

**ActiveX-компонент для подключения ЭККА
к POS-терминалам Ingenico
ECRCommX**

JSC “Bancomzvjazok”, Secure Transactions Systems Department

Address: Ukraine – 04080, Kyiv, 69 Frunze st.

Tel.: +380444960096

Email: support@bkc.com.ua

WEB: <http://www.bkc.com.ua>

Общая информация.

Компонент реализует кассовый протокол взаимодействия с POS-терминалами компании Ingenico и предназначен для разработки программного обеспечения (ПО) для персонального компьютера (ПК), использующего этот протокол для взаимодействия с POS-терминалом.

Описание интерфейса

Компонент реализует интерфейс *ECRLibSgn*, посредством которого и выполняются все действия.

Запуск транзакций на исполнение, а так же установка начальных параметров выполняется с помощью методов интерфейса *ECRLibSgn*. Методы сразу же возвращают управление вызывающей стороне. Выполнение транзакции происходит в отдельном потоке.

Использование интерфейса

Ниже приведен пример (для 1Cv8):

```
Terminal = Новый СОМОбъект("ECRCommX.ECRLibSgn");
Terminal.setUserTimeout(40);
Terminal.setCommNum(1, 115200);
Terminal.PosPurchase(1000, 0, "");

Пока l=1 Цикл
    если НЕ (Terminal.GetLastResult() = 2) тогда
        Прервать;
    конесесли;
КонецЦикла;

если (Terminal.GetLastResult() = 1) тогда
    Сообщить("Ошибка: " + Terminal.GetLastErrorCode());
    Сообщить(Terminal.GetLastErrorText());
иначе
    Сообщить("Авторизация успешна");
конесесли;
```

При инициировании транзакции класс *ECRLibSgn* создает поток для связи с терминалом и возвращает управление. Для получения результатов выполнения транзакции используется метод *GetLastResult*. Если он возвращает 0, то транзакция прошла успешно, иначе возникла ошибка, код которой можно получить с помощью метода *GetLastErrorCode*, а описание – *GetLastErrorText*. Список кодов ошибок см. ниже.

Методы интерфейса

public void Cancel()

Метод предназначен для прекращения выполнения транзакции в случае продолжительного времени ожидания.

public void PosPurchase(long tot_amount, long add_amount, String HostText)

Метод проводит транзакцию Продажа(Purchase)/Наличные(Cash). tot_amount – общая сумма в копейках (далее вся сумма указывается в копейках), add_amount – дополнительная сумма (скидка/комиссия). HostText – данные для отправки на хост.

public void PosPurchaseMer(long tot_amount, long add_amount, long mer, String HostText)

Метод аналогичный PosPurchase. В дополнение передается номер мерчанта (параметр mer).

public void PosRefund(long tot_amount, long add_amount, String HostText)

Метод проводит транзакцию Возврат(Refund)/Депозит(Deposit). tot_amount – общая сумма, add_amount – дополнительная сумма (комиссия на внесение). HostText – данные для отправки на хост.

public void PosRefundMer(long tot_amount, long add_amount, long mer, String HostText)

Метод аналогичный PosRefund. В дополнение передается номер мерчанта (параметр mer).

public void PosDiscountInit()

Инициирование 2-х проходной продажи со скидкой или наличных с комиссией. Метод иницирует считывание магнитной карты клиента, по считанному треку со стороны кассы определяется размер скидки. Для завершения транзакции вызывается метод PosDiscountCompletion.

public void PosDiscountInitEx()

То же что и **PosDiscountInit**, только терминал вместо данных по карте передает информацию о скидках терминала по данной карте. Формат данных см. ниже функцию getDiscountInfo.

public void PosDiscountInitAccIdSale(StringAccID)

Инициирование 2-х проходной операции «ОПЛАТА» со скидкой. Метод иницирует считывание магнитной карты клиента, по считанному треку со стороны кассы определяется размер скидки. Для завершения транзакции вызывается метод PosDiscountCompletion.

public void PosDiscountInitAccIdCash(StringAccID)

Инициирование 2-х проходной операции «ВЫДАЧА НАЛИЧНЫХ» с комиссией. Метод иницирует считывание магнитной карты клиента, по считанному треку со стороны кассы определяется размер скидки. Для завершения транзакции вызывается метод PosDiscountCompletion.

public void PosDiscountCompletion(long tot_amount, long add_amount, String HostText)

Завершение 2-х проходной продажи со скидкой или наличных с комиссией. tot_amount – общая сумма, add_amount – дополнительная сумма (скидка/комиссия). HostText – данные для отправки на хост. Метод PosDiscountCompletion аналогичен методу PosPurchase, разница состоит в том, что для завершения транзакции используется трек карты, считанный при помощи метода PosDiscountInit.

public void PosDiscountCompletionMer(long tot_amount, long add_amount, String HostText, long mer)

То же что и **PosDiscountCompletion**, только в дополнение передается номер мерчанта (параметр mer) по которому выполнять завершение. Используется в связке с функцией **PosDiscountInitEx**.

public void PosCardread()

Метод инициирует считывание магнитной карты клиента.

public void PosSettle()

Метод предназначен для инициирования операции Z-баланс (выгрузка итогов) на терминале.

public void PosSettleMer(long mer)

Метод аналогичный PosSettle. В дополнение передается номер мерчанта (параметр mer). При mer = 0 выполняется выгрузка по всем мерчантам.

public void PosPrepaid()

Метод предназначен для инициирования транзакции продажи PrePaid.

public void PosVoid(long InvoiceNum)

Метод проводит транзакцию Void(отмена) по заданному номеру чека.

public void PosAudit(String AccID)

Метод проводит аудит по заданному AccID(CARDACCEPTORMERCHANT) После успешного завершения, вызывать метод **public String getAudit()**, для получения аудита.

public void PosGetInfo()

Метод инициирует получение версии ПО, номера терминала, серийного номера терминала. После успешного завершения, вызывать метод **public String getInfoBiplan()**, для получения данной информации.

public void PosSettleAccID (String AccID)

Метод аналогичный PosSettle. В дополнение передается номер мерчанта (8 байт CardAcceptorMerchant).

public void PosBalance()

Иницирует запрос баланса на терминале. Терминал передает значение полей 4 и 63.

public void PosService(long tot_amount, long merchNum, long serviceNum, String ServiceParams)

Метод проводит продажу сервиса на терминале. tot_amount – общая сумма (которая будет отправляться на хост в 4м поле), merchNum –порядковый номер мерчанта, который используется для сервисных операций, serviceNum – уникальный номер сервиса, ServiceParams – строка, содержащая номинал сервиса и набор параметров.

Формат передачи номинала и параметров: Номинал/Параметр1/Параметр2/ /ПараметрN. В качестве разделителя номинал-параметры и параметр-параметр используется символ "/" (0x2F).

public void PosService(String RRN)

Метод проводит установку RRN для последующего использования с операцией Refund (возврат товара).

public long getLastResult()

Возвращает результат выполнения последней транзакции. Возможные значения:

2 – в процессе выполнения.

0 – транзакция успешно завершена

1 – ошибка выполнения транзакции.

public long getLastErrorCode()

Возвращает код ошибки в случае если getLastResult вернула 1. Возможные значения:

1 – ошибка связи с роутером.

2 – ошибка передачи данных между роутером и компонентой

3 – ошибка передачи между роутером и терминалом

4 – транзакция завершена не успешно, т.е. getRspCode не равен 0.

public long getLastErrorText()

Возвращает текстовое описание ошибки в случае если getLastResult вернула 1.

Свойства интерфейса

public void setUserTimeOut(long newTimeOut)

Устанавливает новый таймаут ожидания (в секундах) данных от терминала. По умолчанию равен 40с.

public void setSocketNum(long socket)

Устанавливает номер сокета для соединения с роутером JavaPOS. По умолчанию равен 4779.

STDMETHOD setCommNum(BYTE bPort, LONG lBaudRate)

После вызова этого метода ECRLibSgn переключается на работу с терминалом без использования JAVAPOS FourInOne, то есть сообщения будут отправляться напрямую на терминал (BYTE bPort, LONG lBaudRate).

В таком режиме работы поддерживаются следующие операции:

- setUserTimeout
- setRequirePIN
- Cancel
- PosPurchase
- PosPurchaseMer
- PosRefund
- PosRefundMer
- PosXBalance
- PosXBalanceMer
- PosCardread
- PosSettle
- PosSettleMer
- PosVoid
- PosGetInfo
- PosService
- PosDiscountInitEx
- PosDiscountCompletion
- PosDiscountCompletionMer
- Все get<> методы (такие как getPAN, getCardExpY, getCardExpM...)

Все дальнейшие свойства использовать после того как отработал поток работы с терминалом, т.е. (getLastResult() != 2)

public long isTransactionComplete()

Возвращает признак завершения транзакции. Имеется в виду успешность сеанса передачи между библиотекой и терминалом. Успешность самой транзакции смотрится с помощью getRspCode()

public long getTotalAmount()

Возвращает сумму сделки (сумма поля 4)

public long getAddAmount()

Возвращает дополнительную сумму сделки (сумма поля 54)

public String getField63()

Возвращает поле 63 ответа хоста

public String getTrack1()

Возвращает Track1 карты

public String getTrack2()

Возвращает Track2 карты

public String getTrack3()

Возвращает Track3 карты

public long getRspCode()

Возвращает Response Code транзакции

public String getPAN()

Возвращает PAN карты

public long getCardExpY()

Возвращает год Expiration Date карты

public long getCardExpM()

Возвращает месяц Expiration Date карты

public String getInvoiceNum()

Возвращает номер инвойса транзакции

public String getAppCode()

Возвращает код авторизации транзакции

public long getTranDateD()

Возвращает день даты транзакции

public long getTranDateM()

Возвращает месяц даты транзакции

public long getTranTimeH()

Возвращает час времени транзакции

public long getTranTimeM()

Возвращает минуту времени транзакции

public String getIssName()

Возвращает Issuer Name карты

public String getMerchantNum()

Возвращает номер мерчанта

public String getProcCode()

Возвращает Processing Code транзакции

public String getPOSEntMode()

Возвращает POS Entry Mode транзакции

public String getPOSCondCode()

Возвращает POS Condition Code транзакции

public String getRRN()

Возвращает RRN транзакции

public String getHolderName()

Возвращает имя владельца карты

public String getTerminalId()

Возвращает ID POS-терминала

public String getDiscountName()

Возвращает имя бонусной карты

public long getDiscountAttribute()

Возвращает атрибут (признак) бонусной карты

public String getAudit()

Полученная строка имеет формат:

NAME	LENGTH	VALUE	DESCRIPTION
type_error	1	30	успешная передача, все данные отгружены
		31	успешная передача, все данные не отгружены
curr_num_tran	4		Порядковый номер передаваемой транзакции в ASCII
all_num_tran	4		Всего передаваемых транзакций в ASCII
tran_inv_num	6		Номер чека в ASCII
pan	20		Пан карты в ASCII
approval_code	6		Код авторизации в ASCII
total_amount	12		сумма в ASCII
total_amount2	12		Сумма с процентом в ASCII
type_operation	1		Типооперации «S» - Sale, "C" - cash, "R" - refund, "V" – void в ASCII
tran_data_year	2		год в ASCII
tran_date	4		Месяц/число в ASCII
tran_time	6		Часы/минуты/секунды в ASCII

public String getInfoBiplan()

Полученная строка имеет формат:

NAME	LENGTH	VALUE	DESCRIPTION
version	10		Версия ПО в ASCII
number_terminal	9		Номер терминала в ASCII
serial_number_terminal	12		Серийный номер терминала в ASCII

public String getBalanceAndField63()

NAME	LENGTH	VALUE	DESCRIPTION
Field4	12		Баланс ASCII
Field63	0...999		Поле 63 (формат не изменяем)

public String getDiscountInfo()

Полученная строка состоит из полей вида:

NAME	LENGTH	VALUE	DESCRIPTION
merchant	1..2		Номер мерчанта
;	1		разделитель
merchant_name	1..2		Имя мерчанта
;	1		разделитель
discount_rate			Процент скидки в сотых долях. Т.е. 350 – это 3,5%
;	1		разделитель