



Программно-аппаратный комплекс Lan4POS

Описание функциональных характеристик

ООО «ЛАНТЕР»

2021

Содержание

1. ВВЕДЕНИЕ.....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
2. ПОДДЕРЖИВАЕМЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ.	3
3. ЛОГИКА РАБОТЫ С LAN4POS	3
4. ОПИСАНИЕ ИНТЕРФЕЙСОВ.....	4
4.1. ITRANSDATA – ИНТЕРФЕЙС ДАННЫХ ТРАНЗАКЦИИ.....	4
4.2. ITRANSENTER – ИНТЕРФЕЙС ВЫПОЛНЕНИЯ ТРАНЗАКЦИИ.....	6
4.3. ITRANSENTEREVENTS – ИНТЕРФЕЙС СОБЫТИЙ ТРАНЗАКЦИИ.....	8

1. Введение

Lan4POS – это платежный модуль, предназначенный для интеграции с внешними приложениями (в основном, на устройствах самообслуживания) с целью надления их функциональностью проведения безналичных платежей с использованием банковских карт.

Lan4POS поддерживает выполнение финансовых транзакций по дебетовым/кредитным картам:

- Банковские карты с магнитной полосой
- Банковские микропроцессорные карты
- Банковские микропроцессорные карты с технологией бесконтактных платежей
- Специализированные бесконтактные карты

Взаимодействие с внешним приложением Lan4POS осуществляет через COM-объект - LanCOM. Он обеспечивает единый интерфейс для выполнения банковских операций, независимо от используемого хостового протокола и установленного оборудования на устройствах самообслуживания.

Этот документ описывает функции интерфейсы COM-объекта, которые должна использовать программа для выполнения финансовых операций с использованием банковских карт.

2. Поддерживаемые операционные системы:

Lan4POS работает в среде 32-битных операционных систем Microsoft: Windows XP, 2003, Windows 7, но может быть использован и в других ОС Microsoft (Windows Vista, Windows 2008, а также в 64-битных версиях ОС, включая Windows 10).

Есть примеры использования LanCOM под Windows Embedded и под Linux (x86 с Wine)

3. Логика работы с Lan4POS:

Логика работы построена по схеме – ‘запрос-ответ’. Перед выполнением запроса на выполнение операции, необходимо заполнить неким набором значений (данные зависят от выполняемой операции) объект данных Request, передать этот объект данных в интерфейс методов, выполняющий запрос и после выполнения операции Lan4POS, считать результат выполнения операции в объект данных Response.

Lan4POS поддерживает две схемы работы:

1. Совмещение запроса-ответа в одном методе (метод Exchange ()). Используется в режиме ‘виртуального’ POS-терминала (отсутствуют устройства чтения карт и ввода пин-кода)
2. Режим работы с событиями. Для запроса и получения ответа, используются отдельные методы (методы StartAsyncExchange() и EndAsyncExchange()) . После отправки запроса, приложение может получать события, через интерфейс событий, по мере прохождения этапов транзакции. Это позволяет организовать пользовательский интерфейс (GUI) с клиентом. Этот режим предназначен для использования на устройствах самообслуживания.

4. Описание интерфейсов:

4.1. *ITransData – Интерфейс данных транзакции*

Свойства интерфейса:

Name	Description	Type/Size (S-string, N - numeric)
Amount	Сумма операции (в копейках, центах и т.п.)	S/12
CurrencyCode	Код валюты операции	S/3
DateTimeHost	Дата и время операции на хосте, (YYYYMMDDHHMMSS)	S/14
CardEntryMode	Режим работы с картами	N/1
PAN	Primary Account Number	S/13..19
CardExpiryDate	Срок действия карты (YYMM)	S/4
AuthorizationCode	Код авторизации	S/8
ReferenceNumber	Код ссылки (RRN)	S/12
ResponseCodeHost	Код ответа хоста	S/3
TextResponse	Дополнительные данные ответа	S/99
DateTimeCRM	Дата и время операции на устройстве, (YYYYMMDDHHMMSS)	S/14
Operation Code	Код выполняемой операции	N/2
OrigOperation	Код первичной операции	N/2
TrxID	Номер транзакции на устройстве	N/6
TrxIDHost	Номер транзакции на хосте	N/6
PathParameters	Доп. данные для печати	S/128
ReceiptData	Данные для печати на чеке	S/128
AmountFee	Сумма комиссии для операции (в копейках, центах и т.п.)	S/12
LogicalTermID	Логический номер терминала (для поддержки мульти-терминальности)	N/8
TerminalOutID	Номер устройства на хосте	S/15
Status	Статус выполнения операции	N/3
Commodity Code	Код платежа	S/255
PaymentDetails	Детали платежа	S/255
ProviderCode	Код провайдера	S/255
EncPinBlock	Зарезервированное поле	S/16

OperationCode – значение этого поля, определяет тип выполняемой операции (финансовой или служебной) при вызове методов интерфейса TransSender.Exchange (...) и TransSender.StartAsyncExchange(...). Значения должны быть из следующего диапазона

Финансовые операции:

Значение	Описание
1	Оплата товаров и услуг
3	Пополнение счета (Кредит)
4	Отмена первичной операции
13	Баланс
14	Баланс (безпиновый)
29	Возврат
61	Платеж (коммунальный)
62	Платеж в сторону 3-х лиц
71	Пополнение счета (Cash to Card)
74	Мини-выписка

Служебные операции:

Значение	Описание
26	Проверка связи с хостом
59	Сверка итогов (закрытие дня)
60	Просмотр журнала транзакций
70	Смена пин-кода карты
75	Загрузка ключей с хоста
72	Установка пин-кода на карту
80	Загрузка параметров с TMS для Lan4POS
85	Проверка статуса подключенных устройств
86	Считывание номера (UID) бесконтактной карты
87	Работа с контактной картой по шагам
88	Создать электронный счет
89	Считывание номера парковочной (Mifare) карты
90	Считывание карты, для дальнейшего проведения операции 'Оплата'

Значение поля **Card Entry Mode** определяет (в запросе) как должна выполняться работа с картами. В ответе, поле несет информацию о том, как реально была задействована карта. Поддерживается следующий диапазон значений:

Значение	Описание
0	Любой вариант предъявления карты (общий случай) через любой доступный ридер карт
1	Ручной ввод номера карты (без использования ридера карт)
2	Карта с магнитной полосой через ридер карт с магнитной полосой
3	Магнитная или чиповая карта через любой доступный ридер контактных карт
4	Чиповая карта через ридер карт с магнитной полосой (EMV FALLBACK)
5	Чиповая карта через ридер чиповых карт
6	Без карты (наличные)
7	Бесконтактная банковская карта через ридер бесконтактных карт
8	Бесконтактная (MIFARE) карта через ридер бесконтактных карт
9	Специальный режим для работы с контактными картами

Поле **Status** - поле (в ответе), которое показывает результат выполнения операции (финансовой или служебной).

Успешное выполнение операции определяется только двумя значениями данного поля: “1” или “17”, все другие значения должны трактоваться как неуспех выполнения операции!

Возможные значения кодов поля Status:

Значение	Описание
0	Неопределенный статус
1	<u>ОДОБРЕНО</u>
3	Необходимо использовать чип (при отработке с бесконтактной картой)
4	Сравниваемые значения не равны (при установке пин-кода)
5	Неподдерживаемая карта
6	Недостаточно средств
7	Карта просрочена
8	Ошибка обработки карты (возможен повтор)
9	Устройство недоступно
10	Ошибка настройки устройства
11	Отсутствие файла(ов) параметров
12	Неизвестное (неверное) значение параметра
13	Прерывание операции по ошибке
14	Неизвестный номер счета
16	Отказано
17	<u>ОДОБРЕНО В OFFLINE</u>
34	Нет связи с хостом
53	Операция прервана (пользователем)
54	Таймаут выполнения операции

Методы интерфейса: Отсутствуют

4.2. *ITransSender* – Интерфейс выполнения транзакции:

Свойства (только для чтения):

BSTR ErrorDescription – текстовое описание ошибки, которая произошла при последнем вызове методов интерфейса (не путать с ошибками при прохождении транзакций)

int ErrorCode - код ошибки, которая произошла при последнем вызове методов интерфейса

значение 0 – означает, что не было ошибки

Методы:

HRESULT InitResources([out, retval] long * Value) – Инициализация ресурсов объекта

HRESULT FreeResources (void) – Освобождение ресурсов, занятых объектом

HRESULT SetChannelParam([in] BSTR IPAddress, [in] long IPPort, [in] BSTR X25Script, [out, retval] long * Value) – установка коммуникационных параметров (TCP-Address/Port) для связи с Lan4POS (Lan4POS может работать на другом компьютере и в другой подсетке).

ООО «ЛАНТЕР», Служба тех. поддержки: тел: +7(495) 967 6650,

e-mail: support_POS@lanit.ru

HRESULT Exchange([in] ITransData ** Request, [in, out] ITransData ** Response, [in] long Timeout, [out, retval] long * Value) – синхронный метод для проведения операции. Возврат из метода происходит по окончании операции.

Параметры:

Request – указатель на объект данных, заполненный соответствующим образом для выполнения операции

Response – указатель на объект данных, заполняемый по результату выполнения Lan4POS -ом операции

Timeout – максимальное время (в секундах) для выполнения операции (метода).

HRESULT StartAsyncExchange([in] ITransData ** Request, [in] long Timeout, [in] long TimeoutForPresentCard, [out, retval] long * Value) – метод для запуска выполнения операции.

Возврат из метода происходит немедленно. Отслеживание шагов прохождения операции происходит через интерфейс событий. Для получения результата выполнения операции, необходимо вызвать метод EndAsyncExchange(...).

Параметры:

Request – указатель на объект данных, заполненный соответствующим образом для выполнения операции

Timeout – максимальное время (в секундах) для выполнения операции (не метода)

TimeoutForPresentCard – максимальное время ожидания (в секундах) для предъявления карты пользователем.

Логика параметров - таймаут ожидания карты и таймаут выполнения операции такова:

- Если за время **TimeoutForPresentCard** карта не будет предъявлена, выполнение операции закончится с ошибкой таймаута выполнения (Status=54)
- Если карта будет предъявлена в промежуток **TimeoutForPresentCard**, тогда с этого момента начинается отсчет времени выполнения операции **Timeout**. Если операция не сможет выполняться за **Timeout** секунд, тогда также будет ошибка таймаута выполнения (Status=54)
- В самом плохом случае – операция будет завершена по таймауту выполнения через **TimeoutForPresentCard + Timeout** секунд.

HRESULT EndAsyncExchange([in, out] ITransData ** Response, [out, retval] long * Value) – метод для получения результата выполнения операции, начатой с помощью StartAsyncExchange(...). Этот метод вернет 0 (no error) и заполнит объект **Response** правильными значениями, только если он будет вызван после получения события – ‘Transaction FINISH’ в методе интерфейса событий - **ITransSenderEvents.OnExchange(...)**.

Параметры:

Response – указатель на объект данных, заполняемый результатом выполнения операции

HRESULT CancelExchange([out, retval] long * Value) – метод для прерывания выполнения операции, запущенной вызовом метода StartAsyncExchange(). После вызова **CancelExchange**

нет необходимости в вызове EndAsyncExchange(..). События прохождения операции, после CancelExchange, перестают поступать. Перед запуском новой операции с помощью StartAsyncExchange(), необходимо выдержать паузу в 2-3 секунды.

4.3. ItransSenderEvents – Интерфейс событий транзакции:

Свойства: отсутствуют

Методы:

void OnExchange([in] long InfoStatus, [in] BSTR StatusDescription) – этот метод может вызываться многократно, по мере выполнения операции (реализация метода должна быть сделана в программе)

Параметры:

InfoStatus – Код события

StatusDescription – текстовое описание события (или дополнительные данные события).

Возможные значения для кода InfoStatus (последовательность событий случайная, гарантируется только, что порядок событий будет такой же, как он возникает на стороне Lan4POS):

Значение	Описание
101	Готовность к выполнению транзакции
102	Ожидание предъявления карты
103	Чтение карты
104	Считывание карты успешно
105	Ожидание ввода пин-кода (в StatusDescription – число оставшихся попыток (для offline pin) или ‘’(пусто) для online pin)
106	Нажата кнопка на пин-клавиатуре
107	Нажата кнопка ‘Clear’(корректировка) на пин-клавиатуре (очистка всех введенных нажатий) (см. также 138 событие)
108	Закончен ввод пин-кода
109	Ввод пин-кода отменен
110	Установка соединения с хостом
111	Соединение с хостом установлено
112	Нет соединения с хостом
113	Отправка запроса на хост
114	Ожидание ответа с хоста
115	Обработка ответа хоста
116	Выполнение автоматической отмены
117	Проверка готовности устройств
118	Отсутствие параметров

119	Ошибка ридера бесконтактных карт
120	Ошибка ридера контактных карт
121	Ошибка дополнительного SAM-ридера шифрования
122	Ошибка основного SAM-ридера шифрования
123	Ошибка дополнительной SAM-карты шифрования
124	Ошибка основной SAM-карты шифрования
125	Ошибка пин-клавиатуры
....	Зарезервированные значения
129	Ошибка считывания карты
130	Карта не может быть обслужена
131	Ожидание ввода нового пин-кода
132	Ожидание повторного ввода пин-кода
133	Пустой стекер карт (для card dispenser)
134	Переполнено отделение для сброшенных карт (для cad dispenser)
135	Ожидание изъятия карты из ридера
136	Произошел захват карты
137	Информация о транзакции в журнале
138	Нажата кнопка 'Clear' (корректировка) на пин-клавиатуре (очистка только последней введенной цифры) (см. также 107 событие)
....	Зарезервированные значения
163	Transaction timeout ERROR
164	Transaction FINISH

После получения события с кодом 164, необходимо вызвать метод EndAsyncExchange(...) для получения результата выполнения операции.

При выполнении любой **финансовой** операции, обязательным к заполнению в запросе, является следующий набор свойств интерфейса данных:

- OperationCode
- CardEntryMode
- Amount
- CurrencyCode

Требования к заполнению других свойств в запросе может зависеть от выполняемой финансовой операции:

(Ниже в таблицах: М – Mandatory Field; О – Optional Field; С – Conditional Field (Mandatory for Card Entry Mode = 1))

Оплата (“OperationCode” - 1)

Name	Request	Response
------	---------	----------

Amount	M	M
CurrencyCode	M	M
DateTimeHost		O
CardEntryMode	M	M
PAN	C	M
CardExpiryDate	C	O
AuthorizationCode		O
ReferenceNumber		O
ResponseCodeHost		O
TextResponse		O
DateTimeCRM		M
Operation Code	1	1
OrigOperation		
TrxID		M
TrxIDHost		O
PathParameters		O
ReceiptData		O
AmountFee	O	O
TerminalOutID		M
Status		M
Commodity Code		
PaymentDetails		O (Cardholder Name)
ProviderCode		

Пополнение счета (Кредит) (“OperationCode” - 3)

Name	Request	Response
Amount	M	M
CurrencyCode	M	M
DateTimeHost		O
CardEntryMode	M	M
PAN	C	M
CardExpiryDate	C	O
AuthorizationCode		O
ReferenceNumber		O
ResponseCodeHost		O
TextResponse		O
DateTimeCRM		M
Operation Code	3	3
OrigOperation		
TrxID		M

TrxIDHost		O
PathParameters		O
ReceiptData		O
AmountFee	O	O
TerminalOutID		M
Status		M
Commodity Code		
PaymentDetails		
ProviderCode		

Отмена (“Operation Code” - 4)

Name	Request	Response
Amount	M	M
CurrencyCode	M	M
DateTimeHost		O
CardEntryMode	M	M
PAN	O	M
CardExpiryDate	O	O
AuthorizationCode		O
ReferenceNumber	M	O
ResponseCodeHost		O
TextResponse		O
DateTimeCRM		M
Operation Code	4	4
OrigOperation	M	M
TrxID		M
TrxIDHost		O
PathParameters		O
ReceiptData		O
AmountFee		O
TerminalOutID		M
Status		M
Commodity Code		
PaymentDetails		O (Cardholder Name)
ProviderCode		

Баланс (“Operation Code” - 13)

Name	Request	Response
------	---------	----------

Amount	0	M
CurrencyCode	M	M
DateTimeHost		O
CardEntryMode	M	M
PAN	C	M
CardExpiryDate	C	O
AuthorizationCode		
ReferenceNumber		O
ResponseCodeHost		O
TextResponse		O
DateTimeCRM		M
Operation Code	13	13
OrigOperation		
TrxID		M
TrxIDHost		O
PathParameters		O
ReceiptData		O
AmountFee	O	O
TerminalOutID		M
Status		M
Commodity Code		
PaymentDetails		O (Cardholder Name)
ProviderCode		

Баланс (безпиновЫЙ, только протокол- OW) (“Operation Code” - 14)

Name	Request	Response
Amount	0	M
CurrencyCode	M	M
DateTimeHost		O
CardEntryMode	M	M
PAN	C	M
CardExpiryDate	C	O
AuthorizationCode		
ReferenceNumber		O
ResponseCodeHost		O
TextResponse		O
DateTimeCRM		M
Operation Code	14	14
OrigOperation		

TrxID		M
TrxIDHost		O
PathParameters		O
ReceiptData		O
AmountFee	O	O
TerminalOutID		M
Status		M
Commodity Code		
PaymentDetails		O (Cardholder Name)
ProviderCode		

Возврат (“Operation Code” - 29)

Name	Request	Response
Amount	M	M
CurrencyCode	M	M
DateTimeHost		O
CardEntryMode	M	M
PAN	C	M
CardExpiryDate	C	O
AuthorizationCode		O
ReferenceNumber	M	O
ResponseCodeHost		O
TextResponse		O
DateTimeCRM		M
Operation Code	29	29
OrigOperation		
TrxID		M
TrxIDHost		O
PathParameters		O
ReceiptData		O
AmountFee		O
TerminalOutID		M
Status		M
Commodity Code		
PaymentDetails		O (Cardholder Name)
ProviderCode		

Платеж (коммунальный) (“Operation Code” - 61)

Name	Request	Response
Amount	M	M
CurrencyCode	M	M
DateTimeHost		O
CardEntryMode	M	M
PAN	C	M
CardExpiryDate	C	O
AuthorizationCode		O
ReferenceNumber		O
ResponseCodeHost		O
TextResponse		O
DateTimeCRM		M
Operation Code	61	61
OrigOperation		
TrxID		M
TrxIDHost		O
PathParameters		O
ReceiptData		O
AmountFee	O	O
TerminalOutID		M
Status		M
Commodity Code	M	M
PaymentDetails	M	M
ProviderCode	M	M

Платеж в сторону 3-х лиц (“Operation Code” - 62)

Name	Request	Response
Amount	M	M
CurrencyCode	M	M
DateTimeHost		O
CardEntryMode	M	M
PAN	C	M
CardExpiryDate	C	O
AuthorizationCode		O
ReferenceNumber		O
ResponseCodeHost		O
TextResponse		O
DateTimeCRM		M

Operation Code	62	62
OrigOperation		
TrxID		M
TrxIDHost		O
PathParameters		O
ReceiptData		O
AmountFee	O	O
TerminalOutID		M
Status		M
Commodity Code	M	M
PaymentDetails	M	M
ProviderCode		

Пополнение счета (Cash to Card) (“OperationCode” - 71)

Name	Request	Response
Amount	M	M
CurrencyCode	M	M
DateTimeHost		O
CardEntryMode	M	M
PAN	C	O
CardExpiryDate	C	O
AuthorizationCode		O
ReferenceNumber		O
ResponseCodeHost		O
TextResponse		O
DateTimeCRM		M
Operation Code	71	71
OrigOperation		
TrxID		M
TrxIDHost		O
PathParameters		O
ReceiptData		O
AmountFee	O	O
TerminalOutID		M
Status		M
Commodity Code		
PaymentDetails		O (Cardholder Name)
ProviderCode		

Мини-выписка (“Operation Code” - 74)

Name	Request	Response
Amount	0	O
CurrencyCode	M	M
DateTimeHost		O
CardEntryMode	M	M
PAN	C	M
CardExpiryDate	C	O
AuthorizationCode		O
ReferenceNumber		O
ResponseCodeHost		O
TextResponse		O
DateTimeCRM		M
Operation Code	74	M
TrxID		M
TrxIDHost		O
PathParameters		O
ReceiptData		O
AmountFee	O	O
TerminalOutID		M
Status		M
Commodity Code		
PaymentDetails		O (Mini-Statement)
ProviderCode		

Служебные операции:

При выполнении любой **служебной** операции (**кроме 90**), ЗАПРЕЩЕНО устанавливать в запросе поле Amount (даже в 0 значение)!

Чтение карты (“OperationCode” - 90)

Name	Request	Response
Amount	M	S
CurrencyCode	M	S
CardEntryMode	0	M
PAN		M
CardExpiryDate		O
TextResponse		O
DateTimeCRM		M
Operation Code	89	89
Status		M

Сверка итогов (закрытие дня) (“OperationCode” - 59)

Name	Request	Response
Amount		М (Итоговая сумма по операциям на устройстве за день)
CurrencyCode		
DateTimeHost		О
CardEntryMode		
PAN		
CardExpiryDate		
AuthorizationCode		
ReferenceNumber		
ResponseCodeHost		О
TextResponse		О
DateTimeCRM		М
Operation Code	59	59
OrigOperation		
TrxID		М
TrxIDHost		О
PathParameters		О
ReceiptData		М (Кол-во успешных операций на устройстве за день)
AmountFee		О (Сумма по операциям на хосте за день (только протокол TRTP))
TerminalOutID		М
Status		М
Commodity Code		
PaymentDetails		О (Кол-во операций с устройства на хосте за день (только протокол TRTP))
ProviderCode		

Просмотр журнала (“OperationCode” - 60)

Name	Request	Response
Amount		
CurrencyCode		
DateTimeHost		
CardEntryMode		

PAN		
CardExpiryDate		
AuthorizationCode		
ReferenceNumber		
ResponseCodeHost		
TextResponse		
DateTimeCRM		
Operation Code	60	60
OrigOperation		
TrxID		
TrxIDHost		
PathParameters		О
ReceiptData		М (Кол-во успешных операций в журнале)
AmountFee		
TerminalOutID		
Status		М
Commodity Code		
PaymentDetails		
ProviderCode		

При выполнении этой операции, на каждую запись в журнале будет посылаться событие с InfoStatus=137.

Количество таких событий, будет равно значению, которое вернется в поле ReceiptData, в ответе, после выполнения операции.

StatusDescription такого события имеет следующий формат:

OperationCode~Дата~Время~PAN~Amount~CurrencyCode~AuthorizationCode~Reference Number

Например,

информация о хранящейся транзакции в журнале-

Операция: Оплата Дата:08/10/2015 Время:16:34:54 Карта:*****1234 Сумма:9546.25 RUB AuthCode:AB3459 RRN:000123456789

будет передана как событие с данными:

1~2015/10/08~16:34:54~*****1684~954625~643~AB3459~000123456789

информация о хранящейся транзакции в журнале-

Операция: Возврат Дата:02/09/15 Время:11:18:14 Карта:*****4008 Сумма:100.00 USD AuthCode:674231 RRN:000101

будет передано как событие с данными -

29~2015/09/02~11:18:14~*****4008~10000~840~674231~000101

Пример использования

Ниже приводится пример (на Delphi) выполнения транзакции в режиме с использованием событий. В примере выполняется стандартная операция - Оплата

```
TTransSender = class (TOleServer);
```

```
Exec:=CreateComObject(CLASS_TransSender) as TransSender;
```

```
WrapperExec:=TTransSender.Create(nil);
```

```
WrapperExec.ConnectTo(Exec);
```

```
WrapperExec.OnExchange:=Transaction_EventHandle; // Это метод- обработчик событий,
```

```
// который необходимо реализовать в вашей
```

программе

```
{Objects creating – Request, Response}
```

```
Request:= CreateComObject(CLASS_TransData) as TransData;
```

```
Response:= CreateComObject(CLASS_TransData) as TransData;
```

```
{Initialize the resources}
```

```
Exec.InitResources();
```

```
{Request Creating} // заполняются только необходимые поля, неиспользуемые поля  
// - должны быть пустыми (нулевыми)
```

```
Request.OperationCode:= 1; // выставляем значение кода операции - Оплата
```

```
Request.CardEntryMode:=0; // указываем, что вариант предъявления карты может быть  
// любой
```

```
Request.Amount:= '1020'; // Сумма операции – 10 руб. 20 коп.
```

```
Request.CurrencyCode:= '643'; // код валюты операции (Рубль)
```

```
{Perform the request for operation }
```

```
WrapperExec.StartAsyncExchange(Request,60,30);
```

```
.....
```

```
.....
```

```
{
```

где-то в обработчике OnExchange(..) – в функции Transaction_EventHandle(...)

ожидаем прихода событий. При определенных событиях производится отображение сообщений для клиента.

При получении события с InfoStatus – ‘Transaction FINISH’ переходим к следующему шагу - получению результата.

При получении события с InfoStatus – ‘Transaction timeout ERROR’ можно результат не получать, а просто завершать транзакцию с ошибкой.

.....

Важно: обработчик событий должен как можно быстрее выполнять обработку, лучше всего для обрабатываемых событий запускать отдельный поток или взводить таймер. Также нельзя получать результат выполнения транзакции (с помощью EndAsyncExchange()) непосредственно в обработчике событий, это необходимо делать в отдельном потоке или участке кода.

}

{Get the results}

WrapperExec.EndAsyncExchange(Response);

case Response.Status of

0: Result := 'Undefenite Status';

1: Result := 'Approved'; { read from Response the properties and print the receipt}

5: Result := 'Unsupported card';

13: Result := 'Break by error';

16: Result := 'Decline';

17: Result:=‘Approved OFFLINE’;{read from Response the properties and print the receipt}

34: Result := 'No Connection with host';

53: Result := 'Operation is interrupted';

54: Result:= ‘Timeout of operation’;

end;

{Kill the objects}

Response:=nil;

Request:=nil;

Ниже пример выполнения операции ‘Платеж’ в режиме без использования событий

Exec:=CreateComObject(CLASS_TransSender) as TransSender;

{Objects creating – Request, Response}

```
Request:=CreateComObject(CLASS_TransData) as TransData;  
Response:=CreateComObject(CLASS_TransData) as TransData;
```

```
{Initialize the resources}
```

```
Exec.InitResources();
```

```
{Request Creating}
```

```
Request.OperationCode := 62;    // выставляем код операции – Платеж в сторону 3-х лиц  
Request.CardEntryMode:=7;      // принимаем только бесконтактные банковские карты  
Request.Amount := '10000';     // Сумма платежа – 100 рублей 00 копеек  
Request.CurrencyCode := '643'; // Код валюты ( рубль)  
Request.CommodityCode:=9420;   // Код мобильного оператора  
Request. PaymentDetails: =89161235678; // Номер телефона
```

```
{Perform the request for online operation and get the answer}
```

```
Exec.Exchange(Request,Response,60);
```

```
{Get the results}
```

```
case Response.Status of
```

```
0: Result := 'Undefenite Status';
```

```
1: Result := 'Approved'; { read from Response the properties and print the receipt}
```

```
13: Result := 'Abnormal Termination';
```

```
16: Result := 'Decline';
```

```
34: Result := 'No Connection';
```

```
53: Result := 'Operation is interrupted';
```

```
end;
```

```
{Kill the objects}
```

```
Response:=nil;
```

```
Request:=nil;
```