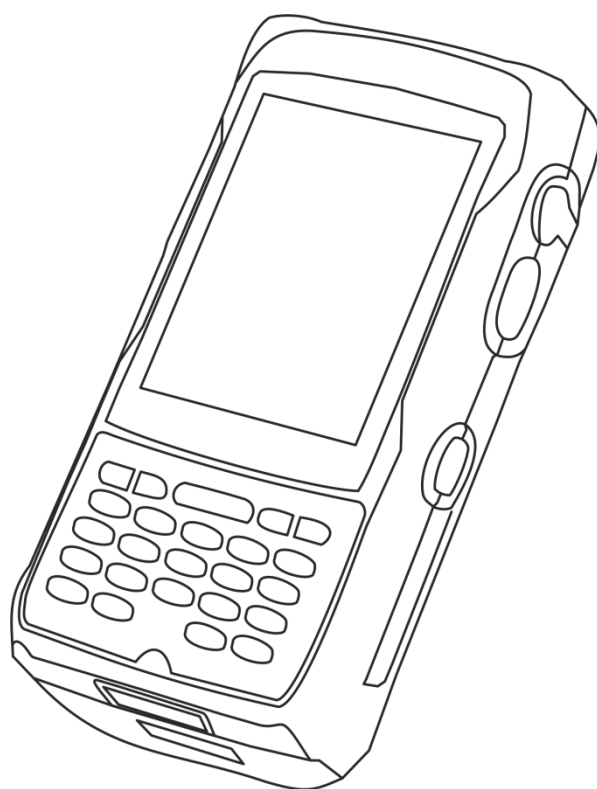


Драйвер терминалов сбора данных

в. 6

2014



Руководство программиста

АТОН

Содержание

Введение	6
Сокращения	6
Условные обозначения	6
О руководстве	6
Описание драйвера	6
Интерфейс драйвера	8
Обработка ошибок	10
ResultCode	10
ResultDescription	11
RaiseException	12
Информация о программе	12
AboutBox() ПоказатьИнформацию()	12
ApplicationHandle	13
Version	13
ServerVersion	13
IsDemo	13
DriverDescription	14
Визуальная страница свойств	14
ShowProperties() ПоказатьСтраницуСвойств()	14
Логические устройства	15
Свойства текущего ЛУ	16
AddDevice() ДобавитьУстройство()	17
DeleteDevice() УдалитьУстройство()	18
CurrentDeviceIndex	20
CurrentDeviceNumber	20
CurrentDeviceName	21
DeviceCount	21
LockDevices	21
Параметры связи с оборудованием	22
PortNumber	23
PortDescription	24
PortIP	24
BaudRate	24
Parity	24
DataBits	25

StopBits.....	25
DeviceIDStr	26
Model.....	26
MachineName.....	27
DeviceEnabled.....	27
Чтение и запись данных.....	28
BeginReport() НачалоОтчета()	28
EndReport() КонецОтчета()	30
GetRecord() ПолучитьЗапись()	30
Clear() Очистить()	31
BeginAdd() НачатьДобавление()	31
SetRecord() УстановитьЗапись()	31
EndAdd() ЗакончитьДобавление()	32
Delimiter	34
ReportFieldName.....	34
ReportFieldIndex	34
FieldDataEnabled	34
FormNumber.....	34
ReportParserEnabled	35
ReportField	35
ReportRecordCount	35
ShowProgress	35
Сервисные функции	35
CancelOperation() ПрерватьОперацию()	35
GetDeviceMetrics() ПолучитьПараметрыУстройства()	36
LoadTask() ЗагрузитьЗадачу()	38
ExchangeFormClose() ФормаОбменаЗакреть()	39
Оставлено для совместимости	39
WriteTSL(FileName: WideString; DeleteData: WordBool) ЗаписатьЗадачу(FileName: WideString; DeleteData: WordBool)	39
Field1 ... Field16.....	40
FormsData.....	40
Обработка событий MobileLogistics v.4.x.....	41
FinishCustomEvent() ЗавершитьОбработкуСобытия()	41
EventDeviceNumber.....	42
Приложение 1. Работа в среде «1С: Предприятие» версий 7.7 и 8.x.	43
Приложение 2. Использование драйвера – ActiveX элемент	45

Excel, Access, Word и т.п.	45
Borland Delphi	46

Введение

Сокращения

ТСД	Терминал сбора данных
ПК	Персональный компьютер
ОС	Операционная система
ПО	Программное обеспечение
ШК	Штрихкод
ML	MobileLogistics

Условные обозначения



Информация, выделенная таким образом, является важной и требует обязательного прочтения и/или выполнения.



Информация, отмеченная такой иконкой, носит ознакомительный и/или рекомендательный характер.



Информация, отмеченная такой иконкой, является примером использования настройки или механизма работы.

О руководстве

Данное руководство программиста является описанием продукта «АТОЛ: Драйвер терминалов сбора данных». При описании подразумевалось, что читатель имеет навыки программирования на одном или нескольких языках программирования для операционных систем: Windows 98 / ME / NT /2000 / 2003 x86 / XP x86 / Vista x86 / 7 x86 / 7 x64, а также знаком с используемым оборудованием (на уровне «Руководства по эксплуатации» из его комплекта поставки).

Ввиду универсальности драйверов не все функциональные возможности оборудования могут быть реализованы в драйвере. Компания «АТОЛ» всегда стремится к поддержке всех функциональных возможностей конкретной модели оборудования, но оставляет за собой право реализации тех функций, которые считает необходимыми. Все возможности драйвера подробно изложены в данном документе, свободно доступном на сайте компании «АТОЛ», с которым можно ознакомиться до приобретения драйвера.

Описание драйвера

«АТОЛ: Драйвер терминалов сбора данных» – программная компонента (драйвер), предназначенная для работы с терминалами сбора данных.

Драйвер представляет собой ActiveX компоненту, работающую под управлением ОС Windows 98 / ME / NT /2000 / 2003 x86 / XP x86 / Vista x86 / 7 x86 / 7 x64 и внешнюю

компоненту для программ системы «1С: Предприятие». Драйвер может использоваться в любых средах разработки поддерживающих технологию OLE Automation:

- 1С: Предприятие;
- Borland Delphi;
- Borland C++ Builder;
- Microsoft Visual C++;
- Microsoft Visual Fox Pro;
- Microsoft Visual Basic;
- Приложения Microsoft Office с VBA (Excel, Word, Access и др.);
- Navision Ахартa и др.

Более подробную информацию о драйверах торгового оборудования и их взаимодействии с оборудованием и программным обеспечением можно найти в документе «АТОЛ: Драйверы торгового оборудования».

Интерфейс драйвера

Интерфейс драйвера состоит из методов и свойств. Все методы представляют собой функции без параметров, возвращающие результат выполнения операции.

Описание метода выглядит следующим образом:

```
// Язык Pascal
function Имя( ) : integer;
// Язык C
int function Имя( );
// Язык Basic
Function Имя( ) As integer;
```

Для передачи параметров используются свойства. Фактически, это глобальные переменные драйвера, в которые можно записать или считать значение. Однажды установленное свойство сохраняет свое значение до последующего его изменения или выгрузки самого драйвера.

Рассмотрим функцию «Добавить логическое устройство» AddDevice (), которой необходимо передать название устройства, а результатом работы будет индекс созданного логического устройства.

Функция могла бы выглядеть следующим образом:

```
// Язык Pascal
function AddDevice (ASrt: string) : integer;
// Язык C
int AddDevice (char * Astr);
// Язык Basic
Function AddDevice (Astr As string) As integer
```

Но в драйвере используются свойства CurrentDeviceName и CurrentDeviceIndex. Таким образом, вызов этого метода будет выглядеть следующим образом:

```
Если Драйвер.AddDevice ( ) <> 0 тогда
// Сообщение пользователю об ошибке
// Выход из программы
КонецЕсли;
Драйвер.CurrentDeviceName = "Устройство на COM1";
Сообщение ("Создано логическое устройство с индексом №" +
ЧислоВСтроку (Драйвер.CurrentDeviceIndex) );
```

Данный подход позволяет:

- устанавливать значения параметров (свойств) перед выполнением метода в любом порядке;
- обращаться к параметрам (свойствам) по именам;
- устанавливать значения только необходимым параметрам (свойствам), а в остальных оставлять последние значения.

Как уже было сказано выше, все методы драйвера являются функциями, возвращающими код результата. Код результата также заносится в свойство `ResultCode`, а его описание – в `ResultDescription`. Также, в драйвере реализована работа с «исключительными ситуациями», что делает использование драйвера более удобным.

Подробнее об использовании исключительных ситуаций, кодов ошибок и их описаний см. в разделе «Интерфейс драйвера \ Обработка ошибок».

При описании интерфейса используются следующие обозначения:

Тип	Описание
[BX]	Входной параметр (применительно к свойствам)
[ВЫХ]	Выходной параметр (применительно к свойствам)

Типы данных:

Тип	Описание
Int	Целое / Integer Целое 32-битное число со знаком. Диапазон значений: -2147483648 ... 2147483647
Db1	Дробное / Double Дробное 64-битное число со знаком. Диапазон значений: $5,0 \times 10^{-324}$... $1,7 \times 10^{+308}$, точность 15 ... 16 знаков после десятичной запятой. Для дробных величин допускается погрешность не более 0,0001. Значения с недопустимой погрешностью рассматриваются как «недопустимое значение» (ошибка – б). Допустимая точность описана для каждого из свойств отдельно
Str	Строка / String Строка символов
Log	Логическое / Logical Целое число, интерпретируемое как «ЛОЖЬ (FALSE)» при значении 0 и «ИСТИНА (TRUE)» в остальных случаях
OLE	Интерфейс для работы через OLE Automation

Типы доступа:

Тип	Описание
R	Только для чтения

Тип	Описание
RW	Для чтения и записи

Обработка ошибок

Свойства

Название	Тип	Дост.	Значение
ResultCode Результат	Int	R	Код ошибки
ResultDescription ОписаниеРезультата	Str	R	Описание кода ошибки
RaiseException ГенерироватьИсключение	Log	RW	Генерировать «исключения» при ошибках

ResultCode

Результат

Содержит код ошибки, возникшей в результате выполнения последней операции (вызова функции или записи значения в свойство драйвера). Если ошибки не произошло, то значение данного свойства устанавливается в 0 (Ошибок нет).

Т.е. ошибки при работе драйвера могут возникать (и, следовательно, свойство ResultCode обновляется) в двух случаях:

- При записи значения в свойство.
- При вызове метода.
- При чтении значения из свойства ошибки не возникают, и свойство ResultCode не обновляется.

Возможны следующие значения:

Код	Описание ошибки
0	Ошибок нет
-1	Нет связи
-3	Порт недоступен
-4	Ключ защиты не найден
-5	Работа драйвера прервана пользователем
-6	Недопустимое значение
-7	Невозможно добавить устройство

Код	Описание ошибки
-8	Невозможно удалить устройство
-9	Устройство не найдено
-10	Неправильная последовательность операций
-11	Устройство не включено
-12	Не поддерживается в данной версии оборудования
-13	Драйвер не смог загрузить необходимые модули
-14	Порт занят приложением: (\\<имя ПК>\<описание приложения-клиента>\<описание драйвера>). Например: «Порт занят приложением: \\Server\1С: Предприятие\Драйвер сканера штрихкода»
-18	Устройство с заданными параметрами не найдено
-199	Неопознанная ошибка
-2400	Нет больше строк отчета
-2401	Таблица содержит данные
-2402	Ошибка от устройства: ошибка при сканировании
-2403	Происходит выполнение команд
-2404	Некорректные данные от устройства
-2405	Неверная контрольная сумма
-2406	Ошибка от устройства: нет больше записей для отчета
-2407	Недопустимое значение параметра устройства
-2409	Неверный формат файла
-2410	Ошибка от устройства: недопустимое значение
-2411	Некорректная настройка таблицы
-2412	Ошибка от устройства: в ТСД нет больше памяти
-2413	Нарушена уникальность данных
-2414	Ошибка от устройства: невозможно добавить запись. Справочник полон

ResultDescription

ОписаниеРезультата

Содержит строку с описанием на русском языке кода ошибки последней выполнявшейся операции (см. столбец «Описание ошибки» в разделе `ResultCode`).

См. также: свойство `ResultCode`.



```
Драйвер.DeviceEnabled = True;
Если Драйвер.ResultCode <> 0 Тогда
  // Сообщить об ошибке
```

RaiseException

Генерировать Исключение

Практически все современные средства разработки (Delphi, C++, VB, «1С: Предприятие» и т.д.) поддерживают работу с исключительными ситуациями (исключениями / Exception). Используя исключение – можно существенно упростить написание программ.

Свойство `RaiseException` определяет – генерировать ли исключительную ситуацию при возникновении ошибки (`ResultCode <> 0`) или нет.

См. также: свойство `ResultCode`, `ResultDescription`.



Попытка

```
Драйвер.Model = 0;  
Драйвер.PortNumber = 1;  
Драйвер.DeviceEnabled = True;
```

Исключение

```
// Выдать сообщение об ошибке
```

Конец Попытки

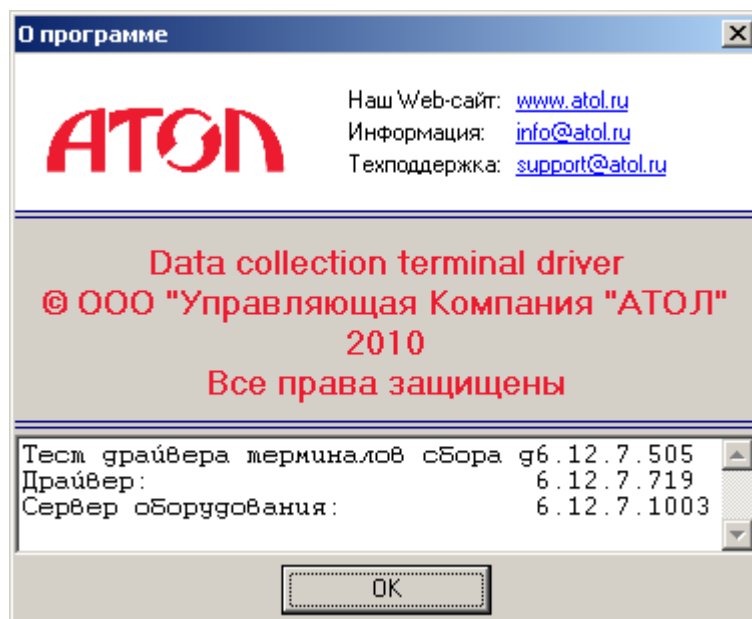
Информация о программе

Методы

AboutBox()

Показать Информацию()

Метод выводит окно с информацией о версии драйвера.



Свойства

Название	Тип	Дост.	Значения
ApplicationHandle	Int	RW	Дескриптор главного окна приложения
Version Версия	Str	R	Версия драйвера
ServerVersion ВерсияСервера	Str	R	Версия сервера
IsDemo ДемонстрационныйРежим	Log	R	Режим работы драйвера
DriverDescription ОписаниеДрайвера	Str	R	Описание драйвера

ApplicationHandle

После загрузки драйвера в данное свойство можно записать дескриптор главного окна приложения-клиента. Это предотвратит появление отдельных кнопок в панели задач при отображении визуальной страницы свойств и других окон драйвера. При использовании драйвера в качестве внешней компоненты (для «1С: Предприятие») данное свойство не поддерживается, так как драйвер самостоятельно при загрузке инициализирует свойство корректным значением. При записи значения в данное свойство следует проявлять особую аккуратность, так как запись некорректного значения может привести к нарушениям работы системы.

Version

Версия

Содержит версию данного драйвера.

ServerVersion

ВерсияСервера

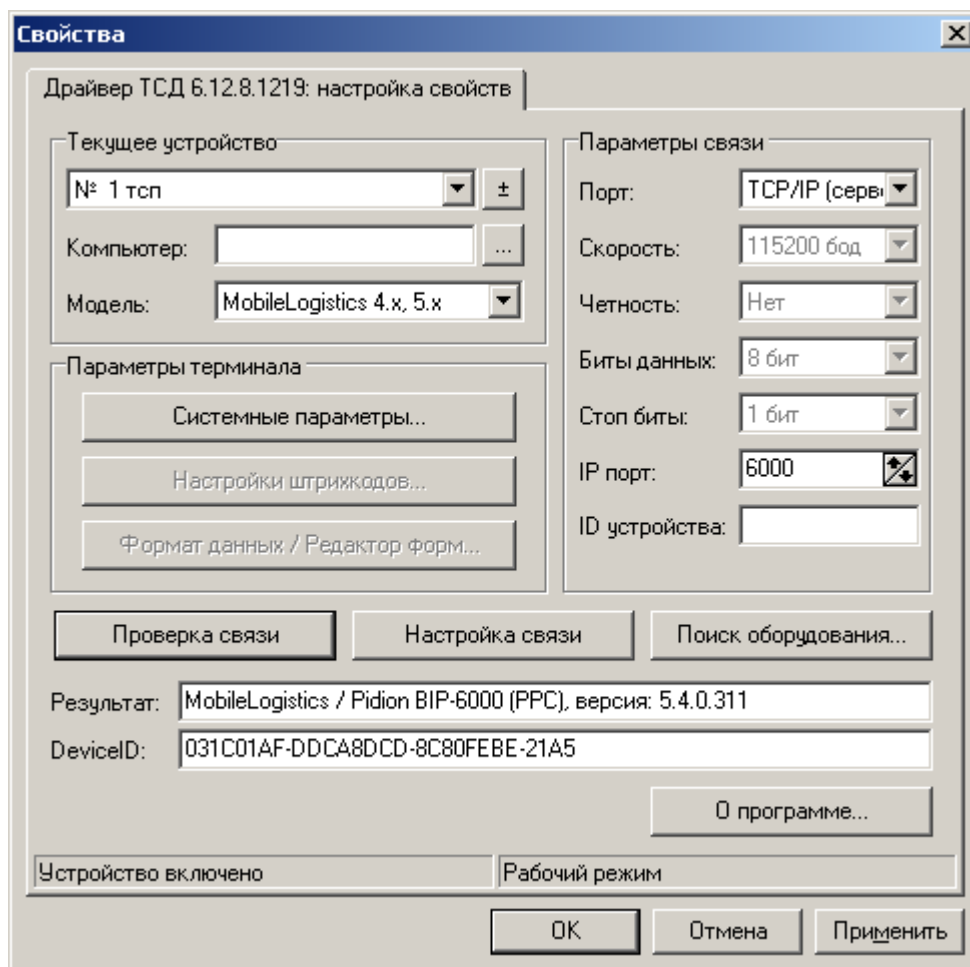
Свойство содержит версию используемого сервера оборудования. Если сервер не был найден и/или подключен, то свойство принимает значение «недоступна».

IsDemo

ДемонстрационныйРежим

Информирует, в каком режиме работает драйвер – демонстрационном или рабочем. Если IsDemo = FALSE, значит драйвер работает в рабочем режиме, без всяких ограничений.

Если `IsDemo = TRUE`, значит драйвер работает в демонстрационном режиме (причиной работы драйвера в демонстрационном режиме является отсутствие ключа защиты программы, подробнее см. разделы «Введение \ Условия распространения» и «Введение \ Демонстрационный режим»). Свойство `IsDemo` обновляется драйвером автоматически при установке свойства `DeviceEnabled = TRUE`, до установки `DeviceEnabled` свойство `IsDemo` не имеет смысла.



DriverDescription

ОписаниеДрайвера

Содержит название драйвера: «Драйвер терминалов сбора данных».

Визуальная страница свойств

ShowProperties() ***ПоказатьСтраницуСвойств()***

Данный метод предоставляет возможность работы с драйвером в более удобной и привычной для пользователя форме. Метод выводит на экран визуальную страницу свойств. Подробнее о странице свойств смотрите в документе «АТОЛ: Драйверы торгового оборудования».

Логические устройства

Логическое устройство – набор свойств драйвера, определяющих параметры связи с оборудованием. Подобных наборов (устройств) одновременно может быть от 1 до 99 штук. Это позволяет, однажды настроив несколько наборов свойств (например: номер порта ПК, скорость обмена данными), быстро применять необходимые параметры, просто переключая устройства. Номер устройству присваивается автоматически при его создании, и изменяться не может. Имя устройства доступно для изменения в любой момент времени.

Все данные о логических устройствах хранятся в системном реестре и загружаются из него же. Если необходимо, чтобы приложение, вызывающее драйвер, не обращалось к реестру, пропишите параметр вида

"Произвольное_имя_переменной"="Имя_Приложения"
в следующих разделах:
[HKEY_CURRENT_USER\Software\ATOL\Drivers\6.0\AppNotLoadDevices]
или
[HKEY_LOCAL_MACHINE\Software\ATOL\Drivers\6.0\AppNotLoadDevices]



[HKEY_CURRENT_USER\Software\ATOL\Drivers\6.0\AppNotLoadDevices]

"Frontol"="Frontol.exe"
"FrontolAdmin"="FrontolAdmin.exe"

или

[HKEY_LOCAL_MACHINE\Software\ATOL\Drivers\6.0\AppNotLoadDevices]

"Frontol"="Frontol.exe"
"FrontolAdmin"="FrontolAdmin.exe"

Текущее устройство. Устройство, свойства которого доступны в текущий момент для чтения и редактирования. Все методы драйвера работают со свойствами именно этого устройства. Чтобы изменить свойства другого устройства, его необходимо предварительно сделать текущим.

Индекс логического устройства. Порядковый номер устройства в списке существующих логических устройств драйвера. То есть при создании нового логического устройства ему присваивается минимальный свободный номер, а индексы пересчитываются так, чтобы номера шли по порядку. При удалении логического устройства из «середины» списка устройств номера остальных устройств не меняются, а индексы изменяются таким образом, чтобы опять получился непрерывный ряд значений.



Существуют три логических устройства с номерами: 1, 2 и 3 с индексами 0, 1 и 2 соответственно. После удаления логического устройства с номером 2 появится «дыра». Т.е. можно сделать текущим логическое устройство с номером 1 или 3, но не 2. Однако индекс устройства № 3 изменился: был «2», а стал «1». При создании нового логического устройства, ему будет присвоен номер 2 и индекс 1 (у устройства №3 индекс поменяется с 1 на 2). Такие «провалы в номерах» создают определенные проблемы при работе с устройствами, по этой причине и было введено понятие «индекс логического устройства».устройства».



Данная технология (логических устройств) позволяет хранить заготовки настроек для разных устройств, подключенных к одному ПК, и оперативно переключать их для работы с нужным устройством.

Все действия, производимые с драйвером (например, получение и обработка данных), производятся непосредственно с текущим логическим устройством.

Свойства текущего ЛУ

В данном разделе описана группа свойств, используемая для изменения параметров текущего логического устройства.

Название	Тип	Дост.	Значения
DevicesSettings ПараметрыУстройств	Str	RW	Параметры логических устройств в виде строки
DeviceSettings ПараметрыУстройства	Str	RW	Параметры логического устройства в виде строки

DevicesSettings

ПараметрыУстройств

Получение и изменение параметров логических устройств в виде строки.



Формат данных в строке параметров может изменяться в новых версиях драйвера. Поэтому не рекомендуется редактировать эту строку в прикладном ПО.



Count=2
CurrentDeviceNumber=3
DeviceNumber0=3
DeviceName0=Феликс
MachineName0=
PortNumber0=1
BaudRate0=18
Model0=24
AccessPassword0=
UseAccessPassword0=1
WriteLogFile0=0
DeviceNumber1=6
DeviceName1=Пилот
MachineName1=
PortNumber1=1
BaudRate1=18
Model1=101
AccessPassword1=1111
UseAccessPassword1=1
WriteLogFile1=0

DeviceSettings

ПараметрыУстройства

Получение и изменение параметров логического устройства в виде строки.



Формат данных в строке параметров может изменяться в новых версиях драйвера. Поэтому не рекомендуется редактировать эту строку в прикладном ПО.



DeviceNumber=6
DeviceName=Пилот
MachineName=
PortNumber=1
BaudRate=18
Model=101
AccessPassword=1111
UseAccessPassword=1
WriteLogFile=0

Методы

AddDevice() **ДобавитьУстройство()**

Выполнение данного метода приводит к созданию нового логического устройства. Свойствам логического устройства после вызова метода присваиваются значения «по умолчанию», кроме номера и индекса, которые присваиваются автоматически и зависят от уже имеющегося количества логических устройств.

Название	Тип	Дост.	Значения
Выходные свойства			
CurrentDeviceIndex ИндексТекущегоУстройства	Int	RW	Индекс текущего ЛУ: 0 .. 98
CurrentDeviceNumber НомерТекущегоУстройства	Int	RW	Номер текущего ЛУ: 1 .. 99
CurrentDeviceName НаименованиеТекущегоУстройства	Str	RW	Название ЛУ: 20 символов
DeviceCount КоличествоУстройств	Int	R	Количество ЛУ: 1 .. 99

CurrentDeviceIndex

ИндексТекущегоУстройства

[ВЫХ]

При добавлении нового логического устройства CurrentDeviceIndex увеличивается на 1.

CurrentDeviceNumber

НомерТекущегоУстройства

[ВЫХ]

Свойство CurrentDeviceNumber содержит минимальный свободный номер логического устройства.

CurrentDeviceName

Наименование Текущего Устройства

[Вых]

Названия логических устройств используются только для удобства пользовательского выбора.

При добавлении нового логического устройства свойство CurrentDeviceName принимает значение «Без названия».

DeviceCount

Количество Устройств

[Вых]

При добавлении нового логического устройства DeviceCount увеличивается на единицу.

Возможные ошибки

Код	Причина
-7	«Невозможно добавить устройство» (Попытка создания более 99 устройств)

DeleteDevice() **Удалить Устройство()**

Метод предназначен для удаления текущего логического устройства. Необходимо помнить, что нельзя удалить все устройства (должно оставаться хотя бы одно), поэтому при попытке удалить последнее логическое устройство, выводится сообщение об ошибке с кодом (-8).

Название	Тип	Дост.	Значения
Входные свойства			
CurrentDeviceIndex Индекс Текущего Устройства	Int	RW	Индекс текущего ЛУ: 0 .. 98
CurrentDeviceNumber Номер Текущего Устройства	Int	RW	Номер текущего ЛУ: 1 .. 99
Выходные свойства			
CurrentDeviceIndex Индекс Текущего Устройства	Int	RW	Индекс текущего ЛУ: 0 .. 98
CurrentDeviceNumber Номер Текущего Устройства	Int	RW	Номер текущего ЛУ: 1 .. 99

Название	Тип	Дост.	Значения
CurrentDeviceName НаименованиеТекущегоУстройства	Str	RW	Название ЛУ: 20 символов
DeviceCount КоличествоУстройств	Int	R	Количество ЛУ: 1 .. 99

CurrentDeviceIndex

ИндексТекущегоУстройства [ВХ/ВЫХ]

При удалении не последнего ЛУ, текущим становится ЛУ следующее за удаляемым. При удалении последнего ЛУ, текущим становится ЛУ, предшествующее удаляемому.

CurrentDeviceNumber

НомерТекущегоУстройства [ВХ/ВЫХ]

При удалении не последнего ЛУ, текущим становится ЛУ, следующее за удаляемым. При удалении последнего ЛУ, текущим становится ЛУ, предшествующее удаляемому.

CurrentDeviceName

НаименованиеТекущегоУстройства [ВЫХ]

При удалении не последнего логического устройства, CurrentDeviceName заполняется значением соответствующим следующему за удаляемым.

При удалении последнего логического устройства, свойство принимает значение предыдущего.

DeviceCount

КоличествоУстройств [ВЫХ]

При удалении текущего логического устройства DeviceCount уменьшается на единицу.

Возможные ошибки

Код	Причина
-8	«Невозможно удалить устройство» (Нельзя удалить все устройства – должно оставаться хотя бы одно)

Свойства

Название	Тип	Дост.	Значения
CurrentDeviceIndex ИндексТекущегоУстройства	Int	RW	Индекс текущего ЛУ: 0 .. 98
CurrentDeviceNumber НомерТекущегоУстройства	Int	RW	Номер текущего ЛУ: 1 .. 99
CurrentDeviceName НаименованиеТекущегоУстройства	Str	RW	Название ЛУ: 20 символов
DeviceCount КоличествоУстройств	Int	R	Количество ЛУ: 1 .. 99
LockDevices БлокироватьУстройства	Log	RW	Запрет добавления / удаления ЛУ

CurrentDeviceIndex

ИндексТекущегоУстройства

Свойство `CurrentDeviceIndex` содержит индекс текущего логического устройства. Изменение данного свойства приводит к установке текущим логического устройства с указанным индексом. Если логического устройства с таким индексом не существует, то выдается сообщение об ошибке с кодом (-9), а `CurrentDeviceIndex` сохраняет значение, содержащееся до операции присвоения. Для определения количества логических устройств, существующих на данный момент, используйте свойство `DeviceCount`. Индекс первого логического устройства равен 0, а последнего `DeviceCount - 1`.

Иногда нужно обработать все логические устройства. Это можно сделать последовательным присвоением свойству `CurrentDeviceIndex` значений от 0 до `DeviceCount - 1`. Если после выполнения такой обработки необходимо восстановить текущее логическое устройство, то для сохранения текущего устройства следует пользоваться свойством `CurrentDeviceNumber`, а не свойством `CurrentDeviceIndex`.

При следующей загрузке драйвера соответствие «индекс-номер» может нарушиться, т.е. можно сказать, что значения этого свойства сохраняются не гарантировано.

См. также: свойства `CurrentDeviceName`, `CurrentDeviceNumber` и `DeviceCount`.

CurrentDeviceNumber

НомерТекущегоУстройства

Данное свойство содержит уникальный номер текущего логического устройства. Каждому логическому устройству при создании присваивается уникальный номер из диапазона от 1 до 99. Номер логического устройства выбирается минимально возможным из незанятых.

Изменение этого свойства приводит к установке текущим логического устройства с указанным номером. Если логического устройства с таким номером не существует, то выдается сообщение об ошибке с кодом (-9), а данное свойство сохраняет свое прежнее значение, содержащееся до операции присвоения.

См. также: свойства `CurrentDeviceName`, `CurrentDeviceIndex` и `CurrentDeviceNumber`.

CurrentDeviceName

НаименованиеТекущегоУстройства

Данное свойство содержит название (описание) логического устройства длиной до 20 символов (при попытке записи более длинной строки будет выдано сообщение об ошибке с кодом (-6), и свойство `CurrentDeviceName` сохранит свое прежнее значение). Название логического устройства используется только для удобного различения их (устройств) между собой. Обычно `CurrentDeviceName` содержит описание той единицы оборудования, для работы с которой это устройство используется. При создании нового логического устройства свойство по умолчанию принимает значение «Без названия»

См. также: свойства `CurrentDeviceIndex` и `CurrentDeviceNumber`.

DeviceCount

КоличествоУстройств

Свойство содержит значение, равное количеству существующих на данный момент логических устройств. При создании устройства свойство увеличивается на единицу, а при удалении – уменьшается на единицу. Свойство общее для всех логических устройств.

См. также: свойства `CurrentDeviceIndex` и `CurrentDeviceNumber`.

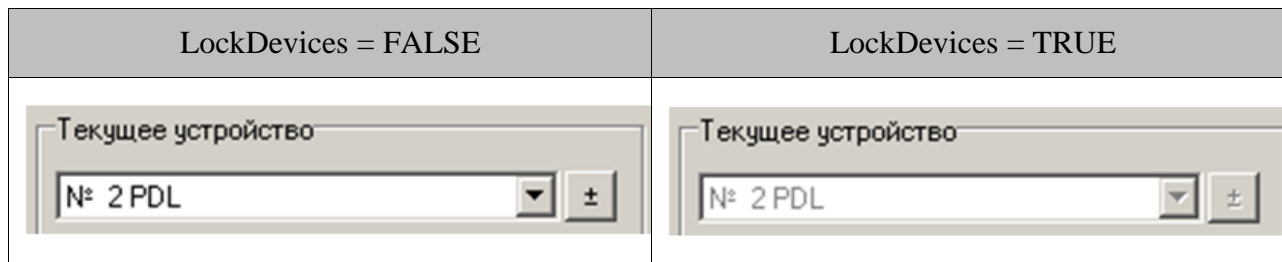
LockDevices

БлокироватьУстройства

Драйвер имеет визуальную страницу свойств, с помощью которой, в частности, можно создавать новые, удалять существующие, изменять текущие логические устройства. Иногда требуется запретить подобные действия, оставив возможность наглядного изменения свойств только текущего логического устройства. Этим и управляет данное свойство. Если `LockDevices = TRUE`, то работа с логическими устройствами при помощи визуальной страницы свойств заблокирована, иначе (`LockDevices = FALSE`) – разрешена.

Отметим, что методы `AddDevice`, `DeleteDevice` и свойства `CurrentDeviceName`, `CurrentDeviceNumber` и `CurrentDeviceIndex` доступны вне зависимости от значения свойства `LockDevices`.

Фрагмент страницы свойств при различных значениях `LockDevices`:



См. также: свойства `CurrentDeviceName`, `CurrentDeviceNumber`, `CurrentDeviceIndex` и методы `AddDevice`, `DeleteDevice` и `ShowProperties`.

Параметры связи с оборудованием

Свойства

Название	Тип	Дост.	Значения
PortNumber НомерПорта	Int	RW	Номер порта ПК: <ul style="list-style-type: none"> • 65 – USB • 101 – TCP/IP • 102 – IRComm (клиент) • 103 – IRComm (сервер) • 1001 – COM1 • ... • 1256 – COM256
PortDescription ОписаниеПорта	Int	R	Содержит строку с описанием порта
PortIP ПортIP	Int	RW	Номер IP-порта 1...65535
BaudRate СкоростьОбмена	Int	RW	Скорость обмена с ПК: <ul style="list-style-type: none"> • 1 – 300 бод; • 2 – 600 бод; • 3 – 1200 бод; • 4 – 2400 бод; • 5 – 4800 бод; • 7 – 9600 бод; • 10 – 19200 бод; • 12 – 38400 бод; • 14 – 57600 бод; • 18 – 115200 бод
Parity Четность	Int	RW	Метод определения четности: <ul style="list-style-type: none"> • 0 – нет;

Название	Тип	Дост.	Значения																				
			<ul style="list-style-type: none"> • 1 – нечетность; • 2 – четность; • 3 – установлен; • 4 – сброшен 																				
DataBits БитыДанных	Int	RW	Количество бит данных: <ul style="list-style-type: none"> • 3 – 7 бит; • 4 – 8 бит 																				
StopBits СтопБиты	Int	RW	Количество стоповых бит: <ul style="list-style-type: none"> • 0 – 1 бит; • 2 – 2 бит 																				
DeviceIDStr ИДУстройства	Str	RW	Строка, по которой драйвер идентифицирует терминал, при работе по интерфейсу USB																				
Model Модель	Int	RW	<table border="1"> <tbody> <tr><td>0</td><td>Zebex PDX</td></tr> <tr><td>1</td><td>Zebex Z-1050</td></tr> <tr><td>3</td><td>Cipher CPT-7xx, CPT-8x00 (провод)</td></tr> <tr><td>4</td><td>Cipher CPT-8x00 (подставка)</td></tr> <tr><td>5</td><td>Cipher 35x0 (радиобаза)</td></tr> <tr><td>6</td><td>Zebex Z-2030</td></tr> <tr><td>7</td><td>АТОЛ: Mobile Logistics</td></tr> <tr><td>8</td><td>АТОЛ: Mobile Logistics Lite Cipher (подставка)</td></tr> <tr><td>9</td><td>АТОЛ: Mobile Logistics Lite</td></tr> <tr><td>10</td><td>MobileLogistics 4.x</td></tr> </tbody> </table>	0	Zebex PDX	1	Zebex Z-1050	3	Cipher CPT-7xx, CPT-8x00 (провод)	4	Cipher CPT-8x00 (подставка)	5	Cipher 35x0 (радиобаза)	6	Zebex Z-2030	7	АТОЛ: Mobile Logistics	8	АТОЛ: Mobile Logistics Lite Cipher (подставка)	9	АТОЛ: Mobile Logistics Lite	10	MobileLogistics 4.x
0	Zebex PDX																						
1	Zebex Z-1050																						
3	Cipher CPT-7xx, CPT-8x00 (провод)																						
4	Cipher CPT-8x00 (подставка)																						
5	Cipher 35x0 (радиобаза)																						
6	Zebex Z-2030																						
7	АТОЛ: Mobile Logistics																						
8	АТОЛ: Mobile Logistics Lite Cipher (подставка)																						
9	АТОЛ: Mobile Logistics Lite																						
10	MobileLogistics 4.x																						
MachineName ИмяКомпьютера	Str	R	Сетевое имя компьютера																				
DeviceEnabled УстройствоВключено	Log	RW	Признак захвата порта драйвером																				

PortNumber

НомерПорта

Определяет номер порта, к которому подключается периферийное устройство.

Значение	65	101	102	103	1001	...	1256
Порт	USB	TCP/IP	IRComm (клиент)	IRComm (сервер)	COM1	...	COM256

Если в момент записи нового значения в данное свойство `DeviceEnabled = TRUE`, то драйвер освобождает текущий порт (устанавливает `DeviceEnabled = FALSE`), изменяет `PortNumber` и пытается самостоятельно захватить новый порт. Если порт занять не удалось, то `DeviceEnabled` остается равным `FALSE`, иначе (порт занять удалось) выставляется `DeviceEnabled = TRUE`.

См. также, свойство `DeviceEnabled`.

PortDescription

ОписаниеПорта

Содержит строку с описанием текущего порта (см. строку «Порт» в разделе `PortNumber`).

См. также: свойство `PortNumber`.

PortIP

ПортIP

Определяет номер IP-порта, на котором ожидается подключение терминала (при использовании интерфейса TCP/IP).

См. также, свойство `DeviceEnabled`.

BaudRate

СкоростьОбмена

Задает значение, определяющее номер скорости обмена данными с периферийным устройством по COM-порту (COM-порт, через который производится связь периферийным устройством указывается через свойство `PortNumber`).

Значен.	1	2	3	4	5	7	10	12	14	18
Скорость бод	300	600	1200	2400	4800	9600	19200	38400	57600	115200

Список разрешенных значений может меняться в зависимости от типа устройства (см. свойство `Model`).

См. также: свойства `PortNumber`, `DataBits`, `Parity`, `StopBits`, `Model` и `DeviceEnabled`.

Parity

Четность

Задает значение, определяющее метод контроля четности при работе с периферийным устройством по COM-порту (COM-порт, через который производится связь с периферийным устройством, указывается через свойство `PortNumber`).

Значение	0	1	2	3	4
Контроль четности	Нет (None)	Нечетность (Odd)	Четность (Even)	Установлен (Mark)	Сброшен (Space)

Список разрешенных значений может меняться в зависимости от типа устройства (см. свойство `Model`).

См. также: свойства `PortNumber`, `BaudRate`, `DataBits`, `StopBits`, `Model` и `DeviceEnabled`.

DataBits

БитыДанных

Задаёт значение, определяющее количество бит данных при работе с периферийным устройством по СОМ-порту (СОМ-порт, через который производится связь с периферийным устройством, указывается через свойство `PortNumber`).

Значение	3	4
Бит в байте	7	8

Список разрешенных значений может меняться в зависимости от типа устройства (см. свойство `Model`).

См. также: свойство `PortNumber`, `BaudRate`, `Parity`, `StopBits`, `Model` и `DeviceEnabled`.

StopBits

СтопБиты

Задаёт значение, определяющее количество стоповых бит при работе с периферийным устройством по СОМ-порту (СОМ-порт, через который производится связь с периферийным устройством, указывается через свойство `PortNumber`).

Значение	0	2
Стоп-бит	1	2

Список разрешенных значений может меняться в зависимости от типа устройства (см. свойство `Model`).

См. также: свойства `PortNumber`, `BaudRate`, `Parity`, `DataBits`, `Model` и `DeviceEnabled`.

DeviceIDStr

ИДУстройства

Свойство задает идентификатор, по которому драйвер опознаёт терминал при работе по интерфейсу USB.

Если используется только один ТСД, можно оставить это свойство пустым, тогда драйвер будет работать с первым найденным устройством.

В случае использования нескольких терминалов удобно заполнять это свойство с помощью кнопки «Закрывать с применением настроек» на странице поиска оборудования.

Если драйвер не находит устройство с заданным идентификатором, возвращается ошибка -18 «Устройство с заданными параметрами не найдено».

См. также: свойства PortNumber, Model, DeviceEnabled.

Model

Модель

Содержит значение, по которому определяется тип периферийного устройства, на работу с которым настроено текущее логическое устройство.

Знач.	Описание	Модели устройств
0	Zebex PDX	Zebex PDL-20, PDC-10, PDL-10, PDW-10, PDM-10, PDT-10
1	Zebex Z-1050	Zebex Z-1050 («АТОЛ»)
3	Cipher CPT-7xx, CPT-8x00 (провод)	Cipher CPT-711, CPT-720, CPT-8300 (соединение с ПК через кабель)
4	Cipher CPT-8x00 (подставка)	Cipher CPT-8000, CPT-8300 (соединение с ПК IR подставку)
5	Cipher CPT-35x0 (радиобаза)	Cipher CPT-3510 для серии CPT-8x1x
6	Zebex Z-2030	Zebex Z-2030
7	АТОЛ: Mobile Logistics	Терминалы с ОС WinCE/PocketPC/ Windows Mobile и установленным ПО АТОЛ: Mobile Logistics. Терминалы Symbol SPT-1800, SPT-1550 с установленным ПО АТОЛ: Mobile Logistics
8	АТОЛ: Mobile Logistics Lite Cipher (подставка)	Cipher CPT-8x00 (соединение с ПК через подставку)
9	АТОЛ: Mobile Logistics Lite	Casio DT-900/DT-930 и Cipher CPT-800x/8300 (через провод)
10	MobileLogistics 4.x	Терминалы с ОС WinCE/PocketPC/ Windows Mobile, а также Casio DT-900/DT-930 с установленным ПО АТОЛ: Mobile Logistics 4.3 и выше.

MachineName

ИмяКомпьютера

Свойство задает имя ПК в сети, к порту которого подключен ТСД. Если свойство пустое, то используется имя ПК по умолчанию, указанное в конфигураторе DCOM для данного драйвера. Если работа с ТСД ведется через порт удаленной машины, то должен быть настроен DCOM (подробнее смотри «АТОЛ: Драйверы торгового оборудования», раздел «Настройка DCOM»). Если DCOM на ПК не установлен, то свойство должно быть пустым и работа будет вестись через порт ПК, на котором запущено клиентское приложение.

При изменении значения свойства MachineName в зависимости от состояния свойства DeviceEnabled, возможны 2 варианта действий:

Свойство DeviceEnabled = FALSE:

1. Если указано имя ПК такое же, как и текущее, то никаких действий не производится.
2. Если значение изменилось, то драйвер запоминает новое значение.

Далее, до включения DeviceEnabled, никаких действий не производится.

Свойство DeviceEnabled = TRUE:

1. Если указано имя ПК такое же, как и текущее, то никаких действий не производится.
2. Драйвер освобождает порт на текущем ПК.
3. Разрушается используемый объект сервера оборудования (*LS.EXE).
4. Запоминает новое значение.
5. Создается новый объект сервера оборудования. При задании пустой строки объект создается на «ПК по умолчанию для данного драйвера» (см. настройку DCOM). Если DCOM не установлен, то объект создается на той же машине, на которой запущено клиентское приложение.

Если создать новый объект сервера оборудования не удалось (на удаленном ПК не установлен драйвер или ПК с таким именем нет в сети), то драйвер возвращает ошибку с кодом (-13).

6. Драйвер пытается занять порт на новом ПК. Если не удалось занять, то DeviceEnabled = FALSE.

После изменения MachineName обновляется свойство ServerVersion.

DeviceEnabled

УстройствоВключено

Свойство определяет: захватил драйвер порт, или нет.

В момент установки свойства (DeviceEnabled = TRUE) драйвер пытается занять порт, а при (DeviceEnabled = FALSE) – освободить.

При захвате порта могут возникнуть ошибочные ситуации: например, когда порт не существует или занят другой программой. При неудачной попытке захвата порта драйвер возвращает ошибку и устанавливает значение DeviceEnabled в FALSE.

При переключении логического устройства порт не освобождается (если он захвачен), это обеспечивает одновременное получение данных от нескольких устройств. Т.е. одновременно у нескольких логических устройств, настроенных на разные порты, может быть установлено свойство `DeviceEnabled = TRUE` (соответственно, если уже существует логическое устройство, занявшее порт, то занять этот же порт другим логическим устройством не удастся).



Перед установкой `DeviceEnabled = TRUE` рекомендуется выставить нужные значения `PortNumber`, `Parity`, `BaudRate`, `DataBits`, `StopBits`. Если использовать визуальную часть свойств драйвера (что предпочтительнее), то нет необходимости каждый раз самостоятельно восстанавливать значения в вышеперечисленных свойствах – драйвер самостоятельно устанавливает необходимые настройки для каждого логического устройства.

Подробнее смотри «АТОЛ: Драйверы торгового оборудования».

Чтение и запись данных

Методы

BeginReport() ***НачалоОтчета()***

Метод выполняет чтение всех данных из памяти терминала в память ПК (буфер драйвера). Номер таблицы, из которой считываются данные, задается в свойстве `FormNumber`. Если свойство `ShowProgress` содержит `TRUE`, то при чтении данных из ТСД, на экран ПК будет выведено окно с индикатором прогресса.

Название	Тип	Дост	Значения
Входные свойства			
<code>FormNumber</code> НомерФормы	Int	RW	Номер таблицы: 0..99
<code>ShowProgress</code> ПоказатьПрогресс	Log	RW	Показывать / не показывать полосу прогресса
Выходные свойства			
<code>ReportFieldCount</code> КоличествоПолейОтчета	Int	R	Количество полей в таблице
<code>ReportRecordCount</code> КоличествоЗаписейОтчета	Int	R	Количество записей в таблице

FormNumber

НомерФормы

[BX]

В данном свойстве задается номер таблицы ТСД, данные из которой требуется прочитать.



Некоторые модели ТСД (*Zebex Z-1050, CipherLab*) не поддерживают подобную возможность, в этом случае выбор таблицы происходит автоматически или осуществляется оператором в ТСД.

ShowProgress

ПоказатьПрогресс

[ВХ]

Если данное свойство равно TRUE , то при передаче данных из ТСД в ПК на экран выводится окно с индикатором прогресса.

ReportFieldCount

КоличествоПолейОтчета

[ВЫХ]

Количество полей в таблице со считанными данными.



Если модель ТСД возвращает информацию о количестве полей в передаваемой таблице, то свойство устанавливается равным этому значению. В противном случае свойство принимает максимальное допустимое значение. Для ТСД *CipherLab* количество полей всегда равно 16.

ReportRecordCount

КоличествоЗаписейОтчета

[ВЫХ]

Количество записей в таблице, полученной от ТСД.

```
//Пример организации чтения данных
//ТАБЛИЦА - таблица лежащая на форме, где нулевая строка - шапка,
//а все остальные - строки с данными.
Процедура ЧтениеДанныхТаблицыИзТСД
Начало
// Начинаем снятие отчета
PDX.BeginReport;
// Количество столбцов получаем из драйвера
ТАБЛИЦА.КоличествоКолонок := PDX.ReportFieldCount;
// Заполняем шапку таблицы
for I := 0 to PDX.ReportFieldCount - 1 do
begin
PDX.ReportFieldIndex := I;
ТАБЛИЦА.Ячейка[I, 0] := PDX.ReportFieldName;
end;
// Цикл по записям (строкам)
for I := 1 to PDX.ReportRecordCount do
begin
PDX.GetRecord;
// Цикл по полям (столбцам)
for J := 0 to (ТАБЛИЦА.КоличествоКолонок - 1) do
begin
```

```

PDX.ReportFieldIndex := J;
ТАБЛИЦА.Ячейка[J, I] := PDX.ReportField;
end;
end;
// Завершаем снятие отчета
PDX.EndReport;
Конец

```

EndReport() **КонецОтчета()**

Метод завершает снятие отчета и очищает внутренний буфер драйвера, если были выполнены методы BeginReport и GetRecord.

GetRecord() **ПолучитьЗапись()**

Команда получения очередной записи отчета из буфера драйвера. После ее выполнения данные записи (поля) доступны для чтения. Структура полей (назначение, тип) зависят от формата таблицы терминала, который настраивает пользователь. Поэтому клиентское приложение должно само интерпретировать получаемые данные. При попытке получения записи до начала отчета (BeginReport) или после его закрытия методом EndReport, GetRecord вернет ошибку с кодом (-10).

Название	Тип	Дост.	Значения
Выходные свойства			
ReportField ПолеОтчета	Str	RW	Строка с данными текущего поля текущей записи

ReportField

ПолеОтчета

[ВЫХ]

Строка с данными текущего поля (индекс поля задается в свойстве ReportFieldIndex) текущей записи.

```

//ТАБЛИЦА - таблица лежащая на форме, где нулевая строка - шапка,
//а все остальные - строки с данными.
// Получаем запись
PDX.GetRecord;
// Цикл по полям
for J := 0 to (ТАБЛИЦА.КоличествоКолонок - 1) do
begin
PDX.ReportFieldIndex := J;
ТАБЛИЦА.Ячейка[J, I] := PDX.ReportField;
end;

```

Clear() **ОЧИСТИТЬ()**

Этот метод стирает все данные в таблице (номер таблицы задается в FormNumber) терминала. После этой команды все данные в памяти терминала безвозвратно теряются.

Название	Тип	Дост.	Значения
Входные свойства			
FormNumber НомерФормы	Int	RW	Номер таблицы. Нумерация 0..99

FormNumber

НомерФормы

[BX]

В данном свойстве задается номер таблицы ТСД, которая будет очищена.

BeginAdd() **НачатьДобавление()**

Метод переводит драйвер в состояние буферизации записей, добавляемых методом SetRecord. Это означает, что все последующие команды записей сохраняют данные во временном буфере (в памяти ПК) и передаются в ТСД только по команде EndAdd. Номер таблицы, в которую добавляются записи, задается в FormNumber.

Название	Тип	Дост.	Значения
Входные свойства			
FormNumber НомерФормы	Int	RW	Номер таблицы. Нумерация 0..99

FormNumber

НомерФормы

[BX]

В данном свойстве задается номер таблицы ТСД, в которую будут добавляться данные.

См. также: методы SetRecord и EndAdd.

SetRecord() **УстановитьЗапись()**

Добавляет запись в таблицу. Поля записи должны быть заполнены до добавления. Значение поля заносится в свойство ReportField, индекс задается свойством ReportFieldIndex.

См. также: методы `BeginAdd`, `EndAdd` и свойства `ReportField`, `ReportFieldIndex`.

EndAdd() ***ЗакончитьДобавление()***

Метод передает все размещенные во временном буфере данные непосредственно в память ТСД.

Название	Тип	Дост.	Значения
Входные свойства			
<code>ShowProgress</code> ПоказатьПрогресс	Log	RW	Показывать / не показывать полосу прогресса
<code>AddMode</code> РежимДобавления	Int	RW	Режим добавления: <ul style="list-style-type: none"> • 0 – добавление или редактирование; • 1 – добавление

ShowProgress

ПоказатьПрогресс

[BX]

Если `ShowProgress = TRUE`, то при записи выводится индикатор прогресса.

См. также: методы `BeginAdd`, `SetRecord` и свойства `ShowProgress`.

AddMode

РежимДобавления

[BX]

В данном свойстве указывается режим добавления данных ТСД.

- Если `AddMode = 0`, то перед добавлением данных в таблицу будет производиться поиск записей по ключевым полям. Если ключевые поля записи в таблице ТСД соответствуют ключевым полям добавляемой записи, то запись в ТСД будет изменена. Иначе запись будет добавлена.
- Если `AddMode = 1`, то добавление записей в ТСД происходит с предварительным удалением всех записей справочника.



В случае, если в таблице ТСД не используются ключевые поля, значение свойства `AddMode` не имеет смысла.

```
//ТАБЛИЦА - таблица лежащая на форме, где нулевая строка - шапка,  
//а все остальные - строки с данными.
```

```
Процедура записи
```

```
Начало
```

```
// Начинаем отчет
```



```

PDX.BeginAdd;
for I := 1 to ТАБЛИЦА.КоличествоСтрок - 1 do
begin
// Заполняем поля
for J := 0 to (ТАБЛИЦА.КоличествоКолонок - 1) do
begin
PDX.ReportFieldIndex := J;
PDX.ReportField := ТАБЛИЦА.Ячейка[J, I];
end;
// Добавляем запись в таблицу (буфер драйвера)
PDX.SetRecord;
end;
// Записываем таблицу (из буфера в ТСД)
PDX.EndAdd;
Конец

```

Свойства

Название	Тип	Дост.	Значения
Delimiter Разделитель	Str	RW	Определяет разделительный символ
ReportFieldName ИмяПоляОтчета	Str	R	Строка с наименованием поля
ReportFieldIndex ИндексПоляОтчета	Int	RW	Определяет индекс поля. 0 .. 62
FieldDataEnabled ПолеДанныхВключено	Log	RW	Устанавливает фильтр для поля
FormNumber НомерФормы	Int	RW	Номер таблицы. Нумерация 0..99
ReportDataNullTerminated НольТерминированныеСтрокиОтчета	Log	RW	Обрезать строки по первому #0 (нультерминированные строки) или нет
ReportField ПолеОтчета	Str	RW	Строка с данными текущего поля текущей записи
ReportRecordCount КоличествоЗаписейОтчета	Int	R	Количество записей в отчете
ReportFieldCount КоличествоПолейОтчета	Int	R	Количество полей в таблице
ShowProgress ПоказатьПрогресс	Log	RW	Показывать / не показывать полосу прогресса

Delimiter

Разделитель

Свойство определяет разделительный символ, использующийся во внутренней структуре данных терминала. Значение свойства – строка из одного символа. Для корректной работы необходимо задать одинаковое значение свойства в ТСД и БД на ПК.

Свойство имеет смысл для следующих моделей ТСД: *Cipher CPT-711, CPT-720, CPT-8000, CPT-8300*, радиобаза *CPT-3510*.

ReportFieldName

ИмяПоляОтчета

Строка с названием поля. Индекс поля задается свойством `ReportFieldIndex`. Названия полей обновляются при вызове метода `BeginReport`.

См. также: метод `BeginReport` и свойства `RecordFieldIndex`, `ReportFieldCount`.

ReportFieldIndex

ИндексПоляОтчета

Определяет индекс поля, название которого можно получить через свойство `ReportFieldName`, а содержимое через `ReportField`. Свойство изменяется от 0 до `ReportFieldCount - 1`. При изменении свойства обновляются значения `ReportFieldName` и `ReportField`.

См. также: методы `BeginReport`, `GetRecord`, свойства `Field1..16`, `ReportFieldName`, `ReportField`, `ReportFieldCount`.

FieldDataEnabled

ПолеДанныхВключено

Свойство позволяет установить фильтр на поле и используется совместно со свойством `ReportFieldIndex`.

Перед вызовом метода `BeginReport` можно указать, следует ли передавать данное поле из ТСД (0 – не передавать поле, 1 – передавать). Перед использованием данного свойство необходимо указать индекс поля (свойство `ReportFieldIndex`).

См. также: методы `BeginReport`, и свойства `ReportFieldIndex`, `ReportFieldCount`.

FormNumber

НомерФормы

В данном свойстве задается номер таблицы ТСД, с которой осуществляется работа. Возможные значения: 0..99.

ReportParserEnabled

Разбирать данные Отчета

Если свойство установлено в TRUE, то:

- строковые значения, возвращаемые свойствами ReportFieldName, ReportField будут «обрезаться» после первого символа с кодом 0 (#0), так называемые нультерминированные строки;
- в дробных числах разделитель (точка) из терминала заменяется на разделитель, настроенный в системе, при получении и передаче данных. Если FALSE, то будут выдаваться в том виде, в каком данные пришли от ТСД.

Значение свойства по умолчанию равно TRUE.

См. также: свойства ReportFieldName, ReportField.

ReportField

Поле Отчета

Свойство содержит строку с данными текущего поля текущей записи. Индекс текущего поля задается в свойстве ReportFieldIndex.

См. также: метод GetRecord, свойства ReportFieldIndex.

ReportRecordCount

Количество Записей Отчета

Количество записей в отчете, снятом методом BeginReport.

См. также: методы BeginReport и GetRecord.

ShowProgress

Показать Прогресс

Если данное свойство равно TRUE, то при передаче данных из ТСД в ПК и наоборот на экран выводится окно с индикатором прогресса.

Сервисные функции

CancelOperation() **Прервать Операцию()**

Метод прерывает все операции обмена данными между терминалом сбора данных и ПК.



Драйвер реализован так, что не блокирует пользовательский интерфейс клиентского приложения и может получать команды от клиента в процессе выполнения других команд. Это позволяет реализовать в обработчике нажатия одной кнопки чтение данных, а в обработчике другой – отмену. Начав снятие отчета нажатием первой кнопки, в случае необходимости его можно отменить, вызвав команду отмены нажатием на другую кнопку. Но в то же время, если необходима блокировка возможности каких-либо действий при выполнении операций, ее разработчику клиентского приложения надо реализовать самому.

GetDeviceMetrics() **ПолучитьПараметрыУстройства()**

При выполнении метода вызывается универсальная команда запроса типа устройства.

Название	Тип	Дост.	Значения
Выходные свойства			
UProtocolVersion УВерсияПротокола	Int	R	Версия формата универсальной команды: 1
UType УТип	Int	R	Тип оборудования: <ul style="list-style-type: none"> • 0 – не определен; • 1 – ККМ; • 2 – весы электронные; • 3 – Memo Plus™ • 5 – ТСД
UModel УМодель	Int	R	Модель оборудования: <ul style="list-style-type: none"> • 0 – ТСД Zebex Z-1050; • 1 – ТСД Zebex PDL-20; • 2 – ТСД Symbol SPT-1800
UMode УРежим	Int	R	Для ТСД (УТип = 5), введены следующие режимы (биты): <ul style="list-style-type: none"> • 15 – ТСД (Накопление данных в памяти); • 14 – Сканер (Сканирование по нажатию кнопки оператором); • 13..0 – Зарезервировано
UMajorVersion УСтаршаяВерсия	Int	R	Версия блока
UMinorVersion УМладшаяВерсия	Int	R	Подверсия блока

Название	Тип	Дост.	Значения
UBuild УСборка	Int	R	Релиз (номер сборки) блока
UCodePage УКодоваяСтраница	Int	R	Используемая блоком кодовая страница
UDescription УОписание	Str	R	Описание оборудования
UVersionStr УВерсия	Str	R	Версия ПО терминала сбора данных

UProtocolVersion

УВерсияПротокола [ВЫХ]

В свойство заносится версия поддерживаемой универсальной команды определения типа оборудования. Данный драйвер корректно может обрабатывать только версию 1 (на сегодня это последняя версия этой команды).

Если значение свойства отлично от 1, то содержимое свойств UType, UModel, UMode, UMajorVersion, UMinorVersion, UCodePage, UDescription не было обновлено из-за того, что драйвер «не умеет» работать с данной версией команды.

UType

УТип [ВЫХ]

В свойство заносится код типа оборудования.

UModel

УМодель [ВЫХ]

В свойство заносится код модели оборудования. Для каждого типа оборудования введено отдельное множество кодов моделей.

UMode

УРежим [ВЫХ]

В свойстве содержится двухбайтовое беззнаковое целое число, представляющее собой битовую кодировку поддерживаемых оборудованием режимов.

UMajorVersion

УСтаршаяВерсия

[ВЫХ]

В свойстве содержится версия оборудования. Данный параметр обычно выводится как первая составляющая версии устройства.

UMinorVersion

УМладшаяВерсия

[ВЫХ]

В свойстве содержится «подверсия» оборудования. Данный параметр обычно выводится как вторая составляющая версии устройства.

UBuild

УСборка

[ВЫХ]

В свойство заносится номер сборки (релиз) оборудования.

UCodePage

УКодоваяСтраница

[ВЫХ]

В свойстве содержится «кодировка» (код языка), для которого адаптированы все ресурсы устройства. Данный параметр обычно выводится как четвертая составляющая версии устройства.

UDescription

УОписание

[ВЫХ]

В свойстве содержится строка, описывающая оборудование. Длина строки может быть произвольной и зависит только от длины названия оборудования, заложенного его разработчиками.

UVersionStr

УВерсия

[ВЫХ]

В свойство заносится версия ПО терминала сбора данных в строковом формате.

LoadTask() ЗагрузитьЗадачу()

Метод загружает файл с конфигурацией в ТСД. Имя файла (с полным путем к нему) заносится в свойство FileName.

Название	Тип	Дост.	Значения
Входные свойства			
FileName ИмяФайла	Str	RW	Имя файла задачи

FileName

ИмяФайла

[BX]

Полное имя файла, т.е. строка, содержащая имя файла и путь к нему.

ExchangeFormClose() **ФормаОбменаЗакреть()**

Метод передает в терминал команду выхода с формы обмена. Данный метод поддерживается только для ТСД с ОС Windows CE/Windows Mobile и установленным ПО MobileLogistics.

Оставлено для совместимости



Методы и свойства, перечисленные в данном разделе, оставлены только для совместимости с предыдущими версиями. Пользуйтесь новыми методами и свойствами, они указаны в пункте «см. также».

Методы

WriteTSL(FileName: WideString; DeleteData: WordBool) **ЗаписатьЗадачу(FileName: WideString; DeleteData: WordBool)**

Записывает слинкованный файл задачи (*.tsl) в ТСД *Zebex PDL-20* и *Z-2030*. Файл создается с помощью утилит, входящих в комплект поставки ТСД. В параметре FileName указывается имя файла, параметр DeleteData указывает записывать задачу, если есть данные или нет, запись приводит к удалению данных.

См. также: метод LoadTask и свойство FileName.

Свойства

Название	Тип	Дост.	Значения
Field1...Field16 Поле1...Поле16	Str	R	Поля с данными
FormsData ДанныеФорм	OLE	R	Возвращает интерфейс IFormsData

Field1 ... Field16

Поле1 ... Поле16

Свойства содержат с 1 по 16 поля, соответственно, очередной строки отчета. Свойства заполняются методом `GetRecord`.

Реальное количество заполняемых полей зависит от настроек ТСД.

См. также: свойства `ReportField` и `ReportFieldIndex`.

FormsData

ДанныеФорм

Данное свойство возвращает интерфейс `IFormsData`, который позволяет получить доступ к интерфейсу `IFormData`, позволяющему записать данные в конкретную таблицу и ее в ТСД.

Обработка событий MobileLogistics v.4.x

В 1С обработка события ML осуществляется в предопределенной процедуре «ОбработкаВнешнегоСобытия». В других средах разработки (Delphi, C++, VBA и т.п.) следует обрабатывать событие «OnCustomEvent» «АТОЛ: Драйвера терминалов сбора данных». Обработчик событий может выполнять различные действия, например, читать/писать/очищать справочники. В конце обработчика события следует заполнить свойства драйвера `EventResultCode` (РезультатСобытия), `EventResultDescription` (РезультатСобытияСтр) и вызвать метод `FinishCustomEvent` (ЗавершитьОбработкуСобытия). Эти параметры будут переданы в функцию ML `ОбменОтправитьЗапрос`, инициировавшую обмен (подробнее о функции в документе «MobileLogistics: Руководство пользователя»).



В серии программных продуктах «1С: Предприятие» аргументы внешнего события заполняются следующим образом: в параметре `Source` (Источник) передаётся `CustomEvent`, в параметре `Message` (Событие) - Код запроса в виде строки, в параметре `Data` (Данные) - Параметр запроса.

FinishCustomEvent() ***ЗавершитьОбработкуСобытия()***

Метод позволяет передать из обработчика события `CustomEvent` в терминал один числовой и один строковый параметр. Его следует вызывать в конце обработчика события.

Название	Тип	Дост.	Значения
Входные свойства			
<code>EventResultCode</code> РезультатСобытия	Int	RW	Код результата обработки события
<code>EventResultDescription</code> РезультатСобытияСтр	Str	RW	Описание результата обработки события

EventResultCode

РезультатСобытия

[BX]

Числовой параметр, который следует передать из обработчика события в ТСД по окончании обработки события.

EventResultDescription

РезультатСобытияСтр

[BX]

Строковый параметр, который следует передать из обработчика события в ТСД по окончании обработки события.

Свойства

Название	Тип	Дост.	Значения
EventDeviceNumber НомерУстройстваСобытия	Int	RW	Номер логического устройства, пославшего запрос

EventDeviceNumber

НомерУстройстваСобытия

Данное свойство содержит номер логического устройства, пославшего запрос. Свойство принимает значения из диапазона от 1 до 99.



Если в обработчике события требуется взаимодействие с ТСД, пославшим запрос (например, чтение или запись справочников), то необходимо явно установить номер текущего логического устройства, присвоив свойству `CurrentDeviceNumber` значение свойства `EventDeviceNumber`. Соответствующая строка кода может выглядеть так:

```
Driver.CurrentDeviceNumber := Driver.EventDeviceNumber.
```

Приложение 1.

Работа в среде «1С: Предприятие» версий 7.7 и 8.x.

Подключение драйвера происходит в глобальном модуле конфигурации для «1С: Предприятие» версии 7.7.

```
Перем Терминал Экспорт; // Глоб. переменная для работы с драйвером
```

```
Процедура ПриНачалеРаботыСистемы()  
// Загрузка внешней компоненты  
Если ЗагрузитьВнешнююКомпоненту("PDX1C.dll") <> 0 Тогда  
// Создание объекта  
Терминал = СоздатьОбъект("AddIn.PDX45");  
Сообщить("Объект загружен");  
Иначе  
Сообщить("Внешняя компонента драйвера ТСД не найдена");  
КонецЕсли;  
КонецПроцедуры
```

```
Процедура ПриЗавершенииРаботыСистемы()  
Терминал = 0; // Отсоединение объекта  
КонецПроцедуры
```

Подключение драйвера происходит в модуле приложения конфигурации для «1С: Предприятие» версии 8.x.

```
Перем Терминал Экспорт; // Глоб. переменная для работы с драйвером
```

```
Процедура ПриНачалеРаботыСистемы()  
Попытка  
// Загрузка внешней компоненты  
ЗагрузитьВнешнююКомпоненту("PDX1C.dll");  
Сообщить("Внешняя компонента загружена");  
Попытка  
// Создание объекта  
Терминал = Новый("AddIn.PDX45");  
Исключение  
Сообщить("Объект не найден!");  
КонецПопытки;  
Исключение  
Сообщить("Внешняя компонента драйвера ТСД не найдена");  
КонецПопытки;  
КонецПроцедуры
```

```
Процедура ПриЗавершенииРаботыСистемы()
```

Терминал = 0; // Отсоединение объекта
КонецПроцедуры



При работе с радиотерминалами через Cipher base CPT-3510 (Mode1 = 5) взаимодействие с драйвером происходит в асинхронном режиме. Для поддержки этого режима необходимо реализовать работу с внешними сообщениями. Для этой цели в 1С присутствует предопределенная процедура «ОбработкаВнешнегоСобытия». Обработка в данной процедуре должна вестись так, как описано в разделе «Асинхронный режим работы».

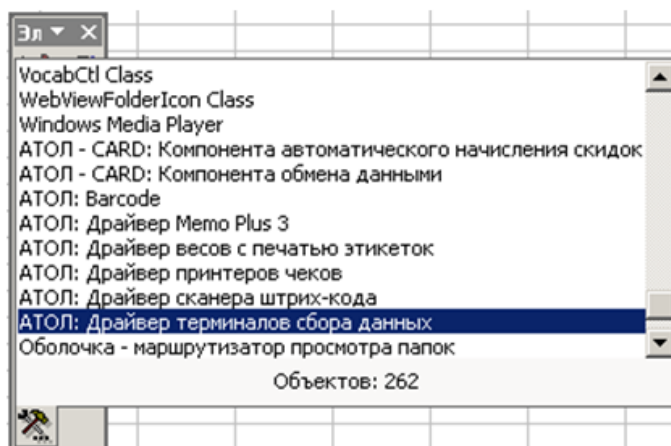
Приложение 2. Использование драйвера – ActiveX элемент

Excel, Access, Word и т.п.

Откройте панель инструментов
«Элементы управления».



Нажмите кнопку «другие элементы» и
выберите в списке «АТОЛ: Драйвер
терминалов сбора данных».

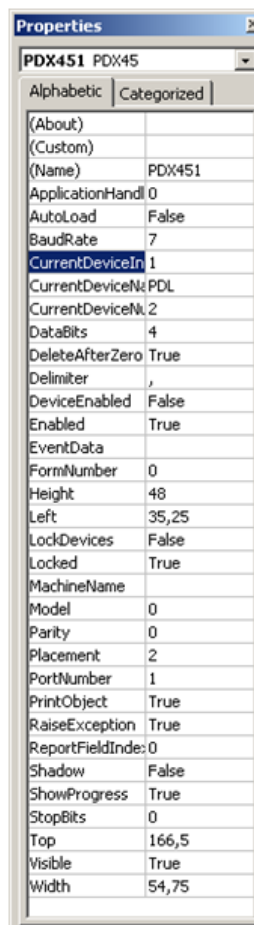


Вставьте компонент на лист (форму).



Далее можно работать с драйвером как с ActiveX компонентой.

Через контекстное меню можно отобразить редактор свойств или визуальную страницу свойств драйвера.



При работе с радиотерминалами *CipherLab* (Model = 5) драйвер работает в асинхронном режиме, т.е. посылает события клиентскому приложению сам. Для получения асинхронных событий напишите нужный код в обработчике события DataEvent (по двойному щелчку мышью открывается редактор Visual Basic на функции – обработчике события).

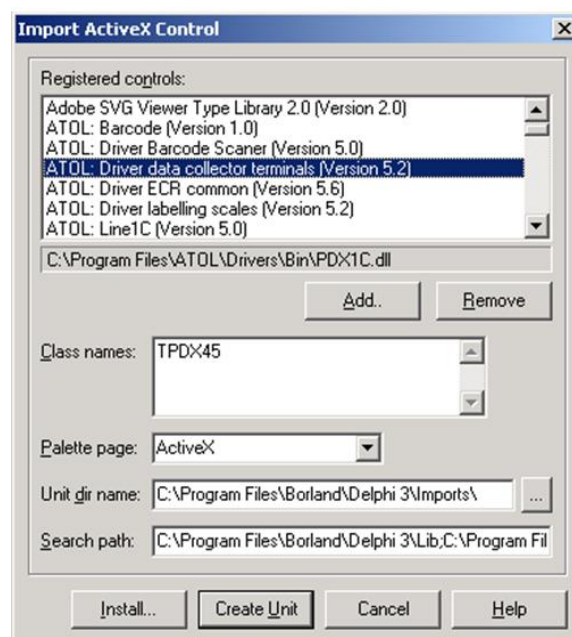
```
//Простейший обработчик внешнего события
Private Sub PDX451_DataEvent()
    StringForMessage = PDX451.PortDescription + ": " + _
    PDX451.EventData
    A = MsgBox(StringForMessage, , _
    "Драйвер терминалов сбора данных")
End Sub
```

Borland Delphi

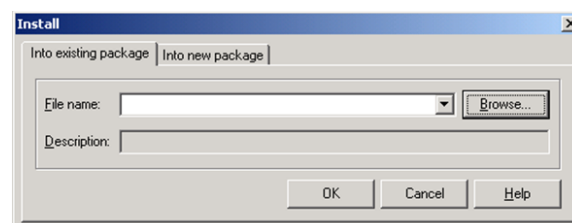
В среде Delphi ActiveX компоненту «АТОЛ: Драйвер терминалов сбора данных» можно поместить на палитру компонентов.

Выберите пункт «Import ActiveX Control...» в меню «Component».

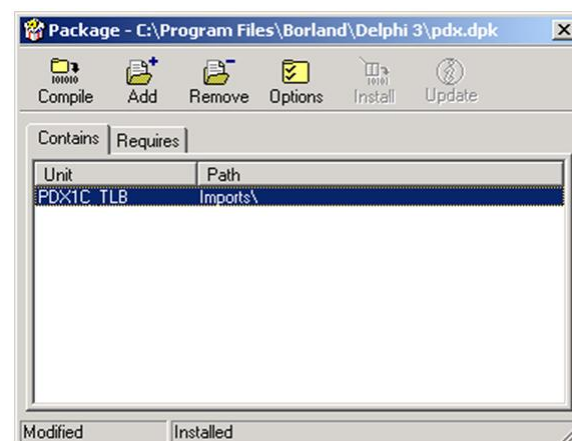
1. Выберите элемент «АТОЛ: Драйвер терминалов сбора данных»;
2. В поле «Palette Page» укажите страницу палитры, на которую желаете добавить компонент;
3. В поле «Unit dir name» задайте директорию, в которой будет сохранен импортируемый модуль;
4. Нажмите кнопку «Install...».



5. Укажите имя существующего (закладка «Into existing package») или нового (закладка «Into new package») пакетного файла.




6. Нажмите «Yes» для подтверждения перекомпиляции пакетного файла.
7. Нажмите «ОК» в окне сообщения об удачной перекомпиляции.
8. Закройте окно пакетного файла.
9. Нажмите «Yes» для подтверждения сохранения пакетного файла.



После этого на закладке «ActiveX» будет расположен компонент «TPDX45». Экземпляры данного компонента имеют обработчик внешних событий – OnDataEvent. Для получения асинхронных событий назначьте обработчик события OnDataEvent для компонента.

+7(495) 730-7420
www.atol.ru



Исключительные права
на программное обеспечение
и документацию принадлежат
компания АТОЛ