

Приложения и базы

Приложение **Mobile SMARTS** (см. [App \[ПриложениеСМАРТС\]](#)) представляет собой некоторое прикладное решение на платформе **Mobile SMARTS** ("Магазин 15", "Учет имущества", и т.д.).

Каждое приложение устанавливается из отдельного дистрибутива (если при этом платформа **Mobile SMARTS** на компьютере не установлена, сначала будет предложено загрузить и установить платформу). Например, с [этой страницы](#) можно скачать продукт **Mobile SMARTS** для ЕГАИС, предназначенный для учета алкогольной продукции при помощи терминалов сбора данных.

После установки приложения появляется возможность создавать базы данных на основе шаблона установленного приложения. ([Подробнее](#)).


База данных **Mobile SMARTS** (см. [AppInstance \[БазаСМАРТС\]](#)) - это экземпляр определенного приложения. База данных хранит конфигурацию **Mobile SMARTS** и текущие данные (справочники, документы, настройки). База данных находится в определенной папке на диске компьютера. В зависимости от того, какой способ обмена данными с терминалами требуется, база данных может работать в следующих режимах: обмен с сервером **Mobile SMARTS**, прямое подключение терминала к компьютеру через провод, обмен через папку на диске (Режим RDP - общая папка, терминал подключается физически к другому компьютеру, обмен непосредственно с терминалом выполняет специальная утилита, установленная на компьютере, к которому подключается терминал).

См. [подробнее](#).

После установки **Mobile SMARTS** пользователь имеет возможность работать с базами данных (добавлять базы в список зарегистрированных баз, удалять, изменять настройки) с помощью [Менеджера баз](#). Обмен данными возможен только с базами, зарегистрированными в списке баз на данном компьютере.

Удаленная база. Существует возможность добавить в список (зарегистрировать) не только базу, находящуюся на данном компьютере, но и работающую на сервере **Mobile SMARTS**, расположенном на другом компьютере в сети. В менеджере баз такая база будет находиться в разделе Удаленное подключение:

Удаленное подключение

 База на SERVER-MSK-01

На локальном компьютере удаленная база также, как и локальная, будет находиться в заданной при добавлении базы папке, но в этой папке будут содержаться только настройки соединения с удаленным сервером **Mobile SMARTS** и пользовательские настройки все данные (справочники, документы) при обмене будут загружаться (и выгружаться) с удаленного сервера. Сервер **Mobile SMARTS** представляет собой web-сервис, при обмене данными компонента вызывает функции web-сервиса, данные передаются по протоколу http. Организовывать общий доступ к папке базы на удаленном компьютере не нужно.

Объектам, служащим описанием удаленной базы, является также [AppInstance \[БазаСМАРТС\]](#) и для доступа к данным использует [StorageConnector \[Соединение\]](#). При использовании функций обмена нет никакой разницы, удаленная база или локальная.

- 1 Создание и настройка новой базы данных
- 2 Подключение к базе данных из учетной системы
- 3 Получение списка баз, зарегистрированных на компьютере
- 4 Получение и сохранение произвольных настроек
- 5 Получение режима работы базы

Создание и настройка новой базы данных

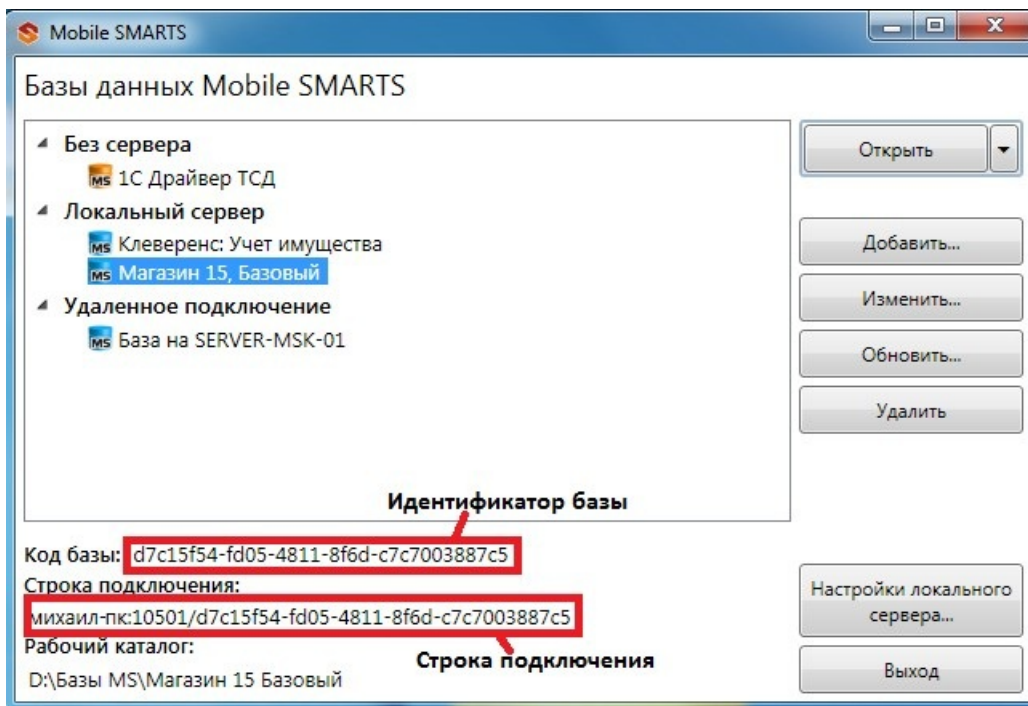
См. [Заведение базы данных с помощью менеджера баз Mobile SMARTS](#)

Для созданной базы данных пользователь указывает нужный режим работы (Прямое подключение к устройству, Прямая работа с папкой (RDP режим), Подключение к серверу). См. [Настройка базы данных Mobile SMARTS](#).

Подключение к базе данных из учетной системы

Для того, чтобы была возможность производить обмен данными с базой **Mobile SMARTS**, необходимо сначала выполнить подключение к базе. Нам потребуется создать экземпляр COM-объекта [Cleverence.Warehouse.StorageConnector](#) и вызвать функцию [SelectCurrentApp](#)

[\[УстановитьПодключениеСБазойСМАРТС\]](#), передав в качестве аргумента Идентификатор (Код) базы или Строку подключения к баз



Идентификатор и код в менеджере баз:

C#:

```
var connector = new Cleverence.Warehouse.StorageConnector(); // Создаем экземпляр объекта try
{
    // Выполняем подключение
    connector.SelectCurrentApp("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-c7c7003887c5");
}
catch
{
    // При подключении возникла ошибка, обрабатываем исключение
}
```

«1С:Предприятие 8»:

```
connector = новый СОМОбъект("Cleverence.Warehouse.StorageConnector"); // Создаем экземпляр объекта Попытка
connector.УстановитьПодключениеСБазойСМАРТС("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-c7c7003887c5");
//Выполняем подключение Исключение
// При подключении возникла ошибка, обрабатываем исключение КонецПопытки;
```

Получение списка баз, зарегистрированных на компьютере

Получить список баз может быть нужно, например, для того, чтобы дать пользователю учетной системы возможность выбирать базу, с которой будет выполняться обмен, из списка баз и хранить это как настройку в учетной системе, вместо того, чтобы жестко прописывать в коде строку подключения.

Воспользуемся для этого функцией [GetAppsList \[ПолучитьСписокБаз\]](#). Перед вызовом этой функции выполнять подключение к какой-либо базе не нужно. Функция возвращает как локальные, так и удаленные базы, зарегистрированные на данном компьютер

C#:

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

try

{

var appInstanceList = connector.GetAppsList(""); //Получаем список баз - объект AppInstanceCollection for(int i=0;
i<appInstanceList.Count; i++) // Обходим в цикле полученный список

{

var appInstance = appInstanceList[i]; //Получаем элемент списка - объект AppInstance var id = appInstance.Id; //Идентификатор
базы

var connectionString = appInstance.ConnectionString; //Строка подключения var name = appInstance.Name; //Имя базы

// Обрабатываем полученные значения

}

}

catch

{

// Обработка исключения

}

}
```

«1С:Предприятие 8»:

```
connector = новый СОМОбъект("Cleverence.Warehouse.StorageConnector"); // Создаем экземпляр объекта Попытка

СписокБаз = connector.ПолучитьСписокБаз(""); //Получаем список баз - объект AppInstanceCollection Для Инд=0 По
СписокБаз.Количество - 1 Цикл // Обходим в цикле полученный список

БазаСмартса = СписокБаз.Item(Инд); //Получаем элемент списка - объект AppInstance Ид = БазаСмартса.Ид;
//Идентификатор базы

СтрокаПодключения = БазаСмартса.СтрокаПодключения; //Строка подключения Имя = БазаСмартса.Имя; //Имя базы

... // Обрабатываем полученные значения КонецЦикла;

Исключение

// Обработка исключения КонецПопытки;
```

Получение и сохранение произвольных настроек

Есть возможность сохранять произвольные настройки в базе данных Mobile SMARTS из учетной системы. Например, это могут бы какие-либо настройки, используемые при выгрузке номенклатуры, параметры отборов документов и др. Настройки сохраняются в виде "Ключ-Значение", Ключ - строка, задающее имя настройки, Значение - данные простого типа (строка, число, булево). Для работы с настройками необходимо получить объект базы **AppInstance [БазаСМАРТС]** и использовать функции

GetSettings [ПолучитьНастройки] и **SaveSettings** [СохранитьНастройки]. Объектом настроек является

AppInstanceSettings [НастройкиБазыСМАРТС].

C#:

```
var connector = new Cleverence.Warehouse.StorageConnector(); // Создаем экземпляр объекта try
{
    // Получаем базу по строке подключения или идентификатору
    var appInstance = connector.GetAppById("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-c7c7003887c5");
    var appInstanceSettings = appInstance.GetSettings(); //Получаем объект настроек AppInstanceSettings var settingNode =
appInstanceSettings.GetSetting("ИмяНастройки"); //Получаем узел настройки по имени
    var value = settingNode.Value; //Значение настройки settingNode.Value = newValue; // Присваиваем новое значение настройке
    appInstance.SaveSettings(appInstanceSettings); // Сохраняем настройки в базу
}
catch
{
    // Обработка исключения
}
```

«1С:Предприятие 8»:

```
connector = новый СОМОбъект("Cleverence.Warehouse.StorageConnector"); // Создаем экземпляр объекта Попытка
БазаСМАРТС = connector.ПолучитьБазуСМАРТСПоИД("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-c7c7003887c5");
//Получаем базу по строке подключения или идентификатору
НастройкиБазы = БазаСМАРТС.ПолучитьНастройки(); //Получаем объект настроек AppInstanceSettings УзелНастройки =
НастройкиБазы.Настройка("ИмяНастройки"); //Получаем узел настройки по имени
ЗначениеНастройки = УзелНастройки.Значение; //Значение настройки УзелНастройки.Значение = НовоеЗначение; //
Присваиваем новое значение настройке
БазаСМАРТС.СохранитьНастройки(НастройкиБазы); // Сохраняем настройки в базу Исключение
// Обработка исключения КонецПопытки;
```

Получение режима работы базы

Рассмотрим, как получить, в каком режиме работает база данных (Прямое подключение к устройству, Прямая работа с папкой (RD режим) или Подключение к серверу, см. **Настройка базы данных Mobile SMARTS**). Нам нужно будет получить объект базы **AppInstance** [БазаСМАРТС], получить настройки базы с помощью функции **GetSettings** [ПолучитьНастройки] и использовать следующие свойства объекта настроек **AppInstanceSettings** [НастройкиБазыСМАРТС]:

Свойство	Тип	Описание
ServerMode [РаботаССервером]	Булево	Режим работы с сервером. Истина - база работает в режиме с сервером, Ложь - без сервер

LocalServerMode [РаботаСЛокальнымСервером]	Булево	<p>Возвращает признак, работает ли серверная база на локальном компьютере.</p> <p>Истина - серверная база на локальном компьютере, Ложь - удаленная серверная база или база не серверная (прямой обмен с ТСД или с папкой).</p>
RemoteServerMode [РаботаСУдаленнымСервером]	Булево	<p>Возвращает признак того, что база работает на удаленном сервере.</p> <p>Истина - серверная база на удаленном компьютере, Ложь - локальная серверная база или база не серверная (прямой обмен с ТСД или с папкой).</p>
FolderMode [РаботаСКаталогом]	Булево	<p>Возвращает признак того, что база работает в режиме "с каталогом".</p> <p>Истина - база работает в режиме "с каталогом".</p> <p>Режим RDP - обмен данными через общую папку, терминал подключается физически к другому компьютеру, обмен непосредственно с терминалом выполняет специальная утилита установленная на компьютере, к которому подключается терминал.</p>
DeviceMode [РаботаСУстройствомНапрямую]	Булево	<p>Возвращает признак того, что база работает в режиме напрямую с терминалом (батч-режим)</p> <p>Истина - база работает в режиме напрямую с терминалом (батч-режим).</p> <p>Обмен с терминалом происходит через проводное подключение.</p>

C#:

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

try

{

// Получаем базу по строке подключения или идентификатору

var appInstance = connector.GetAppById("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-c7c7003887c5");

    var appInstanceSettings = appInstance.GetSettings(); //Получаем объект настроек AppInstanceSettings
if(appInstanceSettings.ServerMode == true)

{

//Работа с сервером if(appInstanceSettings.LocalServerMode == true)

{

// Локальный сервер

}

else

{

// Удаленный сервер

}

}

else if(appInstanceSettings.FolderMode == true)

{

// Работа с папкой

}

else if(appInstanceSettings.DeviceMode == true)

{

// Работа с ТСД напрямую

}

}

catch

{

// Обработка исключения

}
```

```
connector = новый СОМОбъект("Cleverence.Warehouse.StorageConnector"); // Создаем экземпляр объекта

//соединения Попытка

БазаСМАРТС = connector.ПолучитьБазуСМАРТСПоИД("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-c7c7003887c5");

//Получаем базу по строке подключения или идентификатору

НастройкиБазы = БазаСМАРТС.ПолучитьНастройки(); //Получаем объект настроек ApplInstanceSettings Если
НастройкиБазы.РаботаССервером = Истина Тогда //Работа с сервером

Если НастройкиБазы.РаботаСЛокальнымСервером = Истина Тогда // Локальный сервер

...

Иначе // Удаленный сервер

...

КонецЕсли;

ИначеЕсли НастройкиБазы.РаботаСКаталогом = Истина Тогда //Работа с папкой

...

ИначеЕсли НастройкиБазы.РаботаСУстройствомНапрямую = Истина Тогда //Работа с ТСД напрямую

...

КонецЕсли;

Исключение

// Обработка исключения КонецПопытки;
```

Была ли статья полезна?

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