# **Пример запроса к базе MSSQL с помощью коннектора из Mobile SMARTS**

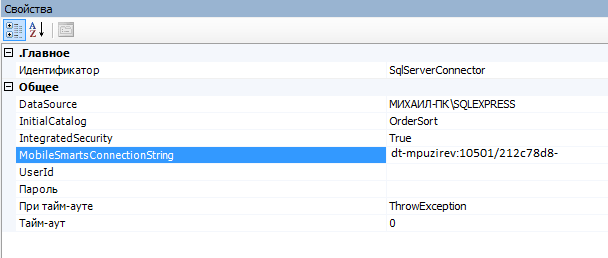
Платформа Mobile SMARTS интегрирована с MSSQL.

В комплекте поставки есть специальный коннектор, который позволяет обращаться напрямую к базе данных с терминала сбора данных. Это даёт возможность использовать решение на основе платформы Mobile SMARTS совместно с MSSQL без необходимости самостоятельно писать средства обмена данными.

Кроме хранимых процедур, которые должны возвращать xml, можно использовать маппинг таблиц базы SQL на объекты Mobile SMARTS.

[**Скачать пример базы Mobile SMARTS с настроенным маппингом и базы SQL**](https://disk.yandex.ru/d/ZwMwsyaCuuHH3g)

Для работы с базой SQL Server в конфигурацию Mobile SMARTS нужно [добавить коннектор к SQL Server](https://www.cleverence.ru/support/2756/) и настроить подключение к базе.

[](https://www.cleverence.ru/upload/medialibrary/ca9/t38surkp5k5l7wagcr0c3l2zmztly84o/content_img.png)

Для получения данных из базы можно использовать действие «[Вызов внешней системы](https://www.cleverence.ru/support/2539/)». Также есть возможность с помощью коннектора обрабатывать [события сервера Mobile SMARTS](https://www.cleverence.ru/support/category:601/). Для получения номенклатуры онлайн по запросу с терминала используется событие «[Получить товар](https://www.cleverence.ru/support/72168/?cat=601)».

После этого создайте в базе SQL хранимую процедуру:

CREATE PROCEDURE [dbo].[GetProduct]

(

@productId varchar(50),

@packingId varchar(50),

@userId varchar(50),

@mode int,

@resultXml xml OUTPUT

)

AS

BEGIN

...

END

В обработчике события «[Получить товар](https://www.cleverence.ru/support/72168/?cat=601)» укажите SqlServerConnector:serverEvent\_GetProduct (SqlServerConnector — ID коннектора, serverEvent\_ — указание на обработчик серверного события). Процедура должна возвращать через параметр @resultXml xml-файл особого вида, который представляет собой сериализованные объекты Mobile SMARTS. В случае режима 1 (поиск по ШК или другим атрибутам товара) это объекты Cleverence.Warehouse.PackedProduct (если найден один товар) или Cleverence.Warehouse.PackedProductCollection (несколько объектов), [подробнее здесь](http://www.cleverence.ru/devlib/TerminalConnector_v3/index.html). В процедуре нужно создать соответствующие COM-объекты, получить xml с помощью вызова ToXml объекта StorageConnector, можно также самим сформировать нужный xml, пример:

IF @mode = 1

BEGIN

SET @resultXml = '<?xml version="1.0" encoding="windows-1251"?>

<PackedProduct expiredDate="0001-01-01T00:00:00" quantity="1" registrationDate="0001-01-01T00:00:00"><Fields />

<Product barcode="" basePackingId="шт" id="cbcf492a-55bc-11d9-848a-00112f43529a" marking="Х-1234" name="BOSCH">

<Packings><Packing barcode="888999" id="шт" marking="" Int32\_qty="0" Int32\_price="27960" currency="RUB"></Packing></Packings>

</Product>

<Packing barcode="888999" id="шт" marking="" Int32\_qty="0" Int32\_price="27960" currency="RUB"></Packing>

<Quantities />

</PackedProduct>'

END

При попадании завершенного документа с ТСД на сервер Mobile SMARTS вызывается событие «ДокументЗавершен» — с помощью обработчика этого события можно загрузить документ в базу SQL. Есть два варианта обработчика для SQL

|  |  |
| --- | --- |
| 1. SqlServerConnector:serverEvent\_DocumentCompleted) | 2. SqlServerConnector:serverEvent\_DocumentCompletedXml |
| CREATE PROCEDURE [dbo].[DocumentCompleted]  (  @documentId varchar(50),  @resultXml xml OUTPUT  )  AS  BEGIN  /\* | CREATE PROCEDURE [dbo].[DocumentCompletedXml]  (  @documentXml xml,  @result int OUTPUT  )  AS  BEGIN  SET @result = 1;  END |
| Нужно создать COM-объект Cleverence.Warehouse.StorageConnector, выполнить подключение к базе Mobile SMARTS c помощью SelectCurrentApp (<строка подключения к базе Mobile SMARTS или ID базы>), загрузить с сервера документ при помощи GetDocument (@documentId), обработать документ и, если загрузка выполнилась успешно, удалить с сервера документ при помощи RemoveDocument (@documentId).  \*/  SET @resultXml = '';  END | documentXml содержит xml документа Mobile SMARTS, который можно разобрать внутри процедуры.  Если через result возвращается 1, коннектор удалит документ с сервера. |

Для получения списка документов для работы на терминале и выгрузке из базы SQL используются события «[ПолучитьСписокДокументов](https://www.cleverence.ru/support/2767/?cat=601)» и «[ПолучитьДокумент](https://www.cleverence.ru/support/2766/?cat=601)».

Примеры процедур:

CREATE PROCEDURE [dbo].[GetDocumentsList]

(

@userId varchar(50),

@documentTypeName varchar(50),

@resultXml xml OUTPUT

)

AS

BEGIN

SET @resultXml = '<?xml version="1.0" encoding="windows-1251"?>

<DocumentDescriptionCollection xmlns:clr="http://schemas.cleverence.ru/clr">

<DocumentDescription barcode="123" createDate="0001-01-01T00:00:00" distributeByBarcode="True" documentTypeName="Инвентаризация" id="123" name="Инвентаризация 123" warehouseId="1"><Fields />

</DocumentDescription>

</DocumentDescriptionCollection>';

END

CREATE PROCEDURE [dbo].[GetDocument]

(

@userId varchar(50),

@documentTypeName varchar(50),

@documentId varchar(50),

@mode int,

@resultXml xml OUTPUT

)

AS

BEGIN

SET @resultXml = '<?xml version="1.0" encoding="windows-1251"?>

<Document xmlns:clr="http://schemas.cleverence.ru/clr" barcode="00003" createDate="2014-03-06T16:53:33" deviceId="" deviceIP="" deviceName="" documentTypeName="Инвентаризация" id="123" name="Инвентаризация товаров на складе РТЦУТД00003 от 26.06.2008 11:22:45" appointment="оператор" userId="оператор" userName="оператор" warehouseId="1">

<DeclaredItems>

<DocumentItem declaredQuantity="8" expiredDate="0001-01-01T00:00:00" packingId="пара" productId="dee6e1d0-55bc-11d9-848a-00112f43529a" registeredDate="0001-01-01T00:00:00"><Fields capacity="4"><FieldValue fieldName="price"><Value clr:Type="Int32">1424</Value></FieldValue></Fields></DocumentItem>

<DocumentItem declaredQuantity="20" expiredDate="0001-01-01T00:00:00" packingId="3" productId="dee6e1d0-55bc-11d9-848a-00112f43529a" registeredDate="0001-01-01T00:00:00"><Fields capacity="4"><FieldValue fieldName="descr"><Value clr:Type="String">6, 39, Зеленый</Value></FieldValue><FieldValue fieldName="price"><Value clr:Type="Int32">1424</Value></FieldValue></Fields></DocumentItem>

<DocumentItem declaredQuantity="9" expiredDate="0001-01-01T00:00:00" packingId="пара" productId="dee6e1d3-55bc-11d9-848a-00112f43529a" registeredDate="0001-01-01T00:00:00"><Fields capacity="4"><FieldValue fieldName="price"><Value clr:Type="Int32">1780</Value></FieldValue></Fields></DocumentItem>

<DocumentItem declaredQuantity="20" expiredDate="0001-01-01T00:00:00" packingId="3" productId="dee6e1d3-55bc-11d9-848a-00112f43529a" registeredDate="0001-01-01T00:00:00"><Fields capacity="4"><FieldValue fieldName="descr"><Value clr:Type="String">6, 39, Белый</Value></FieldValue><FieldValue fieldName="price"><Value clr:Type="Int32">1780</Value></FieldValue></Fields></DocumentItem>

</DeclaredItems><CurrentItems capacity="0" />

<ClassificatorIds capacity="0" />

<Classificators capacity="0" />

<ClassificatorUsings capacity="0" /><Errors capacity="0" />

<Fields capacity="4"><FieldValue fieldName="КонтрольКолва"><Value clr:Type="Int32">0</Value></FieldValue><FieldValue fieldName="ПоЯчейкам"><Value clr:Type="Int32">0</Value></FieldValue>

</Fields>

<Tables capacity="0" />

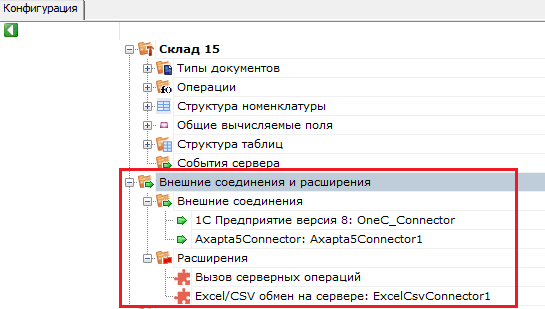
</Document>'

END

Также не обязательно формировать xml вручную, можно использовать COM-объекты Mobile SMARTS.

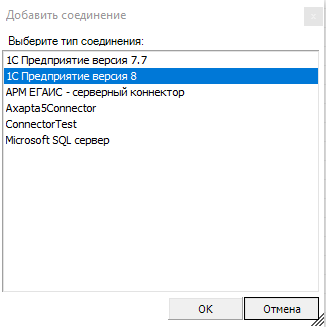
# **Регистрация и запуск коннекторов**

Для того, чтобы начать использовать коннектор, необходимо сначала его зарегистрировать (добавить), а затем запустить. Эти операции выполняются в [панели управления Mobile SMARTS](https://www.cleverence.ru/support/category:67/) и находятся внутри узла «Внешние соединения и расширения».

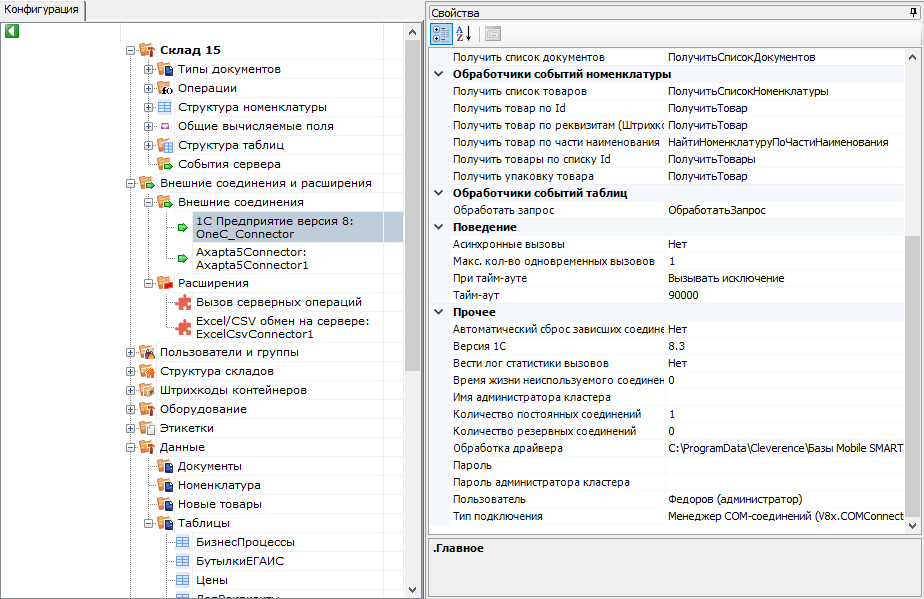
[](https://www.cleverence.ru/upload/medialibrary/136/1367b0752f836f6a642e1b2c9089e869.png)

Регистрация коннектора

Добавление в конфигурацию нового коннектора выполняется с помощью щелчка правой кнопкой мыши на узле «Внешние соединения», в контекстном меню нужно выбрать «Добавить внешнее соединение...», в списке выбрать нужный модуль и нажать ОК:

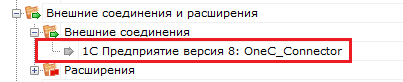
[](https://www.cleverence.ru/upload/medialibrary/cf3/cf3a2421c53ed5a31385623945a04777.png)

Настройка параметров коннектора выполняется через боковую панель свойств:

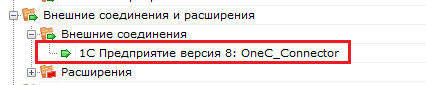
[](https://www.cleverence.ru/upload/medialibrary/0af/0afb72812fd0250cafb72edf6d387567.png)

После добавления коннектора и настройки параметров нужно сохранить конфигурацию.

Если узел коннектора в дереве конфигурации отображается в виде серой стрелки, то это означает, что вызовы через коннектор запрещены (отключены):

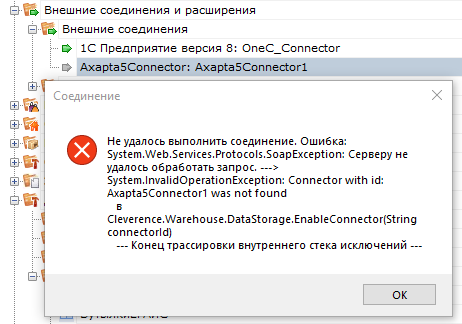
[](https://www.cleverence.ru/upload/medialibrary/830/830f868deac3f1634c006384d61fd414.png)

Для того, чтобы разрешить вызовы, нажмите правой кнопкой мыши на узле дерева, выберите в контекстном меню «Разрешить». Модуль должен перейти в разрешенное состояние (зеленая стрелка):

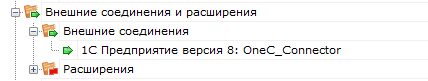
[](https://www.cleverence.ru/upload/medialibrary/10a/10a40e009aeace7eac2ff0208b4827bd.png)

Для работы модуля под управлением сервера Mobile SMARTS и настройки параметров модуля через панель управления используются две версии библиотеки (файлы dll). Первая dll, предназначенная для сервера, размещается в <Папка базы Mobile SMARTS>\Server\DataService\bin\. Вторая dll, для панели управления - в <Папка базы Mobile SMARTS>\Control panel\Addins. Некоторые стандартные модули, входящие в дистрибутив платформы Mobile SMARTS, не требуют дополнительных файлов dll (см. [Список существующих стандартных  модулей](https://www.cleverence.ru/support/2754/)).

Если библиотека модуля для панели управления загружена, а для сервера библиотека отсутствует (или не была загружена в процесс сервера Mobile SMARTS), то при сохранении конфигурации Mobile SMARTS после добавления модуля в панели управления данные о добавленном модуле не запишутся в файлы конфигурации. При попытке разрешить вызовы через коннектор будет происходить ошибка.

[](https://www.cleverence.ru/upload/medialibrary/8e6/8e6386a5410ec4cd0d7ed02bb99d8735.png)

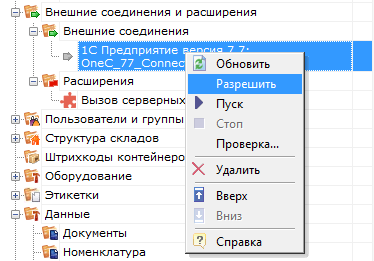
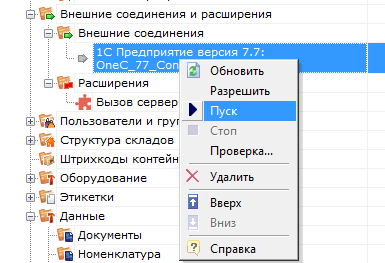
Если загрузить конфигурацию с сервера (используя кнопку «Обновить» в панели управления), отображаются данные о том, что добавленного модуля нет:

[](https://www.cleverence.ru/upload/medialibrary/57b/57b94f47e96cb53c7144ac8ad39663c7.png)

Для устранения ошибки необходимо переместить серверную версию dll модуля в <Папка базы Mobile SMARTS>\Server\DataService\bin\, перезапустить сервер базы данных Mobile SMARTS и заново добавить модуль через панель управления.

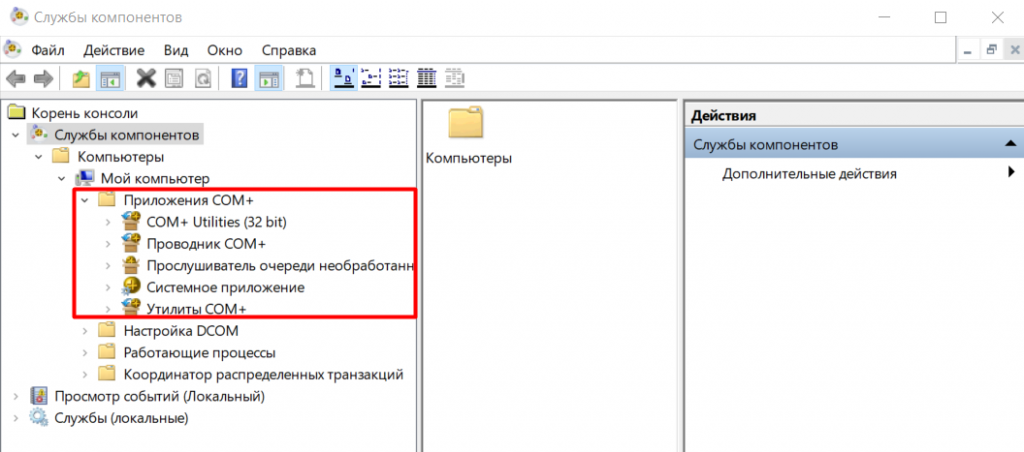
Запуск коннектора

После регистрации коннектора следует проверить, выполняется ли подключение. Для этого следует разрешить доступ к коннектору («Разрешить»), и затем запустить его («Пуск»).

[](https://www.cleverence.ru/upload/medialibrary/13d/13dc18a6624e685e11498ba1e8caa7b4.png)  [](https://www.cleverence.ru/upload/medialibrary/e17/e17e504798277622f7b9b9b83526c586.png)

В ряде случаев случаев попытка запуска завершится неудачей, т.к. у [сервера Mobile SMARTS](https://www.cleverence.ru/support/13831/) не будет хватать пользовательских прав на подключение к внешней системе. Основные причины таких проблем и способы их решения описаны в статье [«Диагностика и исправление проблем»](https://www.cleverence.ru/support/61970/) на сайте «Клеверенс».

Если в «Службах компонентов» есть установленные вручную COM+ приложения, это может помешать запуску COM-коннектора. В таком случае потребуется удалить COM+ приложения, после чего заново [зарегистрировать](https://www.cleverence.ru/support/43817/) и [запустить](https://www.cleverence.ru/support/2756/) COM-коннектор.

[](https://www.cleverence.ru/upload/medialibrary/295/content_img.png)

Свойства коннектора

Microsoft SQL Server

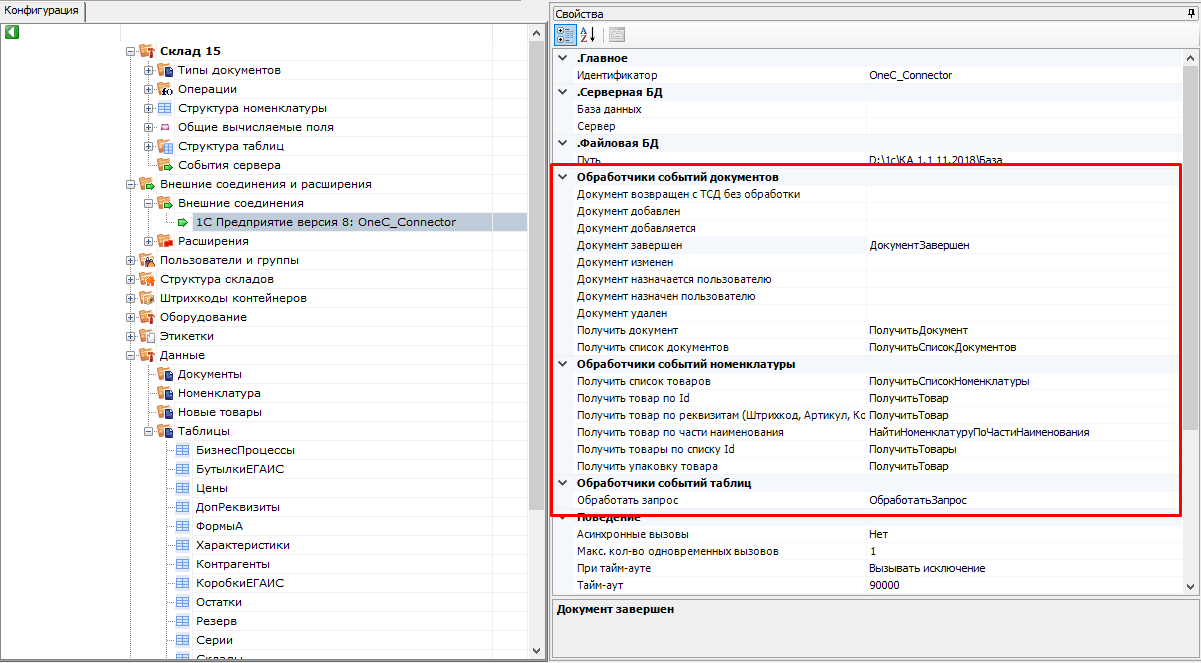
|  |  |  |  |
| --- | --- | --- | --- |
| **Свойство** | **Заполняется автоматически** | **Принимаемые значения** | **Расшифровка** |
| **DataSource** | - | Произвольная строка | Имя или сетевой адрес SQL-сервера |
| **InitialCatalog** | - | Произвольная строка | Имя базы данных на SQL-сервере |
| **IntegratedSecurity** | True | True/ False | True — для подключения используется текущая учетная запись Windows False — используется заданные имя пользователя базы SQL и пароль пользователя |
| **UserID** | - | Произвольная строка | Идентификатор пользователя базы SQL |
| **Пароль** | - | Произвольная строка | Пароль пользователя базы SQL |
| **При тайм-ауте** | - | Вызывать исключение/ Переподключаться | Поведение при истечении тайм-аута. Вызывать исключение — выполнение вызова останавливается и возвращается ошибка Переподключаться — отключить от системы и повторить вызов |
| **Тайм-аут** | - | Произвольное число (в мс) | При истечении указанного времени вызов будет прерван и выполнится действие, указанное в пункте «При тайм-ауте» |

# **Для самостоятельной разработки/дорабоотки коннектора**

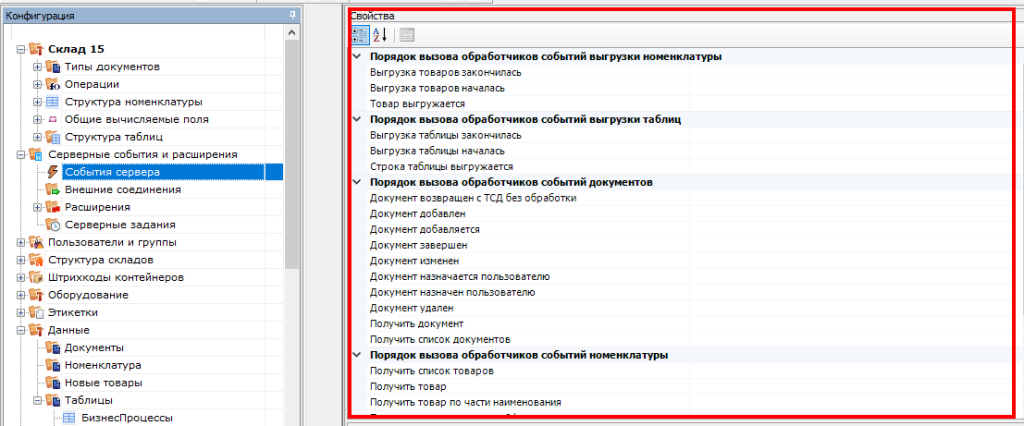
# **События сервера Mobile SMARTS**

События сервера Mobile SMARTS — специальный механизм, позволяющий выполнять вызов функций [внешних соединений](https://www.cleverence.ru/support/category:612/) (коннекторов) при наступлении определенных событий (например, запрос номенклатуры по идентификатору или штрихкоду с ТСД, получение сервером с ТСД завершенного документа и др.). При вызове функции-обработчика события коннектор, как правило, выполняет обращение к внешней системе, передавая параметры события во внешнюю систему (например, штрихкод, по которому нужно найти номенклатуру или идентификатор завершенного документа, по которому документ можно получить из базы Mobile SMARTS), внешняя система возвращает результат в требуемом виде (например, объект «Товар с упаковкой» PackedProduct, сериализованный в xml), коннектор передает результат серверу для возврата его на ТСД (подробнее написано в статье «[Основной принцип работы коннекторов](https://www.cleverence.ru/support/2753/)»). В некоторых случаях логика обработки определенных событий может быть реализована непосредственно в самом коннекторе, без вызова внешней системы (например, при добавлении в базу документа для обработки на ТСД коннектор может выполнить построение дополнительных таблиц документа или запретить добавление документа, если не заполнены определенные поля).

Настройка обработчиков событий сервера выполняется через [панель управления](https://www.cleverence.ru/support/category:67/). Для большинства стандартных коннекторов, входящих в дистрибутив Mobile SMARTS, функции-обработчики указываются в параметрах самого коннектора:

[](https://www.cleverence.ru/upload/manager/012/6ro3wdcjjw3u2xkv1twzxjoecs61yd2w/content_img.png)

Указание обработчиков событий сервера в свойствах коннектора позволяет организовать вызов нескольких обработчиков для одного события. Порядок вызова обработчиков редактируется в [панели управления](https://www.cleverence.ru/support/category:67/) в свойствах узла «События сервера»:

[](https://www.cleverence.ru/upload/medialibrary/134/58ehmfcs4b3kkt1y3utcp0xrizmfdep6/content_img.png)

**Примечание.** Задание имени функции-обработчика события в параметрах коннектора не является обязательным условием для вызова обработчика. В платформе Mobile SMARTS версий до 3.1 (3.0.xx.xxxx) обработчики событий указывались только в узле «События сервера» в формате «Ид коннектора:Имя функции-обработчика», и коннекторы не имели параметров для задания обработчиков. Такой механизм поддерживается и в версиях 3.1.xx.xxxx. Т. е. если используется коннектор, разработанный для платформы 3.0, не имеющий параметров для указания обработчиков, и обработчик задан в узле «События сервера», вызов обработчика сервером будет выполняться.

Параметры, передаваемые в функцию-обработчик, для каждого события, фиксированные, и функция для успешного вызова должна соответствовать этим параметрам.

### **События сервера**

|  |  |
| --- | --- |
| **Название** | **Описание** |
| **События документов** | |
| [ДокументДобавляется](https://www.cleverence.ru/support/72151/) | Событие, возникающее в процессе добавления документа на сервере. Вызывается при выгрузке документа в базу Mobile SMARTS из внешней системы. |
| [ДокументДобавлен](https://www.cleverence.ru/support/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%94%D0%BE%D0%B1%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD/) | Событие, возникающее после добавления документа на сервере. Вызывается при выгрузке документа в базу Mobile SMARTS из внешней системы. |
| [ДокументНазначаетсяПользователю](https://www.cleverence.ru/support/72152/) | Событие,сообщающее о том что документ готов передаваться на мобильное устройство. Вызывается при запросе с ТСД получения документа для работы. |
| [ДокументНазначенПользователю](https://www.cleverence.ru/support/72166/) | Событие о том, что документ захвачен на обработку. Вызывается в момент, когда документ был передан на ТСД для работы с ним. |
| [ДокументИзменен](https://www.cleverence.ru/support/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%98%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD/) | Событие об изменении документа. Вызывается при сохранении документа на сервер в процессе работы на ТСД при использовании в конфигурации Mobile SMARTS действия «Сохранение документа на сервер». |
| [ДокументЗавершен](https://www.cleverence.ru/support/2760/) | Событие о завершении обработки документа. Вызывается при получении сервером завершенного документа с ТСД. |
| [ДокументВозвращенСТсдБезОбработки](https://www.cleverence.ru/support/72167/) | Вызывается в случае если документ с терминала был возвращен пользователем вызовом release (вернуть документ без изменений). |
| [ДокументУдален](https://www.cleverence.ru/support/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%A3%D0%B4%D0%B0%D0%BB%D0%B5%D0%BD/) | Событие об удалении документа с сервера. Удаление выполняется внешней системой. |
| [ПолучитьДокумент](https://www.cleverence.ru/support/%D0%9F%D0%BE%D0%BB%D1%83%D1%87%D0%B8%D1%82%D1%8C%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82/) | Вызывается при выборе в списке документов на ТСД того документа, который еще не был выгружен (попал в этот список с помощью события «Получить список документов»). |
| [ПолучитьСписокДокументов](https://www.cleverence.ru/support/%D0%9F%D0%BE%D0%BB%D1%83%D1%87%D0%B8%D1%82%D1%8C%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2/) | Вызывается при входе в список документов на ТСД. Используется в паре с событием «Получить документ» для того, чтобы реализовать выбор документов прямо из учетной системы, без предварительной выгрузки. |
| **События номенклатуры** | |
| [ПолучитьТовар](https://www.cleverence.ru/support/72168/) | Событие о том, что продукт был запрошен терминалом по коду (штрихкод, артикул, код) с сервера и не был найден в серверном справочнике. |
| [ПолучитьТоварыПоСпискуId](https://www.cleverence.ru/support/72169/) | Событие о том, что несколько товаров были запрошены терминалом по идентификаторам с сервера и не были найден в серверном справочнике. Позволяет реализовывать получение сразу нескольких товаров из базы учетной системы без предварительной выгрузки. |
| [ПолучитьТоварПоЧастиНаименования](https://www.cleverence.ru/support/32470/) | Вызывается при вводе в строку поиска в списке номенклатуры на ТСД. Используется для того, чтобы реализовать поиск номенклатуры по части наименования прямо из учетной системы. |
| [ПолучитьСписокТоваров](https://www.cleverence.ru/support/32469/) | Вызывается при входе в список номенклатуры на ТСД. Используется для того, чтобы реализовать выбор номенклатуры прямо из учетной системы, без предварительной выгрузки. |
| **События таблиц** | |
| [ОбработатьЗапрос](https://www.cleverence.ru/support/32488/) | Событие вызывается при запросе данных из таблицы, определенной в конфигурации Mobile SMARTS. Позволяет получать строки таблицы онлайн по запросу с ТСД. |

# **Событие сервера Mobile SMARTS «ДокументДобавляется»**

Событие, возникающее в процессе добавления документа на сервере. Вызывается при выгрузке документа в базу Mobile SMARTS из внешней системы.

Параметры

|  |  |  |
| --- | --- | --- |
| **Имя параметра** | **Тип данных** | **Описание** |
| ИдДокумента | string (строка) | Код документа, который выгружается на сервер. |
| ИдПользователя | string (строка) | Передается null |
| XmlСессии | string (строка) | Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий». |

Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию передается сам документ (переменная сессии Document). Если выполнение серверной операции прерывается (переход [прервать операцию]), добавление документа на сервер отменяется. Если отмена добавления не требуется, операция должна завершиться по переходу [завершить операцию].

В случае обычных коннекторов, если обработчик события вернул null, добавление документа на сервер отменяется.

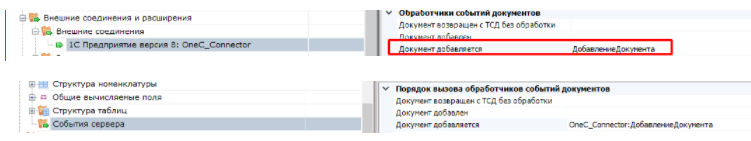
Описание в [панели управления](https://www.cleverence.ru/support/category:67/):

C#

<ид. коннектора>:ДобавлениеДокумента

Ид. коннектора — задается в панели управления.

Например: OneC\_DriverConnector:ДобавлениеДокумента

[](https://www.cleverence.ru/upload/manager/873/notmkr5tajw1dvmrd5o34epupj86zej4/content_img.png)

# **Событие сервера Mobile SMARTS «ДокументДобавлен»**

Событие, возникающее после добавления документа на сервере. Вызывается при выгрузке документа в базу Mobile SMARTS из внешней системы.

Параметры

|  |  |  |
| --- | --- | --- |
| **Имя параметра** | **Тип данных** | **Описание** |
| ИдДокумента | string (строка) | Код документа, который выгружен на сервер. |
| ИдПользователя | string (строка) | Передается null. |
| XmlСессии | string (строка) | Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий». |

Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию передается сам документ (переменная сессии Document).

Обработчик может внести изменения в документ. После вызова обработчика документ сохраняется на сервере.

Описание в  [панели управления](https://www.cleverence.ru/support/category:67/):

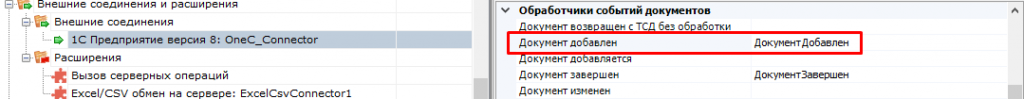
C#

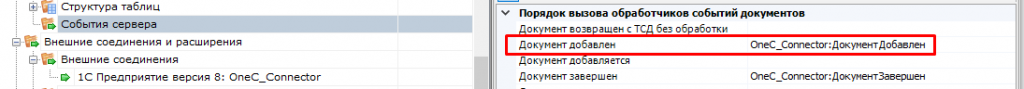
<ид. коннектора>:ДокументДобавлен

Описание в панели управления:

Ид. коннектора - задается в панели управления.

Например: OneC\_Connector:ДокументДобавлен

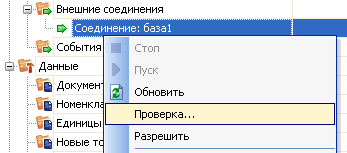
[](https://www.cleverence.ru/upload/medialibrary/563/56384ab62fd83733be11d61cde80ae85.png)

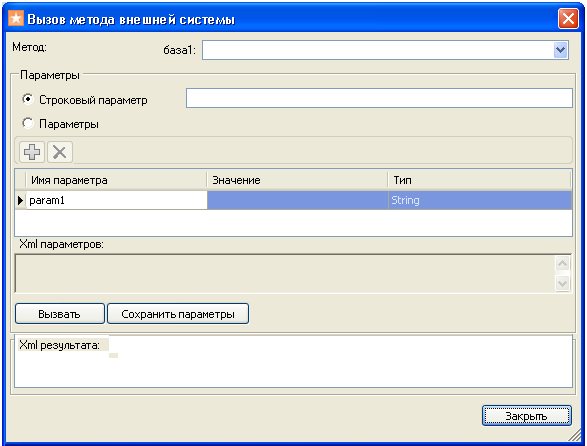
[](https://www.cleverence.ru/upload/medialibrary/59f/59f7ba90b55f4c789c7f22271ea63e3d.png)

# **Отладка вызовов к внешней системе через коннекторы**

Когда оператор терминала сбора данных сканирует штрихкод для запроса остатков товара, со стороны внешней системы (учетной системы, товарной базы) кто-то должен обработать этот запрос и вернуть результат. В случае 1С это будет метод глобального контекста или контекста внешнего соединения, для Axapta это будет публичный статический метод в каком-то классе, а в случае SQL базы данных это может быть хранимая процедура (хотя можно выполнить и простой запрос).

Для отладки работы кода внешней системы, который должен будет вызываться с ТСД, в панели управления Mobile SMARTS предусмотрено специальное окно, вызываемое из контекстного меню коннектора:

[](https://www.cleverence.ru/upload/iblock/c18/c18c80ade2cab00d009ffeba27e1a406.png)

[](https://www.cleverence.ru/upload/iblock/e2f/e2ffca0a26955ef2d39a2ebd052fc05c.png)

В этом окне можно задать имя вызываемого метода внешней системы, указать передаваемые параметры, вызвать внешнюю систему при помощи коннектора и посмотреть на результат.

Как видно, коннектору передаются именованные параметры. Метод внешней системы, который будет принимать эти параметры, может получить их в двух вариантах, в зависимости от реализации коннектора. Первый вариант – будет учитываться только порядок передачи параметров, а имена не имеют значения и отбрасываются. Второй вариант – метод принимает единственный строковой аргумент, в котором передается XML с сериализованным объектом Cleverence.Warehosue.InvokeArgs; это выглядит примерно как «<?xml version="1.0" encoding="utf-8" ?><InvokeArgs xmlns:clr="[http://schemas...»](http://schemas...xn--yba/).

# **Разработка коннекторов Mobile SMARTS к внешним системам**

[Коннекторы](https://www.cleverence.ru/support/category:612/) к внешним системам в платформе Mobile SMARTS позволяют получать данные из внешней системы по запросу с ТСД, а также выполнять обработку [событий сервера Mobile SMARTS](https://www.cleverence.ru/support/category:601/) с помощью вызова функций внешней системы.

В состав дистрибутива Mobile SMARTS входит [ряд коннекторов](https://www.cleverence.ru/support/2754/) к наиболее распространенным учетным системам и СУБД. Если Вы используете учетную систему или СУБД, для которой не имеется стандартного коннектора, есть возможность разработать свой собственный коннектор.

## Требования к разработке

Разработка коннектора выполняется в среде Microsoft Visual Studio на языке C# (можно использовать и другие языки, поддерживаемые платформой .Net Framework). Предполагается, что Вы знакомы с разработкой в среде Microsoft Visual Studio.

Мы рекомендуем использовать Microsoft Visual Studio 2019, скачать бесплатную версию Microsoft Visual Studio Community 2019 можно с [сайта Microsoft](https://visualstudio.microsoft.com/ru/license-terms/microsoft-visual-studio-community-2019/).

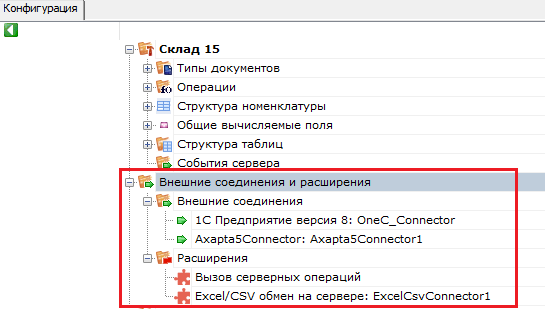
Разработка будет выполняться под .Net Framework 4.6.1, скачать дистрибутив можно [здесь](https://www.microsoft.com/ru-RU/download/details.aspx?id=49982). Если установлена более новая версия .Net Framework 4 (4.7 и др.), устанавливать отдельно версию 4.6.1 не требуется.

Последний релиз платформы Mobile SMARTS (MobileSMARTS.msi) можно скачать на [странице](https://www.cleverence.ru/downloads/distr/).

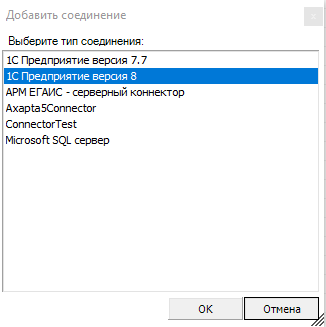
## Основы

Подключаемые модули платформы Mobile SMARTS разделяются на внешние соединения (коннекторы) и расширения (плагины). Коннекторы предназначены для вызова функций внешних систем по запросу с ТСД, с их помощью также может выполняться обработка событий сервера. Расширения (плагины) используются для выполнения какой-либо работы под управлением сервера Mobile SMARTS без обработки вызовов c ТСД и событий сервера (например, плагин может следить за содержимым заданной папки и загружать данные при изменении файлов или делать бэкап по расписанию).

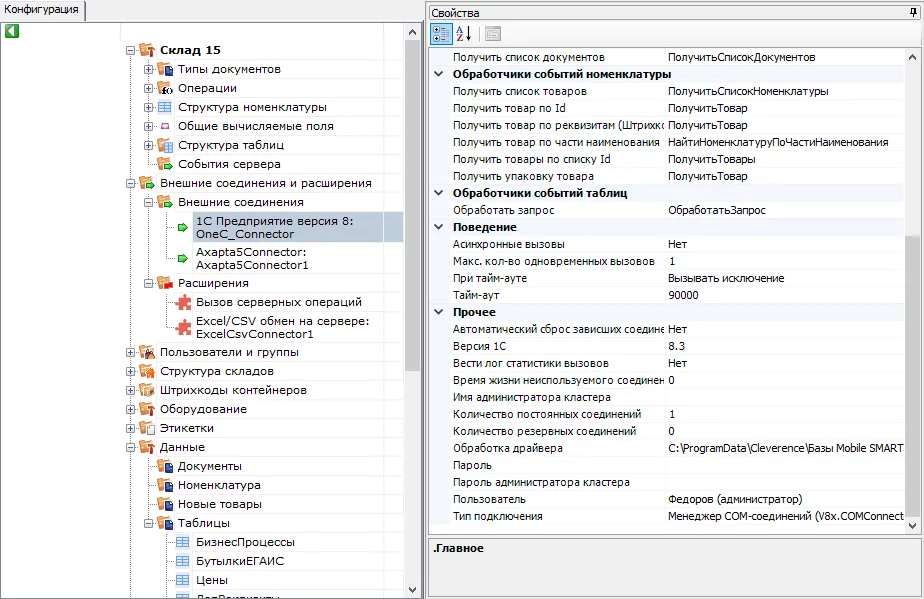
В [панели управления](https://www.cleverence.ru/support/category:67/) все подключаемые модули, добавленные в конфигурацию Mobile SMARTS, находятся внутри узла «Внешние соединения и расширения», разбитые на подгруппы «Внешние соединения» и «Расширения»:

[](https://www.cleverence.ru/upload/manager/721/l3lyq40koukpb1k2h9cqw6av0fqsyniu/content_img.png)

Добавление в конфигурацию нового модуля выполняется с помощью щелчка правой кнопкой мыши на узле «Внешние соединения» или «Расширения», в контекстном меню нужно выбрать «Добавить внешнее соединение…» (или «Добавить расширение…»), в списке выбрать нужный модуль и нажать ОК:

[](https://www.cleverence.ru/upload/manager/b75/rc5u14sfoedt2xc8w9xwi2ut4j11b4dr/content_img.png)

Настройка параметров модуля выполняется через боковую панель свойств:

[](https://www.cleverence.ru/upload/manager/a6e/d6zqcalsrqc47kagn5hfov0dsuvcv3mk/content_img.png)

После добавления модуля и настройки параметров нужно сохранить конфигурацию.

# **Объектная модель подключаемых модулей**

ассмотрим объектную модель подключаемых модулей. Базовым интерфейсом для всех подключаемых модулей является**IConnectivityBase**:

[С#]

/// <summary>

/// Базовый интерфейс для подключаемых модулей

/// </summary>

public interface IConnectivityBase

{

/// <summary>

/// Уникальный идентификатор модуля.

/// </summary>

string Id

{

get;

}

/// <summary>

/// Флаг, указывающий разрешена или нет работа модуля.

/// </summary>

bool Enabled

{

get;

set;

}

/// <summary>

/// Инициализация (запуск) модуля. После вызова этой функции он должен перейти в  рабочее состояние

/// </summary>

void Initialize();

/// <summary>

/// Инициализировано соединение или нет.

/// </summary>

bool Initialized

{

get;

}

/// <summary>

/// Функция проверки дополнительных лицензионных ограничений модуля

/// </summary>

param name="supportedSystems">Список внешних модулей, упомянутых в лицензионном файле</param>

void CheckLicenseLimitations(System.Collections.Generic.List<Cleverence.Licensing.LicenseExternalSystem>

supportedSystems);

}

От интерфейса**IConnectivityBase** наследуется интерфейс соединения с внешней системой**IConnecto**r и интерфейс расширения (плагина) **IPlugin**.

[С#]

/// <summary>

/// Интерфейс соединения с внешней системой. Важное отличие - наличие метода Invoke, позволяющего вызывать любые

какие-то методы из внешней системы

/// </summary>

public interface IConnector: IConnectivityBase

{

/// <summary>

/// Тайм-аут при вызове функций внешней системы через коннектор.

/// </summary>

int Timeout

{

get;

set;

}

/// <summary>

/// Поведение при наступлении тайм-аута. ThrowException - вызывать исключение, ReInvoke - завершить попытку вызова,

переинициализировать подключение и сделать снова вызов.

/// </summary>

TimeoutBehavaior TimeoutBehavaior

{

get;

set;

}

/// <summary>

/// Сообщает, что коннектор сам внутри обрабатывает таймауты, и внешняя обработка не требуется

/// </summary>

bool IsSelfTimeoutBehavior { get; }

/// <summary>

/// Для ActiveMQ коннектора добавляет в аргументы DeviceInfo.

/// </summary>

bool IsSupportDeviceInfoInArgs

{

get;

}

/// <summary>

/// Вызов метода внешней системы.

/// </summary>

/// <param name="methodName">Имя метода.</param>

/// <param name="args">Параметры.</param>

/// <returns>Результат (null, если метод ничего не возвращает).</returns>

object InvokeMethod(string methodName, object[] args);

}

/// <summary>

/// Базовый интерфейс плагина.

/// Немного расширяет стандартный IConnectivityBase явным методом остановки модуля.

/// </summary>

public interface IPlugin: IConnectivityBase

{

/// <summary>

/// Деинициализация (остановка) модуля. После вызова этой функции он должен остановить свою работу.

/// </summary>

void Deinitialize();

}

Основное отличие между [внешним соединением (коннектором)](https://www.cleverence.ru/support/category:965/) и расширением (плагином) в том, что коннектор имеет функцию InvokeMethod, с помощью которой выполняется вызов внешней системы при запросах с ТСД и при обработке событий сервера. Свойства Timeout, TimeoutBehavaior, IsSelfTimeoutBehavior позволяют установить тайм-аут при вызовах и задать поведение при наступлении тайм-аута. Плагин, в отличии от коннектора, не поддерживает вызовы. И коннектор и плагин запускаются при вызове функции Initialize [сервером Mobile SMARTS](https://www.cleverence.ru/support/category:601/). Вызов Initialize происходит при старте сервера [базы данных Mobile SMARTS](https://www.cleverence.ru/support/17309/) (если в настройках базы включена опция «Инициализировать подключения к внешним системам при старте») или при обращении к коннектору (вызов с ТСД или обработка [события сервера](https://www.cleverence.ru/support/category:601/)). Конкретная реализация Initialize определяется особенностями системы, к которой выполняется подключение. Например, может быть выполнено создание [COM-объекта](https://www.cleverence.ru/support/category:735/) для работы со внешней системой и выполнено подключение к базе данных системы с заданным логином/паролем. В случае плагина Initialize запускает модуль в работу. Например, может начаться слежение за папкой или запущен таймер для выполнения периодических операций. Плагин имеет функцию Deinitialize, при вызове которой работа плагина прекращается. Работа плагина может быть остановлена через [панель управления](https://www.cleverence.ru/support/category:67/) (контекстное меню узла плагина -> «Остановить»), также остановка происходит при остановке сервера базы данных Mobile SMARTS.

Еще одним интерфейсом, производным от**IConnectivityBase**, является **IEventsProcessor**:

[С#]

/// <summary>

/// Базовый интерфейс для модуля, умеющего обрабатывать события.

/// Для реализации обработки событий необходимо пользоваться

/// атрибутами EventProcessorAttribute, DocumentEventProcessorAttribute

/// </summary>

public interface IEventsProcessor: IConnectivityBase

{

}

Интерфейс не добавляет каких-либо функций в**IConnectivityBase**, а служит маркером для модулей, которые используются при обработке событий сервера.

Существует базовая реализация интерфейса**IConnectivityBase**, от которой можно наследоваться при реализации своих коннекторов — класс **ConnectorBase**.

Если разрабатываемый коннектор будет использоваться для обработки [событий сервера](https://www.cleverence.ru/support/category:601/), рекомендуется в качестве базового использовать другой класс — **ConnectorTypical**, производный от **ConnectorBase**. Данный класс позволяет указывать имена обработчиков событий сервера в свойствах самого коннектора.

[С#]

Код класса приведен не полностью

/// <summary>

/// Базовый класс коннектора, позволяющий указывать имена обработчиков событий в свойствах самого коннектора.

/// При обработке событий вызываются функции класса, помеченные атрибутом EventProcessor или DocumentEventProcessor.

/// При разработке новых коннекторов рекомендуется наследоваться от данного класса.

/// </summary>

public abstract class ConnectorTypical : ConnectorBase

{

/// <summary>

/// Имя функции-обработчика события "Документ добавляется".

/// Событие вызывается перед добавлением документа в базу Mobile SMARTS из внешней системы.

/// </summary>

[EventHandlerProperty(EventType.DocumentAdding)]

public string DocumentAddingHandler { get; set; }

…..

/// <summary>

/// Обработчик вызывается перед добавлением документа на сервер из внешней системы.

/// </summary>

/// <param name="di"></param>

/// <param name="documentTypeName"></param>

/// <param name="doc"></param>

/// <returns>Возвращается переданный документ или null, если требуется отклонить добавление документа (будет вызвано исключение и документ не будет сохранен на сервере).</returns>

[DocumentEventProcessor(DocumentType = "\*", EventType = EventType.DocumentAdding)]

public virtual object DocumentAdding(DeviceInfo di, string documentTypeName, Document doc)

...

}

Имена обработчиков в [панели управления](https://www.cleverence.ru/support/category:67/) указываются в свойствах коннектора:

Указание обработчиков событий сервера в свойствах коннектора позволяет организовать вызов нескольких обработчиков для одного события. Порядок вызова обработчиков редактируется в панели управления в свойствах узла «События сервера»:

Класс **ConnectorTypical** содержит виртуальные функции, вызываемые при обработке событий сервера, данные функции помечены атрибутом DocumentEventProcessor для событий документов и атрибутом EventProcessor для остальных событий (номенклатуры и таблиц). Например:

[С#]

<summary>

/// Обработчик вызывается сервером при запросе документа с ТСД.

/// </summary>

/// <returns>Возвращается документ (Document), может в сериализованном в xml-строку виде. Если документ не был получен, то возвращается null. Также может возвращаться InvokeResult.</returns>

[DocumentEventProcessor(DocumentType = "\*", EventType = EventType.GetDocument)]

public virtual object GetDocument(DeviceInfo di, string documentTypeName, string  identity, GetDocumentMode mode)

Атрибут DocumentEventProcessor имеет параметры: DocumentType, позволяет ограничить вызов данного обработчика определенными типами документов, «\*» — вызов будет выполнен для всех типов документов, EventType — тип события. Атрибут EventProcessor имеет один параметр: EventType — тип события. При реализации своих обработчиков в производных классах также следует использовать атрибуты DocumentEventProcessor и EventProcessor.

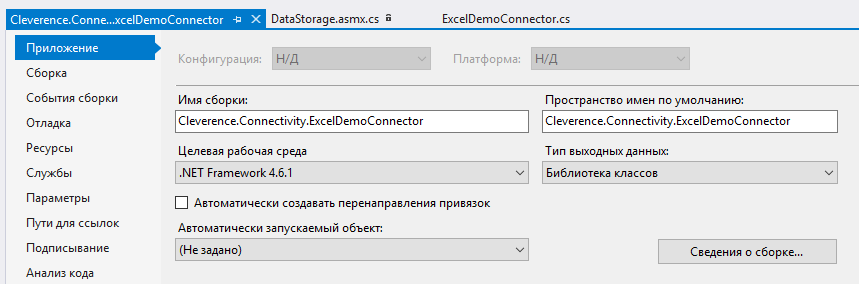
Типы событий, задаваемые перечислением EventType:

|  |  |
| --- | --- |
| **События документов** | |
| **DocumentAdding** | Документ добавляется. Событие, возникающее в процессе добавления документа на сервере. Вызывается при выгрузке документа в базу Mobile SMARTS из внешней системы. |
| **DocumentAdded** | Документ добавлен. Событие, возникающее после добавления документа на сервере. Вызывается при выгрузке документа в базу Mobile SMARTS из внешней системы. |
| **DocumentAppointing** | Документ назначается пользователю. Событие о том, что документ готов передаваться на мобильное устройство. Вызывается при запросе с ТСД получения документа для работы. |
| **DocumentAppointed** | Документ назначен пользователю. Событие о том, что документ захвачен на обработку. Вызывается в момент, когда документ был передан на ТСД для работы с ним. |
| **DocumentChanged** | Документ изменен. Событие об изменении документа. Вызывается при сохранении документа на сервер в процессе работы на ТСД при использовании в конфигурации Mobile SMARTS действия «Сохранение документа на сервер». |
| **DocumentFinished** | Документ завершен. Событие о завершении обработки документа. Вызывается при получении сервером завершенного документа с ТСД. |
| **DocumentReleased** | Документ возвращен с ТСД без обработки. Вызывается когда документ с терминала был возвращен пользователем вызовом release (вернуть документ без изменений). |
| **DocumentRemoved** | Документ удален. Событие об удалении документа с сервера. Удаление выполняется внешней системой. |
| **ListDocuments** | Получить список документов. Вызывается при входе в список документов на ТСД. Используется в паре с событием ПолучитьДокумент для того, чтобы реализовать выбор документов прямо из учетной системы, без предварительной выгрузки. |
| **GetDocument** | Получить документ. Вызывается при выборе в списке документов на ТСД того документа, который еще не был выгружен (попал в этот список с помощью события ПолучитьСписокДокументов). |
| **События номенклатуры** | |
| **ProductNotFound** | Получить товар. Событие о том, что продукт был запрошен терминалом по коду (штрихкод, артикул, код) с сервера и не был найден в серверном справочнике. |
| **ProductsNotFound** | Получить товары по списку Id. Событие о том, что несколько товаров были запрошены терминалом по идентификаторам с сервера и не были найден в серверном справочнике. Позволяет реализовывать получение сразу нескольких товаров из базы учетной системы без предварительной выгрузки. |
| **GetProductByPartOfName** | Получить товар по части наименования. Вызывается при вводе в строку поиска в списке номенклатуры на ТСД. Используется для того, чтобы реализовать поиск номенклатуры по части наименования прямо из учетной системы. |
| **GetCellById** | Получить ячейку по идентификатору |
| **GetCellByBarcode** | Получить ячейку по штрихкоду |
| **ListProducts** | Получить список товаров. Вызывается при входе в список номенклатуры на ТСД. Используется для того, чтобы реализовать выбор номенклатуры прямо из учетной системы, без предварительной выгрузки. |
| **События таблиц** | |
| **TableRequest** | Обработать запрос. Событие вызывается при запросе данных из таблицы, определенной в конфигурации Mobile SMARTS. Позволяет получать строки таблицы он-лайн по запросу с ТСД. |

# **Разработка собственного коннектора к внешней системе**

Для работы коннектора под управлением [сервера Mobile SMARTS](https://www.cleverence.ru/support/category:601/) и настройки параметров коннектора через [панель управления](https://www.cleverence.ru/support/category:67/) создаются два файла dll: первая dll, предназначенная для сервера, размещается в <Папка базы Mobile SMARTS>\ Server\DataService\ bin\. Вторая dll, для панели управления — в <Папка базы Mobile SMARTS>\ Control panel\ Addins. Имена файлов должны иметь вид Cleverence.Connectivity.\*.dll (например, Cleverence.Connectivity.MyConnector.dll — для сервера и Cleverence.Connectivity.MyConnector.Panel.dll — для панели управления).

В «Решении» (Solution) в Visual Studio нужно создать два проекта с типом выходных данных «Библиотека классов»:

[](https://www.cleverence.ru/upload/medialibrary/fb9/fb9e1d2644a2cb19837c8b9335190c83.png)

Скачать заготовку [коннектора](https://www.cleverence.ru/files/22194/MobileSMARTS_TestConnector.zip) (Решение Visual Studio 2019 с двумя проектами).

Пространство имен класса коннектора должно начинаться на Cleverence.Connectivity, целевая рабочая среда .NET Framework 4.6.1.

Для серверной версии коннектора в «Ссылки» (Reference Assemblies) нужно добавить следующие dll:

Cleverence.Barcoding.dll

Cleverence.Common.dll

Cleverence.Connectivity.dll

Cleverence.DataCollection.dll

Cleverence.MobileSMARTS.dll

Данные библиотеки находятся по пути <папка установки Mobile SMARTS>\ Server\DataService\ Bin (по умолчанию, C:\ Program Files (x86)\ Cleverence Soft\ Mobile SMARTS\ Server\ DataService\ Bin).

В проект коннектора для [панели управления](https://www.cleverence.ru/support/category:67/) нужно добавить в «Ссылки» (Reference Assemblies):

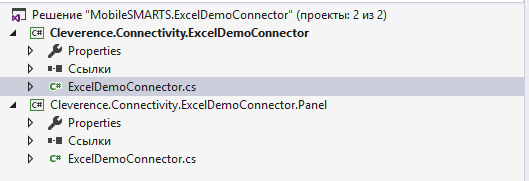
Cleverence.DataCollection.dll

Cleverence.MobileSMARTS.ComConnector.dll

Библиотека**Cleverence.MobileSMARTS.ComConnector.dll** находится по пути <папка установки Mobile SMARTS>\ Control Panel (по умолчанию C:\ Program Files (x86)\ Cleverence Soft\ Mobile SMARTS\ Control Panel).

Рекомендуется скопировать все нужные dll в отдельную папку рядом с файлом «Решения» (Solution) и добавить ссылки на библиотеки из этой папки.

В каждый из проектов (для сервера и для панели управления) нужно добавить по файлу \*.cs для исходного кода класса коннектора.

[](https://www.cleverence.ru/upload/medialibrary/8c5/8c555706b6abac88d4d818137a8bdfd3.png)

Класс коннектора должен реализовывать интерфейс IConnector (находится в пространстве имен Cleverence.Connectivity). Можно наследовать класс своего коннектора от класса ConnectorTypical (также из Cleverence.Connectivity), который реализует IConnector и, кроме того, содержит свойства для указания имен обработчиков событий сервера и виртуальные функции для вызова обработчиков.

Обе версии класса коннектора (для cервера и для [панели управления](https://www.cleverence.ru/support/category:67/)) должны иметь одинаковое имя и находится в одинаковых пространствах имен (namespace), начинающихся на Cleverence.Connectivity (например, Cleverence.Connectivity.MyConnector).

При наследовании от ConnectorTypical в серверной версии коннектора следует перегрузить функции: Initialize (выполняет инициализацию коннектора), Deinitialize (выполняет деинициализацию), InvokeMethod (выполняет вызов во внешнюю систему). Именно InvokeMethod реализует основной функционал коннектора по работе с внешней системой. Также следует перегрузить свойство Initialized (возвращает признак, инициализирован ли коннектор, true — инициализирован, false — нет). При необходимости, могут быть перегружены функции обработки событий сервера (например, GetProduct — получение товара при запросе с ТСД, DocumentFinished — на сервер с ТСД попал завершенный документ и др.). Если не перегружать функции обработки событий, при возникновении событий будет вызываться функция InvokeMethod, в которую передается имя обработчика события, указанное в настройках и соответствующие событию аргументы (см. [События сервера](https://www.cleverence.ru/support/category:601/)).

Версия коннектора для [панели управления](https://www.cleverence.ru/support/category:67/) также наследуется от ConnectorTypical. Перегружать какие-либо функции в этом случае не нужно. Для подключения коннектора к базе внешней системы обычно требуются определенные настройки (адрес базы внешней системы, имя пользователя/ пароль и т.п). Для того, чтобы иметь возможность редактировать нужные настройки и чтобы настройки сохранялись, требуется добавить свойства в классы коннектора для панели управления и сервера. Заготовка коннектора:

[С#]

Серверная часть:

namespace Cleverence.Connectivity.DemoConnector

{

public class DemoConnector : ConnectorTypical

{

public DemoConnector()

{

}

public string MyProperty

{

get;

set;

}

public override bool Initialized

{

get

{

throw new NotImplementedException();

}

}

public override void Initialize()

{

throw new NotImplementedException();

}

public override void Deinitialize()

{

throw new NotImplementedException();

}

public override object InvokeMethod(string methodName, object[] args)

{

throw new NotImplementedException();

}

}

}

Часть для панели управления:

namespace Cleverence.Connectivity.DemoConnector

{

public class DemoConnector : ConnectorTypical

{

public DemoConnector()

{

}

public string MyProperty

{

get;

set;

}

}

}

В серверной части нужно реализовать Initialized, Initialize, Deinitialize, InvokeMethod. Для примера добавлено свойство MyProperty.

Собранную dll серверной версии размещаем в <Папка базы Mobile SMARTS>\Server\DataService\bin\, версию для панели управления в <Папка базы Mobile SMARTS>\Control panel\Addins.

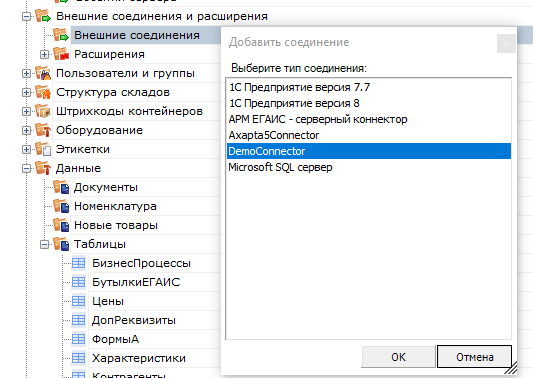
Сервер Mobile SMARTS в целях безопасности выполняет проверку цифровой подписи загружаемых сборок. Если после размещения неподписанного файла dll в <Папка базы Mobile SMARTS>\Server\DataService\bin\> перезапустить службу сервера Mobile SMARTS, в логе сервера базы данных dataserver\_\*.log (в C:\ProgramData\Cleverence\Logs) появится сообщение:

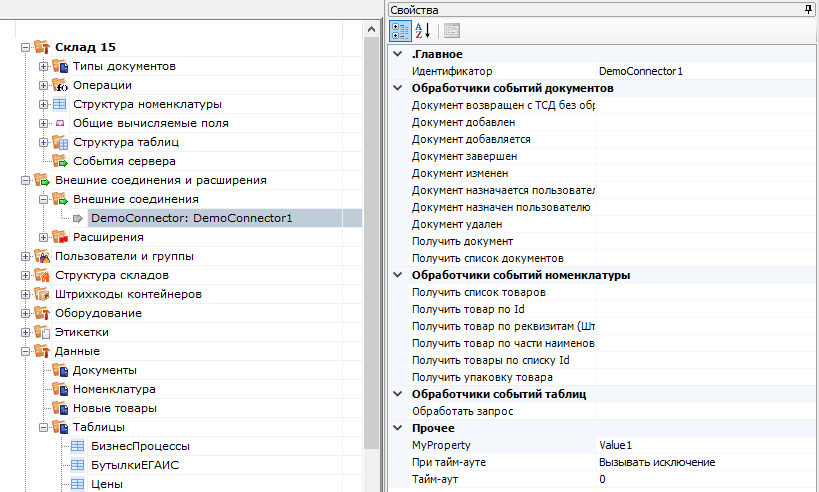
2019-08-05 10:08:08.6537|ERROR|NLogger.WriteNlogEvent| Коннекторы не загружены! Обнаружена неподписанная сборка: C:\ProgramData\Cleverence\Базы Mobile SMARTS\e3945857-308f-4829-92e2-720dc11d1bec\Server\DataService\bin\Cleverence.Connectivity.DemoConnector.dll

Видим, что dll коннектора не загрузилась. В некоторых случаях загрузка неподписанной сборки приводит к остановке сервера. Для того, чтобы можно было проверить работу коннектора и выполнить отладку, сервер нужно запустить с ключом Debug из командной строки (службу сервера перед этим нужно остановить):

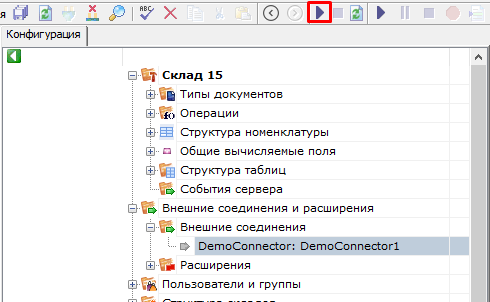
C:\Program Files (x86)\Cleverence Soft\Mobile SMARTS\Server\Cleverence.MobileSMARTS.Server.exe /debug

После этого в [панели управления](https://www.cleverence.ru/support/category:67/) можно добавить коннектор в конфигурацию:

[](https://www.cleverence.ru/upload/medialibrary/699/699f51a1f30eb521781217e4f3cb9419.png)

[](https://www.cleverence.ru/upload/medialibrary/621/621484758442d0cccb1ed8d1fec0cfbe.png)

Настройка параметров коннектора выполняется через панель свойств. Когда настройка выполнена, сохраним конфигурацию. После этого можно запустить коннектор с помощью кнопки «Пуск»:

[](https://www.cleverence.ru/upload/medialibrary/1d8/1d8bd2b9cfd400b46294c53d631c5d19.png)

Если требуется отладка кода коннектора, рядом с файлом коннектора в <Папка базы Mobile SMARTS>\Server\DataService\bin\ следует разместить файл \*.pdb (см. [Тестирование и выпуск разработанного коннектора](https://www.cleverence.ru/support/66149/)). Когда отладка закончена, обратитесь в [техническую поддержку](https://www.cleverence.ru/sd) «Клеверенс» для подписания dll.

Скачать заготовку [коннектора](https://www.dropbox.com/s/tlifyoihr2179du/MobileSMARTS.DemoConnector.zip?dl=0) (Решение Visual Studio 2019 с двумя проектами)

# **Пример разработки коннектора к внешней системе**

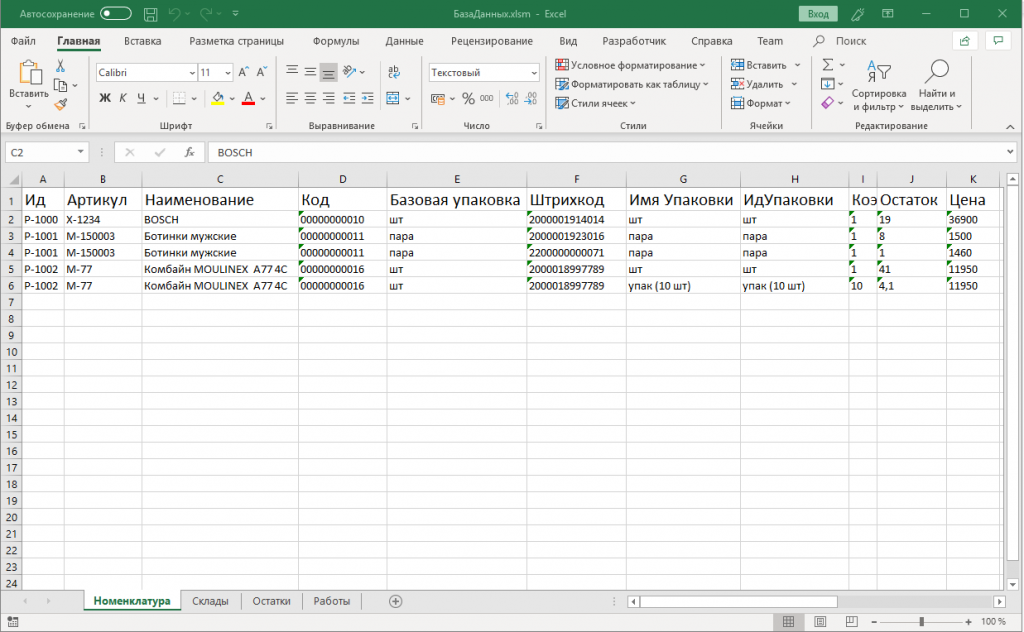
Для примера разработаем коннектор, который будет выполнять получение данных из файла Excel онлайн по запросам с ТСД. В примере будет рассмотрена, в том числе, перегрузка функций-обработчиков событий сервера для получения номенклатуры, документов и обработки запросов к таблицам.

## Постановка задачи

Требуется разработать коннектор, который будет по запросам с ТСД получать данные из файлов Excel. Используются следующие справочники: «Номенклатура», «Склады», «Остатки». Должна быть обеспечена работа с документами «Поступление», на ТСД должен отображаться список документов для работы, подготовленных на сервере. Пользователь ТСД выбирает документ, принимает товар, после завершения работы данные записываются в исходный документ. Используется база Mobile SMARTS «[Магазин 15](https://www.cleverence.ru/RTL15/)».

## Подготовка данных

Справочники будут храниться в одном файле (книге) Excel на листах: «Номенклатура», «Склады», «Остатки». Файл со справочниками имеет фиксированное название «БазаДанных.xlsm» (книга Excel с поддержкой макросов) и располагается в заданной папке (путь к папке указывается в настройках коннектора). Файлы документов «Поступления» будут находиться в этой же папке и называться «Поступление №00001.xlsx», «Поступление №00002.xlsx» и т. д. В архиве с исходниками данные находятся в папке Xlsx, состав полей см. в файлах.

[](https://www.cleverence.ru/upload/medialibrary/704/704a5adc997de052d79f8b4ca9c7c587.png)

### **Разработка и тестирование коннектора**

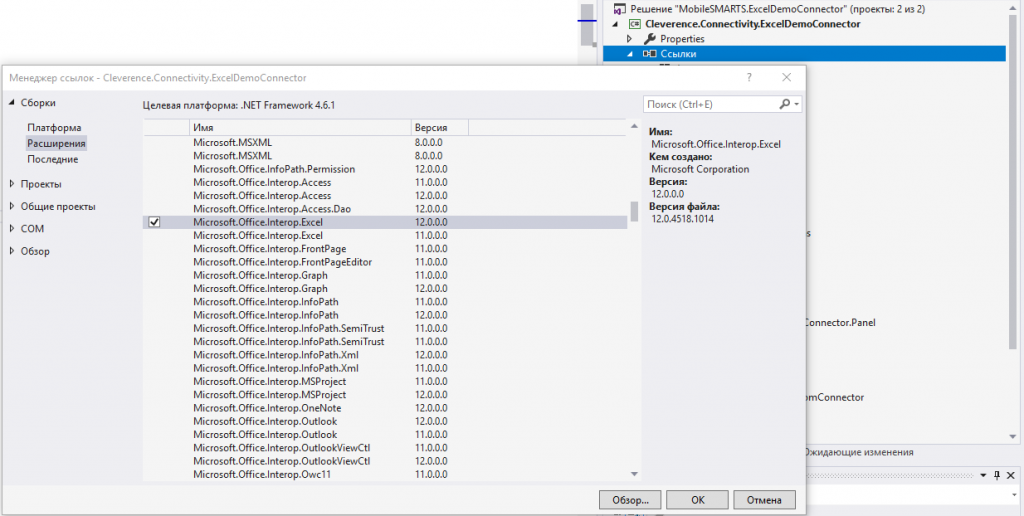
В Visual Studio создаем «Решение» (solution) и два проекта в нем:

Cleverence.Connectivity.ExcelDemoConnector (для сервера) и  
Cleverence.Connectivity.ExcelDemoConnector.Panel (для панели управления).

Добавляем в ссылки (Reference assemblies) в серверный проект Cleverence.Connectivity.ExcelDemoConnector: Cleverence.Common.dll, Cleverence.Connectivity.dll, Cleverence.DataCollection.dll, Cleverence.MobileSMARTS.dll.

В проект для панели управления Cleverence.Connectivity.ExcelDemoConnector.Panel: Cleverence.DataCollection.dll, Cleverence.MobileSMARTS.ComConnector.dll.

Для работы с файлом Excel будем использовать COM-объект Excel.Application (на ПК должен быть установлен Microsoft Excel), добавим в ссылки в проект для сервера Microsoft.Office.Interop.Excel:

[](https://www.cleverence.ru/upload/medialibrary/545/5459b254d0a9646d42b0b5ed6944558f.png)

Добавим в каждый из проектов по файлу ExcelDemoConnector.cs, в котором будет находится класс коннектора ExcelDemoConnector. Базовым классом для ExcelDemoConnector является ConnectorTypical.

Добавим свойство, с помощью которого можно задать путь к папке с файлами Excel, используемыми для обмена данными:

[С#]

public string DatabaseFolder

{

get;

set;

}

Свойство должно быть как в серверном варианте коннектора, так и в варианте для Панели управления. Других настроек для нашего коннектора не требуется, займемся разработкой серверной части, которая будет выполнять обмен данными с Excel.

Добавим приватное поле Excel.Application excel для хранения ссылки на COM-объект Excel. Теперь первым делом нам нужно реализовать, функцию Initialize, которая переводит коннектор в рабочее состояние:

[С#]

public override void Initialize()

{

this.Dispose();

if (!this.Enabled)

throw new InvalidOperationException(this.GetType().Name + " is not enabled.");

this.excel = new Excel.Application();

this.excel.Visible = false;

}

Вызываем Dispose, чтобы выполнить деиницилиазацию, если ранее коннектор был инициализирован. Если вызовы коннектора запрещены через gанель управления (Enabled == false), вызываем исключение. Если вызовы разрешены, создаем COM-объект.

Деинициализация:

[С#]

public override void Deinitialize()

{

if (this.excel != null)

{

this.excel.Quit();

this.excel = null;

}

}

Признак того, что коннектор инициализирован:

public override bool Initialized

{

get

{

return this.excel != null;

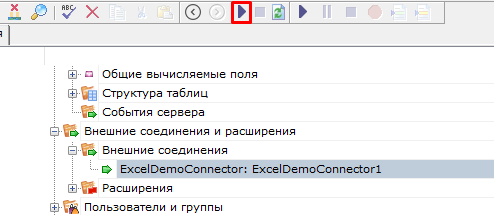
}

}

Этого достаточно, чтобы проверить запуск коннектора через панель управления.

Соберем решение в Visual Studio. Развернем базу «[Магазин 15 Расширенный](https://www.cleverence.ru/software/licenses/RTL15B-1C83/#spec)» или «[Мегамаркет](https://www.cleverence.ru/software/licenses/RTL15C-1C83/)». Скопируем dll для сервера в <Папка базы Mobile SMARTS>\Server\DataService\bin\, для панели управления в <Папка базы Mobile SMARTS>\Control panel\Addins. Запустим сервер в режиме отладки из командной строки: C:\Program Files (x86)\Cleverence Soft\Mobile SMARTS\Server\Cleverence.MobileSMARTS.Server.exe/debug.

Добавим коннектор в конфигурацию, сохраним и проверим, что выполняется запуск.

[](https://www.cleverence.ru/upload/medialibrary/e9a/e9add86799c88b63bcaf365d9de7da75.png)

Реализуем получение номенклатуры онлайн из файла Excel. Когда товар запрашивается терминалом на сервере по каким-либо реквизитам (Ид, Штрихкод, Артикул) и товар не найден в выгруженном справочнике номенклатуры, выполняется вызов обработчика события «[Получить товар](https://www.cleverence.ru/support/2768/)»

В случае наследования от ConnectorTypical обработчик события может быть реализован как во внешней системе (вызов через InvokeMethod функции из внешней системы по имени), так непосредственно в коннекторе. Для реализации обработчика в самом коннекторе нужно перегрузить функцию GetProduct:

[С#]

[EventProcessor(EventType = EventType.ProductNotFound)]

public override object GetProduct(Cleverence.Warehouse.DeviceInfo di, string productId, string packingId,

SearchProductMode searchMode, Cleverence.Barcoding.BarcodeData barcodeData)

{

CheckInit();

return GetProductInternal(productId, packingId, searchMode);

}

Не забываем указать атрибут EventProcessor (EventType = EventType.ProductNotFound). Реализацию поиска товаров по заданным реквизитам в зависимости от режима (searchMode) см. в GetProductInternal.

Событие «Получить список товаров» возникает, когда пользователь на ТСД заходит в список номенклатуры (выбор товара по 0 в действии «Выбор номенклатуры») и в базе нет выгруженного справочника. Перегрузим в коннекторе функцию GetProductsList:

[С#]

[EventProcessor(EventType = EventType.ListProducts)]

public override object GetProductsList(DeviceInfo di, string searchStr)

{

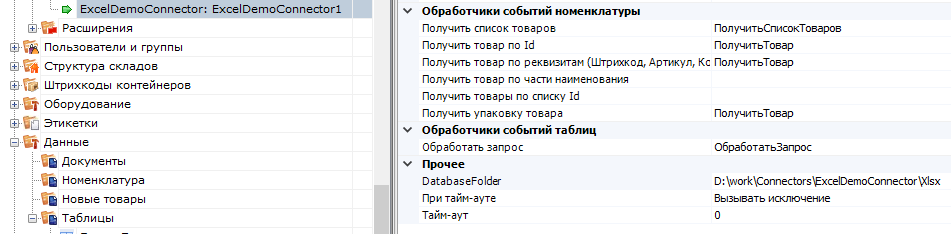
CheckInit();

return GetProductsListInternal(searchStr);

}

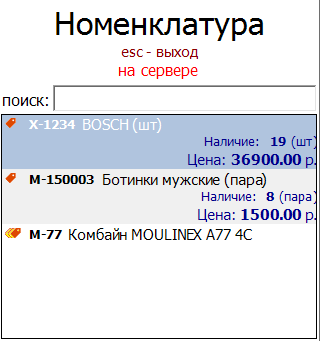
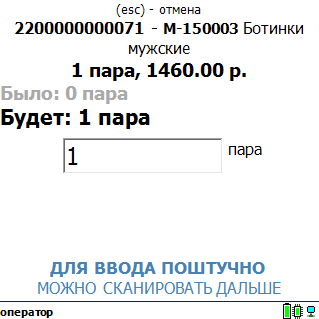
Функция возвращает PackedProductCollection со всем списком номенклатуры.

Проверим, как происходит получение номенклатуры. В настройках коннектора нужно указать путь к папке, в которой находится файл Excel со справочниками (DatabaseFolder). Нужно заполнить обработчики событий номенклатуры: «Получить список товаров», «Получить товар по Id», «Получить товар по реквизитам» (штрихкод, артикул, код), «Получить упаковку товара».

[](https://www.cleverence.ru/upload/medialibrary/b9d/b9d294adb74be98d5cb24a6f299e7e54.png)

Задание имен обработчиков требуется для того, чтобы указать, что коннектор имеет подписку на определенные события. При вызове обработчика через InvokeMethod указанные имена обработчиков передаются в InvokeMethod.

Проверить получение номенклатуры можно с помощью клиента Mobile SMARTS для ПК. Для получения списка товаров заходим в «Просмотр справочников -> Товары». Проверить получение по штрихкоду можно, например, в операции «Сбор штрихкодов»:

[](https://www.cleverence.ru/upload/medialibrary/86f/86f84d2b6476a75982ad262728bf6f35.png) [](https://www.cleverence.ru/upload/medialibrary/157/157bd7f413d3755488e165d94a24a254.png)

Реализуем получение списка документов. Обработчик события «[Получить список документов](https://www.cleverence.ru/support/2767/)» вызывается, когда пользователь на ТСД заходит в список выбора документов по кнопке типа документа.

[С#]

[DocumentEventProcessor(EventType = EventType.ListDocuments)]

public override object GetDocumentsList(DeviceInfo di, string documentTypeName)

{

...

}

Функция возвращает DocumentDescriptionCollection со списком описаний документов (DocumentDescription), готовых для отдачи на ТСД. Список получаем на основе имен файлов xlsx в папке с данными.

Для получения документа в работу на ТСД нужно реализовать обработчик события «[Получить документ](https://www.cleverence.ru/support/2766/)». Обработчик вызывается, когда пользователь на ТСД выбирает документ из списка, полученного с помощью события «Получить список документов», или сканирует штрихкод документа в окне выбора документов.

[С#]

[DocumentEventProcessor(EventType = EventType.GetDocument)]

public override object GetDocument(DeviceInfo di, string documentTypeName, string identity, GetDocumentMode mode)

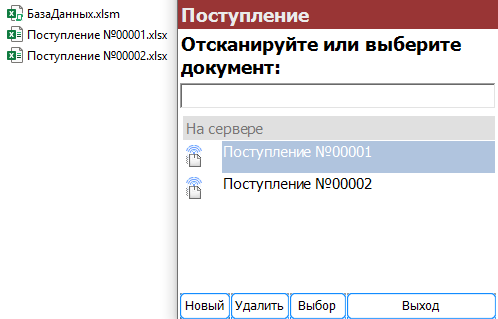
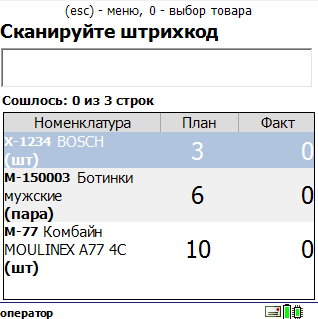
{

...

}

Функция возвращает объект Document, если документ найден по переданным параметрам.

Проверим работу на примере документов «Поступление»:

[](https://www.cleverence.ru/upload/medialibrary/b83/b83ba6fe74fa642e8a4afef69ccbdfc8.png)[](https://www.cleverence.ru/upload/medialibrary/317/317eb9b7e0b00d3f8d5bb7476621a427.png)

Для загрузки завершенного на ТСД документа используем событие «[Документ завершен](https://www.cleverence.ru/support/2760/)».

[С#]

[DocumentEventProcessor(EventType = EventType.DocumentFinished)]

public override object DocumentFinished(DeviceInfo di, string documentTypeName, Document doc)

{

CheckInit();

string fileName = GetFileNameByDocId(doc.Id, documentTypeName);

if (!string.IsNullOrEmpty(fileName))

{

if(LoadDocumentFromTerminal(fileName, doc))

{

MessageCenter.AddNewMessage(string.Format("Документ '{0}' загружен.", doc.Name), null, doc.UserId, false);

BaseRuntimeContext.Current.DocumentsManager.Delete(doc);

return doc;

}

}

return base.DocumentFinished(di, documentTypeName, doc);

}

В данной функции после загрузки документа на ТСД отправляется сообщение с помощью MessageCenter.AddNewMessage. Завершенный документ удаляется из базы Mobile SMARTS с помощью BaseRuntimeContext.Current.DocumentsManager.Delete (doc).

Реализуем получение данных онлайн при запросах к таблицам Mobile SMARTS. Например, в конфигурации «[Магазин 15](https://www.cleverence.ru/RTL15/)» есть таблицы «[Склады](https://www.cleverence.ru/support/37973/?cat=705)», «[Остатки](https://www.cleverence.ru/support/37958/?cat=705)» и др. На ТСД может выполняться получение списка всех складов, запрос остатков определенного товар и др. Если таблица хранится на сервере, в настройках таблицы включен поиск во внешней системе и задан обработчик события «[Обработать запрос](https://www.cleverence.ru/support/32488/)», то сервер вызывает указанный обработчик. Перегрузим в нашем коннекторе функцию ProcessTableRequest.

[С#]

[EventProcessor(EventType = EventType.TableRequest)]

public override object ProcessTableRequest(DeviceInfo di, DocumentQuery dq, DocumentTableInfo tableInfo)

{

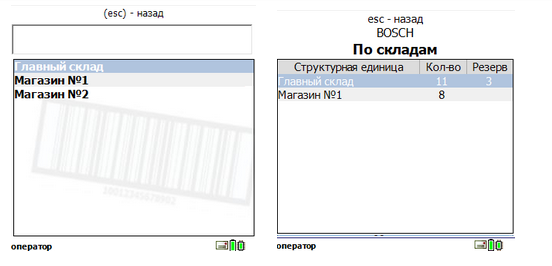
...

}

В функцию передается: DocumentQuery dq — объект запроса, в свойстве WhereRootElement содержится корень синтаксического дерева запроса. Обойдя дерево, можно сформировать запрос в том виде, который ожидает внешняя система. DocumentTableInfo tableInfo — описание таблицы, в свойстве Name содержится имя таблицы из конфигурации Mobile SMARTS. В простейшем случае можно возвращать все строки запрошенной таблицы, не накладывая условие из DocumentQuery, сервер Mobile SMARTS сам выполнит выборку данных по заданным условиям. Однако, в случае реальной работы с большими объемами данных, так делать не рекомендуется.

Функция возвращает коллекцию строк RowCollection.

Проверим получение складов и остатков:

[](https://www.cleverence.ru/upload/medialibrary/8aa/8aa390136fce8f802527468a12c24a99.png)

Кроме обработки событий сервера, есть возможность вызова произвольных функций внешней системы, которые будут возвращать некоторые данные на ТСД или выполнять какую-то работу во внешней системе. В конфигурации Mobile SMARTS для этого используется действие «[Вызов внешней системы](https://www.cleverence.ru/support/2539/)». В коннекторе должна быть реализована функция InvokeMethod.

В нашем случае InvokeMethod будет вызывать макрос Excel с указанным именем, передавая полученные с ТСД аргументы:

[С#]

public override object InvokeMethod(string methodName, object[] args)

{

CheckInit();

string dbPath = Path.Combine(DatabaseFolder, DbFileName);

Excel.Workbook workbook = this.excel.Workbooks.Open(dbPath);

try

{

return RunMacro(methodName, args);

}

finally

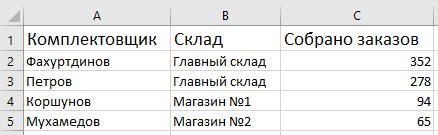
{

workbook.Close();

}

}

Добавим в книгу Excel лист «Работы», на котором будет таблица с количеством собранных сотрудниками заказов за неделю:

[](https://www.cleverence.ru/upload/medialibrary/64e/64ea0b5f3f934345a6075ef298471a60.png)

На VB в Excel напишем функцию, которая будет возвращать таблицу с колонками «Комплектовщик», «Собрано заказов» по переданному складу.

[С#]

Function СобраноЗаказов(Склад As String)

Dim sh\_src As Worksheet

Dim storageConnector As Object, rowCollection As Object, row As Object

Dim picker As String

Dim ordCnt As Integer

Dim rng As Excel.Range

Set storageConnector = CreateObject("Cleverence.Warehouse.StorageConnector")

Set rowCollection = CreateObject("Cleverence.Warehouse.RowCollection")

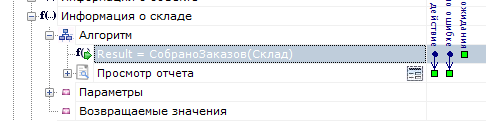
Set sh\_src = Worksheets("Работы")

…..

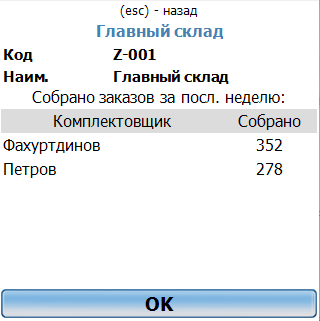
СобраноЗаказов = storageConnector.ToXml(rowCollection)

End Function

Функция возвращает объект Cleverence.Warehouse.RowCollection, сериализованный в xml. В конфигурации Mobile SMARTS сделаем вызов данной функции при просмотре информации о складе:

[](https://www.cleverence.ru/upload/medialibrary/d79/d796386517bcf8da7b48ff511a2f5c76.png)

В действии «Просмотр отчета» выведем полученную таблицу:

[](https://www.cleverence.ru/upload/medialibrary/b07/b077807031abbbe025df0b203c656ea2.png)

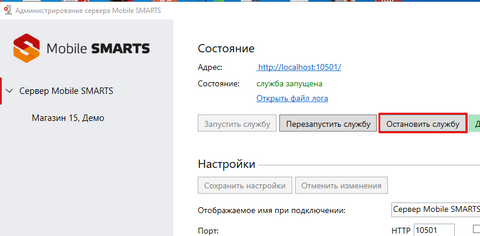
# **Тестирование и выпуск разработанного коннектора к внешней системе**

В связи с повышением требований к безопасности сервера все разработанные коннекторы должны пройти процедуру проверки и подписания у «Клеверенс».

Попытка использования неподписанного коннектора в «продуктивном» режиме приводят к полной остановке сервера с записью в лог файле «Коннекторы не загружены! Обнаружена неподписанная сборка: хххх.dll», либо «Коннекторы не загружены! Ошибка проверки подписи dll: xxxx.dll».

Для отладки коннектора при разработке следует использовать запуск сервера в тестовом режиме.

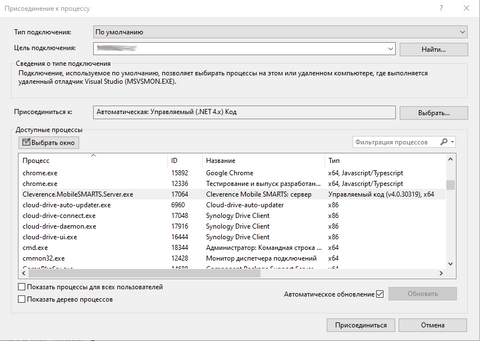
Для этого остановите сервис сервера:

[](https://www.cleverence.ru/upload/medialibrary/356/wjgeihp7yv34mv3qprderp0ujy2kvsjv/content_img.png)

И запускайте его из командной строки с параметром /debug:

c:\Program Files (x86)\Cleverence Soft\Mobile SMARTS\Server\Cleverence.MobileSMARTS.Server.exe /debug

Далее Вы можете просто подключаться отладчиком Visual Studio к запущенному процессу и вести отладку вашего коннектора.

[](https://www.cleverence.ru/upload/medialibrary/68b/vsa3ru4v60kg02g5ds27rrlth3rva3yb/content_img.png)

После завершения разработки вашего коннектора создайте запрос в техническую поддержку «Клеверенс» о его проверке и подписании.