

Работа с компонентой AddIn.Cl.TerminalConnector

Последние изменения: 2024-03-26

Компонента AddIn.Cl.TerminalConnector предназначена для использования в среде «1С:Предприятие» (версий 7.7, 8.x). Компонента реализует стандарт 1С на внешние компоненты, а также стандарты 1С на компоненты драйверов торгового оборудования (терминалов сбора данных). Компонента позволяет выполнять соединение с базой данных Mobile SMARTS, выгружать справочник номенклатуры и дополнительные таблицы, выгружать документы Mobile SMARTS для исполнения на терминале, загружать завершённые документы Mobile SMARTS в 1С. Отличие от Cleverence.Warehouse.StorageConnector заключается в том, что для обмена данными используются объекты 1С (массивы, списки значений, таблицы значений и др.), что существенно упрощает написание кода обмена в 1С. При этом возможно совместное использование Add.Cl.TerminalConnector и Cleverence.Warehouse.StorageConnector, а также всех других COM-объектов Mobile SMARTS ([Document](#)

[[Документ](#)], [DocumentItem](#) [[СтрокаДокумента](#)], [Product](#) [[Товар](#)] и др.).

Подключение компоненты и создание объекта

«1С:Предприятие 8»:

```
ПрогИД = "AddIn.Cl.TerminalConnector";
```

```
Если ПодключитьВнешнююКомпоненту(ПрогИД) Тогда КомДляMS = Новый (ПрогИД);
```

```
Иначе
```

```
//Ошибка подключения компоненты КонецЕсли;
```

«1С:Предприятие 7.7»:

```
ПрогИД = "AddIn.Cl.TerminalConnector";
```

```
Если ПодключитьВнешнююКомпоненту(ПрогИД) <> 0 Тогда ОбъектТСД =  
СоздатьОбъект(Компонента);
```

```
Иначе
```

```
//Ошибка подключения компоненты КонецЕсли;
```

Если по каким-то причинам использование ПодключитьВнешнююКомпоненту с передачей программного идентификатора невозможно (например, на сервере «1С:Предприятие»), компонента может быть создана как COM-объект:

«1С:Предприятие 8»:

```
ПрогиД = "AddIn.Cl.TerminalConnector";
```

```
КомДляМС = Новый СОМОбъект(ПрогиД);
```

```
КомДляМС.УстановитьВерсию1С("v8"); //Необходимо сообщить компоненте версию платформы 1С ("v7" или "v8")
```

«1С:Предприятие 7.7»:

```
ПрогиД = "AddIn.Cl.TerminalConnector";
```

```
КомДляМС = СоздатьОбъект(ПрогиД);
```

```
КомДляМС.УстановитьВерсию1С("v7"); //Необходимо сообщить компоненте версию платформы 1С ("v7" или "v8")
```

Подключение к базе данных Mobile SMARTS

Перед выполнением обмена данными (выгрузка номенклатуры, документов и др.) необходимо установить подключение к базе данных Mobile SMARTS, для этого используется функция `Open` [Подключить].

Наименование
Параметры и возвращаемое значение
Описание
<code>Open</code> [Подключить]
<p>Параметры:</p> <p><code>ValuesArray</code> [МассивЗначений],</p> <p><code>DeviceID</code> [ИДУстройства]</p> <p><code>ValuesArray</code> [МассивЗначений] - Массив (для 1С 8.x) или</p> <p>Список значений (для 1С 7.7), содержащий один элемент, в котором должен находиться идентификатор базы Mobile SMARTS (или строка подключения), к которой происходит подключение.</p> <p><code>DeviceID</code> [ИДУстройства] - Возвращается идентификатор ТСД или пустая строка.</p> <p>Возвращаемое значение: Истина (1) — подключение успешно выполнено, Ложь (0) — возникла ошибка. Для получения описания ошибки используйте функцию GetLastError [ПолучитьОшибку].</p>
<p>Выполняет подключение к базе данных Mobile SMARTS и инициализацию компоненты драйвера для последующего обмена с терминалом, папкой обмена или сервером Mobile SMARTS.</p>
<p>Функция входит в один из стандартов 1С на Драйвер для ТСД (См. Описание стандарта).</p>

«1С:Предприятие 8»:

```
ПрогИД = "AddIn.Cl.TerminalConnector";

Если ПодключитьВнешнююКомпоненту(ПрогИД) Тогда КомДляMS = Новый (ПрогИД);

МассивЗначений = Новый Массив;

МассивЗначений.Добавить("михаил-пк:10501/3cb8cb43-ca60-42c5-94d9-bb352dcba9e1");
//Строка

// подключения или ид. базы

ИДУстройства = "";

Если КомДляMS.Подключить(МассивЗначений, ИДУстройства) Тогда

//Успешное подключение Иначе

//Ошибка при подключении к базе ОписаниеОшибки = "";
КомДляMS.ПолучитьОшибку(ОписаниеОшибки);

КонецЕсли;

Иначе

//Ошибка подключения компоненты КонецЕсли;
```

«1С:Предприятие 7.7»:

```

ПрогИД = "AddIn.Cl.TerminalConnector";

Если ПодключитьВнешнююКомпоненту(ПрогИД) <> 0 Тогда КомДляMS =
СоздатьОбъект(ПрогИД);

Параметры = СоздатьОбъект("СписокЗначений");

Параметры.ДобавитьЗначение("михаил-пк:10501/3cb8cb43-ca60-42c5-94d9-bb352dcba9e1");
//Строка

// подключения или ид. базы

ИДУстройства = "";

Если КомДляMS.Подключить(Параметры, ИДУстройства) <> 0 Тогда

//Успешное подключение Иначе

//Ошибка при подключении к базе ОписаниеОшибки = "";

КомДляMS.ПолучитьОшибку(ОписаниеОшибки); КонецЕсли;

Иначе

//Ошибка подключения компоненты КонецЕсли;

```

Выгрузка номенклатуры

Для того, чтобы реализовать выгрузку справочника номенклатуры из 1С в Mobile SMARTS нужно определить, какие данные потребуются на терминале для работы и как отобразятся эти данные на объекты Mobile SMARTS [Product \[Товар\]](#) и [Packing \[Упаковка\]](#) при выгрузке. См. также [Выгрузка](#)

номенклатуры. Самое очевидное, что должны выгружаться штрихкоды товаров и наименования (для того, чтобы при сканировании на терминале происходила идентификация товара по штрихкоду). Штрихкод в базе 1С, как правило, привязан не только к позиции номенклатуры, но и к упаковке (единице измерения), а часто и к характеристике номенклатуры. Таким образом, позиции номенклатуры соответствует объект

[Product \[Товар\]](#), упаковкам и характеристикам — [Packing \[Упаковка\]](#). Наименование характеристики записывается в дополнительное поле Характеристика (descr) упаковки [Packing \[Упаковка\]](#). Для объектов

[Product \[Товар\]](#) требуются уникальные идентификаторы (поле Id), лучше всего использовать Guid-ы позиций справочника номенклатуры (для 1С 7.7 — коды). Кроме того, при выгрузке могут заполняться различные дополнительные поля номенклатуры, определенные в конфигурации Mobile SMARTS. Например, Колво (qty) — остаток товара на складе, Алко — признак, что товар является алкоголем и на терминале следует сканировать ШК акцизной марки. Когда определены все нужные для выгрузки данные, можно приступить к написанию запроса для их получения. Чаще всего запрос выполняется к справочнику номенклатуры и используются соединения для регистров штрихкодов, остатков, цен и др. Товары, не имеющие штрихкодов, тоже обычно выгружаются в Mobile SMARTS, т.к. могут выгружаться документы, содержащие такие товары и чтобы видеть на терминале наименование и другие данные по товару, нужно, чтобы он присутствовал в справочнике номенклатуры. Кроме того, штрихкод к товару можно присвоить на самом терминале.

Для выгрузки используются следующие функции объекта `AddIn.Cl.TerminalConnector`:

Наименование
Параметры и возвращаемое значение
Описание
BeginUploadProducts [НачатьВыгрузкуТоваров]
<p>Параметры:</p> <p><code>schema</code> [СхемаВыгрузки]</p> <p>Массив (для 1С 8.x) или Список значений (для 1С 7.7), описывающий схему выгрузки.</p> <p>Содержит наименования полей объектов Product [Товар] и Packing [Упаковка] в виде «<code>Product.ИмяПоля</code>» или «<code>Packing.ИмяПоля</code>». Значения полей передаются с помощью функции AddProductToUpload [ДобавитьВВыгрузкуТоваров].</p> <p>Возвращаемое значение: Истина</p> <p>(1) — выполнено успешно, Ложь (0)</p> <p>- возникла ошибка. Для получения описания ошибки используйте функцию GetLastError [ПолучитьОшибку].</p> <p>Переводит коннектор в режим выгрузки товаров. Выгрузка выполняется с помощью функции AddProductToUpload [ДобавитьВВыгрузкуТоваров].</p>

AddProductToUpload [ДобавитьВВыгрузкуТоваров]

Параметры:

object row [Данные]

Массив (для 1С 8.x) или

Список значений (для 1С 7.7), содержащий поля объектов

Product [Товар] и **Packing**

[Упаковка], заданные при помощи функции **BeginUploadProducts** [НачатьВыгрузкуТоваров].

Возвращаемое значение: Истина

(1) — выполнено успешно, Ложь (0)

- возникла ошибка. Для

получения описания ошибки используйте функцию

GetLastError [ПолучитьОшибку].

Добавляет в выгрузку товаров товар с упаковкой.

EndUploadProducts [ЗавершитьВыгрузкуТоваров]

Параметры: нет

Возвращаемое значение: Истина

(1) — выполнено успешно, Ложь (0)

- возникла ошибка. Для

получения описания ошибки используйте функцию

GetLastError [ПолучитьОшибку].

Завершает выгрузку товаров.

ResetUploadProducts [СброситьВыгрузкуТоваров]

Параметры: нет

Возвращаемое значение: Истина

(1) — выполнено успешно, Ложь (0)

- возникла ошибка. Для

получения описания ошибки используйте функцию

[GetLastError \[ПолучитьОшибку\]](#).

Сбрасывает начатую выгрузку. Функция применяется в случае ошибки в процессе выгрузки номенклатуры.

«1С:Предприятие 8»:

```
ПрогИД = "AddIn.Cl.TerminalConnector"; ПодключитьВнешнююКомпоненту(ПрогИД);
Соединение = Новый (ПрогИД);
```

```
МассивЗначений = Новый Массив;
```

```
МассивЗначений.Добавить("михаил-пк:10501/3cb8cb43-ca60-42c5-94d9-bb352dcba9e1");
//Строка
```

```
// подключения или ид. базы
```

```
ИДУстройства = "";
```

```
Если КомДляMS.Подключить(МассивЗначений, ИДУстройства) Тогда Запрос = Новый
Запрос(ТекстЗапроса);
```

```
ТаблицаТоваров = Запрос.Выполнить().Выгрузить(); //Выполняем запрос
```

```
//Описываем схему выгрузки мДанные = Новый Массив(26);
```

```
//Основные поля товара и упаковки
```

```
мДанные.Установить( 0, "Product.Id" );
```

```
мДанные.Установить( 1, "Product.Marking" );
```

```
мДанные.Установить( 2, "Product.Barcode" );
```

```
мДанные.Установить( 3, "Packing.Barcode" );
```

```
мДанные.Установить( 4, "Product.Name" );
```

```
мДанные.Установить( 5, "Product.BasePackingId" );
```

```
мДанные.Установить( 6, "Packing.Id" );
```

```
мДанные.Установить( 7, "Packing.Name" );
```

```
мДанные.Установить( 8, "Packing.UnitsQuantity" );
```

```
//Дополнительные поля
```

```
мДанные.Установить( 9, "Packing.descr" );
```

```
мДанные.Установить( 10, "Packing.serial" );
```

```
мДанные.Установить( 11, "Packing.price" );
```

```
мДанные.Установить( 12, "Packing.qty" );
```

```
мДанные.Установить( 13, "Product.withserial" );
```

```
мДанные.Установить( 14, "Product.withsn" );
```

```
мДанные.Установить( 15, "Product.КлючСерий" );
```

```
//Дополнительные поля для работы с алкоголем
```

```
мДанные.Установить( 16, "Product.Алко" );
```

```
мДанные.Установить( 17, "Packing.АлкоОбъем" );
```

```
мДанные.Установить( 18, "Packing.АлкоКодВ" );
```

```
мДанные.Установить( 19, "Packing.АлкоНаимВ" );
```

```
мДанные.Установить( 20, "Product.АлкоКрепость" );
```

```
мДанные.Установить( 21, "Product.Производитель" );
```

```
мДанные.Установить( 22, "Product.ПроизВИНН" );
```

```
мДанные.Установить( 23, "Product.ПроизвКПП" );
```

```
мДанные.Установить( 24, "Packing.АлкоКод" );
```

```
мДанные.Установить( 25, "Packing.АлкоМарк" );
```

```
Соединение.УстановитьПоискПоНаименованиюИАртикулу(Истина); //Устанавливаем
```

признак, что будет //доступен поиск по наименованию товара на терминале

//Начинаем выгрузку

Если Не Соединение.НачатьВыгрузкуТоваров(мДанные) Тогда

//Ошибка при начале выгрузки ОписаниеОшибки = "";

Соединение.ПолучитьОшибку(ОписаниеОшибки); Соединение.ОсвободитьРесурсы();

Сообщить(ОписаниеОшибки); Возврат;

КонецЕсли;

Для Каждого СтрокаТовара из ТаблицаТоваров Цикл

ИдТовара = XMLСтрока(СтрокаТовара["НоменклатураСсылка"]); //В качестве идентификатора товара //будем выгружать Guid позиции номенклатуры

Наименование = Строка(СтрокаТовара["Номенклатура"]); Код = СокрЛП(СтрокаТовара["Код"]);

ШК = СокрЛП(СтрокаТовара["Штрихкод"]);

ЕдиницаИзмерения = Строка(СтрокаТовара["Упаковка"]); Характеристика = СокрЛП(СтрокаТовара["Характеристика"]); Серия = СокрЛП(СтрокаТовара["Серия"]);

Цена = СтрокаТовара["Цена"]; Количество = СтрокаТовара["Количество"]; Артикул = СокрЛП(СтрокаТовара["Артикул"]); Коэфф = СтрокаТовара["Коэффициент"];

БазоваяЕдиница =?(СтрокаТовара["ФлагБазовойЕдиницы"], Строка(СтрокаТовара["Упаковка"]), ""); ЕстьСерии = СтрокаТовара["ЕстьСерии"];

СтатусУказанияСерий = СтрокаТовара["СтатусУказанияСерий"];

мДанные.Установить(0, ИдТовара); //Product.Id мДанные.Установить(1, Артикул); //Product.Marking

мДанные.Установить(2, Код); //Product.Barcode - выгружаем Код позиции номенклатуры, чтобы на

//терминале была возможность поиска по коду

мДанные.Установить(3, ШК); //Packing.Barcode - штрихкод упаковки мДанные.Установить(4, Наименование); //Product.Name

мДанные.Установить(5, БазоваяЕдиница); //Product.BasePackingId - ид. базовой упаковки (единицы) //или

пустая строка, если данная единица не является базовой

```
мДанные.Установить( 6, ЕдиницаИзмерения ); //Packing.Id

мДанные.Установить( 7, ЕдиницаИзмерения ); //Packing.Name - в качестве имени и ид.
упаковки

//используем сокращенное наименование (например, "шт", "кг") мДанные.Установить( 8,
Коэфф ); //Packing.UnitsQuantity

мДанные.Установить( 9, Характеристика ); //Packing.descr - наименование характеристики
или пустая

//строка, если нет характеристики

мДанные.Установить( 10, Серия ); //Packing.serial - наименование серии или пустая строка,
если нет

//серии

мДанные.Установить( 11, Цена ); //Packing.price - цена мДанные.Установить( 12, Количество
); //Packing.qty - остаток на складе

мДанные.Установить( 13, ЕстьСерии ); //Product.withserial - ведется учет по сериям
мДанные.Установить( 14, СтатусУказанияСерий ); //Product.withsn - есть серийные номера
мДанные.Установить( 15, ИдТовара ); //Product.КлючСерий - поле для связи с таблицей
Серии (если

//таковая выгружается), ид. позиции номенклатуры или ид. вида номенклатуры

//Алко поля

мДанные.Установить( 16, СтрокаТовара["Алко"]);

мДанные.Установить( 17, СтрокаТовара["АлкоОбъем"]);

мДанные.Установить( 18, СтрокаТовара["АлкоКодВ"]);

мДанные.Установить( 19, СтрокаТовара["АлкоНаимВ"]);

мДанные.Установить( 20, СтрокаТовара["АлкоКрепость"]);

мДанные.Установить( 21, СтрокаТовара["Производитель"]);

мДанные.Установить( 22, СтрокаТовара["ПроизвИНН"]);

мДанные.Установить( 23, СтрокаТовара["ПроизвКПП"]);

мДанные.Установить( 24, СтрокаТовара["АлкоКод"]);

мДанные.Установить( 25, СтрокаТовара["АлкоМарк"]);

//Добавляем массив данных в выгрузку
```

```

Если Не Соединение.ДобавитьВВыгрузкуТоваров(мДанные) Тогда
//Произошла ошибка ОписаниеОшибки = "";
Соединение.ПолучитьОшибку(ОписаниеОшибки); Сообщить(ОписаниеОшибки);
Соединение.СброситьВыгрузкуТоваров(); //Прерываем выгрузку
Соединение.ОсвободитьРесурсы();
Возврат; КонецЕсли;
КонецЦикла;
//Завершение выгрузки
Если Не Соединение.ЗавершитьВыгрузкуТоваров() Тогда ОписаниеОшибки = "";
Соединение.ПолучитьОшибку(ОписаниеОшибки); Сообщить(ОписаниеОшибки);
КонецЕсли;
Соединение.ОсвободитьРесурсы();
Иначе
//Ошибка при подключении к базе ОписаниеОшибки = "";
Соединение.ПолучитьОшибку(ОписаниеОшибки); Сообщить(ОписаниеОшибки);
КонецЕсли;

```

«1С:Предприятие 7.7»:

```

ПрогИД = "AddIn.Cl.TerminalConnector";
ПодключитьВнешнююКомпоненту(ПрогИД); Соединение = СоздатьОбъект(ПрогИД);
Параметры = СоздатьОбъект("СписокЗначений");
Параметры.ДобавитьЗначение("михаил-пк:10501/3cb8cb43-ca60-42c5-94d9-bb352dcba9e1");
//Строка
// подключения или ид. базы
ИДУстройства = "";
Если Соединение.Подключить(Параметры, ИДУстройства) <> 0 Тогда
//Успешное подключение
ТаблицаНоменклатуры = ПолучитьТаблицуНоменклатурыДляВыгрузки();

```

```
//Описываем схему выгрузки
```

```
мДанные = СоздатьОбъект("СписокЗначений");
```

```
мДанные.ДобавитьЗначение("Product.Id" );
```

```
мДанные.ДобавитьЗначение("Product.Marking" );
```

```
мДанные.ДобавитьЗначение("Product.Barcode" );
```

```
мДанные.ДобавитьЗначение("Packing.Barcode" );
```

```
мДанные.ДобавитьЗначение("Product.Name" );
```

```
мДанные.ДобавитьЗначение("Product.BasePackingId" );
```

```
мДанные.ДобавитьЗначение("Packing.Id" );
```

```
мДанные.ДобавитьЗначение("Packing.Name" );
```

```
мДанные.ДобавитьЗначение("Packing.UnitsQuantity" );
```

```
//Дополнительные поля
```

```
мДанные.ДобавитьЗначение("Packing.descr" );
```

```
мДанные.ДобавитьЗначение("Packing.serial" );
```

```
мДанные.ДобавитьЗначение("Packing.price" );
```

```
мДанные.ДобавитьЗначение("Packing.qty" );
```

```
мДанные.ДобавитьЗначение("Product.withserial" );
```

```
мДанные.ДобавитьЗначение("Product.withsn" );
```

```
мДанные.ДобавитьЗначение("Product.КлючСерий" );
```

```
//Дополнительные поля для работы с алкоголем
```

```
мДанные.ДобавитьЗначение("Product.Алко" );
```

```
мДанные.ДобавитьЗначение("Packing.АлкоОбъем" );
```

```
мДанные.ДобавитьЗначение("Packing.АлкоКодВ" );
```

```
мДанные.ДобавитьЗначение("Packing.АлкоНаимВ" );
```

```
мДанные.ДобавитьЗначение("Product.АлкоКрепость" );
```

```
мДанные.ДобавитьЗначение("Product.Производитель" );
```

```

мДанные.ДобавитьЗначение("Product.ПроизвИНН" );
мДанные.ДобавитьЗначение("Product.ПроизвКПП" );
мДанные.ДобавитьЗначение("Packing.АлкоКод" );
мДанные.ДобавитьЗначение("Packing.АлкоМарк" );

Соединение.УстановитьПоискПоНаименованиюИАртикулу(1); //Устанавливаем признак, что
будет

//доступен поиск по наименованию товара на терминале

//Начинаем выгрузку

Если Соединение.НачатьВыгрузкуТоваров(мДанные) = 0 Тогда

//Ошибка при начале выгрузки ОписаниеОшибки = "";

Соединение.ПолучитьОшибку(ОписаниеОшибки); Соединение.ОсвободитьРесурсы();

Сообщить(ОписаниеОшибки); Возврат;

КонецЕсли;

Пока ТаблицаНоменклатуры.ПолучитьСтроку() = 1 Цикл

мДанные.УдалитьВсе();

Код = СокрЛП(ТаблицаНоменклатуры.Номенклатура.Код); //В качестве идентификатора
товара будем выгружать код позиции номенклатуры

Наименование =
СокрЛП(ТаблицаНоменклатуры.Номенклатура.Наименование); ШК = Запрос.Штрихкод;

ЕдиницаИзмерения = СокрЛП(ТаблицаНоменклатуры.ЕдиницаИзмерения);
Характеристика =
СокрЛП(ТаблицаНоменклатуры.Характеристика); Серия =
СокрЛП(ТаблицаНоменклатуры.Серия);

Цена = ТаблицаНоменклатуры.Цена; Количество =
ТаблицаНоменклатуры.Количество; Артикул = СокрЛП(ТаблицаНоменклатуры.Артикул);

....

мДанные.ДобавитьЗначение(Код); мДанные.ДобавитьЗначение(Артикул);
мДанные.ДобавитьЗначение(Код); мДанные.ДобавитьЗначение(ШК);
мДанные.ДобавитьЗначение(Наименование);

.... мДанные.ДобавитьЗначение(СокрЛП(ТаблицаНоменклатуры.АлкоМарк));

//Добавляем массив данных в выгрузку

```

```
//добавляем массив данных в выгрузку

Если Соединение.ДобавитьВВыгрузкуТоваров(мДанные) = 0 Тогда

//Произошла ошибка ОписаниеОшибки = "";

Соединение.ПолучитьОшибку(ОписаниеОшибки); Сообщить(ОписаниеОшибки);

    Соединение.СброситьВыгрузкуТоваров(); //Прерываем выгрузку
Соединение.ОсвободитьРесурсы();

Возврат; КонецЕсли;

    КонецЦикла;

//Завершение выгрузки

Если Соединение.ЗавершитьВыгрузкуТоваров() = 0 Тогда ОписаниеОшибки = "";
Соединение.ПолучитьОшибку(ОписаниеОшибки); Сообщить(ОписаниеОшибки);

КонецЕсли;

Соединение.ОсвободитьРесурсы();

Иначе

//Ошибка при подключении к базе ОписаниеОшибки = "";
Соединение.ПолучитьОшибку(ОписаниеОшибки);

КонецЕсли;
```

 COM-компонента

Не нашли что искали?



Задать вопрос в техническую поддержку