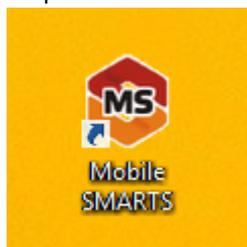


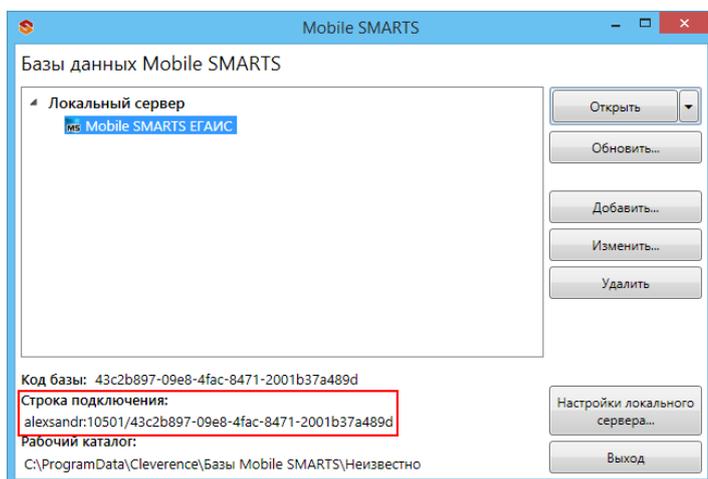
## Инструкция по интеграции базы Mobile SMARTS для произвольной базы «1С:Предприятия» (v8)

### 1. Первоначальная настройка и подключение

1. Устанавливаем платформу и конфигурацию Mobile SMARTS ЕГАИС;
2. На рабочем столе появится иконка для запуска Mobile SMARTS



3. Зайдя в Mobile SMARTS, мы можем проверить адрес подключения и рабочий каталог, по которому расположена база «Mobile SMARTS: ЕГАИС»:



4. В установку входит внешняя компонента Cl.TerminalConnector, которая (если установка прошла корректно), уже будет зарегистрирована в системе.
5. Если эта компонента встала успешно, то мы получили возможность создавать com объект драйвера, и можем обращаться напрямую к нашей базе «Mobile SMARTS: ЕГАИС»

Вызываем компоненту и подключаем необходимую базу:

```
//мДрайверТСД = новый СОМОбъект("Cleverage.Warehouse.StorageConnector"); // Для серверной версии
ПрогиД = "AddIn.Cl.TerminalConnector";
ПодключитьВнешнююКомпоненту(ПрогиД);
мДрайверТСД = Новый(ПрогиД);
СтрокаПодключения = "dt-501-2:10501/6d7e3a2f-3967-4734-bb9c-d1ba605aaef8";
МассивПодключения = Новый Массив;
МассивПодключения.Добавить(СтрокаПодключения);
Если Не мДрайверТСД.Подключить(МассивПодключения, "") Тогда
    Сообщить("Не удалось подключиться к базе данных MS");
КонецЕсли;
// СтрокаПодключения - строка подключения из настройки базы MS
```

## 2. Обмен данными

Обмен данными происходит в три этапа:

Этап 1 – выгрузка номенклатуры

Этап 2 – выгрузка документов

Этап 3 – загрузка документов

### Этап первый – выгрузка номенклатуры

Перед выгрузкой номенклатуры мы должны её инициализировать:

```
Если Не мОбъектТсД.НачатьВыгрузкуТоваров(мДанные) Тогда
    мОбъектТсД.ПолучитьОшибку(мОбъектТсД.ОписаниеОшибки);
    ВывестиПредупреждение("Ошибка при инициализации выгрузки: " + мОбъектТсД.ОписаниеОшибки);
    Сообщить("Ошибка при инициализации выгрузки: " + мОбъектТсД.ОписаниеОшибки,
    СтатусСообщения.Важное);
    мОбъектТсД.ОсвободитьРесурсы();
    Возврат;
КонецЕсли;
```

Если по какой-либо причине начать выгрузку товаров не удалось, то система выдаст сообщение об ошибке. Далее мы определяем, что есть определенная структура полей, которая неизменна. На 24.11.2015 она имеет следующий состав: (сначала формируем массив строк в 1С для выгрузки, затем циклом последовательно его обходим – по каждой позиции.

## мДанные – строка выгрузки.

```
// +++ Перед началом выгрузки необходимо передать в Mobile SMARTS структуру полей выгрузки
// там, где тип указан явно - int или Boolean, они и есть. В других случаях тип поля всегда
строка
мДанные = Новый Массив(25);
мДанные.Установить( 0, "Product.Ид" );
мДанные.Установить( 1, "Product.Marking" );
...
...
мДанные.Установить( 24, "Product.АлкоКод" );
    Если Не мДрайверТСД.НачатьВыгрузкуТоваров(мДанные) Тогда // Происходит проверка на
соответствие структуры полей
        Ошибка = мДрайверТСД.ПолучитьОшибка("Описание ошибки");
        Сообщить("Ошибка при инициализации выгрузки: " + Ошибка,
СтатусСообщения.Важное );
        мДрайверТСД.ОсвободитьРесурсы();
        Возврат;
    КонецЕсли;
СправочникНоменклатуры = Справочники.Номенклатура.Выбрать();
    Пока СправочникНоменклатуры.Следующий() Цикл
Если НЕ СправочникНоменклатуры.ЭтоГруппа Тогда
мДанные.Установить( 0, XMLСтрока(СправочникНоменклатуры.Ссылка)); //"Product.Ид" );
мДанные.Установить( 1, СправочникНоменклатуры.Артикул); //"Product.Marking" );
мДанные.Установить( 2, СправочникНоменклатуры.Код); //"Product.Barcode" );
мДанные.Установить( 3, "0000000000000"); //"Packing.Barcode" );
мДанные.Установить( 4, СправочникНоменклатуры.Наименование); //"Product.Name" );
мДанные.Установить( 5,
XMLСтрока(СправочникНоменклатуры.ЕдиницаДляОтчетов.Ссылка)); //"Product.BasePackingId" );
мДанные.Установить( 6,
XMLСтрока(СправочникНоменклатуры.ЕдиницаДляОтчетов.Ссылка)); //"Packing.Ид" );
мДанные.Установить( 7,
СправочникНоменклатуры.ЕдиницаДляОтчетов.Наименование); //"Packing.Name" );
мДанные.Установить( 8, 1); //"Packing.UnitsQuantity" );
мДанные.Установить( 9, ""); //"Packing.descr" );
мДанные.Установить( 10, ""); //"Packing.serial" );
мДанные.Установить( 11, 0); //"Packing.price" );
мДанные.Установить( 12, 1); //"Packing.qty" );
мДанные.Установить( 13, ложь); //"Product.withserial" );
мДанные.Установить( 14, Истина); //"Product.Алко" ); продукцией (bool)
мДанные.Установить( 15, "лицензия на продажу спирта"); //"Product.АлкоВидЛиц" );
мДанные.Установить( 16, Истина); //"Product.АлкоМарк" );
мДанные.Установить( 17, "420"); //"Product.АлкоКодВ" );
мДанные.Установить( 18, "СправочникНоменклатуры.Наименование"); //"Product.АлкоНаимВ" );
мДанные.Установить( 19, 1); //"Product.АлкоОбъем" );
мДанные.Установить( 20, 25); //"Product.АлкоКрепость" );
мДанные.Установить( 21, "РуОпт"); //"Product.Производитель" );
мДанные.Установить( 22, "7774444454"); //"Product.ПроизвИНН" );
мДанные.Установить( 23, "7707707700"); //"Product.ПроизвКПП" );
мДанные.Установить( 24, "22550"); //"Product.АлкоКод" );
Если Не мДрайверТСД.ДобавитьВВыгрузкуТоваров(мДанные) Тогда
    Ошибка = мДрайверТСД.ПолучитьОшибка("Описание ошибки");
    Сообщить( "Ошибка при выгрузке данных: " + ". " + Ошибка, СтатусСообщения.Важное );
    Прервано = Истина;
Прервать;
КонецЕсли;
КонецЕсли;
КонецЦикла;
// ++ После того, как весь товар передан на СТД, необходимо поверить, что в процессе
передачи не возникло исключений
Если Не мДрайверТСД.ЗавершитьВыгрузкуТоваров() Тогда
    НомерОшибки = мДрайверТСД.ПолучитьОшибка("Номер ошибки");
    Сообщить("Ошибка при завершении выгрузки: " + НомерОшибки);
    Прервано = Истина;
КонецЕсли;
```

Понятно, что в тестовой выгрузке номенклатуры ряд полей заполнены произвольно. (алкоКод, АлкоОбъем, и т.д.). Эти поля заполняются в зависимости от используемой конфигурации.

Список полей для заполнения в **товаре**:

Поле	Тип	Основное/дополнительное	Описание
<b>Id</b>	string	основное	Уникальный идентификатор товара.
<b>Name</b>	string	основное	Наименование товара.
<b>BasePackngId</b>	string	основное	Уникальный идентификатор базовой упаковки товара.
<b>Marking</b>	string	основное	Артикул.
<b>withserial</b>	int	дополнительное	Ведется ли серийный учет по товару (1 – ведется, 0 – нет).
<b>Алко</b>	bool	дополнительное	Является алкоголе содержащей продукцией.
<b>АлкоКод</b>	string	дополнительное	Код алкогольной продукции.
<b>АлкоВидЛиц</b>	string	дополнительное	Вид лицензии (просто информационное поле).
<b>АлкоМарк</b>	bool	дополнительное	Является маркируемой продукцией.
<b>АлкоКодВ</b>	string	дополнительное	Код вида алкогольной продукции.
<b>АлкоНаимВ</b>	string	дополнительное	Наименование вида алкогольной продукции.
<b>АлкоОбъем</b>	double	дополнительное	Объем.
<b>АлкоКрепость</b>	double	дополнительное	Крепость.
<b>Производитель</b>	string	дополнительное	Производитель.
<b>ПроизвИНН</b>	string	дополнительное	ИНН Производителя.
<b>ПроизвКПП</b>	string	дополнительное	КПП Производителя.

Список полей для заполнения в **упаковке**:

Поле	Тип	Основное/дополнительное	Описание
<b>Id</b>	string	основное	Уникальный идентификатор товара.
<b>Name</b>	string	основное	Наименование упаковки.
<b>UnitsQuantity</b>	double	основное	Коэффициент (кратность) единицы измерения.
<b>Barcode</b>	string	основное	Штрихкод товара (обычно EAN13) в текущей упаковке.
<b>price</b>	double	дополнительное	Стоимость.
<b>qty</b>	double	дополнительное	Количество.
<b>serial</b>	string	дополнительное	Серия, привязанная к упаковке (для серийного учета).
<b>descr</b>	string	дополнительное	Характеристика товара, привязанная к упаковке.

Заполнив таким образом данные, мы можем отправить их в терминал (по адресу и порту сервера, который получили при первоначальной настройке).

```
// Выгрузка
Если Не мОбъектТсД.ДобавитьВВыгрузкуТоваров(мДанные) Тогда
    мОбъектТсД.ПолучитьОшибка(мОбъектТсД.ОписаниеОшибки);
    Сообщить("Ошибка при выгрузке данных: " + ". " + мОбъектТсД.ОписаниеОшибки,
        СтатусСообщения.Важное);
    Прервано = Истина;
    Прервать;
КонецЕсли;
```

Если данные не отправлены, получим ошибку.

После того, как все данные выгружены, необходимо завершить выгрузку товаров:

```
Если Не мОбъектТсД.ЗавершитьВыгрузкуТоваров() Тогда
    НомерОшибки = мОбъектТсД.ПолучитьОшибка(мОбъектТсД.ОписаниеОшибки);
    Сообщить("Ошибка при завершении выгрузки: " + мОбъектТсД.ОписаниеОшибки);
    Прервано = Истина;
КонецЕсли;
```

Реализуем с проверкой на ошибку.

## Этап второй. Выгрузка документов

Аналогично работе с номенклатурой, инициализируем внешнюю компоненту Клеверенс. После чего, определяем, какой документ, и куда мы будем выгружать. На сервере Mobile SMARTS ЕГАИС по умолчанию доступны следующие документы:

- Собрать штрихкоды;
- Сбор начальных остатков;
- Приход на склад;
- Подбор заказа;
- Инвентаризация.

Применительно к алкогольной тематике, мы можем проставить следующее соответствие:

- сбор начальных остатков = Акт постановки на баланс;
- приход на склад = ТТН ЕГАИС в статусе «от поставщика»;
- подбор заказа = ТТН ЕГАИС в статусе «покупателю».

### Сбор начальных остатков.

Описание полей:

#### Реквизиты «шапки» документа

Реквизит	Тип	Основное/доп.	Описание
<b>Id</b>	string	основное	Идентификатор документа. Обязательно заполняется при выгрузке документа и должен иметь уникальное значение.
<b>Name</b>	string	основное	Отображаемое имя документа. Обязательно заполняется при выгрузке документа.
<b>Appointment</b>	string	основное	Назначение документа - код пользователя или имя группы, которым данный документ назначается на исполнение. Если значение пустое, то документ попадает к первому свободному пользователю, которому разрешен тип документа.

<b>UserID</b>	string	основное	Идентификатор пользователя, который выполнил документ. Должен иметь уникальное значение для каждого пользователя.
<b>UserName</b>	string	основное	Имя пользователя.
<b>DocumentTypeName</b>	string	основное	Имя типа документа. Обязательно заполняется при выгрузке документа и должно иметь уникальное значение.
<b>Modified</b>	boolean	основное	Признак того, что документ был изменен. Заполняется при изменении документа одним из пользователей.
<b>InProgress</b>	boolean	основное	Признак того, что документ захвачен пользователем на обработку. Заполняется, когда документ был захвачен пользователем.
<b>Finished</b>	boolean	основное	Признак того, что обработка документа пользователем была завершена, и его можно забирать назад в учетную систему. Заполняется, когда работа с документом была завершена на терминале.
<b>WareHouseID</b>	string	основное	Идентификатор склада, к которому привязан документ. Должен иметь уникальное значение для каждого склада.
<b>Barcode</b>	string	основное	Штрихкод документа. Для выбора документа по штрихкоду на терминале или на сервере должен быть обязательно заполнен.
<b>Priority</b>	Int32	основное	Приоритет документа. Более приоритетные документы раньше отдаются на терминал для обработки.
<b>DistributeByBarCode</b>	boolean	основное	Признак выдачи документа по штрихкоду. Документы с таким признаком не поступают на мобильный терминал автоматически, а могут быть выбраны с сервера только по штрихкоду. Свойство может успешно применяться только при наличии постоянной связи с сервером.
<b>ServerHosted</b>	boolean	основное	Признак того, что документ должен выполняться "на сервере". Такой документ могут одновременно открыть на редактирование несколько пользователей. Все изменения в документе будут происходить одновременно для всех работающих с ним пользователей. Работа в таком режиме требует наличия постоянной связи с сервером.
<b>КонтрольКолва</b>	Int32	дополнительное	Контроль количества по заявке. 1 – контролируется, в документ нельзя принимать товары, которых нет в заявке, а также нельзя превышать количество товара, который в заявке есть 0 – без контроля.
<b>НомерЕгаис</b>	string	дополнительное	Номер ТТН ЕГАИС.

## Реквизиты «строк» документа

Реквизит	Тип	Основное/доп.	Описание
CreatedBy	CreateBy	основное	Проверка упаковки.
ProductID	string	основное	Идентификатор товара, для которого описана данная позиция.
DeclaredQuantity	Double	основное	Заданное количество в позиции. Свойство заполняется при выгрузке документа из учетной системы, чтобы ограничить объем товара данного типа, с которым оперирует пользователь в документе.
CurrentQuantity	Double	основное	Текущее количество товара в позиции. Показывает какое реальное количество товара задействовано пользователем.
FirstCellID	string	основное	
FirstStorageBarcode	string	основное	Штрихкод первого места хранения к которой привязан товар. В качестве места может выступать ячейка или паллета.
PackingID	string	основное	Идентификатор упаковки для заданного товара. Товар задается свойством ProductId [ИдТовара].
SSCC	string	основное	Уникальный номер единицы хранения. Служит для идентификации конкретных экземпляров товара. Заполняется из штрихкода товара по шаблону при выполнении операции.
ExpiredDate	DateTime	основное	Срок годности товара. Инициализируется при занесении конкретного товара в позицию документа пользователем либо из штрихкода по шаблону, либо ручным выбором.
SecondCellID	string		
SecondStorageBarcode	string	основное	Штрихкод второго места хранения к которой привязан товар. В качестве места может выступать ячейка или паллета.
BindedLine	DocumentItem	основное	Задаёт связь строки из CurrentItems [СтрокиФакт] со строкой в DeclaredItems [СтрокиПлан].
Barcode	string	дополнительное	Штрихкод.
Serial	string	дополнительное	Серия товара (если используется учет по сериям).
Sn	string	дополнительное	Серийный номер.
Price	Decimal	дополнительное	Цена единицы товара в строке.
ЦенаСклад	string	дополнительное	
Алко	boolean	дополнительное	Признак того, что товар является алкогольной или спиртосодержащей продукцией.
ПроверкаЧМ	boolean	дополнительное	Нужно проверять Check Mark или нет.
АлкоСН	string	дополнительное	Серийный номер бутылки.
АлкоКод	string	дополнительное	Строка с кодом алкогольной

			продукции в ЕГАИС.
<b>АлкоНаим</b>	string	дополнительное	Наименование товара.
<b>АлкоКодВ</b>	string	дополнительное	Строка с кодом вида алкогольной продукции.
<b>АлкоОбъем</b>	Decimal	дополнительное	Ёмкость тары в литрах. (в базе может быть в декалитрах, например, в Рознице).
<b>АлкоКрепость</b>	Decimal	дополнительное	Процентное содержание спирта.
<b>Производитель</b>	string	дополнительное	Строка с наименованием производителя (уже есть реализация в мобильной печати). Для алкоголя должно выгружаться вне зависимости от того, включена ли мобильная печать и стоит ли там галочка.
<b>ПроизводительИНН</b>	string	дополнительное	ИНН производителя, если уже известен для данного товара.
<b>ПроизводительКПП</b>	string	дополнительное	КПП производителя, если уже известен для данного товара.
<b>АлкоПДФ</b>	string	дополнительное	Строка с PDF 417.

**Реквизиты строка дополнительной табличной части документа «ФормыА»**

Реквизит	Тип	Основное/доп.	Описание
<b>КодФормы</b>	string	дополнительное	Код формы в системе ЕГАИС.
<b>КодНоменклатуры</b>	string	дополнительное	Уникальный идентификатор номенклатуры из справочника номенклатуры, если есть.
<b>АлкоКод</b>	string	дополнительное	Строка с кодом алкогольной продукции в ЕГАИС.
<b>Начало</b>	int32	дополнительное	Начало диапазона серийных номеров.
<b>Конец</b>	int32	дополнительное	Конец диапазона серийных номеров.

Основной метод записи в поля документ – это команда «SetField» («УстановитьПоле» - русск)

Соответственно, метод получения данных из полей – «GetField» («ПолучитьПоле» - русск)

**ИМЕНА ПОЛЕЙ ЧУВСТВИТЕЛЬНЫ К РЕГИСТРУ!**

Давайте попробуем создать документ «Сбор начальных остатков»

Перед началом выгрузки документа, мы должны определить его тип (в какой документ Mobile SMARTS мы будем выгружать документ 1С).

```
ДокументТсД = Новый СОМОбъект("Cleverence.Warehouse.Document");
ДокументТсД.Ид - uuid Документа = XmlСтрокой(СсылкаНаНашДокумент);
ДокументТсД.ИмяТипаДокумента = «Сбор начальных остатков»;
```

Таким образом мы создали новый документ. После его создания, необходимо явным образом объявить создание строк документа.

Строки документа создаются следующим образом:

```
СтрокаДокументаТСД = Новый СОМОбъект("Cleverence.Warehouse.DocumentItem");
//
// тут происходит заполнение полей объекта DocumentItem
СтрокаДокументаТСД.КоличествоПлан = 100
// Например, вышеприведённый код выгрузит 1 строку, где поле DeclaredQuantity будет равно
100
ДокументТСД.СтрокиПлан.Добавить(СтрокаДокументаТСД);
// После заполнения строк документа, нам необходимо зафиксировать сам объект:
Ответ = мДрайверТСД.ВыгрузитьДокумент(ДокументТСД);
Если Не Ответ Тогда
    НомерОшибки = мОбъектТСД.ПолучитьОшибку(мОбъектТСД.ОписаниеОшибки);
    Сообщить("Ошибка: " + мОбъектТСД.ОписаниеОшибки, СтатусСообщения.Важное);
Иначе
    Сообщить("Документ "" + Строка(Документ1С) + "" выгружен на ТСД.",
СтатусСообщения.Обычное);
КонецЕсли;
```

В наших конфигурациях обычно используется следующий алгоритм выгрузки строк документа: (метод «установитьполе» равен в нашем случае команде «SetField»).

```
Для каждого строкаРекв из СтрокаФормата.Реквизиты Цикл // Тут у нас определена сущность
«строка формата», которая как раз и определяет, какие реквизиты документа 1С мы передаём в
реквизиты MS
Если Не строкаРекв.Выгружать Тогда
    Продолжить; // не выгружаем ненужные реквизиты
КонецЕсли;

значение = РекДок[строкаРекв.Реквизит];
Если не ТипЗначенияДоступенВСМАРТС(значение) Тогда
    значение = Строка(значение);
    значение = ?(значение = "<>", "", значение);
КонецЕсли;
Если ЗначениеЗаполнено(значение) Тогда
    ДокументТСД.УстановитьПоле(строкаРекв.ПолеДокументаТСД, значение);
    // Метод УстановитьПоле («НаименованиеПоляДокумента», «ПередаваемоеЗначение»)
КонецЕсли;
КонецЦикла;
```

После выгрузки шапки документа (заголовок, организация, код ЕГАИС организации, и т.д.), мы можем приступить к выгрузке непосредственно строк документа. Необходимо уточнить, что структура базы данных Mobile SMARTS построена следующим образом – существуют «плановые» строки, которые выгружаются непосредственно из 1С, а после считывания штрих-кода с товара, в документ Mobile SMARTS добавляются строки «факта», которые затем мы с вами и будем обрабатывать.

Выгружаем плановые строки:

```
Для каждого строкаДок из МассивСтрок Цикл
    Товар = Неопределено;
    СтрокаДокументаТСД = Новый СОМОбъект("Cleverence.Warehouse.DocumentItem");
КонецЦикла;
```

Если есть возможность выгрузить диапазоны справок А, то используется следующий метод:

```

ТаблицаСправкаА = ДокументТСД.Таблицы.ДобавитьЭлемент ();
ТаблицаСправкаА.Name = "ФормыА";
мДрайверТСД.ВыгрузитьДопТаблицу (ТаблицаСправкаА);
Для Каждого СтрокаСправок из Документ1С.Товары Цикл
НоваяСтрокаТаблицы = ТаблицаСправкаА.Строки.ДобавитьЭлемент ();
НоваяСтрокаТаблицы.SetField ("КодФормы", СтрокаСправок.СправкаА.Код);
НоваяСтрокаТаблицы.SetField ("Начало", СтрокаСправок.СправкаА.ДиапазоныНомеров [0].НачальныйНомер);
НоваяСтрокаТаблицы.SetField ("Конец", СтрокаСправок.СправкаА.ДиапазоныНомеров [0].КонечныйНомер);
НоваяСтрокаТаблицы.SetField ("АлкоКод", СтрокаСправок.АлкоГольнаяПродукция.Код);
НоваяСтрокаТаблицы.SetField ("КодНоменклатуры", ОпределитьНоменклатуруНаСервере (СтрокаСправок.АлкоГольнаяПродукция));
КонецЦикла;
    
```

Поле	Тип	Основное/дополнительное	Описание
<b>КодФормы</b>	String	дополнительное	Код формы в системе ЕГАИС.
<b>КодНоменклатуры</b>	String	дополнительное	Уникальный идентификатор номенклатуры из справочника номенклатуры, если есть.
<b>АлкоКод</b>	String	дополнительное	Строка с кодом алкогольной продукции в ЕГАИС.
<b>Начало</b>	Int32	дополнительное	Начало диапазона серийных номеров.
<b>Конец</b>	Int32	дополнительное	Конец диапазона серийных номеров.

После того, как мы закончили выгружать документ, используется следующая конструкция:

```

Ответ = мОбъектТСД.ВыгрузитьДокумент (ДокументТСД);
Если Не Ответ Тогда
НомерОшибки = мОбъектТСД.ПолучитьОшибку (мОбъектТСД.ОписаниеОшибки);
Сообщить ("Ошибка: " + мОбъектТСД.ОписаниеОшибки, СтатусСообщения.Важное);
Иначе
Сообщить ("Документ "" + Строка (Документ1С) + "" выгружен на ТСД.",
СтатусСообщения.Обычное);
КонецЕсли;
    
```

Таким образом мы передали информацию о том, что документ выгружен, либо, если произошла ошибка, получим её описание

## Этап третий. Загрузка документов

Загрузка документов аналогична выгрузке, исключение составляет лишь несколько моментов:

1. Для чтения полей читаем строки факта.

```
Для Каждого СтрокаТаблицыДокумента из ДокументТСД["СтрокиФакт"]
```

2. Поля читаем командой «GetField», либо «ПолучитьПоле».
3. После того, как мы загрузили документ, обработали его определенным образом, необходимо выполнить команду очистки списка документов.

```
мОбъектТСД.Драйвер.УдалитьДокументы («Список документов»);
```

**Пример Б – загрузка с ТСД документа «Ввод начальных остатков»**

```
// Демонстрационный пример для формирования документа "сбор начальных остатков"
// Создаём таблицу значений, и загружаем в неё данные, полученные с терминала
// Тут мы можем перебрать все документы, которые были выгружены на терминал.
// Вниманию - у документов есть несколько статусов, доступных через свойства
// ДокументыТерминалов.Изменен - boolean; // ДокументыТерминалов.Завершен - boolean;
// Если для вас необходимо собирать только завершённые документы, пользуйтесь значениями
этих полей.
// После чего можем сохранить документы // Единственное, что необходимо типизировать
таблицу значений перед тем, как загружать в неё данные
ТаблицаДанныхСТерминала = Новый ТаблицаЗначений;
ТаблицаДанныхСТерминала.Колонки.Добавить ("CreatedBy");
ТаблицаДанныхСТерминала.Колонки.Добавить ("ProductID");
ТаблицаДанныхСТерминала.Колонки.Добавить ("DeclaredQuantity");
ТаблицаДанныхСТерминала.Колонки.Добавить ("CurrentQuantity");
ТаблицаДанныхСТерминала.Колонки.Добавить ("FirstCellID");
ТаблицаДанныхСТерминала.Колонки.Добавить ("FirstStorageBarcode");
ТаблицаДанныхСТерминала.Колонки.Добавить ("PackingID");
ТаблицаДанныхСТерминала.Колонки.Добавить ("SSCC");
ТаблицаДанныхСТерминала.Колонки.Добавить ("Index");
ТаблицаДанныхСТерминала.Колонки.Добавить ("RegisteredDate");
ТаблицаДанныхСТерминала.Колонки.Добавить ("RegistrationDate");
ТаблицаДанныхСТерминала.Колонки.Добавить ("ExpiredDate");
ТаблицаДанныхСТерминала.Колонки.Добавить ("SecondCellID");
ТаблицаДанныхСТерминала.Колонки.Добавить ("SecondStorageBarCode");
ТаблицаДанныхСТерминала.Колонки.Добавить ("BindedLine");
ТаблицаДанныхСТерминала.Колонки.Добавить ("code");
ТаблицаДанныхСТерминала.Колонки.Добавить ("barcode");
ТаблицаДанныхСТерминала.Колонки.Добавить ("serial");
ТаблицаДанныхСТерминала.Колонки.Добавить ("desc");
ТаблицаДанныхСТерминала.Колонки.Добавить ("sn");
ТаблицаДанныхСТерминала.Колонки.Добавить ("price");
ТаблицаДанныхСТерминала.Колонки.Добавить ("Ячейка");
ТаблицаДанныхСТерминала.Колонки.Добавить ("Заблокировано");
ТаблицаДанныхСТерминала.Колонки.Добавить ("ЦенаСклад");
ТаблицаДанныхСТерминала.Колонки.Добавить ("Алко");
ТаблицаДанныхСТерминала.Колонки.Добавить ("ПроверкаЧМ");
ТаблицаДанныхСТерминала.Колонки.Добавить ("АлкоСН");
ТаблицаДанныхСТерминала.Колонки.Добавить ("АлкоКод");
ТаблицаДанныхСТерминала.Колонки.Добавить ("АлкоНаим");
ТаблицаДанныхСТерминала.Колонки.Добавить ("АлкоКодВ");
ТаблицаДанныхСТерминала.Колонки.Добавить ("АлкоОбъем");
ТаблицаДанныхСТерминала.Колонки.Добавить ("АлкоКрепость");
ТаблицаДанныхСТерминала.Колонки.Добавить ("Производитель");
ТаблицаДанныхСТерминала.Колонки.Добавить ("ПроизвИНН");
ТаблицаДанныхСТерминала.Колонки.Добавить ("ПроизвКПП");
ТаблицаДанныхСТерминала.Колонки.Добавить ("АлкоПДФ");
Для инд = 0 По ДокументыТерминалов.Количество - 1 Цикл
ДокументТерминала = ДокументыТерминалов.Элемент (инд);
// Все документы сбора начальных остатков мы можем собрать в одну ТЗ
Если ДокументыТерминалов.Элемент (инд).ИмяТипаДокумента = "Сбор начальных остатков" Тогда
    СтрокиДокументаПлан = ДокументТерминала.СтрокиПлан;
    КоличествоСтрокПлан = СтрокиДокументаПлан.Количество;
    СтрокиДокументаФакт = ДокументТерминала.СтрокиФакт;
    КоличествоСтрокФакт = СтрокиДокументаФакт.Количество;
    Для СтрокаФакт = 0 по СтрокиДокументаФакт.Количество - 1 Цикл
        ТекущаяСтрокаДокумента = СтрокиДокументаФакт.Элемент (СтрокаФакт);
        СтрокаТЗ = ТаблицаДанныхСТерминала.Добавить ();
        Для Каждого Колонка из ТаблицаДанныхСТерминала.Колонки Цикл
            СтрокаТЗ[Колонка.Имя] = ТекущаяСтрокаДокумента.ПолучитьПоле (Колонка.Имя);
        КонецЦикла;
    КонецЦикла;
КонецЕсли;
КонецЦикла;
// Всё, таблица готова для дальнейшей обработки в 1С
```

## Поля строк документа для получения из фактических строк

Поле	Тип	Основное/доп.	Описание
ProductID	String	основное	Идентификатор товара, для которого описана данная позиция.
PackingID	String	основное	Идентификатор упаковки для заданного товара. Товар задается свойством ProductId [ИдТовара].
CurrentQuantity	Double	основное	Фактическое количество товара в заданном виде упаковки.
descr	String	дополнительное	Характеристика товара (если ведется учет с характеристиками).
serial	String	дополнительное	Серия товара (если используется учет по сериям).
price	Decimal	дополнительное	Цена единицы товара в строке.
Алко	Boolean	дополнительное	Признак того, что товар является алкогольной или спиртосодержащей продукцией.
АлкоКод	String	дополнительное	Код алкогольной продукции в ЕГАИС, полученный из отсканированных марок.
АлкоПДФ	String	дополнительное	Строка с PDF 417.
АлкоСН	string	дополнительное	Серийный номер бутылки, если сканировался.
ПроверкаЧМ	Boolean	дополнительное	Признак того, что <b>данная конкретная марка проверялась в CheckMark на легальность.</b>
Производитель	String	дополнительное	Наименование производителя продукции в ЕГАИС, если уже известно для данного товара, <b>либо если было получено из CheckMark.</b>
ПроизводительИНН	String	дополнительное	ИНН производителя, если уже известен для данного товара, <b>либо если был получен из CheckMark.</b>
ПроизводительКПП	String	дополнительное	КПП производителя, если уже известен для данного товара, <b>либо если был получен из CheckMark.</b>
АлкоНаим	String	дополнительное	Наименование товара из ЕГАИС, если уже известно для данного товара, <b>либо если было получено из CheckMark.</b>
АлкоКодВ	String	дополнительное	Код вида алкогольной продукции из ЕГАИС, если уже известен для данного товара, <b>либо если был получен из CheckMark.</b>
АлкоОбъем	Decimal	дополнительное	Ёмкость тары в литрах из ЕГАИС, если уже известна для данного товара, <b>либо если была получена из CheckMark.</b>
АлкоКрепость	Decimal	дополнительное	Процентное содержание спирта из ЕГАИС, если уже известно для данного товара, <b>либо если было получено из CheckMark.</b>
АлкоЕстьВФормах	Boolean	дополнительное	Признак того, что данная марка была найдена в выгруженных формах А.
КодФормыА	String	дополнительное	Код формы А, где была найдена марка.
ДатаРозлива	DateTime	дополнительное	Дата розлива позиции, если вводилась, иначе null.

## Контакты

Все права на программное обеспечение Mobile SMARTS принадлежат ООО «Клеверенс Софт». По вопросам поддержки обращайтесь по указанным реквизитам компании:

Глоссарий:

Наименование компоненты:

Cleverence Soft,

email: [support@cleverence.ru](mailto:support@cleverence.ru)

[www.cleverence.ru](http://www.cleverence.ru)