates) ;</code>	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта тренда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>EnableUpdates</i>	0 для остановки обновления или 1 для запуска обновления.
Пример(ы)	<code>ptSetTrend("PenTrend_1", 1) ;</code> Запустит обновление объекта тренда с именем PenTrend_1 . <code>ptSetTrend(GroupTrendObjName , 0) ;</code> Запустит обновление объекта тренда, указанного тэгом InTouch GroupTrendObjName .	

ptSetTrendType()

	Эта функция устанавливает тип тренда, который будет показываться.	
Категория	перьевой тренд	
Синтаксис	<code>ptSetTrendType (TrendObjName, TrendType) ;</code>	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта тренда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>TrendType</i>	Аналоговое значение для установки типа тренда: 0 для исторического тренда; 1 или 2 для тренда времени исполнения.
Пример(ы)	<code>ptSetTrendType ("PenTrend_1", 2) ;</code>	

ptZoomCurrentPen()

	Эта функция уменьшит масштаб перьевой оси указанного 16-перьевого трэнда в указанное число раз.	
Категория	перьевой трэнд	
Синтаксис	<code>ptZoomCurrentPen (TrendObjName, ZoomFactor);</code>	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>ZoomFactor</i>	Используйте для уменьшения масштаба число больше 1.0 и для укрупнения масштаба число между 0 и 1.0. Коэффициент пропорциональности относится к диапазону числовой оси текущего объекта трэнда.
Пример(ы)	<p><code>ptZoomCurrentPen ("PenTrend_1", 1.02);</code> Это приведёт к уменьшению на 2% масштаба оси значений объекта трэнда PenTrend_1.</p> <p><code>ptZoomCurrentPen (TrendObj, 0.7);</code> Это приведёт к укрупнению на 30% оси значений объекта трэнда, указанного тэгом сообщения <i>Trendobj</i>. Если перьевой трэнд имел до увеличения шкалу, простирающуюся от 0 до 100, после выполнения данной функции он будет показывать шкалу, простирающуюся от 15 до 85.</p>	

ptZoomTime()

	Эта функция уменьшит масштаб оси времени указанного 16-перьевого трэнда в указанное число раз.	
Категория	перьевой трэнд	
Синтаксис	<code>ptZoomTime (TrendObjName, ZoomFactor);</code>	
	Параметр	Описание
	<i>TrendObjName</i>	Имя объекта трэнда. Должно быть либо строковой константой, либо тэгом сообщения.
	<i>ZoomFactor</i>	Используйте для уменьшения масштаба число больше 1.0 и для укрупнения масштаба число между 0 и 1.0. Коэффициент пропорциональности относится к диапазону оси времени текущего объекта трэнда.
Пример(ы)	<p><code>ptZoomTime ("PenTrend_1", 1.17);</code> Это приведёт к уменьшению на 17% масштаба оси времени объекта трэнда PenTrend_1.</p> <p><code>ptZoomCurrentPen (TrendObj, 0.5);</code> Это приведёт к укрупнению на 50% оси времени объекта трэнда, указанного тэгом сообщения <i>Trendobj</i>. Если перьевой трэнд имел до увеличения шкалу времени, покрывающую 1 минуту, после выполнения данной функции он показывать шкалу времени, покрывающую 30 секунд.</p>	

QueryGroupMembership()

Категория	контроль доступа						
Синтаксис	DiscreteTag=QueryGroupMembership("Domain", "");						
	<table border="1"><thead><tr><th>Аргумент</th><th>Описание</th></tr></thead><tbody><tr><td>Domain</td><td>Название домена либо локальной машины, в которой определена группа.</td></tr><tr><td>Group</td><td>Название группы.</td></tr></tbody></table>	Аргумент	Описание	Domain	Название домена либо локальной машины, в которой определена группа.	Group	Название группы.
Аргумент	Описание						
Domain	Название домена либо локальной машины, в которой определена группа.						
Group	Название группы.						
Примечание	Функция действительна только для режима контроля доступа средствами операционной системы и применима к зарегистрированному в текущий момент пользователю. Если пользователь зарегистрирован и входит в указанную группу указанного домена, то возвращается значение TRUE, в противном случае возвращается FALSE.						
Примеры	DiscreteTag=QueryGroupMembership("corporate_hq", "InTouchAdmins"); DiscreteTag=QueryGroupMembership("JohnS01", "InTouchUsers");						
См. также	BOOL PostLogonDialog(), InvisibleVerifyCredentials(), BOOL IsAssignedRole(), AttemptInvisibleLogon(), AddPermission()						

RecipeDelete()

	Удаляет название рецепта из указанного файла шаблонов рецептов	
Категория	рецепты	
Синтаксис	RecipeDelete("Filename","RecipeName");	
	Аргумент	Описание
	Filename	Название файла шаблонов рецептов. Литерал либо название тега типа "сообщение".
	RecipeName	Название рецепта, удаляемого из указанного файла шаблонов рецептов. Функции RecipeLoad() , RecipeSave() и RecipeDelete() требуют обязательного указания названия рецепта. Определить значение этого аргумента можно с помощью функции RecipeSelectRecipe() .
Примеры	Следующий оператор удаляет рецепт Recipe1 из файла recfile.csv :	
	RecipeDelete("c:\recipe\recfile.csv", "Recipe1");	

RecipeGetMessage()

	Записывает код ошибки в аналоговый тег, а соответствующее сообщение об ошибке – тег типа "сообщение"	
Категория	рецепты	
Синтаксис	RecipeGetMessage(Analog_Tag, Message_Tag, Number);	
	Аргумент	Описание
	Analog_Tag	Название тега целого или вещественного типа без кавычек либо константа.
	Message_Tag	Название тега типа "сообщение" без кавычек либо литерал.
	Number	Максимальная длина строки, записываемой в тег Message_Tag . Теги InTouch типа "сообщение" могут хранить строки длиной не более 131 символа. Следует указывать значение 131, если только в определении тега в Словаре тегов для него не была указана другая максимальная длина допустимых значений. Данный аргумент может быть как целым числом, так и названием тега целого типа.
Примечание		
Примеры	<p>С помощью функции RecipeGetMessage() в скрипте по изменении данных код ошибки можно записать в аналоговый тег, а соответствующее сообщение об ошибке – в тег типа "сообщение":</p> <p>Название тега в скрипте: ErrorCode</p> <p>Скрипт:</p> <p>RecipeGetMessage(ErrorCode, ErrorMessage, 131);</p> <p>Этот скрипт будет автоматически выполняться всегда, как только значение аналогового тега ErrorCode изменится. В этот момент функция RecipeGetMessage() считает текущее числовое значение тега ErrorCode и возвратит соответствующее текстовое сообщение в тег ErrorMessage.</p> <p>ErrorCode = RecipeLoad("c:\App\recipe.csv", "Unit1", "cookies");</p> <p>RecipeGetMessage(ErrorCode, ErrorMessageTag, 131);</p>	

RecipeLoad()

	Загружает указанный рецепт в указанный блок тегов	
Категория	рецепты	
Синтаксис	RecipeLoad("Filename", "UnitName", "RecipeName");	
	Аргумент	Описание
	Filename	Название файла шаблонов рецептов. Допустимые значения – литерал либо название тега типа "сообщение".
	UnitName	Название блока тегов в указанном файле шаблонов рецептов. Данный аргумент в функции RecipeLoad() является обязательным. Функция RecipeSelectRecipe() возвращает значение в этот параметр. Допустимые значения – литерал либо название тега типа "сообщение".
	RecipeName	Название рецепта в указанном файле шаблонов рецептов. Функции RecipeLoad() , RecipeSave() и RecipeDelete() требуют обязательного указания названия рецепта. Функция RecipeSelectRecipe() возвращает значение в этот аргумент. Допустимые значения – литерал либо название тега типа "сообщение".
Примеры	Следующий оператор загружает значения, определённые для рецепта с названием Recipe1 (в файле recfile.csv), в теги, определённые для указанного блока Unit1 : RecipeLoad("c:\recipe\recfile.csv", "Unit1", "Recipe1");	

RecipeSave()

	Сохраняет вновь созданный рецепт или сохраняет сделанные в существующем рецепте изменения в указанном файле шаблонов рецептов	
Категория	рецепты	
Синтаксис	RecipeSave("Filename", "UnitName", "RecipeName");	
	Аргумент	Описание
	Filename	Название файла шаблонов рецептов. Допустимые значения – литерал либо название тега типа "сообщение".
	UnitName	Название блока тегов в указанном файле шаблонов рецептов. Данный аргумент в функции RecipeSave() является обязательным. Функция RecipeSelectUnit() возвращает значение в этот параметр. Допустимые значения – литерал либо название тега типа "сообщение" без кавычек.
	RecipeName	Название рецепта в указанном файле шаблонов рецептов. Функции RecipeLoad() , RecipeSave() и RecipeDelete() требуют обязательного указания названия рецепта. Функция RecipeSelectRecipe() возвращает значение в этот аргумент. Допустимые значения – литерал либо название тега типа "сообщение" без кавычек.
Примеры	Следующий оператор сохраняет все сделанные в рецепте с названием Recipe3 изменения в файле recfile.csv . Если рецепт Recipe3 не существует, он будет создан. Параметры рецепта будут записаны в теги, определённые в блоке Unit2 : RecipeSave("c:\recipe\recfile.csv", "Unit2", "Recipe3");	

RecipeSelectNextRecipe()

	Выбирает следующий рецепт из файла шаблонов рецептов	
Категория	рецепты	
Синтаксис	RecipeSelectNextRecipe("Filename", RecipeName, Number);	
	Аргумент	Описание
	Filename	Название файла шаблонов рецептов. Допустимые значения – литерал либо название тега типа "сообщение".
	RecipeName	Название рецепта в указанном файле шаблонов рецептов. Функции RecipeLoad() , RecipeSave() и RecipeDelete() требуют обязательного указания названия рецепта. Функция RecipeSelectRecipe() возвращает значение в этот аргумент. Допустимые значения – литерал либо название тега типа "сообщение" без кавычек.
	Number	Если функция должна заполнять параметр символами – максимальная длина строки. Строки InTouch (теги типа "сообщение") могут хранить не более 131 символа. Следует указывать значение 131, если только в определении тега в Словаре тегов для него не была указана другая максимальная длина допустимых значений.
Примеры	Следующий оператор считывает текущее значение тега RecipeName и возвращает следующий рецепт в файле. Если значение RecipeName представляет собой пустую строку или не может быть определено, возвращается первый рецепт в файле. Если RecipeName хранит название последнего рецепта в файле, то оно остаётся неизменным. (Рецепты хранятся в файле в том порядке, каком создавались.) RecipeSelectNextRecipe("c:\recipe\recfile.csv", RecipeName, 131);	

RecipeSelectPreviousRecipe()

	Выбирает предыдущий рецепт из файла шаблонов рецептов	
Категория	рецепты	
Синтаксис	RecipeSelectPreviousRecipe("Filename", RecipeName, Number);	
	Аргумент	Описание
	Filename	Название файла шаблонов рецептов. Допустимые значения – литерал либо название тега типа "сообщение".
	RecipeName	Название рецепта в указанном файле шаблонов рецептов. Функции RecipeLoad() , RecipeSave() и RecipeDelete() требуют обязательного указания названия рецепта. Функция RecipeSelectRecipe() возвращает значение в этот аргумент. Допустимые значения – литерал либо название тега типа "сообщение" без кавычек.
	Number	Если функция должна заполнять параметр символами – максимальная длина строки. Строки InTouch (теги типа "сообщение") могут хранить не более 131 символа. Следует указывать значение 131, если только в определении тега в Словаре тегов для него не была указана другая максимальная длина допустимых значений.
Примеры	<p>Следующий оператор считывает текущее значение тега RecipeName и возвращает предыдущий рецепт в файле. Возвращаемая строка будет записана в тег RecipeName и таким образом замещает его текущее значение. Если значение RecipeName представляет собой пустую строку или не может быть определено, возвращается последний рецепт в файле. Если RecipeName хранит название первого рецепта в файле, то оно остаётся неизменным. (Рецепты хранятся в файле в том порядке, каком создавались.)</p> <pre>RecipeSelectPreviousRecipe("c:\recipe\recfile.csv", RecipeName, 131);</pre>	

RecipeSelectRecipe()

	Выбирает рецепт с указанным названием из файла шаблонов рецептов	
Категория	рецепты	
Синтаксис	RecipeSelectRecipe("Filename", RecipeName, Number);	
	Аргумент	Описание
	Filename	Название файла шаблонов рецептов. Допустимые значения – литерал либо название тега типа "сообщение".
	RecipeName	Название рецепта в указанном файле шаблонов рецептов. Функции RecipeLoad() , RecipeSave() и RecipeDelete() требуют обязательного указания названия рецепта. Функция RecipeSelectRecipe() возвращает значение в этот аргумент. Допустимые значения – литерал либо название тега типа "сообщение" без кавычек.
	Number	Если функция должна заполнять параметр символами – максимальная длина строки. Строки InTouch (теги типа "сообщение") могут хранить не более 131 символа. Следует указывать значение 131, если только в определении тега в Словаре тегов для него не была указана другая максимальная длина допустимых значений. Допускается указание целого числа или названия тега целого типа.
Примеры	При выполнении следующего оператора на экране открывается окно Select a Recipe (Выбор рецепта) : RecipeSelectRecipe("c:\recipe\recfile.csv", RecipeName, 131); После того как рецепт будет выбран, его название будет записано в тег RecipeName.	

RecipeSelectUnit()

	Выбирает блок тегов, в которые будут загружены текущие параметры рецепта	
Категория	рецепты	
Синтаксис	RecipeSelectUnit("Filename", UnitName, Number);	
	Аргумент	Описание
	Filename	Название файла шаблонов рецептов. Допустимые значения – литерал либо название тега типа "сообщение".
	UnitName	Название блока тегов в указанном файле шаблонов рецептов. Данный аргумент в функции RecipeLoad() является обязательным. Функция RecipeSelectUnit() возвращает значение в этот параметр. Допустимые значения – литерал либо название тега типа "сообщение" без кавычек.
	Number	Если функция должна заполнять параметр символами – максимальная длина строки. Строки InTouch (теги типа "сообщение") могут хранить не более 131 символа. Следует указывать значение 131, если только в определении тега в Словаре тегов для него не была указана другая максимальная длина допустимых значений. Допускается указание целого числа или названия тега целого типа.
Примеры	<p>При выполнении следующего на экране открывается окно Select a Unit (Выбор блока тегов):</p> <pre>RecipeSelectRecipe("c:\recipe\recfile.csv", UnitName, 131);</pre> <p>После того как блок будет выбран, его название будет записано в тег UnitName.</p> <p>И функция RecipeSelectRecipe(), и функция RecipeSelectUnit() используются в сочетании с функцией RecipeLoad().</p> <p>Дополнительные сведения о совместном использовании функций можно найти в параграфе "Комбинирование функций рецептов" системного <i>Руководства пользователя Менеджера рецептов Recipe Manager</i>.</p>	

ReloadWindowViewer()

	Запускает WindowViewer повторно
Категория	системная
Синтаксис	ReloadWindowViewer()
Примечание	Данная функция осуществляет автоматический перезапуск приложения WindowViewer. Она может использоваться вместо функции NAD (Network Application Development – Система разработки сетевых приложений) автоматического обновления приложения. Тег \$ApplicationChanged позволяет определять моменты изменения главного приложения, а функция ReloadWindowViewer() – обновлять узел View без прерывания операций этого узла. Данную функцию можно использовать в кнопочном скрипте, для того чтобы оператор мог выполнить обновление приложения (получив соответствующее уведомление) в удобное для него время позднее. Фактически эта функция заменяет собой функцию RestartWindowViewer() .
См. также	\$ApplicationChanged

RestartWindowViewer()

	Прекращает выполнение и запускает приложение WindowViewer повторно
Категория	системная
Синтаксис	RestartWindowViewer()
Примечание	Данная функция осуществляет автоматическое завершение и повторный перезапуск приложения WindowViewer. Она может использоваться вместо функции NAD (Network Application Development – Система разработки сетевых приложений) автоматического обновления приложения. Тег \$ApplicationChanged позволяет определять моменты изменения главного приложения, а функция RestartWindowViewer() – обновлять узел View путём завершения приложения и повторного его запуска. Данную функцию можно в кнопочном скрипте, для того чтобы оператор мог выполнить останов и перезапуск приложения (получив соответствующее уведомление) в удобное для него время позднее. В настоящее время вместо неё всё чаще используется функция ReloadWindowViewer() , которая позволяет обновлять узел View без прекращения его операций.
См. также	\$ApplicationChanged, ReloadWindowViewer()

Round()

	Округляет указанное действительное число с указанной точностью
--	--

Категория	математическая	
Синтаксис	RealResult=Round(Number, Precision);	
	Аргумент	Описание
	Number	Любое число либо название тега целого или вещественного типа.
	Precision	Точность округления. Может представлять собой число либо тег целого или вещественного типа.
Примечание	Аргумент Precision задаёт точность округления аргумента Number .	
Примеры	Round(4.3, 1) возвращает 4 Round(4.3, .01) возвращает 4.30 Round(4.5, 1) возвращает 5 Round(-4.5, 1) возвращает -4 Round(106, 5) возвращает 105 Round(43.7, .5) возвращает 43.5	
См. также	Trunc()	

SendKeys

	Пересылает указанные коды клавиш другому приложению, которое будет воспринимать их так же, как и обычное нажатие на клавиши. Эту возможность можно применять для ввода в приложение данных или команд. В операторе SendKeys могут указываться большинство клавиш. Каждая из них представляется одним или более символом, например A для буквы А или {ENTER} для клавиши Enter			
Категория	прочие			
Синтаксис	SendKeys KeySequence;			
	Аргумент	Описание		
	KeySequence	Последовательность клавиш или название тега типа "сообщение".		
Примечание	Чтобы указать более одной клавиши, перечислите подряд коды для каждого символа. Например, для передачи кода знака доллара и буквы "b" введите последовательность \$b. Далее перечислены обозначения многих функциональных клавиш клавиатуры.			
	Клавиша	Обозначение	Клавиша	Обозначение
	BACKSPACE	{BACKSPACE}	HOME	{HOME}
	BREAK	{BREAK}	INSERT	{INSERT}
	CAPSLOCK	{CAPSLOCK}	LEFT	{LEFT}
	DELETE	{DELETE} или {DEL}	NUMLOCK	{NUMLOCK}
	DOWN	{DOWN}	PAGE DOWN	{PGDN}
	END	{END}	PAGE UP	{PGUP}
	ENTER	{ENTER} или "~" (тильда)	PRTSC	{PRTSC}
	ESCAPE	{ESCAPE} или {ESC}	RIGHT	{RIGHT}
	F1	{F1}*	TAB	{TAB}
	UP	{UP}		
	* Все функциональные клавиши вводятся аналогичным образом.			
	Для специальных клавиш (SHIFT, CTRL и ALT) существуют собственные обозначения:			
	SHIFT	+ (плюс)		
	CTRL	^ (каретка)		
	ALT	% (процент)		
Примеры	Если нужно использовать вместе две или более специальных клавиш, требуется вторая пара скобок. Следующий оператор			

	<p>используется для имитации удержания клавиши CTRL во время нажатия клавиши ALT, а затем p:</p> <p>SendKeys "^(%p)";</p> <p>Вызову функции SendKeys может предшествовать команда ActiveApp для того, чтобы направить коды клавиш нужному приложению.</p> <p>Следующий оператор активизирует приложение Excel и передаёт ему комбинацию клавиш CTRL+P (которая запускает макрос печати):</p> <p>ActivateApp "Microsoft Excel";</p> <p>SendKeys "^(%p)";</p> <p>Следующий оператор откроет на экране окно регистрации в приложении WindowViewer:</p> <p>SendKeys "%(SYL)";</p> <p>Для кнопки вызова справки можно определить следующий кнопочный скрипт:</p> <p>SendKeys "{F1}";</p> <p>В некоторых операционных системах Windows эта функция может не работать в связи с особенностями реализации уровня аппаратных абстракций HAL (Hardware Abstract Layer).</p>
--	--

SetPropertyD()

	Во время исполнения программы устанавливает указанное логическое значение указанного свойства	
Категория	GOT	
Синтаксис	[ErrorNumber=] SetPropertyD("ControlName.Property", DiscreteTag);	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ChkBox_1 , либо название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	.Property	Свойство оконного объекта либо объекта вывода сведений о распределённых алармах. Перечень свойств объектов с их описаниями можно найти в Главе 2 "Поля тегов".
	DiscreteTag	Логическое значение либо название тега логического типа, значение которого будет присвоено указанному свойству в момент выполнения функции. Типичный значения: 0 = запрет объекта, 1 = разрешение объекта.
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
См. также	GetPropertyD(), GetPropertyI(), GetPropertyM(), SetPropertyI(), SetPropertyM()	

SetPropertyI()

	Во время исполнения программы присваивает указанное целое значение указанному свойству	
Категория	GOT	
Синтаксис	[ErrorNumber=]SetPropertyI("ControlName.Property", Integer);	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ChkBox_1 , либо название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	.Property	Свойство оконного объекта либо объекта вывода сведений о распределённых алармах. Перечень свойств объектов с их описаниями можно найти в Главе 2 "Поля тегов".
	Integer	Целое число либо название тега целого типа.
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
См. также	GetPropertyD(), GetPropertyI(), GetPropertyM(), SetPropertyD(), SetPropertyM()	

SetPropertyM()

	Во время исполнения программы присваивает указанное символьное значение указанному свойству	
Категория	GOT	
Синтаксис	[ErrorNumber=]SetPropertyM("ControlName.Property", "MessageTag");	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ChkBox_1 , либо название объекта вывода сведений о распределённых алармах, например AlmObj_1 .
	.Property	Свойство оконного объекта либо объекта вывода сведений о распределённых алармах. Перечень свойств объектов с их описаниями можно найти в Главе 2 "Поля тегов".
	MessageTag	Строка символов, которая должна быть записана в указанное свойство. Допустимые значения – литерал либо название тега типа "сообщение".
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
См. также	GetPropertyD(), GetPropertyI(), GetPropertyM(), SetPropertyI(), SetPropertyI()	

SetWindowPrinter()

	Разоешает пользователю динамически выбирать принтер, используемый во время исполнения функциями PrintWindow, PrintScreen и PrintHT functions at runtime. Принтер, отличный от принтера по умолчанию, может быть установлен в скрипте запуске приложения (On-Start) или каждый раз, когда это понадобится.	
Категория	дополнительная	
Синтаксис	SetWindowPrinter(PrinterName);	
	Параметр	Описание
	PrinterName	Имя узла и разделяемое имя принтера, имя принтера, как оно показывается в окне Properties (Свойства) принтера, или тэг сообщения, содержащий имя установленного принтера Windows или путь к разделяемому сетевому принтеру.
Комментарии	Команда/кнопка Print (печать) в окне Historical Trend Object (объект исторического тренда) также приведёт к установленному с помощью SetWindowPrinter принтеру, если до выбора Print (Печать) из объекта тренда вызывалась эта функция.	
Пример(ы)	<pre>SetWindowPrinter ("\\PRTSRV1\PRT22SW1");</pre> <p>где PRTSRV1 является именем узла и PRT22SW1 является разделяемым именем, присвоенным принтеру.</p> <pre>SetWindowPrinter("Epson LX-300")</pre> <p>где Epson LX-300 является именем принтера, как оно показывается в окне свойств принтера.</p> <pre>SetWindowPrinter(MyPrinter)</pre> <p>где MyPrinter является сообщением, содержащем имя установленного в windows принтера или путь к разделяемому сетевому принтеру.</p>	
См. также	PrintWindow(), PrintScreen(), PrintHT().	

Sgn()

	Определяет знак указанной величины (является ли она положительной, отрицательной либо равна нулю)	
Категория	математическая	
Синтаксис	IntegerResult=Sgn(Number);	
	Аргумент	Описание
	Number	Любое число либо название тега целого или вещественного типа.
Примечание	Если указанное число положительное, функция возвращает 1; если отрицательное, возвращается -1; если равно 0, возвращается 0.	
Примеры	Sgn(425) возвращает 1 Sgn(0) возвращает 0 Sgn(-37.3) возвращает -1	

Show

	Открывает на экране указанное окно (название окна должно быть в кавычках)	
Категория	прочие	
Синтаксис	Show "Window";	
	Аргумент	Описание
	Window	Название требуемого окна. Допустимые значения – строка символов либо название тега типа "сообщение".
Примечание	<p>В аргументе функции необходимо указание реально существующего окна. Если окно не существует в режиме исполнения, Window Viewer проигнорирует эту функцию. Если название окна изменилось, соответствующим образом необходимо изменить и аргумент функции.</p> <p>При указании в названия окна названия локального тега на экране появится следующее сообщение об ошибке: "Exprecting window name - must be string expression" (Отсутствует название окна – ожидается строка символов)</p> <p>Для того чтобы ошибка не возникала, перед названием тега типа "сообщение" следует указать символ "+". Например:</p> <p>DIM Test AS message; Test = "MyWindow1"; Show ""+Test;</p>	
Примеры	<p>Show "Alarm Summary Window";</p> <p>Если требуется лишь скрывать и показывать окна, то вместо этой функции рекомендуется пользоваться кнопками типа Show Window ("Показать окно") и Hide Window ("Скрыть окно"). В этом случае при изменении названия окна система InTouch автоматически сделает соответствующие изменения.</p>	
См. также	Hide, Hideself, ShowAt(), ShowHome, ShowTopLeftAt()	

ShowAt()

	Устанавливает вертикальные и горизонтальные координаты окна в пикселях при его открытии на экране	
Категория	прочие	
Синтаксис	ShowAt("Window", Horiz, Vert);	
	Аргумент	Описание
	Window	Название окна: строка символов или название тега типа "сообщение".
	Horiz	Горизонтальная координата окна в пикселях: число либо название тега целого типа.
	Vert	Вертикальная координата окна в пикселях: число либо название тега целого типа.
Примечание	Указанные координаты являются координатами центра окна. Окно не будет отцентрировано относительно указанных координат, если какая-либо из его сторон выходит за пределы экрана. В этом случае окно будет смещено соответствующим образом.	
Примеры	<p>В следующем примере горизонтальная координата в пикселях равна 100, а вертикальная – 200: ShowAt("Window Name", 100, 200);</p> <p>Для динамического изменения координат центра окна во время исполнения программы можно использовать теги аналогового типа (в следующем примере – TagHoriz и TagVert): ShowAt("Boiler Room 7 Details", TagHoriz, TagVert);</p> <p>В следующем операторе используются внутренние теги \$ObjHor и \$ObjVer, благодаря чему окно будет открываться строго над выбранным в текущий момент графическим объектом (его координаты при этом могут отображаться в полях вывода аналоговых значений). Указанный оператор может использоваться для открытия окна в соответствующем кнопочном скрипте, связанном с экранной кнопкой либо графическим объектом): ShowAt("Window Name", \$ObjHor, \$ObjVer);</p>	
См. также	\$ObjHor, \$ObjVer, ShowTopLeftAt(), Show, ShowHome, Hide, Hideself	

ShowHome

	Открывает "базовое" окна (окна). Базовые окна – те, которое должны автоматически открываться при запуске WindowViewer. (Выбор базовых окон осуществляется на странице с вкладкой Home Windows (Базовые окна) окна WindowViewer Properties (Свойства WindowViewer))
Категория	прочие
Синтаксис	ShowHome;
См. также	Show, ShowAt(), ShowTopLeftAt(), ShowHome, Hide, Hideself

ShowTopLeftAt()

	Устанавливает вертикальные и горизонтальные координаты (в пикселях) верхнего левого угла указанного окна при его открытии на экране	
Категория	прочие	
Синтаксис	ShowTopLeftAt("Window", Horiz, Vert);	
	Аргумент	Описание
	Window	Название окна: строка символов или название тега типа "сообщение".
	Horiz	Горизонтальная координата окна в пикселях: число либо название тега целого типа.
	Vert	Вертикальная координата окна в пикселях: число либо название тега целого типа.
Примечание	<p>При открытии окна его левый верхний угол размещается в точке с указанными координатами (по умолчанию, координаты левого верхнего угла экрана равны (0,0)). Данная функция полностью аналогична функции ShowAt() за исключением того, что указанные координаты являются координатами не центра окна, а его левого верхнего угла.</p> <p>Функция сначала загружает окно в память, а затем отображает начиная с указанной позиции, при этом исходные координаты левого верхнего угла (которые он имел в момент сохранения окна в WindowMaker) остаются неизменными независимо от текущего отображения окна.</p>	
См. также	Show, ShowAt(), ShowHome(), Hide, Hideself	

Sin()

	Возвращает значение <i>синуса</i> угла, выраженного в градусах	
Категория	математическая	
Синтаксис	Result=Sin(AngleNumber);	
	Аргумент	Описание
	AngleNumber	Угол, выраженный в градусах: число либо название тега целого или вещественного типа.
Примечание	В тег Result возвращается значение синуса угла, заданного аргументом AngleNumber .	
Примеры	Sin(90) возвращает 1 Sin(0) возвращает 0 wave = 100 * Sin (6 * \$Second);	
См. также	Cos(), Tan(), ArcCos(), ArcSin(), ArcTan()	

SPCConnect()

	Данная функция используется в сочетании с наборами данных системы автоматического сбора данных. Прежде чем требуемые сведения будут записываться в набор данных, необходимо выполнить эту функцию, чтобы указать системе SPC, каким пользователь является узел.	
Категория	SPC	
Синтаксис	SPCConnect("User", "Password");	
	Аргумент	Описание
	User	Имя пользователя базы данных: строка символов либо название тега типа "сообщение".
	Password	Пароль пользователя: строка символов либо название тега типа "сообщение".
Примечание	С помощью этой функции осуществляется подключение пользователя к базе данных и автоматический сбор информации в наборы данных под указанным идентификатором пользователя. Если для доступа к базе данных пароль не нужен, вызов функции может выглядеть следующим образом:	
Примеры	SPCConnect("User1", "");	
См. также	SPCDisconnect()	

SPCDataSetDlg()

	<p>Открывает окно SPCPro Dataset Configuration (Конфигурация набора данных SPCPro) в окне WindowViewer, в котором можно добавлять новые наборы и удалять существующие. Функция не имеет никаких входных и выходных параметров.</p> <p>Некоторые поля характеристики наборов данных и изделий будут закрашены в окне WindowViewer в серый цвет, и изменить их будет нельзя.</p> <p>В открытом окне можно определить новые наборы данных и изделия.</p> <p>При добавлении и сохранении сведений о новых наборах и изделиях:</p> <p>в режиме Autocollection (Автоматический сбор данных) цикл автоматического сбора данных будет перезапущен, при этом во время инициализации наборов данных возможна потеря информации.</p> <p>Добавление нового продукта во время работы программы лучше выполнять путём использования DDE-элемента SPC NewProduct, чем с помощью функции SPCDataSetDlg(), благодаря тому что перезапуск режима автоматического сбора данных не осуществляется.</p>
Категория	SPC
Синтаксис	SPCDataSetDlg();
Примечание	При выполнении данной функции в окне WindowViewer открывается окно SPCPro Dataset Configuration (Конфигурация набора данных SPCPro) .
Примеры	SPCDataSetDlg();
См. также	SPCSelectDataset()

SPCDisconnect()

	Отключает агента от базы данных SPCPro. При выполнении этой функции сбор информации в наборы данных, связанных с данным агентом, прекращается.
Категория	SPC
Синтаксис	SPCDisconnect();
Примечание	При выполнении данной функции осуществляется отключение пользователя от базы данных и прекращение накопления информации во всех связанных с ним наборах данных.
Примеры	SPCDisconnect();
См. также	SPCConnect()

SPCDisplayData()

	Прокручивает график до указанной даты и времени суток. Результат поиска SPC-данных может быть записан в тег InTouch. Тег Status имеет значение 0, если SPC-данные найдены, и 1, если для указанного временного периода SPC-данные не найдены.	
Категория	SPC	
Синтаксис	[Status=]SPCDisplayData("Dataset", "DateString", "TimeString", RangeInHours);	
	Аргумент	Описание
	Dataset	Название существующего набора данных: строка символов либо название тега типа "сообщение".
	DateString	Дата в формате ДД/ММ/ГГ: строка символов либо название тега типа "сообщение".
	TimeString	Время суток в формате ЧЧ:ММ:СС: строка символов либо название тега типа "сообщение".
	RangeInHours	Длительность требуемого периода времени в часах: число либо название тега целого типа.
Примеры	StatusTag = SPCDisplayData("Dataset", "DateString", "TimeString", RangeInHours);	

SPCLocateScooter()

	Данная функция используется для перемещения визира к требуемому номеру выборки. Тег визира, определённый в наборе данных, принимает значение выборки столбиковой диаграммы. Установка аргумента SampleNumber в 0 скрывает (отключает) визир	
Категория	SPC	
Синтаксис	SPCLocateScooter("Dataset", SampleNumber);	
	Аргумент	Описание
	Dataset	Название существующего набора данных: строка символов либо название тега типа "сообщение".
	SampleNumber	Допустимый номер выборки: число либо название тега целого типа.
Примеры	SPCMoveScooter()	

SPCMoveScooter()

	Данная функция используется для перемещения визира к требуемому номеру выборки. Тег визира, определённый в наборе данных, принимает значение выборки столбиковой диаграммы.	
Категория	SPC	
Синтаксис	SPCMoveScooter("Dataset", IncrementValue);	
	Аргумент	Описание
	Dataset	Название существующего набора данных: строка символов либо название тега типа "сообщение".
	IncrementValue	Смещение визира (положительное число для прокрутки графика в прямом направлении и отрицательное для прокрутки в обратном): число либо название тега целого типа.
Примеры	SPCLocateScooter()	

SPCSaveSample()

	Сохраняет значение выборки, введенной вручную. Данная функция используется совместно с функцией SPCSetMeasurement()	
Категория	SPC	
Синтаксис	SPCSaveSample("Dataset");	
	Аргумент	Описание
	Dataset	Название существующего набора данных: строка символов либо название тега типа "сообщение".
Примечание	При выполнении этой функции выборка вставляется в указанный набор данных. Функция использует введенные вручную значения переменных в качестве результатов измерений, составляющих выборку данных. Установка значений выполняется с помощью либо DDE-тегов MI_Mx (где x – это номер измерения от 1 до 25), либо функции SPCSetMeasurement() (для числа измерений от 1 до 300). Для обновления значений тегов MI_Mx необходимо инициализировать DDE-каналы повторно.	
См. также	SPCSetMeasurement()	

SPCSelectDataset()

	Открывает окно выбора непосредственного набора данных
Категория	SPC
Синтаксис	DatasetName=SPCSelectDataset();
Примечание	При выполнении этого оператора на экране открывается окно Select a Dataset (Выбор набора данных) . Название выбранного набора данных возвращается в тег DatasetName . Таким же образом можно изменить значение косвенного набора данных.
См. также	SPCSelectProduct(), SPCDatasetDlg()

SPCSelectProduct()

	Открывает окно выбора продукта в указанном наборе данных
Категория	SPC
Синтаксис	ProductName=SPCSelectProduct(Dataset);
Примечание	При выполнении этого оператора на экране открывается окно Select a Product (Выбор продукта) . Название выбранного продукта возвращается в тег ProductName , после чего его можно использовать для изменения накопления информации в наборе данных.
См. также	SPCSelectDataset(), SPCSetProductDisplayed(), SPCSetProductCollected()

SPCSetControlLimits()

	Функция ручного либо определяемого событиями ввода пределов управления для контрольных графиков (Control Chart)	
Категория	SPC	
Синтаксис	SPCSetControlLimits("Dataset", XUCL, XLCL);	
	Аргумент	Описание
	Dataset	Название существующего набора данных: строка символов либо название тега типа "сообщение".
	XUCL	Значение верхнего предела управления: число либо название тега вещественного типа.
	XLCL	Значение нижнего предела управления: число либо название тега вещественного типа.
Примечание	Эти параметры сохраняются в выборке при выполнении функции SPCSaveSample() .	
См. также	SPCSaveSample(), SPCSetRangeLimits(), SPCSetSpecLimits(), SPCSetMeasurement()	

SPCSetMeasurement()

	Функция ручного либо определяемого событиями ввода аналоговых значений измерений при выполнении скриптов	
Категория	SPC	
Синтаксис	SPCSetMeasurement("Dataset", Measurement, Value);	
	Аргумент	Описание
	Dataset	Название существующего набора данных: строка символов либо название тега типа "сообщение".
	Measurement	Номер измерения (от 1 до 300): число или название тега целого типа.
	Value	Значение, присваиваемое измерению с указанным номером: число или название тега вещественного типа.
Примечание	Сохранение всех измерений в базе данных выполняется с помощью функции SPCSaveSample() . Для обновления значений тегов MI_Mx необходимо инициировать DDE-каналы повторно.	
См. также	SPCSaveSample()	

SPCSetProductCollected()

	Изменяет название продукта, под которым осуществляется накопление сведений в указанном наборе данных	
Категория	SPC	
Синтаксис	SPCSetProductCollected("Dataset", "Product");	
	Аргумент	Описание
	Dataset	Название существующего набора данных: строка символов либо название тега типа "сообщение".
	Product	Название продукта, под которым должно осуществляться накопление данных: строка символов либо название тега типа "сообщение".
Примечание	Данная функция не меняет продукт, сведения о котором выводятся на экран. В приложении можно одновременно накапливать сведения для одного продукта, а отображать для другого (с помощью функции SPCSetProductDisplayed()).	
Примеры	SPCSetProductCollected("Data5838", "Widgets");	
См. также	SPCSelectProduct(), SPCSetProductDisplayed()	

SPCSetProductDisplayed()

	Устанавливает вывод сведений для другого продукта из указанного набора данных.	
Категория	SPC	
Синтаксис	SPCSetProductDisplayed("Dataset", "Product");	
	Аргумент	Описание
	Dataset	Название существующего набора данных: строка символов либо название тега типа "сообщение".
	Product	Название продукта, данные которого должны выводиться на экран: строка символов либо название тега типа "сообщение".
Примеры	SPCSetProductDisplayed("ADataSetName", "AProductName");	
См. также	SPCSelectProductt(), SPCSetProductCollected()	

SPCSetRangeLimits()

	Функция ручного или управляемого событиями ввода пределов регулирования для диапазонной диаграммы	
Категория	SPC	
Синтаксис	SPCSetRangeLimits("Dataset", RUCL, RLCL);	
	Аргумент	Описание
	Dataset	Название существующего набора данных: строка символов либо название тэга типа «сообщение».
	RUCL	Значение верхнего предела спецификации: число либо название тэга вещественного типа.
	RLCL	Значение нижнего предела спецификации: число либо название тэга вещественного типа.
См. также	SPCSetControlLimits(), SPCSetSpecLimits()	

SPCSetSpecLimits()

	Функция ручного или управляемого событиями ввода пределов спецификации для контрольных графиков	
Категория	SPC	
Синтаксис	SPCSetSpecLimits("Dataset", XUSL, XLSL);	
	Аргумент	Описание
	Dataset	Название существующего набора данных: строка символов либо название тэга типа «сообщение».
	XUSL	Значение верхнего предела спецификации: число либо название тэга вещественного типа.
	XLSL	Значение нижнего предела спецификации: число либо название тэга вещественного типа.
См. также	SPCSetControlLimits(), SPCSetRangeLimits()	

SQLAppendStatement()

	Дополняет текст указанной строки. Функцией возвращаются коды ошибок	
Категория	SQL	
Синтаксис	[ResultCode=]SQLAppendStatement(ConnectionID, "SQLStatement");	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	SQLStatement	Добавляемая строка.
Примеры	ResultCode=SQLAppendStatement(ConnectionID, "where tablename.columnname=(любое значение или строка символов)");	
См. также	SQLConnect(), SQLClearStatement()	

SQLClearParam()

	Сбрасывает значение указанного параметра. После вызова этой функции до обращения к функции SQLExecute() необходимо вызывать функцию SQLSetParam() .	
Категория	SQL	
Синтаксис	[ResultCode=]SQLClearParam(SQLHandle, ParameterNumber);	
	Аргумент	Описание
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .
	ParameterNumber	Параметр SQL-оператора, который должен быть сброшен.
См. также	SQLPrepareStatement(), SQLExecute()	

SQLClearStatement()

	Возвращает в систему ресурсы, выделенные SQL-выражению, который определяется аргументом SQLHandle	
Категория	SQL	
Синтаксис	[ResultCode=]SQLClearStatement(ConnectionID, SQLHandle);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .
См. также	SQLConnect() , SQLPrepareStatement()	

SQLClearTable()

	Удаляет все строки в таблице базы данных, сохраняя при этом саму таблицу	
Категория	SQL	
Синтаксис	[ResultCode=]SQLClearTable(ConnectionID, "TableName");	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	TableName	Название таблицы в базе данных.
Примеры	Для удаления всех строк таблицы BATCH1 можно воспользоваться следующим оператором: ResultCode=SQLClearTable(ConnectionID, "BATCH1");	
См. также	SQLConnect() , SQLClearStatement()	

SQLCommit()

	<p>Команда SQLCommit() определяет конец группы операций с базой данных. Группа команд между командой SQLTransact() и командой SQLCommit() называется набором транзакций. Набор транзакций выполняется как единое целое. После команды SQLTransact() результаты выполнения всех последующих операций не будут фиксироваться в базе данных, пока в потоке команд не встретится вызов SQLCommit().</p>	
Категория	SQL	
Синтаксис	[ResultCode=]SQLCommit(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
Примечание	<p>Командами SQLTransact() и SQLCommit() следует пользоваться обдуманно. Поскольку время выполнения команд увеличивается пропорционально числу транзакций, большое их число в наборе может существенно ухудшить общую производительность системы.</p>	
Примеры	<p>ResultCode = SQLTransact(ConnectionID); ResultCode = SQLInsertPrepare(ConnectionID, TableName, BindList, SQLHandle); ResultCode = SQLInsertExecute(ConnectionID, BindList, SQLHandle); ResultCode = SQLInsertExecute(ConnectionID, BindList, SQLHandle); ResultCode = SQLInsertExecute(ConnectionID, BindList, SQLHandle); ResultCode = SQLInsertEnd(ConnectionID, SQLHandle); ResultCode = SQLCommit(ConnectionID); {В базу данных вставлены три записи}</p>	
См. также	SQLRollback(), SQLTransact(), SQLCommit()	

SQLConnect()

	Подключает систему InTouch к базе данных, определяемой строкой ConnectionString	
Категория	SQL	
Синтаксис	[ResultCode=]SQLConnect(ConnectionID, "ConnectionString");	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	ConnectionString	Строка с указанием на базу данных и дополнительными сведениями, необходимыми функции SQLConnect() для регистрации в этой базе.
Примеры	<p>Следующий оператор осуществляет соединение с СУБД IBM OS/2 Database Manager и базой данных с названием SAMPLE:</p> <pre>[ResultCode=]SQLConnect(ConnectionID, "DSN=OS2DM;DB=SAMPLE");</pre> <p>В тэг ConnectionID функция возвращает некоторое значение, которое должно указываться в качестве аргумента во всех последующих SQL-функциях.</p> <p>Аргумент ConnectionString идентифицирует требуемую базу данных и содержит дополнительные регистрационные сведения в следующем формате:</p> <pre>"DSN=название источника данных[;атрибут=значение[;атрибут=значение]...]"</pre> <p>Атрибуты для разных баз данных могут быть разными. QELIB распознает следующие атрибуты для всех типов баз данных:</p>	
	Атрибут	Описание
	DSN	Имя источника данных, сконфигурированное в приложении Microsoft ODBC Administrator.
	DLG	Если параметр DLG установлен в 1, то на экране будет открыто окно, в котором пользователь может ввести сведения, составляющие строку подключения.
	DRV	Указываемое для совместимости с SQL Access версии 4.11 для InTouch, введённое значение используется, если название источника данных (DSN) не присутствует в строке подключения. QELIB заменяет его на название источника данных.
	UID	Регистрационный идентификатор.
	PWD	Пароль доступа.

	MODIFYSQL	Используется библиотекой QELIB для обеспечения совместимости между версией языка SQL, используемым в приложении, и версией языка SQL, поддерживаемого базой данных. Если этот параметр установлен в 1 (значение по умолчанию), драйвер базы данных ожидает синтаксис оператора, совместимый с ODBC, который он по необходимости модифицирует с учётом характеристик базы данных. Если этот параметр установлен в 0, драйвер базы данных ожидает и поддерживает операторы в синтаксисе конкретной базы данных. Это позволяет использовать приложения, операторы в которых написаны на языке SQL, поддерживаемом драйверами базы данных QELIB версии 1.0.
	REREADAFTE RUPDATE	Если этот параметр установлен в 1, QELIB повторно считывает запись из базы данных после её обновления. Это необходимо делать для получения правильного значения автоматически обновляемых столбцов, например столбцов с отметками времени.
	REREADAFTE RINSERT	Если этот параметр установлен в 1, QELIB повторно считывает запись из базы данных после её вставления. Это необходимо делать для получения правильного значения автоматически обновляемых столбцов, например столбцов с отметками времени.
	Более подробные сведения об атрибутах, поддерживаемых различными базами данных, можно найти в системном <i>Руководстве пользователя SQL Access Manager</i> .	
См. также	SQLDisconnect()	

SQLCreateTable()

	Создает новую таблицу базы данных с использованием сведений из шаблона таблицы. Шаблоны таблиц (хранящиеся в файле SQL.DEF) определяют структуру таблицы базы данных	
Категория	SQL	
Синтаксис	[ResultCode=]SQLCreateTable(ConnectionID, TableName, TemplateName);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	TableName	Название создаваемой таблицы базы данных.
	TemplateName	Название таблицы с шаблоном новой таблицы.
Примеры	<p>Следующий оператор создает таблицу с названием BATCH1, названия и типы столбцов которой указаны в шаблоне TEMPLATE:</p> <pre>[ResultCode=]SQLCreateTable(ConnectionID, "BATCH1", "TEMPLATE");</pre> <p>Если аргумент функции заключён в двойные кавычки (как, например, "Parameter1"), то используется указанная в кавычках строка. Если аргумент указан без кавычек (например, Parameter1), то считается, что Parameter1 – это тэг, и система будет обращаться в Словарь InTouch за его значением. Например:</p> <p>"c:\main\file" и location</p> <p>location – это тэг типа «сообщение», а строка "c:\main\file" – это литерал.</p>	
См. также	SQLConnect()	

SQLDelete()

	Удаляет одну или несколько записей из таблицы	
Категория	SQL	
Синтаксис	[ResultCode=]SQLDelete(ConnectionID, TableName, WhereExpr); Примечание: аргумент WhereExpr данной функции не может быть опущен либо равен пустой строке.	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	TableName	Название таблицы, из которой требуется удалить строки.
	WhereExpr	Условие, которое для каждой строки таблицы принимает значение либо «истина», либо «ложь». Функция удаляет из таблицы только те строки, для которых это условие истинно. Этот аргумент должен указываться в следующем формате: Название_столбца <i>оператор_сравнения</i> выражение. Примечание: если тип столбца – символьный, <i>выражение</i> должно быть заключено в одинарные кавычки.
Примеры	<p>В следующем примере будут выбраны все строки таблицы, в которых столбец имен содержит значение EmployeeID: name='EmployeeID'</p> <p>В следующем примере будут выбраны все строки таблицы, в которых в поле partno содержатся числа от 100 до 199: partno>=100 and partno<200</p> <p>В следующем примере будут выбраны все строки таблицы, в которых значение поля temperature превышает 350: temperature>350</p> <p>Следующий оператор удаляет из таблицы BATCH1 все записи, в которых номер партии равен 65: [ResultCode=]SQLDelete(ConnectionID, "BATCH1", "lotno=65");</p>	
См. также	SQLConnect()	

SQLDisconnect()

	Отключает пользователя от базы данных	
Категория	SQL	
Синтаксис	[ResultCode=]SQLDisconnect(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
См. также	SQLConnect()	

SQLDropTable()

	Удаляет указанную таблицу из базы данных вместе с её определением	
Категория	SQL	
Синтаксис	[ResultCode=]SQLDropTable(ConnectionID, TableName);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	TableName	Название удаляемой таблицы.
Примеры	Следующий оператор удаляет из базы данных таблицу BATCH1 . После команды удаления это название становится неизвестным системе, никакие операторы с данным названием выполняться не будут. [ResultCode=]SQLDropTable(ConnectionID, "BATCH1");	
См. также	SQLConnect()	

SQLEnd()

	Используется после функции SQLSelect() для освобождения ресурсов, выделенных для хранения таблицы результатов	
Категория	SQL	
Синтаксис	[ResultCode=]SQLEnd(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
См. также	SQLConnect() , SQLSelect()	

SQLExceptionMsg()

	Возвращает текстовое сообщение об ошибке, соответствующее указанному коду результат ResultCode . ErrorMsg – внутренний тэг типа «сообщение» (может хранить строки символов длиной не более 131 символа)	
Категория	SQL	
Синтаксис	SQLExceptionMsg(ResultCode);	
	Аргумент	Описание
	ResultCode	Целая величина, возвращаемая большинством SQL-функций: 0, если выполнение завершилось успешно, и отрицательное число, если во время выполнения функции возникла какая-либо ошибка.
Примеры	ErrorMsg=SQLExceptionMsg(ResultCode);	
См. также	SQLConnect()	

SQLExecute()

	Выполняет SQL-оператор. Если указанный оператор является оператором выбора (SELECT), то аргумент BindList обозначает название списка связей, определяющего взаимосвязь столбцов базы данных с тэгами InTouch. Если аргумент BindList имеет нулевое значение (NULL), то взаимосвязи столбцов и тэгов определены не будут. В качестве SQL-выражения могут использоваться любые операторы, например Create View (Создать подсхему), Insert (Вставить) и т. д. Функция возвращает код ошибки. Если оператор был “подготовлен”, то функции SQLExecute() в качестве аргумента необходимо передать указатель выражения (возвращаемый функцией SQLPrepareStatement()). Если оператор не был “подготовлен”, функции необходимо передать нулевой указатель (0)	
Категория	SQL	
Синтаксис	[ResultCode=]SQLExecute(ConnectionID, BindList, SQLHandle);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	BindList	Список связей. Указывает, какие тэги InTouch используются и каким столбцам таблицы они соответствуют.
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement()
См. также	SQLConnect(), SQLPrepareStatement()	
	Примечание: для выполнения оператора, который не был “подготовлен”, функция SQLExecute() может быть вызвана только один раз. Если оператор был заранее “подготовлен”, эта функция может вызываться много раз.	

SQLFirst()

	Выбирает первую запись из таблицы результатов, созданной при выполнении последней функций SQLSelect() . Функция SQLSelect() должна вызываться перед использованием этой команды	
Категория	SQL	
Синтаксис	[ResultCode=]SQLFirst(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
См. также	SQLConnect() , SQLSelect()	

SQLGetRecord()

	Считывает запись с номером RecordNumber из буфера текущей выборки	
Категория	SQL	
Синтаксис	[ResultCode=]SQLGetRecord(ConnectionID, RecordNumber);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	RecordNumber	Номер требуемой записи.
Примеры	[ResultCode=]SQLGetRecord(ConnectionID, 3);	
См. также	SQLConnect()	

SQLInsert()

	Вставляет новую запись в указанную таблицу с использованием значений тэгов из списка связей BindList . Список связей указывает, какие тэги InTouch используются и каким столбцам таблицы они соответствуют	
Категория	SQL	
Синтаксис	[ResultCode=]SQLInsert(ConnectionID, TableName, BindList);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	TableName	Название таблицы, в которую вставляется новая строка.
	BindList	Список связей. Указывает, какие тэги InTouch используются и каким столбцам таблицы они соответствуют.
Примеры	<p>Следующий оператор вставляет новую запись в таблицу ORG, заполнив её значениями тэгов, указанных в списке связей List1:</p> <pre>[ResultCode=]SQLInsert(ConnectionID, "ORG", "List1");</pre> <p>Примечание: вместо стандартной функции SQLInsert() для быстрой вставки записей в файл можно пользоваться указанными далее тремя функциями. Функция SQLInsert() – одноэтапная операция, заключающаяся в выполнении вставки и высвобождении всех выделенных её ресурсов. Поэтому при следующем вызове этой функции все требуемые действия будут проделаны с самого начала еще раз. Это займет гораздо больше времени, чем использование следующих трёх функций. Эти три функции выполняют требуемые действия отдельно: после подготовки операции функцией SQLInsertPrepare() вставку строк можно проделывать с помощью функции SQLInsertExecute() сколько угодно раз и очень быстро, а затем для высвобождения ресурсов вызвать функцию SQLInsertEnd().</p>	

SQLInsertEnd()

	Высвобождает ресурсы, выделенные выражению	
Категория	SQL	
Синтаксис	[ResultCode=]SQLInsertEnd(ConnectionID, SQLHandle);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement()
См. также	SQLConnect(), SQLPrepareStatement()	

SQLInsertExecute()

	Выполняет заранее подготовленный оператор	
Категория	SQL	
Синтаксис	[ResultCode=]SQLInsertExecute(ConnectionID, BindList, SQLHandle);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	BindList	Список связей. Указывает, какие тэги InTouch используются и каким столбцам таблицы они соответствуют.
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement()
См. также	SQLConnect(), SQLPrepareStatement()	

SQLInsertPrepare()

	Создает и подготавливает оператор вставки для выполнения. Вставка данных в таблицу не осуществляется. После выполнения функции InsertPrepare() тэг целого типа SQLHandle будет содержать указатель SQL-оператора	
Категория	SQL	
Синтаксис	[ResultCode=]SQLInsertPrepare(ConnectionID, TableName, BindList, SQLHandle);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	TableName	Название таблицы, в которую должны быть вставлены данные.
	BindList	Список связей. Указывает, какие тэги InTouch используются и каким столбцам таблицы они соответствуют.
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement()
См. также	SQLConnect(), SQLPrepareStatement()	

SQLLast()

	Выбирает последнюю запись в таблице результатов, созданной последним вызовом функции SQLSelect() . Функция SQLSelect() должна быть выполнена перед обращением к данной функции	
Категория	SQL	
Синтаксис	[ResultCode=]SQLLast(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
Примеры	ResultCode=SQLLast(ConnectionID);	
См. также	SQLConnect(), SQLSelect()	

SQLLoadStatement()

	Считывает SQL-оператор, содержащийся в файле FileName . В текущий момент этот оператор будет похож на создаваемый функцией SQLSetStatement() и может быть расширен с помощью функции SQLAppendStatement() или выполнен функцией SQLExecute() . В файле должен содержаться только один SQL-оператор. Однако, если функции SQLPrepareStatement() и SQLExecute() еще не вызывались, считанный из файла оператор можно дополнять с помощью функции SQLAppendStatement() произвольное число раз	
Категория	SQL	
Синтаксис	[ResultCode=]SQLLoadStatement(ConnectionID, FileName);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	FileName	Название файла с текстом SQL-оператора.
Примечание	Данная функция подготавливает к выполнению SQL-оператор, который был создан с помощью функции SQLSetStatement() либо функции SQLLoadStatement() . Функция возвращает указатель оператора.	
Примеры	ResultCode=SQLLoadStatement(ConnectionID, "C:\InTouchAppname\SQL.txt") SQL.txt = Select ColumnName from TableName where ColumnName>100;	
См. также	SQLConnect(), SQLAppendStatement(), SQLExecute(), SQLPrepareStatement	

SQLManageDSN()

	Запускает программу конфигурирования Microsoft ODBC Manager. С её помощью можно добавлять, удалять и модифицировать любые источники данных	
Категория	SQL	
Синтаксис	SQLManageDSN(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.

SQLNext()

	Выбирает следующую запись в таблице результатов, созданной последним вызовом функции SQLSelect() . Функция SQLSelect() должна вызываться прежде обращения к этой команде	
Категория	SQL	
Синтаксис	[ResultCode=]SQLNext(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
Примеры	[ResultCode=]SQLNext(ConnectionID);	
См. также	SQLConnect(), SQLSelect()	

SQLNumRows()

	Показывает, сколько строк в таблице удовлетворяет критерию, указанному в последнем вызове функции SQLSelect() . Например, число строк таблицы, в которых в поле AGE хранится число 45 (критерий выборки задаётся в конструкции WhereExpression), может быть как 40, так и 4000. Возвращаемое данной функцией значение позволяет решать, какую функцию вызывать следующей	
Категория	SQL	
Синтаксис	SQLNumRows(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
Примеры	Следующий оператор возвращает число найденных строк в тэг NumRows целого типа: NumRows=SQLNumRows(ConnectionID);	
См. также	SQLConnect()	

SQLPrepareStatement()

	Данная функция подготавливает SQL-оператор для его обработки функцией SQLSetParam() . Оператор может быть создан как функцией SQLSetStatement() , так и функцией SQLLoadStatement() . Возвращаемым значением является указатель оператора.	
Категория	SQL	
Синтаксис	[ResultCode=]SQLPrepareStatement(ConnectionID, SQLHandle);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement()
Примечание	Для высвобождения выделенных ресурсов необходимо вызвать соответствующую функцию SQLClearStatement() . Вызвать хранимую процедуру можно следующим образом: ResultCode = SQLSetStatement(ConnectionID, "exec sp_MyStoredProc");	
Примеры	[ResultCode=]SQLPrepareStatement(ConnectionID, SQLHandle);	
См. также	SQLConnect(), SQLSelect(), SQLSetStatement(), SQLLoadStatement()	

SQLPrev()

	Выбирает предшествующую запись в таблице результатов, созданной последним вызовом функции SQLSelect()	
Категория	SQL	
Синтаксис	[ResultCode=]SQLPrev(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
Примечание	Перед вызовом этой функции необходимо выполнить команду SQLSelect()	
Примеры	[ResultCode=]SQLPrev(ConnectionID);	
См. также	SQLConnect(), SQLSelect()	

SQLRollback()

	<p>Команда SQLRollback() отменяет результаты выполнения, или осуществляет “откат”, последнего набора транзакций. Набор транзакций – это множество команд, выполняемых между функциями SQLTransact() и SQLCommit(). Набор транзакций выполняется как одна команда. После выполнения функции SQLTransact() результаты всех последующих команд не будут фиксироваться в базе данных до тех пор, пока не встретится команда SQLCommit()</p>	
Категория	SQL	
Синтаксис	[ResultCode=]SQLRollback(ConnectionID,);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
Примеры	<p>ResultCode =SQLTransact(ConnectionID); ResultCode = SQLInsertPrepare(ConnectionID, TableName, BindList, SQLHandle); ResultCode = SQLInsertExecute(ConnectionID, BindList, SQLHandle); ResultCode = SQLInsertEnd(ConnectionID, SQLHandle); ResultCode =SQLRollback(ConnectionID);</p> <p>{Отменяет результаты всех выполненных после вызова функции SQLTransact() команд}</p>	
См. также	SQLCommit(), SQLTransact()	

SQLSelect()

	<p>Выбирает данные из таблицы базы данных. После выполнения этой функции в памяти создается временная таблица результатов со строками, которые можно находить и считывать с помощью функций SQLFirst(), SQLLast(), SQLNext() и SQLPrev().</p> <p>Примечание: после вызова функции SQLSelect() всегда выполняйте функцию SQLEnd(), для того чтобы высвободить ресурсы, выделенные для таблицы результатов</p>	
Категория	SQL	
Синтаксис	[ResultCode=]SQLSelect(ConnectionID, TableName, BindList, WhereExpr, OrderByExpr);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	TableName	Название таблицы, в которую должны быть вставлены данные.
	BindList	Список связей. Указывает, какие тэги InTouch используются и каким столбцам таблицы они соответствуют.
	WhereExpr	<p>Условие, которое для каждой строки таблицы принимает значение либо «истина», либо «ложь». Функция удаляет из таблицы только те строки, для которых это условие истинно. Этот аргумент должен указываться в следующем формате:</p> <p>Название_столбца <i>оператор_сравнения</i> выражение.</p> <p>Примечание: если тип столбца – символьный, <i>выражение</i> должно быть заключено в одинарные кавычки.</p>
		<p>В следующем примере будут выбраны все строки таблицы, в которых столбец имен содержит значение EmployeeID:</p> <p>name='EmployeeID'</p> <p>В следующем примере будут выбраны все строки таблицы, в которых в поле partno содержатся числа от 100 до 199:</p> <p>partno>=100 and partno<200</p> <p>В следующем примере будут выбраны все строки таблицы, в которых значение поля temperature превышает 350:</p> <p>temperature>350</p>

	<p>Примеры использования WhereExpression</p>	<p>WhereExpr – внутренний тэг типа «сообщение», OrderByExpr – внутренний тэг типа «сообщение», Speed_Input – внутренний тэг вещественного типа для ввода данных пользователем, Serial_Input – внутренний тэг типа «сообщение» для ввода данных пользователем</p>
	<p>Использование чисел</p>	<p>WhereExpr = "Speed = " + text(Speed_Input, "#.##"); Подсказка: так как значение Speed_Input – число, оно должно быть преобразовано в символьный вид для конкатенации со строкой.</p>
	<p>Использование строк символов</p>	<p>WhereExpr = "Ser_No = ' " + Serial_input + "'"; Подсказка: так как значение Serial_input – строка символов, она должна быть заключена в одинарные кавычки, например WhereExpr = "Ser_No='125gh'";</p>
	<p>Использование строк и конструкции LIKE</p>	<p>WhereExpr = "Ser_No like ' " + "125%' " Подсказка: в операторах этого типа символ процента "%" можно интерпретировать как символ подстановки.</p>
	<p>Использование чисел и строк символов и оператора AND</p>	<p>WhereExpr = "Ser_No = ' " + Serial_input + "' " + " and " + "Speed = " + text(Speed_Input, "#.##"); OrderByExpr = ""; Подсказка: если порядок сортировки неважен, определяйте его так, как указано в этом примере.</p>
	<p>SQLSelect() с тэгом WhereExpr</p>	<p>ResultCode = SQLSelect(Connect_Id, TableName, BindList, WhereExpr, OrderByExpr); Error_msg = SQLErrorMsg(ResultCode);</p>
	<p>SQLSelect() с тэгом WhereExpr в функции</p>	<p>ResultCode = SQLSelect(Connect_Id, TableName, BindList, "Ser_No = ' " + Serial_input + "' ", OrderByExpr); Error_msg = SQLErrorMsg(ResultCode);</p>
<p>После вызова функции SQLSelect() всегда выполняйте команду SQLEnd(). В противном случае выделяемые функции SQLSelect() ресурсы не будут возвращаться в систему, и может возникнуть ситуация нехватки памяти.</p>		

	OrderByExpr	<p>Аргумент, определяющий столбцы и порядок сортировки. Указываться должны только названия столбцов в следующем формате: Название_столбца [ASC DESC]</p> <p>Следующая запись задаёт упорядочивание строк таблицы по возрастающим значениям столбца manager: "manager ASC"</p> <p>Упорядочивать строки таблицы можно по значениям нескольких столбцов, указывая их следующим образом: Название_столбца1 [ASC DESC], Название_столбца2 [ASC DESC]...</p> <p>Следующая запись задаёт упорядочивание строк таблицы по возрастающим значениям столбца temperature и убывающим значениям столбца time: "temperature ASC, time DESC"</p>
Примеры		<p>Следующий оператор в соответствии со списком связей List1 выбирает строки из таблицы BATCH, в которых в ячейках столбца type хранятся строки 'cookie'. Таблица результатов будет упорядочена по возрастающим значениям столбца amount и убывающим значениям столбца sugar:</p> <pre>[ResultCode=]SQLSelect(ConnectionID, "BATCH", "List1", "type='cookie'", "amount ASC, sugar DESC");</pre> <p>Следующий оператор выбирает все данные в базе данных, причём критерии выборки и порядок сортировки отсутствуют:</p> <pre>[ResultCode=]SQLSelect(ConnectionID, "BATCH", "List1", "", "");</pre>
См. также		SQLFirst(), SQLConnect(), SQLLast(), SQLNext(), SQLPrev(), SQLEnd(), SQLSelect()

SQLSetParamChar()

	<p>Записывает в указанный параметр указанную строку. Перед исполнением оператора функцию можно вызывать несколько раз, при этом окончательное значение параметра будет равным конкатенации всех ранее указанных в аргументе функции строк. Значение длины строки, равное 0, игнорируется.</p>	
Категория	SQL	
Синтаксис	[ResultCode=]SQLSetParamChar(SQLHandle, ParameterNumber, ParameterValue, MaxLen);	
	Аргумент	Описание
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .
	ParameterNumber	Номер параметра в операторе.
	ParameterValue	Значение, передаваемое параметру.
	MaxLen	Максимальный размер столбца, которому соответствует данный параметр. Указанное значение определяет, является ли тип столбца varying character либо long varying character . Если значение аргумента MaxLen не превышает максимально допустимой длины символьных строк для базы данных, то тип параметра принимается varying character , в противном случае – long varying character .
Примеры	ResultCode=SQLSetParamChar(SQLHandle, ParameterNumber, ParameterValue, MaxLen);	
См. также	SQLPrepareStatement()	

SQLSetParamDate()

	Записывает в указанный параметр типа date указанную строку	
Категория	SQL	
Синтаксис	[ResultCode=]SQLSetParamDate(SQLHandle, ParameterNumber, ParameterValue);	
	Аргумент	Описание
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .
	ParameterNumber	Номер параметра в операторе.
	ParameterValue	Значение, передаваемое параметру.
Примеры	ResultCode=SQLSetParamDate(SQLHandle, ParameterNumber, ParameterValue);	
См. также	SQLPrepareStatement()	

SQLSetParamDateTime()

	Записывает в указанный параметр типа datetime указанную строку	
Категория	SQL	
Синтаксис	[ResultCode=]SQLSetParamDateTime(SQLHandle, ParameterNumber, ParameterValue, Precision);	
	Аргумент	Описание
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .
	ParameterNumber	Номер параметра в операторе.
	ParameterValue	Значение, передаваемое параметру.
	Precision	Длина присваиваемого значения даты и времени. Равно количеству символов, которое должно быть взято из аргумента ParameterValue
Примеры	ResultCode=SQLSetParamDateTime(SQLHandle, ParameterNumber, ParameterValue, Precision);	
См. также	SQLPrepareStatement()	

SQLSetParamDecimal()

	Записывает в указанный десятичный параметр (decimal) указанную строку. Аргумент Precision определяет общее число позиций в значении, аргумент Scale – число позиций справа от десятичной точки.	
Категория	SQL	
Синтаксис	[ResultCode=]SQLSetParamDecimal(SQLHandle, ParameterNumber, ParameterValue, Precision, Scale);	
	Аргумент	Описание
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .
	ParameterNumber	Номер параметра в операторе.
	ParameterValue	Значение, передаваемое параметру.
	Precision	Длина присваиваемого значения. Равно количеству символов, которое должно быть взято из аргумента ParameterValue
	Scale	Число позиций справа от десятичной точки
Примеры	ResultCode=SQLSetParamDecimal(SQLHandle, ParameterNumber, ParameterValue, Precision, Scale);	
См. также	SQLPrepareStatement()	

SQLSetParamFloat()

	Записывает в указанный параметр указанную строку	
Категория	SQL	
Синтаксис	[ResultCode=]SQLSetParamFloat(SQLHandle, ParameterNumber, ParameterValue);	
	Аргумент	Описание
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .
	ParameterNumber	Номер параметра в операторе.
	ParameterValue	Значение, передаваемое параметру.
Примеры	ResultCode=SQLSetParamFloat(SQLHandle, ParameterNumber, ParameterValue);	
См. также	SQLPrepareStatement()	

SQLSetParamInt()

	Записывает в указанный параметр указанную строку	
Категория	SQL	

Синтаксис	[ResultCode=]SQLSetParamInt(SQLHandle, ParameterNumber, ParameterValue);	
	Аргумент	Описание
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .
	ParameterNumber	Номер параметра в операторе.
	ParameterValue	Значение, передаваемое параметру.
Примеры	ResultCode=SQLSetParamInt(SQLHandle, ParameterNumber, ParameterValue);	
См. также	SQLPrepareStatement()	

SQLSetParamLong()

	Записывает в указанный параметр указанную строку	
Категория	SQL	
Синтаксис	[ResultCode=]SQLSetParamLong(SQLHandle, ParameterNumber, ParameterValue);	
	Аргумент	Описание
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .
	ParameterNumber	Номер параметра в операторе.
	ParameterValue	Значение, передаваемое параметру.
Примеры	ResultCode=SQLSetParamLong(SQLHandle, ParameterNumber, ParameterValue);	
См. также	SQLPrepareStatement()	

SQLSetParamNull()

	Записывает в указанный параметр значение NULL		
Категория	SQL		
Синтаксис	[ResultCode=]SQLSetParamNull(SQLHandle, ParameterNumber, ParameterType, Precision, Scale);		
	Аргумент	Описание	
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .	
	ParameterNumber	Номер параметра в операторе.	
	ParameterType	Тип данных параметра:	
		Char	Строка фиксированной длины, заполненная пробелами
		Var Char	Строка переменной длины
		Decimal	BCD-число (в двоично-десятичном формате)
		Integer	Целое число со знаком, занимающее 4 байта
		Small Integer	Целое число со знаком, занимающее 2 байта
		Float	Вещественное число с плавающей запятой, занимающее 4 байта
		Double Precision Float	Вещественное число с плавающей запятой, занимающее 8 байтов
		DateTime	Значение даты и времени, занимающее 4 байта
		Date	Значение даты и времени, занимающее 2 байта
		Time	Значение даты и времени, занимающее 2 байта
		No Type	Типа данных нет
		Precision	Точность десятичного значения, максимальная длина строки в символах либо длина значения даты и времени в байтах
	Scale	Количество знаков справа от десятичной точки. Данная величина требуется, только если она применима к устанавливаемому в нулевое значение параметру.	

Примечание	Значение No Type может указываться, если нулевое значение уже присваивалось данному параметру с помощью вызова SQLSetParam() .
Примеры	ResultCode=SQLSetParamNull(SQLHandle, ParameterNumber, ParameterType, Precision, Scale);
См. также	SQLPrepareStatement()

SQLSetParamTime()

	Записывает в указанный параметр типа time указанную строку	
Категория	SQL	
Синтаксис	[ResultCode=]SQLSetParamTime(SQLHandle, ParameterNumber, ParameterValue);	
	Аргумент	Описание
	SQLHandle	Целая величина, возвращаемая функцией SQLPrepareStatement() .
	ParameterNumber	Номер параметра в операторе.
	ParameterValue	Значение, передаваемое параметру.
Примеры	ResultCode=SQLSetParamTime(SQLHandle, ParameterNumber, ParameterValue);	
См. также	SQLPrepareStatement()	

SQLSetStatement()

	Создаёт для соединения ConnectionID буфер SQL-оператора, содержащегося в указанной строке. Для каждого соединения с базой данных может существовать только один буфер. Возвращаемым значением функции является код ошибки	
Категория	SQL	
Синтаксис	[ResultCode=]SQLSetStatement(ConnectionID, SQLStatement);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	SQLStatement	Текст SQL-оператора
Примеры	<p>[ResultCode=]SQLSetStatement(ConnectionID, "Select LotNo, LotName from LotInfo");</p> <p>В следующем выражении значение указателя SQLHandle установлено равным 0, для того чтобы не вызывать функцию SQLPrepare(Connect_Id, SQLHandle) перед выполнением этого оператора. Поскольку указатель SQLhandle не был создан функцией SQLPrepare(), то для правильного завершения указанного оператора выборки вместо функции SQLClearStatement() следует использовать функцию SQLEnd().</p> <p>•</p> <p>SQLSetStatement(Connect_Id, "Select Speed, Ser_No from tablename where Ser_No =" + Serial_input + "");</p> <p>SQLExceute(Connect_Id,0);</p> <p>В следующем выражении указатель SQLhandle создается функцией SQLPrepareStatement() и используется в функции SQLExecute(). Для завершения указанного оператора выборки следует использовать функцию SQLClearStatement(), чтобы освободить ресурсы и освободить SQLhandle.</p> <p>SQLSetStatement(Connect_Id, "Select Speed, Ser_No from tablename where Ser_No =" + Serial_input + "");</p> <p>SQLPrepareStatement(Connect_Id, SQLHandle);</p> <p>SQLExecute(Connect_Id, SqlHandle);</p>	
См. также	SQLConnect()	

SQLTransact()

	<p>Команда SQLTransact() определяет начало группы транзакции. Группа команд, выполняемая между вызовом SQLTransact() и вызовом SQLCommit(), называется набором транзакций. Набор транзакций выполняется как единая команда. После вызова функции SQLTransact() результаты всех последующих команд не будут фиксироваться в базе данных, пока не будет выполнена команда SQLCommit()</p>	
Категория	SQL	
Синтаксис	[ResultCode=]SQLTransact(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	<p>Командами SQLTransact() и SQLCommit() следует пользоваться обдуманно. Поскольку время выполнения команд увеличивается пропорционально числу транзакций, большое их число в наборе может существенно ухудшить общую производительность системы.</p>	
См. также	SQLCommit(), SQLRollback()	

SQLUpdate()

	Модифицирует запись в базе данных, обновляя её текущим значениями тэгов, указанных в списке связей BindList	
Категория	SQL	
Синтаксис	[ResultCode=]SQLUpdate(ConnectionID, TableName, BindList, WhereExpr);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
	TableName	Название таблицы, в которой обновляется строка.
	BindList	Список связей. Указывает, какие тэги InTouch используются и каким столбцам таблицы они соответствуют.
	WhereExpr	<p>Условие, которое для каждой строки таблицы принимает значение либо «истина», либо «ложь». Функция удаляет из таблицы только те строки, для которых это условие истинно. Этот аргумент должен указываться в следующем формате:</p> <p>Название_столбца <i>оператор_сравнения</i> выражение.</p> <p>Примечание: если тип столбца – символьный, <i>выражение</i> должно быть заключено в одинарные кавычки.</p> <p>В следующем примере будут выбраны все строки таблицы, в которых столбец имен содержит значение EmployeeID:</p> <p>name='EmployeeID'</p> <p>В следующем примере будут выбраны все строки таблицы, в которых в поле partno содержатся числа от 100 до 199:</p> <p>partno>=100 and partno<200</p> <p>В следующем примере будут выбраны все строки таблицы, в которых значение поля temperature превышает 350:</p> <p>temperature>350</p>
Примеры	<p>Следующий оператор обновляет в таблице BATCH1 все записи, в которых номер партии равен 65, значениями тэгов, указанных в списке связей List1:</p> <p>[ResultCode=]SQLDelete(ConnectionID, "BATCH1", "List1", "lotno=65");</p> <p>Примечание: если в таблице данных будут присутствовать одинаковые записи, то они будут изменены все.</p>	
См. также	SQLConnect()	

SQLUpdateCurrent()

	Обновляет текущую запись таблицы существующими новыми значениями тэгов InTouch. В примере обновляются все выделенные в текущий момент записи	
Категория	SQL	
Синтаксис	[ResultCode=]SQLUpdateCurrent(ConnectionID);	
	Аргумент	Описание
	ConnectionID	Название внутреннего тэга целого типа, используемого для хранения идентификатора, который функция SQLConnect() создаёт для каждого соединения с базой данных.
Примеры	ResultCode=SQLUpdateCurrent(ConnectionID);	
См. также	SQLConnect()	

Sqrt()

	Возвращает значение квадратного корня из указанного числа	
Категория	SQL	
Синтаксис	RealResult=Sqrt(Number);	
	Аргумент	Описание
	Number	Любое число или название тэга целого или вещественного типа.
Примеры	AnalogTag1=Sqrt(AnalogTag2);	
См. также		

StartApp

	Автоматически запускает другое приложение Windows	
Категория	SQL	
Синтаксис	StartApp "AppName";	
	Аргумент	Описание
	AppName	Название файла запускаемого приложения, например Wordpad.exe .
	<p>Рекомендуется указывать расширение программы .EXE. Можно также добавить в командную строку ключи программы, если они допустимы для данного приложения. Длинные названия файлов не поддерживаются, однако можно использовать их эквиваленты в DOS.</p> <p>Если длинное имя пути файла – C:\Program files\Microsoft Office\Office\Excel то указывать следует строку C:\Progra~1\Micros~2\Office\Excel (эквивалент в DOS):</p> <p>StartApp "C:\Progra~1\Micros~2\Office\Excel";</p> <p>При отображении в операционной системе Windows списка файлов в виде таблицы в её центральном столбце будут показаны эквивалентные для DOS значения путей доступа.</p>	
Примеры	<p>Следующий оператор запускает программу Microsoft Windows Wordpad.</p> <p>StartApp "Wordpad.exe";</p>	
См. также	ActivateApp()	

StringASCII()

	Возвращает ASCII-код первого символа в указанном тэге типа «сообщение»	
Категория	символьная	
Синтаксис	IntegerResult=StringASCII("Char");	
	Аргумент	Описание
	Char	Алфавитно-цифровой символ либо тэг типа «сообщение».
Примечание	В тэг IntegerResult возвращается ASCII-код первого символа аргумента Char . Функция обрабатывает только один символ. Если в тэге типа «сообщение» содержится строка, состоящая из более чем одного символа, функция вернёт ASCII-код только самого первого символа строки.	
Примеры	StringASCII("A") возвращает 65 StringASCII("A Mixer is Running") возвращает 65 StringASCII("a mixer is running") возвращает 97	
См. также	StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringChar()

	Возвращает символ, соответствующий указанному ASCII-коду	
Категория	SQL	
Синтаксис	MessageResult=StringChar(ASCII);	
	Аргумент	Описание
	ASCII	ASCII-код либо название тэга целого типа.
Примечание	В тэг MessageResult возвращается символ, ASCII-код которого задан аргументом ASCII . Одним из примеров применения этой функции может быть добавление к символьной строке таких символов ASCII, которые нельзя обычным образом ввести с клавиатуры.	
Примеры	<p>ControlString=MessageTag + StringChar(13) + StringChar(10);</p> <p>Данный оператор добавляет символы возврата каретки (ASCII-код 13) и перевода строки (ASCII-код 10) в конец строки MessageTag и записывает результат в тэг ControlString. Как правило, вставка символов, коды которых не попадают в диапазон 32-126 (печатные символы), выполняется для программного управления такими внешними устройствами, как принтеры или модемы.</p> <p>Данная функция также широко используется в SQL-операторах. Иногда в конструкциях Where требуется указание строк, содержащих двойные кавычки, в этом случае указывается StringChar(34).</p>	
См. также	StringASCII(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringCompare()

	Сравнивает строковое значение с другой строка.	
Категория	Строка	
Синтаксис	<code>StringCompare (Text1, Text2);</code>	
	Параметр	Описание
	Text1	Первая сравниваемая строка.
	Text2	Вторая сравниваемая строка.
Комментарии	Возвращаемое значение равно нулю, если строки идентичны, -1, если значение Text1 меньше, чем Text2, или 1, если значение Text1 больше, чем Text2.	
Пример(ы)	<pre>Результат = StringCompare ("Text1", "Text2"); (или) Результат = StringCompare (MText1, MText2);</pre> <p>где Результат является целым или вещественным тэгом и MText1 и MText2 являются внутренними (Memory) тэгами сообщений.</p>	
См. также	StringASCII(), StringChar(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringCompareEncrypted()

	Сравнивает зашифрованную строку с незашифрованной строкой и и возвращает Булевское значение.	
Категория	Строка	
Синтаксис	<code>StringCompareEncrypted (sMessageTag, sEncryptedMessageTag);</code>	
	Параметр	Описание
	sMessageTag	Открытый текст, с которым сравнивается зашифрованный текст.
	sEncryptedMessageTag	Текст из зашифрованного ввода пользователя.
Пример(ы)	<pre>Результат = StringCompareEncrypted(PlainTxt, Passwd);</pre> <p>Где <i>Passwd</i> является тэгом сообщения, содержащим значение из зашифрованного ввода пользователя, <i>PlainTxt</i> является тэгом сообщения, с которым сравнивается пользовательский ввод, и <i>Результат</i> является логическим тэгом, который устанавливается в TRUE (истина), когда открытый текст и зашифрованный текст совпадают, в противном случае – FALSE (ложь).</p>	
См. также	StringASCII(), StringChar(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringCompareNoCase()

	Сравнивает значение строки с другой строкой, не учитывая регистр.	
Категория	Строка	
Синтаксис	StringCompareNoCase(Text1, Text2);	
	Параметр	Описание
	Text1	Первая сравниваемая строка.
	Text2	Вторая сравниваемая строка.
Комментарии	Возвращаемое значение равно нулю, если строки идентичны (без учёта регистра), -1, если значение Text1 меньше (без учёта регистра), чем Text2, или 1, если значение Text1 больше (без учёта регистра), чем Text2.	
Пример(ы)	Результат = StringCompareNoCase ("Text1", "TEXT1"); (или) Результат = StringCompareNoCase (MText1, MText2); Где <i>Результат</i> является целым (Integer) или вещественным (Real) тэгом и MText1 и MText2 являются внутренними (Memory) тэгами сообщения.	
См. также	StringASCII(), StringChar(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringFromGMTTimeToLocal()

	Преобразовывает метку времени в UTC-формате (Coordinated Universal Time – универсальное координатное время) в локальное время и выводит результат в виде строки. Эта функция переводит метку времени, выражающую количество секунд, прошедших с 1 января 1970 года в строку, выражающую эту метку времени как дату, время и день недели в соответствии с выбранным методом представления. Это аналогично StringFromTime().		
Категория	строка		
Синтаксис	MessageРезультат=StringFromGMTTimeToLocal (SecsSince1-1-70, StringType) ;		
	Параметр	Описание	
	<i>SecsSince1-1-70</i>	Преобразуется в заданный StringType, и результат записывается в MessageResult.	
	<i>StringType</i>	Задаёт метод представления.	
		Тип	Описание
		1	Показывает дату в том же формате, что и установлен в Панели управленияWindows. (Аналогично представлению для \$DateString.)
		2	Показывает время в том же формате, что и установлен в Панели управленияWindows. (Аналогично представлению для \$DateString.)
		3	Показывает 24-символьную строку, представляющую как дату, так и время: "Wed Jan 02 02:03:55 1993" (среда, 2 января, 2ч.3мин.55 с 1993 г.)
		4	Показывает краткую форму дня недели: "Wed" (среда).
		5	Показывает длинную форму дня недели: "Wednesday" (среда).

<p>Пример(ы)</p>	<p>Этот пример предполагает, что временная зона локального узла является тихоокеанским стандартным временем (Pacific Standard Time – UTC-0800). UTC-время, передаваемое в функцию, составляет 12:00:00 в пятницу 02.01.1970. Так как PST на 8 часов позже UTC, функция возвратит следующие результаты:</p> <p>StringFromGMTTimeToLocal(86400, 1) возвратит "1/1/70"</p> <p>StringFromGMTTimeToLocal(86400, 2) возвратит "04:00:00 PM"</p> <p>StringFromGMTTimeToLocal(86400, 3) возвратит "Thu Jan 01 16:00:00 1970"</p> <p>StringFromGMTTimeToLocal(86400, 4) возвратит "Thu"</p> <p>StringFromGMTTimeToLocal(86400, 5) возвратит "Thursday"</p>
<p>Комментарии</p>	<p>Любые настройки, связанные с переходом на летнее время (Daylight Savings Time – DST), будут автоматически применяться к возвращаемому результату. Следовательно, нет необходимости вносить во входное значение какие-либо ручные поправки, связанные с DST.</p>
<p>См. также</p>	<p>StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()</p>

StringFromIntg()

	Преобразует целое значение в символьное представление по другому основанию	
Категория	символьная	
Синтаксис	MessageResult=StringFromIntg(Number, Base);	
	Аргумент	Описание
	Number	Преобразуемая величина: число либо название тэга целого типа.
	Base	Основание счисления: число либо название тэга целого типа.
Примечание	Функция преобразует значение аргумента Number в его символьное представление по указанному основанию и записывает результат в тэг MessageResult типа «сообщение».	
Примеры	StringFromIntg(26, 2) возвращает "11010" StringFromIntg(26, 8) возвращает "32" StringFromIntg(26, 16) возвращает "1A"	
См. также	StringASCII(), StringChar(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringFromReal()

	Преобразует указанную вещественную величину в символьное представление либо в формате с плавающей запятой, либо в экспоненциальной форме		
Категория	символьная		
Синтаксис	MessageResult=StringFromReal(Number, Precision, "Type");		
	Аргумент	Описание	
	Number	Величина, преобразуемая в символьное представление указанного типа с указанной точностью: число или название тэга вещественного типа.	
	Precision	Число позиций в символьном представлении: число или название тэга целого типа.	
	Type	Тип представления:	
		f	в формате с плавающей запятой,
		e	в экспоненциальной форме с показателем степени "e" (в нижнем регистре).
		E	в экспоненциальной форме с показателем степени "E" (в верхнем регистре).
Примеры	StringFromReal(263.355, 2,"f") возвращает "263.36" StringFromReal(263.355, 2,"e") возвращает "2.63e2" StringFromReal(263.55, 3,"E") возвращает "2.636E2"		
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()		

StringFromTime()

	Преобразует значение времени (число секунд, прошедших с 1 января 1970) в символьное представление в указанном формате. Время должно указываться в UTC-эквиваленте (число секунд, прошедших с 1 января 1970 по гринвичскому времени). Возвращаемая величина указывается по местному времени	
Категория	символьная	
Синтаксис	MessageResult=StringFromTime(SecsSince1-1-70, StringType);	
	Аргумент	Описание
	SecsSince1-1-70	Число секунд, прошедших с 1 января 1970.
	StringType	Тип представления результата:
	1	показывается дата в формате, определённом в Панели управления Windows. (Аналогично представлению в тэге \$DateString.)
	2	показывается время в формате, определённом в Панели управления Windows. (Аналогично представлению в тэге \$TimeString.)
	3	выводятся значения даты и времени в 24-символьной строке следующего формата: "Wed Jan 02 02:03:55 1993"
	4	выводится сокращённое название дня недели: "Wed"
	5	выводится полное название дня недели: "Wednesday"
Примеры	StringFromTime(86400, 1) возвращает "1/2/70" StringFromTime(86400, 2) возвращает "12:00:00 AM" StringFromTime(86400, 3) возвращает "Fri Jan 02 00:00:00 1970" StringFromTime(86400, 4) возвращает "Fri" StringFromTime(86400, 5) возвращает "Friday"	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringFromTimeLocal()

	Преобразует значение времени (число секунд, прошедших с 1 января 1970) в символьное представление в указанном формате. Возвращаемая величина указывается по местному времени		
Категория	символьная		
Синтаксис	MessageResult=StringFromTimeLocal(SecsSince1-1-70, StringType);		
	Аргумент	Описание	
	SecsSince1-1-70	Число секунд, прошедших с 1 января 1970.	
	StringType	Тип представления результата:	
		1	показывается дата в формате, определённом в Панели управления Windows. (Аналогично представлению в тэге \$DateString.)
		2	показывается время в формате, определённом в Панели управления Windows. (Аналогично представлению в тэге \$TimeString.)
		3	выводятся значения даты и времени в 24-символьной строке следующего формата: "Wed Jan 02 02:03:55 1993"
		4	выводится сокращённое название дня недели: "Wed"
	5	выводится полное название дня недели: "Wednesday"	
Примеры	StringFromTimeLocal(86400, 1) возвращает "1/2/70" StringFromTimeLocal (86400, 2) возвращает "12:00:00 AM" StringFromTimeLocal (86400, 3) возвращает "Fri Jan 02 00:00:00 1970" StringFromTimeLocal (86400, 4) возвращает "Fri" StringFromTimeLocal (86400, 5) возвращает "Friday"		
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()		

StringInString()

	Возвращает позицию в строке Text , начиная с которой впервые встречается подстрока поиска SearchFor	
Категория	символьная	
Синтаксис	IntegerResult=StringInString("Text", "SearchFor", StartPos, CaseSens);	
	Аргумент	Описание
	Text	Строка, в которой осуществляется поиск подстроки SearchFor . Если эта подстрока присутствует в строке Text несколько раз, то функция возвращает позицию первого вхождения.
	SearchFor	Искомая подстрока: литерал либо название тэга типа «сообщение».
	StartPos	Начальная позиция поиска подстроки в указанной строке: число или название тэга целого типа.
	CaseSens	Данный аргумент определяет, должен ли поиск осуществляться с учётом регистра символов в подстроке (1) или нет (0): число или название тэга целого типа.
Примеры	StringInString("The mixer is running", "mix", 1, 0) возвратит 5 StringInString("Today is Thursday", "day", 1, 0) возвратит 3 StringInString("Today is Thursday", "day", 10, 0) возвратит 15 StringInString("Today is Veteran's Day", "Day", 1, 1) возвратит 20 StringInString("Today is Veteran's Day", "Night", 1, 1) возвратит 0	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringLeft()

	Возвращает указанное число символов строки Text , начиная с первой позиции	
Категория	символьная	
Синтаксис	MessageResult=StringLeft("Text", Chars);	
	Аргумент	Описание
	Text	Исходная строка символов: литерал или название тэга типа «сообщение».
	Chars	Число возвращаемых символов: число или название тэга целого типа.
Примечание	Если значение аргумента Chars равно 0, то возвращается вся исходная строка.	
Примеры	StringLeft("The Control Pump is On", 3) возвращает "The" StringLeft("Pump 01 is On", 4) возвращает "Pump" StringLeft("Pump 01 is On", 96) возвращает "Pump 01 is On" StringLeft("The Control Pump is On", 0) возвращает "The Control Pump is On"	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringInString(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringLen()

	Возвращает длину указанной строки символов	
Категория	символьная	
Синтаксис	IntegerResult=StringLen("Text");	
	Аргумент	Описание
	Text	Строка символов: литерал либо название тэга типа «сообщение».
Примечание	Функция возвращает длину строки Text в тэг IntegerResult целого типа. Учитываются все символы строки, включая непечатаемые.	
Примеры	StringLen("Twelve percent") возвращает 14 StringLen("12%") возвращает 3 StringLen("The end." + StringChar(13)) возвращает 9 13 – ASCII-код символа CR (возврат каретки).	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringLower()

	Преобразует в указанной строке все символы в верхнем регистре в символы в нижнем регистре	
Категория	символьная	
Синтаксис	MessageResult=StringLower("Text");	
	Аргумент	Описание
	Text	Строка символов: литерал либо название тэга типа «сообщение».
Примечание	Символы в нижнем регистре, специальные символы и цифры остаются без изменений.	
Примеры	StringLower("TURBINE") возвращает "turbine" StringLower("22.2 Is The Value") возвращает "22.2 is the value"	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringMid()

	Возвращает указанное число символов из указанной строки, начиная с указанной позиции. Данная функция отличается от аналогичных функций StringLeft() и StringRight() тем, что позволяет определять начальную и конечную позиции извлекаемой части исходной строки	
Категория	символьная	
Синтаксис	MessageResult=StringMid("Text", StartChar, Chars);	
	Аргумент	Описание
	Text	Исходная строка символов: литерал либо название тэга типа «сообщение».
	StartChar	Начальная позиция извлекаемой подстроки в исходной строке: число или название тэга целого типа.
	Chars	Количество извлекаемых символов: число или название тэга целого типа.
Примеры	StringMid("The Furnace is Overheating", 5, 7) возвращает "Furnace" StringMid("The Furnace is Overheating", 13, 3) возвращает "is " StringMid("The Furnace is Overheating", 16, 50) возвращает "Overheating"	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringReplace()

	Изменяет указанные части исходной строки. С помощью этой функции можно изменять отдельные символы, слова и даже целые фразы	
Категория	символьная	
Синтаксис	MessageResult=StringReplace(Text, SearchFor, ReplaceWith, CaseSens, NumToReplace, MatchWholeWords);	
	Аргумент	Описание
	Text	Исходная строка: литерал или название тэга типа «сообщение».
	SearchFor	Искомая подстрока: литерал или название тэга типа «сообщение».
	ReplaceWith	Подстрока замены: литерал или название тэга типа «сообщение».
	CaseSens	Данный аргумент определяет, должен ли поиск осуществляться с учётом регистра символов в подстроке (1) или нет (0): число или название тэга целого типа.
	NumToReplace	Данный аргумент определяет, сколько раз искомая подстрока должна в исходной строке заменяться на подстроку замены (-1 означает замену всех вхождений): число или название тэга целого типа.
	MatchWholeWords	Данный аргумент указывает, должна ли функция заменять только слова целиком (0 = нет, 1 = да): число или название тэга целого типа. Если значение аргумента MatchWholeWords равно 1, а значение аргумента SearchFor – “and”, то эти символы в строке “handle” заменяться не будут. Если же аргумент MatchWholeWords будет равен 0, то найденные символы будут заменяться на указанные независимо от местоположения.
Примеры	<p>StringReplace("In From Within", "In", "Out", 0, 1, 0) возвращает "Out From Within" (изменено только одно вхождение)</p> <p>StringReplace("In From Within", "In", "Out", 0, -1, 0) возвращает "Out From without" (изменены все вхождения)</p> <p>StringReplace("In From Within", "In", "Out", 1, -1, 0) возвращает "Out From Within" (изменены все вхождения, совпадающие с учётом регистра)</p> <p>StringReplace("In From Within", "In", "Out", 0, -1, 1) возвращает "Out From Within" (изменены все вхождения, представляющие собой отдельные слова)</p> <p>Функция StringReplace() не распознает специальные символы типа @#\$%&*(). Указанные символы воспринимаются как ограничители. В частности, при выполнении функции StringReplace("abc#", "abc#", "1234", 0, 1, 1) никакие замены выполняться не будут. Знак “#” будет воспринят как ограничитель, а не символ.</p>	

См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()
-----------	--

StringRight()

	Возвращает указанное число символов строки Text , начиная с последней позиции	
Категория	символьная	
Синтаксис	MessageResult=StringRight("Text", Chars);	
	Аргумент	Описание
	Text	Исходная строка символов: литерал или название тэга типа «сообщение».
	Chars	Число возвращаемых символов: число или название тэга целого типа.
Примечание	Если значение аргумента Chars равно 0, то возвращается вся исходная строка.	
Примеры	StringRight("The Pump is On", 2) возвращает "On" StringRight("The Pump is On", 5) возвращает "is On" StringRight("The Pump is On", 87) возвращает "The Pump is On" StringRight("The Pump is On", 0) возвращает "The Pump is On"	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringSpace()

	Формирует строку пробелов внутри тэга типа «сообщения» или выражения	
Категория	символьная	
Синтаксис	MessageResult=StringSpace(NumSpaces);	
	Аргумент	Описание
	NumSpaces	Количество пробелов: число либо название тэга целого типа.
Примечание	Функция StringSpace() возвращает строку пробелов, длина которой задана аргументом NumSpaces .	
Примеры	В следующем примере символы пробела изображены символами "□": StringSpace(4) возвратит "□□□□" "Pump" + StringSpace(1) + "Station" возвратит "Pump□Station" .	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringTest()

	Проверяет, соответствует ли первый символ строки указанному типу		
Категория	символьная		
Синтаксис	DiscreteResult=StringTest("Text", TestType)		
	Аргумент	Описание	
	Text	Исходная строка символов: литерал либо название тэга типа «сообщение».	
	TestType	Проверяемый тип символа:	
		1	алфавитно-цифровой ('A'-'Z', 'a'-'z' и '0'-'9')
		2	цифровой ('0'-'9')
		3	алфавитный ('A'-'Z', 'a'-'z')
		4	алфавитный в верхнем регистре ('A'-'Z')
		5	алфавитный в нижнем регистре ('a'-'z')
		6	знак пунктуации (0x21-0x2F)
		7	символ ASCII (0x00-0x7F)
		8	шестнадцатеричный ('A'-'F', 'a'-'f' и '0'-'9')
		9	печатаемый символ (0x20-0x7E)
	10	управляющий символ (0x00-0x1F, 0x7F)	
	11	непечатаемый символ (0x09-0x0D и 0x20)	
Примечание	<p>Функция StringTest() возвращает положительное число в тэг DiscreteResult, если первый символ в строке Text соответствует типу, указываемому аргументом TestType.</p> <p>Как и в других функциях, которые обрабатывают только один символ, если строка содержит более одного символа, проверен будет только первый.</p>		
Примеры	<p>StringTest("ACB123", 1) возвращает 1</p> <p>StringTest("ABC123", 5) возвращает 0</p>		
См. также	<p>StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()</p>		

StringToIntg()

	Преобразует символьное представление числа в тэге типа «сообщение» в целое значение, которое можно использовать в математических расчётах	
Категория	символьная	
Синтаксис	IntegerResult=StringToIntg("Text");	
	Аргумент	Описание
	Text	Исходная строка символов: литерал либо название тэга типа «сообщение».
Примечание	При выполнении данной функции сначала проверяется, является ли первый символ строки цифрой. Если первый символ отличен от цифры (пробелы игнорируются), то возвращаемым значением будет 0. Если первый символ – цифра, выполняется проверка следующего символа, и так далее до первого нецифрового символа.	
Примеры	Если Text="ABCD" , то тэг IntegerResult будет равен 0 . Если Text="22.2 is the Value" , то тэг IntegerResult будет равен 22 (так как первые символы – цифры). Если Text="The Value is 22" , то тэг IntegerResult будет равен 0 .	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToReal(), StringTrim(), StringUpper(), Text()	

StringToReal()

	Преобразует символьное представление числа в тэге типа «сообщение» в вещественное значение (с плавающей запятой), которое можно использовать в математических расчётах	
Категория	символьная	
Синтаксис	RealResult=StringToReal("Text");	
	Аргумент	Описание
	Text	Исходная строка символов: литерал либо название тэга типа «сообщение».
Примечание	При выполнении данной функции сначала проверяется, является ли первый символ строки цифрой. Если первый символ отличен от цифры (пробелы игнорируются), то возвращаемым значением будет 0. Если первый символ – цифра, выполняется проверка следующего символа, и так далее до первого нецифрового символа.	
Примеры	Если Text="ABCD" , то тэг RealResult будет равен 0 . Если Text="22.261 is the Value" , то тэг RealResult будет равен 22.261 . Если Text="The Value is 22" , то тэг RealResult будет равен 0 .	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringTrim(), StringUpper(), Text()	

StringTrim()

	Удаляет лишние пробелы из строки	
Категория	символьная	
Синтаксис	MessageResult=StringTrim("Text", TrimType);	
	Аргумент	Описание
	Text	Исходная строка символов: литерал либо название тэга типа «сообщение».
	TrimType	Вид удаления:
	1	начальных пробелов (до первого символа, отличного от пробела)
	2	конечных пробелов (после последнего символа, отличного от пробела)
	3	всех пробелов за исключением единичных пробелов между словами.
Примечание	Функция ищет и удаляет в строке Text символы, аналогичные пробелам (ASCII-коды 0x09-0x0D и 0x20). Вид удаления определяется значением аргумента TrimType .	
Примеры	<p>Пробелы в следующих примерах представлены символами "□".</p> <p>StringTrim("□□□□This□is□a□□test□□□□", 1) возвращает "This□is□a□□test□□□□"</p> <p>StringTrim("□□□□This□is□a□□test□□□□", 2) возвращает "□□□□This□is□a□□test"</p> <p>StringTrim("□□□□This□is□a□□test□□□□", 3) возвращает "This□is□a□□test"</p> <p>Удалить ВСЕ пробелы из строки можно с помощью функции StringReplace(). Для этого символ пробела нужно заменять на символ "null".</p>	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringUpper(), Text()	

StringUpper()

	Преобразует в указанной строке все символы в нижнем регистре в символы в верхнем регистре	
Категория	символьная	
Синтаксис	MessageResult=StringUpper("Text");	
	Аргумент	Описание
	Text	Строка символов: литерал либо название тэга типа «сообщение».
Примечание	Символы в верхнем регистре, специальные символы и цифры остаются без изменений.	
Примеры	StringUpper("abcd") возвращает "ABCD" StringUpper("22.2 is the value") возвращает "22.2 IS THE VALUE"	
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), Text()	

Tan()

	Возвращает <i>тангенс</i> угла, заданного в градусах	
Категория	математическая	
Синтаксис	Result=Tan(AngleNumber);	
	Аргумент	Описание
	AngleNumber	Угол в градусах: число либо название тэга целого или вещественного типа.
Примеры	Wave = 10 + 50 * Tan(6 * \$Second); Tan(45) возвращает 1 Tan(0) возвращает 0	
См. также	Cos(), Sin(), ArcCos(), ArcSin(), ArcTan()	

Text()

	Записывает в тэг типа «сообщение» символьное представление числового значения аналогового тэга (целого или вещественного типа) в соответствии с указанным способом форматирования результата	
Категория	символьная	
Синтаксис	MessageResult=Text(Analog_Tag, "Format_Text");	
	Аргумент	Описание
	Analog_Tag	Преобразуемая величина: число либо название тэга целого или вещественного типа.
	Format_Text	Формат преобразования: литерал либо название тэга типа «сообщение».
Примеры	<p>MessageTag=Text(Analog_Tag, "#.00"); Здесь MessageTag – название тэга типа «сообщение», Analog_Tag – тэг целого либо вещественного типа, строка "#.00" указывает на способ представления полученного результата: Если тэг Analog_Tag имеет значение 66, то в тэг MessageTag записывается строка "66.00". Если тэг Analog_Tag имеет значение 22.269, то в тэг MessageTag записывается строка "22/27". Если тэг Analog_Tag имеет значение 9.999, то в тэг MessageTag записывается строка "10.00". LogMessage("Текущее значение FreezerRoomTemp:" + Text(FreezerRoomTemp, "#.#")); В следующем примере тэг MessageTag получит значение "One=1 Two=2". MessageTag = "One + " + Text(1,"#") + StringChar(32) + "Two + " + Text(2,"#");</p>	
См. также	StringFromIntg(), StringToIntg(), StringFromReal(), StringToReal()	

Trunc()

	Округляет вещественное число путём отбрасывания дробной части	
Категория	математическая	
Синтаксис	ResultNumericTag=Trunc(Number);	
	Аргумент	Описание
	Number	Любое число либо тэг целого или вещественного типа.
Примечание	Результат выполнения функции такой же, как и результат присвоения целому тэгу значения вещественного.	
Примеры	Trunc(4.3) возвращает 4 Trunc(-4.3) возвращает -4	
См. также	Round()	

TseGetClientId()

	Если в клиенте терминального сервиса выполняется приложение View, возвращает идентификатор клиента (его адрес TCP/IP), в противном случае возвращает пустую строку	
Категория	терминальный сервис	
Синтаксис	MessageResult=TseGetClientId();	
	Параметр	Описание
	MessageResult	Название тэга типа «сообщение», в который будет записано значение идентификатора клиента в символьном виде.
Примечание	Данный идентификатор, как правило, используется внутри системы для образования названий серверов SuiteLink и для регистрации названий файлов.	
Примеры	MsgTag=TseGetClientID(); Если IP-адрес клиента равен, например, 10.103.202.1, то в тэг MsgTag будет записана строка "10.103.202.1".	

TseGetClientNodeName()

	Возвращает имя узла клиента Windows, на котором выполняется сеанс.	
Категория	дополнительная	
Синтаксис	TseGetClientNodeName();	
	Параметр	Описание
Комментарии	Если клиент запущена, как консольный сеанс, он возвращает имя узла самого терминального сервера. Если клиент запущена, как тонкий клиент, имеющий имя, которое может распознать Windows, он возвращает это имя. Если клиент не имеет имени, он возвращает пустую строку.	
Пример(ы)		

TseQueryRunningOnClient()

	Возвращает ненулевое целое значение, если приложение Представления (View) выполняется на клиенте терминального сервера. В противном случае возвращается нулевое значение (0).	
Категория	Терминальные сервисы	
Синтаксис	Результат=TseQueryRunningOnClient();	
	Параметр	Описание
	Результат	Возвращает 0, если Представления (View) работает, не на клиенте TS.
Комментарии	Нет	
Пример	IntTag=TseQueryRunningOnClient(); Возвращает IntTag=1, если Представления (View) работает на клиенте Терминального сервера.	

TseQueryRunningOnConsole()

	Если в консоли терминального сервиса выполняется приложение View, возвращает ненулевое целое значение, в противном случае возвращает 0.	
Категория	терминальный сервис	
Синтаксис	Result=TseQueryRunningOnConsole();	
	Параметр	Описание
	Result	Тэг, в который записывается 0, если приложение View не выполняется в консоли терминального сервиса.
Примеры	IntTag=TseQueryRunningOnConsole(); Тэг IntTag принимает значение 1, если приложение View выполняется в текущий момент в консоли терминального сервиса.	

TseQueryRunningOnClient()

	Если в клиенте терминального сервиса выполняется приложение View, возвращает ненулевое целое значение, в противном случае возвращает 0.	
Категория	терминальный сервис	
Синтаксис	Result=TseQueryRunningOnClient();	
	Параметр	Описание
	Result	Тэг, в который записывается 0, если приложение View не выполняется в клиенте терминального сервиса.
Примеры	IntTag=TseQueryRunningOnClient(); Тэг IntTag принимает значение 1, если приложение View выполняется в текущий момент в клиенте терминального сервиса.	

UTCDateTime()

	Получает время UTC, текущую временную зонку и смещение GMT и статус перехода на летнее время.	
Категория	дополнительная	
Синтаксис	UTCDateTime(Format);	
	Параметр	Описание
	Format	Строковый параметр, который задаёт формат выхода. Допустимые значения см. ниже.
Комментарии	UTCDateTime() возвращает значение, которое должно соответствующим образом присвоено тэгу сообщения.	
Пример	<p>В 9:24 тихоокеанского времени будет выведено следующее:</p> <pre>UTCDateTime ("UTC_SHORT") даст 17:24:05 UTCDateTime ("UTC_LONG") даст 01/06/2003 17:24:05 UTCDateTime ("UTC_LOCAL") даст тихоокеанское стандартное время -8:0: 1</pre> <p>Если в качестве аргумента будет переданная любой недопустимая (отличная от UTC_SHORT, UTC_LONG, UTC_LOCAL) строка, возвращаемым значением будет значение по умолчанию (ddd mmm dd hh:mm:ss yyyy)</p> <p>Для вышеупомянутой даты и времени (9:24 6 января 2003 года) возвращаемым значением было бы "Mon Jan 06 17:24:05 2003".</p>	

wcAddItem()

	Добавляет указанную строку в простой (List box) или развёртывающийся (Combo box) список с полем ввода. Если список не предусматривает упорядочивания, строка добавляется в его конец. В противном случае строка вставляется внутрь списка, после чего выполняется его сортировка	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcAddItem("ControlName", "MessageTag");	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	MessageTag	Вставляемая в список строка: литерал либо название тэга типа «сообщение».
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Объекты применения	простые и развёртывающиеся списки	
Примеры	Следующие операторы (в оконном скрипте типа On Show) добавляют указанные строки в список ListBox_1 при открытии содержащего его окна: wcAddItem("ListBox_1", "Chocolate"); wcAddItem("ListBox_1", "Vanilla"); wcAddItem("ListBox_1", "Strawberry");	
См. также	wcInsertItem()	

wcClear()

	Удаляет все элементы простого или развёртывающегося списка	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcClear("ControlName");	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Объекты применения	простые и развёртывающиеся списки	
Примеры	Следующий оператор удаляет все элементы развёртывающегося списка ListBox_1 при выполнении кнопочного скрипта: wcClear("ListBox_1");	

wcDeleteItem()

	Удаляет элемент простого или развёртывающегося списка с указанным порядковым номером	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcDeleteItem("ControlName", ItemIndex);	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	ItemIndex	Порядковый номер (индекс) элемента списка: число либо название тэга целого типа.
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Объекты применения	простые и развёртывающиеся списки	
Примеры	При выполнении кнопочного скрипта следующий оператор удаляет из развёртывающегося списка ListBox_1 элемент с номером 3: wcDeleteItem("ListBox_1", 3);	

wcDeleteSelection()

	Удаляет из списка пункт, выбранный в текущий момент	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcDeleteSelection("ControlName");	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Объекты применения	простые и развёртывающиеся списки	
Примеры	При выполнении кнопочного скрипта следующий оператор удаляет из развёртывающегося списка Combobox_1 выбранный в текущий момент элемент: wcDeleteSelection ("Combobox_1");	

wcErrorMessage()

	Возвращает текст сообщения об ошибке по указанному номеру	
Категория	управление окнами	
Синтаксис	ErrorMessage=wcErrorMessage(ErrorNumber);	
	Аргумент	Описание
	ErrorMessage	Название тэга типа «сообщение».
	ErrorNumber	Код ошибки, который возвращают все функции категории управления окнами: число либо название тэга целого типа.
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Объекты применения	простые и развёртываемые списки, текстовые поля, окошки выбора и переключатели.	
Примеры	<p>При возникновении ошибки во время загрузки списка её текстовое описание можно получить в тэге ErrorDescription. Показать значение этого тэга на экране можно с помощью связанной с ним анимационной функции отображения символьных значений типа String Value Output.</p> <p>Кнопочный скрипт типа On Show (При показе):</p> <pre>ErrorNumber=wcLoadList("ListBox_1", "c:\InTouch\recipe.txt"); ErrorDescription=wcErrorMessage(errornumber);</pre> <p>Данная функция может быть также использована с любыми элементами графических окон для вывода сообщений об ошибках:</p> <pre>ErrorNumber=wcAddItem("ListBox_1", "United States"); ErrorMsg=wcErrorMessage(ErrorNumber);</pre>	

wcFindItem()

	Возвращает соответствующий индекс первого элемента простого или развёртывающегося списка, значение которого совпадает с указанной строкой	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcFindItem("ControlName", "MessageTag", DiscreteTag, Tagname);	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	MessageTag	Строка сравнения: литерал либо название тэга типа «сообщение».
	DiscreteTag	Способ сравнения строк: 0 = без учёта регистра, 1 -= с учётом регистра
	Tagname	Название тэга целого типа
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Объекты применения	простые и развёртывающиеся списки	

wcGetItem()

	Возвращает значение элемента списка с указанным порядковым номером	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcGetItem("ControlName", ItemIndex, Tagname);	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	ItemIndex	Индекс элемента списка: число либо название тэга целого типа.
	Tagname	Название тэга целого либо вещественного типа. При выполнении функции wcGetItem() в него будет записано числовое значение, соответствующее данному элементу списка.
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Объекты применения	простые и развёртываемые списки	
Примеры	<p>При выполнении кнопочного скрипта следующий оператор возвратит в тэг ListSelection типа «сообщение» значение десятого элемента развёртываемого списка Combobox_1:</p> <p>wcGetItem("Combobox_1", 10, ListSelection);</p> <p>Если в списке под десятым номером стоит Vanilla, то эта строка и будет записана в тэг ListSelection.</p>	

wcGetItemData()

	Возвращает целое значение, соответствующее элементу списка с порядковым номером ItemIndex	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcGetItemData("ControlName", ItemIndex, Tagname);	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	ItemIndex	Индекс элемента списка: число либо название тэга целого типа.
	Tagname	Название тэга целого либо вещественного типа. При выполнении функции wcGetItem() в него будет записано числовое значение, соответствующее данному элементу списка.
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Объекты применения	простые и развёртываемые списки	
Примеры	<p>При выполнении кнопочного скрипта следующий оператор возвратит в тэг ItemValue целого типа числовое значение, связанное с пятым элементом простого списка ListBox_1:</p> <p>wcGetItemData("ListBox_1", 5, ItemValue);</p> <p>Если пятому элементу списка было сопоставлено целое число 4500, то это число и будет записано в тэг ItemValue.</p>	
См. также	wcSetItemData()	

wcInsertItem()

	Вставляет строку Message в список. Аргумент ItemIndex задаёт позицию в списке, в которую будет вставлена указанная строка. В отличие от функции wcAddItem() , функция wcInsertItem() не упорядочивает список, даже если он был создан как сортируемый	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcInsertItem("ControlName", ItemIndex, "MessageTag");	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	ItemIndex	Номер позиции добавляемого элемента списка: число либо название тэга целого типа. Если данный аргумент имеет значение -1, то элемент добавляется в конец списка.
	MessageTag	Строка, вставляемая в список в указанной позиции: литерал либо название тэга типа «сообщение».
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Объекты применения	простые и развёртываемые списки	
Примеры	При выполнении кнопочного скрипта следующий оператор вставит строку "Blueberry" в четвёртую позицию сверху списка ListBox_1 : wcInsertItem("ListBox_1", 4, "Blueberry");	
См. также	wcAddItem()	

wcLoadList()

	Заменяет содержимое простого или развёртывающегося списка строками, содержащимися в указанном файле	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcLoadList("ControlName", "Filename");	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	Filename	Название файла с данными в ASCII-формате. Если каталог не указан, функция ищет указанный файл в текущем каталоге приложения. Значение аргумента – литерал либо название тэга типа «сообщение».
Примечание	Перечень кодов возможных ошибок можно найти в Приложении А.	
Объекты применения	простые и развёртывающиеся списки	
Примеры	При выполнении оконного скрипта типа On Show (При показе) следующий оператор загружает в список Combobox_1 данные, хранящиеся в отформатированном соответствующим образом файле c:\InTouch.32\wclist.txt : wcLoadList("Combobox_1", "c:\InTouch.32\wclist.txt");	
См. также	wcAddItem(), wcSaveList()	

wcLoadText()

	Заменяет содержимое текстового объекта строками из указанного файла	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcLoadText("ControlName", "Filename");	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	Filename	Название файла с данными в ASCII-формате. Если каталог не указан, функция ищет указанный файл в текущем каталоге приложения. Значение аргумента – литерал либо название тэга типа «сообщение».
Примечание	<p>Поддерживаются файлы только в формате ASCII (типа создаваемых программой Notepad операционной системы Windows).</p> <p>Перечень кодов возможных ошибок можно найти в Приложении А.</p> <p>Расширенные возможности просмотра файлов доступны только при использовании программы просмотра документов из пакета <i>Productivity Pack</i>, входящего в систему Factory Suite.</p>	
Объекты применения	текстовые объекты	
Примеры	<p>Следующий оператор оконного скрипта типа On Show (При показе) загружает в текстовый объект Textbox_1 содержимое файла c:\InTouch.32\readme.txt, созданного программой Notepad операционной системы Windows:</p> <pre>wcLoadText("Textbox_1", "c:\InTouch.32\readme.txt");</pre>	

wcSaveList()

	Заменяет содержимое указанного файла элементами простого или развёртываемого списка	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcSaveList("ControlName", "Filename");	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	Filename	Название файла с данными в ASCII-формате. Если файл не существует, он создаётся. Впоследствии сохранённые элементы могут быть вновь загружены в список с помощью функции wcLoadList() . Значение аргумента – литерал либо название тэга типа «сообщение».
Примечание	Перечень кодов возможных ошибок можно найти в приложении А.	
Объекты применения	простые и развёртываемые списки	
Примеры	<p>Следующий оператор кнопочного скрипта сохраняет текущие элементы списка ListBox_1 в файле c:\InTouch.32\newlist.txt при нажатии на экранную кнопку:</p> <pre>wcSaveList("ListBox_1", "c:\InTouch.32\newlist.txt");</pre> <p>Если для заполнения списков используются ASCII-файлы, то содержащаяся в них информация должна быть оформлена таким образом:</p> <p>Название_объекта, Число_элементов Элемент_списка, Связанное_значение Элемент_списка, Связанное_значение ... Элемент_списка, Связанное_значение</p> <p>Пример: предположим, что содержимое файла должно быть загружено в развёртываемый список. В файле описано три элемента списка, для которых не будет определено никаких соответствующих связанных значений (более подробные сведения о связанных значениях можно найти в описании функции wcSetItemData()). Записи файла должны выглядеть следующим образом:</p> <pre>COMBOBOX, 3 Chocolate, 0 Vanilla, 0 Strawberry, 0</pre> <p>Здесь COMBOBOX – название списка. Число_элементов равно трём (3), поскольку в списке будут три элемента: Chocolate, Vanilla и Strawberry. Элемент Chocolate указан первым в списке и потому имеет порядковый номер 1. Элемент Vanilla имеет</p>	

	порядковый номер 2, а элемент Strawberry – порядковый номер 3. Связанное_значение каждого из этих элементов равно 0.
См. также	wcLoadList(), wcSetItemData()

wcSaveText()

	Сохраняет содержимое текстового объекта в указанном файле. Если файл не существует, он создаётся. Если он существует, он должен иметь атрибут “для чтения/записи”	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcSaveText("ControlName", "Filename");	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	Filename	Название файла в ASCII-формате. Если в названии файла путь доступа не указан, данные будут сохраняться в каталоге приложения. Если файл не существует, он создаётся. Впоследствии сохранённые элементы могут быть вновь загружены в список с помощью функции wcLoadList() . Значение аргумента – литерал либо название тэга типа «сообщение».
Примечание		
Объекты применения	текстовые объекты	
Примеры	Следующий оператор кнопочного скрипта сохраняет текущее содержимое текстового объекта Textbox_1 в файле c:\InTouch.32\newtext.txt при нажатии на экранную кнопку: wcSaveText("Textbox_1", "c:\InTouch.32\newtext.txt"); Поддерживаются файлы только в формате ASCII (типа создаваемых программой Notepad операционной системы Windows).	
См. также	wcLoadList()	

wcSetItemData()

	Связывает указанное целое значение с элементом списка с указанным порядковым номером. Данная функция позволяет связывать числа со строками символов	
Категория	управление окнами	
Синтаксис	[ErrorNumber=]wcSetItemData("ControlName", ItemIndex, Number);	
	Аргумент	Описание
	ControlName	Название объекта в окне, например ListBox_1 : литерал либо название тэга типа «сообщение».
	ItemIndex	Порядковый номер элемента в списке: число либо название тэга целого типа.
	Number	Связываемая с этим элементом списка целая величина: число либо название тэга целого типа.
Примечание	Элементы списков могут быть созданы во внешней программе (например, в программе Notepad) и загружены в список за один вызов функции. Они должны быть отформатированы так, как показано в описании функции wcSaveList(). Перечень кодов возможных ошибок можно найти в приложении А.	
Примеры	<p>Дан рецепт с тремя ингредиентами: мука, сахар и соль. Масса муки в рецепте 4500 г, сахара 1500 г и соли 325 г. Указанные значения сопоставляются с каждым компонентом рецепта (т.е. элементом списка) с помощью скрипта по изменению данных, который запускается при выборе нового рецепта (при изменении значения тэга названия рецепта):</p> <p>wcSetItemData("ListBox_1", 1, 4500); {устанавливает значение для первого элемента (мука) – 4500}</p> <p>wcSetItemData("ListBox_1", 2, 1500); {устанавливает значение для второго элемента (сахар) – 1500}</p> <p>wcSetItemData("ListBox_1", 3, 325); {устанавливает значение для третьего элемента (соль) – 325}</p> <p>Для определения связанного с элементом списка значения используется функция wcGetItemData(), аргумент TagName которой содержит требуемое число. Этот аргумент может быть тэгом целого типа для ввода/вывода, при помощи которого осуществляется передача параметров во внешнее устройство.</p>	
См. также	wcLoadList(), wcSaveList(), wcGetItemData()	

WindowState()

	Возвращает для указанного окна его статус.	
Категория	дополнительная	
Синтаксис	<code>IntegerResult = WindowState(WindowName);</code>	
	Параметр	Описание
	WindowName	Имя окна, статус которого должен быть определён. Допускается строка или тэг сообщения.
Комментарии	Возвращается 0, 1 или 2 и присваивается IntegerTag.	
Пример(ы)	<p><code>WindowState("Win1")</code> возвратит 1, где "Win1" является именем определённого и открытого в настоящий момент окна.</p> <p><code>WindowState("Win2")</code> возвратит 0, где "Win2" является именем определённого, но не открытого в настоящий момент окна.</p> <p><code>WindowState("XXXX")</code> возвратит 2, где "XXXX" является именем окна, которое не определено.</p>	
См. также	OpenWindowsList()	

WWControl()

	Используется для восстановления, минимизации, максимизации и закрытия окна внешней программы из приложения InTouch	
Категория	прочие	
Синтаксис	WWControl("AppTitle", "ControlType");	
	Аргумент	Описание
	AppTitle	Название внешнего приложения. Его можно также определить с помощью функции InfoAppTitle() . Допустимые значения аргумента – литерал либо название тэга типа «сообщение».
	ControlType	Способ управления приложением (аналогично выбору соответствующей команды в управляющем меню этого приложения): литерал либо название тэга типа «сообщение».
		"Restore" Активизировать и показать окно приложения.
		"Minimize" Активизировать окно приложения и свернуть его в пиктограмму.
		"Maximize" Активизировать окно приложения и развернуть его.
		"Close" Завершить исполнение приложения.
Примеры	WWControl("Calculator","Restore"); WWControl(InfoAppTitle("View"), "Close");	
См. также	InFoAppTitle(), ActivateApp(), StartApp()	

WWExecute()

	Посылает команду (по протоколу DDE) указанному приложению и группе данных	
Категория	WWDDE	
Синтаксис	[Status=]WWExecute("Application", "Topic", "Command");	
	Аргумент	Описание
	Application	Название приложения, которое должно исполнить команду: литерал либо название тэга типа «сообщение».
	Topic	Название группы данных в приложении, которому посылается команда: литерал либо название тэга типа «сообщение».
	Command	Команда: литерал либо название тэга типа «сообщение».
Примечание	Указанная в аргументе Command команды посылается указанному приложению и группе данных.	
Примеры	<p>Следующий скрипт посылает приложению excel команду исполнения макроса.</p> <pre>Macro="Macro1!TestMacro"; Command="[Run(" + StringChar(34) + Macro + StringChar(34) + ",0)]"; WWExecute("excel", "system", Command);</pre> <p>При выполнении функции WWExecute("excel", "system", Command) приложению excel будет послана следующая команда:</p> <pre>[Run("Macro1!TestMacro")]</pre> <p>Следующий оператор запускает макрос в приложении Microsoft Access:</p> <pre>WWExecute("MSAccess", "system", "MyMacro");</pre> <p>Функция WWExecute() возвращает 1, если указанное приложение исполняется, указанная группа данных в нём существует и команда была послана успешно. Она возвращает 0, если приложение занято, и -1 при возникновении ошибки. Таким образом можно контролировать выполнение команды:</p> <pre>Status=WWExecute("excel", "system", Command);</pre> <p>Status – тэг целого типа, в который записываются значения 1, -1 или 0.</p>	
См. также		

WWIsDayLightSaving()

	Возвращает из InTouch режим перехода на летнее время.
Категория	дополнительная
Синтаксис	WWIsDayLightSaving();
Remarks	Возвращает 1, если переход на летнее время (Daylight Savings) разрешён, и 0, если переход на летнее время (Daylight Savings) запрещён.

WWMoveWindow()

	Перемещает указанное окно к координатам, которые вы задали.	
Категория	дополнительная	
Синтаксис	WWMoveWindow(WindowName, Left, Top, Width, Height);	
	Параметр	Описание
	WindowName	Имя перемещаемого окна.
	Left	Желаемая левая координата окна.
	Top	Желаемая верхняя координата окна.
	Width	Желаемая ширина окна.
	Height	Желаемая высота окна.

WWPoke()

	Записывает указанное значение (по протоколу DDE) в указанный элемент группы данных указанного приложения	
Категория	WWDDE	
Синтаксис	[Status=]WWPoke("Application", "Topic", "Item", "TextValue");	
	Аргумент	Описание
	Application	Название приложения, которому посылается команда РОКЕ: литерал либо название тэга типа «сообщение».
	Topic	Название группы данных в приложении: литерал либо название тэга типа «сообщение».
	Item	Название элемента в группе данных: литерал либо название тэга типа «сообщение».
	TextValue	Передаваемое значение: литерал либо название тэга типа «сообщение». Для передачи числового значения его сначала необходимо преобразовать в символьное представление с помощью функций Text() , StringFromIntg() или StringFromReal() .
Примечание	Значение аргумента TextValue записывается в указанный элемент указанной группы данных указанного приложения.	
Примеры	<p>Следующий оператор преобразует некоторое значение в строку символов и записывает его в ячейку таблицы Excel:</p> <p>String=Text(Value, "0"); WWPoke("excel", "sheet1", "r1c1", String);</p> <p>Более подробные сведения о взаимодействии систем Excel 5.0 и InTouch (по протоколу DDE) можно найти в <i>Руководстве пользователя InTouch</i>.</p> <p>Передача команды WWPoke() из одного приложения View в другое приложение View не определено и не поддерживается. Успешное выполнение функции WWPoke() в этом случае не гарантируется и может завершиться без каких-либо результатов</p> <p>Функция WWPoke() возвращает 1, если указанное приложение исполнено, группа данных и элемент в ней существуют и указанное значение было послано успешно. Она возвращает 0, если приложение занято, и -1 при возникновении ошибки. Таким образом можно контролировать выполнения команды:</p> <p>Status=WWPoke("excel", "sheet1", "r1c1", String); Status – это тэг целого типа, в который записываются значения 1, -1 или 0.</p>	
См. также	Text(), StringFromIntg(), StringFromReal()	

WWRequest()

	Выдаёт разовый запрос на получение значения (по протоколу DDE) из указанного приложения, группы данных и элемента	
Категория	WWDDE	
Синтаксис	WWRequest(Application, Topic, Item, ValueMsg_Tag);	
	Аргумент	Описание
	Application	Название приложения, из которого запрашиваются данные: литерал либо название тэга типа «сообщение».
	Topic	Название группы данных в приложении: литерал либо название тэга типа «сообщение».
	Item	Название элемента в группе данных: литерал либо название тэга типа «сообщение».
	ValueMsg_Tag	Название тэга типа «сообщение», которое должно быть заключено в кавычки. Запрошенные данные будут записаны в указанный тэг. Допустимые значения этого аргумента – литерал либо название тэга типа «сообщение».
Примечание	При выполнении этой функции DDE-значение указанного элемента в указанной группе данных указанного приложения будет записано в тэг, определяемый аргументом ValueMsg_Tag . Значение возвращается в виде строки символов. Если оно является символьным представлением числа, то требуемое числовое значение можно получить с помощью функций StringToIntg() и StringToReal() .	
Примеры	<p>Следующий оператор запрашивает значение из указанной ячейки таблицы Excel и преобразует полученную строку символов в число:</p> <pre>WWRequest("excel", "[Book1.xls]sheet1", "r1c1", Result);</pre> <p>Value=StringToReal(Result);</p> <p>Функция WWRequest() возвращает 1, если указанное приложение исполнено, группа данных и элемент в ней существуют и указанное значение было послано успешно. Она возвращает 0, если приложение занято, и -1 при возникновении ошибки. Таким образом можно контролировать выполнения команды:</p> <pre>Status=WWRequest("excel", "[Book1.xls]sheet1", "r1c1", Result);</pre> <p>Status – это тэг целого типа, в который записываются значения 1, -1 или 0.</p>	
См. также	StringToIntg(), StringToReal()	

WWStringFromTime()

	Преобразует заданную метку времени, заданную в локальном времени, в UTC-время (Coordinated Universal Time – универсальное координатное время) и выводит результат в строку. Эта функция принимает метку времени, выраженную количеством секунд, прошедших с полночи 1 января 1970 года, в строку, выражающую эту метку времени как дату, время и день недели в соответствии с выбранным методом представления.		
Категория	строка		
Синтаксис	MessageResult = wwStringFromTime (SecsSince1-1-70, StringType) ;		
	Параметр	Описание	
	<i>SecsSince1-1-70</i>	Целый тип. Количество секунд, прошедших с 1 января 00:00:00 1970 года.	
	<i>StringType</i>	Задаёт метод представления.	
		Тип	Описание
		1	Показывает дату в том же формате, что и установлен в Панели управления Windows. (Аналогично представлению для \$DateString.)
		2	Показывает время в том же формате, что и установлен в Панели управления Windows. (Аналогично представлению для \$DateString.)
		3	Показывает 24-символьную строку, представляющую как дату, так и время: "Wed Jan 02 02:03:55 1993" (среда, 2 января, 2ч.3мин.55 с 1993 г.)
		4	Показывает краткую форму дня недели: "Wed" (среда).
		5	Показывает длинную форму дня недели: "Wednesday" (среда).
Пример(ы)	<p>Этот пример предполагает, что временная зона локального узла является тихоокеанским стандартным временем (Pacific Standard Time – UTC-0800). Локальное время, передаваемое в функцию, составляет 04:00:00 в пятницу 01.01.1970. Так как PST на 8 часов позже UTC, функция возвратит следующие результаты:</p> <p>wwStringFromTime (57600, 1) возвратит "1/2/70"</p> <p>wwStringFromTime (57600, 2) возвратит "12:00:00 AM"</p> <p>wwStringFromTime (57600, 3) возвратит "Fri Jan 02 00:00:00 1970"</p> <p>wwStringFromTime (57600, 4) возвратит "Fri"</p> <p>wwStringFromTime (57600, 5) возвратит "Friday"</p>		

Комментарии	Любые настройки, связанные с переходом на летнее время (Daylight Savings Time – DST), будут автоматически применяться к возвращаемому результату. Следовательно, нет необходимости вносить во входное значение какие-либо ручные поправки, связанные с DST.
См. также	StringASCII(), StringChar(), StringFromIntg(), StringFromReal(), StringFromTime(), StringFromTimeLocal(), StringInString(), StringLeft(), StringLen(), StringLower(), StringMid(), StringReplace(), StringRight(), StringSpace(), StringTest(), StringToIntg(), StringToReal(), StringTrim(), StringUpper(), Text()

Г Л А В А 4

OLE-автоматизация и система InTouch

Разработанная в компании Microsoft технология OLE (Object Linking and Embedding – Внедрение и связывание объектов) обеспечивает весьма широкий перечень информационно-управляющих операций между клиентом и сервером. Система InTouch может выступать в роли клиента по отношению ко многим из таких серверов. Любое приложение InTouch способно информировать сервер об изменении значений тех или иных тегов и тем самым запускать определённые в сервере программные процедуры дальнейшей обработки. Подобная поддержка позволяет значительно расширять имеющиеся функциональные возможности системы InTouch.

Настоящая глава представляет собой краткий обзор архитектуры OLE Automation (OLE-автоматизация). Описаны также способы использования процедур OLE-автоматизации средствами расширений InTouch. Поскольку приложение InTask является одним из серверов OLE-автоматизации, оно используется во многих приведённых в данной главе примерах.

Элементы OLE-автоматизации

В очень упрощённой форме, OLE – это стандартизованное средство обмена информацией между клиентом и сервером.

Средства объектно-ориентированного программирования обеспечиваются в этой клиент-серверной среде OLE-объектом автоматизации. Сервер предоставляет вычислительные возможности посредством своих классов-объектов. Клиент указывает способы использования этих возможностей путём создания объектов и доступа к его членам. OLE предоставляет канал связи между клиентом и сервером.

Для понимания роли объектов автоматизации необходимо сначала получить представление о принципах объектно-ориентированного программирования.

- **Объекты** являются адресуемыми и уникальными сущностями, которые могут быть созданы на основе **класса** объекта.
- Классы определяют соответствующий набор **свойств** (данные) и **методов** (функции). Таким образом, все объекты одного и того же класса имеют один и тот же набор свойств и методов, хотя значения свойств у каждого объекта могут быть разные.

Клиентские расширения OLE-автоматизации в InTouch

Расширения InTouch обеспечивают простой доступ к объектам автоматизации InTrack с помощью следующих трёх компонентов:

- Восемью дополнительными встроенными функциями InTouch:

OLE_CreateObject	OLE_IsObjectValid
OLE_GetLastObjectError	OLE_ReleaseObject
OLE_GetLastObjectErrorMessage	OLE_ResetObjectError
OLE_IncrementOnObjectError	OLE_ShowMessageOnObjectError

- Унарного оператора, означающего начало выражения OLE-автоматизации,
- OLE-браузера InTrack с шаблонами и справочной информацией обо всех классах объектов автоматизации InTrack. Окно OLE-браузера открывается при нажатии на кнопку OLE в правом нижнем углу окна редактора скриптов.

Ссылки на объекты OLE-автоматизации в InTouch

Все ссылки на объекты OLE-автоматизации в InTouch должны начинаться со знака процента (%), за которым указывается название объекта, выделяющее его среди всех объектов автоматизации. Это название должно начинаться с буквы и может состоять из букв, цифр и символа подчёркивания. Названия являются глобальными. Регистр символов в названии не учитывается. Примеры различных ссылок на объекты в InTouch:

- %ObjectName (аналогично %OBJECTNAME и %objectname)
- %Lot5
- %lot_5
- %aMachine
- %A_Machine
- %Sublot

Названия объектов OLE-автоматизации не являются названиями тегов InTouch. В приложении могут использоваться объекты OLE-автоматизации и теги InTouch с одинаковыми названиями (различаясь наличием символа процента в начале).

Ссылки на объекты

Начинающиеся на символ процента названия именуются здесь: **ссылками на объекты**, потому что фактически они являются указателями на объекты автоматизации. Два различных названия могут обозначать один и тот же объект в результате выполнения следующего оператора InTouch:

```
%Объект1 = %Объект2;
```

После выполнения этого оператора обе ссылки на объект будут "указывать" на один и тот же объект автоматизации.

Ссылки на объект автоматически разрешаются при первом же использовании во время исполнения рабочей программы. Они являются глобальными, то есть их значение можно определить в одном скрипте и использовать в другом.

Как и в большинстве объектно-ориентированных языков программирования, обращение к свойствам и методам объектов автоматизации осуществляется путём указания их через точку после названия объекта. В следующем примере показано, каким образом свойству объекта присваивается значение 5:

```
%Объект.Свойство = 5;
```

Вложенные объекты

Если один объект является членом другого объекта, для ссылки на него также может использоваться точка "." (произвольное число раз). Установка значения свойства вложенного объекта выполняется следующим образом:

```
%Объект.Вложенный_объект.Свойство = 6;
```

При использовании точки в подобных случаях необходимо соблюдать следующие правила:

1. В спецификациях объекта и вложенного объекта не должно быть никаких пробелов:
 - пробелы между символом процента и первым символом ссылки на объект недопустимы,
 - пробелы между ссылкой на объект, символом точки и названием вложенного объекта недопустимы,
 - пробелы между названием вложенного объекта, символом точки и названием свойства недопустимы.
2. Количество символов в полной спецификации – от символа процента до последнего символа спецификации (включительно) – не должно превышать 98.

Например, длина спецификации `%Объект.Свойство` равна 16, а спецификации `%Объект.Вложенный_объект.Свойство` – 33.

Преобразование типов данных между InTouch и OLE

Расширения OLE поддерживают не все типы данных, поддерживаемые InTouch. В следующей таблице приведены сведения о возможных преобразованиях типов данных. Следует иметь в виду, что преобразование осуществляется во время исполнения программы, а не в WindowMaker. Таким образом, преобразование данных может привести к неожиданным результатам без сообщений об ошибках.

Целый	32-разрядная величина со знаком. Может быть преобразована в следующие типы:	
	В вещественный	Значение является числом с плавающей запятой.

	В сообщении	Стандартное преобразование в символьное представление десятичного числа, с возможным знаком минуса перед числом.
	В логический	Если исходная величина ненулевая, преобразуется в TRUE (1), если нулевая – в FALSE (0).
Вещественный	64-разрядная величина. Система InTouch выполняет расчёты с 64-разрядной точностью, однако результат сохраняет в тегах с 32-разрядной точностью. Интерфейс объектов автоматизации способен обрабатывать числа с более высокой точностью. Этот тип может быть преобразован в следующие типы:	
	В целый	Дробная часть отбрасывается.
	В сообщении	Стандартное преобразование в символьное представление десятичного числа, с возможным знаком минуса перед числом.
	В логический	Если исходная величина ненулевая, преобразуется в TRUE (1), если нулевая – в FALSE (0).
Сообщение	Строка длиной до 131 символа (ограничение InTouch). Преобразование данных для каждого типа более сложное, являясь наиболее вероятным источником ошибок во время исполнения программы.	
	В целый	Аналогично функции InTouch StringToInt() . Если строка начинается с цифры (пробелы игнорируются, знак минуса учитывается), – она преобразуется, в противном случае возвращается нулевое значение.
	В вещественный	Аналогично функции InTouch StringToReal() . Если строка начинается с цифры (пробелы игнорируются, знак минуса учитывается), – она, в противном случае возвращается нулевое значение.
	В логический	Сначала осуществляется преобразование в целый тип, затем выполняется преобразование целого типа в логический.

	В ссылку на объект	Данное преобразование требуется только в нескольких специальных случаях, и его следует избегать. Ошибки преобразования могут привести к завершению работы WindowViewer с фатальной ошибкой. Сообщение об ошибке выдаётся, когда входные данные отличаются от строки, полученной в результате преобразования ссылки на объект.
Логический	Преобразуется в следующие типы:	
	В целый	1, если исходное значение TRUE, 0 – в противном случае.
	В вещественный	1, если исходное значение TRUE, 0 – в противном случае.
	В сообщение	"0", если исходное значение FALSE, "-1", если TRUE.
Ссылка на объект	Как правило, ссылки на объект не следует использовать в случаях, требующих преобразования типов. Может преобразовываться в следующие типы:	
	В сообщение	Строка в специальном формате с шестнадцатеричным представлением ссылки на объект.

Ограничения

На использование расширений объектов автоматизации InTouch накладываются следующие два ограничения:

- расширения могут использоваться только внутри скриптов InTouch;
- в расширениях используется "позднее связывание": большинство сообщений об ошибках появляется только после исполнения скрипта.

Функция создания объекта OLE_CreateObject()

Объекты автоматизации создаются с помощью внутренней функции InTouch OLE_CreateObject(). Синтаксис функции следующий:

OLE_CreateObject(%Объект, "Название_класса");

Здесь:

OLE_CreateObject	Название функции, создающей объект
%Объект	Название, используемое для ссылки на объект автоматизации
Название_класса	Название класса объектов в сервере OLE Server.

Возвращаемые значения отсутствуют.

Пример

Создать объект класса Sublot системы InTrack можно с помощью следующего оператора:

```
OLE_CreateObject(%Sublot, "InTrack.Sublot");
```

Неправильное написание названия класса объекта – одна из часто встречающихся ошибок, которую нельзя обнаружить средствами WindowMaker. Эта ошибка не определяется до тех пор, пока функция не будет запущена в среде WindowViewer. Во избежание данной ошибки рекомендуется пользоваться шаблонами OLE_CreateObject, которые встроены в OLE-браузер InTrack.

При первом создании объекта его свойства автоматически получают значения по умолчанию, определённые в сервере.

Примечание. Инициализировать ссылку на объект путём вызова функции OLE_CreateObject() не обязательно. В предыдущем примере связь ссылки на объект с самим объектом была создана с помощью оператора присваивания.

Функция управления объектами OLE_ReleaseObject()

Объект является действительным всё время, пока на него ссылается хотя бы один другой объект. Хорошей практикой программирования является освобождение объекта после его использования, для того чтобы обеспечить экономию системных ресурсов. Для освобождения объекта:

1. Создайте новый объект с той же самой ссылкой на объект.

При создании нового объекта автоматизации с тем же названием, что и в существующей ссылке на объект, исходный объект автоматизации освобождается, а новый объект связывается с исходным названием. Например:

```
OLE_CreateObject(%Object, "InTrack.Sublot");
OLE_CreateObject(%Object, "InTrack.DateTime");
```

В первом случае создаётся новый объект класса Sublot, ссылка на который осуществляется с помощью %Object. Второй оператор освобождает объект класса Sublot, создаёт новый объект класса DateTime и связывает его со ссылкой %Object.

2. Воспользуйтесь внутренней системной функцией **OLE_ReleaseObject()**. Её синтаксис:

```
OLE_ReleaseObject(%Название_объекта);
```

Здесь:

%Название_объекта Ссылка на объект автоматизации.

Функция не возвращает никаких значений и никогда не завершается с ошибкой.

3. Создайте ссылку на второй объект.

Синтаксис:

```
%Объект1 = %Объект2;
```

Здесь:

%Объект1 Ссылка на объект, которая будет "указывать" на тот же объект, что и %Объект2. Ранее указываемый этой ссылкой объект автоматизации будет освобождён.

%Объект2 Объект автоматизации, на который есть ссылка, с названием **%Объект2**. Должен быть того же самого типа.

В данном примере объект, связанный с названием **%Объект1** (если таковой существовал), будет освобождён.

Функция проверки действительности объекта **OLE_IsObjectValid()**

После отмены взаимосвязи ссылки на объект и объекта OLE-автоматизации с помощью функции **OLE_ReleaseObject()** эта ссылка становится "недействительной". Указание недействительной ссылки в каком-либо из выражений приведёт к появлению ошибки во время исполнения программы.

Проверка действительности ссылки на объект автоматизации в системе InTouch осуществляется с помощью внутренней функции **OLE_IsObjectValid()**. Её синтаксис:

OLE_IsObjectValid(%Название_объекта);

Здесь:

%Название_объекта Ссылка на объект автоматизации.

Функция возвращает логическое значение, отражающее действительность указанной ссылки на объект автоматизации в текущий момент времени.

Определение значения свойства

Свойство – это именованный атрибут объекта, определяющий некоторую его характеристику или поведение во время исполнения программы. Свойство объекта отделяется от ссылки на объект точкой. Ссылка на свойство объекта в скриптах InTouch указывается следующим образом:

%Объект.Свойство

Здесь:

%Объект Ссылка на объект автоматизации.

Свойство Атрибут объекта.

Если указанное свойство объекта само является объектом, то его свойство также отделяется точкой. Например:

%Объект.Свойство1.Свойство2

Здесь:

%Объект Ссылка на объект автоматизации,

Свойство1 Свойство объекта **%Объект**. Следует отметить, что свойство может представлять собой внедрённый объект или ссылку на другой объект. Системы InTouch и OLE поддерживают оба варианта.

Свойство2 Свойство вложенного объекта, значение которого определяется.

Наличие в записи нескольких разделительных точек означает многократное выполнение операций "определения свойства". В данном случае сначала

"определяется свойство" %Объект.Свойство1, которое, согласно стандартам OLE-автоматизации, возвращает ссылку на вложенный объект. Полученная ссылка используется для выполнения следующей операции "определения свойства" – в данном примере Свойства2. Данная запись функционально аналогична следующим двум операторам:

```
%Временный_объект = %Объект.Свойство1;
```

```
%Временный_объект.Свойство2;
```

Такая последовательность операций имеет большое значение по нескольким причинам:

1. Ошибки разрешения названий свойств могут возникать при обработке каждой точки в записи.
2. Многократное использование точек используется в самых разных случаях для упрощения кодирования, например определения или установки значений свойств, а также вызова методов.

Пример 1

Цель – увеличить значение тега InTouch на единицу, если текущий момент находится между 1:00 и 2:00 AM первого дня месяца. Следующий код иллюстрирует ранее описанные принципы:

```
OLE_CreateObject(%DateTime, "InTrack.DateTime");
```

```
%DateTime.SetLocalTime();
```

```
IF (%DateTime.Day == 1) AND (%DateTime.Hour > 1) AND  
(%DateTime.Hour < 2) THEN
```

```
Tag = Tag + 1;
```

```
ENDIF;
```

```
OLE_ReleaseObject(%DateTime);
```

Пример 2

В данном примере используется "многоточечная" запись ссылок. Назначение кода – записать в теги InTouch некоторые сведения о ресурсах вычислительной системы. Пример довольно условный, однако он иллюстрирует методы определения значений свойств различными способами.

Для простоты изложения предположим, что объект %Machine был создан как экземпляр класса InTrack.Machine и что соответствующая информация была загружена из базы данных InTrack (механизмы загрузки рассматриваются в следующих параграфах).

```
NumberUnitsTag = %Machine.Capacity.Quantity;
```

```
TypicalWeightTag = %Machine.Capacity.Quantity * 40.6;
```

```
ContrivedLoadingTag = Log(%Machine.Capacity.Quantity) + 1;
```

```
UnitsNameTag = %Machine.Capacity.Units;
```

Установка значения свойства

Значение свойства объекта может быть установлено тремя способами:

- с помощью оператора присваивания (рассматривается в настоящем параграфе),

- передачей ссылки на свойство в качестве выходного аргумента метода,
- вызовом метода для установки значений одного или нескольких свойств объекта за одну операцию (новые значения передаются как аргументы). Данная возможность определяется характеристиками соответствующего класса. Дополнительные сведения о методах можно найти в последующих параграфах.

Примечание. Ссылки на свойства объектов автоматизации не могут передаваться "по ссылке" (то есть как выходные аргументы) DLL-функциям, вызываемым в скриптах InTouch. Значение свойства объекта изменить с помощью DLL-функций нельзя.

Синтаксис установки значения свойства объекта с помощью оператора присваивания следующий:

`%Объект.Свойство = Значение;`

Здесь:

%Объект	Ссылка на объект автоматизации.
Свойство	Атрибут объекта.
Значение	Значение, которое должно иметь указанный атрибут объекта.

Как и в случае определения значения свойств, если свойство само является объектом, то его свойство указывается также после точки:

`%Объект.Свойство1.Свойство2 = Значение;`

Здесь:

%Объект	Ссылка на объект автоматизации.
Свойство1	Свойство объекта %Объект , которое само является объектом.
Свойство2	Свойство вложенного объекта, которому должно быть присвоено указанное значение.

Данная запись функционально аналогична следующей:

`%Временный_объект = %Объект.Свойство1;`

`%Временный_объект.Свойство2 = Значение;`

Пример

Первый оператор следующего примера создаёт объект автоматизации класса **Amount** с названием **%PrimaryAmt**. Второй и третий операторы устанавливают значения свойств **Quantity** и **Units** равными **100** и **each** соответственно:

`OLE_CreateObject(%PrimaryAmt, "InTrack.Amount");`

`%PrimaryAmt.Quantity = 100;`

`%PrimaryAmt.Units = "each";`

Вызов методов

Вызов метода объекта автоматизации – это обычный способ обращения клиента к серверу для выполнения какого-либо действия. Далее приведён синтаксис вызова методов объектов автоматизации:

Результат = %Объект.Метод(Аргумент1, Аргумент2, ...);

%Объект.Метод(Аргумент1, Аргумент2, ...);

Здесь:

Результат	Значение, возвращаемое методом.
%Объект	Ссылка на объект автоматизации.
Метод	Действие, которое должен выполнить объект автоматизации.
Параметры	Информация, передаваемая методу или получаемая от него. Количество аргументов зависит от вида вызываемого метода. Тип аргументов может быть "только для чтения" или "для чтения/записи".

Ограничения

На вызов методов накладываются следующие ограничения:

- методы могут вызываться как "подпрограммы-процедуры" (без возвращаемых значений). Если метод может возвращать значение, оно игнорируется;
- для использования возвращаемого значения его сначала необходимо присвоить какому-либо тегу. Вызов метода не может использоваться внутри других выражений;
- как правило, исполнение скрипта приостанавливается до завершения метода, если только метод не выполняет действия, требующие взаимодействия с пользователем посредством диалоговых окон;
- необходимо указывать требуемое количество аргументов. Пропуск аргументов не поддерживается серверами OLE-автоматизации.

Типы параметров

Существуют ограничения, определяемые типами параметров, передаваемых методам объектов автоматизации в приложениях InTouch.

Тип данных	Вид представления аргумента
Целый (Integer)	Константа, выражение, тег "только для чтения", тег "для чтения/записи", свойство OLE-автоматизации "для чтения/записи"
Вещественный (Real)	Константа, выражение, тег "только для чтения", тег "для чтения/записи", свойство OLE-автоматизации "для чтения/записи"
Сообщение (Message)	Константа, выражение, тег "только для чтения", тег "для чтения/записи", свойство OLE-автоматизации "для чтения/записи"
Логический	Константа, выражение, тег "только для чтения", тег

(Discrete)	"для чтения/записи", свойство OLE-автоматизации "для чтения/записи"
Объект (Object)	Ссылка на объект или свойство OLE-автоматизации "для чтения/записи"

Следует отметить, что некоторые методы являются "полиморфными" и могут принимать данные разных типов на месте одного и того же аргумента. Эта возможность поддерживается в InTouch только для входных аргументов.

Выходные аргументы

В системе InTouch нет различий между аргументами "только для ввода" и аргументами "для ввода и вывода". Если аргумент может быть изменён, в него записывается значение.

Аргумент может быть изменён, если аргумент представляет собой:

- название тега InTouch, допускающего чтение и установку значений (для чтения/записи);
- ссылку на объект;
- свойство объекта.

В следующем примере значение первых трёх аргументов будет обновлено, поскольку каждый из них соответствует одному из указанных ранее критериев. Значение остальных аргументов не изменится.

```
%Obj1.Method(tag1, %obj2, %obj3.property1.property2, tag1 + 3,  
%obj4.property + 0);
```

Возвращаемые значения

Для получения возвращаемого методом значения слева от оператора присваивания должно стоять:

- название тега InTouch "для чтения/записи";
- ссылка на объект;
- спецификация свойства объекта.

Пример 1

Предположим, что существует объект класса **SerialNumbers** с названием **%SerialNumbers**. Следующий оператор добавит серийный номер 1234 к имеющимся серийным номерам в накопительном объекте **%SerialNumbers**:

```
returnCode = %SerialNumbers.Add("1234");
```

Пример 2

Предположим, что существует объект класса **Sublot** с названием **%Sublot_Object**. Выполнить метод объекта **Ship** с тремя аргументами (два из которых представляют собой объекты автоматизации **%PrimaryAmt** и **%SecondaryAmt**) можно следующим образом:

```
ReturnCode = %Sublot_Object.Ship (%PrimaryAmt, %SecondaryAmt,  
"customerName");
```

Здесь:

<code>%PrimaryAmt</code>	Объект, определяющий первичную характеристику отгрузки.
<code>%SecondaryAmt</code>	Объект, определяющий вторичную характеристику отгрузки.
<code>customerName</code>	Название клиента, которому отгружается продукция.

Предположим, отгружаемая партия состоит из 100 комплектов документации (первичная характеристика) по 2 кг каждая (вторичная характеристика) некоему заказчику (Wonderware). Необходимо создать два объекта автоматизации: `%PrimaryAmt` и `%SecondaryAmt` – и установить значения их свойств перед выполнением метода `Ship`. Далее описан каждый этап требуемых действий:

1. Создание объекта класса **Amount** для первичной характеристики и установка его свойств:


```
OLE_CreateObject(%PrimaryAmt, "InTrack.Amount");
%PrimaryAmt.Quantity = 100;
%PrimaryAmt.Units = "каждый";
```
2. Создание объекта класса **Amount** для вторичной характеристики и установка его свойств:


```
OLE_CreateObject(%SecondaryAmt, "InTrack.Amount");
%SecondaryAmt.Quantity = 2;
%SecondaryAmt.Units = "кг";
```
3. Вызов метода **Ship** объекта `%Sublot_Object` с использованием следующего синтаксиса:


```
returnCode = %Sublot_Object.Ship(%PrimaryAmt, %SecondaryAmt,
"Wonderware");
```

Обработка ошибок в InTouch

При обработке ошибок необходимо сначала определить, была ли она причинена неправильным использованием объектов OLE-автоматизации или же возвращена сервером OLE-автоматизации. InTouch выводит сообщение об ошибке и регистрирует их в журнале Wonderware Logger.

Исполнение скриптов при возникновении ошибок OLE не прекращается. Предпринимаемые действия зависят от выполняемого в момент возникновения ошибки действия:

- если ошибка возникла во время определения значения свойства, возвращается ноль (0) или пустая строка "";
- если ошибка возникла во время установки значения свойства, оно не изменяется;
- если ошибка присутствует в вызове метода, метод не запускается, никакие значения не изменяются, возвращаемым значением является 0 или пустая строка.

Для диагностики ошибок в системе предусмотрены следующие пять дополнительных функций:

- OLE_GetLastError
- OLE_GetLastErrorMessage
- OLE_ResetObjectError
- OLE_ShowMessageOnObjectError
- OLE_IncrementOnObjectError

В следующих параграфах приведено более подробное их описание.

OLE_GetLastError

Функция возвращает код последней возникшей ошибки, хранящийся в переменной Microsoft "последней ошибки". Синтаксис:

Значение = OLE_GetLastError();

где **Значение** – тег целого типа, в который записывается код ошибки. Коды ошибок представляют собой 32-разрядные числа без знака. Система InTouch не поддерживает такую разрядность целых величин, поэтому коды ошибок с установленными старшими разрядами будут возвращаться неверно.

OLE_GetLastErrorMessage

Функция возвращает текстовое описание ошибки по её коду. Синтаксис:

Значение = OLE_GetLastErrorMessage();

где **Значение** – строка символов (тег типа "сообщение"), в который возвращается текстовое описание ошибки. Этот же текст выводится на экран в окне сообщения об ошибке. В состав текста входят также символы возврата каретки и перевода строки. Если длина текста превышает 131 символ (ограничение InTouch), сообщение усекается.

OLE_ResetObjectError

Значение переменной "последней ошибки" изменяется только при возникновении очередной ошибки и при успешном завершении OLE-операции не сбрасывается. В некоторых случаях разработка приложения упрощается, если записать в переменную "последней ошибки" известное состояние, выполнить несколько операций OLE-автоматизации и проверить изменение значения переменной "последней ошибки". Данная функция предназначена для сброса значения внутренней переменной Microsoft "последней ошибки" в 0. Синтаксис:

OLE_ResetObjectError();

OLE_ShowMessageOnObjectError

Данная функция управляет выводом сообщения на экран и вводом оператора в случае возникновения ошибки OLE-интерфейса или исключительной ситуации OLE-автоматизации. После отображения на экране окна сообщения исполнение скрипта, вызвавшего ошибку, приостанавливается до тех пор, пока оператор не подтвердит сообщение (например, пока он не нажмёт кнопку **ОК**). Фоновые же скрипты будут продолжать исполняться.

В число ошибок OLE-интерфейса входит также невозможность взаимодействия с OLE-инфраструктурой Microsoft, поддерживаемой

операционной системой Windows NT. Ошибка OLE-интерфейса не является следствием или отражением вычислительных ошибок OLE-сервера.

Исключительная ситуация OLE-автоматизации может возникнуть в результате действий как OLE-инфраструктуры, так и OLE-сервера.

Синтаксис вызова функции:

```
OLE_ShowMessageOnObjectError(Логический_тег);
```

где **Логический_тег** – логическая величина, управляющая отображением окна сообщения об ошибке OLE-автоматизации или возникновении исключительной ситуации:

1 = выводить сообщение об ошибке (значение по умолчанию);

0 = не выводить сообщение об ошибке.

Следующий скрипт запрещает вывод окна сообщений об ошибках во время вызова метода **Create** объекта **%Lot**. После того как метод отработает, на экране будет открываться окно сообщения о любых возникающих ошибках:

```
OLE_ShowMessageOnObjectError(0);
```

```
%Lot.Create("abc");
```

```
OLE_ShowMessageOnObjectError(1);
```

OLE_IncrementOnObjectError

Данная функция увеличивает значение указанного тега целого типа на единицу при каждой ошибке OLE-интерфейса или возникновении исключительной ситуации OLE.

Примечание. Если функция **OLE_ShowMessageOnObjectError()** была запущена с единичным аргументом, то значение тега будет увеличиваться **после** того, как оператор подтвердит уведомление о возникшей ошибке.

В число ошибок OLE-интерфейса входит также невозможность взаимодействия с OLE-инфраструктурой Microsoft, поддерживаемой операционной системой Windows NT. Ошибка OLE-интерфейса не является следствием или отражением вычислительных ошибок OLE-сервера.

Исключительная ситуация OLE-автоматизации может возникнуть в результате действий как OLE-инфраструктуры, так и OLE-сервера.

Значение указанного тега будет увеличиваться при каждой ошибке до тех пор, пока не будет сделан вызов функции с названием другого тега. Кроме того, оно не запоминается системой WindowViewer во время завершения работы.

Синтаксис функции:

```
OLE_ShowMessageOnObjectError(IntegerTag);
```

где **IntegerTag** – название тега целого типа, значение которого будет увеличиваться на единицу при каждой ошибке или исключительной ситуации.

Следующий оператор определяет тег целого типа **errorCount** как счётчик ошибок OLE-интерфейса и исключительных ситуаций OLE:

```
OLE_ShowMessageOnObjectError(errorCount);
```

Частые ошибки объектов OLE-автоматизации

В настоящем параграфе перечислены наиболее частые ошибки использования расширений OLE-автоматизации в InTouch и методы их устранения.

- Название OLE-объекта не определено.

Название объекта было использовано в ссылке на свойство (%Объект.Свойство) или вызове метода (%Объект.Метод()) до того, как оно было связано с объектом. Проверьте правильность написания названия объекта. Убедитесь, что название объекта было ранее связано с объектом OLE-автоматизации с помощью функции InTouch OLE_CreateObject() или в результате выполнения операции присваивания.

- Неизвестный объект или метод.

Убедитесь, что название свойства или метода объекта указано верно.

- Попытка присвоить значение свойство типа "только для чтения".

Эта ошибка возникает, когда, например, параметр типа "только для чтения" указан слева от оператора присваивания. Гораздо труднее отследить эту ошибку, если это свойство указано в числе выходных аргументов какой-либо функции.

- Несоответствие типов.

Данные, передаваемые в аргументах функции, не соответствуют им по типу. Процедура автоматического преобразования типов не может выполнить указанное действие. Следует явно присвоить требуемое значение какому-либо тегу InTouch соответствующего типа и указать его в числе аргументов.

- Неверное указание класса.

Как правило, это означает ошибку в указании класса при вызове функции OLE_CreateObject() (второй аргумент).

- Аргументы явно не указаны.

Проверьте, что в вызове функции правильно указано требуемое число аргументов.

- Неожиданные результаты при отсутствии ошибки.

Во время исполнения программы было выполнено преобразование типов данных. Присваивание значения строки тегу целого типа не является ошибкой, однако результат преобразования будет равен нулю, если эта строка не начинается с цифр.

- Указанная ссылка не является ссылкой на объект OLE-автоматизации.

Такая ошибка возникает, если в "многоточечной" ссылке на свойство один из компонентов записи не является внедрённым объектом или ссылкой на объект.

ПРИЛОЖЕНИЕ А

Ошибки выполнения функций InTouch

Некоторые скриптовые функции возвращают значения, определяемые результатом их выполнения. По этому значению (коду завершения, или коду ошибки) можно определять, насколько успешно завершилось выполнение данной функции. В настоящем приложении перечислены коды завершения функций категорий управления окнами, рецептов и SQL. Кроме того, приведено описание DDE-элементов системы SPC (статистического управления процессами), которые можно использовать для получения сведений о наборах данных и управления выводом управляющих графиков на экран.

Сообщения об ошибках функций управления окнами и обработки распределённых алармов

Функции управления окнами и распределёнными алармами возвращают то или иное значение в зависимости от успеха выполнения. Возвращаемое значение можно присвоить тегу целого типа и использовать для диагностики ошибок. Например:

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

Здесь: **ErrorNumber** – название тега целого типа, в который будет записан возвращаемый код ошибки. Возвращаемое функцией значение можно указать в качестве аргумента функции **wcErrorMessage()**, которая по коду ошибки возвратит соответствующий текст её описания. Например:

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

Здесь: **ErrorMsg** – название тега типа "сообщение", в который будет записано описание ошибки в текстовом виде.

В следующей таблице приведены коды завершения и их описание:

Код завершения	Описание
0	Успешное завершение
-1	Общая ошибка
-2	Недостаточно памяти
-3	Свойство имеет атрибут "только для чтения"
-4	Указанный элемент данных уже существует
-5	Неизвестное название объекта

-6	Неизвестное название свойства
-x*	Неизвестная ошибка

* -x обозначает любое другое число.

Ошибки исполнения функций рецептурных скриптов

Чтобы проанализировать код завершения рецептурных функций, его необходимо присвоить тегу InTouch целого типа. Например:

```
ErrorCode = RecipeLoad(FileName, UnitName, RecipeName);
```

Функция **RecipeLoad()** возвратит в тег ErrorCode значение 0, если её выполнение завершилось успешно. При неуспешном выполнении она присвоит аналоговому тегу ErrorCode число, определяемое видом ошибки. В следующей таблице перечислены возможные коды ошибок и их описание.

Значение	Текст сообщения	Описание
0	Success (Успешное выполнение)	Выполнение указанной функции завершилось успешно.
-1	No Such Recipe Template (Файл рецептурных шаблонов не найден)	Указанный файл рецептурных шаблонов не существует.
-2	View Not Active (Просмотр не активен)	Запущенная другой программой рецептурная функция не может быть выполнена, потому что приложение WindowViewer в текущий момент не исполняется.
-3	Out of Memory (Недостаточно памяти)	Недостаточно памяти для выполнения текущей операции.
-4	Line too long in recipe template file (Слишком длинная строка в файле рецептурных шаблонов)	Строка в файле рецептурных шаблонов превысила максимально допустимую длину.
-5	Truncated line in recipe file (Строка в файле рецептов усечена)	Строка в файле рецептурных шаблонов была усечена.
-6	Not a valid recipe template file (Недопустимое название файла рецептурных шаблонов)	Указанный файл не является допустимым .CSV-файлом рецептурных шаблонов. Дополнительные сведения о названиях рецептов или блоков можно найти в системном <i>Руководстве пользователя приложения Recipe manager.</i>

-7	Expecting "unit" or "recipe" (Отсутствует "блок" или "рецепт")	Указанное название блока или рецепта не найдено в файле рецептурных шаблонов. Дополнительные сведения о названиях рецептов или блоков можно найти в системном <i>Руководстве пользователя приложения Recipe manager</i> .
-8	No Units defined in recipe template file (Блоки в рецептурном файле не найдены)	В разделе описания блоков файла рецептурных шаблонов не определено ни одного блока.
-9	Recipe name not found in recipe template file (Название рецепта в файле рецептурных шаблонов не найдено)	Указанное название рецепта в файле рецептурных шаблонов не определено.
-10	Unit name not found in recipe template file (Название блока в файле рецептурных шаблонов не найдено)	Указанное название блока не в разделе описания блоков в файле рецептурных шаблонов не определено.
-12	Expecting "Analog", "Discrete", "Message" (Ожидается тип "Analog", "Discrete", "Message")	Для элемента в файле рецептурных шаблонов указан неправильный данных. Допустимыми являются только типы Analog (Аналоговый), Discrete (Логический) и Message (Сообщение).
-13	Type of tagname mismatches "Analog", "Discrete", "Message" (Ожидается тип тега "Analog", "Discrete", "Message")	Указанный тег не соответствует элементу по типу данных, например, элемент данных рецепта был определен как Analog (аналоговый), а в блоке для него указан тег типа Message (сообщение).
-14	Invalid discrete value, expecting "0", "1" (Неверное логическое значение, ожидается "0" или "1")	Для логического тега в файле рецептурных шаблонов указано неверное логическое значение. Допустимыми значениями являются только значения 0 и 1.
-15	Unable to open temporary file (Невозможно открыть временный файл)	Не удастся открыть временный файл; возможно, из-за недостатка дискового пространства.
-16	Write error while saving recipe template file (Ошибка записи во время сохранения файла рецептурных шаблонов)	Во время записи файла рецептурных шаблонов возникла ошибка.

-17	User did not select (Рецепт не выбран пользователем)	Пользователь нажал кнопку Cancel (отмена) в окне выбора рецептов, вместо того чтобы выбрать название рецепта.
-19	Recipe template in use by another application (Шаблон рецепта используется другим приложением)	Указанный файл рецептурных шаблонов уже открыт, поэтому приложение WindowViewer не имеет к нему доступа.

Вывод сообщений с кодами ошибок рецептурных функций

Каждая рецептурная функция возвращает число, представляющее собой код ошибки выполнения функции. В составе системы InTouch есть функция **RecipeGetMessage()**, с помощью которой можно в аналоговый тег записывать код ошибки, а в тег типа "сообщение" – соответствующее этому коду текстовое сообщение об ошибке.

Это можно сделать в скрипте по изменению данных следующим оператором:

```
RecipeGetMessage(ErrorCode, ErrorMessage, 131);
```

Данный оператор будет автоматически выполняться при каждом изменении значения аналогового тега **ErrorCode**. Во время исполнения скрипта функция **RecipeGetMessage()** будет считывать текущее числовое значение тега **ErrorCode** и возвращать связанное с ним текстовое сообщение в тег **ErrorMessage**.

Сообщения об ошибках для скриптовых функций утилиты принтера алармов

Возвращаемыми кодами ошибок для скриптовых функций утилиты принтера алармов будут положительные числа. Нулевой результат означает, что ошибок нет.

Код ошибки	Сообщение ошибки
0	Успешное завершение (ошибок нет)
1	Instance not found or not running (Экземпляр не найден или не запущен).
2	Interface not initialized (Интерфейс не инициализирован).
3	Failure to access virtual memory (Ошибка доступа к виртуальной памяти).
4	Invalid error code (Неверный код ошибки).
5	Too many instances already running (Очень много экземпляров запущено в астоящий момент).

6	Result string would be too long (Результирующая строка была бы очень длинной).
7	Invalid instance index passed to function (В функцию передан неверный индекс экземпляра).
8	Failed to post message to alarm printer program (Ошибка при отправке сообщения программе принтера алармов).
9	Failed wait for response from alarm printer application (Ошибка ожидания отклика от приложения принтера алармов).
20	To priority must be greater or equal to From priority ("Приорите В" должен быть больше или равен "приоритета Из").
21	Invalid priority value (Неверное значение приоритета).
22	Invalid alarm state (Неверное состояние аларма).
23	Fail to execute command because query is running (Невозможно выполнить команду, так как исполняется запрос).
24	Query String is not valid (Строка запроса неверная).
25	Invalid query processing state (Неверное состояние обработки запроса).
26	Invalid printer state selector (Неверный селектор состояния принтера).
27	Command received by alarm printer window is not recognized (Команда, полученная окном принтера алармов, не распознана).
28	Query could not be started (Запрос не может быть запущен).

Названия DDE-элементов SPC

DDE-элементы SPC позволяют извлекать информацию из наборов данных, а также управлять выводом различных графиков и диаграмм. Названием приложения является SPC, названием группы данных – название набора данных.

Примечание. Все DDE-элементы SPC используются одинаковым образом как в распределённых системах статистического управления, так и в не распределённых, за исключением того что в распределённой системе в качестве названия группы данных должно указываться название косвенного набора данных.

DDE-элементы управления и вывода на экран

DDE-элементы управления и вывода используются для управления и вывода на экран сведений о наборах данных. Элементы управления являются общими для всех узлов. Они представляют собой значения, относящиеся к регистрируемому продукту (о котором накапливается информация) из удалённого набора данных. Элементы вывода являются локальными для каждого узла. Они представляют собой значения выборок данных для продукта, сведения о котором отображаются на экране локального узла.

Модификация выборки применима к регистрируемым и к отображаемым продуктам как из локальных, так и удалённых наборов данных. Изменения, производимые путём щелчка кнопкой мышки на диаграмме, влияют на вывод сведений об отображаемом продукте. DDE-элементы SPC изменяют регистрируемый продукт. Для отображаемого и для регистрируемого продукта поддерживаются отдельные текущие выборки, которыми являются последние записанные выборки данных.

Алармы определяются и сохраняются как для регистрируемого, так и для отображаемого продукта. Сведения об алармах выводятся в режиме исполнения только для регистрируемого продукта.

Примечание. После разделения всех продуктов на регистрируемые и отображаемые многие DDE-элементы действительны только для регистрируемого продукта. Эти элементы отмечены в следующей таблице звёздочкой (*), стоящей перед названием DDE-элемента SPC.

Название элемента	DDE-тип	Доступ	Описание
AutoCollection	Логический	Чт./зап.	Разрешает и запрещает автоматическое накопление данных.
*CalculateControlLimits	Логический	Чт./зап.	Устанавливается в 1 для запуска расчёта пределов действия.
DatasetName	Сообщение (32)	Чт./зап.	Определяет название для косвенного набора данных.
HistogramLCL	Вещественный	Только чтение	Хранит значение нижнего контрольного предела гистограммы, определяемого составом совокупности.
HistogramUCL	Вещественный	Только чтение	Хранит значение верхнего контрольного предела гистограммы, определяемого составом совокупности.
Kurtosis	Вещественный	Только чтение	Вид кривой распределения гистограммы.
LastSampleDisplayed	Целый	Чт./зап.	Устанавливает номер последней отображаемой выборки из набора данных.
*ManualInputDialog	Логический	Чт./зап.	Устанавливается в 1 для открытия встроенного окна ручного ввода.
MeasurementsPerSample	Целый	Только чтение	Хранит число измерений в одной выборке.
NewProduct	Сообщение	Чт./зап.	Используется для создания нового

Название элемента	DDE-тип	Доступ	Описание
	(32)		названия продукта.
NewProductCtrlTitle	Сообщение (32)	Чт./зап.	Заголовок управляющей диаграммы для нового продукта, созданного функцией NewProduct.
NewProductParetoTitle	Сообщение (32)	Чт./зап.	Заголовок диаграммы Парето для нового продукта, созданного функцией NewProduct.
NewProductHistTitle	Сообщение (32)	Чт./зап.	Заголовок графика гистограммы для нового продукта, созданного функцией NewProduct.
*ProductCollected	Сообщение (32)	Чт./зап.	Изменяет название продукта, сведения о котором накапливаются в наборе данных.
ProductDisplayed	Сообщение (32)	Чт./зап.	Изменяет название продукта, о котором из набора данных отображаются сведения.
SampleSize	Целый	Только чтение	Размер выборки для NP.
SamplesPerControlChart	Целый	Чт./зап.	Хранит количество выборок, отображаемых на управляющей диаграмме.
SamplesPerHistogram	Целый	Чт./зап.	Хранит количество выборок, отображаемых на гистограмме.
SamplesPerLimitCalc	Целый	Чт./зап.	Устанавливает количество выборок, используемых в вычислении пределов действия.
SamplesPerPareto	Целый	Чт./зап.	Хранит количество выборок, используемых для построения диаграммы Парето.
SelSPCOutSpecMsg	Сообщение	Только чтение	Тег типа "сообщение" аларма "Sample outside specification Limit" (Выборка вне пределов спецификации).
Skewness	Вещественный	Только чтение	Хранит величину отклонения от среднего значения на гистограмме.
SPCAllowSampDelMod	Логический	Чт./зап.	Разрешает и запрещает отображение команд Delete (удалить) и Modify (изменить) в меню, показываемом при щелчке правой кнопкой мыши на выборке.
SPCConnection	Логический	Только чтение	Устанавливается в 0 в случае потери соединения с сервером.
SPCConnectType	Сообщение	Только чтение	Хранит тип соединения узла: как агента (Server) или как клиента (Client).
SPCLowDBSpace	Логический	Только чтение	Используется для контроля базы данных Microsoft SQL Server. Данный

Название элемента	DDE-тип	Доступ	Описание
			элемент применим только к базам данных только этого типа. Устанавливается в 1 при недостатке свободного пространства в базе. Может использоваться для приостановки автоматического накопления данных и уведомления оператора о необходимости выделения дополнительного пространства. Значение этого элемента автоматически изменяется между 0 и 1 в зависимости от состояния базы данных SQL Server
SPCResetRunRules	Логический	Чт./зап.	Используется для управления правилами обработки поступающих выборок. В частности, начиная сбор данных о новой серии продукции, предшествующие правила обработки поступающих выборок данных иногда требуется отменить.

DDE-элементы текущей выборки SPC

Все DDE-элементы текущей выборки относятся к последней накопленной выборке заданного набора данных. Они могут использоваться для изменения исходных данных и предельных значений, установленных для набора данных. Для обновления информации о текущей выборке необходимо записать какое-либо значение в соответствующий DDE-элемент и установить DDE-элемент **CurrentUpdate** в 1. Результат будет таким же, как при повторном вводе выборки с повторным выполнением всех расчётов. После ввода выборки программа SPC сбрасывает значение DDE-элемента **CurrentUpdate** в 0. При этом после начала цикла накопления следующей выборки DDE-элементы текущей выборки обновить будет нельзя.

DDE-элементы текущей выборки доступны всем узлам. Эти значения соответствуют последней накопленной выборке регистрируемого продукта.

В распределённых системах SPC все значения первоначально равны нулю. Система SPC соединяется с базой данных и проверяет наличие новой информации каждые 5 секунд. Значения элементов данных изменяются при нахождении любых новых сведений. Изменённые значения текущей выборки хранятся в буфере локального узла до тех пор, пока элемент **CurrentUpdate** не будет установлен в 1. После этого значения помещаются в пакет текущей выборки и посылаются удалённому узлу для анализа и хранения. Если изменения текущей выборки относятся к другому продукту, а номер текущей выборки не является номером последней записанной выборки, сервер проигнорирует посланный запрос.

Примечание. После разделения всех продуктов на регистрируемые и отображаемые все DDE-элементы с префиксом "Current" действительны только для регистрируемого продукта.

Название элемента	DDE-тип	Доступ	Описание
-------------------	---------	--------	----------

Название элемента	DDE-тип	Доступ	Описание
CurrentCauseCode	Целый	Чт./зап.	Устанавливает код специальной причины для текущей выборки.
CurrentCauseString	Сообщение (128)	Только чтение	Выводит описание кода специальной причины для текущей выборки.
CurrentComment	Сообщение (50)	Чт./зап.	Используется для чтения и записи любых комментариев для текущей выборки.
CurrentCp	Вещественный	Только чтение	Выводит индекс устойчивости Cp для текущей выборки.
CurrentCpk	Вещественный	Только чтение	Выводит индекс устойчивости Cpk (центрированный Cp) текущей выборки.
CurrentDate	Сообщение (8)	Чт./зап.	Устанавливает дату для текущей выборки в формате ДД/ММ/ГГ. Если дата введена неверно, принимается текущая дата.
CurrentFlag	Логический	Чт./зап.	Устанавливает флаг для текущей выборки.
CurrentIgnoreValue	Логический	Чт./зап.	Устанавливает флаг игнорирования текущей выборки при выводе контрольного графика в режиме AutoScaled (автоматическое масштабирование).
CurrentMx	Вещественный	Чт./зап.	Хранит индивидуальное значение измерения для текущей выборки (x = от 1 до 25).
CurrentR	Вещественный	Только чтение	Хранит диапазон значений текущей выборки.
CurrentRBar	Вещественный	Чт./зап.	Устанавливает средний диапазон текущей выборки.
CurrentRLCL	Вещественный	Чт./зап.	Устанавливает нижний предел действия диапазона.
CurrentRUCL	Вещественный	Чт./зап.	Устанавливает верхний предел действия диапазона.
CurrentSample	Вещественный	Только чтение	Выводит значение последней точки выборки (например X, C, P).
CurrentSampleBar	Вещественный	Чт./зап.	Устанавливает среднее текущей выборки в данной точке выборки.
CurrentSampleNumber	Целый	Только чтение	Выводит число измерений в последней накопленной выборке.
CurrentTarget	Вещественный	Чт./зап.	Устанавливает эталонное значение в данной точке выборки.
CurrentTime	Сообщение (8)	Чт./зап.	Устанавливает время для текущей

Название элемента	DDE-тип	Доступ	Описание
			выборки в формате ЧЧ:ММ:СС. Если время введено неправильно, принимается текущее время.
CurrentUpdate	Логический	Чт./зап.	Для обновления информации о выборке после изменения значений любого из этих полей установите этот элемент в 1.
CurrentXLCL	Вещественный	Чт./зап.	Устанавливает нижний предел действия для текущей выборки.
CurrentXLSL	Вещественный	Чт./зап.	Устанавливает нижний предел спецификации для текущей выборки.
CurrentXUCL	Вещественный	Чт./зап.	Устанавливает верхний предел действия для текущей выборки.
CurrentXUSL	Вещественный	Чт./зап.	Устанавливает верхний предел спецификации для текущей выборки.
SPC2L3Out2SD	Целый	Только чтение	Значение счётчика аларма "2 of the last 3 samples outside of 2 standard Deviation SS" (Отклонение двух из последних трёх выборок больше удвоенного стандартного отклонения SS).
SPC2L3Out2SDMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "2 of the last 3 samples outside of 2 standard Deviation SS" (Отклонение двух из последних трёх выборок больше удвоенного стандартного отклонения SS).
SPC4L5Out1SD	Целый	Только чтение	Значение счётчика аларма "4 of the last 5 samples outside of 1 standard Deviation SS" (Отклонение четырёх из последних пяти выборок больше стандартного отклонения SS).
SPC4L5Out1SDMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "4 of the last 5 samples outside of 1 standard Deviation SS" (Отклонение четырёх из последних пяти выборок больше стандартного отклонения SS).
SPCConSampAltUpDn	Целый	Только чтение	Значение счётчика аларма "Consecutive samples Alternating Up and Down" (Значение последовательных выборок то растёт, то падает).
SPCConSampAltUpDnMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Consecutive samples Alternating Up and Down" (Значение

Название элемента	DDE-тип	Доступ	Описание
			последовательных выборок то растёт, то падает).
SPCConSampln1SD	Целый	Только чтение	Значение счётчика аларма "Consecutive samples Inside 1 standard Deviation" (Отклонение последовательных выборок не превышает стандартного отклонения).
SPCConSampln1SDMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Consecutive samples Inside 1 standard Deviation" (Отклонение последовательных выборок не превышает стандартного отклонения).
SPCConSamplncDec	Целый	Только чтение	Значение счётчика аларма "Consecutive samples Increasing or Decreasing" (Значение последовательных выборок постоянно увеличивается или постоянно уменьшается).
SPCConSamplncDecMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Consecutive samples Increasing or Decreasing" (Значение последовательных выборок постоянно увеличивается или постоянно уменьшается).
SPCConSampOneSideCL	Целый	Только чтение	Значение счётчика аларма "Consecutive samples on one side of centerline" (Значения последовательных выборок находятся на одной стороне от центральной линии).
SPCConSampOneSideCLMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Consecutive samples on one side of centerline" (Значения последовательных выборок находятся на одной стороне от центральной линии).
SPCConSampOut1SD	Целый	Только чтение	Значение счётчика аларма "Consecutive samples outside 1 standard Deviation" (Отклонение последовательных выборок превышает величину стандартного отклонения).
SPCConSampOut1SDMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Consecutive samples outside 1 standard Deviation" (Отклонение последовательных выборок превышает величину

Название элемента	DDE-тип	Доступ	Описание
			стандартного отклонения).
SPCNLNOOutNSD	Целый	Только чтение	Значение счётчика аларма "? Of the last ? samples outside ? standard deviations" (Отклонение X из последних Y выборок превышает N величин стандартного отклонения).
SPCNLNOOutNSDMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "? Of the last ? samples outside ? standard deviations" (Отклонение X из последних Y выборок превышает N величин стандартного отклонения).
SPCNLNOOutNSDSS	Целый	Только чтение	Значение счётчика аларма "? Of the last ? samples outside ? standard deviations SS" (Отклонение X из последних Y выборок превышает N величин стандартного отклонения SS).
SPCNLNOOutNSDSSMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "? Of the last ? samples outside ? standard deviations SS" (Отклонение X из последних Y выборок превышает N величин стандартного отклонения SS).
SPCOutRCtrl	Целый	Только чтение	Значение счётчика аларма диаграммы диапазона "Range outside Control Limit" (Диапазон превышает пределы действия).
SPCOutRCtrlMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма диаграммы диапазона "Range outside Control Limit" (Диапазон превышает пределы действия).
SPCOutXCtrl	Целый	Только чтение	Значение счётчика аларма графа X "Sample outside Control Limit" (Выборка вне пределов действия).
SPCOutXCtrlMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма графа X "Sample outside Control Limit" (Выборка вне пределов действия).
SPCOutSpec	Целый	Только чтение	Значение счётчика аларма "Sample outside specification Limit" (Выборка вне пределов спецификации).
SPCOutSpecMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Sample outside specification Limit" (Выборка вне пределов спецификации).

Название элемента	DDE-тип	Доступ	Описание
SPCRecalculateCp	Логический	Чт./зап.	<p>При установке значения этого элемента в 1 выполняется повторный расчёт индексов Cp и Cpk для текущего набора данных, когда начинается накопление следующей выборки. По окончании накопления значение этого элемента автоматически устанавливается в 0. Чтобы повторно вычислить значения индексов Cp и Cpk, необходимо ещё раз установить этот элемент в 1. После установки в 1 значения индексов Cp и Cpk для следующей накопленной выборки будут вычислены повторно. После накопления значение элемента автоматически сбрасывается в 0.</p> <p>Примечание. Изменение значения этого элемента влияет только на расчёт индексов Cp и Cpk, но не на пределы действия и правила обработки.</p>
SPCResetAlarmCounters	Логический	Чт./зап.	Сброс всех счётчиков алармов.
SPCResetRunRules	Логический	Чт./зап.	<p>Используется для управления правилами обработки поступающих выборок. Применяются к текущему регистрируемому продукту. После установки значения правила обработки сбрасываются, и предыдущие выборки в расчётах алармов типа "4 of the last 5 samples outside of 1 standard Deviation" (Отклонение четырёх из последних пяти выборок больше стандартного отклонения) не используются. Данное действие выполняется только один раз, после чего правила применяются обычным образом. Для того чтобы ещё раз отменить правила, необходимо снова сбросить значение этого элемента в 0 и потом установить в 1.</p>

DDE-элементы ручного ввода SPC

DDE-элементы ручного ввода применяются для создания специальных окон ручного ввода. Чтобы воспользоваться DDE-элементами ручного ввода, необходимо определить значения соответствующих элементов, а затем установить значение элемента **MI_Save** в 1. При этом данные других полей ручного ввода с префиксом **MI** будут введены как новая выборка. После

ввода выборки программа SPC сбрасывает значение DDE-элемента **MI_Save** в 0.

В распределённых системах SPC DDE-элементы ручного ввода существуют в каждом узле отдельно. Их значения сохраняются в локальном буфере до тех пор, пока значение DDE-элемента **MI_Save** не будет установлено в 1. После этого значения помещаются в пакет ручного ввода и отсылаются удалённому узлу набора данных для анализа и хранения.

Примечание. после разделения всех продуктов на регистрируемые и отображаемые все DDE-элементы с префиксом "Manual" действительны только для регистрируемого продукта.

Название элемента	DDE-тип	Доступ	Описание
MI_CauseCode	Целый	Чт./зап.	Устанавливает код специальной причины для выборки, введённой вручную.
MI_CauseString	Сообщение (127)	Только чтение	Выводит описание кода специальной причины для выборки.
MI_Comment	Сообщение (50)	Чт./зап.	Используется для чтения и записи любых комментариев для выборки.
MI_Date	Сообщение (8)	Чт./зап.	Устанавливает дату для текущей выборки в формате ДД/ММ/ГГ. Если дата введена неверно, принимается текущая дата.
MI_Flag	Логический	Чт./зап.	Устанавливает флаг для выборки, введённой вручную.
MI_IgnoreValue	Логический	Чт./зап.	Устанавливает флаг игнорирования текущей выборки при выводе контрольного графика в режиме AutoScaled (автоматическое масштабирование).
MI_Mx	Вещественный	Чт./зап.	Хранит индивидуальное значение указанного измерения (x = от 1 до 25) выборки, введённой вручную.
MI_Save	Логический	Чт./зап.	Сохраняет информацию, вручную введённую в другие поля MI, как новую выборку. Примечание. После установки значения MI_Save в 1 значения всех элементов MI копируются в соответствующие DDE-элементы с префиксом Current , после чего значение элемента CurrentSampleNumber увеличивается на 1.
MI_Time	Сообщение (8)	Чт./зап.	Устанавливает время для текущей выборки в формате ЧЧ:ММ:СС. Если время введено неправильно, принимается текущее время.

DDE-элементы выделения

Все DDE-элементы выделения (SPC Selection DDE Items) могут использоваться для просмотра подробной информации о любой выборке. Элемент Selection используется для ввода номера выборки, сведения о которой требуется вывести на экран. После определения этого значения программа SPC автоматически запишет информацию о выборке с указанным номером в остальные элементы.

Старые данные нельзя изменить, однако можно добавить такие элементы, как код специальной причины, флаг и/или комментарий, определив значения соответствующих элементов и затем установив значение элемента **SelectionUpdate** в 1.

В результате этих действий выбранная запись выборки будет обновлена новыми значениями. После ввода обновлённой выборки программа SPC автоматически сбросит DDE-элемент **SelectionUpdate** в 0.

В распределённой системе SPC элементы выделения существуют отдельно в каждом узле. Они представляют собой значения выборки, сохранённые удалённым узлом для конкретного номера выборки данных о регистрируемом продукте. Когда номер выборки записывается в DDE-элемент Selection, соответствующие сведения считываются из файла выборок на удалённом узле. Старые данные изменить нельзя, однако можно добавить такие элементы, как код специальной причины, флаг и/или комментарий, определив значения соответствующих элементов и затем установив значение элемента **SelectionUpdate** в 1. При установке элемента **SelectionUpdate** в 1, значения элементов Special Cause Code, Comment, Flag и Ignore Value объединяются в пакет и пересылаются удалённому узлу для сохранения.

Примечание. После разделения всех продуктов на регистрируемые и отображаемые все DDE-элементы с префиксом "Selection" действительны только для регистрируемого продукта.

Название элемента	DDE-тип	Доступ	Описание
Selection	Целый	Чт./зап.	Запись в этот элемент номера выборки приводит к обновлению всех остальных полей значениями выборки с указанным номером.
SelectionCauseCode	Целый	Чт./зап.	Устанавливает код специальной причины для выделенной выборки.
SelectionCauseString	Сообщение (128)	Только чтение	Выводит описание кода специальной причины.
SelectionComment	Сообщение (50)	Чт./зап.	Используется для чтения и записи любых комментариев для выделенной выборки.
SelectionCp	Вещественный	Только чтение	Хранит индекс устойчивости Cp для выделенной выборки.
SelectionCpk	Вещественный	Только чтение	Хранит индекс устойчивости Cpk (центрированный Cp) выделенной выборки.
SelectionDate	Сообщение (8)	Только	Хранит дату для выделенной

Название элемента	DDE-тип	Доступ	Описание
		чтение	выборки.
SelectionFlag	Логический	Чт./зап.	Устанавливает флаг для выделенной выборки.
SelectionIgnoreValue	Логический	Чт./зап.	Устанавливает флаг игнорирования выделенной выборки при выводе контрольного графика в режиме AutoScaled (Автоматическое масштабирование).
SelectionMx	Вещественный	Только чтение	В этих элементах хранятся индивидуальные значения измерений (x = от 1 до 25), составляющих выборку.
SelectionProduct	Сообщение (32)	Только чтение	Хранит название продукта в выделенной выборке.
SelectionR	Вещественный	Только чтение	Хранит диапазон значений выделенной выборки.
SelectionRBar	Вещественный	Только чтение	Хранит средний диапазон выделенной выборки.
SelectionRLCL	Вещественный	Только чтение	Хранит нижний предел действия диапазона для выделенной выборки.
SelectionRUCL	Вещественный	Только чтение	Хранит верхний предел действия диапазона для выделенной выборки.
SelectionSample	Вещественный	Только чтение	Хранит значение в точке выделенной выборки.
SelectionSampleBar	Вещественный	Только чтение	Хранит среднее выделенной выборки в данной точке выборки.
SelectionTarget	Вещественный	Только чтение	Хранит эталонное значение для выделенной выборки.
SelectionTime	Сообщение (8)	Только чтение	Хранит время для выделенной выборки.
SelectionUpdate	Логический	Чт./зап.	Для обновления информации о выборке после изменения значений любого из полей с префиксом Selection установите этот элемент в 1.
SelectionXLCL	Вещественный	Только чтение	Хранит нижний предел действия выборки.
SelectionXLSL	Вещественный	Только чтение	Хранит нижний предел спецификации выборки.
SelectionXUCL	Вещественный	Чт./зап.	Хранит верхний предел действия выборки.

Название элемента	DDE-тип	Доступ	Описание
SelectionXUSL	Вещественный	Чт./зап.	Хранит верхний предел спецификации выборки.
SelSPC2L3Out2SDMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "2 of the last 3 samples outside of 2 standard Deviation SS" (Отклонение двух из последних трёх выборок больше удвоенного стандартного отклонения SS).
SelSPC4L5Out1SDMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "4 of the last 5 samples outside of 1 standard Deviation SS" (Отклонение четырёх из последних пяти выборок больше стандартного отклонения SS).
SelSPCConSampAltUpDnMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Consecutive samples Alternating Up and Down" (Значение последовательных выборок то растёт, то падает).
SelSPCConSampIn1SDMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Consecutive samples Inside 1 standard Deviation" (Отклонение последовательных выборок не превышает стандартного отклонения).
SelSPCConSampIncDecMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Consecutive samples Increasing or Decreasing" (Значение последовательных выборок постоянно увеличивается или постоянно уменьшается).
SelSPCConSampOneSideCLMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Consecutive samples on one side of centerline" (Значения последовательных выборок находятся на одной стороне от центральной линии).
SelSPCConSampOut1SDMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Consecutive samples outside 1 standard Deviation" (Отклонение последовательных выборок превышает величину стандартного отклонения).
SelSPCNLNOutNSDMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "? Of the last ? samples outside ? standard deviations" (Отклонение X из последних Y выборок превышает N величин стандартного отклонения).

Название элемента	DDE-тип	Доступ	Описание
SelSPCNLNOOutNSDSSMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "? Of the last ? samples outside ? standard deviations SS" (Отклонение X из последних Y выборок превышает N величин стандартного отклонения SS).
SelSPCOutRCtrlMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма диаграммы диапазона "Range outside Control Limit" (Диапазон превышает пределы действия).
SelSPCOutXCtrlMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма графа X "Sample outside Control Limit" (Выборка вне пределов действия).
SelSPCOutSpecMsg	Сообщение	Только чтение	Название тега типа "сообщение" для аларма "Sample outside specification Limit" (Выборка вне пределов спецификации).

Примечание. С одним и тем же реальным набором данных может быть связано несколько косвенных наборов данных. В этом случае в их элементы Selection могут быть записаны разные номер выборки. Это позволит просматривать подробную информацию о нескольких выборках одного и того же набора данных.

Ошибки исполнения SQL-функций

Все SQL-функции возвращают код завершения `ResultCode`, который может быть использован для поиска и устранения ошибок. Текстовое описание ошибки по её коду можно получить с помощью функции `SQLErrorMsg()`. Например:

```
ErrorMsg=SQLErrorMsg(ResultCode);
```

где:

ErrorMsg – внутренний тег типа "сообщение",

ResultCode – целое значение, возвращённое предыдущей SQL-функцией.

Сообщения об ошибках завершения

Сведения об ошибках, коды которых не приведены в следующей таблице, можно найти в документации по используемой базе данных. Кроме того, дополнительную информацию можно найти в журнале программы-регистратора Wonderware Logger.

Функция `SQLErrorMsg()` записывает текстовое описание ошибки в тег **ErrorMsg** типа "сообщение". В следующей таблице приведён перечень кодов возможных ошибок исполнения SQL-функций, а также описание этих ошибок.

Код завершения	Сообщение об ошибке	Причина
----------------	---------------------	---------

Код завершения	Сообщение об ошибке	Причина
0	No errors occurred (Ошибок нет)	Команда была выполнена успешно.
-5	No more rows to fetch (Больше строк нет)	Достигнута последняя строка таблицы.
-1001	Out of memory (Нехватка памяти)	Недостаточно памяти для выполнения указанной функции.
-1002	Invalid connection (Недопустимое соединение)	Переданный функции идентификатор ConnectionID имеет недопустимое значение.
-1003	No bind list found (Список связей не найден)	Список связей с указанным названием не существует.
-1004	No template found (Шаблон не найден)	Шаблон таблицы с указанным названием не существует.
-1005	Internal Error (Внутренняя ошибка)	Возникла внутренняя ошибка. Необходимо обратиться в службу технической поддержки.
-1006	String is null (Строка пуста)	Предупреждение: считанная из базы данных строка символов пуста (null).
-1007	String is truncated (Строка усечена)	Предупреждение: считанная из базы данных строка длиннее 131 символа и потому была усечена.
-1008	No Where clause (Нет конструкции Where)	В операторе удаления Delete отсутствует конструкция Where.
-1009	Connection failed (Сбой соединения)	Подробное описание причины сбоя соединения нужно посмотреть в журнале Wonderware Logger.
-1010	The database specified on the DB= portion of the connect string does not exist (База данных, указанная конструкцией DB= строки подключения, не существует)	Указанной базы данных не существует.
-1011	No rows were selected (Строки не выбраны)	Была предпринята попытка выполнить команду SQLNumRows(), SQLFirst(), SQLNext() или SQLPrev() ранее команды SQLSelect().
-4149	The connection, statement, or query handle you provided is not valid (Указатель соединения, оператора или запроса неверен)	Возможно, неправильно определен тип столбца. Например, если в шаблоне таблицы тип столбца указан character вместо char для файла dBase, будет возвращена ошибка.

Коды ошибок взаимодействия с отдельными базами данных

Oracle

Сообщение об ошибке	Метод устранения
ORA-03112 – Host String Syntax error (Ошибка в строке хоста)	Если программа NETINIT.EXE не запущена, укажите её в группе автоматического запуска Windows Startup. Если программа NETINIT.EXE выполняется и необходимо организовать более одного соединения или сессии, необходимо выделить дополнительную память. Это можно сделать, установив значение параметра WIN_REMOTE_SESSIONS в файле конфигурации CONFIG.ORA равным числу требуемых соединений. В следующем примере будет выделена память для четырёх соединений: WIN_REMOTE_SESSIONS=4
ORA-3121 – No interface driver connected (Драйвер интерфейса не запущен)	Запустите соответствующую сетевой системе резидентную программу SQL*NET ДО входа в Windows и запуска приложения InTouch SQL Access.
ORA-6435 – NetBIOS: Unable to add local name to name table (Невозможно добавить название локального узла в таблицу названий)	Сетевое программное обеспечение (Novell, LAN Manager и т. д.) должно быть запущено.
ORA-09301 – Local kernel only supported in standard mode (Работа локального ядра возможна только в стандартном режиме)	Укажите в строке подключения название сервера ("SRVR=").
ORA-06430 – Unable to make connection (Невозможно установить соединение)	Проверьте точность и правильность указания атрибутов в строке подключения.

Sybase или Microsoft SQL Server

Сообщение об ошибке	Метод устранения
You cannot have more than one statement active at a time (Нельзя одновременно исполнять более одного оператора)	Была выполнена попытка запустить SQL-команду после вызова функции SQLSelect() . Выполните команду SQLEnd() , чтобы освободить системные ресурсы, которые были выделены для выполнения SQLSelect() , или укажите другой идентификатор соединения ConnectionID для второго оператора.
There is not enough memory available to process the command (Недостаточно памяти для выполнения оператора)	Попробуйте перезагрузить рабочую станцию клиента.
Invalid Object name table name (Неверное название таблицы)	Таблица с указанным названием в используемой базе данных не существует. Укажите название базы данных в конструкции DB=.

dBASE

Сообщение об ошибке	Метод устранения
File or DLL not found (Файл или библиотека не найдены)	В операционной системе Windows файл QEDBF.DLL должен быть или в текущем каталоге, или в системном каталоге Windows, указанном в переменной среды Path.
Invalid connection (Неверное соединение)	Убедитесь в том, что переменной среды Path указан путь доступа к требуемым DLL-библиотекам. Для работы с файлами DBF необходима библиотека QLDBF.DLL.
The connection or statement handle you provided is not valid (Неверный указатель соединения или оператора)	Дополнительные сведения содержатся в журнале Wonderware Logger. Возможна синтаксическая ошибка в выражении SQL-оператора.