# Как происходит обмен документами между учетной системой и Mobile SMARTS через COM-компоненту

Последние изменения: 2024-03-26

При выполнении любых операций на терминале (Приемка, Отгрузка, Инвентаризация и др.) пользователь работает с некоторым документом Mobile SMARTS. Документ может быть выгружен из учетной системы и содержать список товаров, которые нужно отсканировать (или предполагается, что они будут отсканированы). Например, должна произойти приемка товара на склад и из учетной системы выгружается накладная со списком товаров, при этом по факту приемки могут быть расхождения (какой-то товар не привезли или привезли не в том количестве, а может придти товар, которого нет по накладной). Фактически отсканированный товар фиксируется в документе Mobile SMARTS. После завершения работы с документом на терминале, документ загружается в учетную систему и происходит регистрация принятого товара. Кроме выгрузки из учетной системы, документ может быть создан на терминале пользователем (если это позволяют настройки данного типа документа, заданные в конфигурации Mobile SMARTS). В таком документе будут только фактически отсканированные товары. Например, приемка на склад может производится по факту без выгрузки какого-либо документа из учетной системы. В этом случае при загрузке завершенного документа с терминала, как правило, создается новый документ учетной системы.

Таким образом, документ Mobile SMARTS служит заданием для выполнения на терминале (если он выгружается из учетной системы) и является результатом работы пользователя на терминале, который требуется отразить в учетной системе.

Алгоритм работы с документом каждого типа (какие данные должен вводить пользователь и как они будут обрабатываться), задается в Панели управления (см. Панель управления, Тип документа).

Документ Mobile SMARTS содержит два списка товаров (Плановая часть - товары, которые должны быть или предполагается отсканировать, Фактическая часть - фактически отсканированные товары), а также, если необходимо, другие данные (поля шапки, дополнительные табличные части).

Объектом документа является Document [Документ]. С помощью этого объекта происходит обмен данными между учетной системой и Mobile SMARTS (выгрузка документов-заданий и загрузка результатов работы с терминалов).

Document [Документ] содержит следующие основные поля и методы:

| Свойства |
| --- |
| Наименование |
| Тип |
| Описание |
| **Свойства шапки документа, назначаемые при выгрузке из учетной системы:** |
| Id [Ид] |
| Строка |
| Уникальный идентификатор документа. При выгрузке из учетной системы должен идентифицировать исходный выгружаемый документ. При создании документа на терминале назначается автоматически. |

| |
|---|
| Name [Имя] |
| Строка |
| Название документа. Произвольное наименование, которое отображается на экране терминала. Назначается при выгрузке из учетной системы (например, “Приемка №327 от 12.07.16”), при создании документа на терминале присваивается автоматически. |
| DocumentTypeName [ИмяТипаДокумента] |
| Строка |
| Должно соответствовать имени одного из типов документов, имеющихся в конфигурации Mobile SMARTS. |
| WarehouseId [ИдСклада] |
| Строка |
| Идентификатор склада Mobile SMARTS, к которому привязан документ. При выгрузке из учетной системы должен быть равен Ид. одного из складов, имеющихся в конфигурации Mobile SMARTS. В типовых конфигурациях по умолчанию существует единственный склад “Общий” с Ид. равным “1”. |
| Appointment [Назначение] |
| Строка |
| Назначение документа - код пользователя или имя группы<br><br>пользователей, которым данный документ назначается на исполнение.<br><br>Если значение пустое, то документ попадает к первому свободному<br><br>пользователю, которому разрешен тип документа. В типовых конфигурациях Mobile SMARTS по умолчанию заведен единственный пользователь с Ид. “оператор”. |
| AutoAppointed<br><br>[ВыдаватьАвтоматически] |
| Булево |
| Флаг автоматической выдачи. Если стоит Истина - документ выдается на ТСД, автоматически, не дожидаясь<br><br>явного выбора его пользователем из списка или по ШК. Имеет смысл только для серверной базы Mobile SMARTS. По умолчанию Истина. Выдача<br><br>автоматически не означает, что документ будет сразу же открыт без участия пользователя. Сервер<br><br>назначает документ для исполнения, он отправляется на терминал,<br><br>пользователь получает звуковое уведомление, дальше документ нужно будет открыть, выбрав его из списка или по ШК. |

| |
|---|
| Barcode [Штрихкод] |
| Строка |
| Штрихкод документа. Документ на терминале может быть открыт путем сканирования штрихкода с бумажной копии документа (если это разрешено настройками типа документа в конфигурации Mobile SMARTS). |
| ServerHosted<br><br>[ИсполняемыйНаСервере] |
| Булево |
| 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. По умолчанию Ложь. |
| CreateDate [ДатаСоздания] |
| ДатаВремя |
| Дата создания документа. Если не назначать, проставится текущая дата. |
| Priority [Приоритет] |
| Целое |
| Приоритет документа. Более<br><br>приоритетные документы раньше отдаются на терминал для обработки. Имеет смысл только для серверной базы Mobile SMARTS. Назначать не обязательно. |

| |
|---|
| Description [Описание] |
| Строка |
| Описание документа, назначать не обязательно. |
| **Свойства шапки документа, которые могут использоваться при загрузке в учетную систему:** |
| Finished [Завершен] |
| Булево |
| Признак того, что обработка документа пользователем была завершена, и его можно забирать назад в учетную систему. |
| Modified [Изменен] |
| Булево |
| Признак того, что документ был изменен. |

| |
|---|
| **InProcess [ВОбработке]** |
| **Булево** |
| **Признак того, что документ захвачен пользователем на обработку.** |
| **Completed**<br><br>**[ВсеСтрокиПланаВыполнены]** |
| **Булево** |
| **Свойство, позволяющее проверить все ли строки задания выполнены**<br><br>**(фактическое количество товара равно плановому). Проверка происходит по**<br><br>**плановым строкам DeclaredItems.** |
| **Overloaded [ЕстьПереполнение]** |
| **Булево** |
| **Признак, есть ли превышение количества товара в строках документа.** |
| **Underloaded [ЕстьНедобор]** |
| **Булево** |
| **Признак, есть ли недобор количества товара в строках документа.** |
| **CreatedOnPDA [СозданНаТСД]** |
| **Булево** |
| **Признак того, что документ был создан на ТСД. Истина - создан на ТСД, Ложь - выгружен из учетной системы.** |
| **UserId [ИдПользователя]** |
| **Строка** |
| **Ид. пользователя ТСД, который выполнил документ.** |
| **UserName [ИмяПользователя]** |
| **Строка** |
| **Имя пользователя ТСД, который выполнил документ.** |
| **DeviceId [ИдУстройства]** |
| **Строка** |
| **Ид. устройства (уникальный код терминала), на котором был завершен документ.** |

| |
|---|
| DeviceIP [ИпУстройства] |
| Строка |
| IP-адрес устройства, на котором был завершен документ. |
| DeviceName [ИмяУстройства] |
| Строка |
| Имя устройства, на котором был завершен документ. |
| **Коллекции строк документа:** |
| DeclaredItems [СтрокиПлан] |
| DocumentItemCollection [КоллекцияСтрокДокумента] |
| Коллекция строк документа DocumentItem [СтрокаДокумента], содержащая данные о товарах, которые предполагается отсканировать по плану. Заполняется при выгрузке из учетной системы. При этом на терминале при работе с документом также может происходить запись в плановые строки. |
| CurrentItems [СтрокиФакт] |
| DocumentItemCollection [КоллекцияСтрокДокумента] |
| Коллекция строк документа DocumentItem [СтрокаДокумента], содержащая данные о фактически отсканированных товарах. Заполняется при работе на терминале. |
| Tables [Таблицы] |
| DocumentTableCollection [КоллекцияТаблиц] |
| Коллекция дополнительных таблиц документа.<br><br>В конфигурации Mobile SMARTS для типа документа могут быть определены дополнительные табличные части с<br><br>произвольным набором полей. Дополнительные табличные части могут быть заполнены при выгрузке документа из учетной системы, также они могут заполняться при работе на терминале. |
| **Методы** |

| Наименование |
| --- |
| Параметры и<br><br>возвращаемое значение |
| Описание |

| GetField [ПолучитьПоле] |
| --- |
| Параметр: string (имя поля)<br><br>Возвращаемое значение: object (значение поля) |
| Возвращает значение дополнительного поля шапки документа. Если поля с<br><br>переданным именем в объекте нет, вернется null.<br><br>Пример:<br><br>var value  = document.GetField("ИмяПоля");<br><br><br><br>значПоля = документ.ПолучитьПоле("ИмяПоля") |
| SetField [УстановитьПоле] |
| Параметры: string (имя поля), object (значение)<br><br>Возвращаемое значение: нет |
| Устанавливает значение дополнительного поля шапки документа. Пример:<br><br>document.SetField("ИмяПоля", value);<br><br>документ.УстановитьПоле("ИмяПоля", значение); |

Объектом строки документа является DocumentItem [СтрокаДокумента]. Данные объекты содержатся в коллекции плановых (DeclaredItems [СтрокиПлан]) и фактических (CurrentItems [СтрокиФакт]) строк документа.

DocumentItem [СтрокаДокумента] содержит следующие основные поля и методы:

| Свойства |
| --- |
| Наименование |
| Тип |
| Описание |

| ProductId [ИдТовара] |
| --- |
| Строка |
| Идентификатор товара. Товар с таким идентификатором должен быть в выгруженном в справочнике номенклатуры. |

| PackingId [ИдУпаковки] |
| --- |
| Строка |
| Идентификатор упаковки товара. Упаковка с таким ид. должна быть в коллекции упаковок товара, заданного в строке документа. |
| DeclaredQuantity [КоличествоПлан] |
| Число |
| Плановое количество товара. Значение заполняется при выгрузке документа из учетной системы. |
| CurrentQuantity [КоличествоФакт] |
| Число |
| Фактическое количество товара. Заполняется при работе на терминале и отражает, сколько фактически было отсканировано данного товара. |
| SSCC [КодЕдиницыХранения] |
| Строка |
| Serial shipping container code - уникальный номер единицы хранения. Поле может использоваться для занесения номера контейнера, палеты и т.д. Может заполняться из штрихкода EAN-128 (см. Интерпретация кода EAN-128 в Mobile SMARTS). |
| FirstStorageBarcode [ШтрихкодПервогоМеста] |
| Строка |
| Штрихкод ячейки, палеты и т.п., в которых находится или из которых перемещается товар. |
| SecondStorageBarcode [ШтрихкодВторогоМеста] |
| Строка |
| Штрихкод ячейки, палеты и т.п. в которые перемещается товар из первого места хранения. |
| ExpiredDate [СрокГодности] |
| ДатаВремя |
| Срок годности товара. Может заполняться из штрихкода EAN-128 (см. Интерпретация кода EAN-128 в Mobile SMARTS). |

| |
|---|
| RegistrationDate [ДатаРегистрации] |
| ДатаВремя |
| Дата регистрации товара. Может заполняться на терминале. |
| BindedLine [СвязаннаяСтрока] |
| DocumentItem<br><br>[СтрокаДокумента] |
| Связанная строка документа. Заполняется на терминале и задает связь строки из фактической части со строкой из плановой части документа. |
| Overload [Переполнение] |
| Число |
| Возвращает превышение фактического количества товара над плановым (разницу между фактическим и плановым количеством). |
| Underload [Недобор] |
| Число |
| Возвращает разницу между плановым и фактическим количеством. |
| UnderloadedOrOverloaded [ЕстьНедоборИлиПереполнение] |
| Булево |
| Позволяет проверить совпадают ли заявленное и фактическое количество в строке. Истина - количества разные, Ложь - количества одинаковые. |
| Методы |
| Наименование |
| Параметры и возвращаемое значение |
| Описание |

| GetField [ПолучитьПоле] |
| --- |
| Параметр: string (имя поля) Возвращаемое значение: object (значение поля) |
| Возвращает значение дополнительного поля шапки документа. Если поля с<br>переданным именем в объекте нет, вернется null.<br>Пример:<br>var value =<br>documentItem.GetField("ИмяПоля");<br>значПоля =<br>строкаДокумента.ПолучитьПоле("ИмяПол<br>я") |
| SetField [УстановитьПоле] |
| Параметры: string (имя поля), object (значение) Возвращаемое значение: нет |
| Устанавливает значение дополнительного поля шапки документа.<br>Пример:<br>document.SetField("ИмяПоля", value);<br>документ.УстановитьПоле("ИмяПоля", значение); |

Для обмена документами используются следующие функции объекта StorageConnector:

StorageConnector [Соединение]

| Методы |
| --- |
| Наименование |
| Параметры и возвращаемое значение |
| Описание |
| Выгрузка документов: |

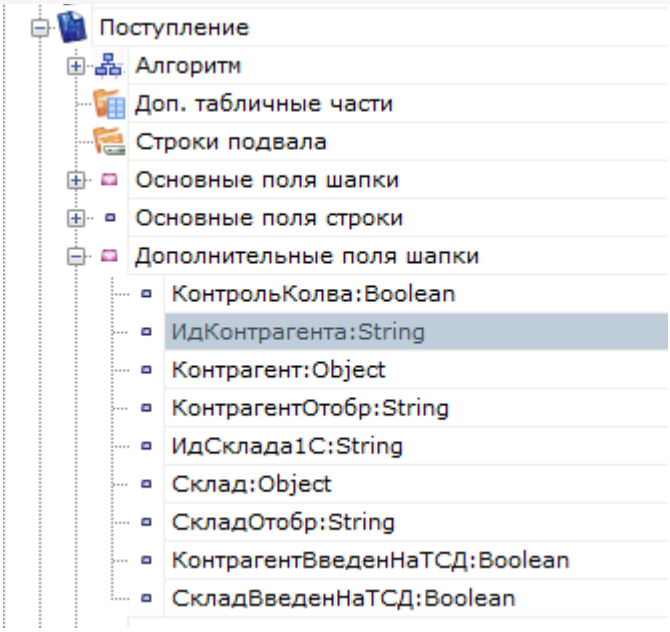
| |
|---|
| SetDocument<br><br>[ВыгрузитьДокумент] |
| Параметры: Document document<br><br>document<br><br>Выгружаемый документ.<br><br>Возвращаемое значение: нет |
| Выгружает документ в базу Mobile SMARTS. |
| SetDocuments<br><br>[ВыгрузитьДокументы] |
| Параметры: DocumentCollection documents<br><br>documents<br><br>Коллекция выгружаемых документов.<br><br>Возвращаемое значение: нет |
| Выгружает сразу несколько документов в базу Mobile SMARTS. |
| **Получение документов:** |
| GetDocuments<br><br>[ПолучитьДокументы] |
| Параметры: string docType,<br><br>bool checkForFinish<br><br>docType Наименование типа<br><br>документов, которые следует получить. Если передать<br><br>пустую строку, возвращает все документы, независимо от<br><br>типа. checkForFinish<br><br>Если Истина, вернутся только завершенные на терминале документы. Если Ложь, все документы.<br><br>Возвращаемое значение: Коллекция документов, отобранных в соответствии с заданными условиями. |
| Получает документы из базы Mobile SMARTS в соответствии с заданными параметрами.<br><br>Если документов, удовлетворяющих условиям нет, вернется пустая коллекция. |

| |
|---|
| **GetDocument**<br><br>**[ПолучитьДокумент]** |
| **Параметры:**<br><br>**string documentId**<br><br>**documentId**<br><br>**Идентификатор документа.**<br><br>**Возвращаемое значение: объект документа Document или null.** |
| **Получает документ из базы Mobile SMARTS по идентификатору.** |
| **GetDocumentsByIds**<br><br>**[ПолучитьДокументыПоИдентиф икаторам]** |
| **Параметры:**<br><br>**StringCollection idList**<br><br>**idList**<br><br>**Коллекция идентификаторов документов.**<br><br>**Возвращаемое значение: коллекция документов DocumentCollection.** |
| **Получает из базы Mobile SMARTS документы с заданными идентификаторами.** |
| **GetDocument**<br><br>**[ПолучитьДокумент]** |
| **Параметры:**<br><br>**string idocumentId**<br><br>**idocumentId**<br><br>**Идентификатор документа.**<br><br>**Возвращаемое значение: объект документа Document или null, если документ не найден.** |
| **Получает документ из базы Mobile SMARTS по идентификатору.** |
| **Удаление документов:** |

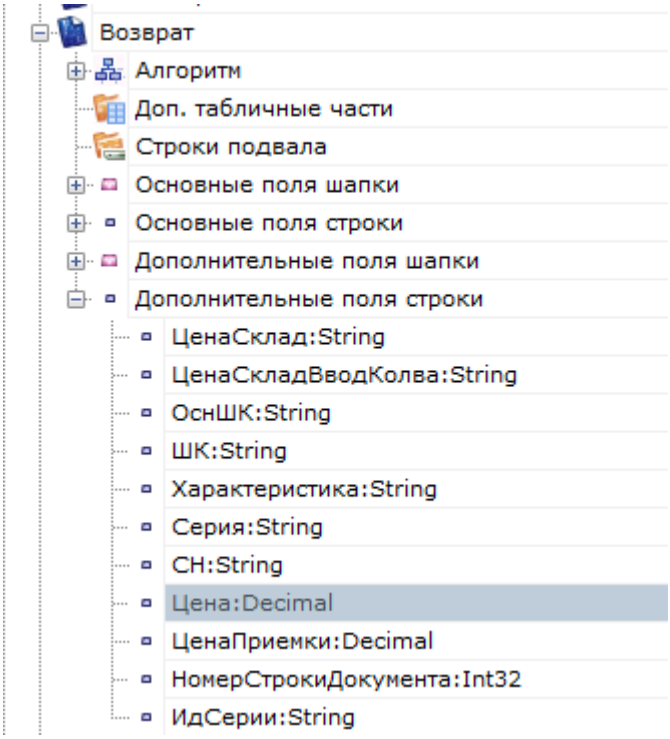
| RemoveDocument [УдалитьДокумент] |
| --- |
| Параметры:<br><br>string documentId<br><br>documentId<br><br>Идентификатор документа.<br><br>Возвращаемое значение: нет. |
| Удаляет документ из базы Mobile SMARTS по идентификатору.<br><br>Если документ не найден, функция ничего не делает. |
| RemoveDocuments [УдалитьДокументы] |
| Параметры: DocumentCollection documents<br><br>documents<br><br>Коллекция документов.<br><br>Возвращаемое значение: нет. |
| Удаляет заданные документы из базы. |

## Выгрузка документов

Выгрузка из учетной системы используется, если требуется отправить на терминал определенное задание для работы. В учетной системе, как правило, имеется некоторый документ, содержащий список товаров с количествами, а также, возможно, другие данные (поля шапки и строк документа, табличные части). Например, накладная на приемку. Чтобы реализовать выгрузку документа из учетной системы, нужно определить, какие данные нужны на терминале при работе с документом, т.е. определить состав выгружаемых полей и соответствие полей документа из учетной системы полям документа Mobile SMARTS. Кроме основных полей, объекты Document [Документ] и DocumentItem [СтрокаДокумента] могут содержать произвольные дополнительные поля, доступ к которым осуществляется с помощью функций SetField [УстановитьПоле] и GetField [ПолучитьПоле]. Состав дополнительных полей шапки и строки определяется для каждого типа документа в Панеле управления Mobile SMARTS (см. Тип документа):




**Поля шапки**




**Поля строки**

**Примечание: некоторые поля, задаваемые в конфигурации Mobile SMARTS являются вычислимыми, т.е. их значение определяется путем некоторых операций над другими полями. Для вычислимого поля задается шаблон значения:**

| | |
|---|---|
| Тип поля | Decimal |
| Шаблон значения | Item.НоваяЦена > 0? Item.НоваяЦена:Item.Цена |

**Присваивать значения вычислимых полей при выгрузке нельзя, однако при загрузке получить значение такого поля можно.**

**Общий алгоритм выгрузки документа. В общем виде алгоритм выгрузки выглядит так:**

1. Создаем объект Cleverence.Warehouse.StorageConnector и устанавливаем подключение к нужной базе Mobile SMARTS при помощи вызова SelectCurrentApp[УстановитьПодключениеСБазойСМАРТС];
2. Получаем в учетной системе необходимые для выгрузки документа данные (объект документа учетной системы, некую выборку и т.п.);
3. Создаем объект Document [Документ];
4. Устанавливаем у созданного объекта Document [Документ] свойства:
   - **Id [Ид]** - обычно равен идентификатору или номеру документа в учетной системе. Может быть так, что в учетной системе документы различных типов имеют одинаковый номер и эти разные типы документов выгружаются в Mobile SMARTS. В этом случае Id [Ид] нужно формировать так, чтобы он однозначно определял документ в Mobile SMARTS и для разных выгружаемых документов не мог повторяться (например, <Имя типа документа><разделитель><номер> - “Приемка#БРД000003”);

   - **Name [Имя]** - название документа, которое отображается на терминале при выборе документа из списка и в шапках окон (если не отключено настройками в конфигурации). Можно использовать наименование документа из учетной системы (например, “Приемка БРД000003 от 12.06.16”). Примечание: иногда в списке документов на терминале нужно отобразить дополнительную информацию (например, контрагента, от которого пришел товар), выделить какую-то часть текста в списке цветом и т.п., чтобы пользователю было проще найти нужный документ. В этом случае следует использовать свойство “Шаблон отображения документов в списке” типа документа в Панеле управления, через шаблон можно вывести значения любых основных и дополнительных полей шапки документа (см. Шаблоны);

   - **DocumentTypeName [ИмяТипаДокумента]** - должно быть равно имени одного из типов документов в конфигурации Mobile SMARTS. Например, “Приемка”;

   - **WarehouseId [ИдСклада]** - должен быть равен Ид. одного из складов, имеющихся в конфигурации Mobile SMARTS. В типовых конфигурациях по умолчанию существует единственный склад “Общий” с Ид. равным “1”;

   - **Appointment [Назначение]** - код пользователя или имя группы пользователей, которым данный документ назначается на исполнение. Может быть пустым (“общий” документ, имеет смысл только для серверной базы Mobile SMARTS). В типовых конфигурациях Mobile SMARTS по умолчанию заведен единственный пользователь с Ид. “оператор”. В самом простом случае присваиваем “оператор”. Если пользователей терминалов несколько, в учетной системе может быть реализован выбор пользователя, которому будет отправлен документ. Для этого нужно получить список пользователей из базы Mobile SMARTS (см. Получение пользователей) и предложить выбор. В этом случае присваивается выбранное значение;

   - **AutoAppointed [ВыдаватьАвтоматически]** - по умолчанию Истина (документ сразу отправляется назначенному пользователю). Если нужно, чтобы документ оставался на сервере Mobile SMARTS, пока пользователь на терминале не выберет его из списка или по ШК, свойство следует проставить в Ложь. Например, оставив поле Appointment [Назначение] пустым, а AutoAppointed [ВыдаватьАвтоматически] выставив в Ложь, можно добиться чтобы документ, выгруженный на сервер, был виден всем пользователям терминалов (“общий” документ). Документ остается видным для всех пользователей, пока кто-то один не откроет его на обработку. Имеет смысл только для серверной базы Mobile SMARTS;

5. Проставляем объекту документа необходимые дополнительные поля шапки при помощи SetField

[УстановитьПоле]. Состав полей зависит от типа документа. Например, для приемки может потребоваться наименование или идентификатор контрагента, которые берем из исходного документа учетной системы;

6. Обходим в цикле строки исходного документа учетной системы (при необходимости перед этим может быть получена какая-то выборка строк. Например, нужно выгружать не все строки или выполнить группировку). На каждой итерации цикла создаем объект DocumentItem[СтрокаДокумента], в строке документа проставляем поля:
   - **ProductId [ИдТовара]** - должен соответствовать ид. одного из товаров из выгруженного справочника номенклатуры. Для “неизвестного товара” (см. ниже) равно “*”. Если номенклатура не выгружается из учетной системы и реализовано он-лайн получение товара с помощью события сервера ТоварНеНайден (см. События сервера, ТоварНеНайден), функция- обработчик события должна возвращать товар по такому идентификатору;

   - **PackingId [ИдУпаковки]** - упаковка с таким Ид. должна быть в коллекции упаковок данного товара в справочнике товаров базы Mobile SMARTS. Для “неизвестного товара” (см. ниже) равно “шт”. Если номенклатура не выгружается из учетной системы и реализовано он-лайн получение товара с помощью события сервера ТоварНеНайден (см. События сервера, ТоварНеНайден), функция-обработчик события должна возвращать товар, содержащий упаковку с таким идентификатором;

   - **DeclaredQuantity [КоличествоПлан]** - плановое количество товара;

   - При необходимости заполняем поля SSCC [КодЕдиницыХранения], FirstStorageBarcode [ШтрихкодПервогоМеста] и др.

   - Проставляем нужные дополнительные поля строки при помощи SetField [УстановитьПоле]. Например, характеристика товара: documentItem.SetField(“Характеристика”, descr) [строкаДок.УстановитьПоле(“Характеристика”, знач) ]

   - **Добавляем** созданную и заполненную строку в коллекцию **DeclaredItems [СтрокиПлан]** документа: document.DeclaredItems.Add(documentItem) [документ.СтрокиПлан.Добавить(строкаДок)];

7. Если кроме заполнения плановых строк требуется выгрузка дополнительных табличных частей документа, подготавливаем данные для выгрузки. Для каждой выгружаемой дополнительной таблицы документа:

   7.1. Создаем объект Table [Таблица], проставляем наименование Name [Имя], как оно задано в конфигурации Mobile SMARTS;
   7.2. Обходим в цикле выгружаемый набор данных (табличную часть документа учетной системы), на каждой итерации цикла создаем объект Row [СтрокаТаблицы];
   - Проставляем поля в строке при помощи SetField [УстановитьПоле], наименования полей должны соответствовать заданным в конфигурации Mobile SMARTS для даннойтабличной части;
   - Добавляем созданную строку в коллекцию строк таблицы: documentTable.Rows.Add(row) [таблицаДокумента.Строки.Добавить(строка)];

   7.3. Добавляем созданную и заполненную таблицу к таблицам документа: document.Tables.Add(table) [документ.Таблицы.Добавить(таблица)];
8. **Выгружаем** заполненный документ: storageConnector.SetDocument(document) [соединение.ВыгрузитьДокумент(документ)].

**“Неизвестный товар”**. Строка документа Mobile SMARTS, как правило, определяет конкретный товар с упаковкой (единицей измерения). В этом случае поля ProductId [ИдТовара] и PackingId [ИдУпаковки] соответствуют идентификатору товара и упаковки из справочника номенклатуры. Но возможна ситуация, когда справочник номенклатуры вообще не используется при работе на терминале, а вся необходимая информация содержится в строках документа. Например, в строке документа записан номер рабочего инструмента (без привязки к номенклатуре), а документ представляет собой список всех инструментов, которые необходимо

**выдать. Тогда номер инструмента записывается в дополнительное поле строки, а при помощи ProductId [ИдТовара] и PackingId [ИдУпаковки] задается “неизвестный товар”. В случае “неизвестного товара” ProductId [ИдТовара] должен быть равен “*”, PackingId [ИдУпаковки] – “шт”.**

## Пример выгрузки

**C#:**

```
var connector = new Cleverence.Warehouse.StorageConnector(); // Создаем экземпляр
 объекта соединения



// Выполняем подключение к базе Mobile SMARTS

connector.SelectCurrentApp("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-c7c7003887c5"); try

{

var document = new Cleverence.Warehouse.Document(); //Создаем объект документа Mobile
SMARTS

var acceptanceDoc = AcceptanceJournal.Find(acceptanceDocId); //Получаем документ учетной
системы



document.Id = acceptanceDoc.Id; //Ид. документа document.Name = acceptanceDoc.Name;
//Имя документа

document.DocumentTypeName = "Поступление"; //Имя
типа документа document.WarehouseId = "1"; //Ид. склада, к которому привязан документ



if(sharedDoc)

{

//"Общий" документ - назначение пустое, остается на сервере до явного выбора.
```

```
document.Appointment = "";

document.AutoAppointed = false;

}

else

{

document.Appointment = selectedUserId; ///Ид. выбранного пользователя Mobile SMARTS или можно

//присвоить "оператор"
для типовых конфигураций Mobile SMARTS

}

//Проставляем дополнительные поля шапки документа document.SetField("ИдКонтрагента", acceptanceDoc.Contractor.Id); document.SetField("ИдСклада", acceptanceDoc.Warehouse.Id);

// Обходим в цикле строки табличной части исходного
докумнта foreach(var acceptanceDocItem in acceptanceDoc.InventItems)

{

//Создаем строку документа Mobile SMARTS и присваиваем Ид. товара, Ид. упаковки и плановое //количество товара

var documentItem = new Cleverence.Warehouse.DocumentItem(); documentItem.ProductId = acceptanceDocItem.InventItem.Id; documentItem.PackingId = acceptanceDocItem.Unit.Name; documentItem.DeclaredQuantity = acceptanceDocItem.Quantity;
```

```
//Проставляем дополнительные поля строки документа if(acceptanceDocItem.InventDim != null)

documentItem.SetField("Характеристика", acceptanceDocItem.InventDim.Name);



documentItem.SetField("Цена", acceptanceDocItem.Price);



//Добавляем строку в коллекцию плановых строк

document.DeclaredItems.Add(documentItem);

}



//Создание дополнительной таблицы документа

var boxesTable = new Cleverence.Warehouse.DocumentTable(); boxesTable.Name = "Короба"; //Присваиваем имя



foreach(var boxItem in acceptanceDoc.Boxes)

{

//Создаем в цикле строки доп. таблицы

var row = new Cleverence.Warehouse.Row(); row.SetField("Номер", boxItem.Id); row.SetField("Описание", boxItem.Description);

boxesTable.Rows.Add(row); //Добавление строки в
 коллекцию строк таблицы

}
```

```
}



//Добавляем таблицу в
 коллекцию таблиц документа document.Tables.Add(boxesTable);



connector.SetDocument(document); //Выгружаем
 документ



}

catch

{

//Обработка ошибки

}
```

**«1С:Предприятие 8»:**

```
connector = новый COMОбъект("Cleverence.Warehouse.StorageConnector"); //
Создаем экземпляр объекта
//соединения Попытка
connector.УстановитьПодключениеСБазойСМАРТС("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-
c7c7003887c5");
//Выполняем подключение
ДокументТСД = новый COMОбъект("Cleverence.Warehouse.Document"); Документ1С =
Документы.ПоступлениеТоваров.НайтиПоНомеру(НомерДокумента);
 ДокументТСД.Ид = “Поступление#”+Документ1С.Номер; //Ид. документа ДокументТСД.Имя
= Строка(Документ1С); //Имя документа ДокументТСД.ИмяТипаДокумента = "Поступление";
//Имя типа документа ДокументТСД.ИдСклада = "1"; //Ид. склада, к которому привязан
документ
 Если
ОбщийДокумент = Истина Тогда
//"Общий" документ - назначение пустое, остается на сервере до явного выбора.
ДокументТСД.Назначение =
"";
ДокументТСД.ВыдаватьАвтоматически = Ложь; Иначе
ДокументТСД.Назначение = ИдПользователяТСД; //Ид. выбранного пользователя Mobile
```

```
SMARTS или можно
//присвоить "оператор"
для типовых конфигураций Mobile SMARTS КонецЕсли;
 //Проставляем дополнительные поля шапки документа
ДокументТСД.УстановитьПоле("ИдКонтрагента", ДокументТСД.Контрагент.Код);
ДокументТСД.УстановитьПоле("ИдСклада", ДокументТСД.Склад.Код);
 Для каждого строкаДок1С из Документ1С.Товары Цикл
//Создаем строку документа Mobile SMARTS и присваиваем Ид. товара, Ид. упаковки и
плановое //количество товара



СтрокаДокументаТСД = Новый COMОбъект("Cleverence.Warehouse.DocumentItem");
СтрокаДокументаТСД.ИдТовара = XMLСтрока(строкаДок1С.Номенклатура);

СтрокаДокументаТСД.ИдУпаковки = строкаДок1С.Упаковка.Наименование;
СтрокаДокументаТСД.КоличествоПлан = строкаДок1С.Количество;
 //Проставляем дополнительные поля строки документа
СтрокаДокументаТСД.УстановитьПоле(“Характеристика”,
Строка(строкаДок1С.Характеристика)); СтрокаДокументаТСД.УстановитьПоле(“Цена”,
строкаДок1С.Цена);
 //Добавляем строку в коллекцию плановых строк
ДокументТСД.СтрокиПлан.Добавить(СтрокаДокументаТСД);
КонецЦикла;
 //Создание дополнительной таблицы документа
ТаблицаКоробовТСД = Новый COMОбъект("Cleverence.Warehouse.Table");
ТаблицаКоробовТСД.Имя =
 “Короба”;
 Для каждого строкаКоробаДок1С из Документ1С.Короба Цикл
//Создаем в цикле строки доп. таблицы
СтрокаКоробаТСД = Новый COMОбъект("Cleverence.Warehouse.Row");
СтрокаКоробаТСД.УстановитьПоле(“Номер”, строкаКоробаДок1С.НомерКороба);
СтрокаКоробаТСД.УстановитьПоле(“Описание”, строкаКоробаДок1С.Описание);
 ТаблицаКоробовТСД.Строки.Добавить(СтрокаКоробаТСД); //Добавление строки в коллекцию
строк таблицы КонецЦикла;
 //Добавляем таблицу в
 коллекцию таблиц документа ДокументТСД.Таблицы.Добавить(ТаблицаКоробовТСД);
connector.ВыгрузитьДокумент(ДокументТСД); //Выгружаем документ Исключение
// При выгрузке возникли ошибки, обрабатываем исключение




КонецПопытки;
```

## Загрузка документов

**Для того, чтобы отразить результаты работы пользователей терминалов в учетной системе, следует загрузить завершенные документы, содержащие фактические данные. Например, после приемки товара, кладовщик завершает документ на терминале, в документе содержится список фактически принятых товаров с количествами, этот список загружается в учетную систему и на его основе заполняется табличная часть документа учетной системы, после чего документ проводится и в учетной системе фиксируется принятый товар.**

**Общий алгоритм загрузки документов.**

1. **Создаем объект Cleverence.Warehouse.StorageConnector и устанавливаем подключение к нужной базе Mobile SMARTS при помощи вызова SelectCurrentApp [УстановитьПодключениеСБазойСМАРТС];**

2. **Получаем из базы Mobile SMARTS документы для загрузки. Если нужно загрузить все завершенные документы (возможно, с фильтром по типу документа), используем функцию GetDocuments [ПолучитьДокументы]. Если известен идентификатор документа, который требуется загрузить, получаем один этот документ с помощью функции GetDocument [ПолучитьДокумент];**

3. **Для загрузки данных из документа Mobile SMARTS может быть создан новый документ учетной системы или загрузка может выполняться в существующий. Если загружается документ ранее выгруженный из учетной системы, загрузка может выполняться в исходный документ (при этом заполняется фактический список товаров и количества) или может быть создан новый документ другого типа (например, выгружали Заказ поставщику, при загрузке создаем Поступление товаров), также пользователь учетной системы может выбрать документ, в который будет происходить загрузка. Конкретный вариант определяется принятым бизнес-процессом. Перед загрузкой в существующий документ нужно обнулить количества товара в табличной части документа, в который происходит загрузка, или очистить табличную часть (Если для количества используется только одно поле и нет двух полей, к примеру, Количество и КоличествоФакт).**

4. **Получаем из документа Mobile SMARTS поля шапки, которые требуются, с помощью вызова GetField [ПолучитьПоле], и записываем нужные данные в поля документа учетной системы;**

5. **Обходим в цикле строки фактической части документа Mobile SMARTS CurrentItems [СтрокиФакт], на каждой итерации цикла:**
   **5.1. На основании значений полей ProductId [ИдТовара], PackingId [ИдУпаковки] строки документа Mobile SMARTS находим в учетной системе товар и упаковку (единицу измерения);**
   **5.2. Получаем из строки документа Mobile SMARTS необходимые при загрузке дополнительные поля при помощи GetField [ПолучитьПоле] (например, Характеристика, Цена);**
   **5.3. Ищем в табличной части документа учетной системы строку с данным товаром и другими соответствующими полями (например, характеристикой). Если строка найдена, прибавляем количество товара. Если нет - создаем новую строку, записываем в нее товар, упаковку, другие поля если нужно (например, характеристику, цену);**

6. **Когда все данные загружены, записываем документ в учетной системе;**

7. **Если загруженный документ Mobile SMARTS больше не нужен, удаляем его из базы при помощи функции RemoveDocument [УдалитьДокумент].**

## Пример загрузки

**C#:**

```
var connector = new Cleverence.Warehouse.StorageConnector(); // Создаем экземпляр
 объекта соединения
 // Выполняем подключение к базе Mobile SMARTS
connector.SelectCurrentApp("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-c7c7003887c5");
try
{
// Получаем из базы все завершенные документы поступления
var documents = connector.GetDocuments("Поступление", true);
 //Обходим в цикле полученные документы и обрабатываем каждый документ foreach(var
document in documents)
```

```
{
var acceptanceDoc = AcceptanceJournal.Find(document.Id); //Ищем в учетной системе
исходный //документ приемки, который был выгружен в Mobile SMARTS
if(acceptanceDoc == null) //Если документ не найден, создаем новый acceptanceDoc =
AcceptanceJournal.CreateDocument();
else
acceptanceDoc.InventItems.Clear(); //Очистим строки, чтобы загрузить фактически
набранный //товар
 //Получаем поля шапки документа Mobile SMARTS
string contractorId = document.GetField("ИдКонтрагента") as string; string warehouseId
= document.GetField("ИдСклада") as string;
 //На основании полей шапки документа
 Mobile SMARTS заполняем
 поля шапки документа учетной системы if(!String.IsNullOrEmpty(contractorId))
{
acceptanceDoc.Contractor = ContractorCatalog.FindById(contractorId);
}
if(!String.IsNullOrEmpty(warehouseId))
{
acceptanceDoc.Warehouse = WarehouseCatalog.FindById(warehouseId);
}
 //Обходим в цикле фактические строки документа
Mobile SMARTS foreach(var documentItem in document.CurrentItems)
{
//Получаем из строки документа Mobile SMARTS Ид. товара, Ид. упаковки и
 количество string productId = documentItem.ProductId;
string packingId = documentItem.PackingId;




decimal currentQuantity = documentItem.CurrentQuantity;

//Получаем дополнительные поля строки документа
string descr = documentItem.GetField("Характеристика"); decimal price =
documentItem.GetField("Цена");
 //Находим в учетной системе
 товар, единицу и характеристику var inventItem = InventItemCatalog.FindById(productId); var
unit = UnitCatalog.FindByName(packingId);
 var inventDim = (from inventDim in InventDimCatalog
where inventDim.Owner == inventItem && inventDim.Name == descr select
inventDim).FirstOrDefault();
 //Ищем строку документа учетной системы
 с полученными товаром, единицей и характеристикой var acceptanceDocItem = (from
docItem in acceptanceDoc.InventItems
where docItem.InventItem == inventItem
&& docItem.Unit == unit &&
 docItem.InventDim == inventDim
select docItem).FirstOrDefault();
 if(acceptanceDocItem != null)
{
//Если строка найдена, прибавляем количество acceptanceDocItem.Quantity +=
```

```
currentQuantity;
}
else
{
//Строка не найдена, добавляем строку и заполняем в ней нужные поля
acceptanceDocItem = acceptanceDoc.InventItems.AddNew(); acceptanceDocItem.InventItem =
inventItem;
acceptanceDocItem.Unit = unit; acceptanceDocItem.InventDim = inventDim;
acceptanceDocItem.Price = price; acceptanceDocItem.Quantity = currentQuantity;
}
}
 //Все строки загрузили, сохраняем документ acceptanceDoc.Save();
 //Удаляем из базы Mobile SMARTS загруженный документ
connector.RemoveDocument(document.Id);
}
}
catch
{
//Обработка ошибки



}
```

**«1С:Предприятие 8»:**

```
connector = новый COMОбъект("Cleverence.Warehouse.StorageConnector"); //
Создаем экземпляр объекта
//соединения Попытка
connector.УстановитьПодключениеСБазойСМАРТС("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-
c7c7003887c5");
//Выполняем подключение
 // Получаем из базы все завершенные документы поступления ДокументыТСД =
connector.ПолучитьДокументы("Поступление", Истина);



//Обходим в цикле полученные документы и обрабатываем каждый документ


  Для ИндДок = 0 По ДокументыТСД.Количество-1 Цикл ДокументТСД =
ДокументыТСД.Элемент(ИндДок);


Документ1С = НайтиДокумент1СПоИдТСД(ДокументТСД.Ид); //Получаем документ 1С по
ид. документа ТСД. //При выгрузке из 1С назначали ид. вида
“Поступление#”+Документ1С.Номер, для созданного на //терминале документ 1С =
Неопределено, создаем новый документ 1С
```

```
Если Документ1С = Неопределено
Тогда

Документ1С = Документы.ПоступлениеТоваров.СоздатьДокумент(); Иначе

Документ1С.Товары.Очистить(); КонецЕсли;



//Получаем поля шапки документа Mobile SMARTS КодКонтрагента =
ДокументТСД.ПолучитьПоле("ИдКонтрагента"); КодСклада =
ДокументТСД.ПолучитьПоле("ИдСклада");



//На основании полей шапки документа Mobile SMARTS заполняем поля шапки
документа 1С Если
ЗначениеЗаполнено(КодКонтрагента) Тогда

Документ1С.Контрагент =
 Справочники.Контрагенты.НайтиПоКоду(КодКонтрагента); КонецЕсли;



Если ЗначениеЗаполнено(КодСклада) Тогда

Документ1С.Склад = Справочники.Склады.НайтиПоКоду(КодСклада); КонецЕсли;



//Обходим в цикле фактические строки документа Mobile SMARTS Для ИндСтроки = 0 По
ДокументТСД.СтрокиФакт.Количество-1 Цикл

СтрокаДокументаТСД = ДокументТСД.СтрокиФакт.Элемент(ИндСтроки); //Получаем строку
документа
```

```
//Получаем из строки документа Mobile SMARTS Ид. товара, Ид. упаковки и
 количество ИдТовара  =  СтрокаДокументаТСД.ИдТовара;


ИмяУпаковки  =  СтрокаДокументаТСД.ИдУпаковки; Количество  =
СтрокаДокументаТСД.КоличествоФакт;




//Получаем дополнительные поля строки документа


ХарактеристикаИмя  =  СтрокаДокументаТСД.ПолучитьПоле(“Характеристика”); Цена  =
СтрокаДокументаТСД.ПолучитьПоле(“Цена”);




//Находим в 1С товар, единицу и характеристику


Номенклатура  =  Справочники.Номенклатура.ПолучитьСсылку(новый
УникальныйИдентификатор(ИдТовара  ));




Упаковка  =
Справочники.ЕдиницыИзмерения.НайтиПоНаименованию(ИмяУпаковки,,,Номенклатура);




Если  ЗначениеЗаполнено(ХарактеристикаИмя)  Тогда


Характеристика  =  Справочники.Характеристики.НайтиПоНаименованию(ХарактеристикаИмя
,,,Номенклатура); КонецЕсли;




//Ищем строку документа 1С с полученными товаром, упаковкой и характеристикой
ПараметрыОтбора  =  Новый  Структура;


ПараметрыОтбора.Вставить("Номенклатура",
 Номенклатура); ПараметрыОтбора.Вставить("Упаковка",
 Упаковка); ПараметрыОтбора.Вставить("Характеристика",  Характеристика);
```

```
НайденныеСтроки = Документ1С.Товары.НайтиСтроки(ПараметрыОтбора);



Если НайденныеСтроки.Количество() > 0 Тогда СтрокаТабличнойЧасти =
НайденныеСтроки[0];


//Если строка найдена, прибавляем количество


СтрокаТабличнойЧасти.Количество = СтрокаТабличнойЧасти.Количество + Количество;
Иначе
//Строка не найдена, добавляем строку и заполняем в ней нужные поля
СтрокаТабличнойЧасти = Документ1С.Товары.Добавить();
СтрокаТабличнойЧасти.Номенклатура = Номенклатура; СтрокаТабличнойЧасти.Упаковка =
Упаковка; СтрокаТабличнойЧасти.Характеристика = Характеристика;
СтрокаТабличнойЧасти.Количество = Количество; СтрокаТабличнойЧасти.Цена = Цена;


КонецЕсли;



//Записываем документ в
 1С Документ1С.Записать(РежимЗаписиДокумента.Проведение);



//Удаляем загруженный документ Mobile SMARTS из базы
connector.УдалитьДокумент(ДокументТСД.Ид);


КонецЦикла; КонецЦикла;


Исключение


// Обработка

ошибки



КонецПопытки;
```

COM-компонента, обмен данными

## Не нашли что искали?

Задать вопрос в техническую поддержку