OSTCARD
OSTPACK GROUP

117997 Москва, ул. Профсоюзная, 65
Тел.: +7 (495) 792 50 41
Факс.: +7 (495) 792 50 42
Internet: www.ostcard.su
E-mail: info@ostpack.ru / ostcard@ostpack.ru

CDP (Chip Data Preparation)

v.1.4

Руководство пользователя.

Оглавление

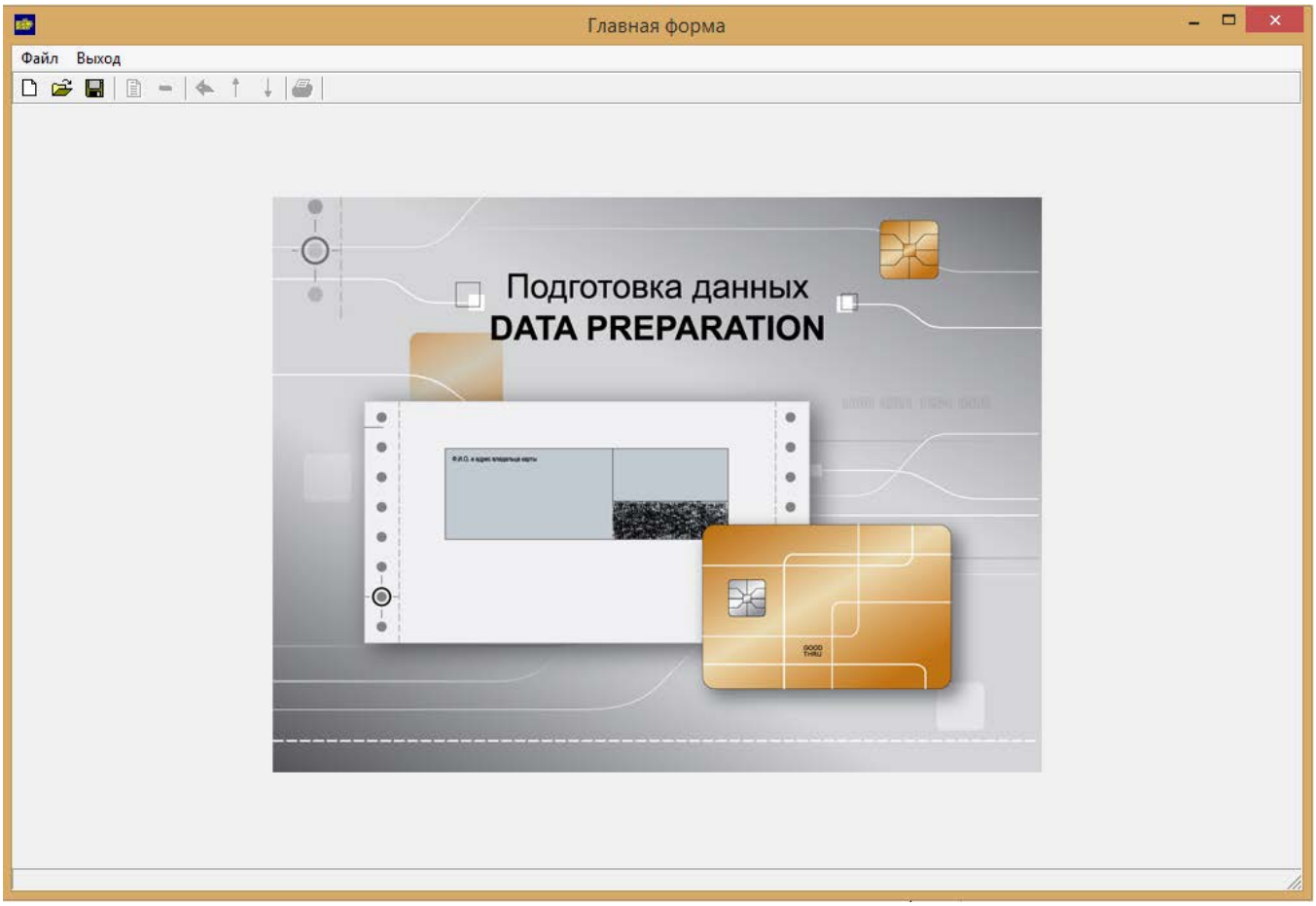
1. Запуск программы
2. Режим управления безопасностью
3. Главное меню
4. Работа с шаблоном
- **4.1. Крипто-модуль**
  - 4.1.1. Типы крипто-модулей
  - 4.1.2. SAM-модуль
  - 4.1.3. Хранилище RSA
  - 4.1.4. Хранилище ключей (файл базы данных)
  - 4.1.5. Хранилище ключей
  - 4.1.6. Импорт/экспорт ключей
- **4.2. Сертификат**
  - 4.2.1. Сформировать запрос
  - 4.2.2. Обработать ответ
- **4.3. Теги**
- **4.4. Дополнительно**
  - 4.4.1. Значения
  - 4.4.2. Длины
- **4.5. Выходной поток – персонализация**
- **4.6. Выходной поток – подготовка ПИНов**
- **4.7. Выходной поток – процессинг**
- **4.8. Выходной поток – база данных**
- **4.9. Устройство печати**
- **4.10. Поля печати**

5. Работа с проектом
- **5.1. Подготовка данных**
  - 5.1.1. Бины
  - 5.1.2. Входная база данных
  - 5.1.3. Выходная база данных
  - 5.1.4. Диверсификационные ключи
  - 5.1.5. Теги+Значения
  - 5.1.6. Составные поля
  - 5.1.7. Запуск задания
- **5.2. Печать данных**
  - 5.2.1. Входная база данных
  - 5.2.2. Поля печати
  - 5.2.3. Запуск задания

6. Словарь терминов
Приложение 1: Формирование входных данных и создание источников ODBC

CDP (Chip Data Preparation) – универсальный программный модуль подготовки данных для приложений на картах с микросхемой (как контактной, так и бесконтактной или дуальной) любой платформы (Native, Open/Global Platform). Обеспечивает подготовку данных для платежных приложений международных платежных систем VISA, MasterCard (VSDC, MChip), Japan Credit Bureau (JCB), национальных платёжных систем China UnionPay (CUP), НСПК/МИР (Российская национальная система платёжных карт), нефинансовых - бонусных, дисконтных, ID и т.д. Модуль предоставляет возможность персонализации нескольких приложений на одной карте, в том числе в случае необходимости использования нескольких источников входных данных. CDP обеспечивает удобство отладки новых шаблонов и тестирования новых карточных приложений.

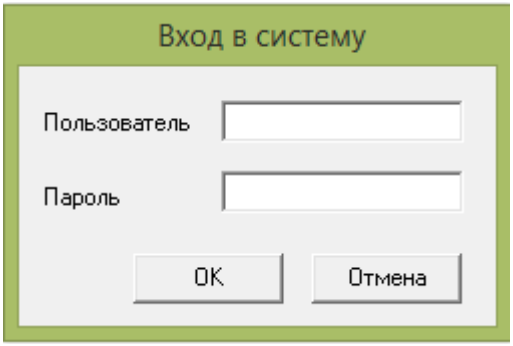
CDP - программный модуль являющийся частью комплексного решения по персонализации пластиковых карт любых типов - **MAP (Multi Application Personalization).**



В случае использования устройств шифрования при подготовке данных предварительно запускается программный модуль Сервер персонализации (HS-сервер). Более подробно о Сервере персонализации можно узнать в документе *«Сервер персонализации HS.exe. Руководство оператора»*.

1. Запуск программы

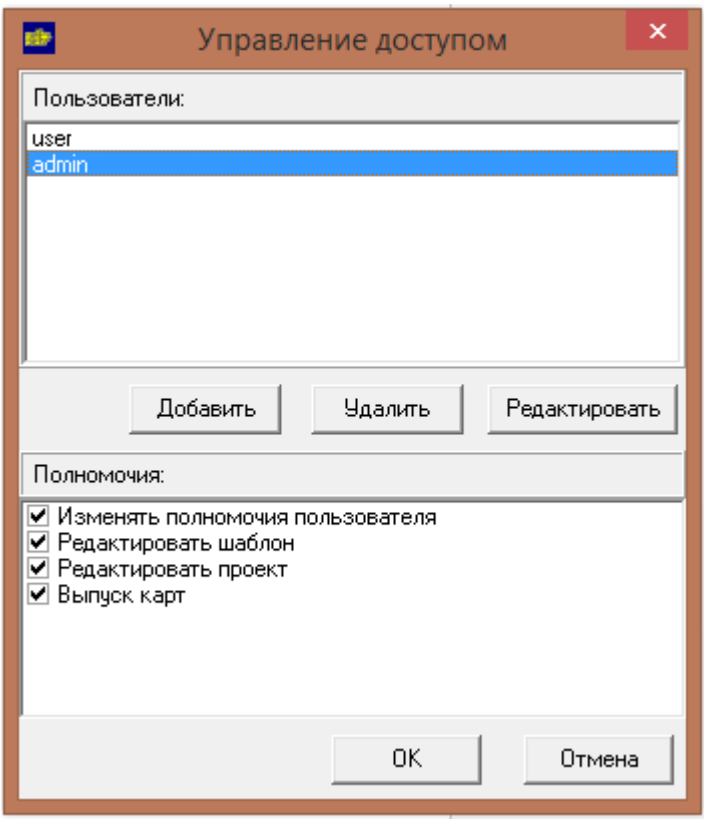
При запуске программы появляется форма, в которой необходимо ввести имя пользователя и пароль для доступа в систему.



Пользователи, пароли и полномочия хранятся в зашифрованном виде в файле **users.kmk**, который должен располагаться в корневой директории программы. Существует встроенный административный пользователь, с полными правами доступа, включая управление пользователями и назначение полномочий: **имя пользователя – admin; пароль – adm**. При первом запуске программы, необходимо войти под этим пользователем. В дальнейшем его можно будет изменить в режиме управления безопасностью.

2. Режим управления безопасностью

Данный режим вызывается из главного меню Файл - Безопасность.

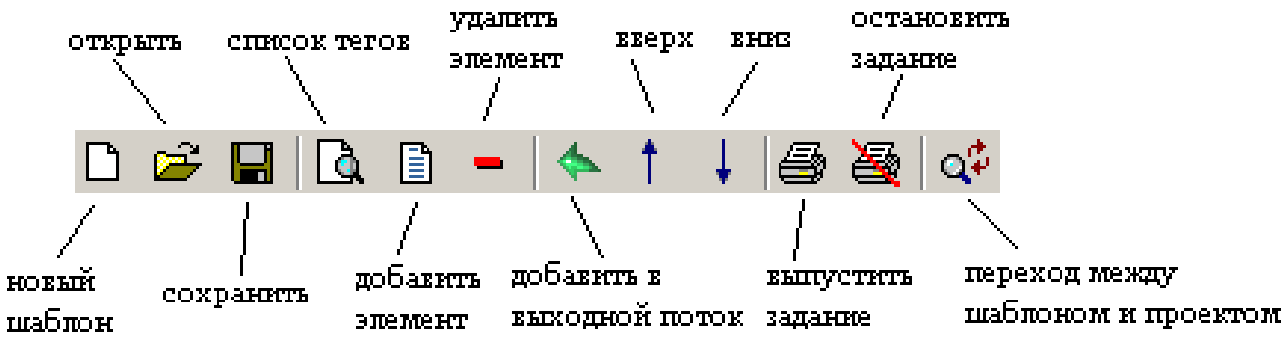


В верхней части экрана показаны пользователи, которые существуют на данный момент в системе. Пользователей можно добавлять, редактировать и удалять, используя соответствующие кнопки. В нижней части показаны полномочия выбранного пользователя.
Существует четыре вида полномочий:
Изменять полномочия пользователя – позволяет заходить в режим безопасности, и изменять права доступа.
Редактировать шаблон* – позволяет редактировать и настраивать шаблон.
Редактировать проект* – позволяет редактировать и настраивать проект.
Выпуск карт – позволяет запускать задания на подготовку и печать данных.

Режим работы с шаблоном и проектом будет описан ниже в соответствующих разделах.

3. Главное меню

После успешного входа в систему под выбранным пользователем открывается главная форма, на которой присутсвует главное меню, используемое для работы с различными модулями[1] системы.



Новый шаблон – создание нового шаблона.
Открыть – открытие файла, существующего (ранее созданного) шаблона или проекта.
Сохранить – сохранение в файл шаблона или проекта.
Сохранить как (доступно в подменю пункта «Файл») – сохранение файла шаблона или проекта под другим именем.
Найти элементы – отображение списка всех тегов, заведённых в шаблон/проект, с именами и входными значениями. Данный пункт меню доступен только в режиме работы с проектом в модуле «Теги» («Подготовка данных» – «бины» – «наименование бина» – «Теги»).
Добавить элемент – добавление нового элемента. Используется для добавления ключей, тегов, значений, длин, полей печати, бинов, составных полей.
Удалить элемент – удаление существующего элемента. Используется для удаления ключей, тегов, значений, длин, элементов выходных потоков, полей печати, бинов, составных полей.
Добавить в выходной поток – добавление элемента в выходной поток:

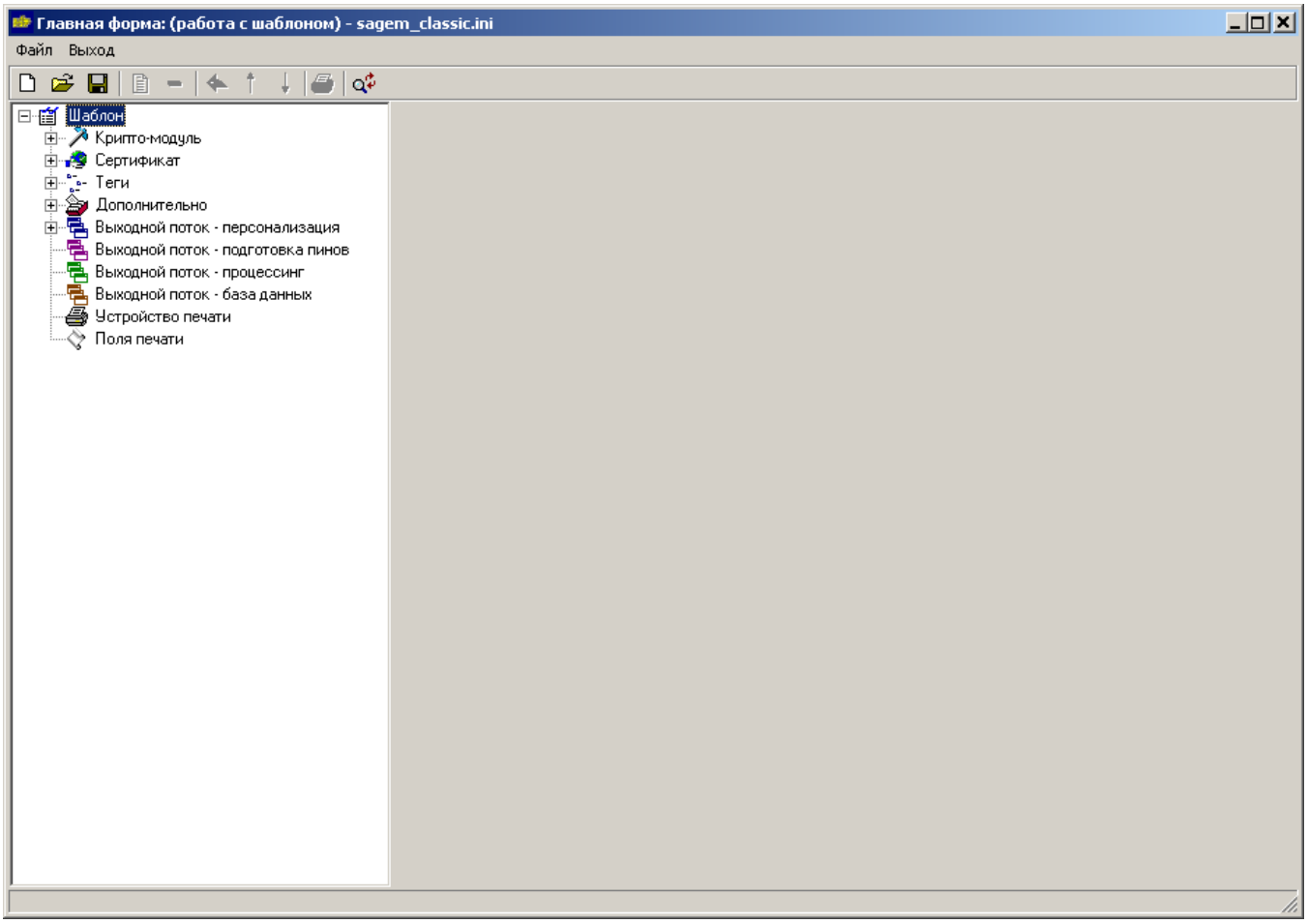
- ***персонализация*** – возможно добавление тегов, значений, длин.
- ***подготовка ПИНов*** – возможно добавление тегов и значений.
- ***процессинг*** – возможно добавление тегов и значений.
- ***база данных*** – возможно добавление тегов и значений.

(Предусмотрено четыре вида выходных потоков, так как могут понадобиться выходные файлы, с различными наборами данных, служащие для разных целей.)
Вверх, вниз – перемещение элемента в основном для удобства отображения. Исключение составляют составные поля и выходные потоки «персонализация», «подготовка ПИНов», а также «Поля печати». В указанных наборах данных важен порядок следования элементов.
Выпустить задание – запуск задания на подготовку или печать данных.
Остановить задание – остановить задание на подготовку или печать данных.
Переход между шаблоном и проектом – быстрый переход между шаблоном и проектом, без открытия диалогового окна.

[1] Некоторые пункты главного меню могут не отображаться или быть недоступными в зависимости от модулей, с которыми работает пользователь системы.

4. Работа с шаблоном

При создании нового шаблона или открытия уже существующего, в левой части экрана в виде дерева будут расположены основные модули, необходимые для настройки шаблона. В правой части показываются свойства данных модулей. Шаблон хранится в виде **ini**-файла, в котором хранятся настройки, не зависящие от входных данных и вида карт.
Обычно для каждого вида карт создается свой шаблон, т.к. наборы тегов, состав значений и выходных потоков могут отличаться в зависимости от вида карт.



Рассмотрим основные модули шаблона:

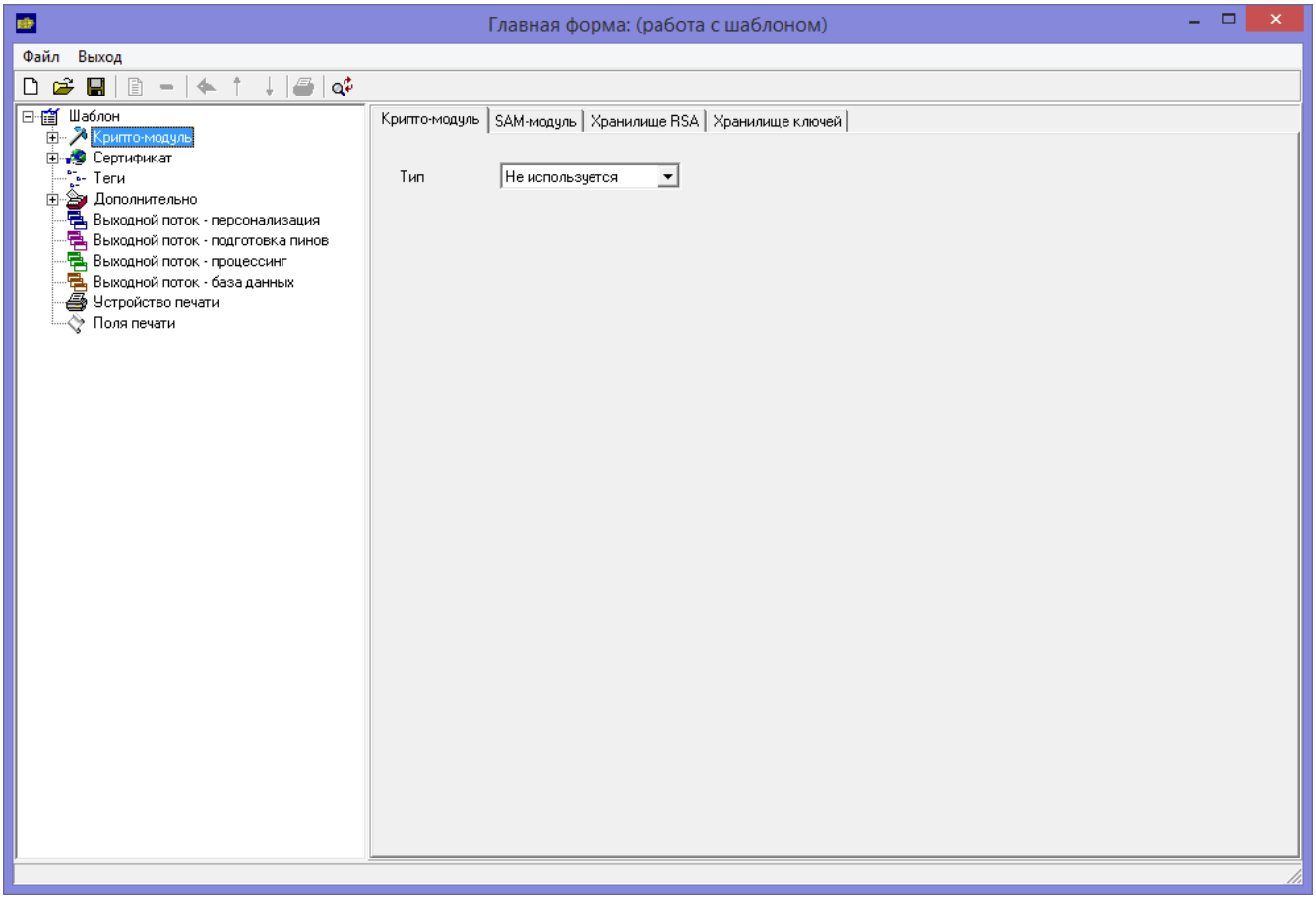
4.1. Крипто-модуль

Если при подготовке данных необходимо использовать шифрование, то требуется выбрать настройки типа крипто-модуля в зависимости от используемого устройства шифрования. В данной версии программы поддерживаются устройства шифрования SafeNet (Eracom) и Thales. Возможен выбор одного из четырёх типов кирптомодулей:
- не используется (установлен по умолчанию);
- Eracom;
- Thales;
- Eracom_EFT.

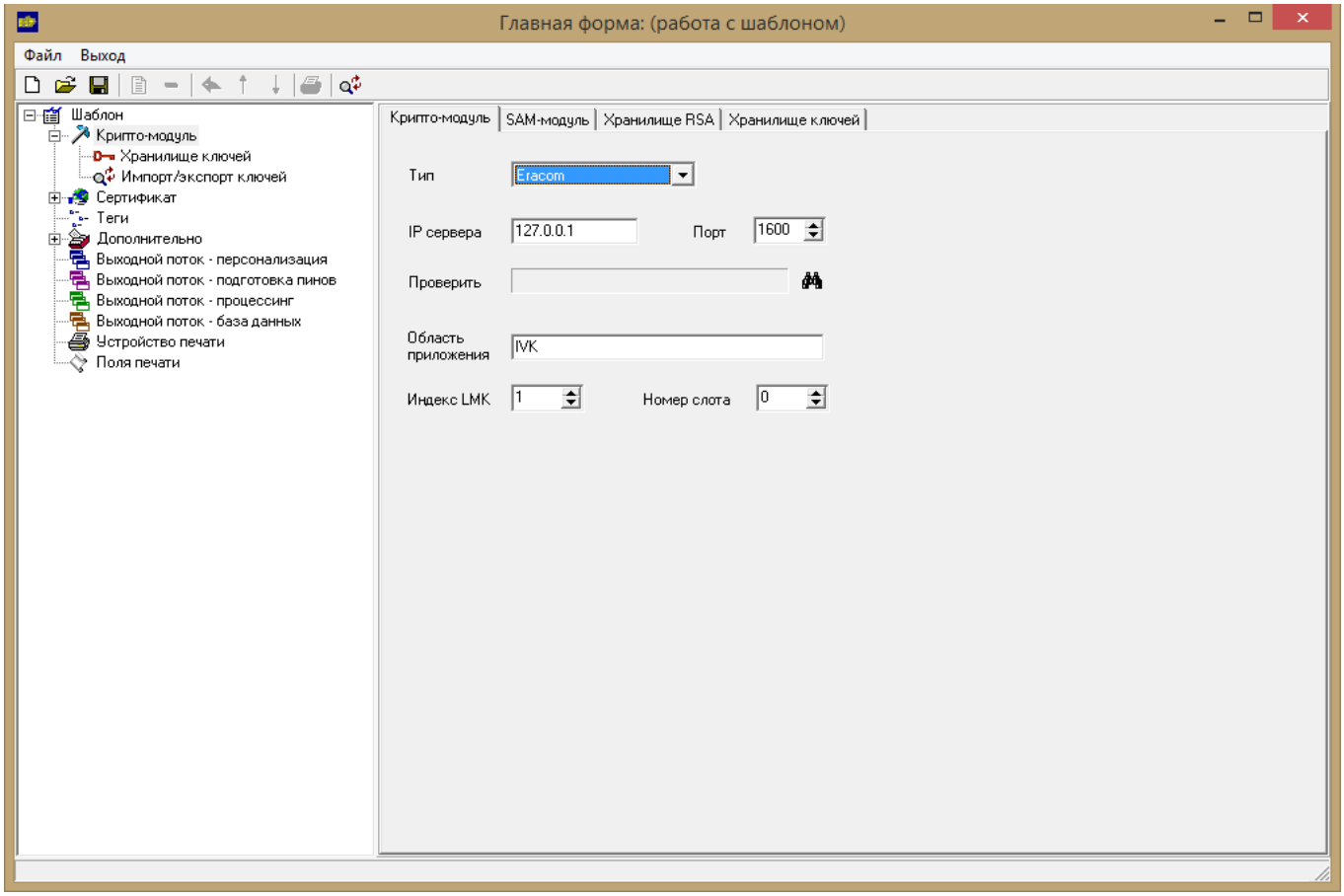
4.1.1. Типы крипто-модулей

4.1.1.1. Крипто-модуль не используется

Данный тип крипто-модуля выбирается в случае отсутствия необходимости шифрования подготавливаемых данных.



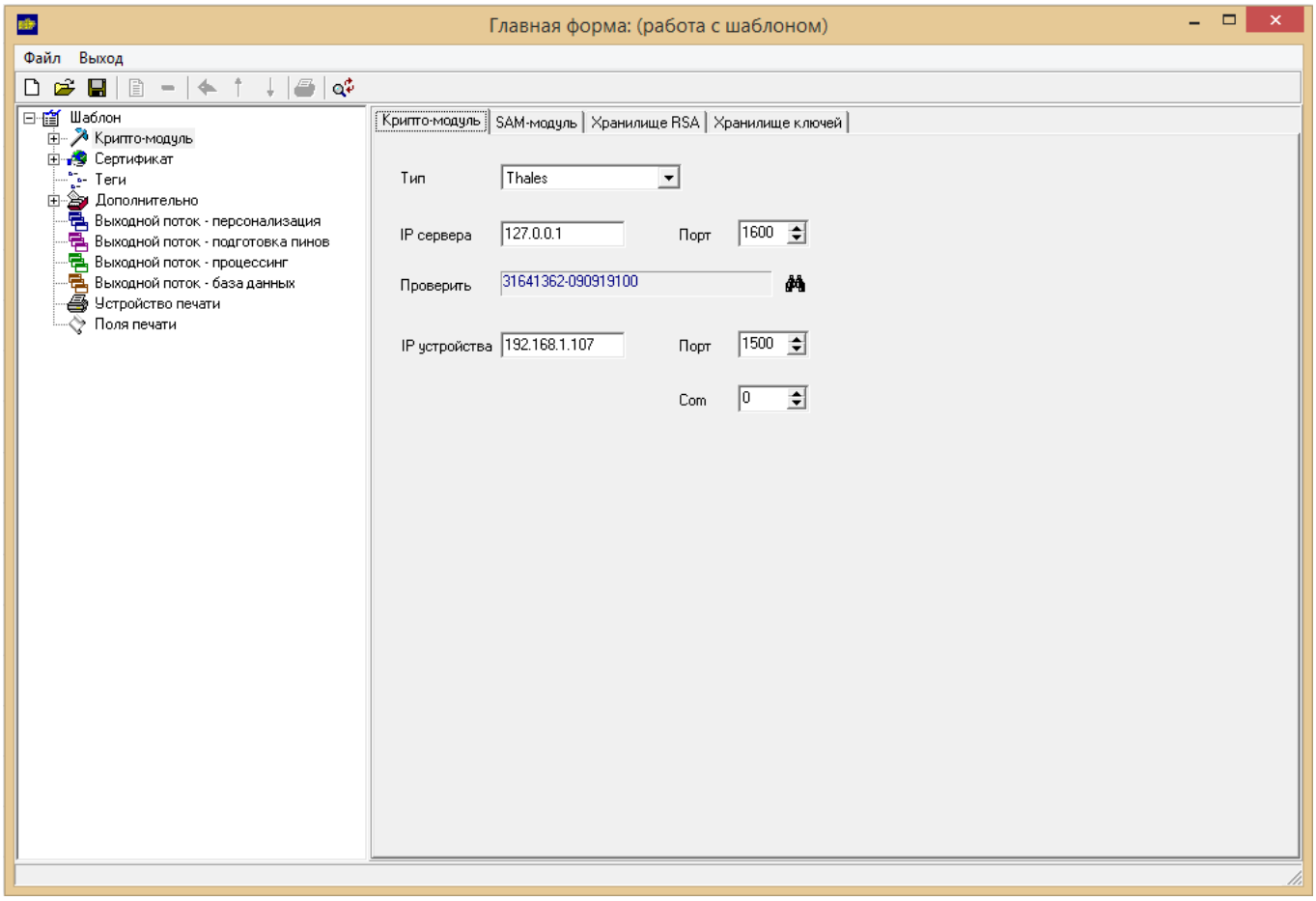
4.1.1.2. Крипто-модуль Eracom



Тип – тип крипто-модуля, определяемого типом устройства шифрования.
Область приложения – область в крипто-модуле, в которой будут храниться данные.
IP сервера, Порт, Номер слота – коммуникационные настройки HS-сервера.
Индекс LMK – идентификатор мастер-ключа, через который будут выполняться операции шифрования.

Настройки крипто-модуля можно проверить используя проверочную кнопку  ***Проверить соединение***. В случае установления правильных настроек в ответ на нажатие данной кнопки вернется проверочное значение, представляющее собой контрольную сумму мастер-ключа указанного индекса, отображённое в поле «Проверить». В противном случае вернется причина ошибки, отображённая в том же поле. Может быть несколько мастер-ключей, каждый из которых будет использоваться для своих целей.

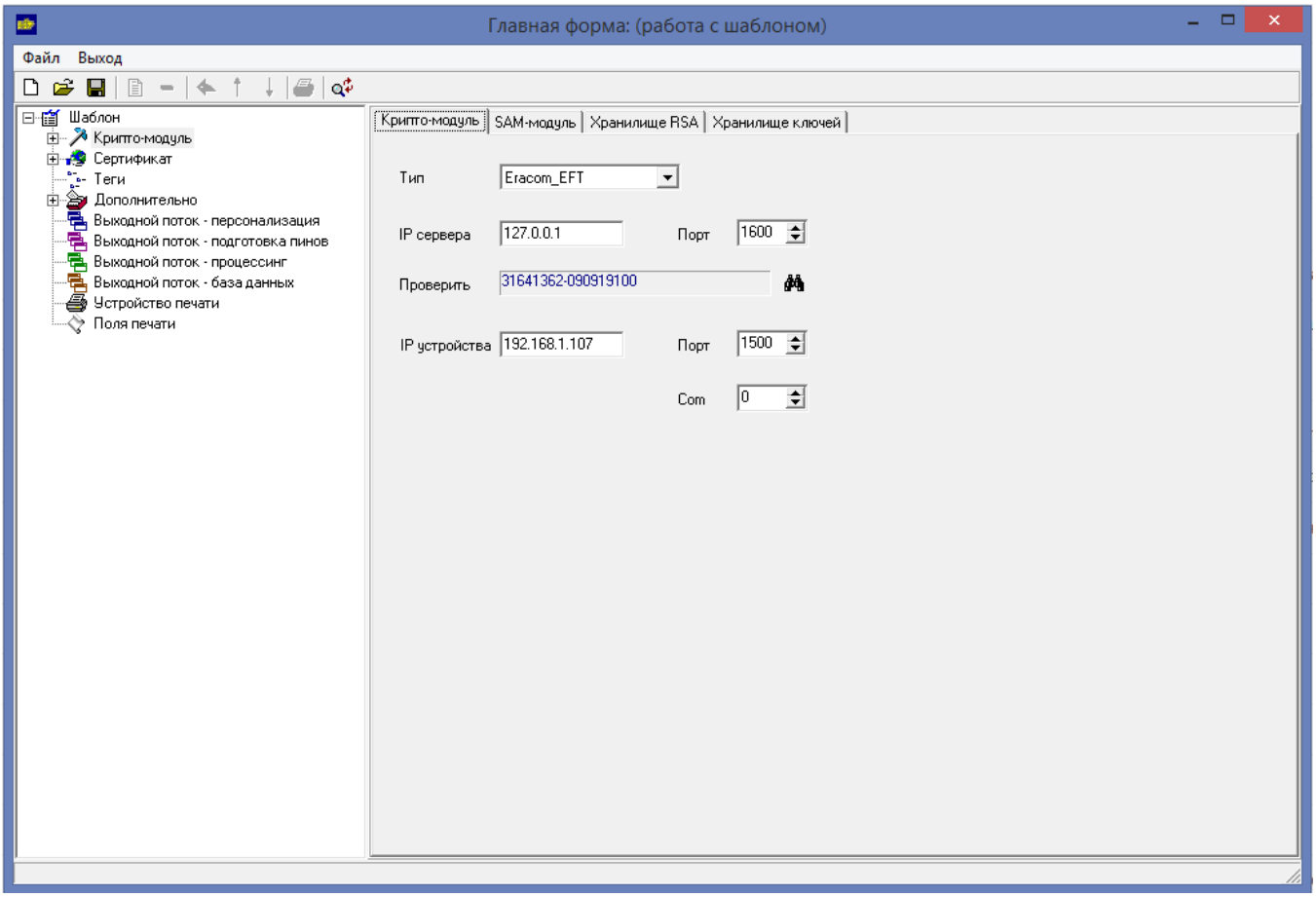
4.1.1.3. Крипто-модуль Thales



Тип – тип крипто-модуля, определяемого типом устройством шифрования.
IP сервера, порт – коммуникационные настройки HS-сервера.
IP устройства, порт, Com – коммуникационные настройки устройства шифрования. Если значение параметра Com=0, то соединение с устройством шифрования осуществляется по IP-адресу. В остальных случаях соединение с устройством шифрования осуществляется по Com-порту по указанному в данном параметре номеру порта.

Настройки крипто-модуля можно проверить используя проверочную кнопку  ***Проверить соединение***. В случае установления правильных настроек в ответ на нажатие данной кнопки вернется номер версии прошивки Thales, отображённый в поле «Проверить». В противном случае вернется причина ошибки, отображённая в том же поле.

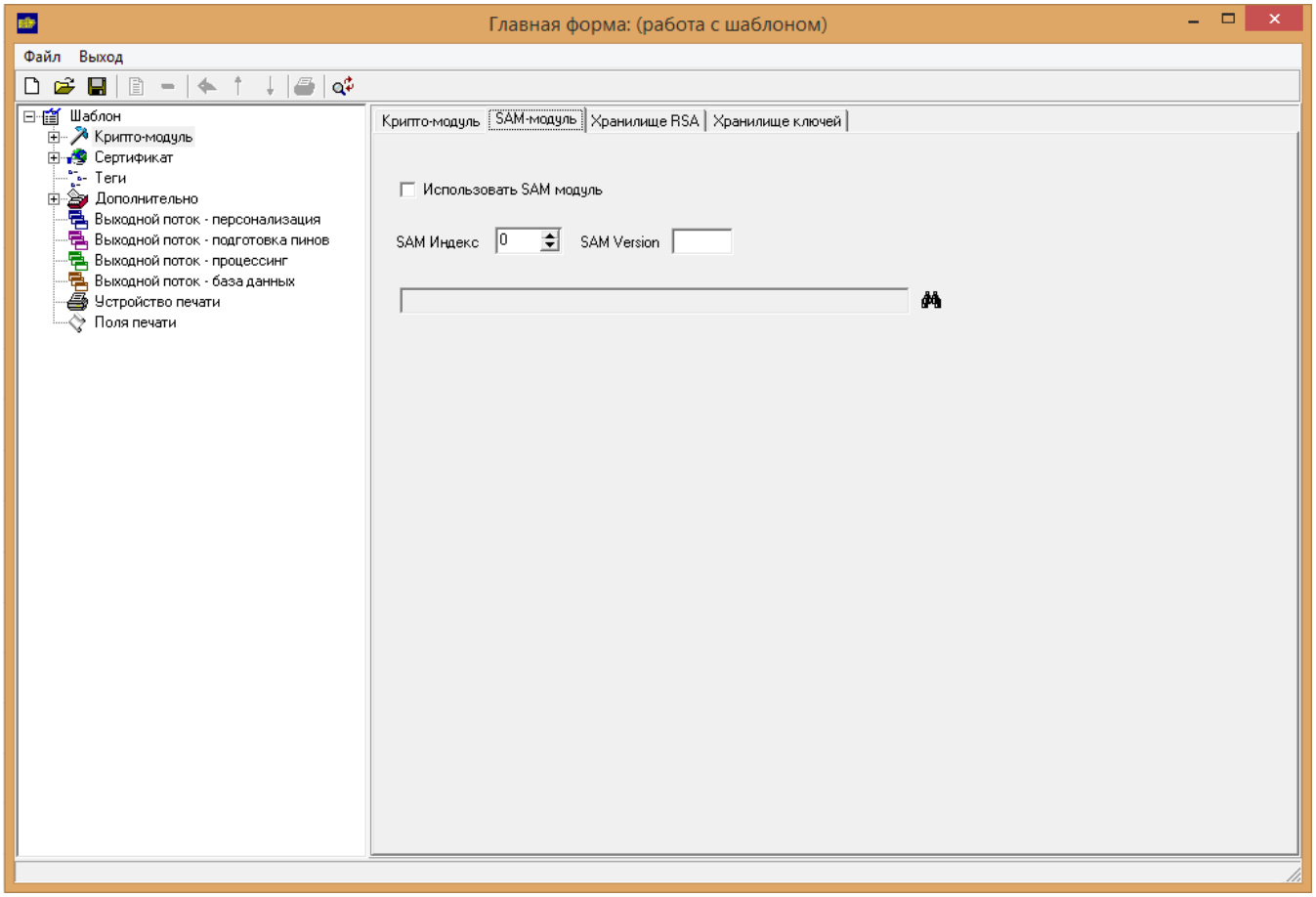
4.1.1.4. Крипто-модуль Eracom_EFT



Тип – тип крипто-модуля, определяемого устройством шифрования.
IP сервера, порт – коммуникационные настройки HS-сервера.
IP устройства, порт, Com – коммуникационные настройки устройства шифрования. Если значение параметра Com=0, то соединение с устройством шифрования осуществляется по IP-адресу. В остальных случаях соединение с устройством шифрования осуществляется по Com-порту по указанному в данном параметре номеру порта.

Настройки крипто-модуля можно проверить используя проверочную кнопку  ***Проверить соединение***. В случае установления правильных настроек в ответ на нажатие данной кнопки вернется номер версии прошивки Eracom_EFT, отображённый в поле «Проверить». В противном случае вернется причина ошибки, отображённая в том же поле.

4.1.2. SAM-модуль

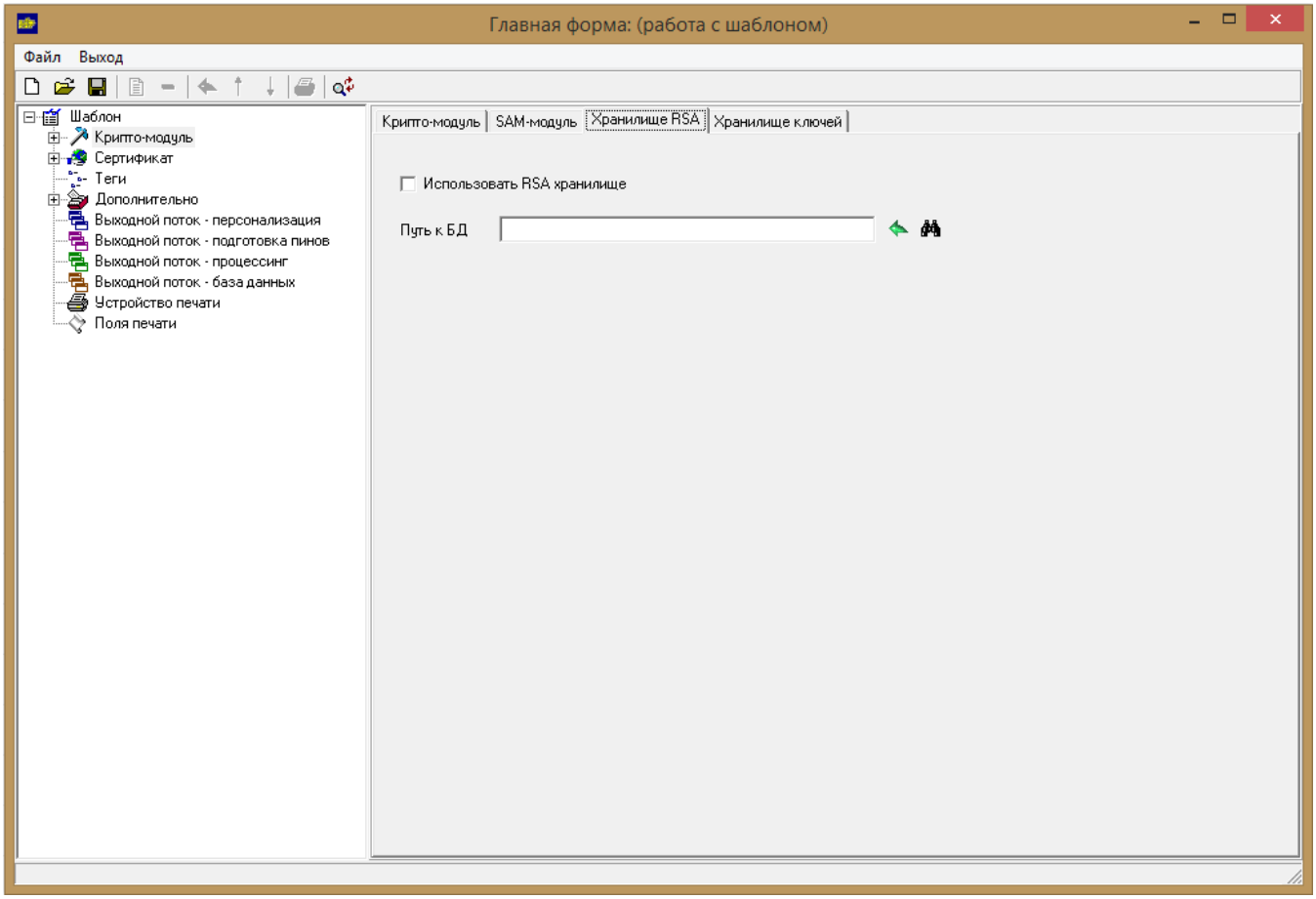


.

SAM Индекс – .
SAM Version – .
Доступность ??? можно проверить используя проверочную кнопку ***Проверить соединение***.

Использование SAM-модуля является дополнительной опцией программы CDP.

4.1.3. Хранилище RSA



Существует возможность ***использовать RSA хранилище*** для работы с RSA ключами. Файл базы данных RSA ключей предварительно готовится отдельной специальной программой. В момент подготовки данных компоненты RSA ключей будут браться из файла базы данных, что существенно ускорит сам процесс подготовки данных, так как сами RSA ключи сгенерированы заранее, и дополнительного времени для расчётов RSA ключей не требуется.

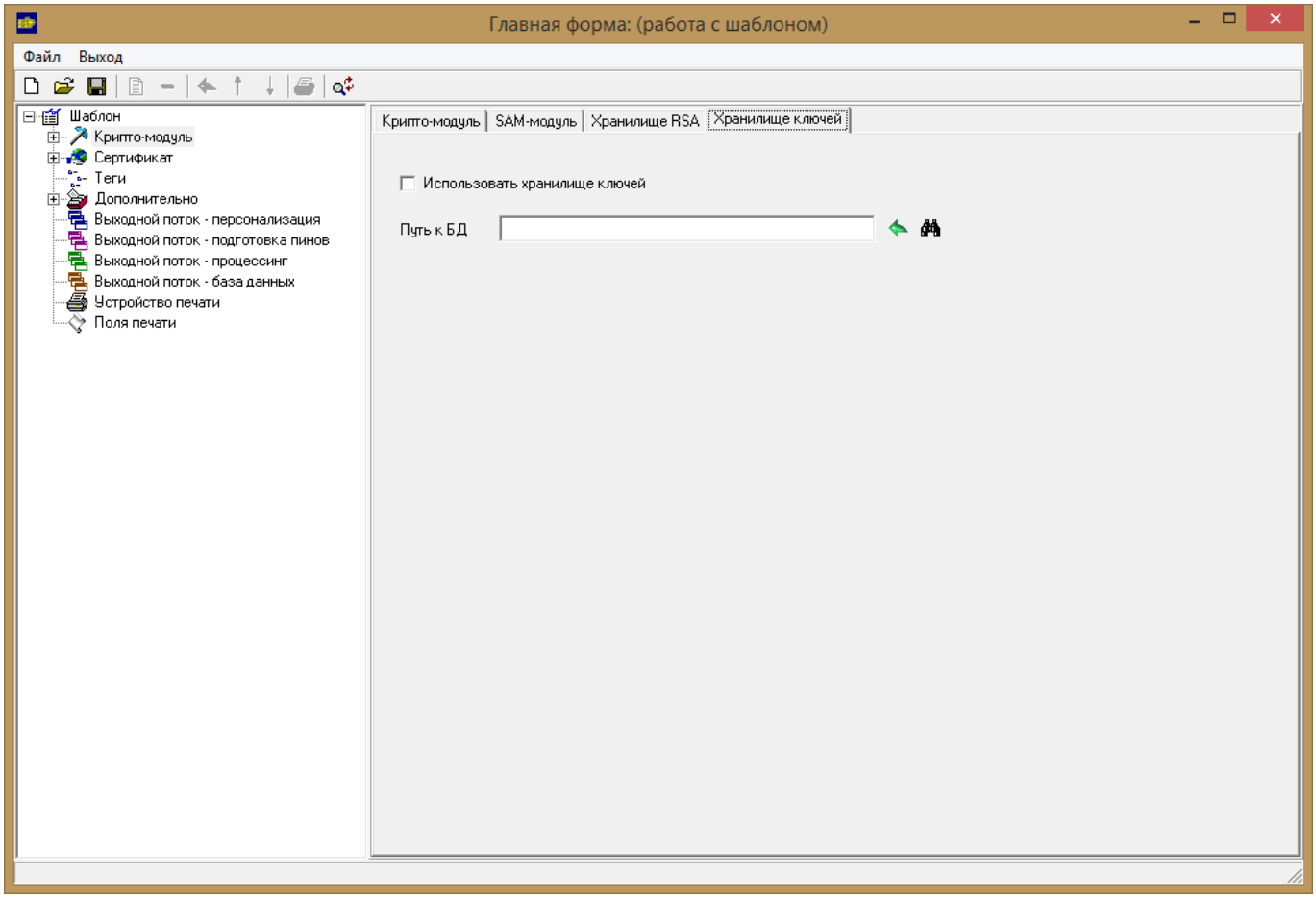
Путь к БД – путь к специальной базе данных в виде файла, в которой хранятся заранее сгенерированные RSA ключи.

Доступность базы данных можно проверить используя проверочную кнопку  ***Проверить соединение***.

Использование RSA хранилища является дополнительной опцией программы CDP.

В дальнейшем некоторые параметры и свойства использования Хранилища RSA будут незначительно отличаться, в зависимости от выбранного типа крипто-модуля.

4.1.4. Хранилище ключей (файл базы данных)



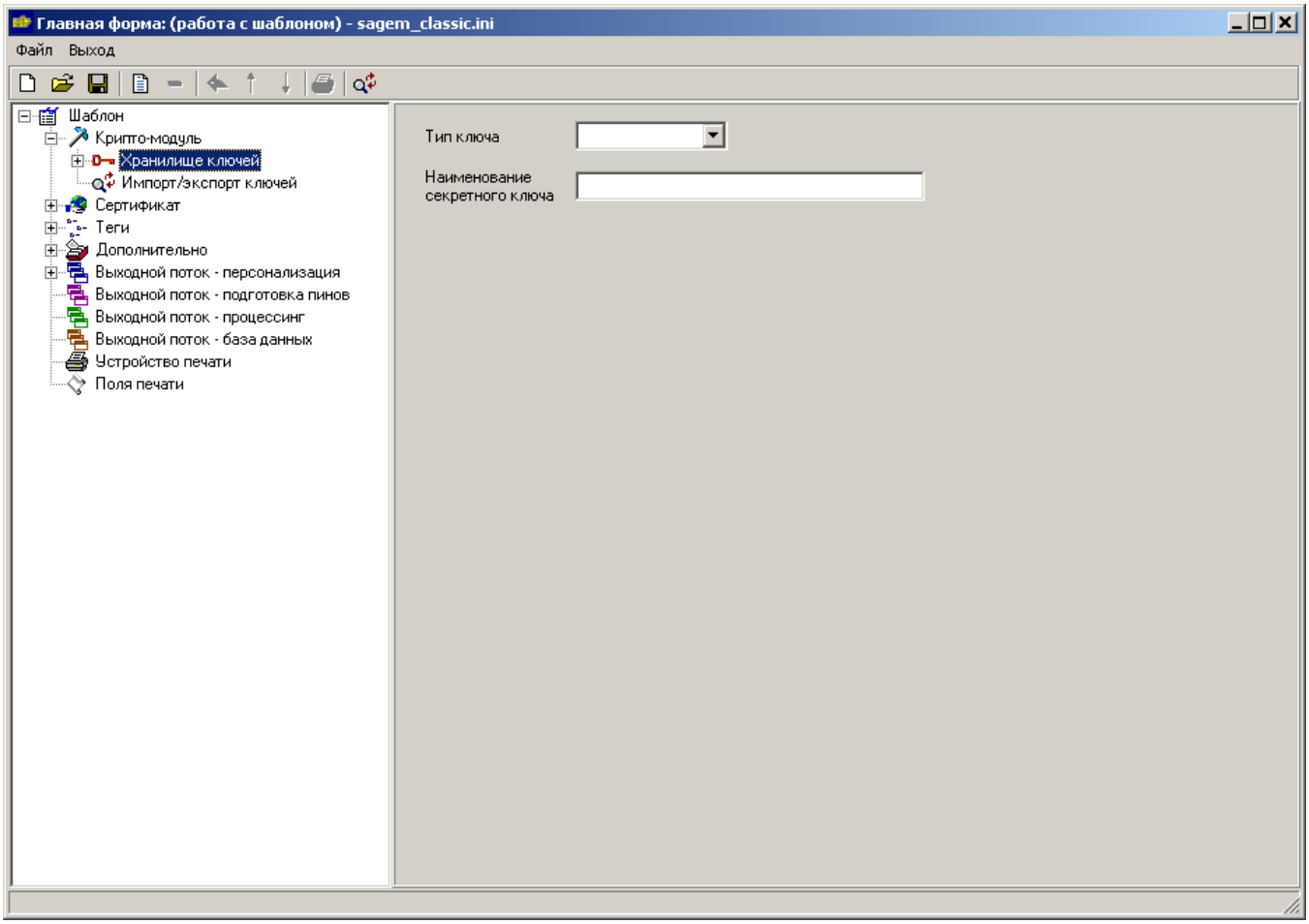
Путь к БД – путь к специальной базе данных в виде файла, в которой хранятся заранее сгенерированные ключи.

Доступность базы данных можно проверить используя проверочную кнопку ***Проверить соединение.***

Использование Хранилища ключей является дополнительной опцией программы CDP.

4.1.5. Хранилище ключей

После настройки крипто-модуля, необходимо определить ключи, которые будут использоваться для шифрования. Их можно либо импортировать из внешних источников, либо сгенерировать новые. Генерировать ключи можно как через специальную консоль, являющейся отдельной программой, выполняющую административные функции, связанные с крипто-модулем, так и через настоящий программный модуль «Хранилище ключей» системы подготовки данных CDP.



Тип ключа – тип создаваемого ключа, существует несколько типов ключей: **DES, DES_2, RSA, RSA_DDA**

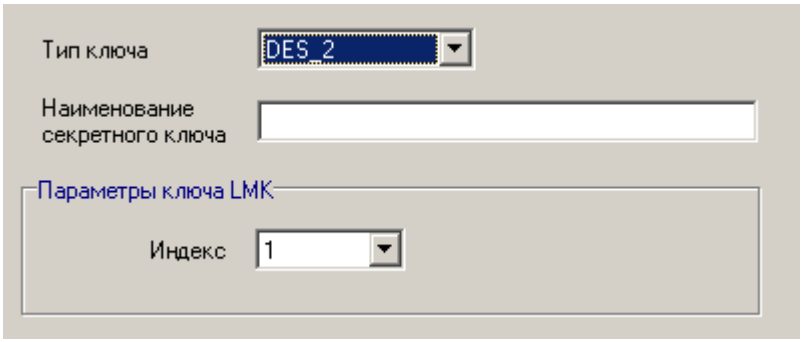
DES – ключ одинарной длины (8 байтов);
DES2 – ключ двойной длины (16 байтов);
RSA – пара ключей двойной длины, имеющие общие свойства;
RSA_DDA – пара ключей двойной длины, имеющие общие свойства, которые создаются в крипто-модуле в процессе подготовки данных;

Наименование секретного ключа – имя секретного ключа.

Ключи типа **DES** или **DES2** хранятся в шаблоне, в виде криптограмм под мастер-ключом.

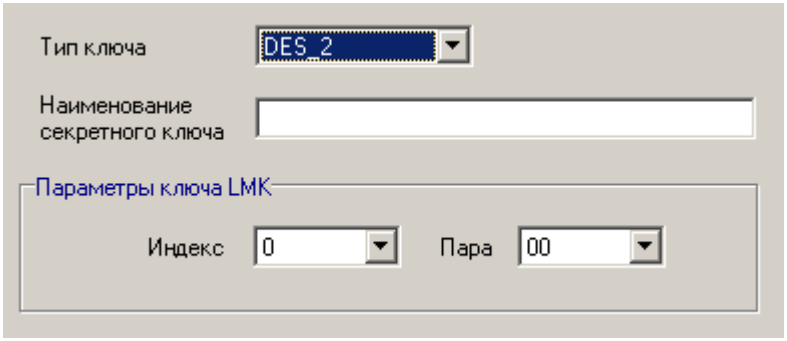
4.1.5.1. Генерация ключей типа DES/DES2.

1) В случае использования крипто-модуля типа Eracom.



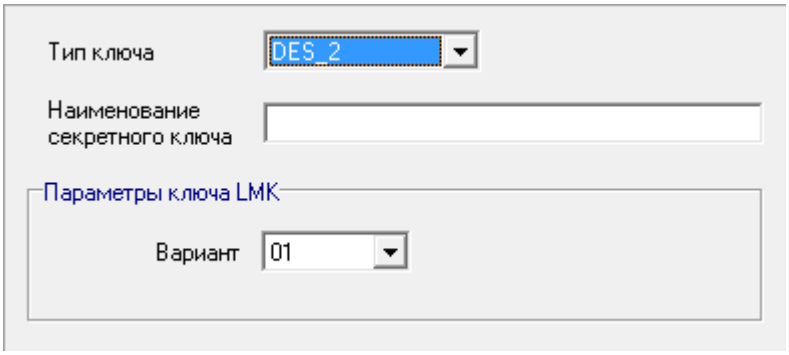
Шифрование происходит под мастер-ключом, который определяется **индексом LMK**.

2) В случае использования крипто-модуля типа Thales.



Шифрование происходит под мастер-ключом, который определяется **индексом (variant) и парой (pair)** (в соответствии с таблицей ключей в Thales).

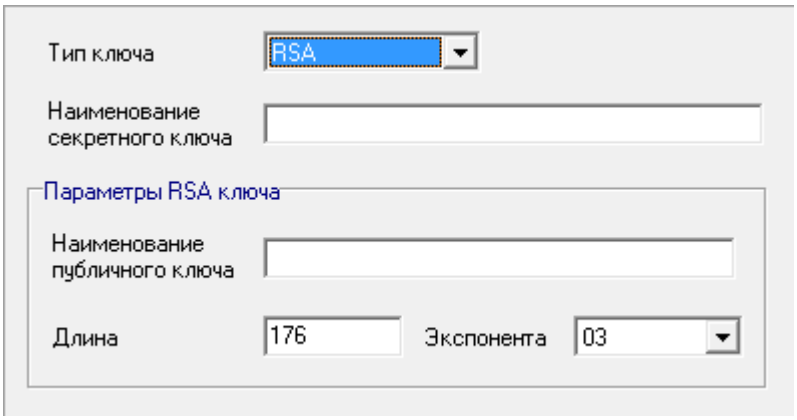
3) В случае использования крипто-модуля типа Eracom_EFT.



Шифрование происходит под мастер-ключом, который определяется **вариантом LMK**.

4.1.5.2. Генерация ключей типа RSA.

Если выбран тип ключа **RSA**, то будет создана пара ключей, с общими свойствами. Один ключ будет Public, другой Private. В случае использования в качестве крипто-модуля Eracom, RSA ключи будут созданы в самом крипто-модуле. В случае использования в качестве крипто-модуля Thales, ключи будут хранится в шаблоне, в виде криптограмм под определенным мастер-ключом.

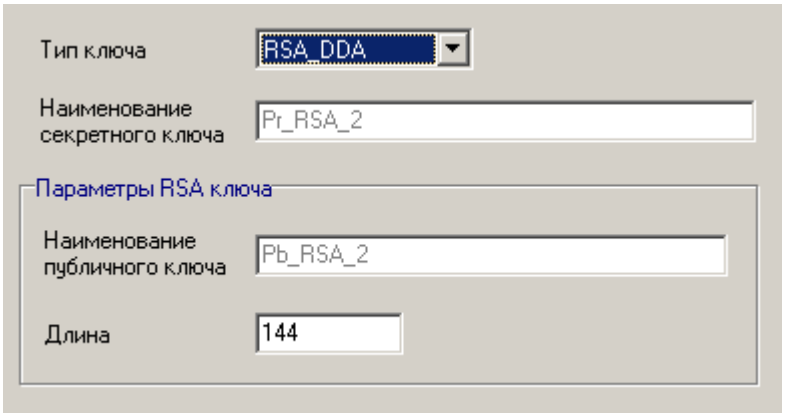


Наименование публичного ключа – имя публичного ключа.
Длина – длина RSA ключа, обычно используется 176 байта.
Экспонента – значение экспоненты RSA ключа.

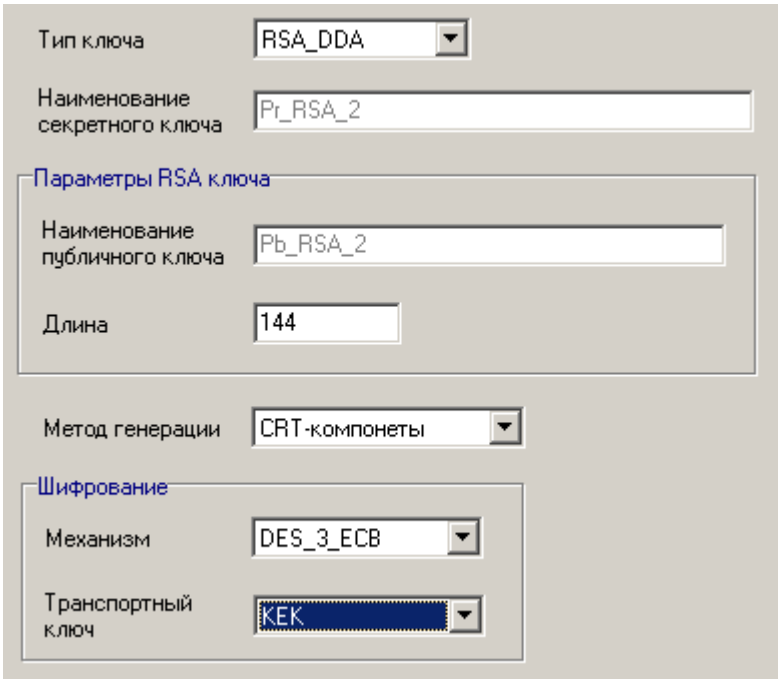
4.1.5.3. Генерация ключей типа RSA_DDA.

Если выбран тип ключа **RSA_DDA**, то в шаблоне будет создана пара ключей с общими свойствами, которые будут динамически создаваться и уничтожаться в крипто-модуле, в процессе выпуска задания. Один ключ будет называться **Pr_RSA_N**, другой **Pb_RSA_N**. Где N количество динамических пар в шаблоне.

1) В случае использования крипто-модуля типа Eracom.



2) В случае использования крипто-модуля типа Thales/Eracom_EFT.



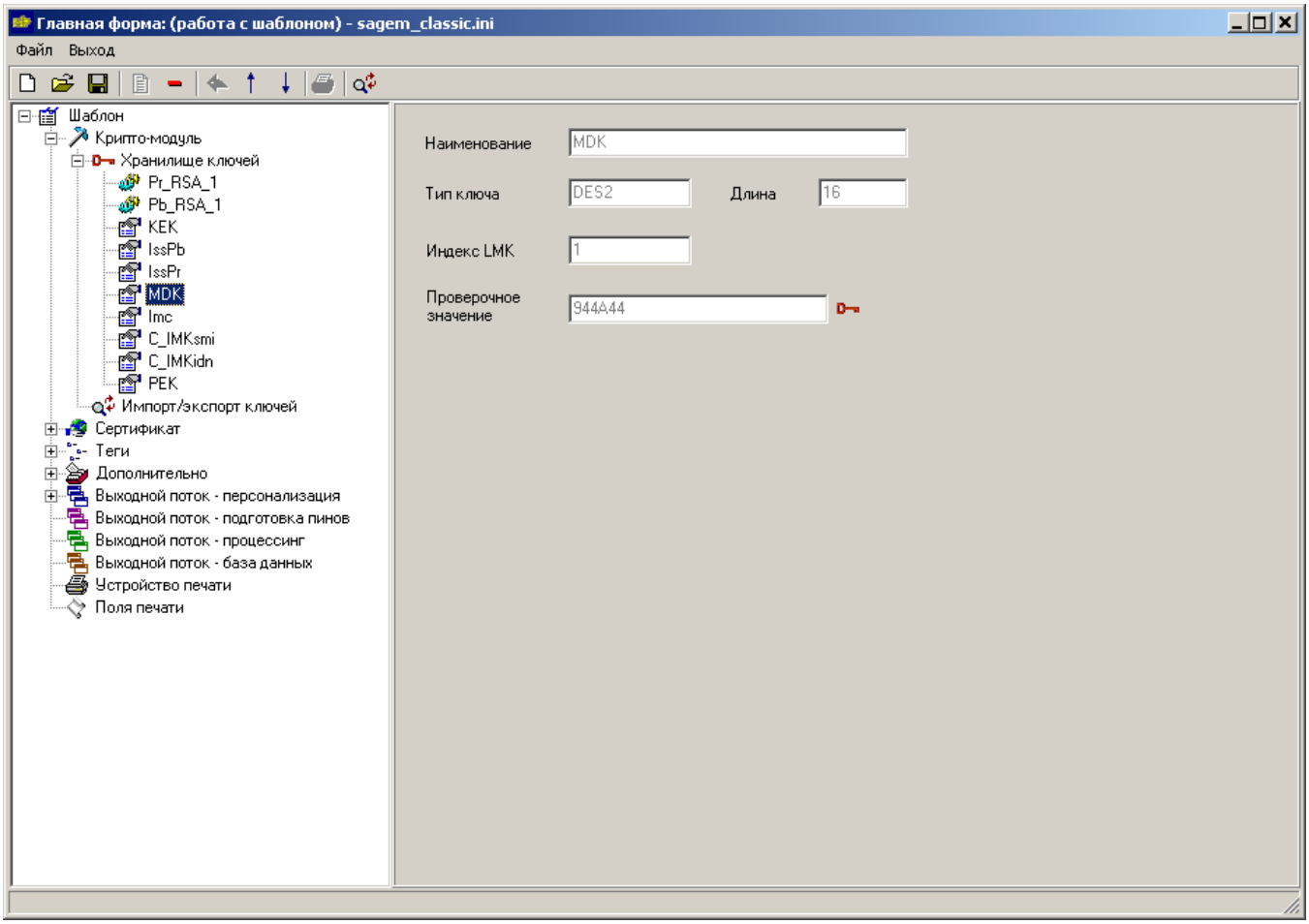
Если используется крипто-модуль Thales/Eracom_EFT, то необходимо задать дополнительные параметры:

Метод генерации **–** существует два вида генерации RSA ключей, в виде CRT-компонент или Экспоненты.
Шифрование **–** используется для шифрования, полученного ключа.
Механизм – механизм, который будет использоваться для шифрования.
Транспортный ключ – один из ключей, который будет использоваться для шифрования.

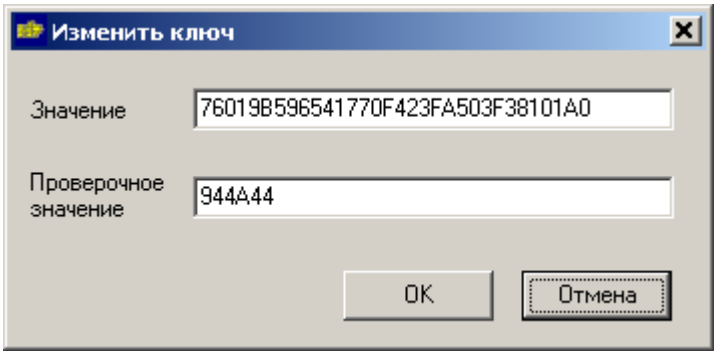
После того как заполнены все необходимые параметры в правой части экрана, необходимо нажать кнопку ***Добавить элемент***, расположенную в главном меню. В случае правильных параметров будет добавлен новый ключ, который отобразится в хранилище ключей. В случае ошибки (неправильные параметры, недоступность крипто-модуля и.т.д.) будет выдано сообщение об ошибке.

4.1.5.4. Работа с ключами в хранилище.

Для просмотра параметров ключей требуется выделить ключ в левой части экрана.
В правой части отобразятся его свойства. Пример отображения параметров ключа указан на приведённом ниже рисунке.



Можно вручную изменить значение ключа и его проверочное значение используя кнопку ***Заменить ключ***. Эта опция доступна только для ключей типа **DES** и **DES2**.



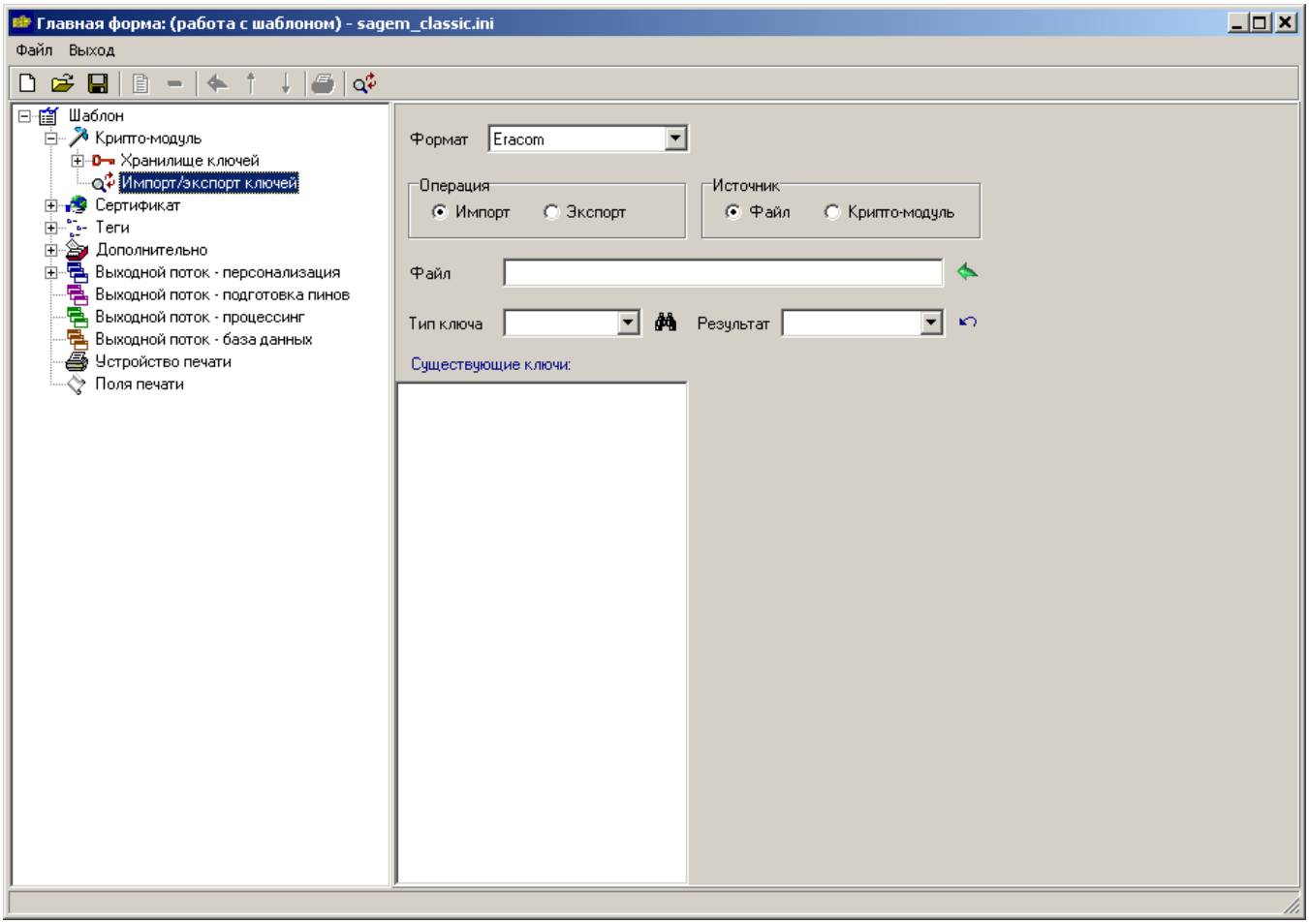
Удаление ключа из хранилища осуществляется использованием кнопки ***Удалить элемент*** главного меню.

Перемещение ключа по списку ключей в хранилище для удобства отображения осуществляется кнопками ***Вверх*** и ***Вниз*** главного меню.

4.1.6. Импорт/экспорт ключей

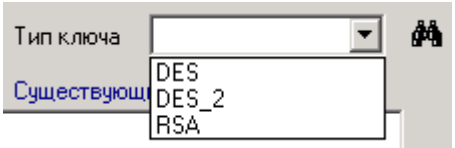
Используется для импорта в шаблон существующих ключей из различных источников (из файла или крипто-модуля), а также для экспорта в файл ключей, хранящихся в шаблоне в хранилище ключей.

4.1.6.1. Формат Eracom

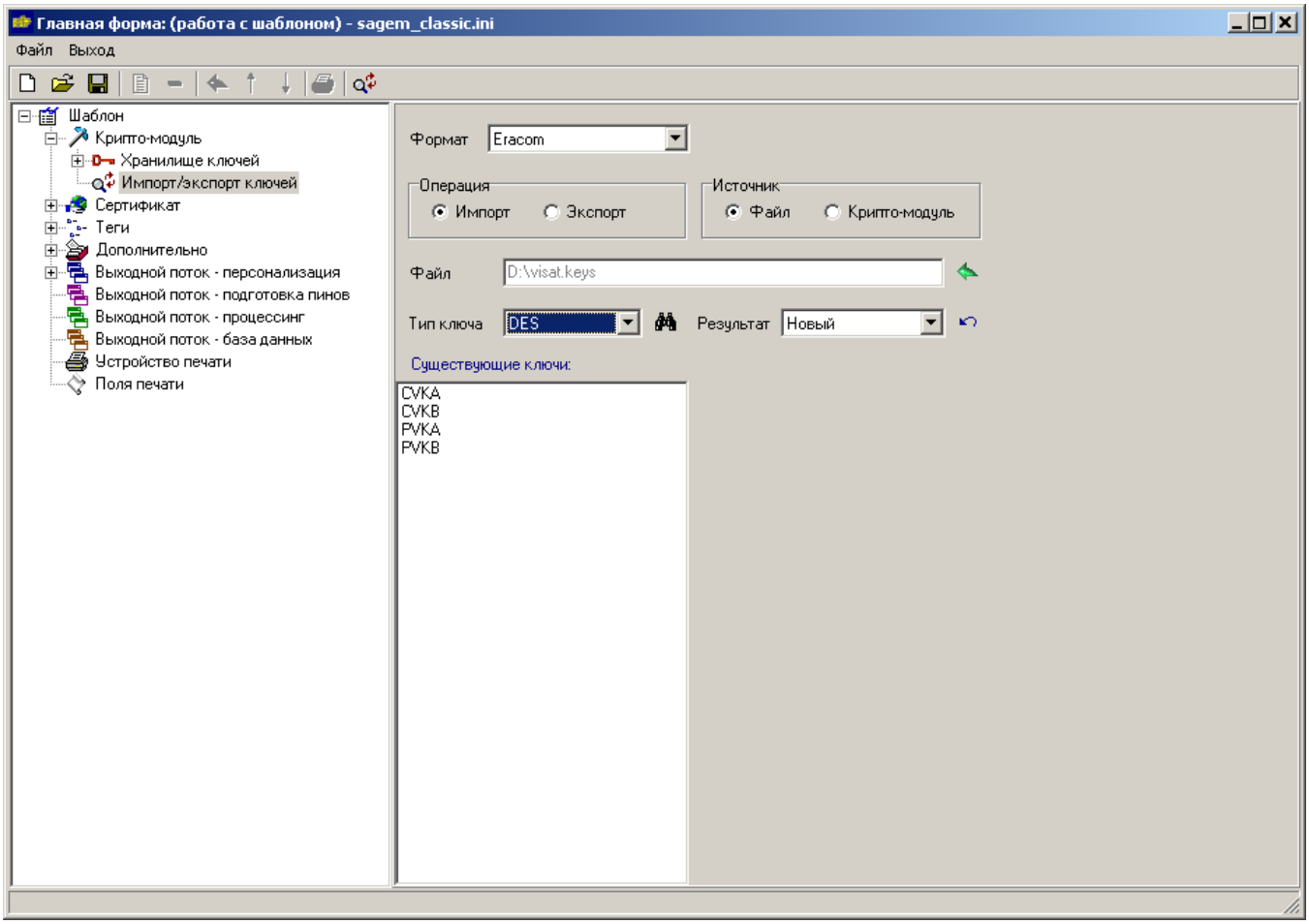


Операция ***Импорт*** – при создании нового шаблона, иногда возникает необходимость использовать уже существующие ключи, хранящиеся в крипто-модуле или специальном файле. Для этого нужно импортировать ключи в текущий шаблон.

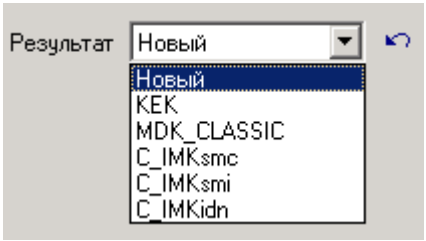
Для импорта ключей необходимо выбрать **источник**, откуда будут импортироваться ключи. Если в качестве источника выбирается **файл**, то нужно указать исходный файл (файлы с ключами имеют расширение .keys). Далее надо выбрать тип ключа и нажать кнопку ***Получить существующие ключи***.



Результатом операции будет список всех ключей (определенного типа), хранящихся во входном источнике.



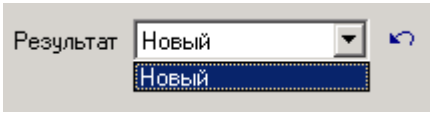
Результат – если требуется импортировать в шаблон новый ключ, то необходимо выбрать **новый**. Если требуется заменить ключ, то необходимо указать один из существующих в шаблоне ключей, вместо которого будет импортироваться ключ.



Далее необходимо выделить ключ и нажать кнопку ***Импортировать ключ***. В случае удачного завершения операции появится сообщение, что ключ успешно импортирован. Сам ключ со всеми свойствами появится в хранилище ключей.

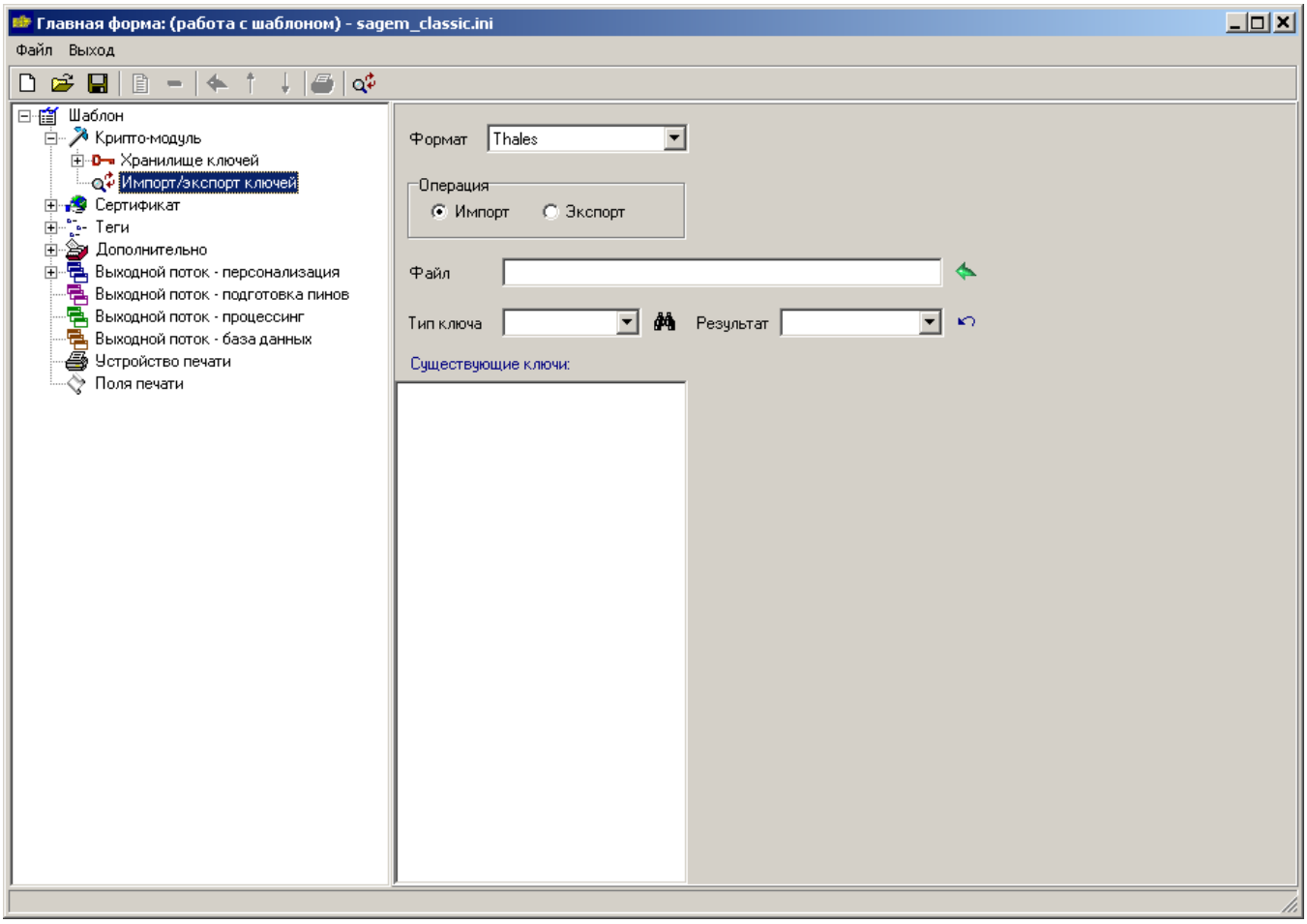
Операция ***Экспорт*** – используется для сохранения ключей в специальном файле, с возможностью дальнейшего восстановления их в данном шаблоне, или импорта в другой шаблон. Для этого необходимо выбрать файл, в который будут экспортироваться ключи (при присвоении имени файлу с ключами необходимо вручную добавить расширение .keys).

Далее надо выбрать тип ключа, и нажать кнопку ***Получить существующие ключи***. Результатом операции будет список всех ключей (определенного типа), хранящихся в шаблоне. В файл можно экспортировать только новый ключ.



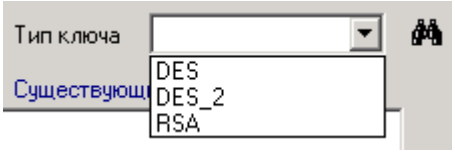
Если необходимо экспортировать ключ в файл, то нужно выделить ключ и нажать кнопку ***Экспортировать ключ***. В случае удачного завершения операции появится сообщение, что ключ успешно экспортирован. Сам ключ со всеми свойствами сохранится в выходном файле.

4.1.6.2. Формат Thales/Eracom_EFT

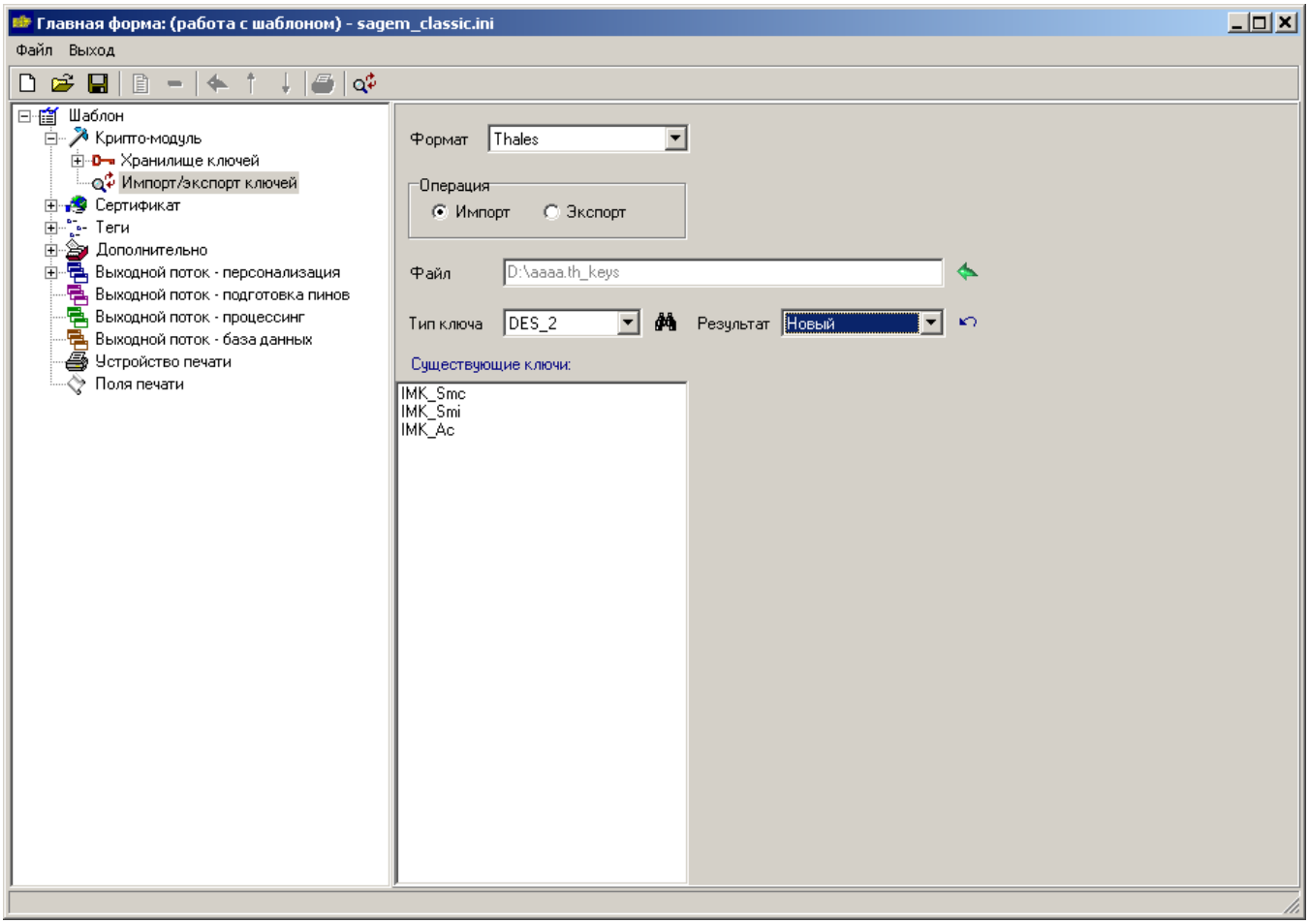


Операция ***Импорт*** – при создании нового шаблона, иногда возникает необходимость использовать уже существующие ключи. Для этого нужно импортировать ключи в текущий шаблон.

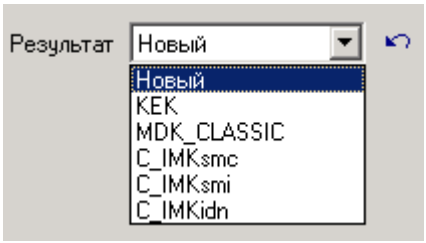
Импортировать ключи можно только из файла (файлы с ключами имеют расширение th_keys). Далее надо выбрать тип ключа и нажать кнопку ***Получить существующие ключи***.



Результатом операции будет список всех ключей (определенного типа), хранящихся в файле.



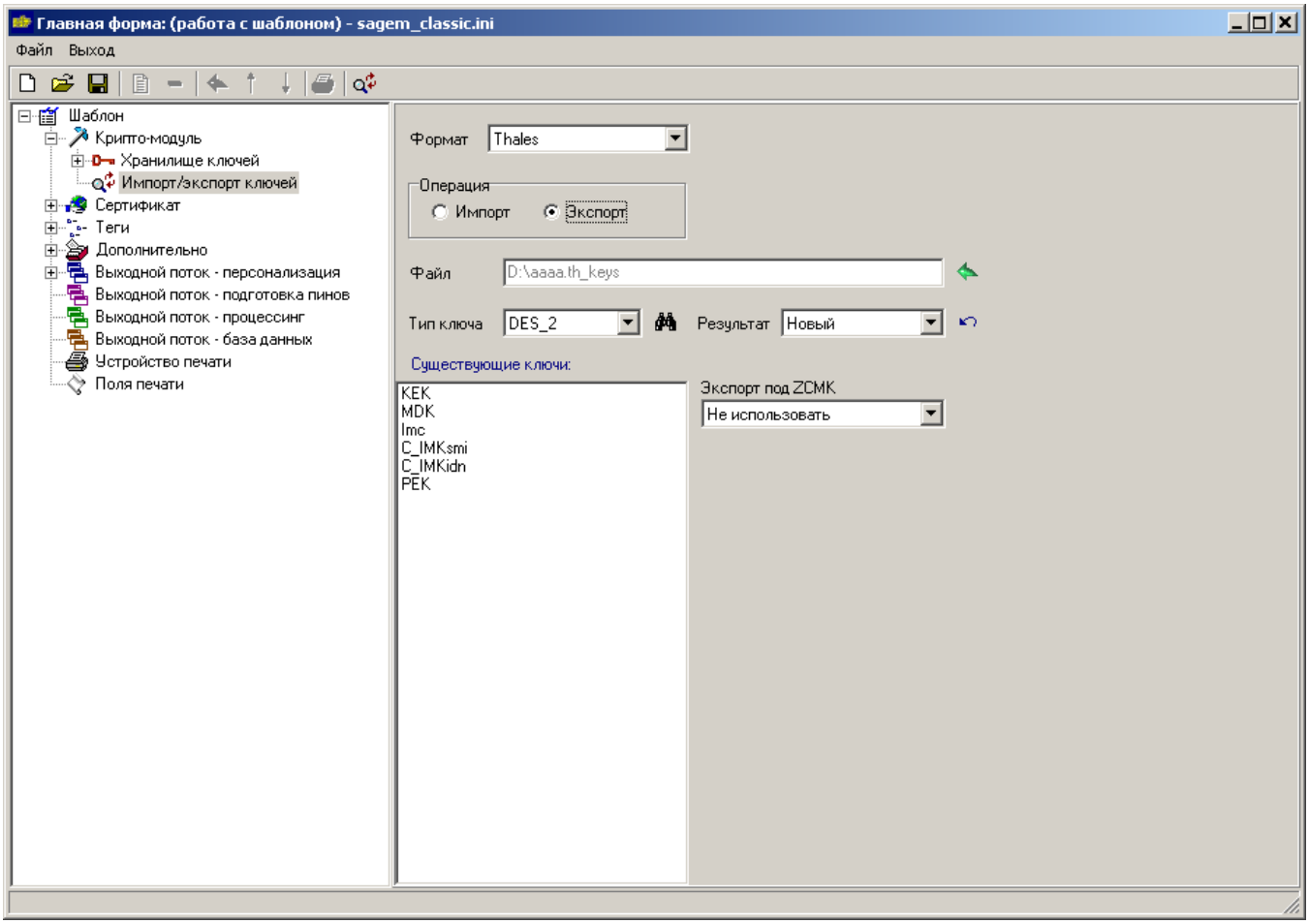
Результат – если требуется импортировать в шаблон новый ключ, то необходимо выбрать **новый**. Если требуется заменить ключ, то необходимо указать один из существующих в шаблоне ключей, вместо которого будет импортироваться ключ.



Далее необходимо выделить ключ и нажать кнопку ***Импортировать ключ***. В случае удачного завершения операции появится сообщение, что ключ успешно импортирован. Сам ключ со всеми свойствами появится в хранилище ключей.

Операция ***Экспорт*** – используется для сохранения ключей в специальном файле, с возможностью дальнейшего восстановления их в данном шаблоне, или импорта в другой шаблон. Для этого необходимо выбрать файл, в который будут экспортироваться ключи (при присвоении имени файлу с ключами необходимо вручную добавить расширение .th_keys).

Далее надо выбрать тип ключа, и нажать кнопку ***Получить существующие ключи***. Результатом операции будет список всех ключей (определенного типа), хранящихся в шаблоне.



Если необходимо экспортировать ключ в файл, то нужно выделить ключ и нажать кнопку ***Экспортировать ключ***.

В файл можно экспортировать как криптограмму под мастер-ключом, так и перешифровать криптограмму под ключ ZCMK. Для этого надо выбрать в поле ***Экспорт под ZCMK*** ключ, под которым осуществится перешифрование.

В случае удачного завершения операции появится сообщение, что ключ успешно экспортирован. Сам ключ со всеми свойствами сохранится в выходном файле.

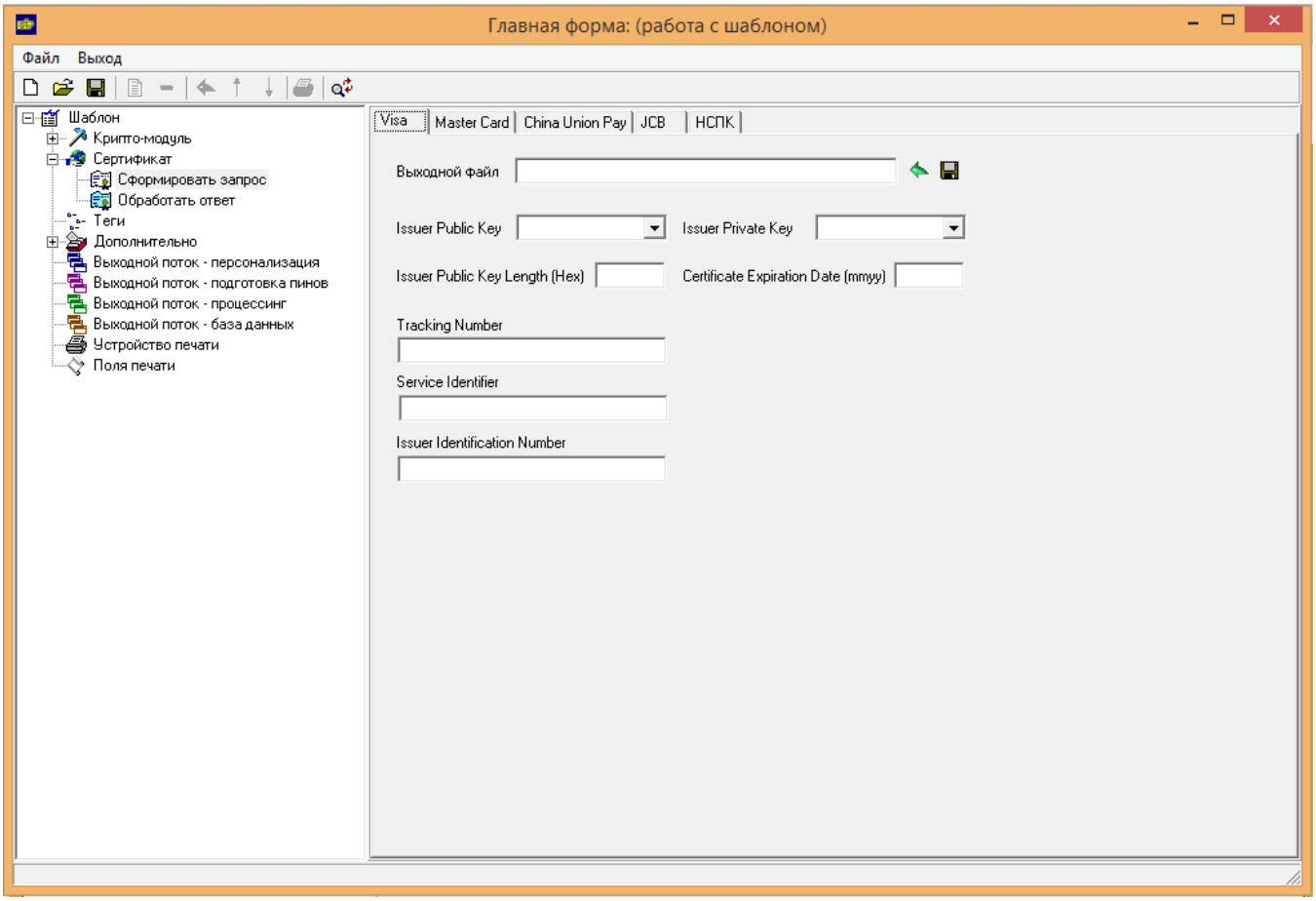
4.2. Сертификат

Служит для формирования запросов в платежные системы на получение сертификатов, и обработке файлов ответа. Запросы и ответы формируются по специальным алгоритмам, описанным в документации платежных систем.

4.2.1. Сформировать запрос

Формирует запрос в платежную систему на получение сертификата.

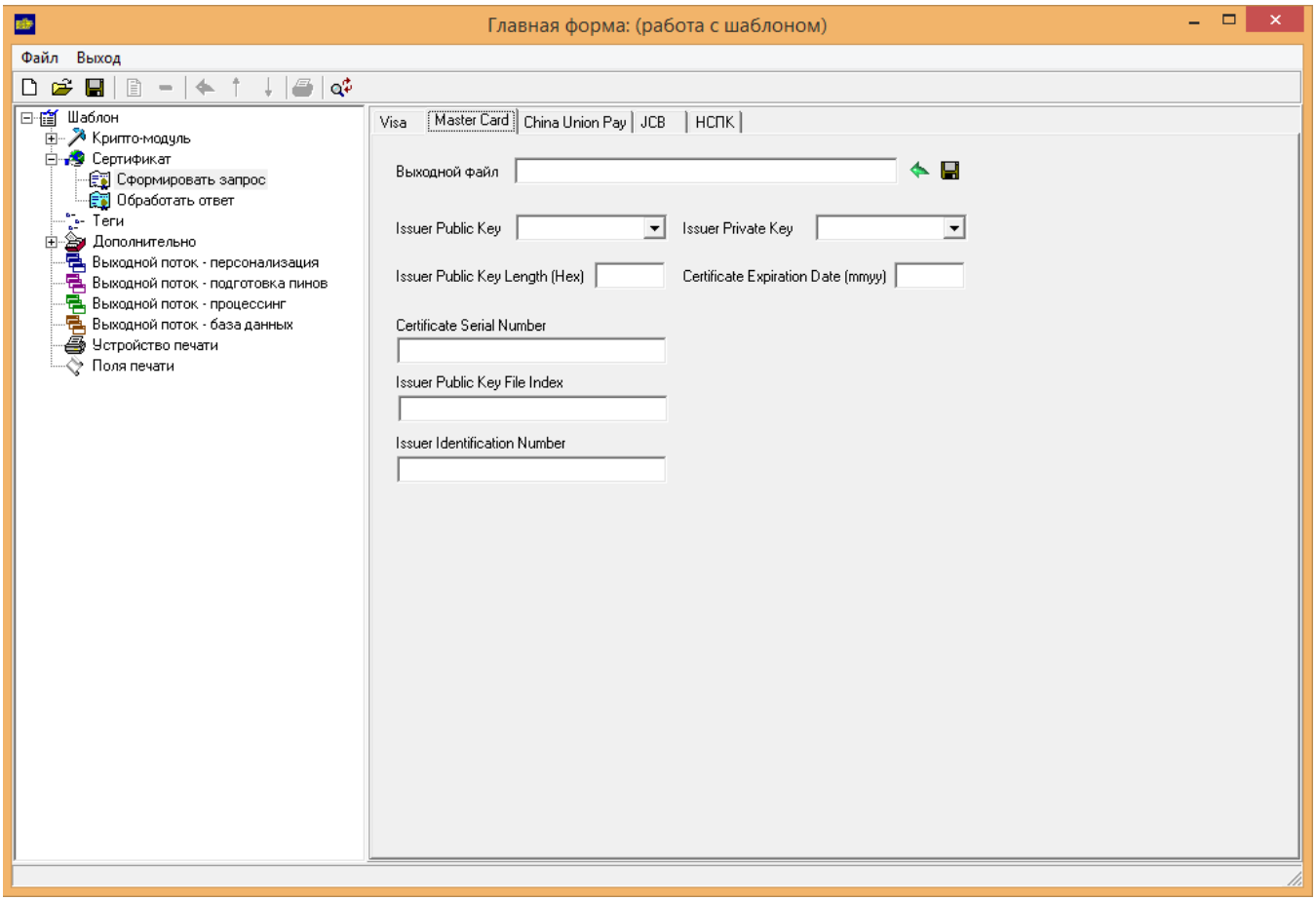
4.2.1.1. Платежная система Visa



Выходной файл – файл, который будет отправлен в платежную систему. Необходимо выбрать директорию, в которую он будет сохранен. Имя файла и расширение сформируется автоматически.
Issuer Public Key – публичный ключ эмитента.
Issuer Private Key – секретный ключ эмитента.
Публичный и секретный ключи имеют тип **RSA**, и должны быть предварительно созданы или импортированы в шаблон.
Issuer Public Key Length (Hex) – длина публичного ключа эмитента в Hex. Если ключ имеет длину 176 байта, соответственно в Hex его длина будет B0.
Certificate Expiration Date (mmyy) – срок действия сертификата.
Tracking Number – 6-значный номер, присылаемый платежной системой.
Service Identifier – 4-значный идентификатор выпускаемого продукта.
Issuer Identification Number – в качестве 6-значного идентификатора используется начало бина, который определяет эмитента и выпускаемый продукт.

После того, как все поля заполнены, необходимо нажать кнопку **Сформировать запрос**. В результате в выбранной директории будет сформирован файл с именем **ccxxxxxx.inp** (xxxxxx - Tracking Number). Правильность полученного файла можно проверить с помощью специальных программ, предоставляемых платежной системой.

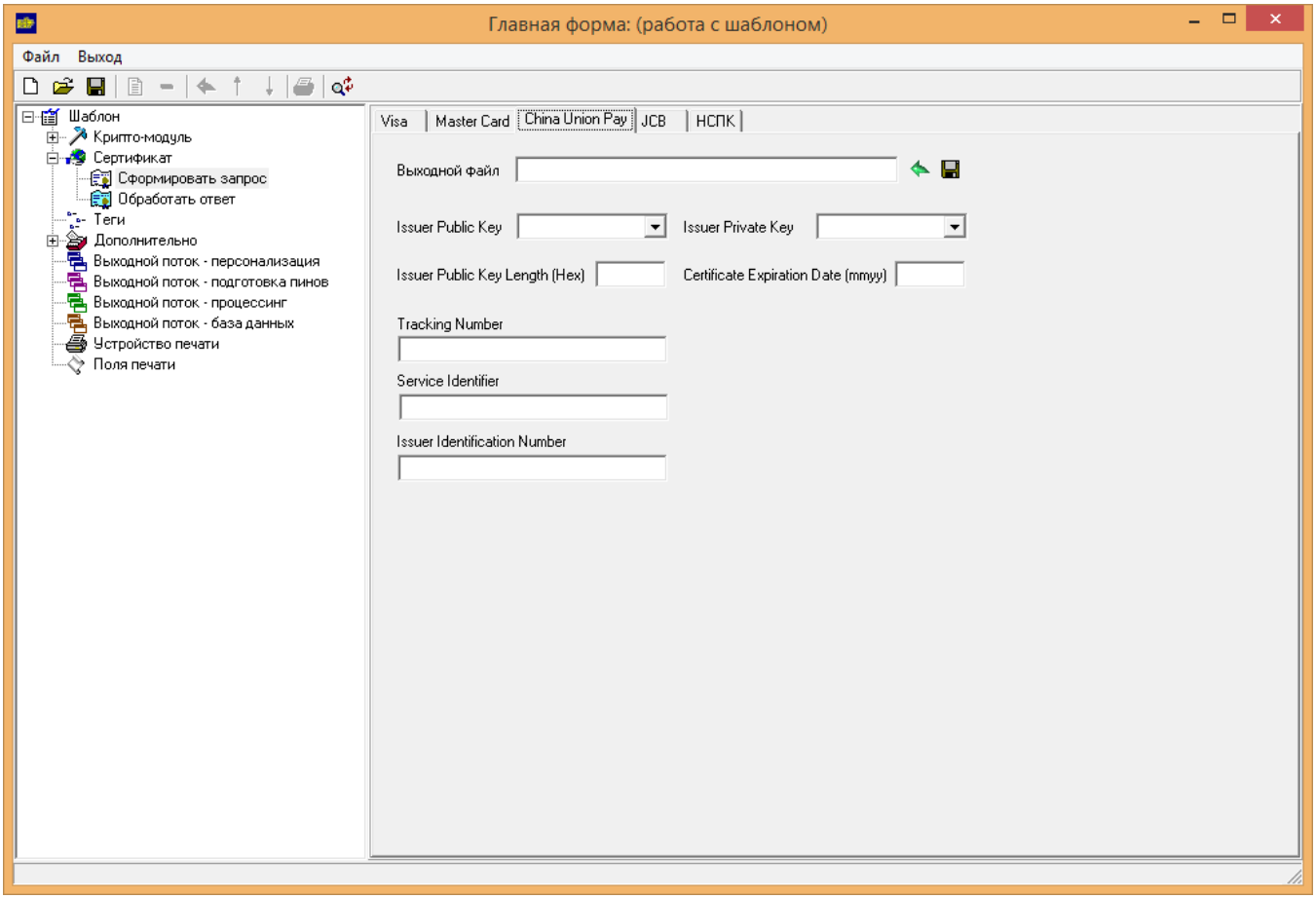
4.2.1.2. Платежная система Master Card



Выходной файл – файлы, которые будут отправлены в платежную систему. Необходимо выбрать директорию, в которую они будут сохранены. Имена файлов и расширение сформируется автоматически.
Issuer Public Key – публичный ключ эмитента.
Issuer Private Key – секретный ключ эмитента. Публичный и секретный ключи имеют тип **RSA**, и должны быть предварительно созданы или импортированы в шаблон.
Issuer Public Key Length (Hex) – длина публичного ключа эмитента в Hex. Если ключ имеет длину 176 байта, соответственно в Hex его длина будет B0.
Certificate Expiration Date – срок действия сертификата.
Certificate Serial Number – 6-значный номер, присылаемый платежной системой (обычно используется начало бина, который определяет эмитента и выпускаемый продукт).
Issuer Public Key File Index – 6-значный идентификатор индекса публичного ключа.
Issuer Identification Number –в качестве 6-значного идентификатора используется начало бина, который определяет эмитента и выпускаемый продукт.

После того, как все поля заполнены, необходимо нажать кнопку **Сформировать запрос**. В результате в выбранной директории будет сформированы два файла с именем **xxxxxx_yyyyyy** (xxxxxx – Certificate Serial Number, yyyyyy – Issuer Public Key File Index) и расширением (**sip** и **hip**). Правильность полученных файлов можно проверить с помощью специальных программ, предоставляемых платежной системой.

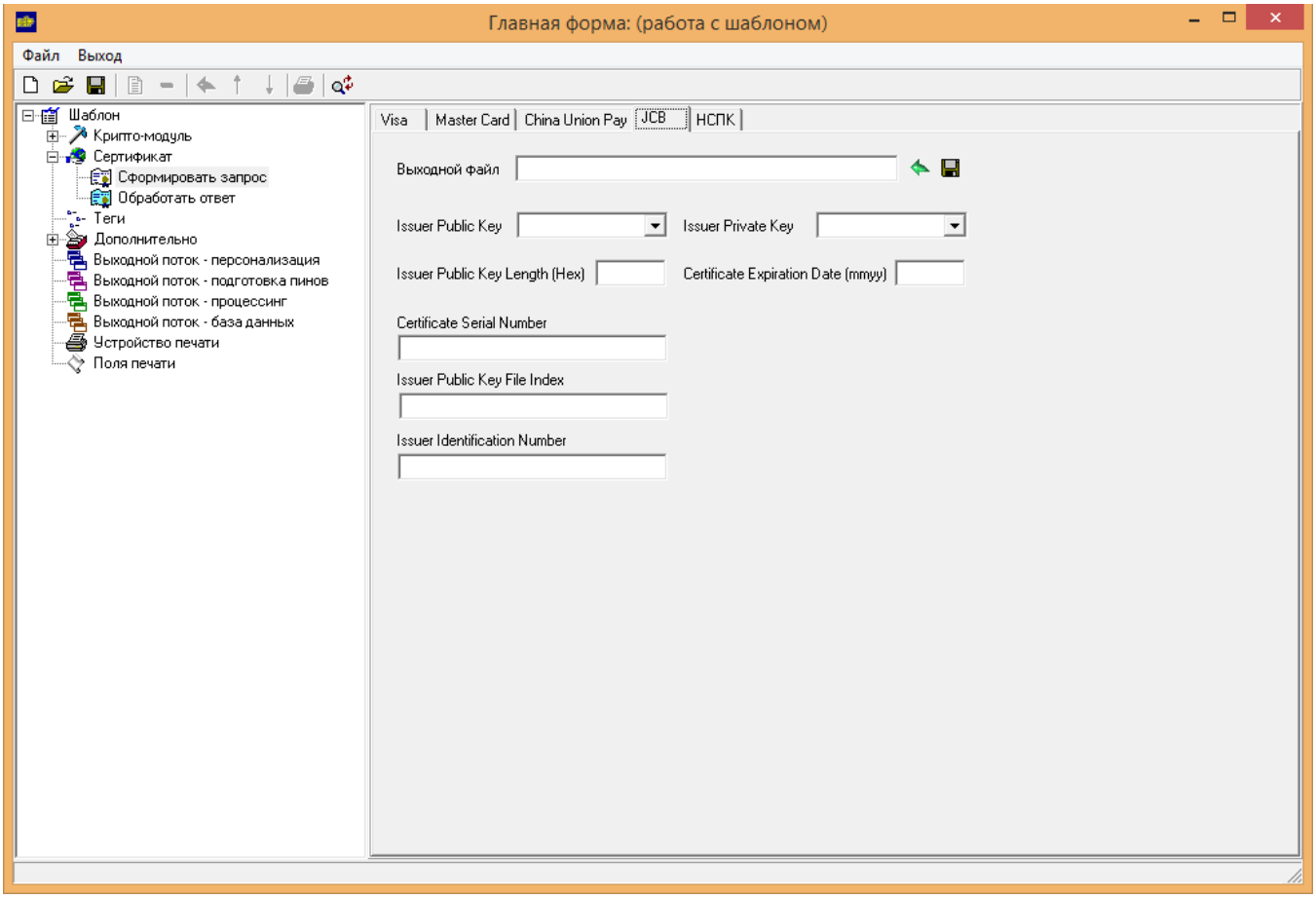
4.2.1.3. Платежная система China UnionPay



Выходной файл – файл, который будет отправлен в платежную систему. Необходимо выбрать директорию, в которую он будет сохранен. Имя файла и расширение сформируется автоматически.
Issuer Public Key – публичный ключ эмитента.
Issuer Private Key – секретный ключ эмитента.
Публичный и секретный ключи имеют тип **RSA**, и должны быть предварительно созданы или импортированы в шаблон.
Issuer Public Key Length (Hex) – длина публичного ключа эмитента в Hex. Если ключ имеет длину 176 байта, соответственно в Hex его длина будет B0.
Certificate Expiration Date (mmyy) – срок действия сертификата.
Tracking Number – 6-значный номер, присылаемый платежной системой.
Service Identifier – 8-значный идентификатор выпускаемого продукта.
Issuer Identification Number – в качестве 6-значного идентификатора используется начало бина, который определяет эмитента и выпускаемый продукт.

После того, как все поля заполнены, необходимо нажать кнопку **Сформировать запрос**. В результате в выбранной директории будет сформирован файл с именем **YLxxxxxx.inp** (xxxxxx - Tracking Number). Правильность полученного файла можно проверить с помощью специальных программ, предоставляемых платежной системой.

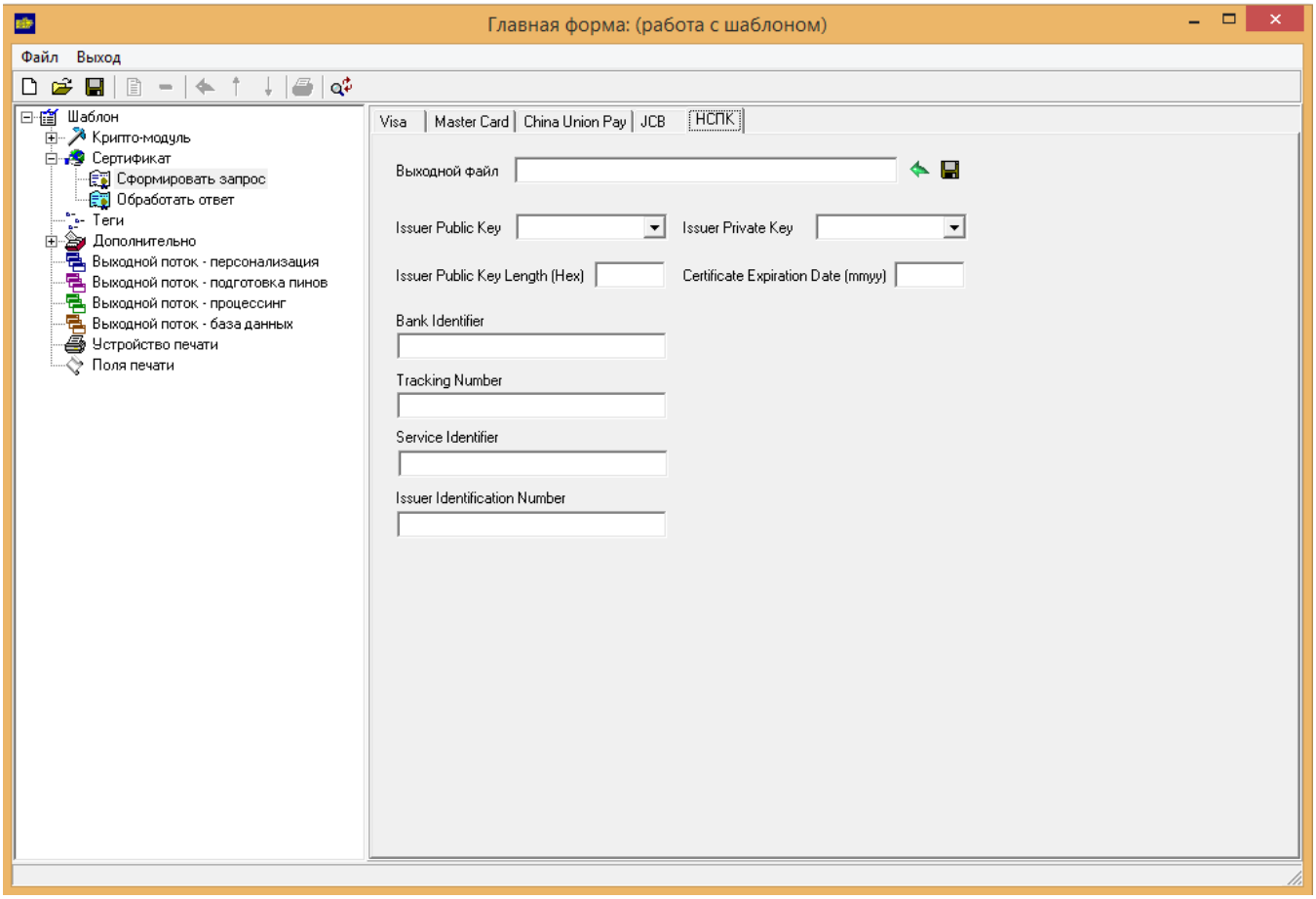
4.2.1.4. Платежная система JCB



Выходной файл – файлы, которые будут отправлены в платежную систему. Необходимо выбрать директорию, в которую они будут сохранены. Имена файлов и расширение сформируется автоматически.
Issuer Public Key – публичный ключ эмитента.
Issuer Private Key – секретный ключ эмитента. Публичный и секретный ключи имеют тип **RSA**, и должны быть предварительно созданы или импортированы в шаблон.
Issuer Public Key Length (Hex) – длина публичного ключа эмитента в Hex. Если ключ имеет длину 176 байта, соответственно в Hex его длина будет B0.
Certificate Expiration Date – срок действия сертификата.
Certificate Serial Number – 6-значный номер, присылаемый платежной системой.
Public Key File Index – 6-значный идентификатор индекса публичного ключа.
Issuer Identification Number – в качестве 8-значного идентификатора используется начало бина, который определяет эмитента и выпускаемый продукт.

После того, как все поля заполнены, необходимо нажать кнопку **Сформировать запрос**. В результате в выбранной директории будет сформирован файл с именем **Kxxxxxxxxx** (xxxxxxxxx –) и без расширения. Правильность полученных файлов можно проверить с помощью специальных программ, предоставляемых платежной системой.

4.2.1.5. Платежная система НСПК



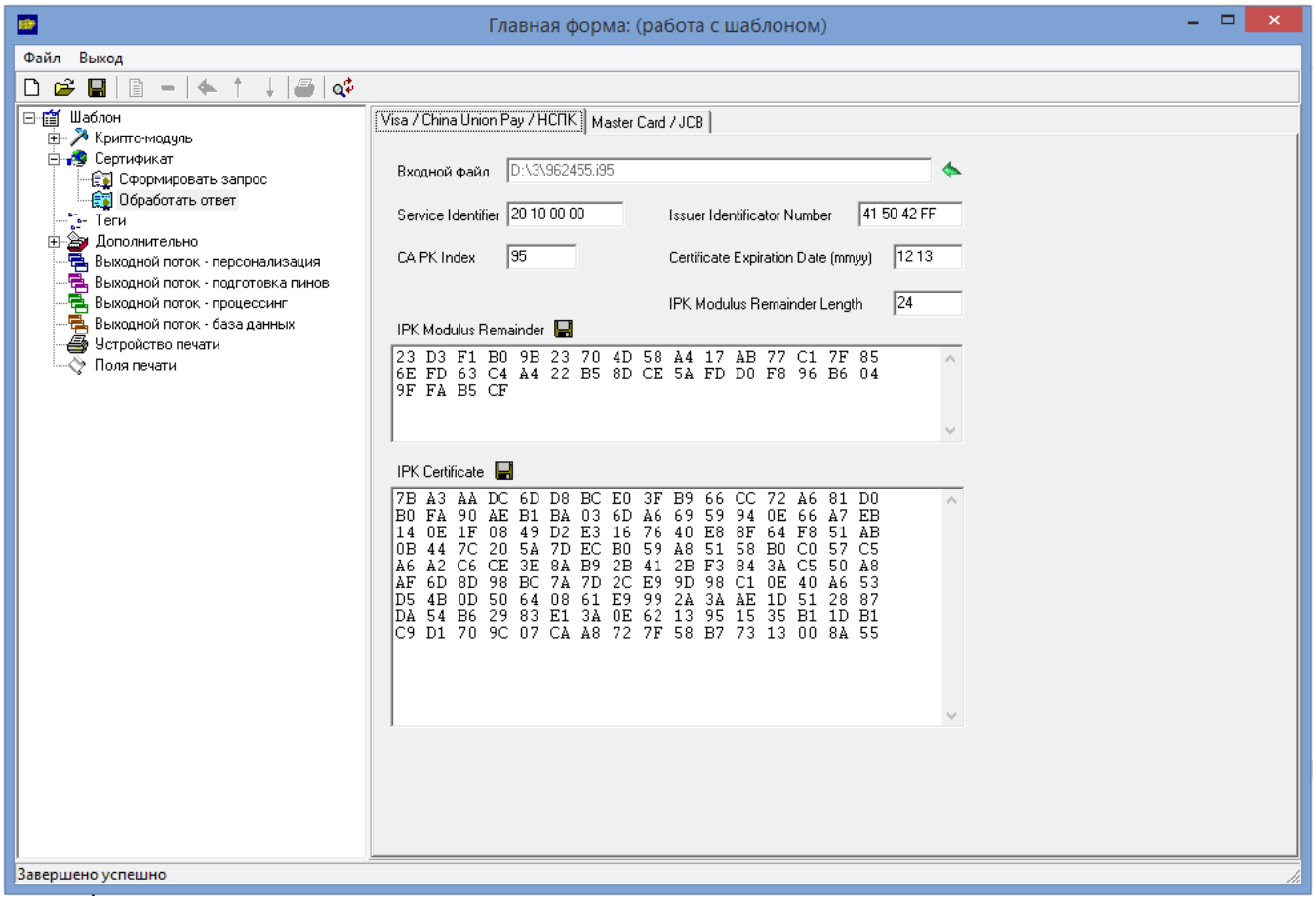
Выходной файл – файл, который будет отправлен в платежную систему. Необходимо выбрать директорию, в которую он будет сохранен. Имя файла и расширение сформируется автоматически.
Issuer Public Key – публичный ключ эмитента.
Issuer Private Key – секретный ключ эмитента.
Публичный и секретный ключи имеют тип **RSA**, и должны быть предварительно созданы или импортированы в шаблон.
Issuer Public Key Length (Hex) – длина публичного ключа эмитента в Hex. Если ключ имеет длину 176 байта, соответственно в Hex его длина будет B0.
Certificate Expiration Date (mmyy) – срок действия сертификата.
Bank Identifier – 4-значный номер, присваиваемый платёжной системой (идентификатор Банка в СЭДО).
Tracking Number – 6-значный номер, присылаемый платежной системой в формате 0xxxxx (номер для запроса тестового сертификата), или 1xxxxx (номер для запроса рабочего (боевого) сертификата), где xxxxx - порядковый номер.
Service Identifier – 4-значный идентификатор выпускаемого продукта.
Issuer Identification Number – в качестве 8-значного идентификатора используется начало бина, который определяет эмитента и выпускаемый продукт.

После того, как все поля заполнены, необходимо нажать кнопку **Сформировать запрос**. В результате в выбранной директории будет сформирован файл с именем **MIR_bbbb_tttttt.REQ** (где bbbb – Bank Identifier, tttttt - Tracking Number). Правильность полученного файла можно проверить с помощью специальных программ, предоставляемых платежной системой.

4.2.2. Обработать ответ

Обрабатывает файл с сертификатом, полученный от платежной системы.

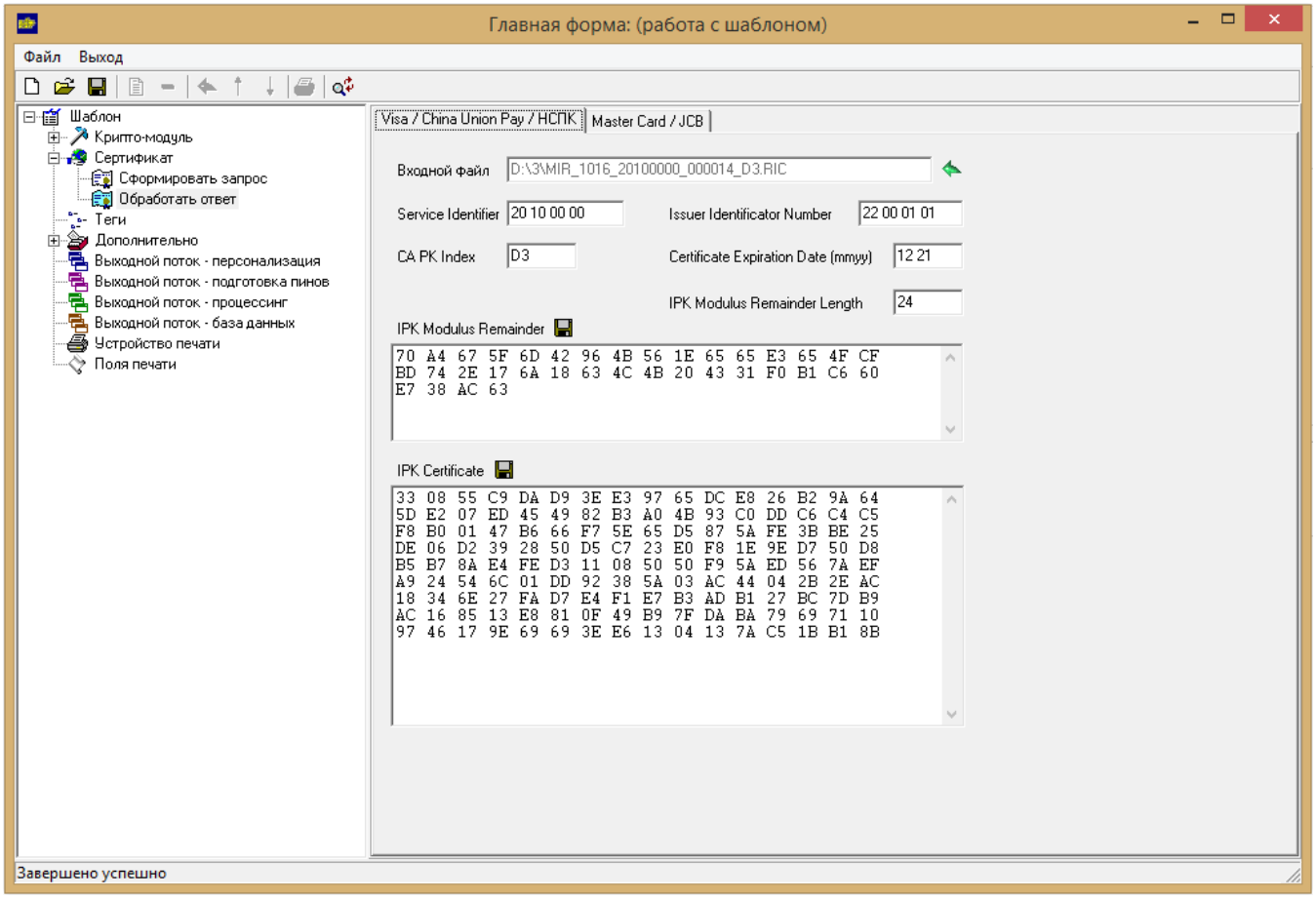
4.2.2.1. Платежная система Visa/China UnionPay



Входной файл – файл, который был получен от платежной системы. Файл имеет расширение **.ixx** (xx – CA PK Index). Необходимо выбрать файл, после чего все остальные поля автоматически заполнятся данными.
Service Identifier – идентификатор выпускаемого продукта.
Issuer Identification Number – начало бина, который определяет эмитента и выпускаемый продукт.
CA PK Index – индекс ключа, на котором платежная система подписывает сертификат.
Certificate Expiration Date – срок действия сертификата.
IPK Modulus Remainder Length – длина в Hex остатка модуля публичного ключа эмитента.
IPK Modulus Remainder – остаток модуля публичного ключа эмитента.
IPK Certificate – сертификат публичного ключа эмитента.

Сертификат и остаток публичного ключа можно сохранить в файл, для дальнейшего использования. Для этого необходимо нажать кнопку **Сохранить**, рядом с соответствующим полем.

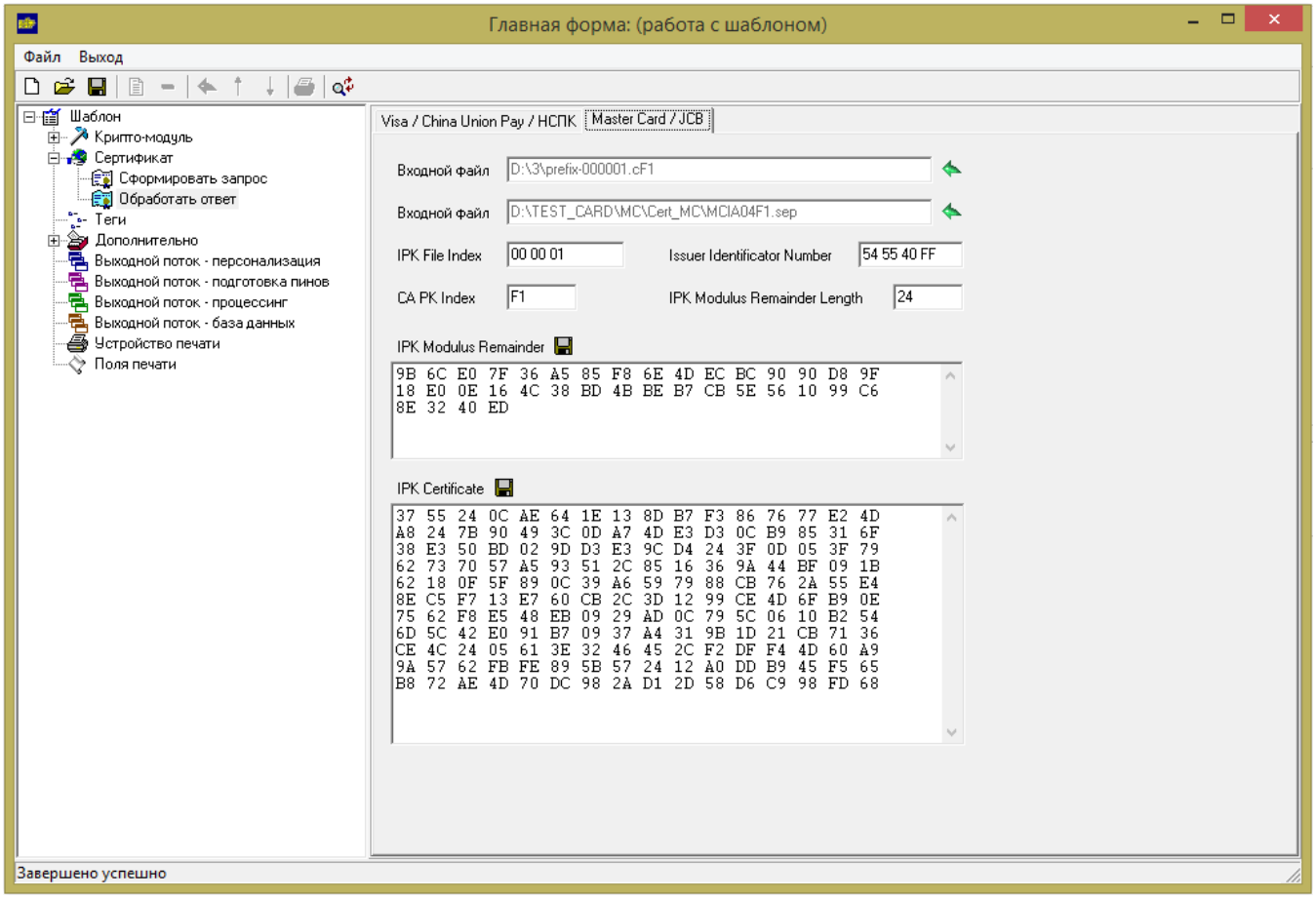
4.2.2.2. Платежная система НСПК



Входной файл – файл, который был получен от платежной системы. Файл имеет расширение **.RIC**. Необходимо выбрать файл, после чего все остальные поля автоматически заполнятся данными.
Service Identifier – идентификатор выпускаемого продукта.
Issuer Identification Number – начало бина, который определяет эмитента и выпускаемый продукт.
CA PK Index – индекс ключа, на котором платежная система подписывает сертификат.
Certificate Expiration Date – срок действия сертификата.
IPK Modulus Remainder Length – длина в Hex остатка модуля публичного ключа эмитента.
IPK Modulus Remainder – остаток модуля публичного ключа эмитента.
IPK Certificate – сертификат публичного ключа эмитента.

Сертификат и остаток публичного ключа можно сохранить в файл, для дальнейшего использования. Для этого необходимо нажать кнопку **Сохранить**, рядом с соответствующим полем.

4.2.2.3. Платежная система Master Card



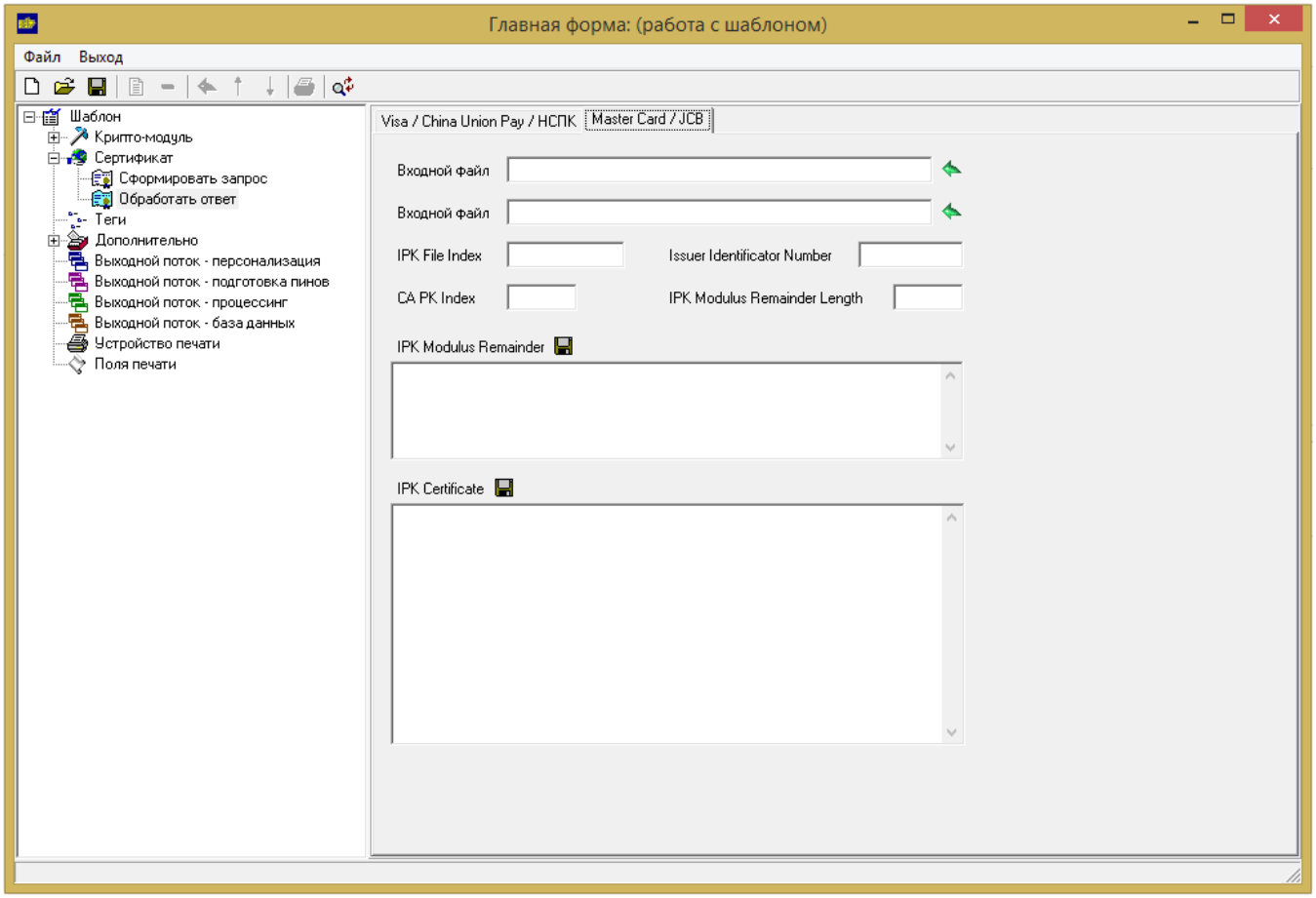
Входной файл – файл, который был получен от платежной системы (содержит сертификат эмитента). Файл имеет расширение **.cxx** (xx – CA PK Index).
Входной файл – файл, который был получен от платежной системы (содержит данные о ключе, которым платежная система подписывала сертификат). Файл имеет расширение **.sep**

Необходимо выбрать оба файла, после чего все остальные поля автоматически заполнятся данными.

IPK File Index – идентификатор индекса публичного ключа.
Issuer Identification Number – начало бина, который определяет эмитента и выпускаемый продукт.
CA PK Index – индекс ключа, на котором платежная система подписывает сертификат.
IPK Modulus Remainder Length – длина в Hex остатка модуля публичного ключа эмитента.
IPK Modulus Remainder – остаток модуля публичного ключа эмитента.
IPK Certificate – сертификат публичного ключа эмитента.

Сертификат и остаток публичного ключа можно сохранить в файл, для дальнейшего использования. Для этого необходимо нажать кнопку **Сохранить**, рядом с соответствующим полем.

4.2.2.4. Платежная система JCB



Входной файл – файл, который был получен от платежной системы (содержит сертификат эмитента). Файл имеет имя, начинающееся с символа L и не имеет расширения.
Входной файл – файл, который был получен от платежной системы (содержит данные о ключе, которым платежная система подписывала сертификат). Файл имеет расширение **.cpk.**

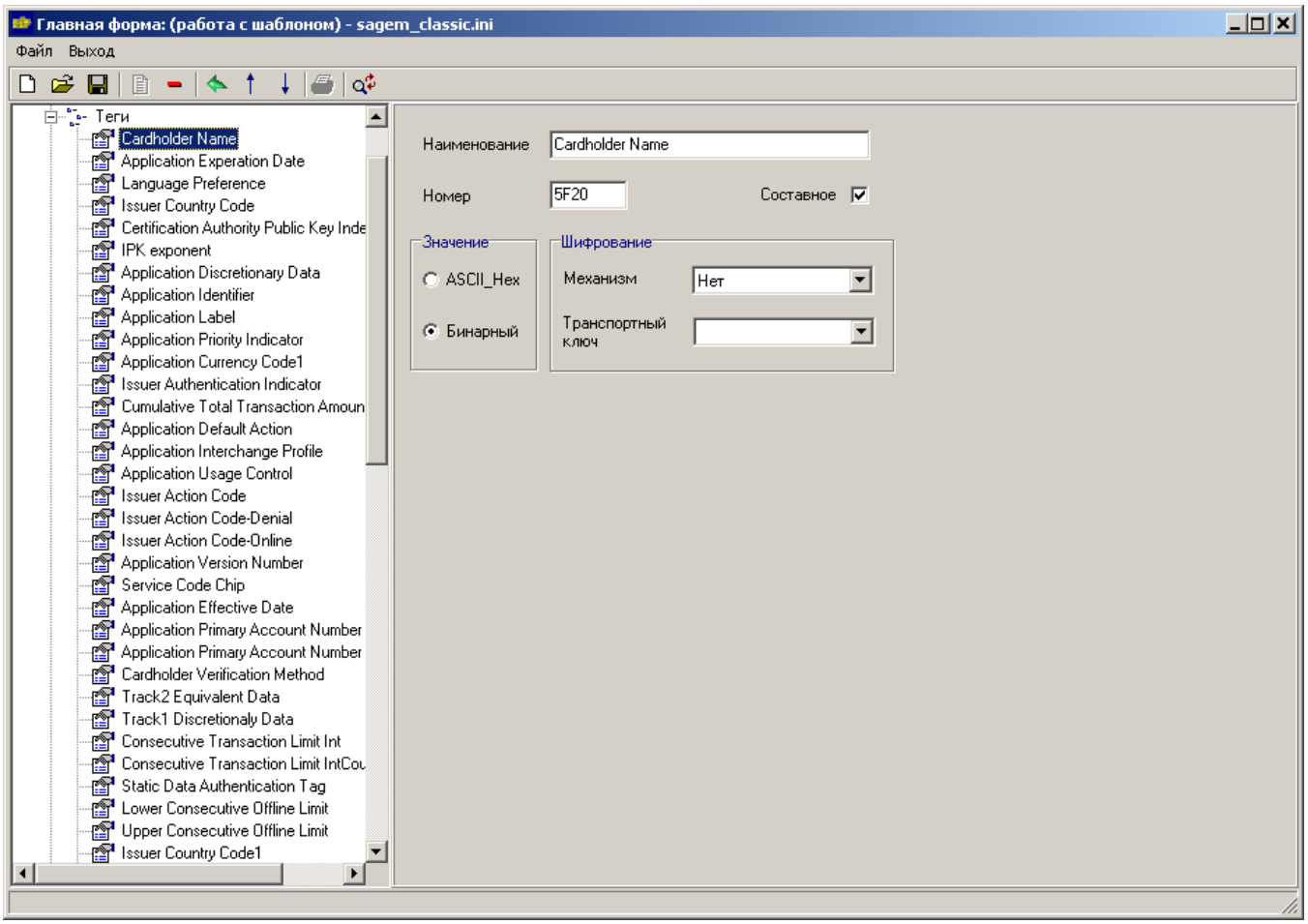
Необходимо выбрать оба файла, после чего все остальные поля автоматически заполнятся данными.

IPK File Index – идентификатор индекса публичного ключа.
Issuer Identification Number – начало бина, который определяет эмитента и выпускаемый продукт.
CA PK Index – индекс ключа, на котором платежная система подписывает сертификат.
IPK Modulus Remainder Length – длина в Hex остатка модуля публичного ключа эмитента.
IPK Modulus Remainder – остаток модуля публичного ключа эмитента.
IPK Certificate – сертификат публичного ключа эмитента.

Сертификат и остаток публичного ключа можно сохранить в файл, для дальнейшего использования. Для этого необходимо нажать кнопку **Сохранить**, рядом с соответствующим полем.

4.3. Теги (TAG)

Теги являются основным элементом, используемым для подготовки данных. Каждый объект данных определяется тремя полями: тегом (Tag), длиной (Length) и значением (Value) – формат TLV. Теги могут быть как определенные стандартом EMV (Europay MasterCard Visa) для данных, которые записываются на карту, так и произвольными, т.е. определенными пользователем.



Наименование – наименование тега.
Номер – номер тега.
Составное – значение будет браться из составного поля, которое определяется в проекте.
Значение – формат входного значения. Поддерживаемые форматы:

ASCII_Hex – символы переведенные в HEX;

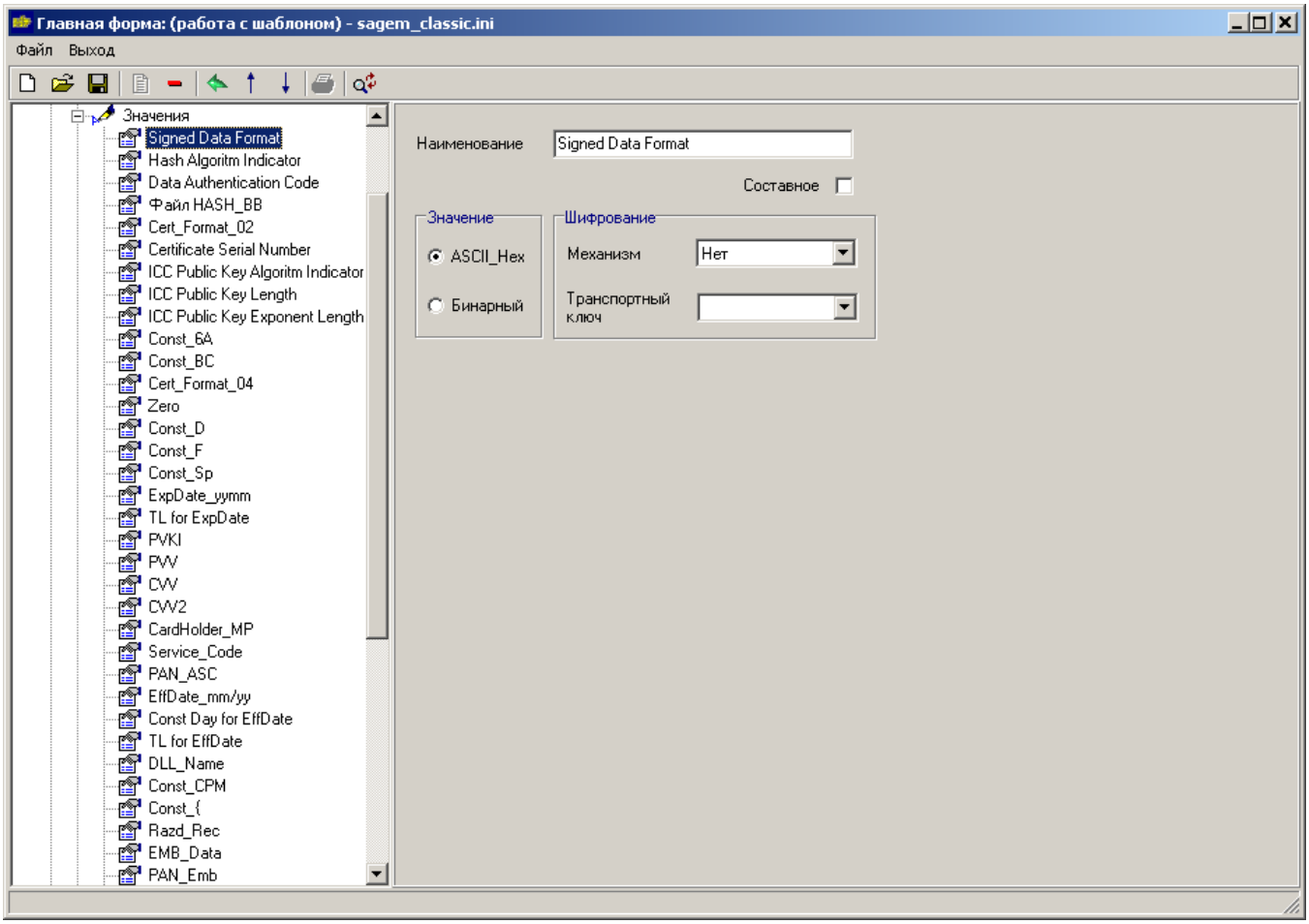
Бинарный – символы;

Поле ***Значение***, если оно доступно в других местах, используется аналогично.
Шифрование – описано выше в пункте 4.1.5.3.

4.4. Дополнительно

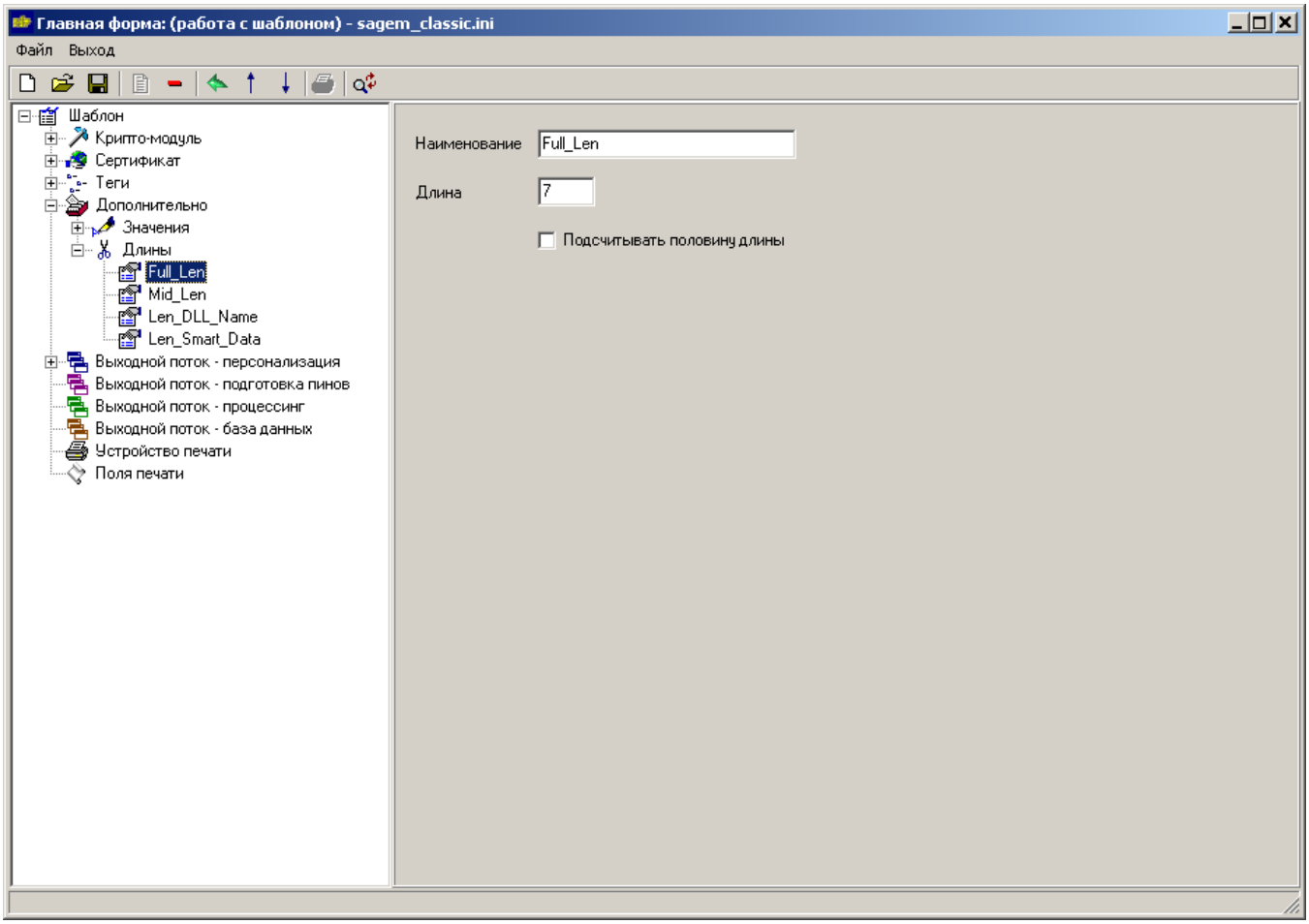
4.4.1. Значения

Если необходимо использовать только значение какого-либо объекта данных (без длины и номера тега), то используется данный элемент структуры меню шаблона. Обычно он используется для создания элементов/констант, участвующих в составных полях и выходных потоках, а также разделителей, участвующих в выходных потоках.
Свойства элемента аналогичны свойствам, используемым при описание тегов (см. пункт 4.3).



4.4.2. Длины

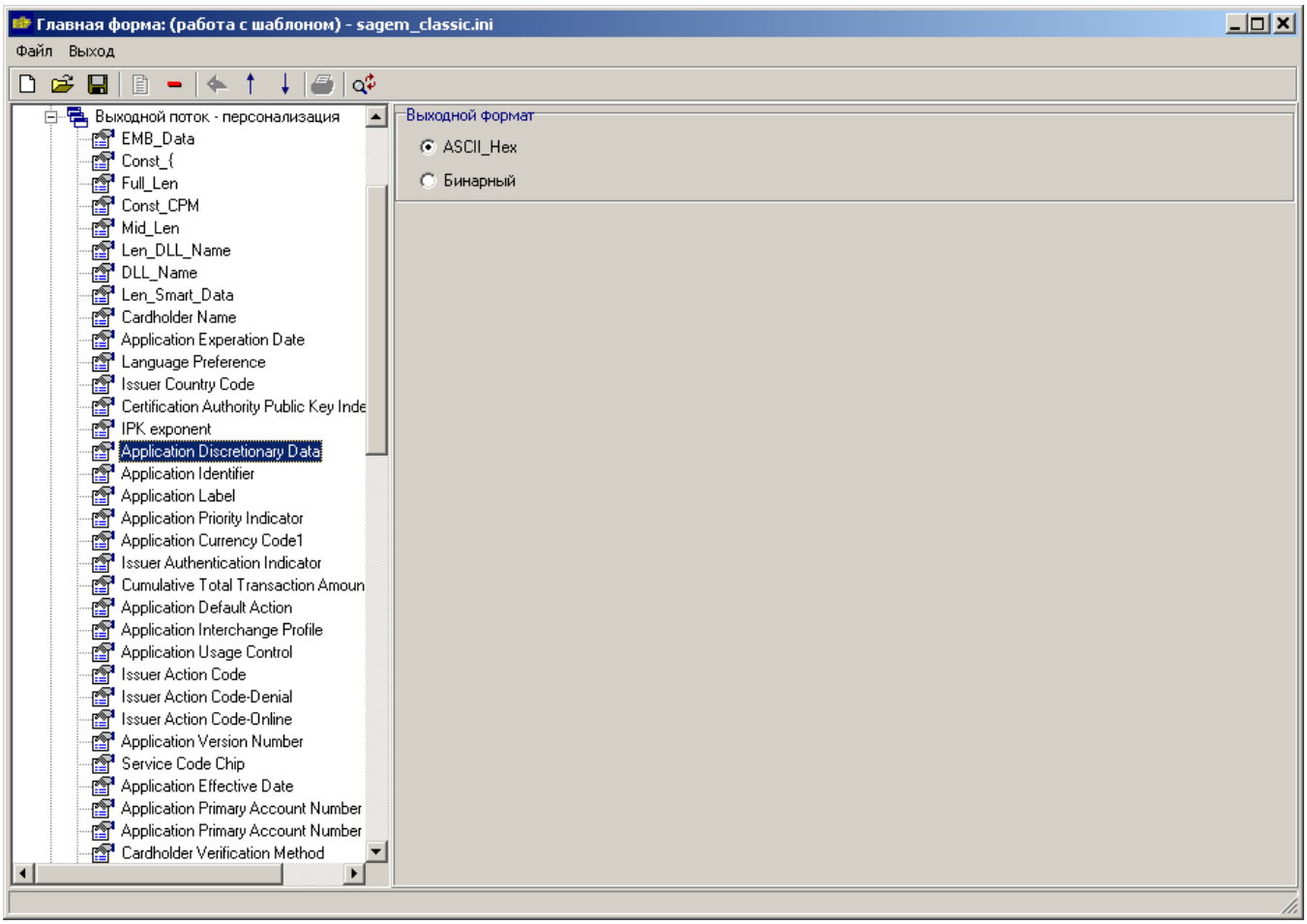
Данный элемент используется, если необходимо подсчитать суммарную длину, каких-либо элементов из выходного потока. Элементы, которые будут использованы в подсчете, выбираются в выходном потоке.



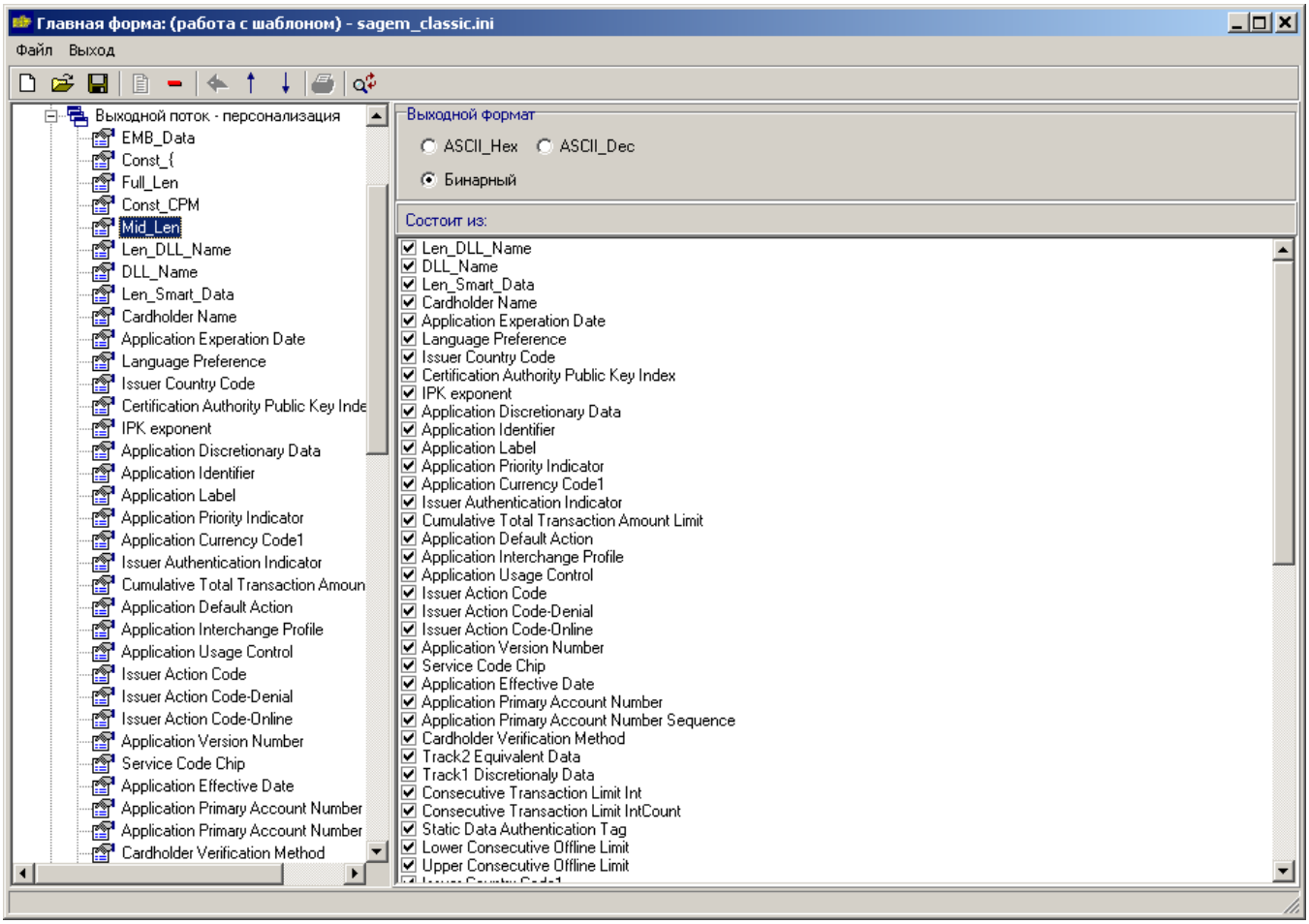
Наименование – наименование длины.
Длина – определяет размер в байтах, который будет использован в выходном потоке.
Подсчитывать половину длины – подсчитывает половину от реальной длины данных. Иногда используется для подсчета длины в тегах.

4.5. Выходной поток - персонализация

В выходной поток добавляются элементы, которые будут использоваться непосредственно при выводе данных. Данные будут выводиться в порядке, определенном в выходном потоке для персонализации.



Если в выходном потоке используется элемент типа длина, то необходимо отметить длины каких элементов будут подсчитываться. Можно отметить только элементы, расположенные ниже в выходном потоке.



Выходной формат – определяет в каком формате будут выводиться данные:

ASCII_Hex – символы переведенные в HEX;

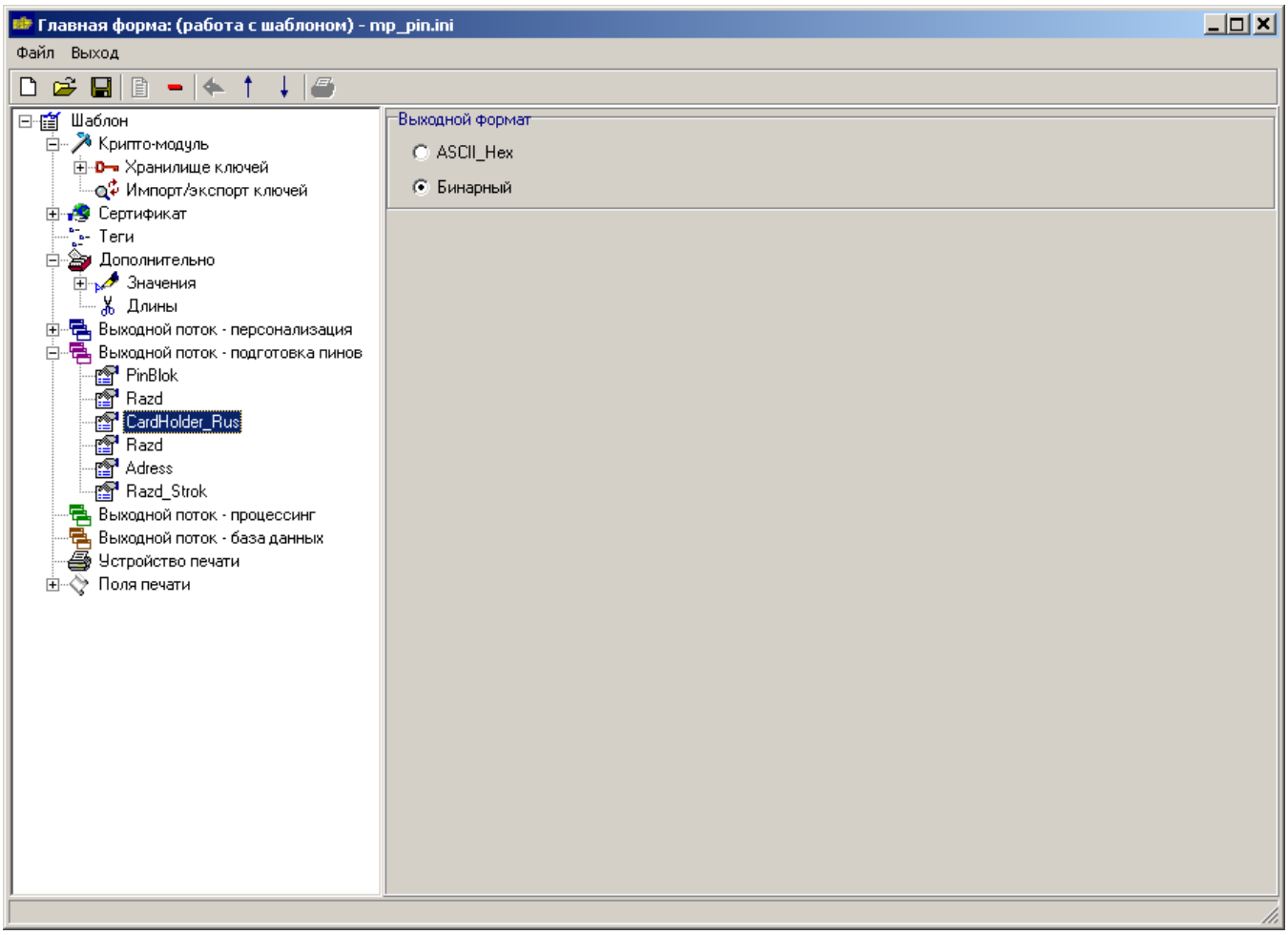
Бинарный – символы;

ASCII_DEC – символы переведенные в DEC, используется только для длин;

В результате получится выходной файл, данные из которого будут использоваться для персонализации карт.

4.6. Выходной поток – подготовка ПИНов

В выходной поток добавляются элементы, которые будут использоваться непосредственно при выводе данных. Данные будут выводиться в порядке, определенном в выходном потоке для подготовки ПИНов.



Выходной формат – определяет в каком формате будут выводиться данные:

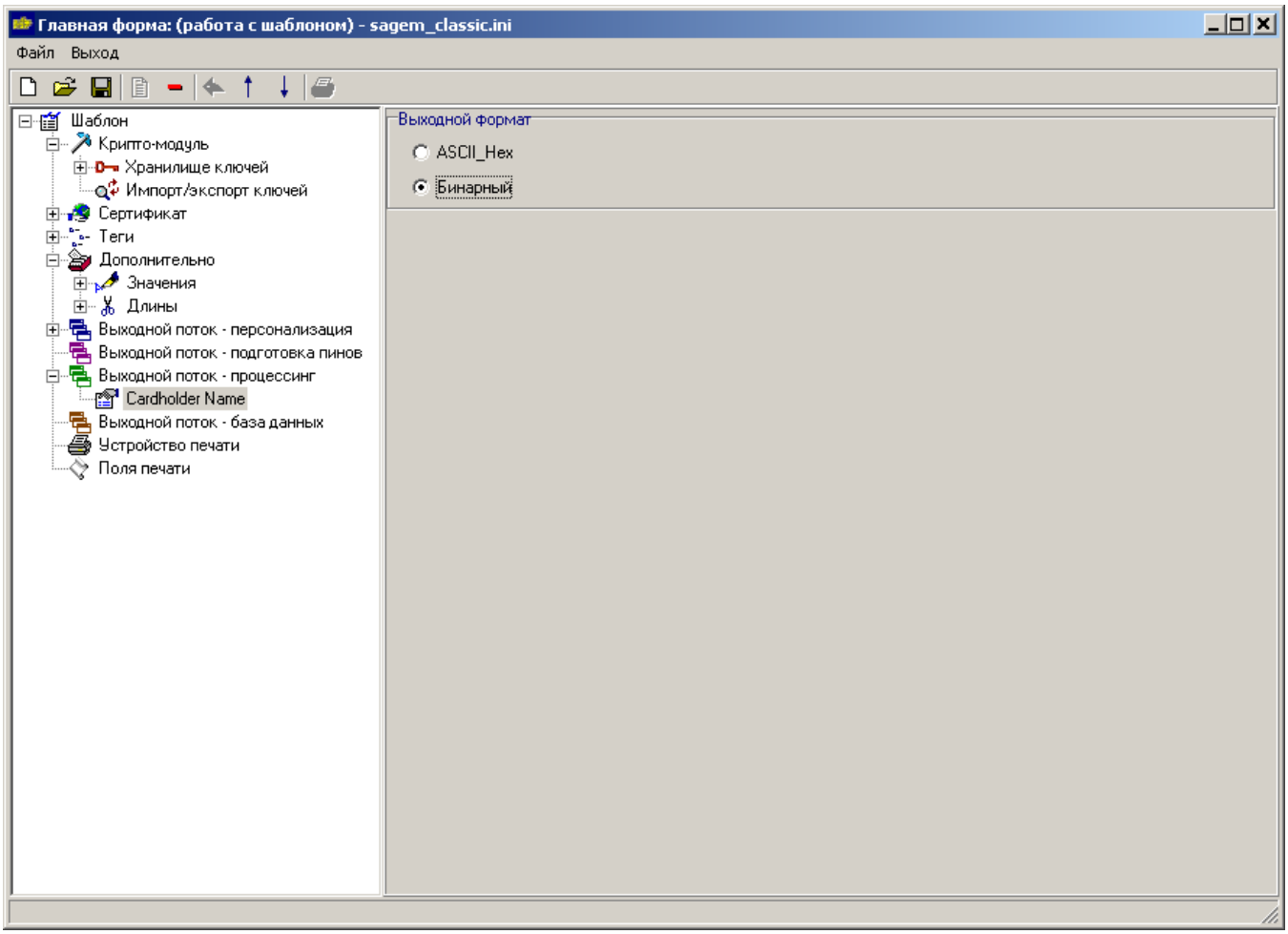
ASCII_Hex – символы переведенные в HEX;

Бинарный – символы;

В результате получится выходной файл, данные из которого будут использоваться для печати данных на ПИН-конвертах*.

4.7. Выходной поток – процессинг

В выходной поток добавляются элементы, которые будут использоваться непосредственно при выводе данных. Данные будут выводиться в порядке, определенном в выходном потоке процессинга.



Выходной формат – определяет в каком формате будут выводиться данные:

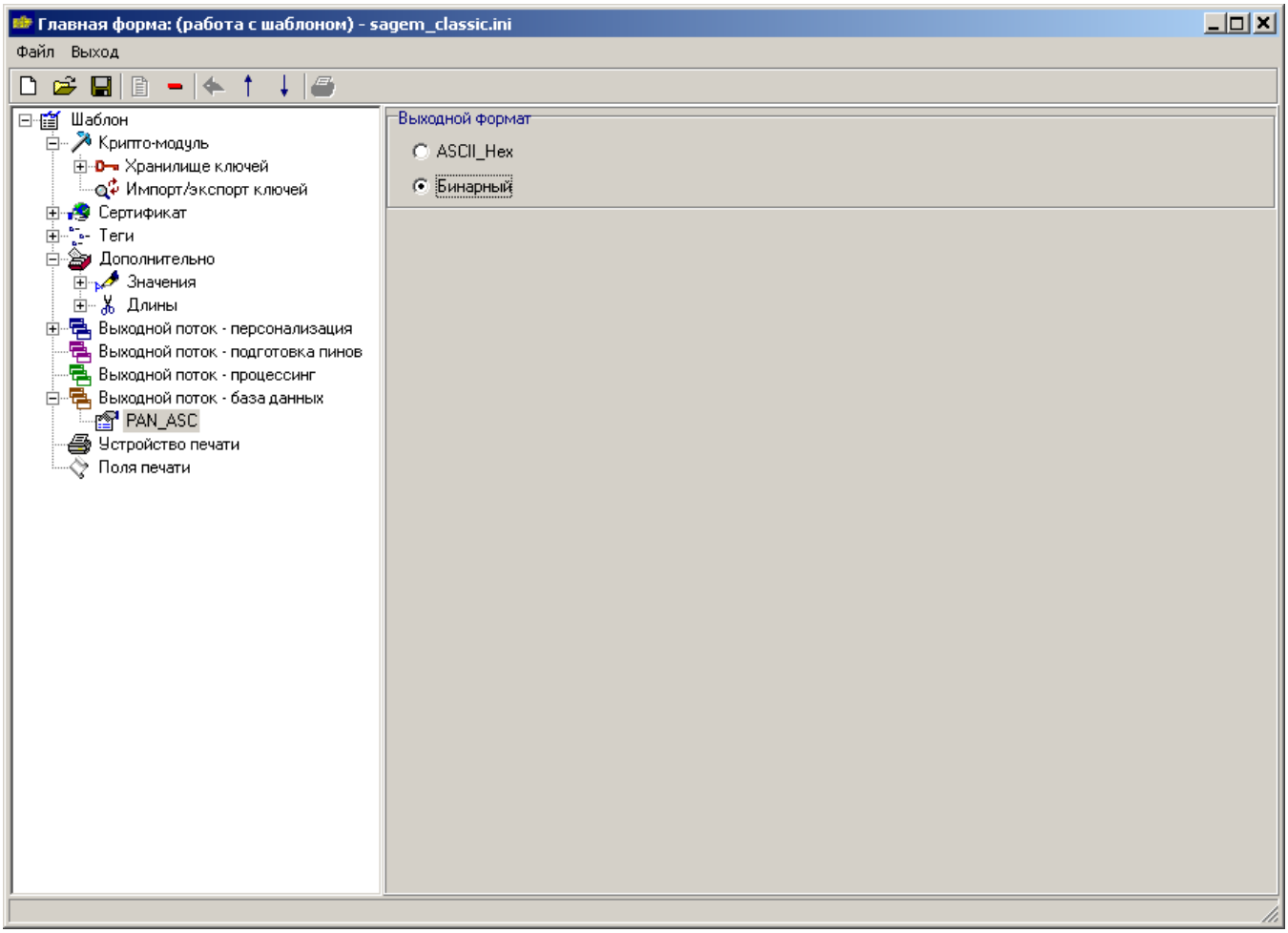
ASCII_Hex – символы переведенные в HEX;

Бинарный – символы;

В результате получится выходной файл, который может быть отправлен в процессинг.

4.8. Выходной поток – база данных

В выходной поток добавляются элементы, которые будут использоваться для записи в выходную базу данных.



Выходной формат – определяет в каком формате данные будут записываться в базу данных:

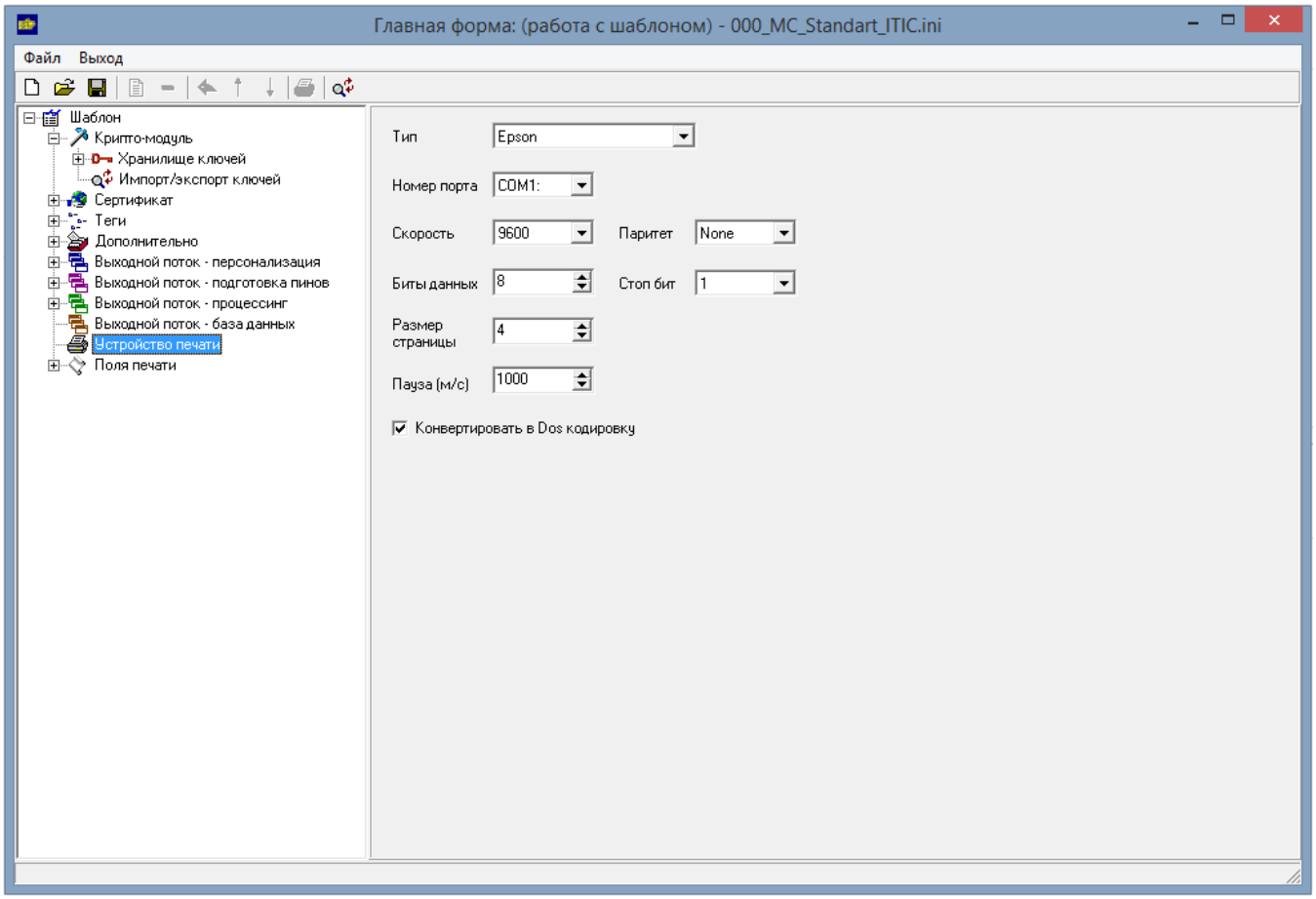
ASCII_Hex – символы переведенные в HEX;

Бинарный – символы;

В результате данные будут записаны в выходную базу данных, которая настраивается в проекте. Данная база данных может в частности использоваться для перевыпуска карт.

4.9. Устройство печати

В данном узле шаблона настраивается устройство, через которое будут печататься данные на ПИН-конвертах.

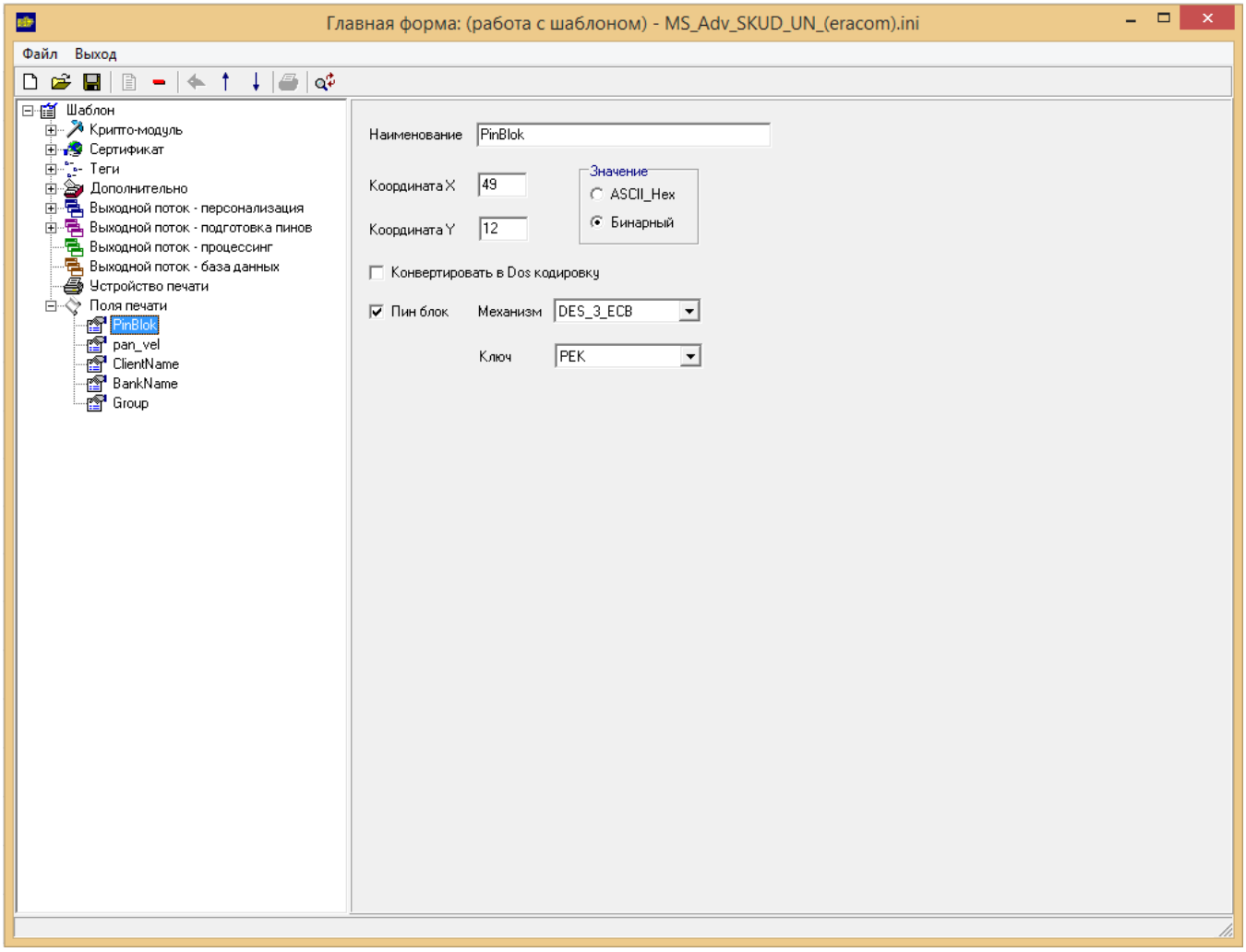


Тип устройства – тип печатающего устройства.
Номер порта, скорость, паритет, биты данных, стоп бит – параметры печатающего устройства.
Размер страницы – длина страницы в дюймах (стандартный размер ПИН-конверта – 4 дюйма).
Пауза (м/с) – интервал времени между посылками на печать ПИН-конвертов данных двух соседних записей входного файла. Используется для исключения переполнения буферной памяти принтера при выводе на печать больших объемов данных.
Конвертировать в Dos кодировку – конвертация кириллицы (в случае её использовании) из Windows-кодировки в Dos-кодировку в случае отсутствия возможности принтера печатать кириллицу в Windows-кодировке.

4.10. Поля печати

После настройки печатающего устройства, необходимо добавить поля и настроить их расположение на бумаге. Настройка полей печати определяется видом ПИН-конверта и зависит от расположения защитных слоев.

1) В случае использования крипто-модуля типа Eracom.



Наименование – наименование поля печати.
Координата X – расположение поля по X. Значение определяет отступ в символах от левого края (по умолчанию 1).
Координата Y – расположение поля по Y. Значение определяет строку, на которой будут печататься данные (по умолчанию 1).
Значение – формат входного значения. Поддерживаемые форматы:

ASCII_Hex – символы переведенные в HEX;
Бинарный – символы;

Конвертировать в Dos кодировку – конвертация кириллицы (в случае её использовании) из Windows-кодировки в Dos-кодировку в случае отсутствия возможности принтера печатать кириллицу в Windows-кодировке.
ПИН-блок – используется для печати ПИНов (признак того что в данном поле находится ПИН-блок).

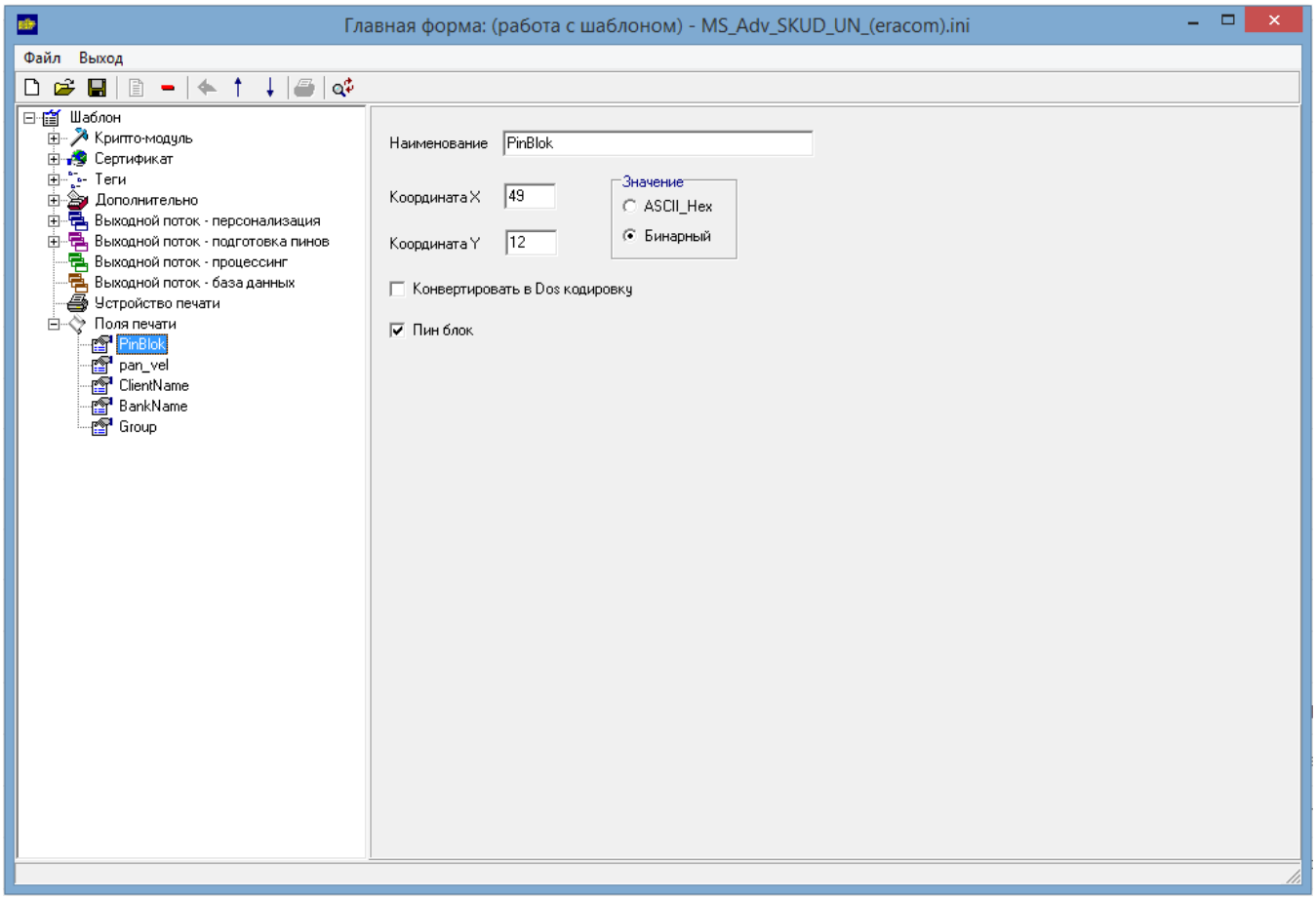
В случае использования в качестве крипто-модуля **Eracom**, должны быть выполнены следующие настройки:

Так как в процессе печати будет происходить расшифровка входного значения, то необходимо указать, что данные в этом поле являются ПИНом, и настроить параметры шифрования.

Входное значение поля, содержащего данные для вывода на печать ПИНа, должно быть сформировано следующим образом: значение, которое вернула функция генерации ПИНа, переведенное в HEX, плюс 12 цифр PAN, заканчивая предпоследней цифрой PAN (например, если PAN имеет длину 16 цифр, то используются цифры PAN с позиции 4 по 15 включительно).
Механизм – механизм, который использовался при шифровании ПИН.
Ключ – ключ, который использовался при генерации ПИН.

Среди полей, предназначенных для вывода на печать, обязательно должно быть одно поле, помеченное как ***ПИН-блок***.

2) В случае использования крипто-модуля типа Thales.



Наименование – наименование поля печати.
Координата X – расположение поля по X. Значение определяет отступ в символах от левого края (по умолчанию 1).
Координата Y – расположение поля по Y. Значение определяет строку, на которой будут печататься данные (по умолчанию 1).
Значение – формат входного значения. Поддерживаемые форматы:

ASCII_Hex – символы переведенные в HEX;

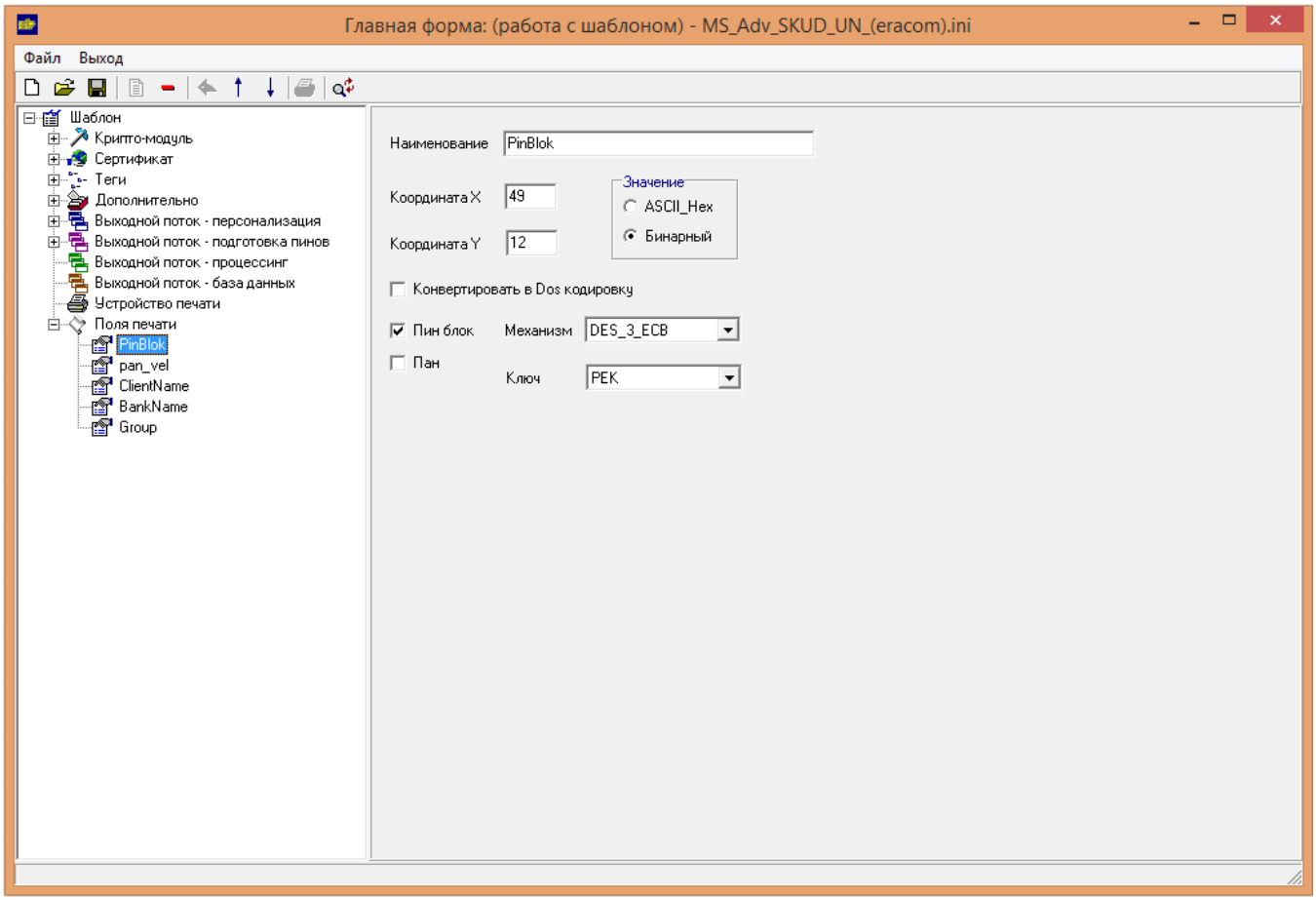
Бинарный – символы;

Конвертировать в Dos кодировку – конвертация кириллицы (при её использовании) из Windows-кодировки в Dos-кодировку в случае отсутствия возможности принтера печатать кириллицу в Windows-кодировке.

ПИН-блок – используется для печати ПИНов (признак того что в данном поле находится ПИН-блок).

Среди полей, предназначенных для вывода на печать, обязательно должно быть одно поле, помеченное как ***ПИН-блок***.

3) В случае использования крипто-модуля типа Eracom_EFT.



Наименование – наименование поля печати.
Координата X – расположение поля по X. Значение определяет отступ в символах от левого края (по умолчанию 1).
Координата Y – расположение поля по Y. Значение определяет строку, на которой будут печататься данные (по умолчанию 1).
Значение – формат входного значения. Поддерживаемые форматы:

ASCII_Hex – символы переведенные в HEX;

Бинарный – символы;

Конвертировать в Dos кодировку – конвертация кириллицы (при её использовании) из Windows-кодировки в Dos-кодировку в случае отсутствия возможности принтера печатать кириллицу в Windows-кодировке.
ПИН-блок – используется для печати ПИНов (признак того что в данном поле находится ПИН-блок).
Пан – .

В случае использования в качестве крипто-модуля **Eracom_EFT**, должны быть выполнены следующие настройки:

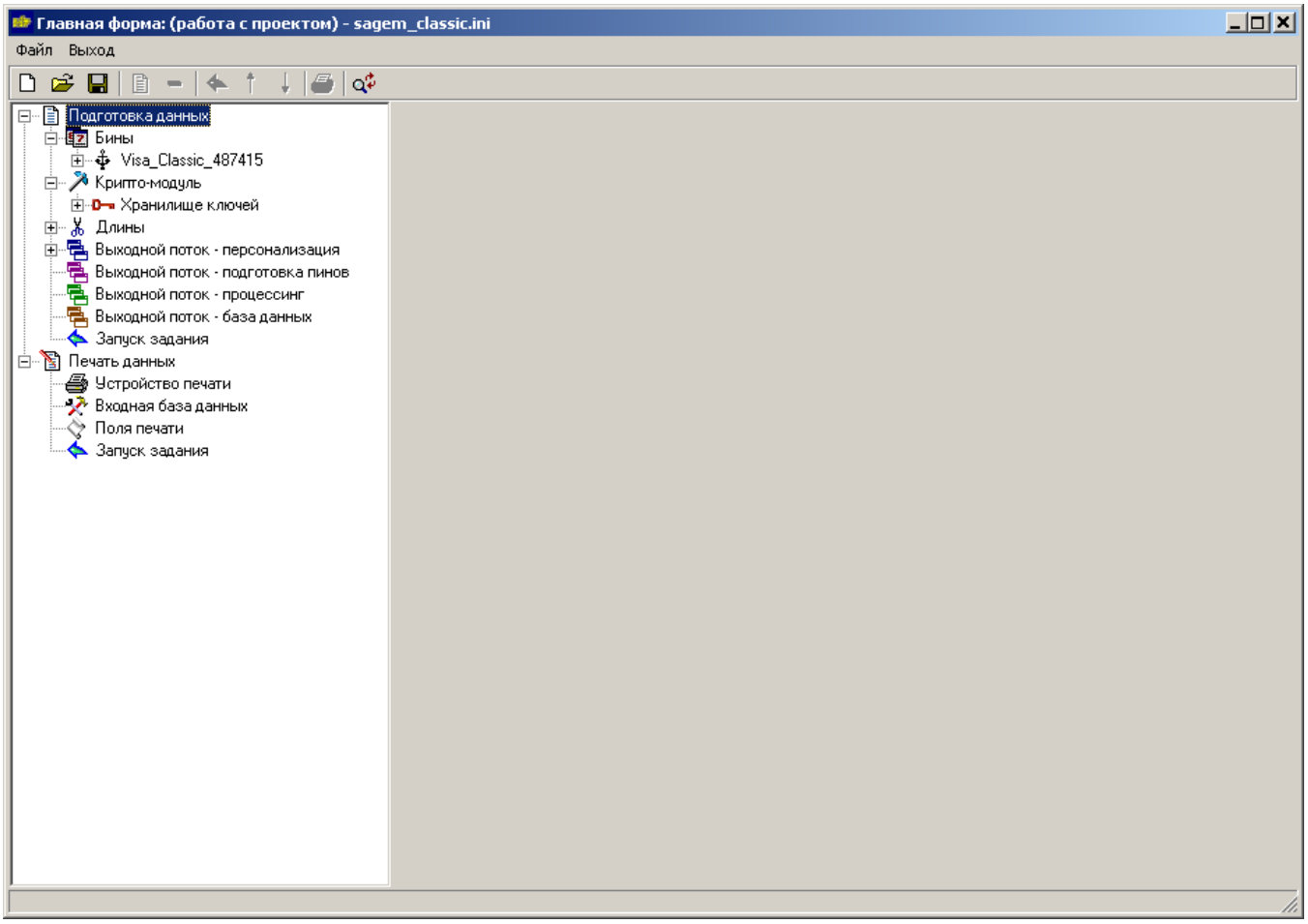
Так как в процессе печати будет происходить расшифровка входного значения, то необходимо указать, что данные в этом поле являются ПИНом, и настроить параметры шифрования.

Входное значение поля, содержащего данные для вывода на печать ПИНа, должно быть сформировано следующим образом: значение, которое вернула функция генерации ПИНа, переведенное в HEX плюс 12 цифр PAN, заканчивая предпоследней цифрой PAN (если PAN имеет длину 16 цифр, то используются цифры PAN с позиции 4 по 15 включительно).
Механизм – механизм, который использовался при шифровании ПИН.
Ключ – ключ, который использовался при генерации ПИН.

Среди полей, предназначенных для вывода на печать, обязательно должно быть одно поле, помеченное как ***ПИН-блок***.

5. Работа с проектом

В качестве исходных настроек для проекта, используются настройки, описанные в шаблоне. После открытия проекта в левой части экрана в виде дерева будут расположены основные модули, необходимые для настройки проекта. В правой части показываются свойства данных модулей. Проект хранится в виде ini-файла, в котором хранятся настройки шаблона и настройки, меняющиеся при подготовке данных (источники данных, входные значения и.т.д.)

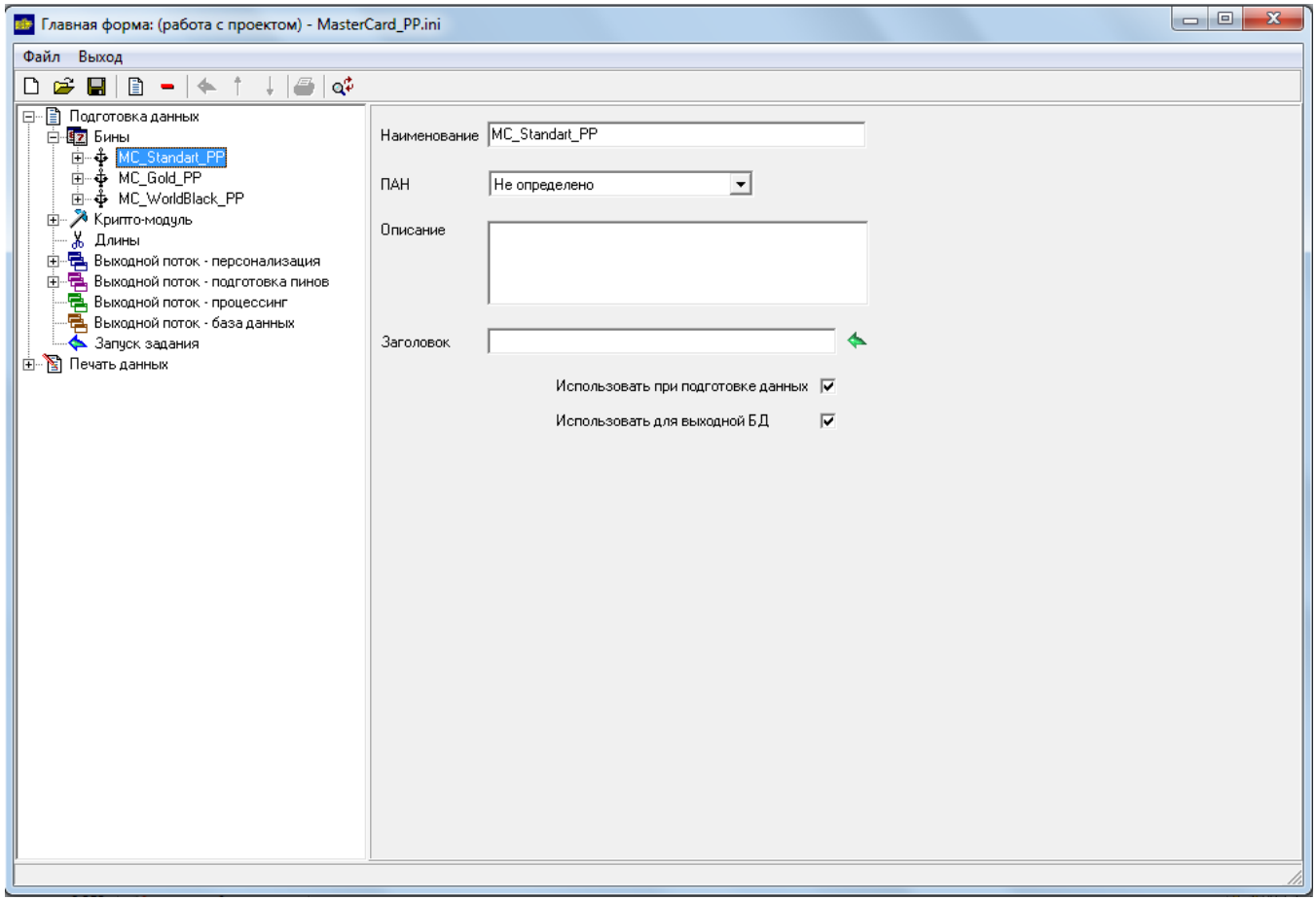


Каждый проект разделяется на два основных подраздела: подготовка данных и печать данных.

5.1. Подготовка данных

5.1.1. Бины

Модуль ***Бины*** обычно определяет банк и выпускаемый продукт. В проекте одновременно могут использоваться несколько бинов, которые отличаются друг от друга некими константами (уникальными для конкретного бина), ключами, а также входными значениями для подготовки данных. Добавление и удаление бинов осуществляется из главного меню. В проекте обязательно должен быть один бин.



Наименование – наименование бина.
ПАН – (данный параметр в текущей версии не используется).
Описание – краткое описание бина.
Заголовок – используется, если перед данными для бина необходимо вставить какую-либо константу. Значение константы находится в файле, который можно выбрать используя кнопку **Выбрать файл**.
Использовать при подготовке данных – используется, когда существует несколько бинов, а подготовить данные нужно для какого-то конкретного бина.
Использовать для выходной БД – используется, когда подготавливаемые данные требуется сохранить в базе данных.

Настройки узлов ***Крипто-модуль***, ***Длины*** и ***Выходной поток*** постоянны и одинаковы для всех Бинов, созданных в Проекте, и описываются в Шаблоне. В Проекте их свойства доступны только для просмотра.

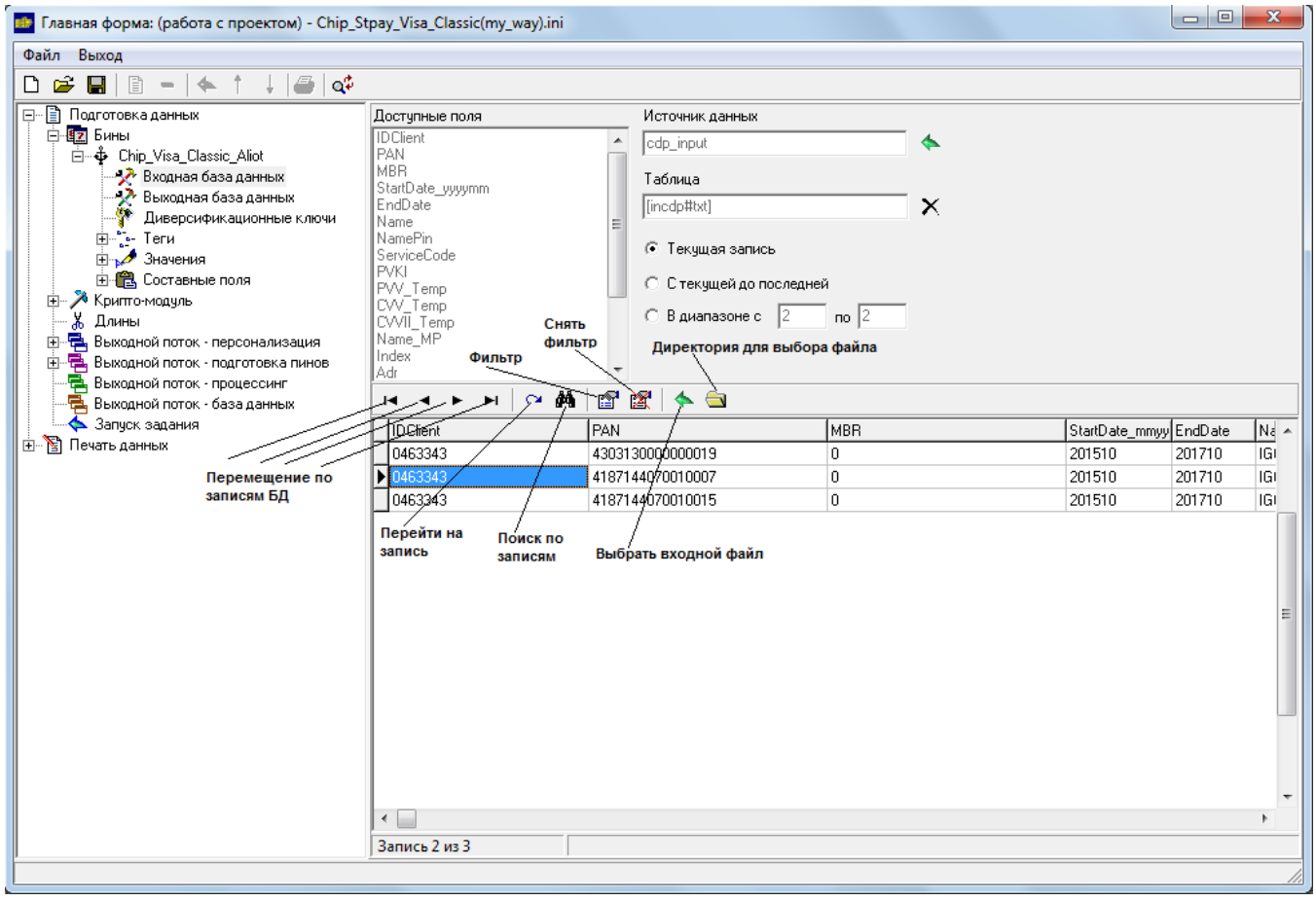
Рассмотрим основные узлы проекта (для одного бина):

5.1.2. Входная база данных

Если для подготовки данных требуются значения, хранящиеся в базе данных, то необходимо произвести следующие действия. В «Панели управления» компьютером выбрать «Администрирование» и настроить «Источник данных (ODBC)» на требуемую базу. Текущая версия программы CDP поддерживает работу со следующими ODBC - драйверами:

- Microsoft Access Driver
- Microsoft dBase Driver
- Microsoft Excel Driver
- Microsoft Text Driver (наиболее часто используемый)
- SQL Server
- Microsoft ODBC for Oracle

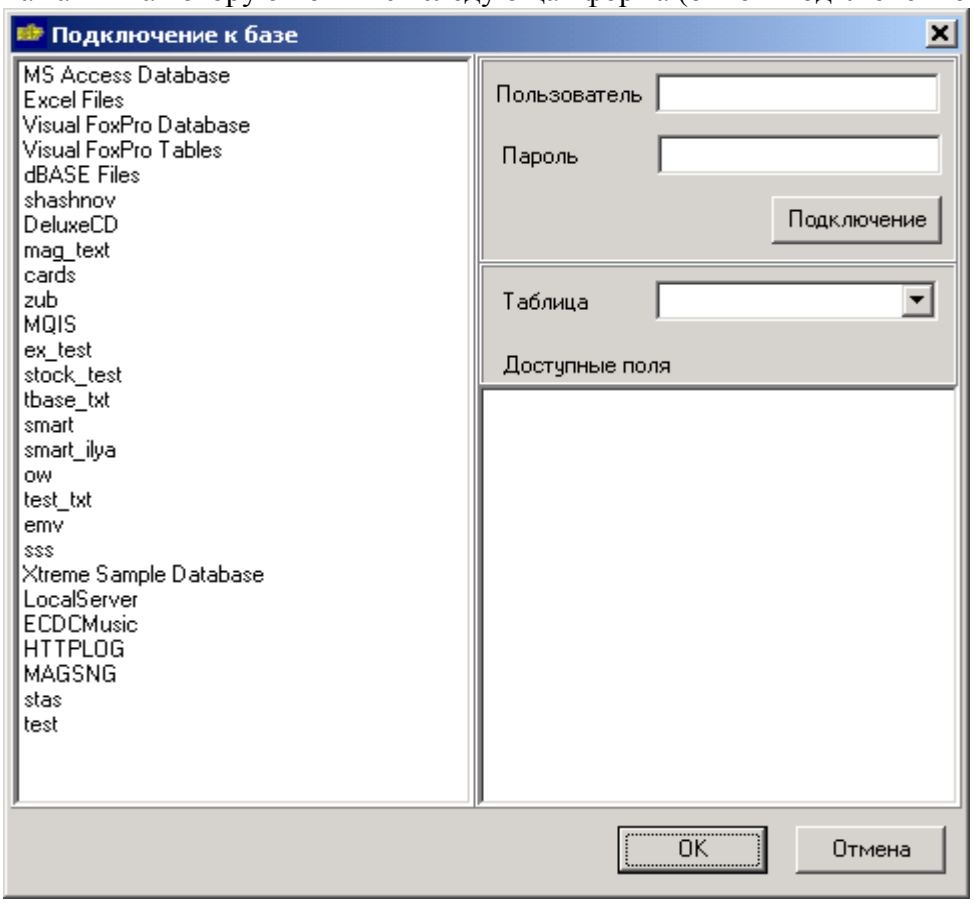
Создание применяемых источников ODBC и описание формата таблиц входных данных, соответствующих определённому источнику ODBC описано в Приложении №1 к настоящему Руководству.



Параметры интерфейса при работе с входной базой данных:

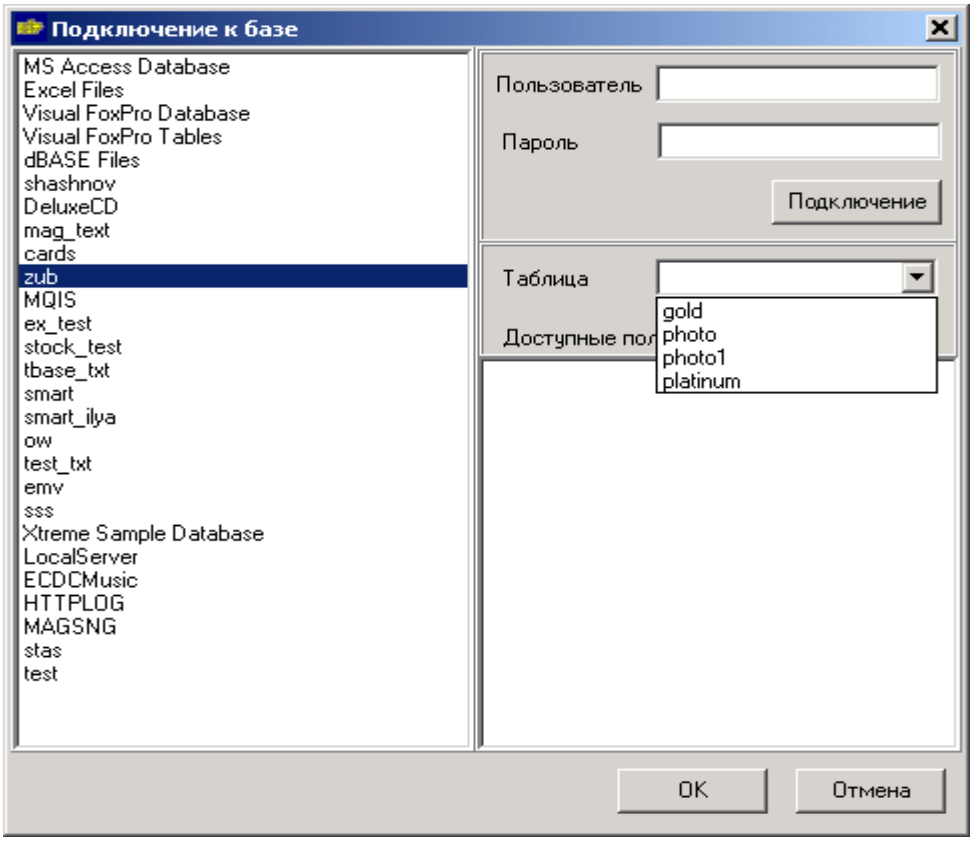
1) ***Источник данных*** – источник ODBC, который используется для доступа к входным данным.

Для привязки к источнику данных ODBC используется кнопка **выбрать источник**, после нажатия на которую появится следующая форма (окно «Подключение к базе»):



В левой части окна показаны все возможные источники данных ODBC, которые настраиваются в панели управления компьютером. Необходимо выбрать требуемый источник ODBC, ввести имя пользователя и пароль (если это необходимо) и нажать на кнопку ***Подключение***.

В случае успешного подключения в поле «Таблица» появится возможность выбора доступной таблицы входных данных, после выбора которой в поле «Доступные поля» будут показаны доступные поля, содержащие входные данные. Завершение настройки происходит по нажатию кнопки **OK**.



2) ***Таблица*** – таблица базы данных, из которой будут браться данные.
Для очистки источника данных используется кнопка **очистить**.

3) ***Доступные поля*** – поля таблицы базы данных.
Если источник данных настроен правильно, то при выборе узла ***Входная база данных*** в нижней части будут показаны исходные данные.

4) ***Текущая запись*** – будет выгружена одна текущая запись.

5) ***С текущей до последней*** – количество выгружаемых записей: с текущей записи таблицы по последнею.

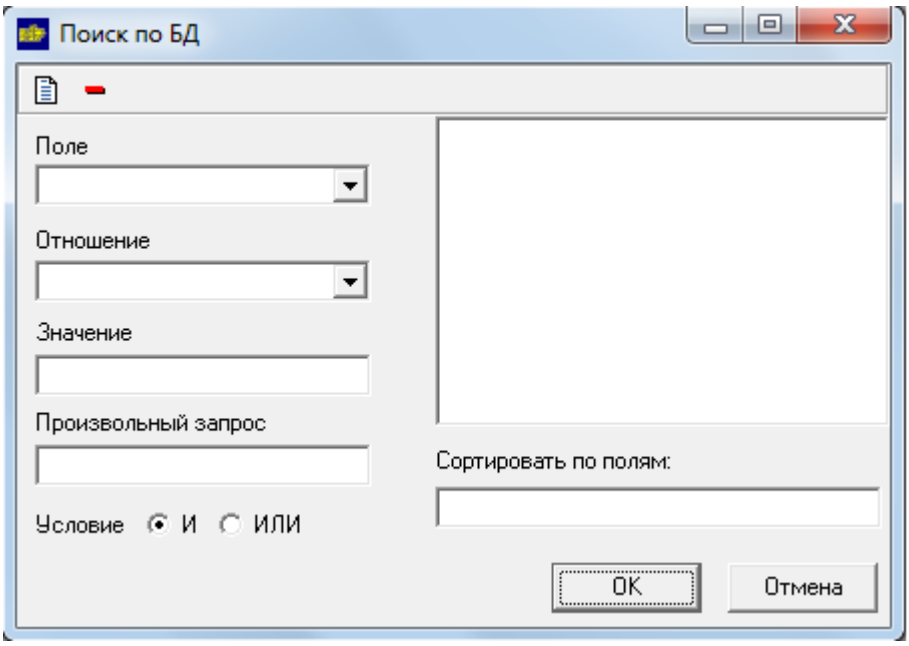
6) ***В диапазоне с … по …*** - определяет с какой по какую записи из таблицы будут выгружаться. Если база данных не используется, а в качестве входных значений используется клавиатура (по сути константы), то будет выгружена одна запись, сформированная на основе данных входных значений.

7) ***Перемещение по записям БД*** – используется для перемещения по таблице БД.

8) ***Перейти на запись*** – переход к записи по указанному порядковому номеру в последовательности записей входного файла.

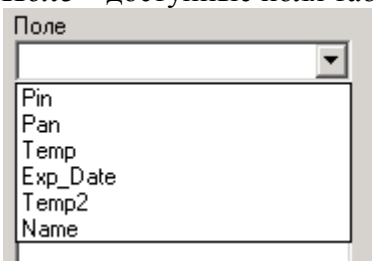
9) ***Поиск по записям*** – переход к записи по указанному определённому значению любого из полей записи входного файла.

10) ***Фильтр*** – используется для поиска по таблице БД. После нажатия на кнопку «Фильтр» появится следующая форма.

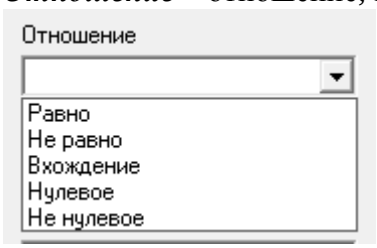


Добавить условие – добавить условие для поиска по таблице базы данных.
Удалить условие – удалить условие из поиска по таблице базы данных.

Поле – доступные поля таблицы базы данных. По этим полям может осуществляться поиск.



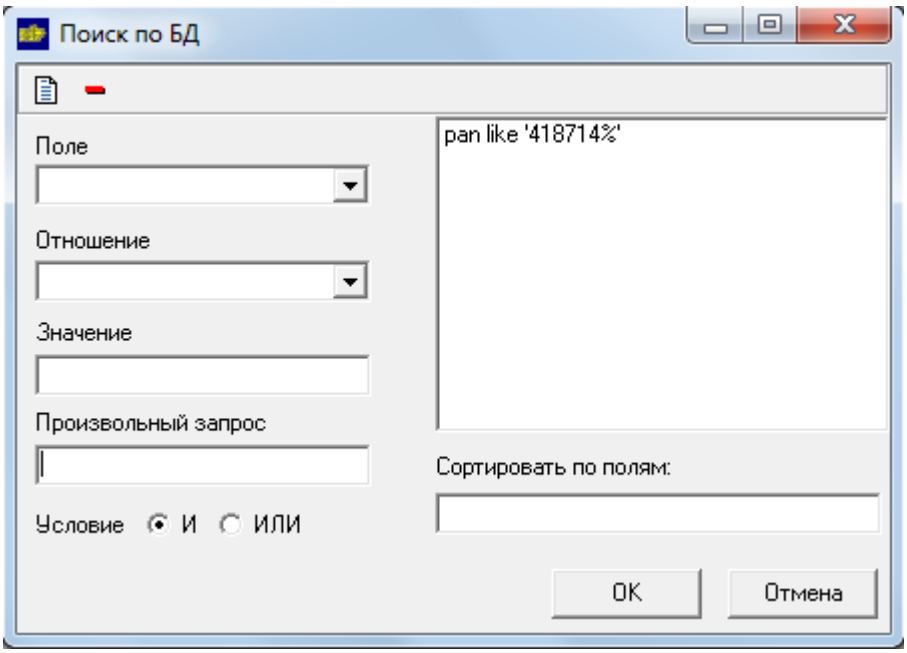
Отношение – отношение, которое будет применяться к выбранному полю.



Некоторые виды отношений требуют ввода конкретного значения в поле **Значение**, некоторые нет.
Произвольный запрос – .

Если добавлено больше одного условия, эти условия будут связаны между собой либо логическим **И**, либо логическим **ИЛИ**, которое задается используя переключатель **условие**.

Сортировать по полям: – сортировка отображения входных данных по содержимому указанного поля таблицы базы данных.



После того как условия настроены, необходимо нажать кнопку **OK**, для применения фильтра к входным данным. Условия наложенные фильтром, сохраняются при сохранении проекта. Таким образом можно например отфильтровать входные данные по продуктам.

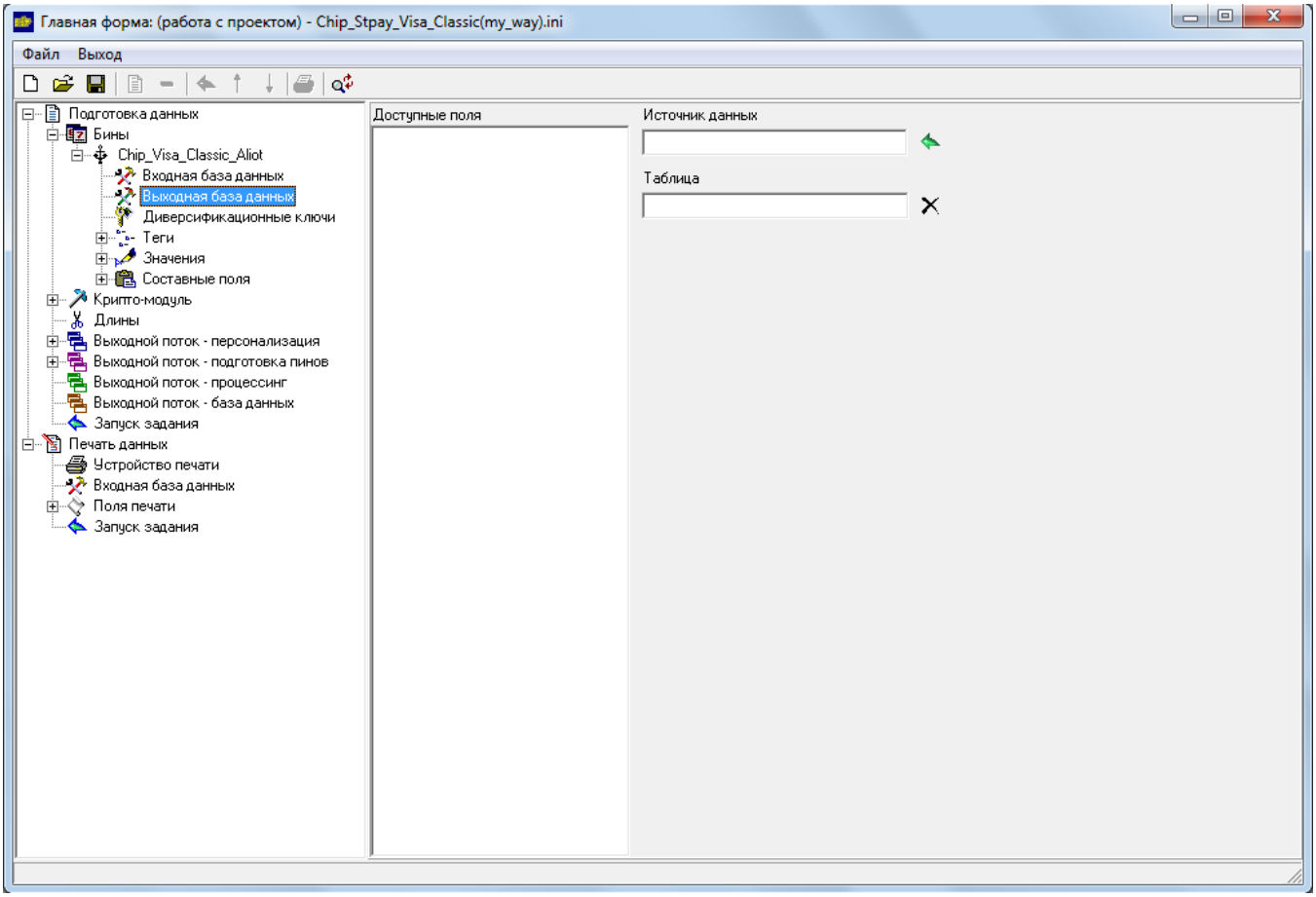
11) ***Снять фильтр*** – снимает ограничения на входные значения. Доступна только в случае, когда на входные данные был установлен фильтр.

12) ***Выбрать входной файл*** – позволяет вручную выбрать файл входных данных.

13) ***Директория для выбора файла*** – позволяет определить директорию расположения файлов входных данных.

5.1.3. Выходная база данных

Если в процессе подготовки данных требуются, сохранить значения в существующей базе данных, то необходимо настроить выходную базу данных.

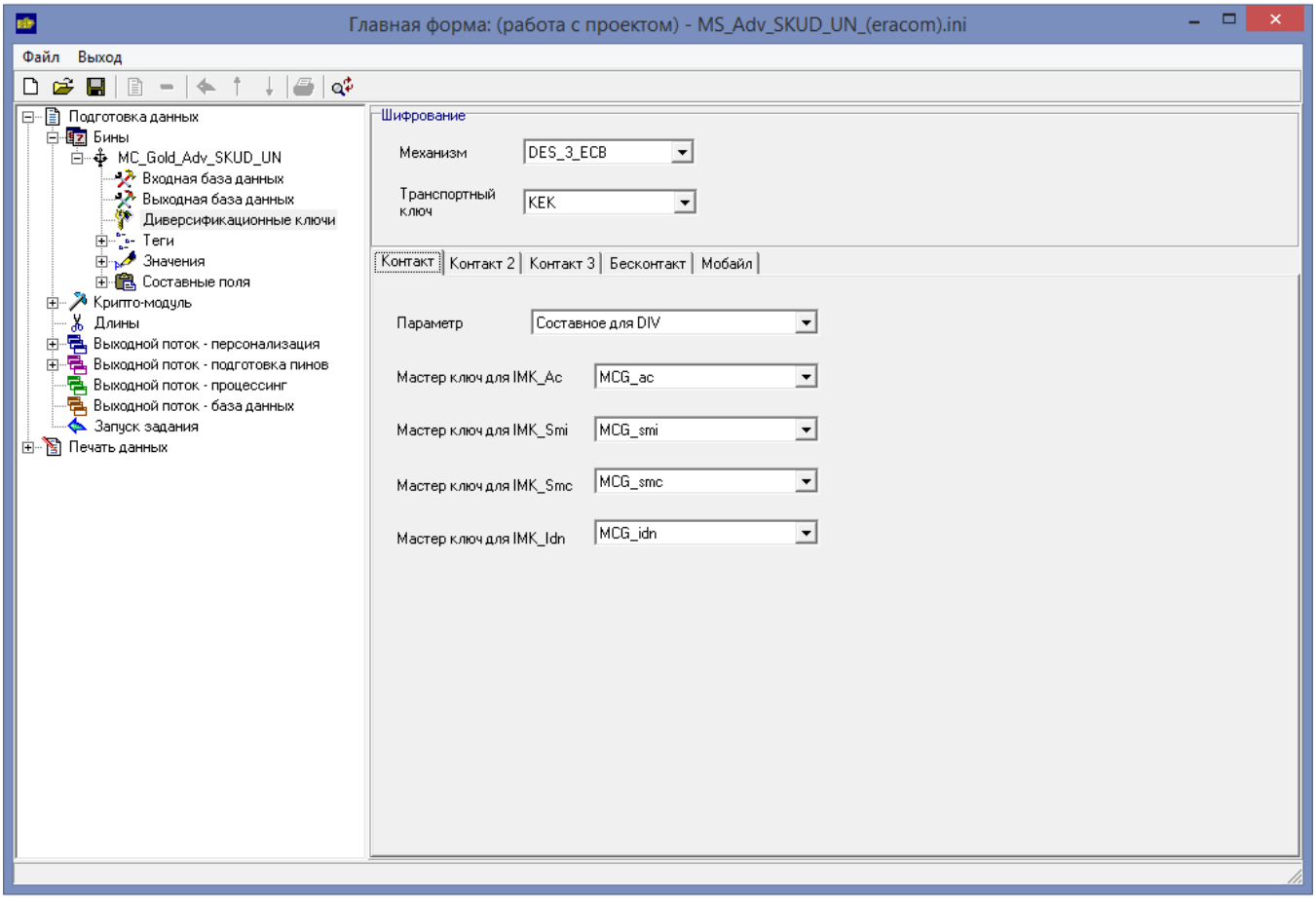


Процесс настройки выходной базы данных аналогичен настройке входной базы данных, за исключением того, что существующие данные не будут отображаться на экране. Данные будут добавляться новой записью в конец таблицы.

5.1.4. Диверсификационные ключи

Диверсификационные ключи представляют собой некое значение, сформированное по специальному алгоритму. Данные ключи используется в выходном потоке для персонализации (в случае подготовки чиповых данных), после чего записываются на карту и служат для проверки других данных, также записанных на карте.

5.1.4.1. Интерфейс *Контакт/Контакт2/Контакт3**.



* отображаемое количество контактных интерфейсов определяется числом наборов эмиссионных ключей, используемых приложениями, для которых осуществляется подготовка чиповых данных. Количество наборов ключей задаётся соответствующим значением параметра **CountApl** в разделе **[Proekt]** при прямом редактировании ini-файла проекта/шаблона. Возможные значения: 1, 2 или 3 максимум.

Шифрование – используется для шифрования, полученного значения.

Механизм – механизм, который будет использоваться для шифрования.

Транспортный ключ – один из ключей, который находится в хранилище ключей. Данный ключ будет использоваться при шифровании.

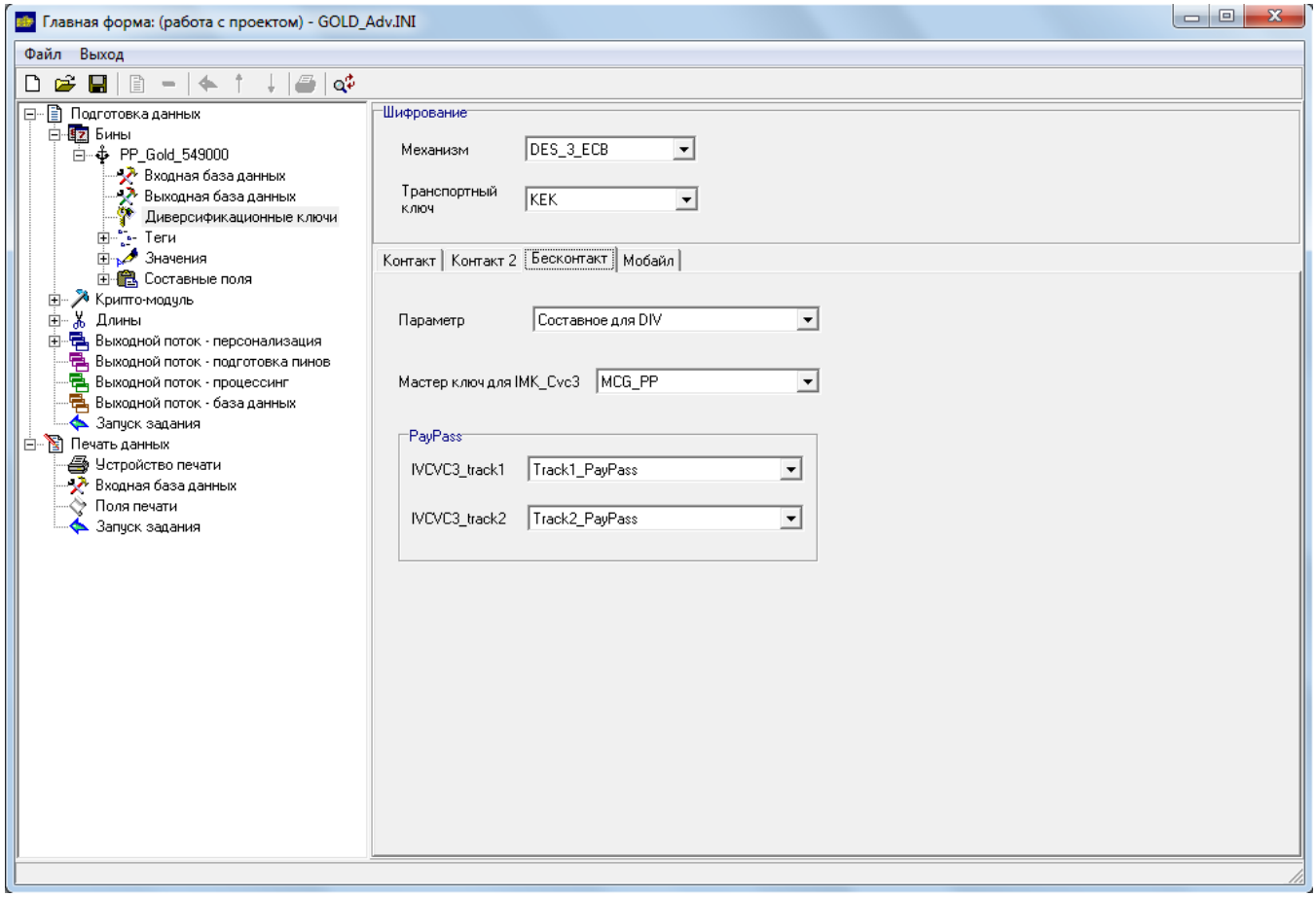
Вкладка ***Шифрование***, если она доступна в других местах, используется аналогично.

Параметр – значение, сформированное по специальному алгоритму. Значение формируется через составные поля, на основе входных данных.

Мастер ключ IMK_Ac/IMK_Smi/IMK_Smc/IMK_Idn – один из ключей, который находится в хранилище ключей. Он будет использоваться при диверсификации.

Одновременно можно сгенерировать четыре диверсификационных ключа (**IMK_Ac**, **IMK_Smi**, **IMK_Smc**, **IMK_Idn**).

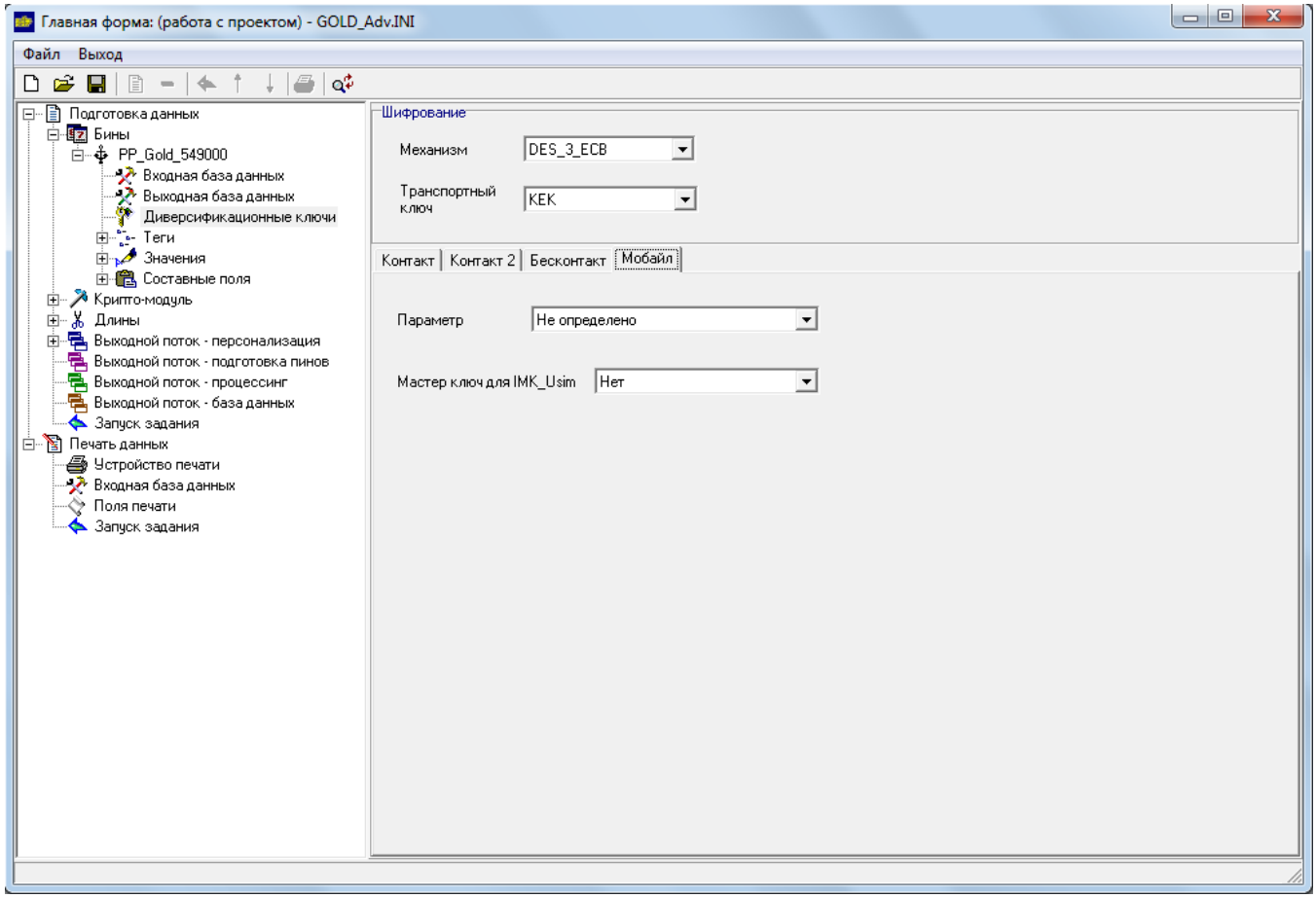
5.1.4.2. Интерфейс *Бесконтакт*.



Шифрование – используется для шифрования, полученного значения.
Механизм – механизм, который будет использоваться для шифрования.
Транспортный ключ – один из ключей, который находится в хранилище ключей. Данный ключ будет использоваться при шифровании.
Вкладка ***Шифрование***, если она доступна в других местах, используется аналогично.
Параметр – значение, сформированное по специальному алгоритму. Значение формируется через составные поля, на основе входных данных.
Мастер ключ IMK_Cvc3 – один из ключей, который находится в хранилище ключей. Он будет использоваться при диверсификации.
PayPass – .

IVCVC3_track1 – .
IVCVC3_track2 – .

5.1.4.3. Интерфейс *Мобайл*.



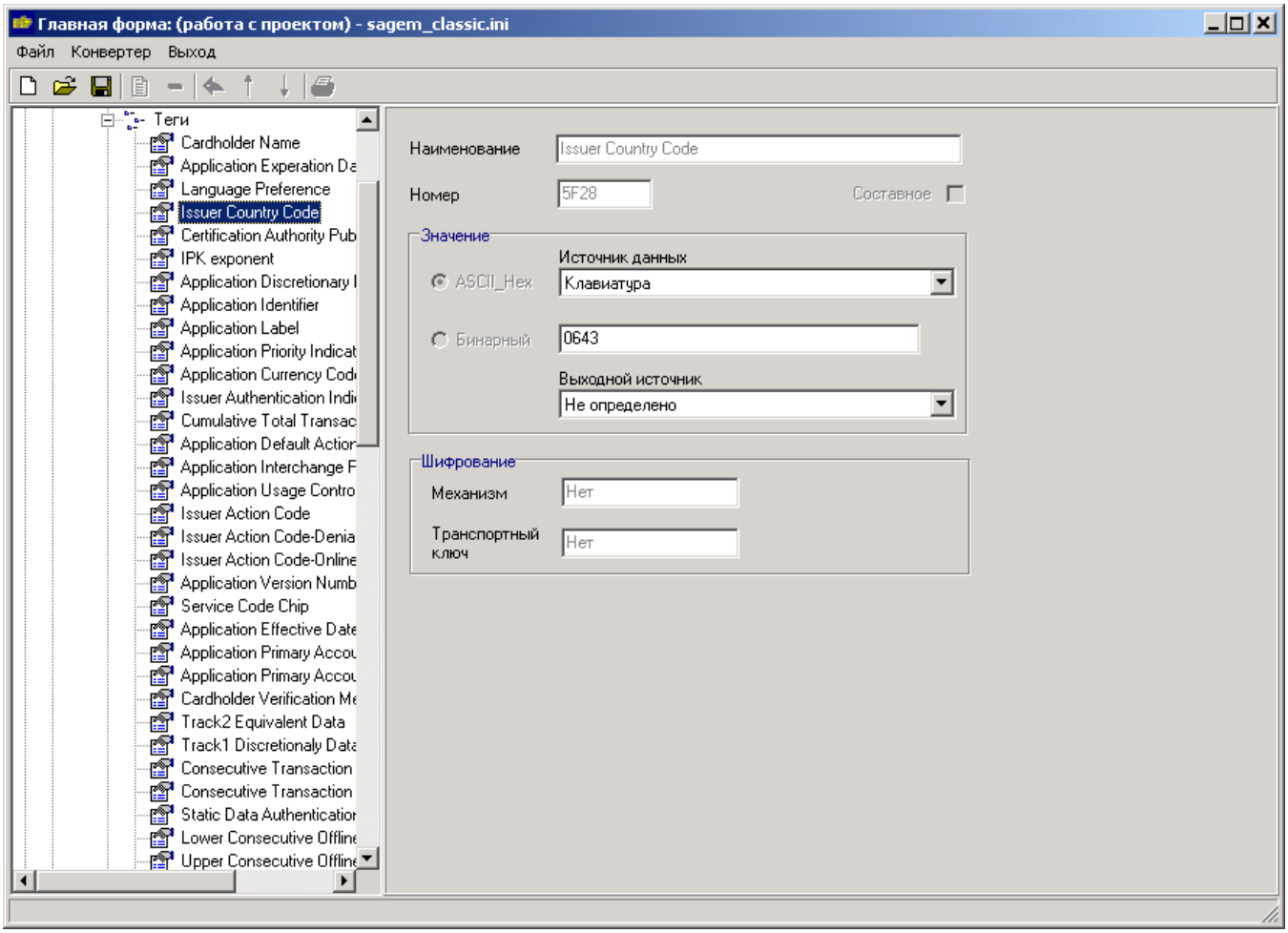
Шифрование – используется для шифрования, полученного значения.
Механизм – механизм, который будет использоваться для шифрования.
Транспортный ключ – один из ключей, который находится в хранилище ключей. Данный ключ будет использоваться при шифровании.
Вкладка ***Шифрование***, если она доступна в других местах, используется аналогично.
Параметр – значение, сформированное по специальному алгоритму. Значение формируется через составные поля, на основе входных данных.
Мастер ключ IMK_Usim – один из ключей, который находится в хранилище ключей. Он будет использоваться при диверсификации.

5.1.5. Теги + Значения

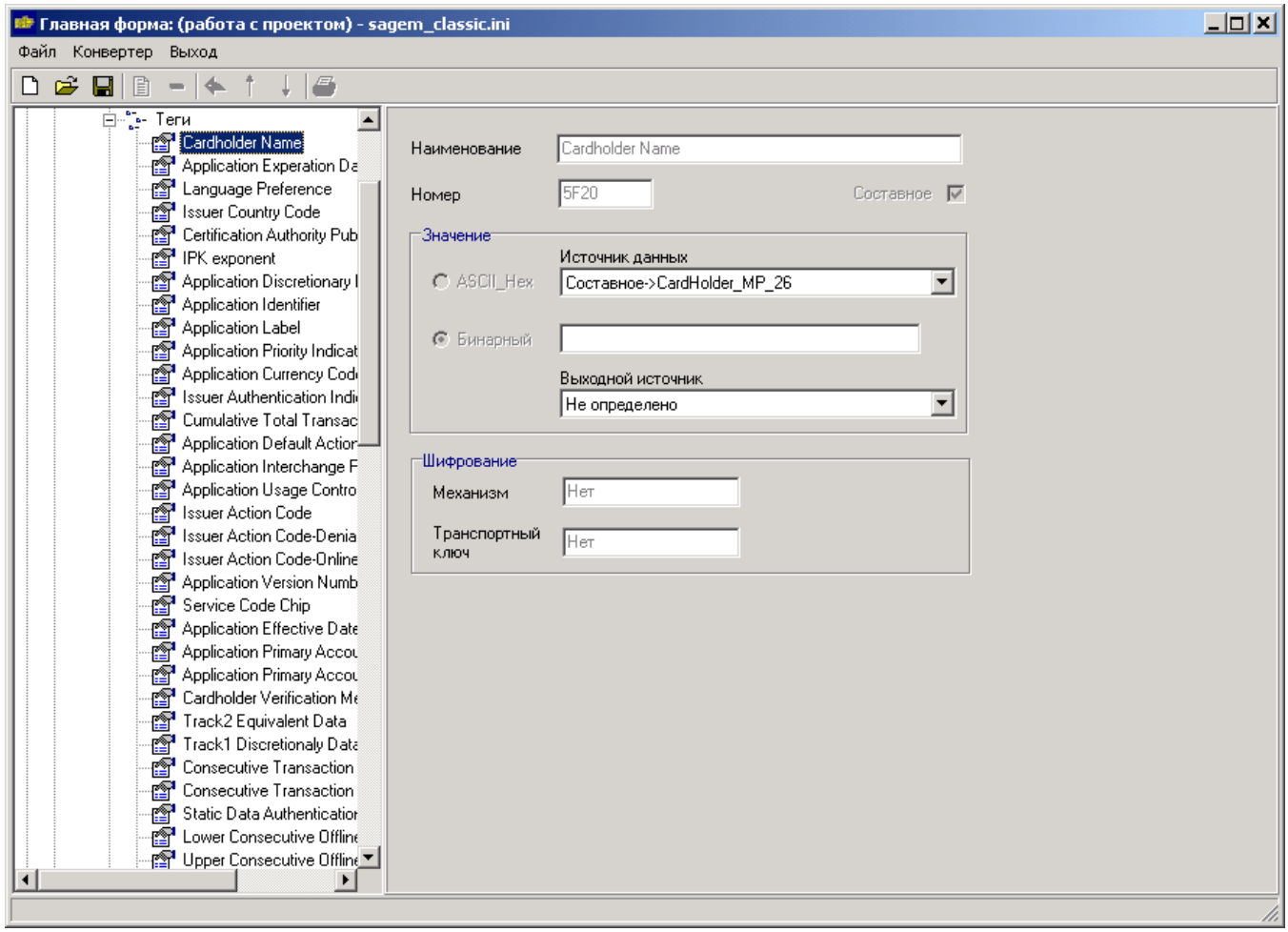
На данных вкладках настраивается источник данных для каждого элемента. Существует несколько источников откуда могут приходить данные.

Дата-время – входным значением является текущая дата и время.
Из файла – выбирается файл, в котором хранятся входные данные.
Клавиатура – входное значение вводится с клавиатуры.
Из базы данных – выбирается значение из входной базы данных, данное поле отображается в виде имя таблицы > имя поля.
Див. ключ->IMK_Ac/IMK_Ac2 – входным значением будет являться значение диверсификационного ключа IMK_Ac/IMK_Ac2.
Див. ключ->IMK_Smi/IMK_Smi2 – входным значением будет являться значение диверсификационного ключа IMK_Smi/IMK_Smi2.
Див. ключ->IMK_Smc/IMK_Smc2 – входным значением будет являться значение диверсификационного ключа IMK_Smc/IMK_Smc2.
Див. ключ->IMK_Idn/IMK_Idn2 – входным значением будет являться значение диверсификационного ключа IMK_Idn/IMK_Idn2.
Див. ключ->IMK_Cvc3 – входным значением будет являться значение диверсификационного ключа IMK_Cvc3.
Див. ключ->IMK_Usim1 – входным значением будет являться значение диверсификационного ключа IMK_Usim1.
Див. ключ->IMK_Usim2 – входным значением будет являться значение диверсификационного ключа IMK_Usim2.
Див. ключ->IMK_Usim3 – входным значением будет являться значение диверсификационного ключа IMK_Usim3.
Див. ключ (check value)->IMK_Ac/IMK_Ac2 – входным значением будет являться контрольная сумма диверсификационного ключа IMK_Ac/IMK_Ac2.
Див. ключ (check value)->IMK_Smi/IMK_Smi2 – входным значением будет являться контрольная сумма диверсификационного ключа IMK_Smi/IMK_Smi2.
Див. ключ (check value)->IMK_Smc/IMK_Smc2 – входным значением будет являться контрольная сумма диверсификационного ключа IMK_Smc/IMK_Smc2.
Див. ключ (check value)->IMK_Idn/IMK_Idn2 – входным значением будет являться контрольная сумма диверсификационного ключа IMK_Idn/IMK_Idn2.
Див. ключ (check value)->IMK_Cvc3 – входным значением будет являться контрольная сумма диверсификационного ключа IMK_Cvc3.
Див. ключ (check value)->IMK_Usim1 – входным значением будет являться контрольная сумма диверсификационного ключа IMK_Usim1.
Див. ключ (check value)->IMK_Usim2 – входным значением будет являться контрольная сумма диверсификационного ключа IMK_Usim2.
Див. ключ (check value)->IMK_Usim3 – входным значением будет являться контрольная сумма диверсификационного ключа IMK_Usim3.

Контрольные суммы диверсификационных ключей (в качестве входных данных) можно использовать только для значений


Главная форма: (работа с проектом) - sagem_classic.ini
Файл
Конвертер
Выход
Теги
Cardholder Name
Application Experation Da
Language Preference
Issuer Country Code
Certification Authority Pub
IPK exponent
Application Discretionary I
Application Identifier
Application Label
Application Priority Indicat
Application Currency Cod
Issuer Authentication Indi
Cumulative Total Transac
Application Default Actior
Application Interchange F
Application Usage Contro
Issuer Action Code
Issuer Action Code-Denia
Issuer Action Code-Online
Application Version Numb
Service Code Chip
Application Effective Date
Application Primary Accou
Application Primary Accou
Cardholder Verification Me
Track2 Equivalent Data
Track1 Discretionaly Data
Consecutive Transaction
Consecutive Transaction
Static Data Authenticatior
Lower Consecutive Offline
Upper Consecutive Offline
Наименование
Issuer Country Code
Номер
5F28
Составное
Значение
Источник данных
ASCII_Hex
Клавиатура
Бинарный
0643
Выходной источник
Не определено
Шифрование
Механизм
Нет
Транспортный ключ
Нет

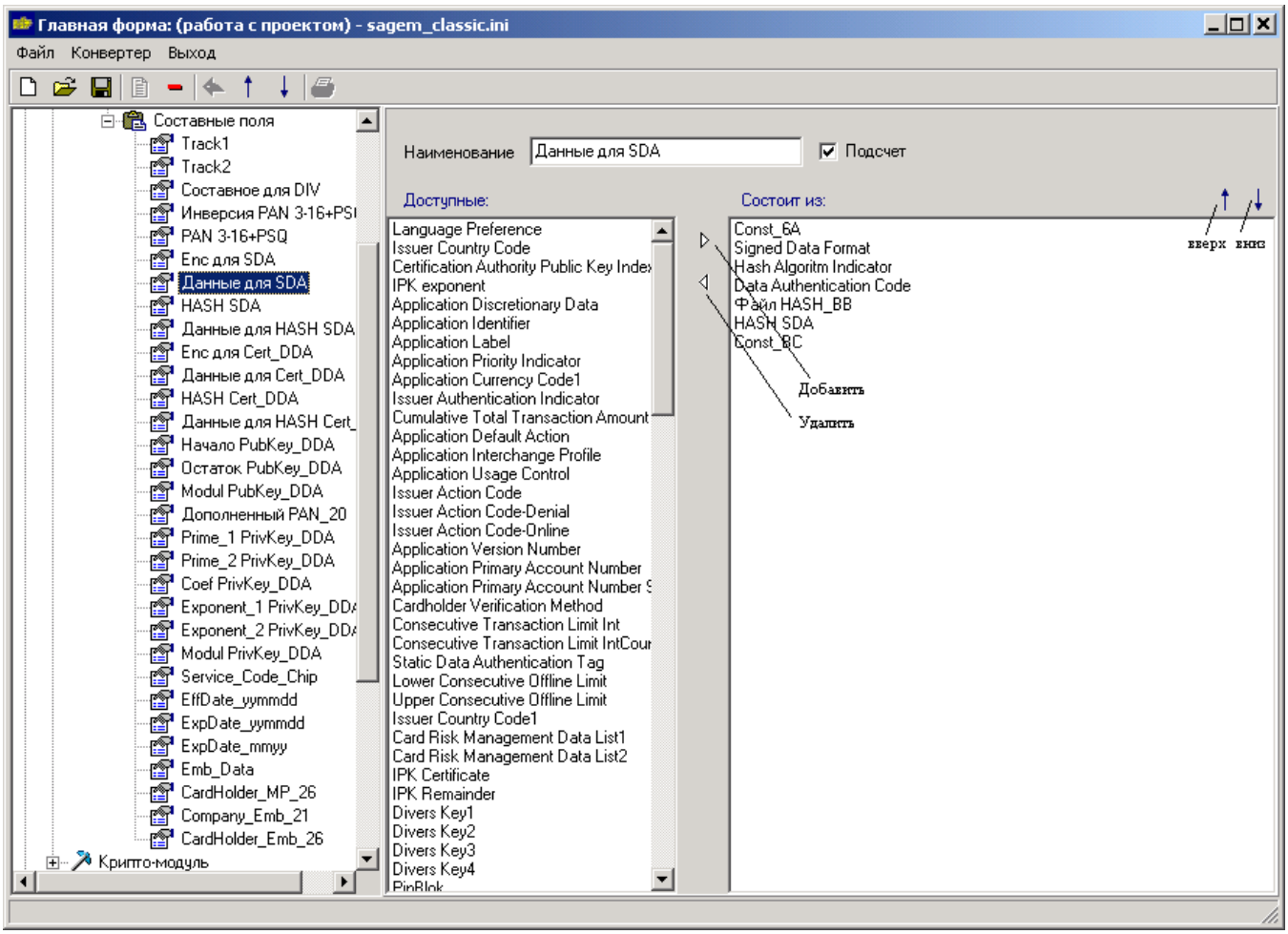
Если входное значение элемента в шаблоне было описано как составное, то входным значением для данного элемента будет составное поле.



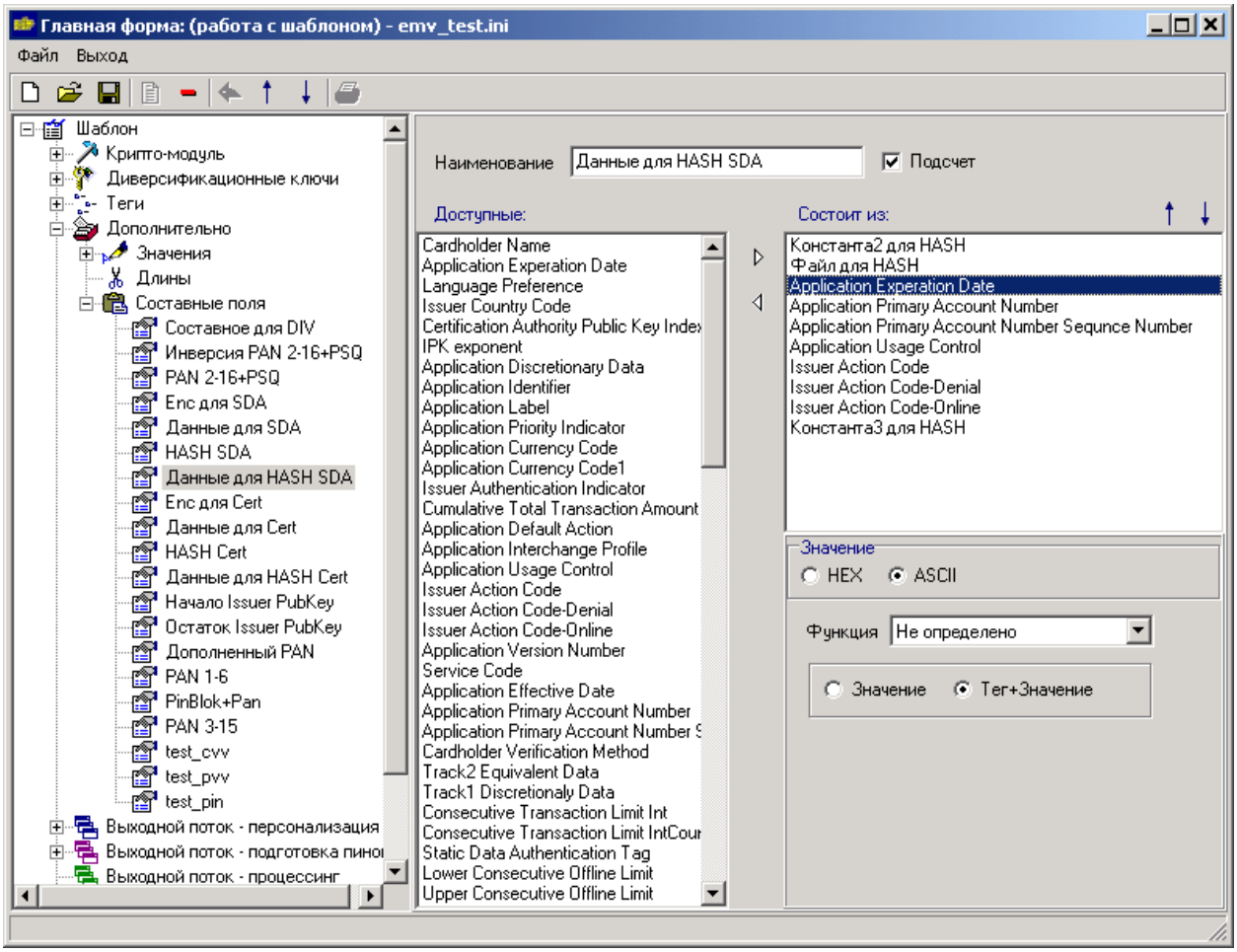
Выходной источник – выбирается поле из выходной базы данных, в которое будет записываться значение, данное поле отображается в виде имя таблицы > имя поля.

5.1.6. Составные поля

Составные поля используются, когда необходимо из других элементов создать составное поле. К составным полям могут применяться определенные функции, изменяющие входное значение. В левой части экрана, очень важен **порядок следования полей**, так как одни составные поля могут включать в себя другие поля. Поле, которое подсчитывается первым, должно быть расположено в самом низу.

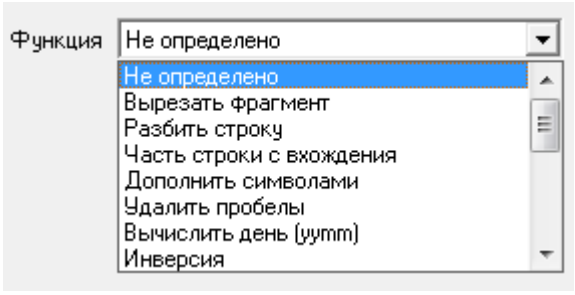


Наименование – наименование составного поля.
Подсчет – указывает необходимо ли подсчитывать значения во время выпуска карт, так как бывает, что некоторые составные поля формируются для одноразовых действий.
Доступные – показывает доступные элементы, которые могут быть использованы для формирования составного поля. В данную группу входят ключи (имеющие тип **RSA** или **RSA_DDA**), диверсификационные ключи, теги, значения и составные поля. Не входят теги и значения, у которых формат входного значение определен как составное.
Состоит из – показывает элементы из которых будет формироваться составное поле.
Добавить, удалить – используется для переноса элементов из одной группы в другую.
Вверх, вниз – определяет порядок следования элементов в составном поле. Результатом составного поля будет сложение входных значений в определенном порядке.



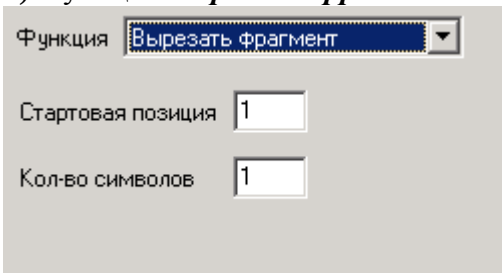
Если в составное поле входит тег, то необходимо выбрать, что будет использоваться в качестве входных данных (TLV или просто значение). Также к входному значению может быть применена одна из ***функций***.

Значение – определяет формат значения, которое будет приходить в функцию (или участвовать в составном поле, если функция не определена). Функция возвращает бинарное значение (ASCII).



Описание функций:

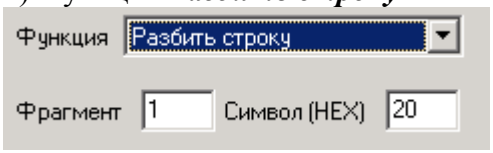
1) Функция ***Вырезать фрагмент***



Вырезает фрагмент с определенной позиции.
Служит для вырезания части входного значения.

Стартовая позиция – стартовая позиция.
Кол-во символов – количество символов со стартовой позиции.

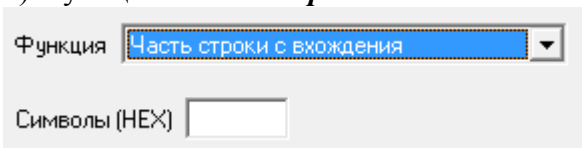
2) Функция ***Разбить строку***



Разбивает строку по заданному разделителю.
Например, если входные данные идут в виде Фамилия/Имя/Титул, то с ее помощью можно получить отдельное значение, используя разделитель /.

Фрагмент – определяет номер вхождения, которое будет использоваться в качестве результата.
Символ (HEX) – код символа, используемого в качестве разделителя. Код задается в HEX.

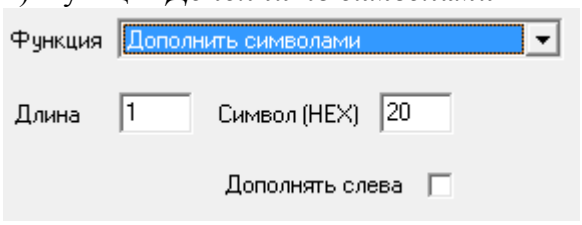
3) Функция ***Часть строки с вхождения***



Служит для вырезания части (фрагмента) входного значения (строки), начиная с определенного символа (или последовательности символов), включая указанный символ (или последовательность символов).

Символ (HEX) – шестнадцатеричный код символа (или последовательности символов), используемого в качестве начала чисти входного значения (строки), предназначенной для выделения.

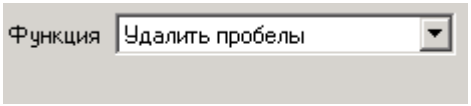
4) Функция ***Дополнить символами***



Дополняет символами до определенной длины.
Обычно используется для дополнения пробелами имени клиента (CardHolder Name) до 26 символов, с последующей записью на первую дорожку магнитной полосы.

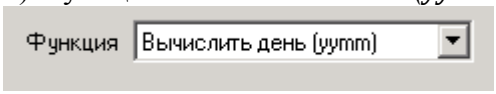
Длина – длина до которой дополняется значение.
Символ (HEX) – код символа, которым будет дополняться значение. Код задается в HEX.
Дополнять слева – в случае необходимости, дополнительные символы будут располагаться слева от используемых данных.

5) Функция ***Удалить пробелы***



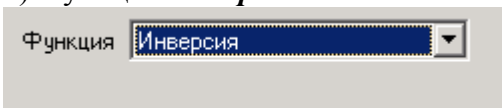
Удаление пробелов слева и справа.
Функция используется для удаления ненужных пробелов из входного значения.

6) Функция ***Вычислить день (yymm)***



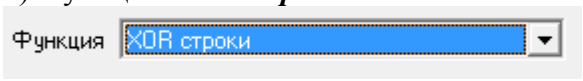
Используется для вычиления последнего дня месяца. Функция должна применяться к данным в формате yymm.

7) Функция ***Инверсия***

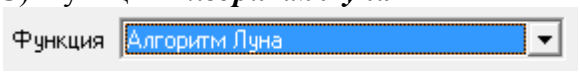


Логическое отрицание входного значение (not).
Используется для создания параметров для диверсификационных ключей.

8) Функция ***XOR строки***

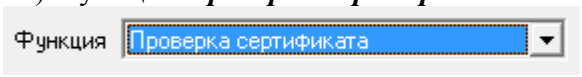


9) Функция ***Алгоритм Луна***



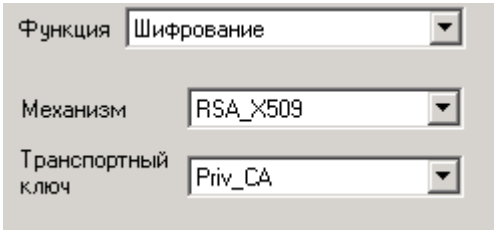
Используется алгоритм Ганса Питера Луна для вычисления последней цифры номера карты PAN. В качестве входного значения должен использоваться PAN кроме последней цифры.

10) Функция ***Проверка сертификата***



Используется для проверки срока действия сертификата. При выполнении функции осуществляется сравнение срока действия сертификата и срока действия выпускаемой карты. В случае, если срок действия сертификата превышает срок действия выпускаемой карты на разницу менее, чем указано в соответствующем значении (с точностью до одного месяца), то процессе подготовки данных появится предупреждение об окончании срока действия сертификата. Процесс подготовки данных будет остановлен.

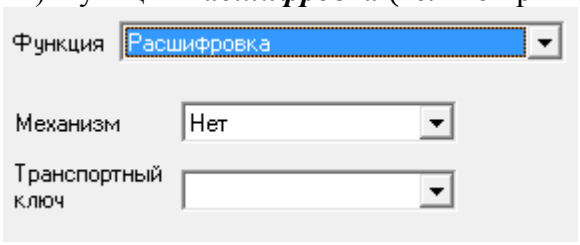
11) Функция ***Шифрование***



Шифрование входного значения.
Используется для шифрования исходного значения, с целью использовать в дальнейшем зашифрованного значения.

Механизм – механизм, который будет использоваться для шифрования.
Транспортный ключ – один из ключей, который находится в хранилище ключей.

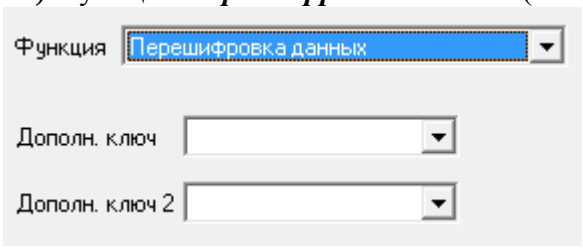
12) Функция ***Расшифровка*** (только при использовании крипто-модуля Eracom)



Расшифровка входного значения.
Используется для дешифрования исходного значения, с целью использовать в дальнейшем дешифрованного значения.

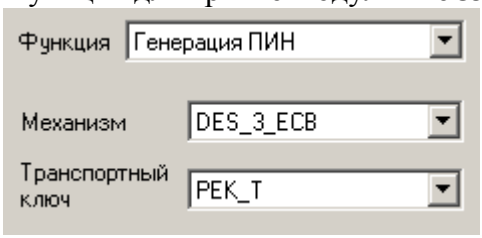
Механизм – механизм, который будет использоваться для расшифровки.
Транспортный ключ – один из ключей, который находится в хранилище ключей.

13) Функция ***Перешифровка данных*** (только при использовании крипто-модуля Eracom)



Дополн. ключ – .
Дополн. ключ2 – .

14) Функция ***Генерация ПИН***
Функция для крипто-модуля **Eracom/Eracom_EFT**:

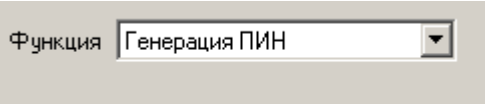


Генерация числа (ПИН-блока) на основе некоторых данных, по определенному алгоритму.

Механизм – механизм, который будет использоваться при шифровании.
Транспортный ключ – один из ключей, который находится в хранилище ключей.

В качестве входного значения должен использоваться PAN.
Функция возвращает 8 символов (PinBlok).

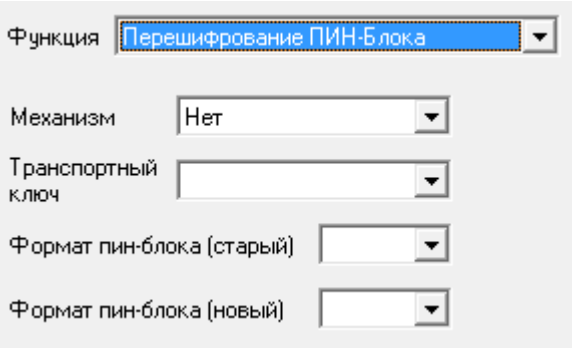
Функция для крипто-модуля **Thales**:



Генерация числа на основе некоторых данных, по определенному алгоритму.

В качестве входного значения должно использоваться:
12 цифр PAN, заканчивая предпоследней цифрой PAN (например, если PAN имеет длину 16 цифр, то используются цифры PAN с 4 по 15 включительно).
Функция возвращает 5 символов (ПИН в формате Thales).

15) Функция ***Перешифрование ПИН-Блока*** (только при использовании крипто-модулей Thales и Eracom_EFT)

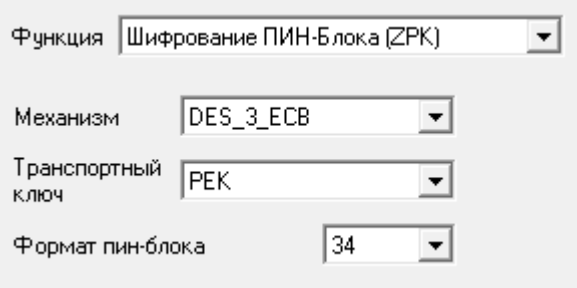


Перешифрование ПИН-блока из одного формата в другой.

Механизм – механизм, который будет использоваться при шифровании.
Транспортный ключ – один из ключей, который находится в хранилище ключей.
Формат ПИН-блока (старый) – исходный формат ПИН-блока.
Формат ПИН-блока (новый) – новый формат ПИН-блока.

В качестве входного значения должно использоваться:
1) 12 цифр PAN, заканчвиая предпоследней цифрой PAN (например, если PAN имеет длину 16 цифр, то используюстя цифры PAN с 4 по 15 включительно)
плюс
2) ПИН-блок в формате 01 (8 символов).
Функция возвращает 8 символов (ПИН-блок) в новом формате.

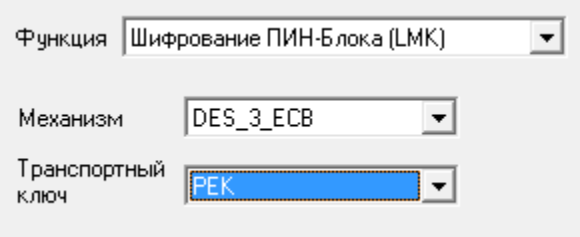
16) Функция ***Шифрование ПИН-Блока (ZPK)*** (только при использовании крипто-модуля Thales)



Генерация числа (ПИН-блока) в определённом формате на основе некоторых данных по определённому алгоритму.
В качестве входного значения должно использоваться:

Механизм – механизм, который будет использоваться при шифровании.
Транспортный ключ – один из ключей, который находится в хранилище ключей.
Формат ПИН-блока – формат ПИН-блока.

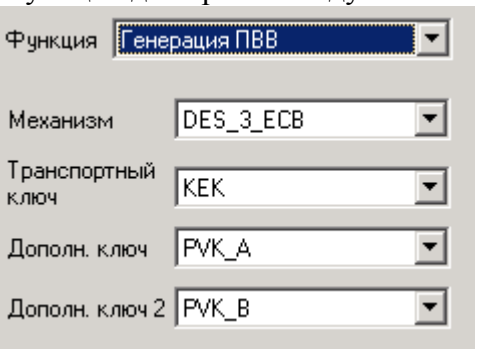
17) Функция ***Шифрование ПИН-Блока (LMK)*** (только при использовании крипто-модуля Thales)



Генерация числа (ПИН-блока) в определённом формате на основе некоторых данных по определённому алгоритму.
В качестве входного значения должно использоваться:

Механизм – механизм, который будет использоваться при шифровании.
Транспортный ключ – один из ключей, который находится в хранилище ключей.

18) Функция ***Генерация ПВВ*** (только при использовании крипто-модулей Thales и Eracom)
Функция для крипто-модуля **Eracom**:



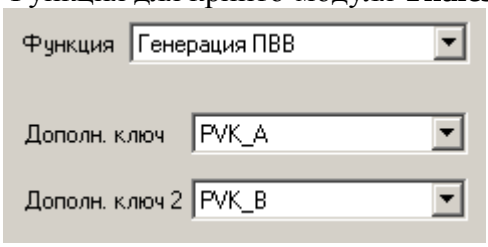
Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Механизм – механизм, который будет использоваться при шифровании.
Транспортный ключ – один из ключей, который находится в хранилище ключей.
Дополнительный ключ – один из ключей, который находится в хранилище ключей.

Дополнительный ключ 2 – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) Значение, которое вернула функция генерации ПИНа (PinBlok), переведенное в HEX
плюс
2) 12 цифр PAN, заканчивая предпоследней цифрой PAN (например, если PAN имеет длину 16 цифр, то используются цифры PAN с позиции 4 по 15 включительно)
плюс
3) PVKI (одна цифра, в дипазоне 1-6).
Функция возвращает 4 символа.

Функция для крипто-модуля **Thales**:

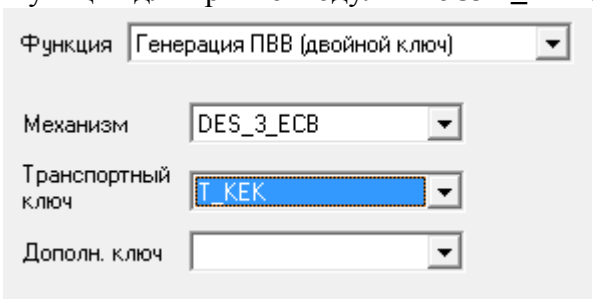


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.
Дополнительный ключ 2 – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) Значение, которое вернула функция генерации ПИНа для Thales (5 символов)
плюс
2) 12 цифр PAN, заканчивая предпоследней цифрой PAN (например, если PAN имеет длину 16 цифр, то используются цифры PAN с позиции 4по 15 включительно)
плюс
3) PVKI (одна цифра, в дипазоне 1-6).
Функция возвращает 4 символа.

19) Функция ***Генерация ПВВ (двойной ключ)*** (только при использовании крипто-модулей Thales и Eracom_EFT)
Функция для крипто-модуля **Eracom_EFT**:

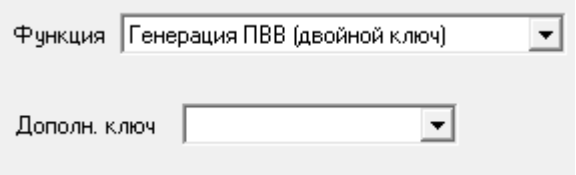


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Механизм – механизм, который будет использоваться при шифровании.
Транспортный ключ – один из ключей, который находится в хранилище ключей.
Дополнительный ключ – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) Значение, которое вернула функция генерации ПИНа (PinBlok), переведенное в HEX
плюс
2) 12 цифр PAN, заканчивая предпоследней цифрой PAN (например, если PAN имеет длину 16 цифр, то используются цифры PAN с позиции 4 по 15 включительно)
плюс
3) PVKI (одна цифра, в дипазоне 1-6).
Функция возвращает 4 символа.

Функция для крипто-модуля **Thales**:

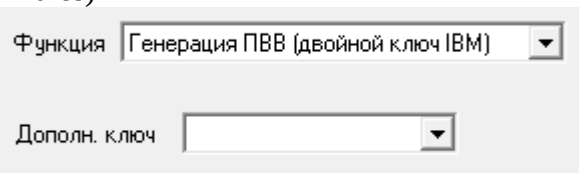


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) Значение, которое вернула функция генерации ПИНа для Thales (5 символов)
плюс
2) 12 цифр PAN, заканчивая предпоследней цифой PAN (например, если PAN имеет длину 16 цифр, то используются цифры PAN с позиции 4 по 15 включительно)
плюс
3) PVKI (одна цифра, в дипазоне 1-6).
Функция возвращает 4 символа.

20) Функция ***Генерация ПВВ (двойной ключ IBM)*** (только при использовании крипто-модуля Thales)

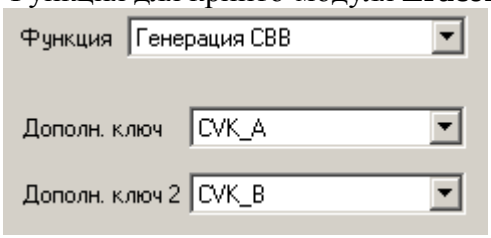


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) Значение, которое вернула функция генерации ПИНа для Thales (5 символов)
плюс
2) 12 цифр PAN, заканчивая предпоследней цифой PAN (например, если PAN имеет длину 16 цифр, то используются цифры PAN с позиции 4 по 15 включительно)
плюс
3) PVKI (одна цифра, в дипазоне 1-6).
Функция возвращает 4 символа.

21) Функция ***Генерация CBB*** (только при использовании крипто-модулей Thales и Eracom)
Функция для крипто-модуля **Eracom**:

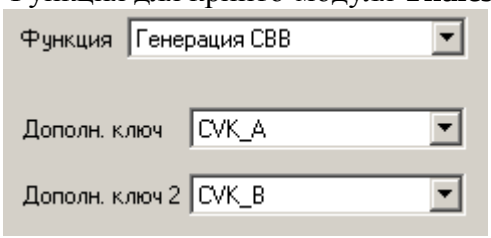


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.
Дополнительный ключ 2 – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) Service Code (три цифры)
плюс
2) Exp Date (4 цифры) - формат yymm
плюс
3) PAN
Функция возвращает 3 символа.

Функция для крипто-модуля **Thales**:



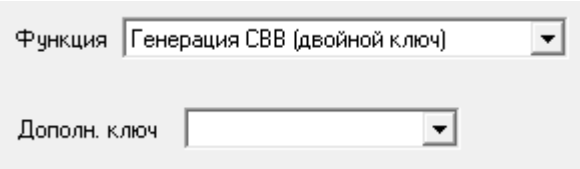
Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.
Дополнительный ключ 2 – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) PAN
плюс
2) Константа **;**
плюс
3) Exp Date (4 цифры) - формат yymm
плюс
4) Service Code (три цифры)
Функция возвращает 3 символа.

22) Функция ***Генерация CBB (двойной ключ)*** (только при использовании крипто-модулей Thales и Eracom_EFT)

Функция для крипто-модуля **Eracom_EFT**:

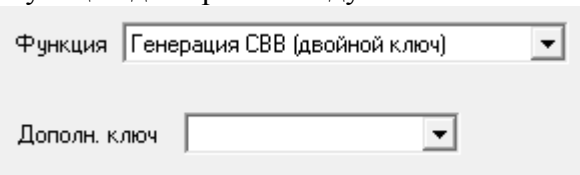


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) Service Code (три цифры)
плюс
2) Exp Date (4 цифры) - формат yymm
плюс
3) PAN
Функция возвращает 3 символа.

Функция для крипто-модуля **Thales**:

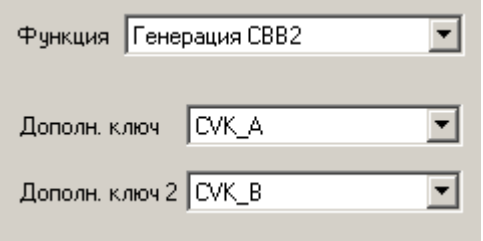


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) PAN
плюс
2) Константа **;**
плюс
3) Exp Date (4 цифры) - формат yymm
плюс
4) Service Code (три цифры)
Функция возвращает 3 символа.

23) Функция ***Генерация CBB2*** (только при использовании крипто-модулей Thales и Eracom)
Функция для крипто-модуля **Eracom**:

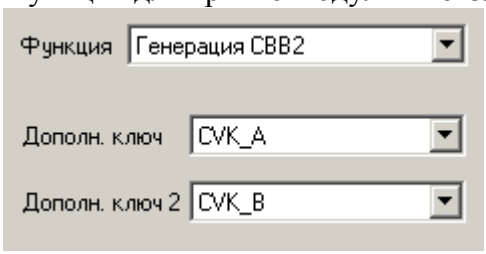


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.
Дополнительный ключ 2 – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) три нуля (вместо Service Code для CBB)
плюс
2) Exp Date (4 цифры) – формат mmyy
плюс
3) PAN
Функция возвращает 3 символа.

Функция для крипто-модуля **Thales**:

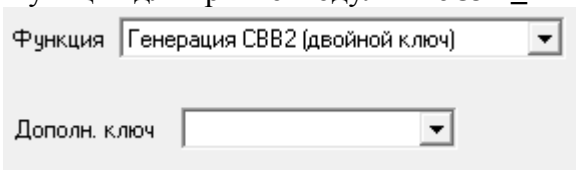


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.
Дополнительный ключ 2 – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) PAN
плюс
2) Константа **;**
плюс
3) Exp Date (4 цифры) - формат mmyy
плюс
4) три нуля (вместо Service Code для CBB)
Функция возвращает 3 символа.

24) Функция ***Генерация CBB2 (двойной ключ)*** (только при использовании крипто-модулей Thales и Eracom_EFT)
Функция для крипто-модуля **Eracom_EFT**:



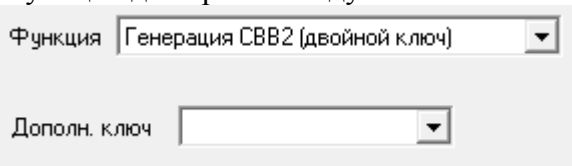
Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:

1) три нуля (вместо Service Code для CBB)
плюс
2) Exp Date (4 цифры) – формат mmyy
плюс
3) PAN
Функция возвращает 3 символа.

Функция для крипто-модуля **Thales**:

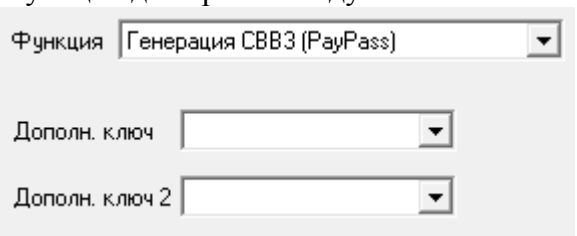


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) PAN
плюс
2) Константа **;**
плюс
3) Exp Date (4 цифры) - формат mmyy
плюс
4) три нуля (вместо Service Code для CBB)
Функция возвращает 3 символа.

25) Функция ***Генерация CBB3 (PayPass)*** (только при использовании крипто-модулей Thales и Eracom)
Функция для крипто-модуля **Eracom**:

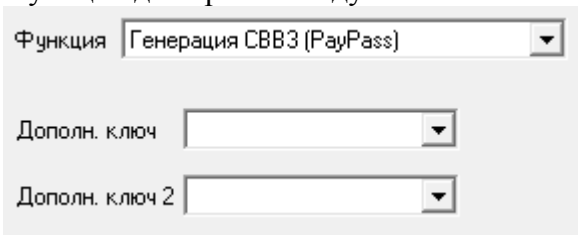


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.
Дополнительный ключ 2 – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) три нуля (вместо Service Code для CBB)
плюс
2) Exp Date (4 цифры) – формат mmyy
плюс
3) PAN
Функция возвращает 3 символа.

Функция для крипто-модуля **Thales**:

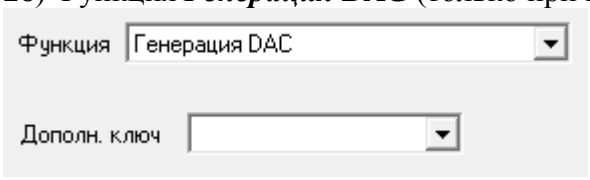


Генерация числа на основе некоторых данных, по определенному алгоритму.
Данное число записывается на карту, и участвует в проверке действительности карты.

Дополнительный ключ – один из ключей, который находится в хранилище ключей.
Дополнительный ключ 2 – один из ключей, который находится в хранилище ключей.

В качестве входного значения должно использоваться:
1) PAN
плюс
2) Константа **;**
плюс
3) Exp Date (4 цифры) - формат mmyy
плюс
4) три нуля (вместо Service Code для CBB)
Функция возвращает 3 символа.

26) Функция ***Генерация DAC*** (только при использовании крипто-модуля Eracom_EFT)



Дополнительный ключ – один из ключей, который находится в хранилище ключей.

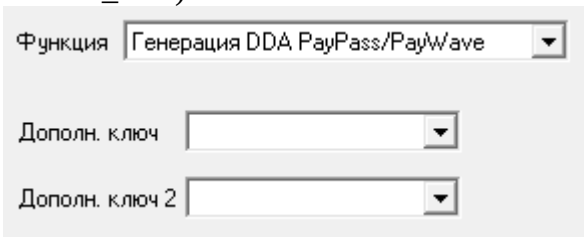
27) Функция ***Генерация SDA*** (только при использовании крипто-модуля Eracom_EFT)

Дополнительный ключ – один из ключей, который находится в хранилище ключей.

28) Функция ***Генерация DDA*** (только при использовании крипто-модуля Eracom_EFT)

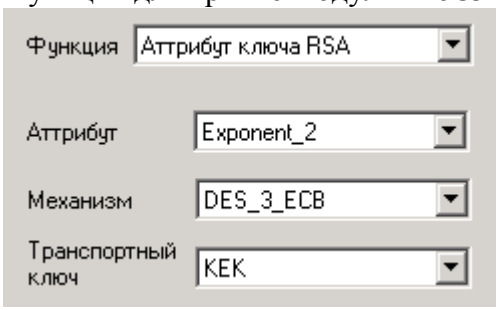
Дополнительный ключ – один из ключей, который находится в хранилище ключей.

29) Функция ***Генерация DDA PayPass/PayWave*** (только при использовании крипто-модуля Eracom_EFT)



Дополнительный ключ – один из ключей, который находится в хранилище ключей.
Дополнительный ключ 2 – один из ключей, который находится в хранилище ключей.

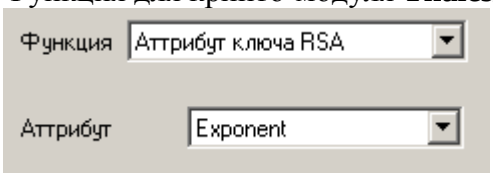
30) Функция ***Аттрибут ключа RSA***
Функция для крипто-модуля **Eracom**:



Если в качестве входного значения используются ключ, то становится доступной функция получить аттрибут ключа RSA.

Аттрибут – аттрибут ключа, который будет возвращен функцией.
Механизм – механизм, который будет использоваться при шифровании.
Транспортный ключ – один из ключей, который находится в хранилище ключей.

Функция для крипто-модуля **Thales/Eracom_EFT**:



Если в качестве входного значения используются ключ, то становится доступной функция получить аттрибут ключа RSA.

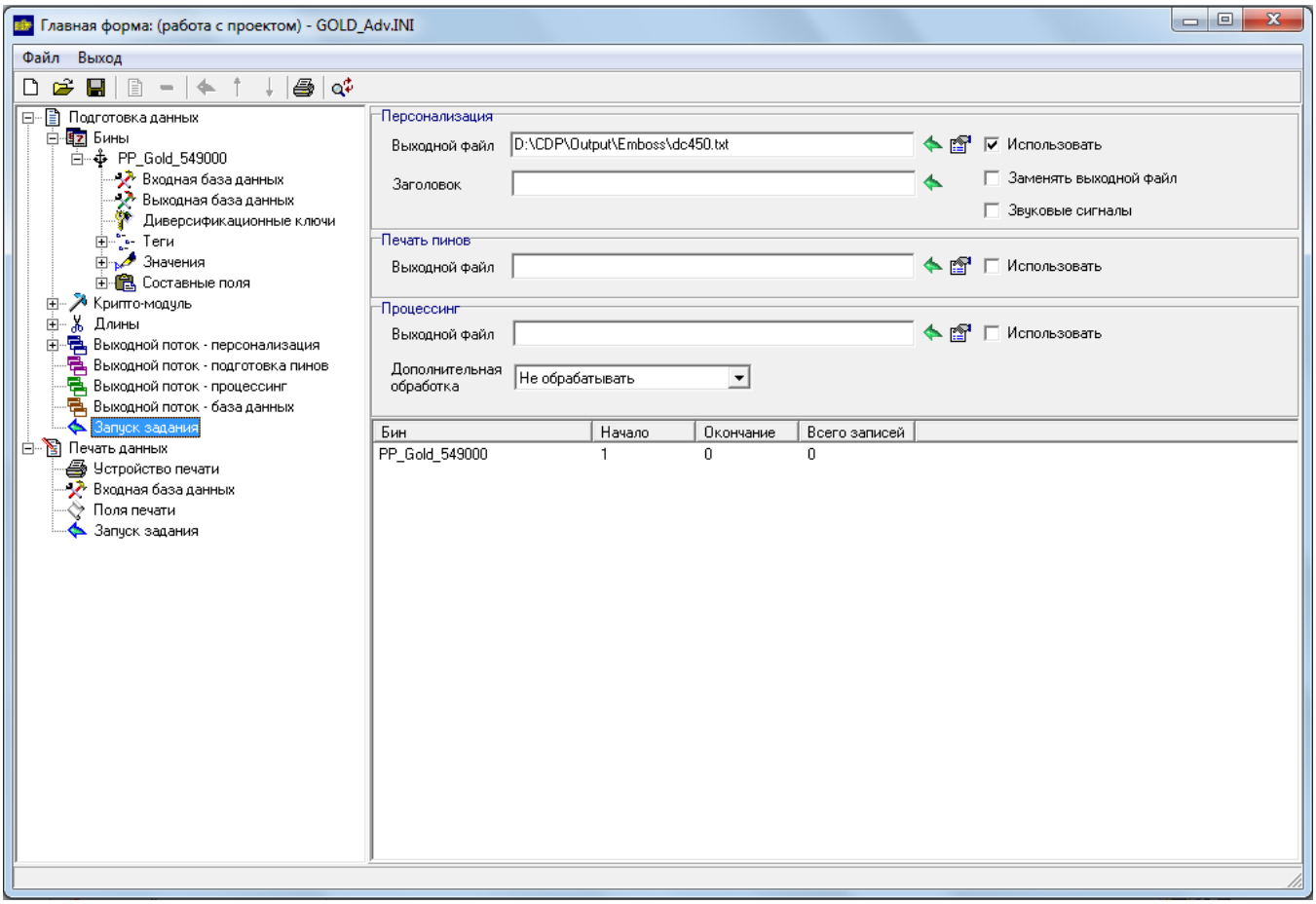
Аттрибут – аттрибут ключа, который будет возвращен функцией.

Возвращаемое значение будет зашифровано под транспортным ключом.

5.1.7. Запуск задания

Служит для настройки выходных файлов и запуска задания на подготовку данных. Количество записей задается в настройке узла входной базы данных для каждого бина. Отображаются только те бины, в настройках которых стоит галочка использовать при подготовке данных. Можно подготавливать данные для персонализации, процессинга и для для последующей печати ПИНов. Необходимо отметить использовать, для необходимых действий.

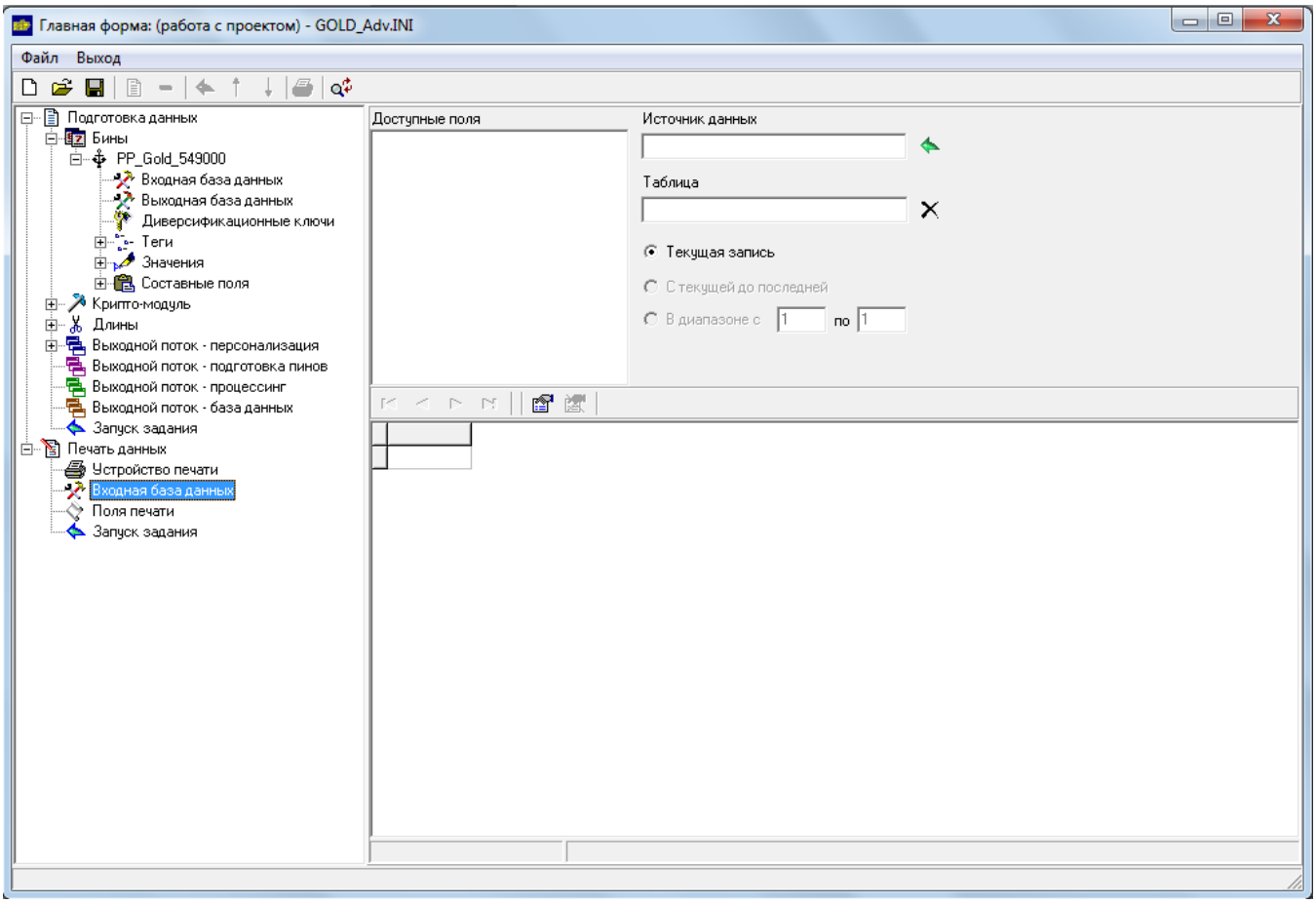
Заголовок используется если перед данными для персонализации необходимо вставить какую-либо константу. Значение константы находится в файле, который нужно будет выбрать.



В случае успешного выполнения задания будет создан или заменен выходной файл, в котором будут находиться подготовленные данные. В случае возникновения ошибки в процессе подготовки данных, о ней будет сообщено в статусной строке.

5.2. Печать данных

5.2.1. Входная база данных

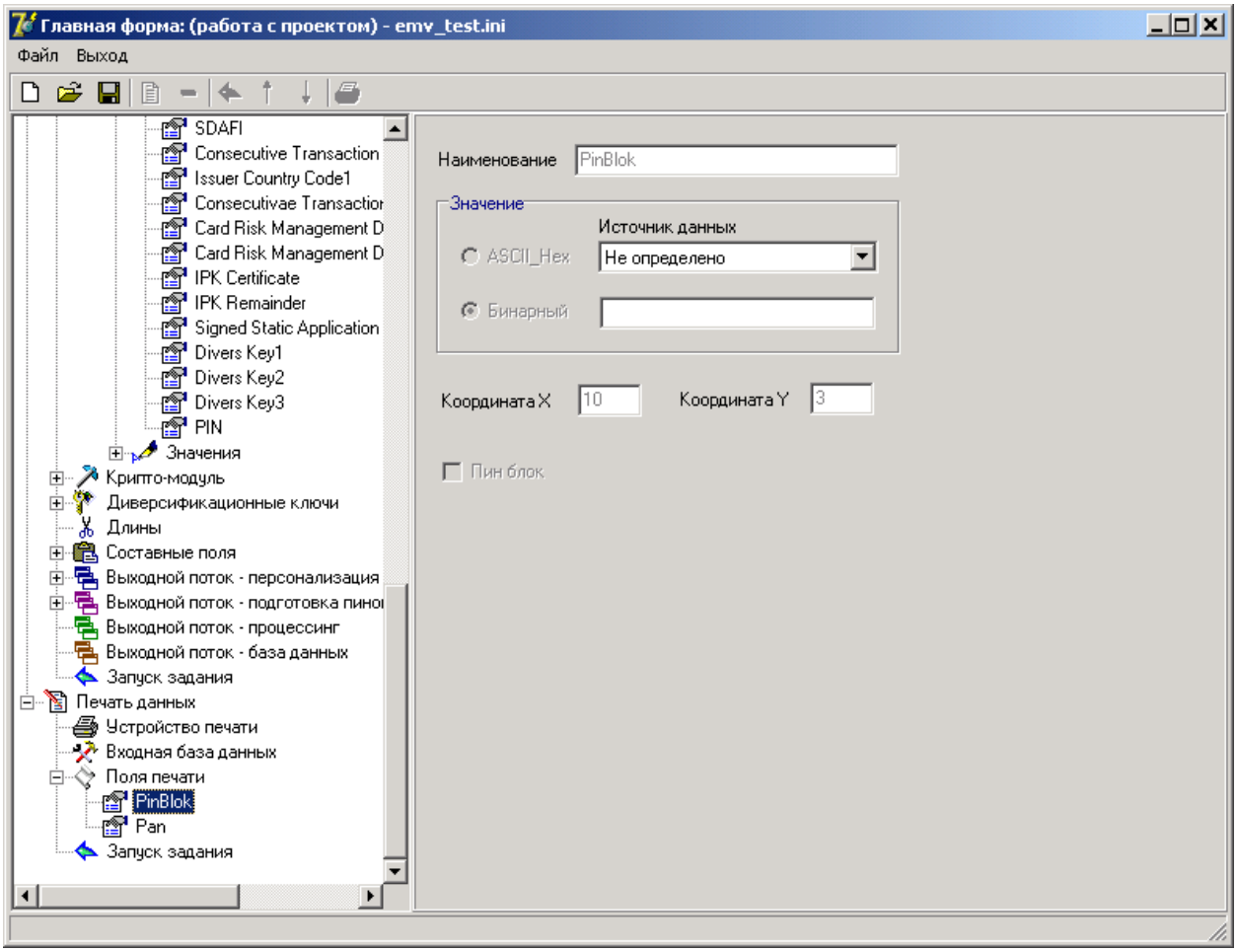


Если для печати данных требуются значения, хранящиеся в базе данных, то необходимо произвести настройки входной базы данных. Настройки аналогичны настройкам, описанным в пункте 5.1.2. настоящего Руководства («Входная база данных»).

5.2.2. Поля печати

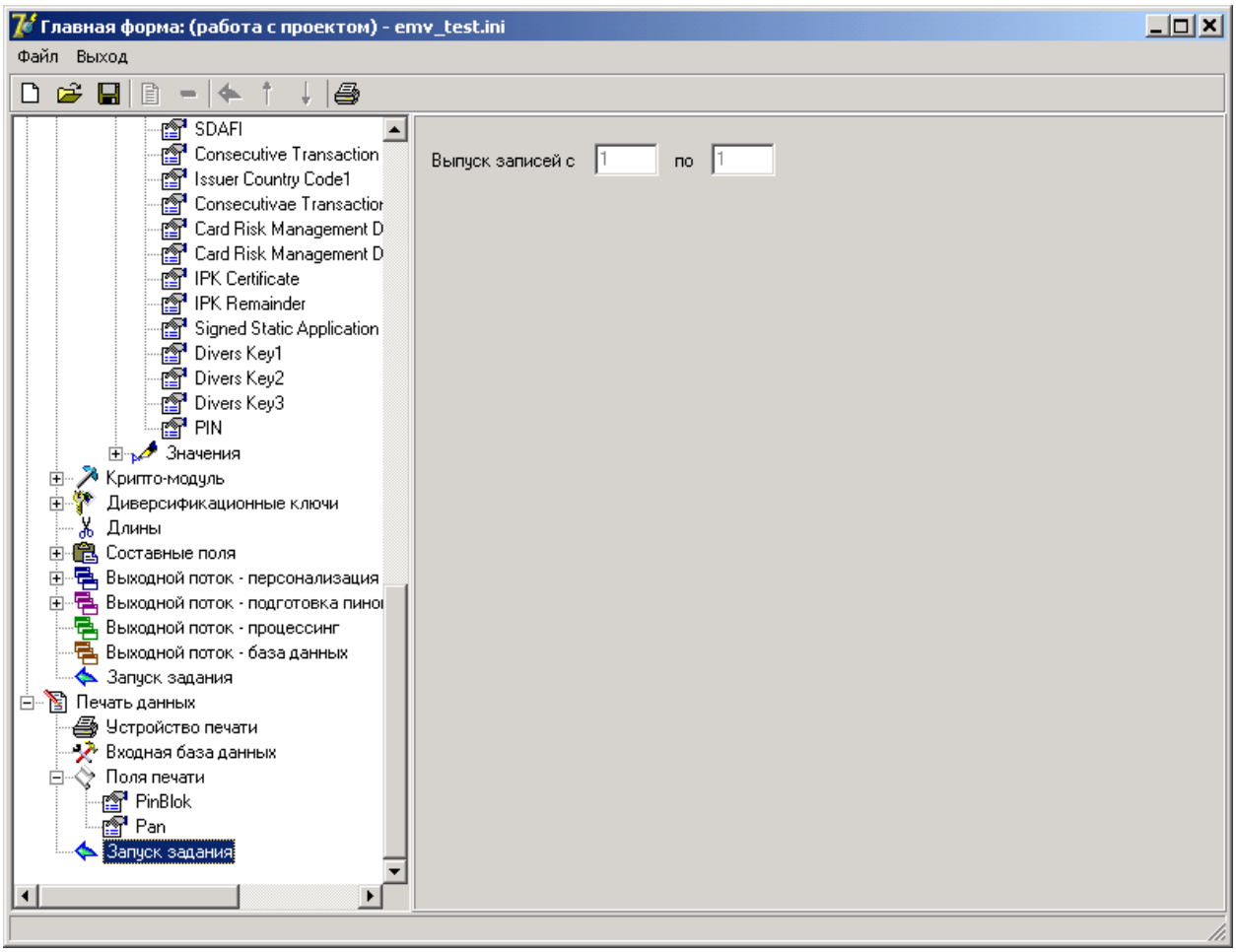
На данной вкладке настраивается источник данных для каждого элемента. Существует несколько источников откуда могут приходить данные.

Клавиатура – входное значение вводится с клавиатуры.
Из базы данных – выбирается значение из входной базы данных, данное поле отображается в виде имя таблицы > имя поля.



5.2.3. Запуск задания

Служит для запуска задания на печать данных на ПИН-конвертах. Количество записей задается в настройке узла входной базы данных.



В случае успешного выполения задания будет производится печать данных. В случае возникновения ошибки в процессе печати данных, о ней будет сообщено в статусной строке.

6. Словарь терминов

Шаблон – используется для настройких модулей, которые являются общими в процессе подготовки данных и не зависят от входных данных.
Проект – используется для настройких модулей, которые изменяются в процессе подготовки данных и зависят от входных данных.
Задание – задание на печать или подготовку данных, зависящее от входных и выходных данных.
Модуль – узел дерева, отображающийся в левой части настройки шаблона или проекта. Каждый модуль содержит настройку, необходимую в процессе подготовки данных.
ПИН-конверт – специальный конверт, имеющий несколько защитных слоев, используемый для печати данных на матричном принтере. Данные печатаются на защитном слое и не видны оператору.
Бин – банковский идентификационный номер.
Эмитент – организация (банк), выпускающая карточный продукт.

Приложение 1. Формирование входных данных и создание источников ODBC

1. Формирование таблицы входных данных и её формат

Входные данные, используемые программой CDP для их обработки, должны состоять из последовательности полей определённого формата. Поля, содержащие значения данных, могут быть фиксированной или переменной длины.

Программа CDP при интерпретации данных, расположенных в файле таблицы входных данных, ориентируется на формат, определённый специальным файлом **schema.ini** текстового формата. Место расположения файла **schema.ini** может быть произвольным на любом локальном диске компьютера вместе с файлом таблицы входных данных.

Программа CDP осуществляет обработку входных данных с использованием файла **schema.ini** в двух вариантах:

1. при подготовке данных персонализации;
2. при печати данных.

В обоих вариантах может применяться либо свой отдельный файл **schema.ini**, либо единый (см. пункт 4 настоящего Приложения).

В файле **schema.ini** указываются:

а) имя файла таблицы входных данных;

б) признак того, фиксированной или переменной длины поля данных. В случае использования полей переменной длины указывается символ разделителя полей;

в) имена полей данных;

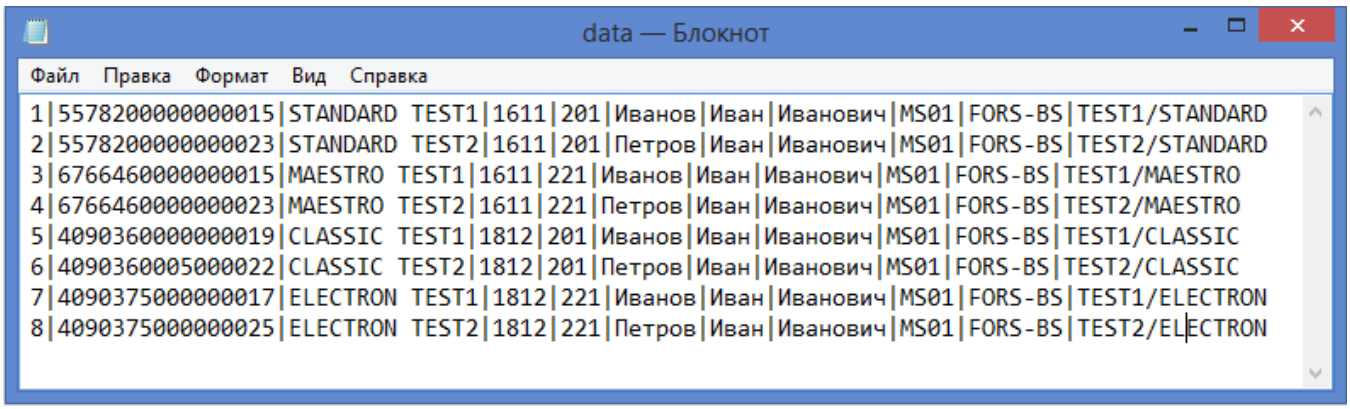
г) длина каждого поля в случае использования полей фиксированной длины. В случае использования полей переменной длины указывается максимально возможное значение длины поля.

2. Входная база данных для подготовки данных персонализации. Примеры.

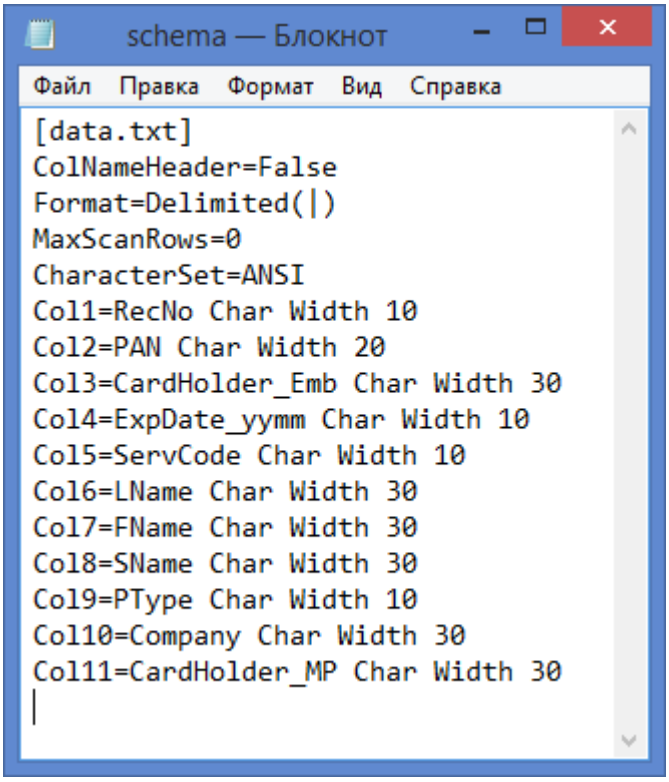
2.1. Пример входных данных с полями переменной длины.

При применении файлов входных данных с полями переменной длины, в файле **schema.ini** параметр **Format** должен иметь значение **Delimited** с указанием символа разделителя полей (например, **#** - решётка, **;** - точка с запятой, **|** - вертикальная полоса, и т.п.). Например, «**Format=Delimited(|)**».

Пример файла таблицы входных данных – **data.txt** (имя файла может быть произвольным):



Пример файла **schema.ini**, описывающего формат вышеуказанной таблицы входных данных:



В данном примере содержимое файла **schema.ini** начинается с указания имени файла таблицы входных данных **data.txt**.

В качестве разделителя полей в строке **«Format=Delimited»** указан символ вертикальной полосы.

Описание каждого поля данных располагается в строке, начинающейся с **«Col»**. В данных строках указывается произвольное имя поля (обычно отражающее его смысловое содержимое), и предельно возможная длина поля. В приведённом здесь примере строка **«Col4=ExpDate_yymm Char Width 10»** сообщает о том, что четвёртое по порядку поле имеет имя **ExpDate_yymm** и может содержать данные длиной не более 10 символов. Это поле содержит данные о сроке действия карты.

Допустим, в нашем примере файлы **schema.ini** и **data.txt** расположены в папке **c:\Ostcard\CDP\Input**. В этом случае необходимо создать и настроить источник данных ODBC с применением драйвера **Microsoft Text Driver** и указанием каталога расположения базы данных **c:\Ostcard\CDP\Input**.

В случае использования в качестве входных данных файлов других форматов, необходимо при настройке источника ODBC применить соответствующий драйвер (**Microsoft Access Driver**, **Microsoft dBase Driver**, **Microsoft Excel Driver**, **SQL Server** или **Microsoft ODBC for Oracle**).

После создания источника данных ODBC необходимо осуществить привязку проекта подготовки данных к описанному источнику данных ODBC в соответствии с пунктом 5.1.2 настоящего Руководства.

Далее, в соответствии с пунктом 5.1.5 настоящего Руководства, необходимо осуществить привязку определённых элементов (Тэгов и/или Значений) к требуемым полям из входной базы данных.

При внесении изменений в состав полей входных данных осуществляется соответствующая корректировка файла **schema.ini**. При необходимости, производится изменение в составе элементов (Тэгов и/или Значений Шаблона) (пункты 4.3. и 4.4. настоящего Руководства).

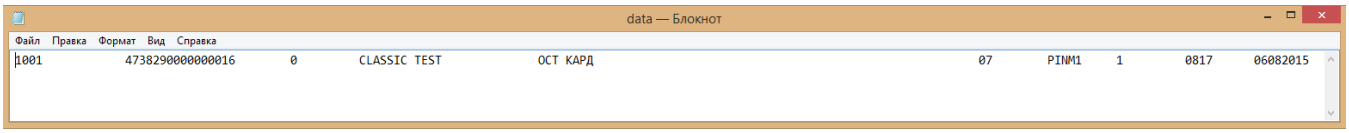
Далее снова осуществляется привязка проекта подготовки данных к источнику данных ODBC в соответствии с пунктом 5.1.2 настоящего Руководства.

Затем снова в соответствии с пунктом 5.1.5 настоящего Руководства, необходимо осуществить привязку определённых элементов (Тэгов и/или Значений) к требуемым полям из входной базы данных.

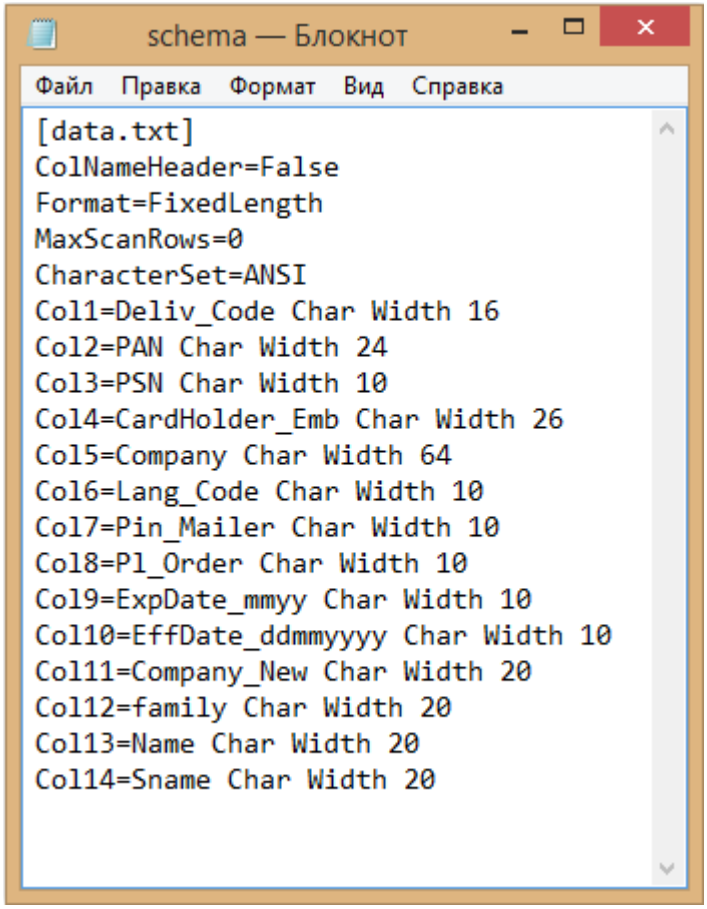
2.2. Пример входных данных с полями фиксированной длины.

При применении файлов входных данных с полями фиксированной длины, в файле **schema.ini** параметр **Format** должен иметь значение **FixedLength**.

Пример файла таблицы входных данных – **data.txt** (имя файла может быть произвольным):



Пример файла **schema.ini**, описывающего формат вышеуказанной таблицы входных данных:



В данном примере содержимое файла **schema.ini** начинается с указания имени файла таблицы входных данных **data.txt**.

Описание каждого поля данных располагается в строке, начинающейся с **«Col»**. В данных строках указывается произвольное имя поля (обычно отражающее его смысловое содержимое), и жёстко фиксированное значение длины поля. В приведённом здесь примере строка **«Col9=ExpDate_mmyy Char Width 10»** сообщает о том, что десятое по порядку поле имеет имя **ExpDate_mmyy** и содержит данные длиной строго 10 символов. Это поле содержит данные о сроке действия карты. Содержимое полей, не несущее полезную информацию заполняется пробелами.

Допустим, в нашем примере файлы **schema.ini** и **data.txt** расположены в папке **c:\Ostcard\CDP\Input**. В этом случае необходимо создать и настроить источник данных ODBC с применением драйвера **Microsoft Text Driver** и указанием каталога расположения базы данных **c:\Ostcard\CDP\Input**.

В случае использования в качестве входных данных файлов других форматов, необходимо при настройке источника ODBC применить соответствующий драйвер (**Microsoft Access Driver**, **Microsoft dBase Driver**, **Microsoft Excel Driver**, **SQL Server** или **Microsoft ODBC for Oracle**).

После создания источника данных ODBC необходимо осуществить привязку проекта подготовки данных к описанному источнику данных ODBC в соответствии с пунктом 5.1.2 настоящего Руководства.

Далее, в соответствии с пунктом 5.1.5 настоящего Руководства, необходимо осуществить привязку определённых элементов (Тэгов и/или Значений) к требуемым полям из входной базы данных.

При внесении изменений в состав полей входных данных осуществляется соответствующая корректировка файла **schema.ini**. При необходимости, производится изменение в составе элементов (Тэгов и/или Значений Шаблона) (пункты 4.3. и 4.4. настоящего Руководства).

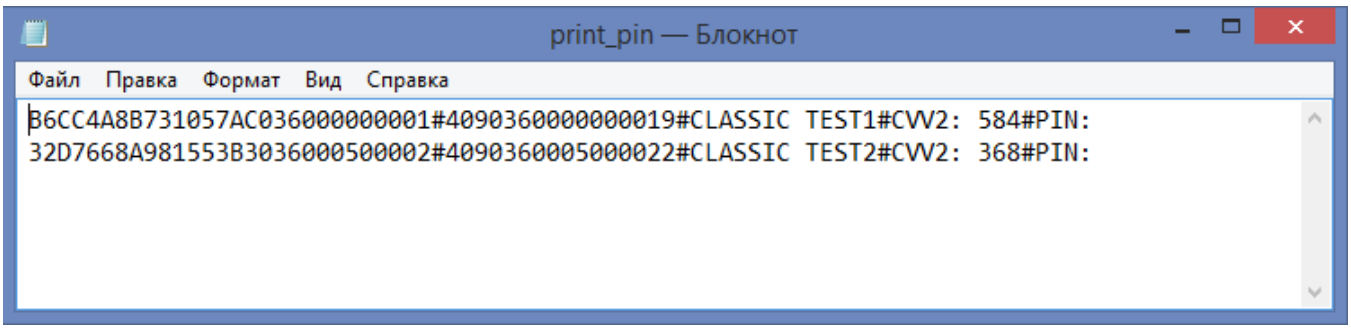
Далее снова осуществляется привязка проекта подготовки данных к источнику данных ODBC в соответствии с пунктом 5.1.2 настоящего Руководства.

Затем снова в соответствии с пунктом 5.1.5 настоящего Руководства, необходимо осуществить привязку определённых элементов (Тэгов и/или Значений) к требуемым полям из входной базы данных.

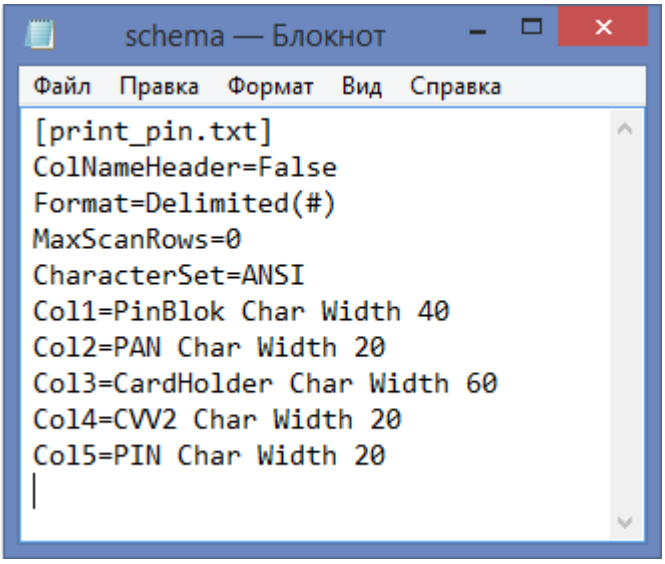
3. Входная база данных для печати данных. Пример.

При применении файлов входных данных для печати данных обычно используются поля переменной длины. В этом случае в файле **schema.ini** параметр **Format** должен иметь значение **Delimited** с указанием символа разделителя полей (например, **#** - решётка, **;** - точка с запятой, **|** - вертикальная полоса, и т.п.). Например, «**Format=Delimited(#)**».

Пример файла таблицы входных данных – **print_pin.txt** (имя файла может быть произвольным):



Пример файла **schema.ini**, описывающего формат вышеуказанной таблицы входных данных:



В данном примере содержимое файла **schema.ini** начинается с указания имени файла таблицы входных данных **print_pin.txt**.

В качестве разделителя полей в строке **«Format=Delimited»** указан символ решётки.

Описание каждого поля данных располагается в строке, начинающейся с **«Col»**. В данных строках указывается произвольное имя поля (обычно отражающее его смысловое содержимое), и предельно возможная длина поля. В приведённом здесь примере строка **«Col2=PAN Char Width 20»** сообщает о том, что второе по порядку поле имеет имя **PAN** и может содержать данные длиной не более 20 символов. Это поле содержит данные о номере карты.

Допустим, в нашем примере файлы **schema.ini** и **print_pin.txt** расположены в папке **c:\Ostcard\CDP\Output\Pin**. В этом случае необходимо создать и настроить источник данных ODBC с применением драйвера **Microsoft Text Driver** и указанием каталога расположения базы данных **c:\Ostcard\CDP\Output\Pin**.

В случае использования в качестве входных данных файлов других форматов, необходимо при настройке источника ODBC применить соответствующий драйвер (**Microsoft Access Driver**, **Microsoft dBase Driver**, **Microsoft Excel Driver**, **SQL Server** или **Microsoft ODBC for Oracle**).

После создания источника данных ODBC необходимо осуществить привязку проекта печати данных к описанному источнику данных ODBC в соответствии с пунктом 5.2.1 настоящего Руководства.

Далее, в соответствии с пунктом 5.2.2 настоящего Руководства, необходимо осуществить привязку определённых полей печати к требуемым полям из входной базы данных.

При внесении изменений в состав полей входных данных осуществляется соответствующая корректировка файла **schema.ini**. При необходимости, производится изменение в составе полей печати (пункт 4.10. настоящего Руководства).

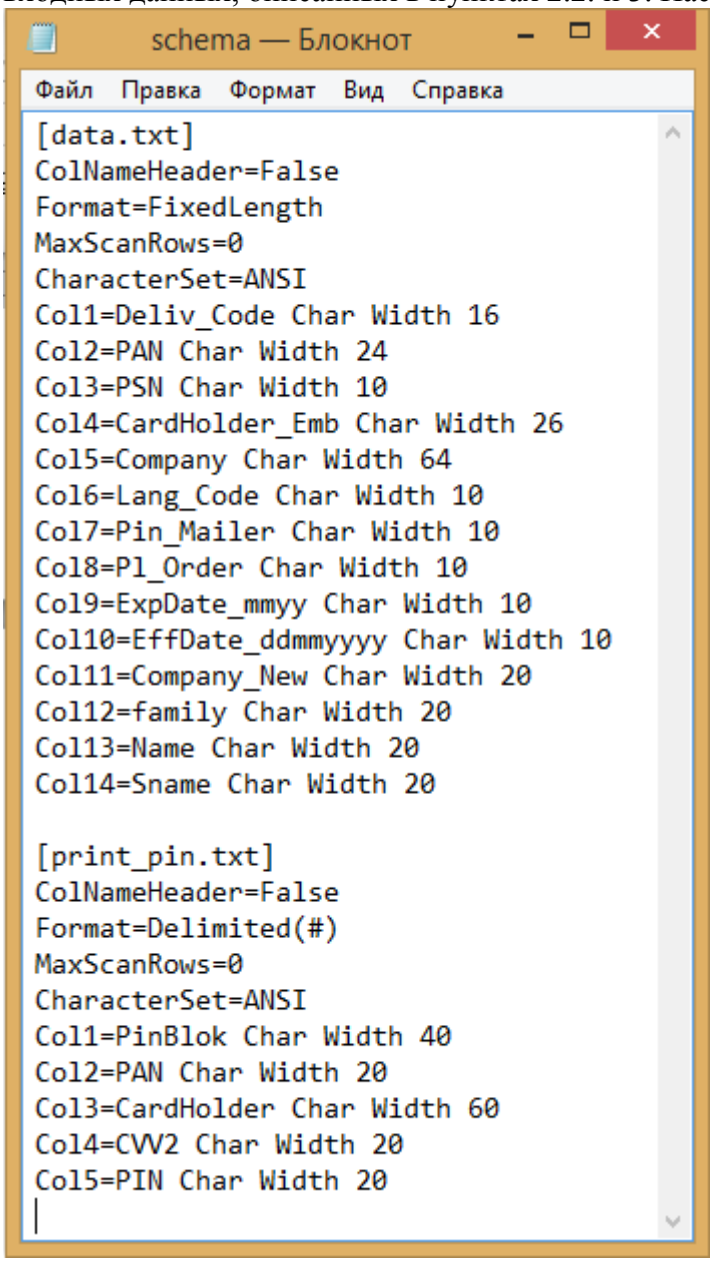
Далее снова осуществляется привязка проекта печати данных к источнику данных ODBC в соответствии с пунктом 5.2.1 настоящего Руководства.

Затем снова в соответствии с пунктом 5.2.2 настоящего Руководства, необходимо осуществить привязку определённых полей печати к требуемым полям из входной базы данных.

4. Единый файл schema.ini для подготовки и печати данных.

При подготовке и печати данных возможно применение единого файла **schema.ini** одновременно для двух и более файлов таблиц входных данных. В этом случае в файле schema.ini описываются форматы таблиц входных данных в отдельных блоках для каждой таблицы.

Ниже приведён пример содержимого файла **schema.ini** одновременно для файлов таблиц входных данных, описанных в пунктах 2.2. и 3. Настоящего Приложения.



При этом файлы **schema.ini**, **data.txt** и **print_pin.txt** должны располагаться в одном каталоге.