

**ООО «Компания Семь Печатей»**

---

117216, Москва, ул. Феодосийская, д. 1, корп. 6; тел.(факс): (495)225-25-31, (495)020-23-46

Email: [2252531@mail.ru](mailto:2252531@mail.ru); Web-page: [www.sevenseals.ru](http://www.sevenseals.ru), [www.shop-sevenseals.ru](http://www.shop-sevenseals.ru)



**Система  
контроля и управления доступом**

***TSS-OFFICE***

***TSS-PROFI***

***ВЕРСИЯ 7***

**Программное обеспечение**

**Интеграция с системой  
заказа пропусков и регистрации посетителей**

*руководство администратора*

**Москва**

**2017**

## Оглавление

<b>1. Принципы работы.....</b>	<b>2</b>
<b>2. Установка и настройка.....</b>	<b>2</b>
<b>3. Алгоритм работы.....</b>	<b>4</b>
<b>4. Приложение 1. Схема взаимодействия СЗП и СКУД TSSProfi....</b>	<b>5</b>
<b>5. Приложение 2. Методы ActiveX библиотеки InterfaceToTSSGuestExchange</b>	<b>6</b>
5.1. Работа с интерфейсной БД.....	6
5.2. Получить список точек доступа.....	6
5.3. Получить уровни доступа .....	6
5.4. Создать посетителя.....	7
5.5. Удаление/блокировка посетителя/карты .....	7
5.6. Получить информацию о проходах (список событий) .....	8

Настоящий документ описывает систему интеграции *СКУД TSS2000 Profi* (SDK) с произвольной системой заказа пропусков и регистрации посетителей (далее СЗП).

Суть интеграции - в автоматизации доступа посетителей на объект посредством СКУД. При регистрации посетителя информация о нем передается в *СКУД TSS2000 Profi* (далее TSS). В результате этого, посетитель перемещается по объекту согласно правам доступа, заданным при его оформлении. Информация о его передвижениях сохраняется в *Системном журнале* СКУД, что позволяет формировать детальные отчеты о пребывании на объекте.

## 1. Принципы работы

Для реализации интеграции со *СКУД TSS* для разработки модуля интеграции предоставляется ActiveX библиотека методов *InterfaceToTSSGuestExchange.dll*. Со стороны TSS на сервере СКУД должна функционировать программа *TSSGuestExchangeServer.exe*, реализованная, как служба Windows.

Схема взаимодействия двух программных комплексов приведена в Приложении 1.

Список методов библиотеки взаимодействия – в Приложении 2.

Интерфейсная БД *InterfaceDB.FDB* и сервер интеграции *TSSGuestExchangeServer.exe* могут располагаться на любом сетевом ПК, однако наиболее целесообразно разместить их на Сервере СКУД.

После установки и настройки системы взаимодействия будет выполняться следующее:

Передача данных о зарегистрированных посетителях из *СЗП* в СКУД TSS.

Назначение посетителям маршрутов передвижения (разрешенных пунктов прохода).

Активация и деактивация карт в СКУД TSS.

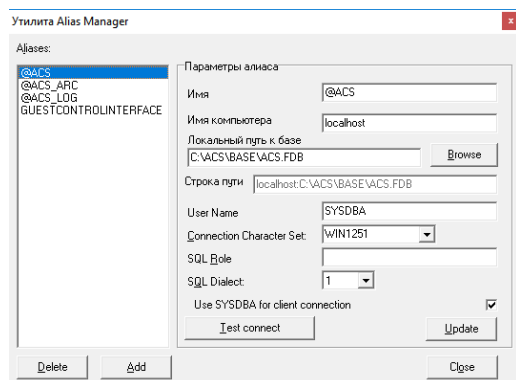
Передача событий о проходах посетителей из СКУД TSS в *СЗП*.

## 2. Установка и настройка<sup>1</sup>

Для работы на стороне Сервера *СЗП* необходимо зарегистрировать библиотеку *InterfaceToTSSGuestExchange.dll* командой *regsvr32*. Для отладки корректности соединений рекомендуется пользоваться тестовым ПО *Project\_test.exe*.

Для запуска системы интеграции на стороне СКУД TSS необходимо:

- Скопировать папку *TSSGuestExchange* с установочного диска в каталог ACS на Сервере СКУД.
- Скопировать файл *INTERFACEDB.FDB* из папки *TSSGuestExchange* в каталог *ACS/Base* на Сервере СКУД.



<sup>1</sup> Предполагается, что СКУД *TSSProfi* и СЗП уже сконфигурированы и работают.

- С помощью утилиты системы TSS *AliasManager.exe* создать алиас, указывающий на эту базу данных. Рекомендованное имя алиаса – GUESTCONTROLINTERFACE.
- Откорректировать настроечный файл *TSSGuestExchangeServer.ini*:
  - aliasbase= алиас базы данных TSS, по умолчанию @acs.
  - aliaslogbase= алиас базы данных системного журнала TSS, по умолчанию @acs\_log.
  - aliasinterface=алиас базы данных заданный в п.3, по умолчанию GUESTCONTROLINTERFACE.
  - appserver= сервер транспорта системы TSS, по умолчанию localhost.
- Проинсталлировать службу *TSSGuestExchangeServer.exe* командой *TSSGuestExchangeServer /install*.
- Для гарантированной работы службы внести ее в список контролируемых служб *Системы управления ACSGMSServer*, добавив в файл *ACSGMSServer.ini* строки (после чего рестартовать службу *ACSGMSServer*):

[SERVO2]

RusName=TSS-СЗП

Name=ServiceTSSGE

NameProcess=TSSGuestExchangeServer

TypeS=S

Path=C:\ACS\TSSGuestExchangeServer.exe

DelayOnStartSystem=5

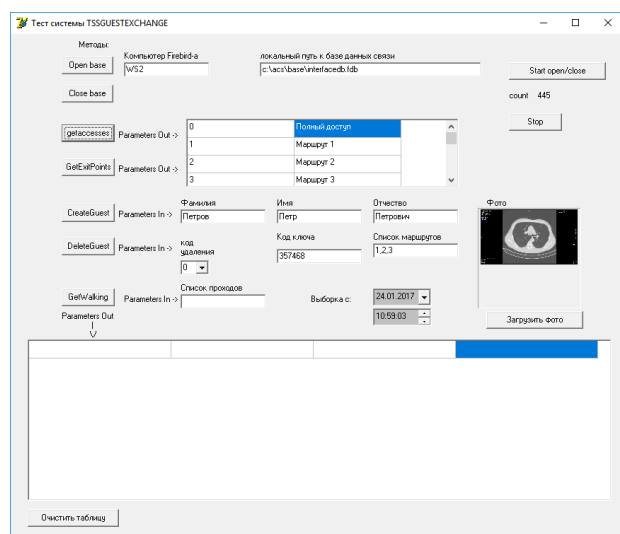
MaxAttemptsStartCounts=5

LifeInterval=-1

Critical=YES

После старта *Сервера интеграции* рекомендуется проверить работоспособность системы обмена данными с помощью тестовой программы *Project\_test.exe*.

Интерфейс программы позволяет выполнить все методы, указанные в [Приложение 2](#).

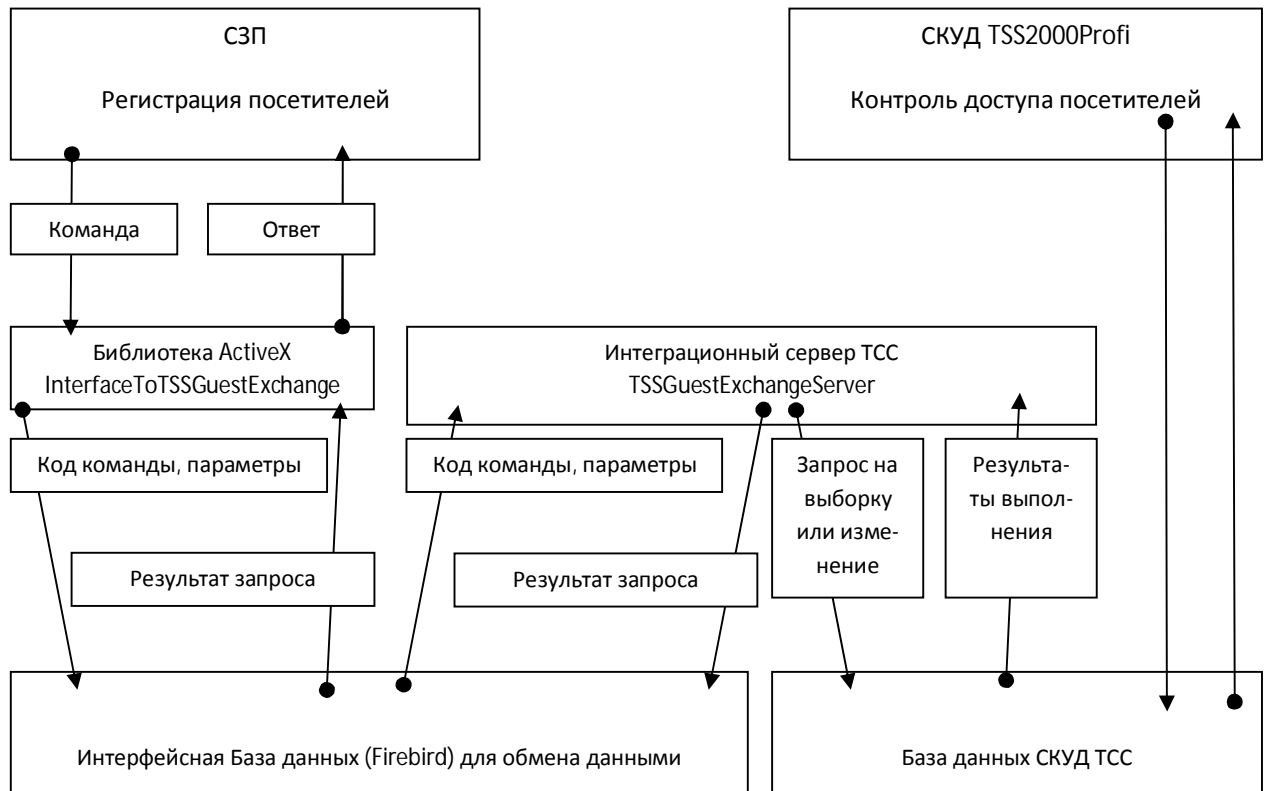


### 3. Алгоритм работы

Система взаимодействия работает по следующему алгоритму:

- При регистрации посетителя в СЗП информация о нем передается в TSS: в БД СКУД создается запись с:
  - Уникальным ID,
  - кодом выданной в VC картой доступа,
  - списком пунктов прохода (маршрутом), через которые посетителю разрешен доступ.
- После регистрации на проходной на вход Посетитель перемещается по объекту СКУД до первого выхода через проходную.
- При выходе его карта аннулируется (удаляется запись в БД СКУД).
- В VC фиксируются события входа и выхода гостя через проходную СКУД.
- В TSS записываются в *Системный журнал* факт добавления в БД гостя («ДОБАВЛЕНИЕ КАРТОЧКИ СТОРОННЕЙ СИСТЕМОЙ»), и факт ее удаления («УДАЛЕНИЕ КАРТОЧКИ СТОРОННЕЙ СИСТЕМОЙ»).
- В TSS записываются в *Системный журнал* все события о проходах гостя, как через проходные, так и во внутренние помещения, включая попытки несанкционированных проходов.

## 4. Приложение 1. Схема взаимодействия СЗП и СКУД TSSProfi



## 5. Приложение 2. Методы ActiveX библиотеки InterfaceToTSSGuestExchange

Каждый метод записывает во входную таблицу интерфейсной базы код операции и параметры и ждет появления результатов в выходной таблице. Чтение и обработку запросов, а также запись результата обработки в выходную таблицу выполняет служба *TSSGuestExchangeServer*. При появлении результата метод возвращает их вызывающей программе и завершает работу.

Примеры работы даны в исходных текстах программы *Project\_test*.

Все методы возвращают:

- 1 – положительный результат.
- 0 – нет результата выполнения.
- -1 – ошибка при выполнении.
- -2 – ошибка по таймауту (например, отсутствие соединения с сервером).

### 5.1. Работа с интерфейсной БД

**OpenBase**(var servername, pathtobase: OleVariant): SYSINT – открыть базу для связи. Все параметры – строковые значения. Servername – имя (или IP адрес) ПК с интерфейсной БД. Pathtobase – полный локальный путь к файлу базы на ПК. При положительном результате – возврат 1.

**CloseBase**: SYSINT – закрыть базу для связи. При положительном результате – возврат 1.

### 5.2. Получить список точек доступа

**getExitPoints**(out listExitPoints: OleVariant): SYSINT - получить список проходов (проходных).

Выходной параметр – массив пар: номер прохода (целое), название прохода (строковое) в системе ТСС. При положительном результате – возврат 1.

№ п.п.	Параметр	Тип	Примечания
1	Номер точки прохода	Целое	-
2	Название точки	Строка	-

### 5.3. Получить уровни доступа

**getAccesses**(out listAccesses: OleVariant): SYSINT – получить список уровней доступа.

Выходной параметр – массив пар: номер маршрута в системе ТСС и его название, если оно задано. При положительном результате – возврат 1.

№ п.п.	Параметр	Тип	Примечания
1	Номер маршрута	Целое	-

2	Название маршрута	Строка	-
---	-------------------	--------	---

#### 5.4. Создать посетителя

**CreateGuest**(var LastName, FirstName, MiddleName, NumCard, Foto, ListUrDost: OleVariant): SYSINT – создать посетителя в базе ТСС.

Параметры: фамилия (строковое), имя (строковое), отчество (строковое), номер карты (строковое, аналог в базе СКУД – табельный номер), фото (целочисленный двумерный массив пикселей), список уровней доступа (массив чисел из списка маршрутов). При положительном результате – возврат 1.

№ п.п.	Параметр	Тип	Примечания
1	Фамилия	Строка	-
2	Имя	Строка	-
3	Отчество	Строка	-
4	№ Карты	Строка	Соответствует полю <i>Код ключа</i> в БД СКУД.
5	Фото посетителя	Массив байт	Двоичный блок.
6	Список уровней доступа	Массив идентификаторов уровней доступа	Задается при выдаче пропуска в СЗП на основе списка уровней доступа (см. метод «Получить уровни доступа»).

#### 5.5. Удаление/блокировка посетителя/карты

**deleteGuest**(var NumCard: OleVariant; CODCOMMAND: SYSINT): SYSINT – удалить посетителя из базы ТСС.

Параметры: номер карты (строковый), код команды (целочисленный: 1-удаление посетителя, 2-удаление карты, 3-блокировка карты). При положительном результате – возврат 1.

№ п.п.	Параметр	Тип	Примечания
1	№ карты	Целое	-
2	Код команды	Целое	1-удаление посетителя, 2-удаление карты, 3-блокировка карты.



## 5.6. Получить информацию о проходах (список событий)

**getWalking**(var listExitPoints: OleVariant; identEvent: SYSINT; dataFiltrEvents: TDateTime; out eventList: OleVariant): SYSINT – получить список проходов.

Параметры: список пунктов прохода (массив целых чисел из списка проходов), identEvent – способ выборки по времени (целое), dataFiltrEvents – время и дата (TDateTime), с которой начинается выборка событий. Выходной параметр – массив событий.

№ п.п.	Параметр	Тип	Примечания
1	Список идентификаторов точек доступа	Массив идентификаторов	События возвращаются по указанным идентификаторам точек доступа или, при задании пустого списка – по всем точкам.
2	Идентификатор события	Целое	Если указано значение 0, то возвращаются события по параметру «Дата фильтра событий».
3	Дата фильтра событий	Дата	Дата, с которой возвращать события. Указывается, если в идентификаторе события задано значение 0.

Каждая строка eventList (событие) представляет собой: дата-время события (TDateTime), тип события (строка), номер прохода (целое), номер карты (строка).

№ п.п.	Параметр	Тип	Описание и способ заполнения
1	Дата и время события	Дата	-
2	Тип события	Строка	-
3	Идентификатор точки доступа	Целое	Идентификатор точки доступа, в которой произошло событие.
4	№ карты	Строка	-

