

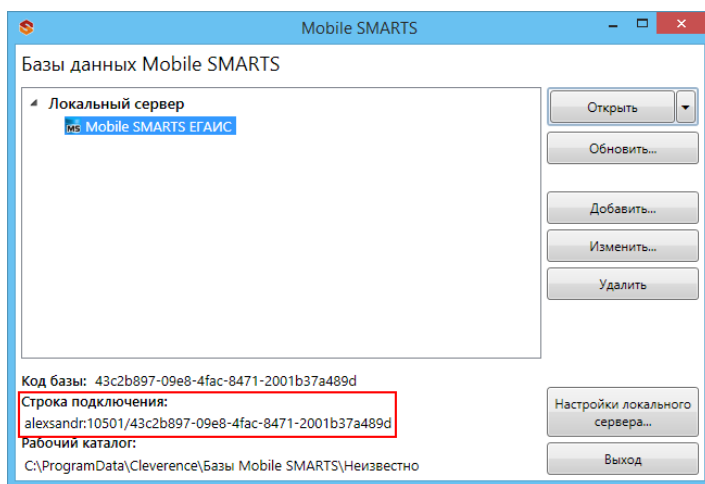
Инструкция по интеграции базы Mobile SMARTS для произвольной базы «1С:Предприятия» (v8)

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 (Внимание, для 1С версии 7, необходимо явным образом указывать режим работы компоненты – 7.7 (мДрайверТСД.УстановитьВерсию1С("v7"))):

```
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;
    Описание = " ";
КонецЕсли;
```

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 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. 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
// так, где тип значения int или Boolean, они будут в цикле выгрузки передаваться в виде
// 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 алкоголесодержащей
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.ПолучитьОшибку ("Описание ошибки");
    Сообщить ( "Ошибка при выгрузке данных: " + ". " + Ошибка, СтатусСообщения.Важное );
    Прервано = 1;
Прервать;
КонецЕсли;
КонецЕсли;
КонецЦикла;

```

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

Строка	Описание
мДанные. ДобавитьЗначение("Product.Ид");	uuid элемента справочника
мДанные. ДобавитьЗначение("Product.Marking");	Артикул
мДанные. ДобавитьЗначение("Product.Barcode");	Код товара
мДанные. ДобавитьЗначение("Packing.Barcode");	Штрихкод
мДанные. ДобавитьЗначение("Product.Name");	Наименование
мДанные. ДобавитьЗначение("Product.BasePackingId");	Является базовой единицей
мДанные. ДобавитьЗначение("Packing.Ид");	Единица измерения
мДанные. ДобавитьЗначение("Packing.Name");	Наименование единицы измерения
мДанные. ДобавитьЗначение("Packing.UnitsQuantity");	Коэффициент (кратность) единицы измерения
мДанные. ДобавитьЗначение("Packing.descr");	Характеристика единицы измерения
мДанные. ДобавитьЗначение("Packing.serial");	Серийный номер единицы измерения
мДанные. ДобавитьЗначение("Packing.price");	Стоимость
мДанные. ДобавитьЗначение("Packing.qty");	Количество
мДанные. ДобавитьЗначение("Product.withserial");	Используются серии
мДанные. ДобавитьЗначение("Packing.Алко");	Является алкоголесодержащей продукцией (bool)
мДанные. ДобавитьЗначение("Packing.АлкоВидЛиц");	Вид лицензии
мДанные. ДобавитьЗначение("Packing.АлкоМарк");	Является маркируемой продукцией
мДанные. ДобавитьЗначение("Packing.АлкоКодВ");	Код вида алкогольной продукции
мДанные. ДобавитьЗначение("Packing.АлкоНаимВ");	Наименование вида алкогольной продукции
мДанные. ДобавитьЗначение("Packing.АлкоОбъем");	Объем
мДанные. ДобавитьЗначение("Packing.АлкоКрепость");	Крепость
мДанные. ДобавитьЗначение("Packing.Производитель");	Производитель
мДанные. ДобавитьЗначение("Packing.ПроизВИНН");	ИНН Производителя
мДанные. ДобавитьЗначение("Packing.ПроизвКПП");	КПП Производителя
мДанные. ДобавитьЗначение("Packing.АлкоКод");	Код алкогольной продукции

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

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

Если данные не отправлены, получим ошибку.

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

```
Если мОбъектТсД.ЗавершитьВыгрузкуТоваров() = 0 Тогда  
    НомерОшибки = мОбъектТсД.ПолучитьОшибку(мОбъектТсД.ОписаниеОшибки);  
    Сообщить("Ошибка при завершении выгрузки: " + мОбъектТсД.ОписаниеОшибки);  
    Прервано = 1;  
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 = фиксация остатков в ЕГАИС;
- 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	Описание
Id: String	Идентификатор документа. Обязательно заполняется при выгрузке документа и должен иметь уникальное значение.
Name: String	Отображаемое имя документа. Обязательно заполняется при выгрузке документа.
Appointment: String	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	Идентификатор пользователя, который выполнил документ. Должен иметь уникальное значение для каждого пользователя.
UserName: String	Имя пользователя.
CreateDate: DateTime	
DocumentTypeName: String	Имя типа документа. Обязательно заполняется при выгрузке документа и должно иметь уникальное значение.
Modified: boolean	Признак того, что документ был изменен. Заполняется при изменении документа одним из пользователей.
InProgress: Boolean	Признак того, что документ захвачен пользователем на обработку. Заполняется, когда документ был захвачен пользователем.
Finished: Boolean	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	Идентификатор склада, к которому привязан документ. Должен иметь уникальное значение для каждого склада.
Barcode: String	Штрихкод документа. Для выбора документа по штрихкоду на терминале или на сервере должен быть обязательно заполнен.
Priority: Int32	Приоритет документа. Более приоритетные документы раньше отдаются на терминал для обработки.

Description: String	
DistributeByBarCode: Boolean	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	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.
DeviceID: String	
DeviceName: String	
DeviceIP: String	
LicenseStatus: Int32	
КонтрольКолва: Int32	
Поячейкам: Int32	
Зона: String	
ЗоняЯч: String	
НомерЕгаис: String	Алкокод.
ИмяОп: String	Имя операции, для занесения настроек.

Реквизиты «строк» документа

Реквизит	Описание
CreatedBy: CreateBy	Проверка упаковки.
ProductID: String	Идентификатор товара для которого описана данная позиция.
DeclaredQuantity: Double	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	Текущее количество товара в позиции. Показывает какое реальное количество товара задействовано пользователем.
FirstCellID: String	
FirstStorageBarcode: String	Штрихкод первого места хранения к которой привязан товар. В качестве места может выступать ячейка или паллета.
PackingID: String	Идентификатор упаковки для заданного товара. Товар задается свойством ProductId [ИдТовара].
SSCC: String	Уникальный номер единицы хранения. Служит для идентификации конкретных экземпляров товара. Заполняется из штрихкода товара по шаблону при выполнении операции.
Index: Int32	
RegisteredDate: DateTime	
RegistrationDate: DateTime	
ExpiredDate: DateTime	Срок годности товара. Инициализируется при занесении конкретного товара в позицию документа пользователем либо из штрихкода по шаблону, либо ручным выбором.
SecondCellID: String	
SecondStorageBarcode: String	Штрихкод второго места хранения к которой привязан товар. В качестве места может выступать ячейка или паллета.
BindedLine: DocumentItem	Задаёт связь строки из CurrentItems [СтрокиФакт] со строкой в DeclaredItems [СтрокиПлан].
Code: String	
Barcode: String	Штрихкод.
Serial: String	
Sn: String	
Ячейка: String	

Заблокировано: int32	
Price: Decimal	
ЦенаСклад: string	
Алко: Boolean	Признак того, что товар является алкогольной или спиртосодержащей продукцией.
ПроверкаЧМ: Boolean	Нужно проверять Check Mark или нет.
АлкоСН: string	Серийный номер бутылки.
АлкоКод: String	Строка с кодом алкогольной продукции в ЕГАИС.
АлкоНаим: String	Наименование товара.
АлкоКодВ: String	Строка с кодом вида алкогольной продукции.
АлкоОбъем: Decimal	Ёмкость тары в литрах. (в базе может быть в декалитрах, например, в Рознице).
АлкоКрепость: Decimal	Процентное содержание спирта.
Производитель: String	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.
ПроизводительИНН: String	Строка с ИНН производителя.
ПроизводительКПП: String	Строка с КПП производителя.
АлкоПДФ: String	Строка с PDF 417.

Реквизиты строка дополнительной табличной части документа «ФормыА»

Реквизит	Описание
КодФормы: String	
КодНоменклатуры: String	
АлкоКод: String	Строка с кодом алкогольной продукции в ЕГАИС.
Начало: Int32	Начало диапазона серийных номеров.
Конец: int32	Конец диапазона серийных номеров.

Основной метод записи в поля документ – это команда «SetField» («УстановитьПоле» - русск)
Соответственно, метод получения данных из полей – «GetField» («ПолучитьПоле» - русск)

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 = СоздатьОбъект("Cleverence.Warehouse.Document");
ДокументТСД.Ид - uuid Документа = ПолучитьUID(СсылкаНаНашДокумент);
ДокументТСД.ИмяТипаДокумента = «Сбор начальных остатков»;

Функция ПолучитьUID(СсылкаНаНашДокумент)

    Стр=ЗначениеВСтрокуВнутри(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 UID для документа

```

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

Строки документа создаются следующим образом:

```
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»).

```
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:

```
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:

```
ТаблицаСправокА = ДокументТДСД.Таблицы.ДобавитьЭлемент ();
ТаблицаСправокА.Name = "ФормыА";
мДрайверТДСД.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.АлкоГольнаяПродукция));
КонецЦикла;
```

Строка таблицы	Описание
"КодФормы"	код ЕГАИС справки А
"Начало"	начало диапазона
"Конец"	конец диапазона
"АлкоКод"	код ЕГАИС продукции
"КодНоменклатуры"	Uid номенклатуры (для сопоставления со строками документа)

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

```
Ответ = мОбъектТДСД.ВыгрузитьДокумент (ДокументТДСД);
Если Ответ = 0 Тогда
НомерОшибки = мОбъектТДСД.ПолучитьОшибку (мОбъектТДСД.ОписаниеОшибки);
Сообщить ("Ошибка: " + мОбъектТДСД.ОписаниеОшибки);
Иначе
Сообщить ("Документ "" + Строка (Документ1С) + "" выгружен на ТДСД);
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 ДокументТДСД["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.

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

Пример Б – загрузка с ТДС 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.
// 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);
        КонечЦикла;
    КонечЦикла;
    КонечЕсли;
КонечЦикла;
// Все, таблица готова для дальнейшей обработки в 1С
```

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

www.cleverence.ru