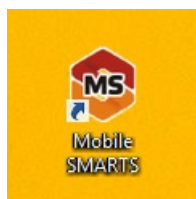


# Интеграция ЕГАИС с произвольной конфигурацией «1С: Предприятия»

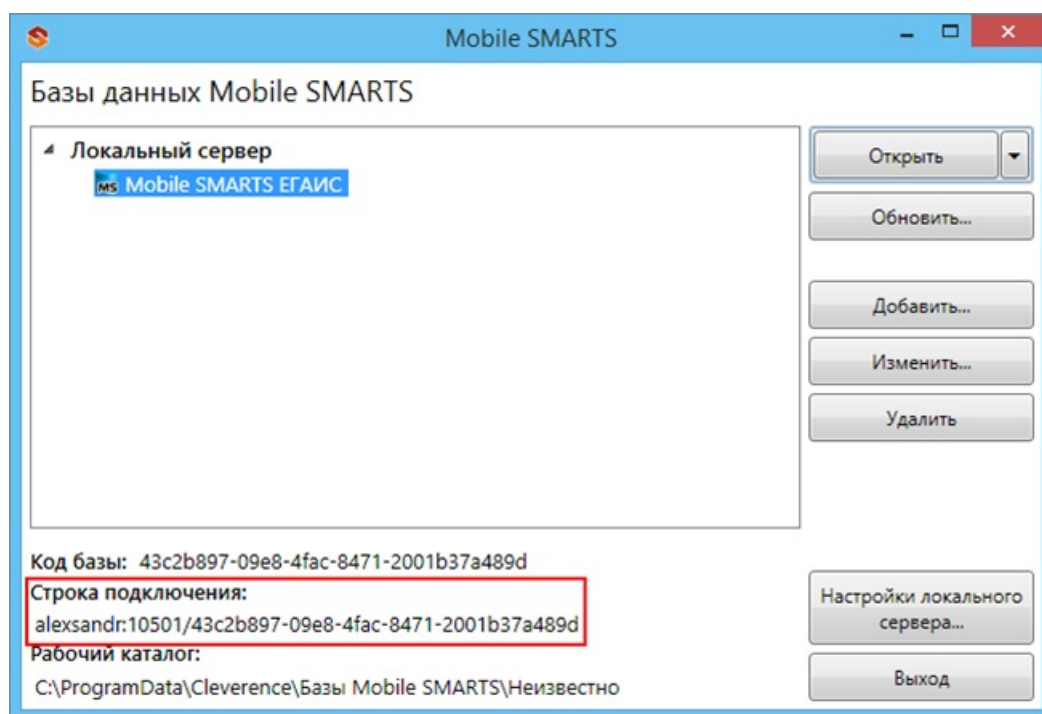
- ☐ [Скачать инструкцию по интеграции базы MSv8](#) (для 1С:Предприятия 8);
- ☐ [Скачать инструкцию по интеграции базы MSv7](#) (для 1С:Предприятия 7.7);
- ☐ [Скачать инструкцию по интеграции «Mobile SMARTS: ЕГАИС» с произвольной системой через COM](#)

## 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 **менеджер баз данных Mobile SMARTS**, мы можем проверить адрес подключения и рабочий каталог, по которому расположена база «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 «ЕГАИС».

Вызываем компоненту и подключаем необходимую базу:

Пример кода 1С:Предприятие 8

```
//мДрайверТСД = новый СОМОбъект("Cleverence.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
```

#### Пример кода 1С:Предприятие 7.7

```
Компонента = "AddIn.Cl.TerminalConnector";
Если ПодключитьВнешнююКомпоненту(Компонента) = 0 Тогда
    ОписаниеРезультата = "ошибка загрузки внешней компоненты "" + Компонента + """";
Иначе
    мДрайверТСД = СоздатьОбъект(Компонента);
    мДрайверТСД.УстановитьВерсию1С("v7");
    СтрокаПодключения = "dt-501-2:10501/6d7e3a2f-3967-4734-bb9c-d1ba605aaef8";
    Сз = СоздатьОбъект("СписокЗначений");
    Сз.ДобавитьЗначение(СтрокаПодключения);
    Если мДрайверТСД.Подключить(Сз, "") = 0 Тогда
        Описание = "";
        мДрайверТСД.ПолучитьОшибку(Описание);
        Сообщить("Ошибка при подключении: " + Описание);
        мДрайверТСД.ОсвободитьРесурсы();
        Возврат; КонецЕсли;
    КонецЕсли;
    Описание = "";
```

#### Пример кода для произвольной учетной системы через компоненту COM

```
connection = new COM("Cleverence.Warehouse.StorageConnector");
// СтрокаПодключения - строка подключения из настройки базы MS
connection.SelectCurrentApp(СтрокаПодключения);
```

## Обмен данными

Разделяем обмен данными на четыре этапа внутри

- Этап 1 – выгрузка номенклатуры;
- Этап 2 – выгрузка документов;
- Этап 3 – выгрузка данных форм «А»;
- Этап 4 – загрузка документов.

### Этап первый – выгрузка номенклатуры

Перед выгрузкой номенклатуры мы должны её инициализировать:

#### Пример кода 1С:Предприятие 8

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

#### Пример кода 1С:Предприятие 7.7

```
Если мОбъектТСД.НачатьВыгрузкуТоваров(мДанные) = 0 Тогда
    мОбъектТСД.ПолучитьОшибку(мОбъектТСД.ОписаниеОшибки);
    ВывестиПредупреждение( "Ошибка при инициализации выгрузки: " + мОбъектТСД.ОписаниеОшибки );
    Сообщить("Ошибка при инициализации выгрузки: " + мОбъектТСД.ОписаниеОшибки);
    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. Сначала формируем массив строк в 1С для выгрузки, затем циклом последовательно его обходим – по каждой позиции.

#### мДанные – строка выгрузки. Пример кода 1С:Предприятие 8

```
// +++ 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.ПолучитьОшибку("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, "000000000000");/"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, "PyOpt");//"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.ПолучитьОшибку("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.ПолучитьОшибку("Номер ошибки");
```

```
Сообщить("Ошибка при завершении выгрузки: " + НомерОшибки);
```

```
Прервано = Истина;
```

```
КонецЕсли;
```

#### **мДанные – строка выгрузки. Пример кода 1С:Предприятие 7.7**

```
// +++ 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 = СоздатьОбъект("СписокЗначений");
мДанные.ДобавитьЗначение("Product.Ид" );
мДанные.ДобавитьЗначение("Product.Marking" );
...
...
мДанные.ДобавитьЗначение(24, " Packing.АлкоКод" );
Если Не мДрайверТСД.НачатьВыгрузкуТоваров(мДанные) Тогда // Происходит проверка на соответствие структуры полей
  Ошибка = мДрайверТСД.ПолучитьОшибку("Описание ошибки");
  Сообщить("Ошибка при инициализации выгрузки: " + Ошибка, СтатусСообщения.Важное );
  мДрайверТСД.ОсвободитьРесурсы();
  Возврат;
КонецЕсли;
Спр = СоздатьОбъект("Справочник.Номенклатура");
Спр.ВыбратьЭлементы();
Пока Спр.ПолучитьЭлемент() = 1 Цикл
  мДанные.УдалитьВсе();
  мДанные.ДобавитьЗначение(Спр.Артикул);//"Product.Ид" ); // uuid элемента справочника
  мДанные.ДобавитьЗначение(Спр.Артикул);//"Product.Marking" ); // Артикул
  мДанные.ДобавитьЗначение(Спр.Код);//"Product.Barcode" ); // Код товара
  мДанные.ДобавитьЗначение("00000000000000");//"Packing.Barcode" ); // ШК
  мДанные.ДобавитьЗначение(Спр.Наименование);//"Product.Name" ); // Наименование
  мДанные.ДобавитьЗначение(Спр.Артикул);//"Product.BasePackingId" ); //Является базовой единицей
  мДанные.ДобавитьЗначение(Спр.Артикул);//"Packing.Ид" ); // Единица измерения
  мДанные.ДобавитьЗначение(Спр.Наименование);//"Packing.Name" ); // Наименование единицы измерения
  мДанные.ДобавитьЗначение(1);//"Packing.UnitsQuantity" ); // коэффициент (кратность) единицы измерения
  мДанные.ДобавитьЗначение("");//"Packing.descr" ); // характеристика единицы измерения
  мДанные.ДобавитьЗначение("");//"Packing.serial" ); // серийный номер единицы измерения
  мДанные.ДобавитьЗначение(0);//"Packing.price" ); // стоимость
  мДанные.ДобавитьЗначение(1);//"Packing.qty" ); //Количество
  мДанные.ДобавитьЗначение(0);//"Product.withserial" ); // Используются серии
```

```
мДанные.ДобавитьЗначение(1);/" Packing.Алко" ); // Является алкоголе содержащей продукцией (bool)
мДанные.ДобавитьЗначение("лицензия на продажу спирта");/" Packing.АлкоВидЛиц" ); // Вид лицензии
мДанные.ДобавитьЗначение(1);/" Packing.АлкоМарк" ); // Является маркируемой продукцией
мДанные.ДобавитьЗначение("420");/" Packing.АлкоКодВ" ); // Код вида алкогольной продукции
мДанные.ДобавитьЗначение("СправочникНоменклатуры.Наименование");/"Product.АлкоНаимВ" ); // Наименование вида
алкогольной продукции
мДанные.ДобавитьЗначение(1);/" Packing.АлкоОбъем" ); // Объем
мДанные.ДобавитьЗначение(25);/" Packing.АлкоКрепость" ); // Крепость
мДанные.ДобавитьЗначение("РуОпт");/" Packing.Производитель" ); // производитель
мДанные.ДобавитьЗначение("7774444454");/" Packing.ПроизвИНН" ); // ИНН Производителя
мДанные.ДобавитьЗначение("7707707700");/" Packing.ПроизвКПП" ); // КПП Производителя
мДанные.ДобавитьЗначение("22550");/" Packing.АлкоКод" ); // Код алкогольной продукции Если
мДрайверТСД.ДобавитьВВыгрузкуТоваров(мДанные) = 0 Тогда
```

```
Ошибка = мДрайверТСД.ПолучитьОшибка("Описание ошибки");
```

```
Сообщить( "Ошибка при выгрузке данных: " + " " + Ошибка, СтатусСообщения.Важное );
```

```
Прервано = 1;
```

```
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.

Для выгрузки номенклатуры используется три функции компоненты подключения:

#### Начать выгрузку товаров

```
BeginUploadProducts(bool anyway, bool overwriteExisting, bool generateFullTextSearch)
```

Параметр
----------

Тип
-----

Описание
----------

anyway
--------

bool
------

Начать новую выгрузку, даже если какая-то другая выгрузка уже была открыта.
---

В случае true – позволяет начинать сразу несколько выгрузок одновременно.
---

overwriteExisting

bool

Флаг, определяющий полностью перезаписывать весь справочник номенклатуры или слить к выгруженным ранее.

generateFullTextSearch

bool

Флаг, определяющий следует ли генерировать индексы для поиска товаров по части имени. Если такой функционал не требуется, то лучше использовать false так как такой индекс значительно увеличивает объем данных, загружаемых на ТСД.

## Выгрузка товаров

UploadProducts(**ProductCollection** products)

Параметр

Тип

Описание

products

**ProductCollection**

Коллекция товаров для выгрузки.

Функция принимает для выгрузки коллекцию товаров (**Product**).

Структура данных Mobile SMARTS требует, чтобы каждый товар содержал хотя бы одну упаковку (**Packing**).

Пример кода для произвольной учетной системы через компоненту COM

```
// создание коллекции
productsColl = new COM("Cleverence.Warehouse.ProductCollection");
// создание товара
product = new COM("Cleverence.Warehouse.Product");
product.SetField("Id", уникИдТовара);
product.SetField("Name", "Товар1");
//создание упаковки
packing = new COM("Cleverence.Warehouse.Product");
packing.SetField("Id", идУпаковки); //уникальный в пределах товара
packing.SetField("Name", "шт");
product.Packings.Add(packing); //добавление упаковки в товар
//обязательно – установка кода базовой упаковки для товара
product.SetField("BasePackingId", packing.Id);
productsColl.Add(product); //добавление товара в коллекцию

//выгрузка
connection.UploadProducts(productsColl);
```

Список полей для заполнения в товаре:

Поле

Тип

Основное/дополнительное

Описание

**Id**

string

основное

Уникальный идентификатор

**Name<sup>3</sup>**

string

основное

Наименование товара.

**BasePackingId**

string

основное

Уникальный идентификатор

**Marking** упаковки товара.

string

основное

Артикул.

**withserial**

int

дополнительное

Ведется ли серийный учет по

**АлкоУ** (1 – ведется, 0 – нет).

bool

дополнительное

Является алкоголе

**АлкоКод** <sup>1</sup>ей продукцией.

string

дополнительное

Код алкогольной продукции.

**АлкоВидЛиц**

string

дополнительное

Вид лицензии (просто

**АлкоМарк** <sup>онное</sup> поле).

bool

дополнительное

Является маркируемой

**АлкоКодВ**<sup>1</sup>.

string

дополнительное

Код вида алкогольной

**АлкоНаимВ**

string

дополнительное

Наименование вида

**АлкоОбъем** продукции.

double

дополнительное

Объем.

**АлкоКрепость**

double

дополнительное

Крепость.

Производитель	string	дополнительное
Производитель.		
ПроизвИНН	string	дополнительное
ИНН Производителя.		
ПроизвКПП	string	дополнительное
КПП Производителя.		

Список полей для заполнения в упаковке:

Поле	Тип	Основное/дополнительное
Описание		
Id	string	основное
Уникальный идентификатор		
Name <sup>2</sup>	string	основное
Наименование упаковки.		
UnitsQuantity	double	основное
Коэффициент (кратность)		
Barcode измерения.	string	основное
Штрихкод товара (обычно price <sup>3</sup> ) в текущей упаковке.	double	дополнительное
Стоимость.		
qty	double	дополнительное
Количество.		
serial	string	дополнительное
Серия, привязанная к desc <sup>1</sup> (для серийного учета)	string	дополнительное
Характеристика товара, привязанная к упаковке		

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

Пример кода 1С:Предприятие 8



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

#### Пример кода 1С:Предприятие 7.7

```
// Выгрузка
Если мОбъектТсД.ДобавитьВВыгрузкуТоваров(мДанные) = 0 Тогда
мОбъектТсД.ПолучитьОшибку(мОбъектТсД.ОписаниеОшибки);
Сообщить( "Ошибка при выгрузке данных: " + ". " + мОбъектТсД.ОписаниеОшибки,
СтатусСообщения.Важное );
Прервано = 1;
Прервать;
КонецЕсли;
```

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

#### Пример кода 1С:Предприятие 8

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

#### Пример кода 1С:Предприятие 7.7

```
Если мОбъектТсД.ЗавершитьВыгрузкуТоваров() = 0 Тогда
НомерОшибки = мОбъектТсД.ПолучитьОшибку(мОбъектТсД.ОписаниеОшибки);
Сообщить("Ошибка при завершении выгрузки: " + мОбъектТсД.ОписаниеОшибки);
Прервано = 1;
КонецЕсли;
```

Заполнение/получение основных полей можно производить как напрямую, так и через функции SetField/GetField. Для дополнительных полей можно использовать только GetField/SetField.

**ИМЕНА ПОЛЕЙ ЧУВСТВИТЕЛЬНЫ К РЕГИСТРУ!**

#### Пример кода для произвольной учетной системы через компоненту COM

```
product.Id = "00001";
//или
product.SetField("Id", "00001");
//доп поле
product.SetField("Алко", true);
```

#### Завершение выгрузки товаров

##### EndUploadProducts()

Завершает процедуру выгрузки товаров. После её вызова сервер будет считать выгрузку завершённой и начнет у себя обновление справочников, генерацию индексов.

До вызова этой функции товары не считаются выгруженными и недоступны.

#### Полный цикл Пример кода для произвольной учетной системы через компоненту COM

```

connection = new COM("Cleverence.Warehouse.StorageConnector");
// СтрокаПодключения - строка подключения из настройки базы MS
connection.SelectCurrentApp(СтрокаПодключения);
//начало выгрузки
//вне зависимости от прошлых выгрузок, с полным переписыванием на сервере, без индекса по //именам товаров
Connection.BeginUploadProducts(true, true, false);
productsColl = new COM("Cleverence.Warehouse.ProductCollection");
for(int i = 0; i < колво_товаров_в_системе; i++) //цикл по товарам в системе
{
//выгружаем блоками по 500 товаров
if(productsColl.Count == 500)
{
//выгрузка блока из 500 товаров
connection.UploadProducts(productsColl);
productsColl = new COM("Cleverence.Warehouse.ProductCollection");
}
// создание товара
product = new COM("Cleverence.Warehouse.Product");
product.SetField("Id", "ид" + i);
product.SetField("Name", "Товар " + i);
//создание упаковки
packing = new COM("Cleverence.Warehouse.Product");
packing.SetField("Id", идУпаковки); //уникальный в пределах товара
packing.SetField("Name", "шт");
...
//заполнение остальных полей упаковки
...
product.Packings.Add(packing); //добавление упаковки в товар
//обязательно – установка кода базовой упаковки для товара
product.SetField("BasePackingId", packing.Id);
productsColl.Add(product); //добавление товара в коллекцию
}
//выгрузка оставшихся товаров
if (productsColl.Count > 0)
connection.UploadProducts(productsColl);
//завершение выгрузки
connection.EndUploadProducts();

```

## 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 по умолчанию доступные 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:

На сервере Mobile SMARTS:ЕГАИС по умолчанию есть следующие типы документов, доступные для выгрузки:

- Приход на склад;
- Подбор заказа;

- Инвентаризация.

Применительно к алкогольной тематике, мы можем проставить следующее соответствие:

- приход на склад = ТТН ЕГАИС в статусе «от поставщика»;
- подбор заказа = ТТН ЕГАИС в статусе «покупателю».

### Функция выгрузки

UploadDocument(Document document)

Параметр
Тип
Описание
document
<b>Document</b>
Документ Mobile SMARTS для выгрузки.

Функция принимает для выгрузки объект документа (**Document**).

### Пример кода для произвольной учетной системы через компоненту COM

```
// Аналогично работе с номенклатурой, инициализируем подключение
connection = new COM("Cleverence.Warehouse.StorageConnector");
// СтрокаПодключения - строка подключения из настройки базы MS
connection.SelectCurrentApp(СтрокаПодключения);
// создание документа
document = new COM("Cleverence.Warehouse.Document");
//заполнение документа
//...
//...
// выгрузка в Mobile SMARTS
connection.UploadDocument(document);
```

Каждый документ Mobile SMARTS имеет шапочную часть документа, и две табличных части (DeclaredItems - строки заявки, CurrentItems – фактические строки с ТСД).

При выгрузке документа нам необходимо заполнить нужные поля шапки и строки заявки.

### Заполнение шапки документа:

#### Реквизиты «шапки» документа

Реквизит
Тип
Основное/доп.
Описание
Id
string
основное
Идентификатор документа. Обязательно заполняется при выгрузке документа и должен иметь уникальное значение.

**Name**

string

основное

Отображаемое имя документа.  
Обязательно заполняется при  
выгрузке документа.

**Appointment**

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.

**UserID**

string

основное

Идентификатор пользователя,  
который выполнил документ.  
Должен иметь уникальное  
значение для каждого  
пользователя.

**UserName**

string

основное

Имя пользователя.

**DocumentTypeName**

string

основное

Имя типа документа.  
Обязательно заполняется при  
выгрузке документа и должно  
иметь уникальное значение.

**Modified**

boolean

основное

Признак того, что документ был  
изменен. Заполняется при  
изменении документа одним из  
пользователей.

### InProgress

boolean

основное

Признак того, что документ захвачен пользователем на обработку. Заполняется, когда документ был захвачен пользователем.

### Finished

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.

### WareHouseID

string

основное

Идентификатор склада, к которому привязан документ. Должен иметь уникальное значение для каждого склада.

### Barcode

string

основное

Штрихкод документа. Для выбора документа по штрихкоду на терминале или на сервере должен быть обязательно заполнен.

### Priority

Int32

основное

Приоритет документа. Более приоритетные документы раньше отдаются на терминал для обработки.

### DistributeByBarCode

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.

### ServerHosted

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.

### КонтрольКолва

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.

### НомерЕгаис

string

дополнительное

Номер ТТН ЕГАИС.

Пример кода для произвольной учетной системы через компоненту COM

```

// создание документа
document = new COM("Cleverence.Warehouse.Document");
document.Id = "5BFF9B72-BA92-409D-BA01-CA909C3CB7C5";
document.DocumentTypeName = "Приход на склад";
document.Name = "Приход на склад №000001";
//для пользователя «оператор», заведенного в стандартной поставке
document.Appointment = "оператор";
//для выбора с сервера произвольным пользователем
//document.Appointment = "";
//document.DistibuteByBarcode = true;
//для склада, заведенного в стандартной поставке
document.WarehouseId = "1";
//дополнительные поля, также как и в товаре, заполняются через SetField
document.SetField("КонтрольКолва", 1);
document.SetField("НомерЕгаис", "4576456456456");

```

### Заполнение строк заявки

Каждая строка документа – это объект **DocumentItem**.

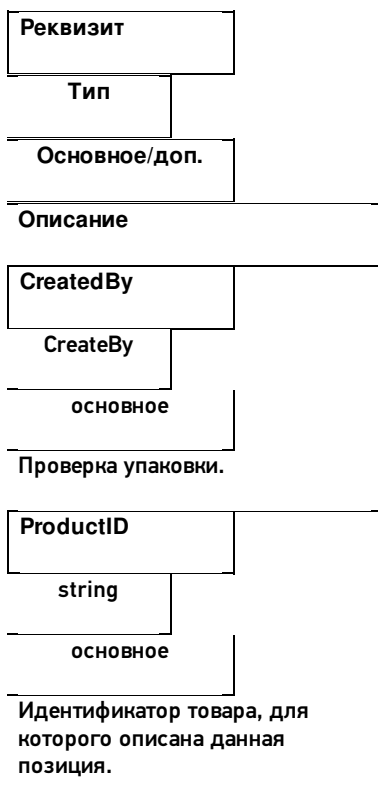
### Пример кода для произвольной учетной системы через компоненту COM

```

// создание строки документа
documentItem = new COM("Cleverence.Warehouse.DocumentItem");

```

### Реквизиты «строк» документа



**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 палета.

**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.

**SecondStorageBarcode**

string

основное

Штрихкод второго места хранения к которой привязан товар. В качестве места может выступать ячейка или палета.

**BindedLine**

DocumentItem

основное

Задаёт связь строки из CurrentItems [СтрокиФакт] со строкой в DeclaredItems [СтрокиПлан].

**Code**

string

дополнительное

**Barcode**

string

дополнительное

Штрихкод.

**Serial**

string

дополнительное

Серия товара (если используется учет по сериям).

**Sn**

string

дополнительное

Серийный номер.

<b>Price</b>
Decimal
дополнительное
Цена единицы товара в строке.
<b>ЦенаСклад</b>
string
дополнительное
<b>Алко</b>
boolean
дополнительное
Признак того, что товар является алкогольной или спиртосодержащей продукцией.
<b>ПроверкаЧМ</b>
boolean
дополнительное
Нужно проверять Check Mark или нет.
<b>АлкоСН</b>
string
дополнительное
Серийный номер бутылки.
<b>АлкоКод</b>
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.

## Этап третий. Выгрузка данных форм «А»

Некоторые операции, например, приход на склад позволяют проводить автоматическую проверку и подбор номеров форм «А», если конечно данные об этих формах выгрузить заранее в Mobile SMARTS.

Формы А выгружаются в виде дополнительной таблицы.

### Формат таблицы форм А

#### Реквизиты строки дополнительной табличной части документа «ФормыА»

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

**Пример кода 1С:Предприятие 8**

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

**Пример кода 1С:Предприятие 7.7**

```
ДокументТсД = СоздатьОбъект("Cleverence.Warehouse.Document");
ДокументТсД.Ид - uuid Документа = ПолучитьUID(СсылкаНаНашДокумент);
ДокументТсД.ИмяТипаДокумента = «Сбор начальных остатков»;
Функция ПолучитьUID(СсылкаНаНашДокумент)
Стр=ЗначениеВСтрокуВнутр(Объект);
СЗ=СоздатьОбъект("СписокЗначений");
СЗ.ИзСтрокиСРазделителями(Сред(Стр,2,СтрДлина(Стр)-2));

Возврат формат(СЗ.ПолучитьЗначение(4),"Ч(0)5")+Прав(СтрЗаменить(Формат("", "С9"), " ", "0") +
СокрЛП(СЗ.ПолучитьЗначение(СЗ.РазмерСписка()), 9);

КонецФункции
// Здесь мы добавили функцию формирования УИД для документа
```

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

Строки документа создаются следующим образом:

**Пример кода 1С:Предприятие 8**

```
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;
```

**Пример кода 1С:Предприятие 7.7**

```

СтрокаДокументаТСД = СоздатьОбъект ("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 Ответ = 0 Тогда
НомерОшибки = мОбъектТСД.ПолучитьОшибку(мОбъектТСД.ОписаниеОшибки);
Сообщить("Ошибка: " + мОбъектТСД.ОписаниеОшибки);
Иначе
Сообщить("Документ "" + Строка(Документ1С) + "" выгружен на ТСД.");
КонецЕсли;

```

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

### **Пример кода 1С:Предприятие 8**

```

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;

```

### **Пример кода 1С:Предприятие 7.7**

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 строкаРекв.Выгружать = 0 Тогда

Продолжить; // не выгружаем ненужные реквизиты

КонецЕсли;

значение = РекДок[строкаРекв.Реквизит];

Если ТипЗначенияДоступенВСМАРТС(значение) = 0 Тогда

значение = Строка(значение);

значение = ?(значение = "<>", "", значение);

КонецЕсли;

Если ПолучитьПустоеЗначение(значение) = 0 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:**

**Пример кода 1С:Предприятие 8**

Для каждого строкаДок из МассивСтрок Цикл

Товар = Неопределено;

СтрокаДокументаТСД = Новый СОМОбъект("Cleverence.Warehouse.DocumentItem");

КонецЦикла;

**Пример кода 1С:Предприятие 7.7**

Для каждого строкаДок из МассивСтрок Цикл

Товар = СтрокаТовара.Номенклатура.Код;

СтрокаДокументаТСД = СоздатьОбъект("Cleverence.Warehouse.DocumentItem");

КонецЦикла;

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

**Пример кода 1С:Предприятие 8**

ТаблицаСправокА = ДокументТСД.Таблицы.ДобавитьЭлемент ();

ТаблицаСправокА.Name = "ФормыА";

мДрайверТСД.ВыгрузитьДопТаблицу(ТаблицаСправокА);

Для Каждого СтрокаСправок из Документ1С.Товары Цикл

НоваяСтрокаТаблицы = ТаблицаСправокА.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.АлкоГольнаяПроду

КонецЦикла;

## Пример кода 1С:Предприятие 7.7

```
ТаблицаСправокА = ДокументТСД.Таблицы.ДобавитьЭлемент ();
ТаблицаСправокА.Name = "ФормыА";
мДрайверТСД.ВыгрузитьДопТаблицу(ТаблицаСправокА);
Для Каждого СтрокаСправок из Документ1С.Товары Цикл
НоваяСтрокаТаблицы = ТаблицаСправокА.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.АлкоГольнаяПроду
КонецЦикла;
```

Поле	Тип
КодФормы	String
КодНоменклатуры	String
АлкоКод	String
Начало	Int32
Конец	Int32

Поэтому, как мы закончили выгружать документ, используется следующая конструкция:

## Пример кода 1С:Предприятие 8

```
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;
```

## Пример кода 1С:Предприятие 7.7



```
Ответ = мОбъектТСД.ВыгрузитьДокумент(ДокументТСД); Если Ответ = 0 Тогда  
НомерОшибки = мОбъектТСД.ПолучитьОшибку(мОбъектТСД.ОписаниеОшибки);  
Сообщить("Ошибка: " + мОбъектТСД.ОписаниеОшибки);  
Иначе  
Сообщить("Документ "" + Строка(Документ1С) + "" выгружен на ТСД);  
КонецЕсли;
```

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

### Начало работы с дополнительной таблицей

Для доступа к доп. таблице необходимо запросить специальный объект для работы с ней.

`GetTableAccessor(string tableName)`

Параметр
Тип
Описание
tableName
String
Имя таблицы.

### Пример кода для произвольной учетной системы через компоненту COM

```
var formTable = connection.GetTableAccessor("ФормыА");
```

Далее работа с таблицей ведется через этот объект. Выгрузка производится аналогично выгрузке номенклатуры.

### Начать выгрузку таблицы

`BeginUpload(bool overwriteAllTable)`

Параметр
Тип
Описание
overwriteAllTable
bool
Флаг, определяющий полностью перезаписывать всю таблицу или слить с выгруженной ранее.

### Пример кода для произвольной учетной системы через компоненту COM

```
var formTable = connection.GetTableAccessor("ФормыА");  
formTable.BeginUpload(true);
```

### Выгрузка строк таблицы

`Upload(RowCollection rows)`

Параметр
Тип
Описание

rows

RowCollection

Коллекция строк таблицы для выгрузки.

Функция принимает для выгрузки коллекцию строк таблицы (**Row**).

**Пример кода для произвольной учетной системы через компоненту COM**

```
// создание коллекции строк таблицы
var rowCol = new COM("Cleverence.Warehouse.RowCollection");
// создание строки таблицы
var row = new COM("Cleverence.Warehouse.Row");
row.SetField("КодФормы", код_формы);
row.SetField("КодНоменклатуры", код_номенклатуры);
row.SetField("АлкоКод", "0345345345435");
row.SetField("Начало", 10234955554);
row.SetField("Конец", 10234955800);
//добавление строки в коллекцию
rowCol.Add(row);
//...
//заполнение строк
//...
//выгрузка коллекции строк
formTable.Upload(rowCol);
```

**Закончить выгрузку таблицы**

EndUpload()

```
formTable.EndUpload();
```

Завершает процедуру выгрузки строк таблицы. После её вызова сервер будет считать выгрузку завершенной и начнет у себя обновление таблицы.

До вызова этой функции новые строки не считаются выгруженными и недоступны.

**Полный цикл Пример кода для произвольной учетной системы через компоненту COM**

```

connection = new COM("Cleverence.Warehouse.StorageConnector");
// СтрокаПодключения - строка подключения из настройки базы MS
connection.SelectCurrentApp(СтрокаПодключения);
var formTable = connection.GetTableAccessor("ФормыА");
//начало выгрузки
//полная выгрузка, с переписыванием всей таблицы на сервере
formTable.BeginUpload(true);
// создание коллекции строк таблицы
var rowCol = new COM("Cleverence.Warehouse.RowCollection");
for(int i = 0; i < колво_форм_А; i++) //цикл по формам А в системе
{
//выгружаем блоками по 500 строк
if(rowCol.Count == 500)
{
//выгрузка блока из 500 товаров
formTable.Upload(rowCol);
rowCol = new COM("Cleverence.Warehouse.RowCollection");
}
// создание строки таблицы
var row = new COM("Cleverence.Warehouse.Row");
row.SetField("КодФормы", код_формы);
row.SetField("КодНоменклатуры", код_номенклатуры);
row.SetField("АлкоКод", алко_код);
row.SetField("Начало", начало_интервала_серийных_номеров);
row.SetField("Конец", конец_интервала_серийных_номеров);
//добавление строки в коллекцию
rowCol.Add(row);
}
//выгрузка оставшихся товаров
if (rowCol.Count > 0)
formTable.Upload(rowCol);
//завершение выгрузки
formTable.EndUpload();

```

## Этап четвертый. 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» (1С:Предприятие 8)

```

// 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.
// Внимание - 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

```

#### Пример Б – загрузка с ТСД документа «Ввод начальных остатков» (1С:Предприятие 7.7)

```

// 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.
// Внимание - 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 = СоздатьОбъект("ТаблицаЗначений");
ТаблицаДанныхСТерминала.НоваяКолонка("CreatedBy");
ТаблицаДанныхСТерминала.НоваяКолонка("ProductID");
ТаблицаДанныхСТерминала.НоваяКолонка("DeclaredQuantity");
ТаблицаДанныхСТерминала.НоваяКолонка("CurrentQuantity");
ТаблицаДанныхСТерминала.НоваяКолонка("FirstCellID");
ТаблицаДанныхСТерминала.НоваяКолонка("FirstStorageBarcode");
ТаблицаДанныхСТерминала.НоваяКолонка("PackingID");
ТаблицаДанныхСТерминала.НоваяКолонка("SSCC");
ТаблицаДанныхСТерминала.НоваяКолонка("Index");
ТаблицаДанныхСТерминала.НоваяКолонка("RegisteredDate");
ТаблицаДанныхСТерминала.НоваяКолонка("RegistrationDate");
ТаблицаДанныхСТерминала.НоваяКолонка("ExpiredDate");
ТаблицаДанныхСТерминала.НоваяКолонка("SecondCellID");
ТаблицаДанныхСТерминала.НоваяКолонка("SecondStorageBarCode");
ТаблицаДанныхСТерминала.НоваяКолонка("BindedLine");
ТаблицаДанныхСТерминала.НоваяКолонка("code");
ТаблицаДанныхСТерминала.НоваяКолонка("barcode");
ТаблицаДанныхСТерминала.НоваяКолонка("serial");
ТаблицаДанныхСТерминала.НоваяКолонка("desc");
ТаблицаДанныхСТерминала.НоваяКолонка("sn");
ТаблицаДанныхСТерминала.НоваяКолонка("price");
ТаблицаДанныхСТерминала.НоваяКолонка("Ячейка");
ТаблицаДанныхСТерминала.НоваяКолонка("Заблокировано");
ТаблицаДанныхСТерминала.НоваяКолонка("ЦенаСклад");
ТаблицаДанныхСТерминала.НоваяКолонка("Алко");
ТаблицаДанныхСТерминала.НоваяКолонка("ПроверкаЧМ");
ТаблицаДанныхСТерминала.НоваяКолонка("АлкоСН");
ТаблицаДанныхСТерминала.НоваяКолонка("АлкоКод");
ТаблицаДанныхСТерминала.НоваяКолонка("АлкоНаим");
ТаблицаДанныхСТерминала.НоваяКолонка("АлкоКодВ");
ТаблицаДанныхСТерминала.НоваяКолонка("АлкоОбъем");
ТаблицаДанныхСТерминала.НоваяКолонка("АлкоКрепость");
ТаблицаДанныхСТерминала.НоваяКолонка("Производитель");
ТаблицаДанныхСТерминала.НоваяКолонка("ПроизвИНН");
ТаблицаДанныхСТерминала.НоваяКолонка("ПроизвКПП");
ТаблицаДанныхСТерминала.НоваяКолонка("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

```

Document GetDocument(string id)

Параметр

Тип

Описание

id

string

Идентификатор документа для получения. Если такой документ не найден будет возвращен null.

DocumentCollection GetDocuments(string docType, bool checkForFinish)

Параметр

Тип

Описание

docType

string

Имя типа документов для возврата. Если передана пустая строка, то будут возвращены документы всех типов.

checkForFinish

bool

Флаг, указывающий возвращать только завершенные документы или все подряд.

В отличие от выгрузки, в загруженном документе извлекать данные следует не из строк плана, а из фактических строк CurrentItems.

**Пример кода для произвольной учетной системы через компоненту COM**

```
connection = new COM("Cleverence.Warehouse.StorageConnector");
connection.SelectCurrentApp(СтрокаПодключения);
//запрашиваем завершенные документы всех типов
var docs = connection.GetDocuments("", true);
for(int i = 0; i < docs.Count; i++) //цикл по полученным документам
{
    var doc = docs.Item(i);
    // var doc = docs[i];
    for(int j = 0; j < doc.CurrentItems.Count; j++) //цикл по строкам факт документа
    {
        var dItem = doc.CurrentItems.Item(j);
        //запрос полей из строки
        string prodId = dItem.GetField("ProductId"); //код товара
        string packId = dItem.GetField("PackingId"); //код упаковки
        double qty = dItem.GetField("CurrentQuantity"); //факт кол-во в строке
        string alcoCode = dItem.GetField("АлкоКод"); //код упаковки
        //...
        //получение остальных полей, согласно таблице доступных полей
        //...
    }
}
```

Поля строк документа для получения из фактических строк

<b>Поле</b>
<b>Тип</b>
<b>Основное/доп.</b>
<b>Описание</b>
<b>ProductID</b>
String
основное
Идентификатор товара, для которого
<b>PackingID</b> нная позиция.
String
основное
Идентификатор упаковки для заданного
<b>CurrentQuantity</b> ается свойством
Double
основное
Фактическое количество товара в
<b>desc</b> ном виде упаковки.
String
дополнительное
Характеристика товара (если ведется
<b>serial</b> характеристиками).
String
дополнительное
Серия товара (если используется учет
<b>pric</b> ериям).
Decimal
дополнительное
Цена единицы товара в строке.
<b>Алко</b>
Boolean
дополнительное
Признак того, что товар является
<b>АлкоКод</b> ой или спиртосодержащей
прсStringей.
дополнительное
Код алкогольной продукции в ЕГАИС,
<b>АлкоПДФ</b> И из отсканированных марок.
String
дополнительное
Строка с PDF 417.
<b>АлкоСН</b>
string
дополнительное
Серийный номер бутылки, если
<b>ПроверкаЧМ</b>
Boolean
дополнительное
Признак того, что данная конкретная
<b>Производитель</b> сь в CheckMark на
легStringсть.
дополнительное
Наименование производителя
<b>ПроизводительИНН</b> сли уже известно
дляString того товара, либо если было
получено из CheckMark.
дополнительное
ИНН производителя, если уже известен
для данного товара, либо если был

ПроизводительКПП.	String
дополнительное КПП производителя, если уже известен	
АлкоНаим	о товара, либо если был получен из CheckMark. String
дополнительное	
Наименование товара из ЕГАИС, если	
АлкоКодВю	для данного товара, либо если получено из CheckMark. String
дополнительное	
Код вида алкогольной продукции из	
АлкоОбъем	уже известен для данного товара, либо если был получен из CheckMark. Decimal
дополнительное	
Ёмкость тары в литрах из ЕГАИС, если	
АлкоКрепость	я данного товара, либо если получена из CheckMark. Decimal
дополнительное	
Процентное содержание спирта из	
АлкоЕстьВФормах	известно для данного товара, либо если было получено из CheckMark. Boolean
дополнительное	
Признак того, что данная марка была	
КодФормыА	руженных формах А. String
дополнительное	
Код формы А, где была найдена марка.	
ДатаРозлива	DateTime
дополнительное	
Дата розлива позиции, если вводилась,	

#### Удаление документа

После успешной загрузки его необходимо удалить из Mobile SMARTS, иначе при следующем запросе он опять будет возвращен.

RemoveDocument(string documentId)

Параметр
Тип
Описание
id
string
Идентификатор документа для удаления.

#### Пример кода для произвольной учетной системы через компоненту COM

```
connection = new COM("Cleverence.Warehouse.StorageConnector");
connection.SelectCurrentApp(СтрокаПодключения);
//удаление конкретного документа по его идентификатору
connection.RemoveDocument(doc.Id);
```

#### Была ли статья полезна?

<input type="radio"/> Нет
<input type="radio"/> Да



