Оглавление

[Общее описание 2](#_Toc438222465)

[Установка 4](#_Toc438222466)

[Порядок работы 4](#_Toc438222467)

[Этапы обработки 5](#_Toc438222468)

[Обработка входящих файлов 5](#_Toc438222469)

[Проверка контрольной суммы 5](#_Toc438222470)

[Конвертирование 5](#_Toc438222471)

[Настройки 6](#_Toc438222472)

[Журналирование 9](#_Toc438222473)

# Общее описание

Программа в фоновом режиме осуществляет проверку входящего каталога. При появлении входящих файлов SBRF3 проводит проверку файлов на хэш, в случае успешной проверки, считывает файлы, записывает входящие документы в базу данных, осуществляет проверку на дубли по дате, номеру и сумме и конвертирует файл в формат свободного формата пригодный для загрузки в АБС Банкир.

Общий вид представлен на рис.1

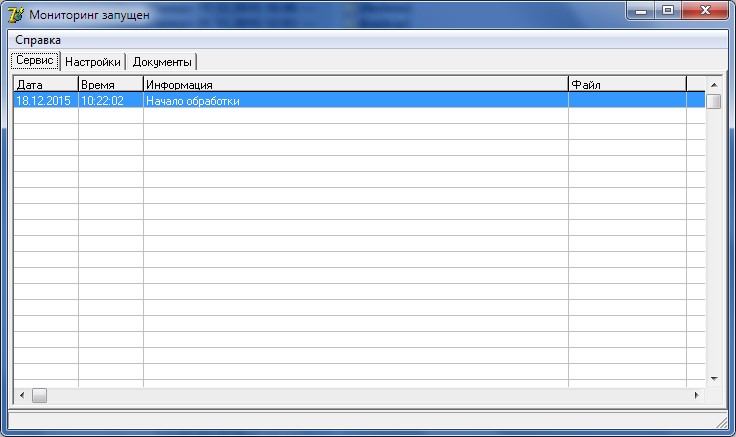


Рис. 1

На закладке сервис представлен ход работы. При прохождении каждого этапа осуществляется вывод соответствующей информации.

На закладке документы, рис. 2., отображаются обработанные и сохраненные документы. Есть возможность фильтровать по датам От и ДО. При нажатии на кнопку «Все документы» отображаются все документы.

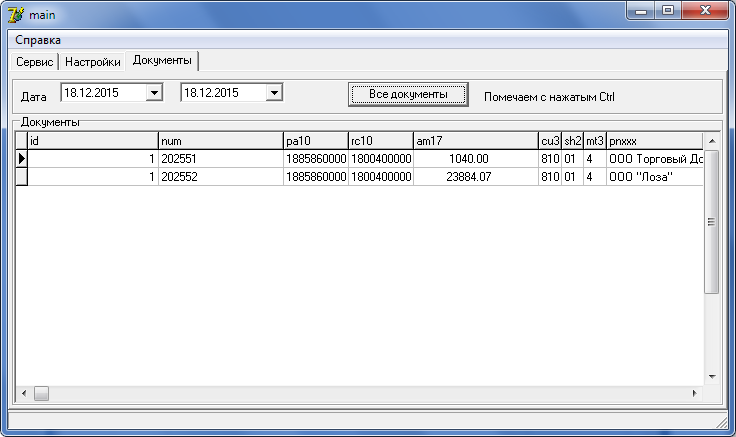


Рис. 2

При нажатии правой кнопки мыши появляется меню для печати документа, рис.3

Есть возможность печати одного текущего документа. Печати всех документов, а также отметить с нажатым Ctrl нужные документы и распечатать только отмеченные документы, выбрав соответствующие пункты. Документы печатаются через MS Word.

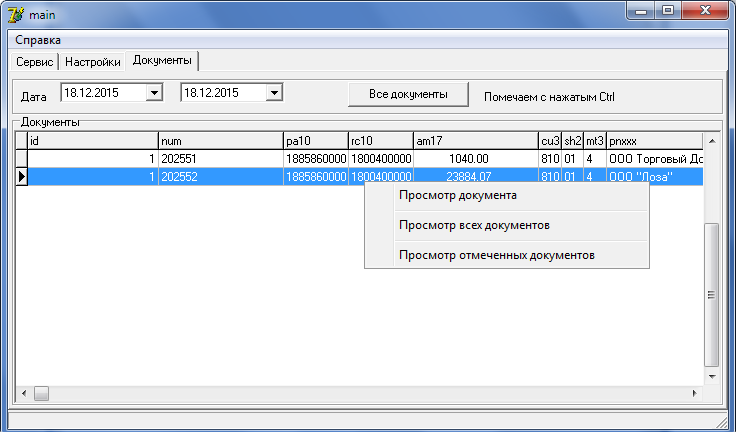


Рис.3

# Установка

Список необходимого ПО:

1. Microsoft OLE DB Provider for Visual FoxPro 9.0 (VFPOLEDBSetup.msi)
2. MS Word 2010 и выше

# Порядок работы

Запускаем файл sbrf3ToBankier.exe

Программа начнет свою работу. Весь функционал работает в отдельном потоке.

# Этапы обработки

## Обработка входящих файлов

На этом этапе сканируется входящий каталог, заданный параметром PATH\_IN\_ на наличие файлов по маске Sb\*.D06 (задается параметром mask\_file) при наличии файлов, попадающих под заданную маску они перемещаются для проверки хэш-суммы в каталог из параметра PATH\_IN\_KEY

## Проверка контрольной суммы

На данном этапе запускается скрипт check.bat, который проверяет все файлы из каталога

PATH\_IN\_KEY, при правильности контрольной суммы файлы перемещаются в каталог заданный параметром PATH\_IN, при ошибочной в каталог err (задается скриптом check.bat ) и исключаются из обработки.

## Конвертирование

В каталоге PATH\_IN проверяется наличие файлов. При наличии файлы считываются, каждый документ проверяется на повторяемость по номеру сумме и дате, сохраняется.

Далее файл конвертируются в файл свободного формата. Сохраняется в каталог заданный параметром PATH\_OUT с именем fr<исходное имя> .txt

Исходный файл перемещается в архив. В архиве все входящие файлы сохраняются по датам, к имени файла добавляется текущее время.

В каталоге PATH\_OUT находятся файлы готовые для загрузки.

# Настройки

Настройки осуществляются в файле sp.ini

[Settings]

PATH\_IN\_= входящий каталог первого этапа

PATH\_IN\_KEY= входящий каталог второго этапа, для проверки хэш

PATH\_IN=входящий каталог третьего этапа

PATH\_OUT=выходной каталог в котором формируются файлы для загрузки в Банкир

INTERVAL= Интервал сканирования каталога, задается в миллисекундах. По умолчанию 10 секунд.

PATH\_CHECK= путь до check.bat

mask\_file= маска файлов

шаблон файла свободного формата, состоит из формул и статических строковых значений, статические значения задаются в кавычках, все формулы перечислены ниже.

SHABLON=LD6#NUMH#'01'#'C'#####'803'#'4'#CU3#AM17#AM17##IN16\_4####RAxx#BCxx#BKxx##BCxx##'30110810000000000849'#BCxx###BCxx##PPxxx+PNxxx###RNxxx#'БАЙКАЛЬСКИЙ БАНК ПАО СБЕРБАНК'#SBxxx#BNxxx###########'e'#RAxx#####'e'#'30110810000000000849'##LD6#######BCxx#SBxxx#BCxx#BNxxx########RIxx#'7707083893'################KPx#KRxd#########ED6###

delimeter= разделитель

PATH\_LOGI= файл с логами

LOGI= //Логи: 0 - выключено, 1 - включено

ARHIV= каталог архива

Формулы для параметра SHABLON

# DT3:string; // Тип бизнес-сообщения

# PA10:string; // Участник-отправитель

# RC10:string; // Участник-получатель

# AM17:string; // Сумма

# DE1:string; // Точность для сумм в полях AM, QA, AX

# CU3:string; // Код валюты РФ (код ISO)

# SH2:string; // Вид операции

# MT3:string; // Тип сообщения

# // банк плательщик Дебет

# PNxxx:string; // Наименование плательщика

# SIxx:string; // ИНН плательщика

# SAxx:string; // Номер счета плательщика

# SBxxx:string; // Наименование банка пла-тельщика

# SNxx:string; // Код банка плательщика бик

# SKxx:string; // Корсчет банка плательщика

# SSx:string; // Расчетная система банка плательщика

# // банк получатель Кредит

# RNxxx:string; // Наименование получателя

# RIxx:string; // ИНН получателя

# RAxx:string; // Номер счета получателя

# BNxxx:string; // Наименование банка получа-теля

# BCxx:string; // Код банка получателя бик

# BKxx:string; // Корсчет банка получателя

# RSx:string; // Расчетная система банка получателя

# PPxxx:string; // Назначение платежа

# LD6:string; // Дата обработки

# IN16:string; // Информация о внесистемном документе

# IN16\_1:string; // префикс

# IN16\_2:string; // " номер клиентского поручения

# IN16\_3:string; // " дата клиентского поручения

# IN16\_4:string; // " очередность платежа

# SC2:string; // Условия перевода

# SD6:string; // ДПП дата

# PT4:string; // Тип обслуживания

# DH6:string; // Cписано со счета плательщи-ка дата

# ED6:string; // Поступило в банк плательщи-ка дата

# KPx:string; // КПП плательщика

# KRx:string; // КПП получателя

Все документы сохраняются в файл base.dbf

# Журналирование

При включенной опции LOGI и PATH\_LOGI ведется подробный журнал работы.

Отражается каждый этап.

Пример:

18.12.2015 11:30:04 Начало обработки

18.12.2015 11:30:42 Входящий SB000017.D06 *🡨 первый этап*

18.12.2015 11:30:46 Проверка хэш *🡨 второй этап*

18.12.2015 11:30:51 Обработан SB000017.D06 *🡨 третий этап*

18.12.2015 11:31:46 Конец обработки