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; Web-page: www.sevenseals.ru, 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
VisitorControl

руководство администратора

Москва

2018 - 2020

Оглавление

1. Принципы работы 2

2. Установка и настройка 2

3. Алгоритм работы 4

4. Приложение 1. Схема взаимодействия ПК VizitorControl и СКУД TSSProfi 5

5. Приложение 2. Методы ActiveX библиотеки InterfaceToTSSGuestExchange 6

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

6. Приложение 3. Формат интерфейсной базы данных 10

В документе используются специальные термины и выражения. Для полного понимания информации, изложенной в данном тексте, рекомендуем ознакомиться с глоссарием «TSS0011_Словарь терминов».

Настоящий документ описывает систему интеграции *СКУД TSS2000 Profi* с системой заказа пропусков и регистрации посетителей *VisitorControl* компании *Инсайрес*[1].

Программный комплекс *VisitorControl* (далее VC) автоматизирует процедуру заказа пропусков и регистрации посетителей. При регистрации посетителя информация о нем передается в *СКУД TSS2000 Profi* (далееTSS). 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

Работа выполняется посредством модуля *VisitorControl* расширенной интеграции со *СКУД TSS*, ActiveX библиотеки методов *InterfaceToTSSGuestExchange.dll* и программы *TSSGuestExchangeServer.exe*, реализованной, как служба Windows.

Схема взаимодействия двух программных комплексов приведена в Приложении 1.

Список методов библиотеки взаимодействия – в Приложении 2.

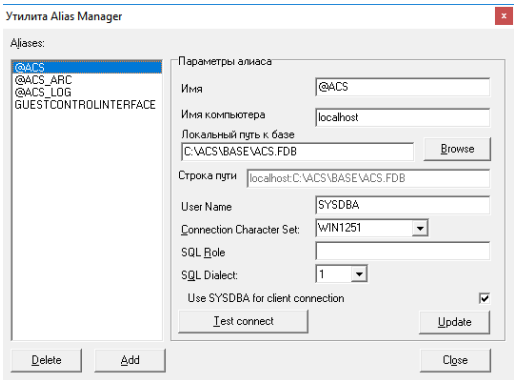
Интерфейсная БД *InterfaceDB.FDB* и сервер интеграции *TSSGuestExchangeServer.exe* могут располагаться на любом сетевом ПК, однако наиболее целесообразно разместить их на Сервере СКУД.

После установки и настройки системы взаимодействия будет выполняться следующее:

- Передача данных о зарегистрированных посетителях из *VisitorControl* 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 *VisitorControl*.

2. Установка и настройка[2]

Для работы на стороне Сервера *VisitorControl* необходимо зарегистрировать библиотеку *InterfaceToTSSGuestExchange.dll* командой *regsvr32*. Для отладки корректности соединений рекомендуется пользоваться тестовым ПО *Project_test.exe*.



[1] Описание установки и настройки системы взаимодействия со стороны *VisitorControl* смотрите в документации компании *Инсайрес*.

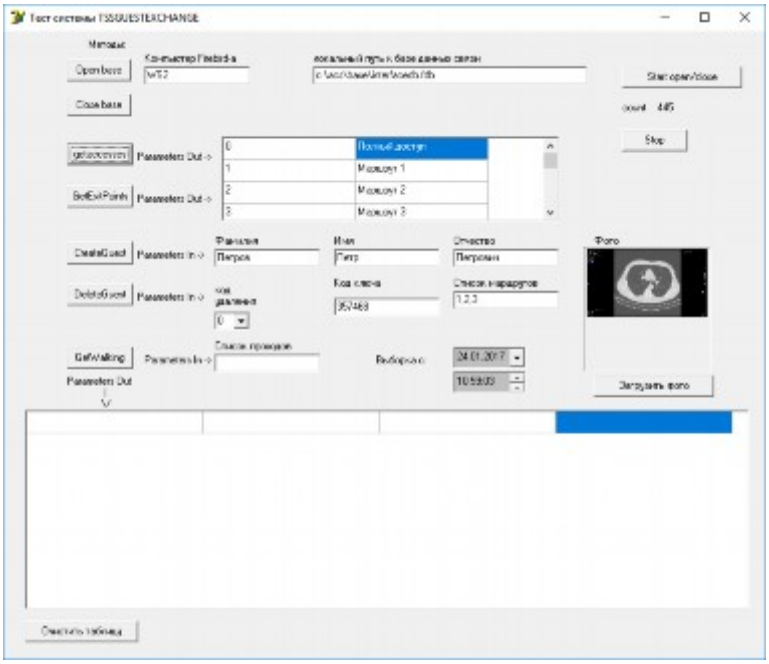
[2] Предполагается, что СКУД *TSSProfi* и система *VisitorControl* уже сконфигурированы и работают.

Для запуска системы интеграции на стороне СКУД ТСС необходимо:

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

```
[SERV02]
RusName=TSS-VisitorControl
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 VC информация о нем передается в TSS: в БД СКУД создается запись с:
  - Уникальным ID,
  - ФИО,
  - кодом выданной в VC картой доступа,
  - списком пунктов прохода (маршрутом), через которые посетителю разрешен доступ.
  - фотоизображением (сканом паспорта)
- Срок действия карты устанавливается либо до первого выхода через проходную, либо до конца текущих суток.
- 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.

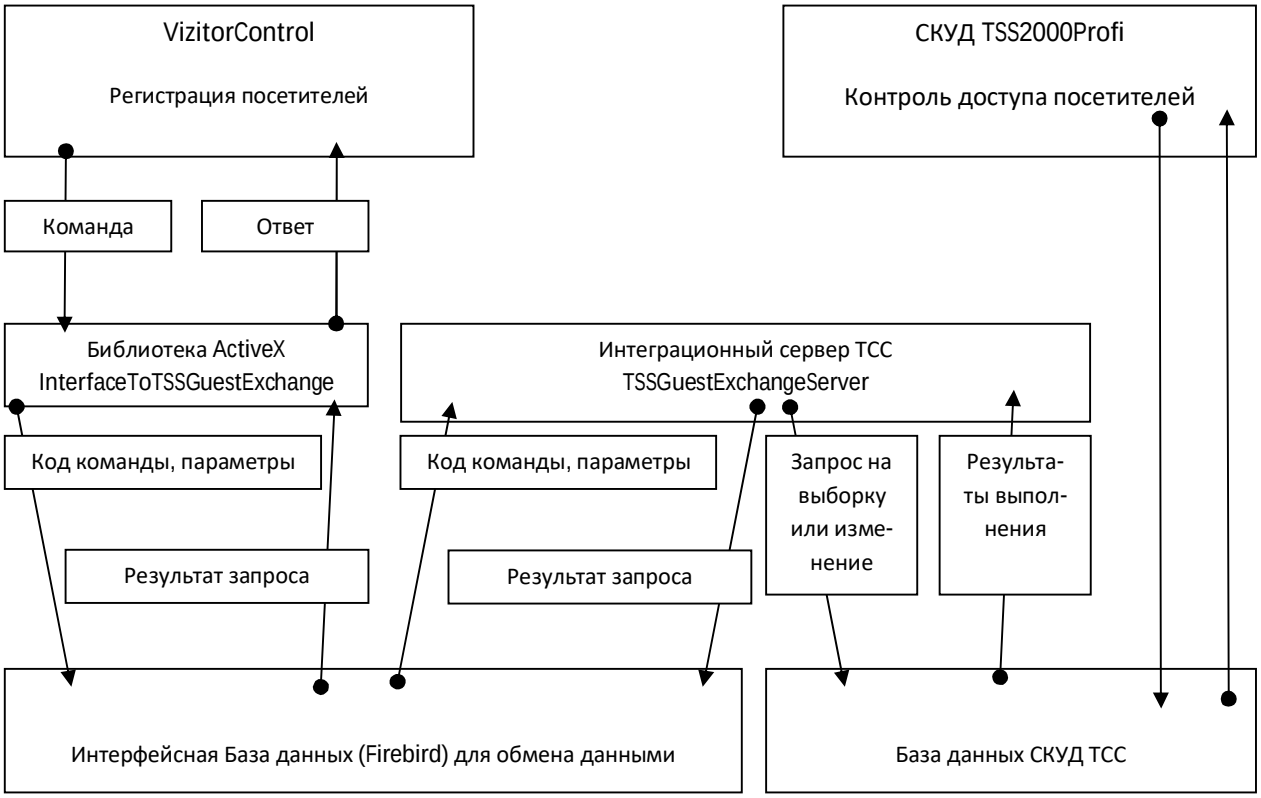
Данные, передаваемые и принимаемые VC, располагаются в буферной базе данных *InterfaceDB*. Входная информация (команды) помещаются в таблицу INPUT_TABLE, выходная (события) в OUTPUT_TABLE.

Формат буферной БД приведен в Приложении 3. Формат интерфейсной базы данных.

Данные гостя временно хранятся в таблице GUESTS, откуда они удаляются автоматически сервером интеграции через 30 суток.

Необходимо периодически выполнять резервное копирование базы данных *Interfacedb.fdb* и ее очистку посредством операции восстановление базы из резервной копии.

4. Приложение 1. Схема взаимодействия ПК VizitorControl 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.

Сами команды со списком параметров (кроме *OpenBase* и *CloseBase*) заносятся в таблицу INPUT_TABLE. Содержательные результаты команд (то есть события по проходам гостей) заносятся в таблицу OUTPUT_TABLE.

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, MidleName, 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.

При создании посетителя информация об этом сохраняется в таблице интерфейсной базы GUESTS, и хранится там 30 дней.

№ п.п.	Параметр	Тип	Примечания
1	Фамилия	Строка	-
2	Имя	Строка	-
3	Отчество	Строка	-
4	№ Карты	Строка	Соответствует полю *Код ключа* в БД СКУД.
5	Фото посетителя	Массив байт	Двоичный блоб.
6	Список уровней доступа	Массив идентификаторов уровней доступа	Задается при выдаче пропуска в *VisitorControl* 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	

6. Приложение 3. Формат интерфейсной базы данных

Интерфейсная база InterfaceDB.fdb состоит из трех таблиц: INPUT_TABLE, OUTPUT_TABLE, GUESTS.

INPUT_TABLE - для команд СЗП – СКУД..

OUTPUT_TABLE - для событий СКУД – СЗП.

Форматы таблиц:

GUESTS

```
ID          INTEGER,
FIRSTNAME   VARCHAR(10),
LASTNAME    VARCHAR(20),
MIDLENAME   VARCHAR(20),
NUMCARD     CHAR(12),
LISTURDOST  VARCHAR(30),
FOTO        BLOB SUB_TYPE 0 SEGMENT SIZE 4096,
DCREATE     TIMESTAMP
```

INPUT_TABLE

```
ID          INTEGER,
COMMAND     VARCHAR(20) NOT NULL,
PARAMETERS  BLOB SUB_TYPE 1 SEGMENT SIZE 80,
PICTURE     BLOB SUB_TYPE 0 SEGMENT SIZE 80
```

OUTPUT_TABLE

```
ID          INTEGER,
RESULT      INTEGER,
RESULT_STR  BLOB SUB_TYPE 1 SEGMENT SIZE 80,
ID_COMMAND  INTEGER
```