Интеграция с ПО UPOS через библиотеку PILOT_NT.

1.0

Создано системой Doxygen 1.8.11

Оглавление

Глава 1

Основные сведения

Для обслуживания банковских карт Сбербанк предоставляет программную библиотеку PILOT_↩ NT.DLL, обеспечивающую проведение авторизаций по картам, а также формирование карточных чеков и итоговых отчетов за день. Для работы с картами кассовый аппарат оснащается дополнительным внешним устройством – ПИН-клавиатурой. Это устройство используется для считывания карт, для ввода ПИН-кода клиентом, а также выполняет функции криптографической защиты при обмене данными с процессинговой системой Сбербанка. ПИН-клавиатура может подключаться к COM-порту, порту USB или по протоколу ТТК2 через Ethernet Tcp/Ip. Сбербанк предоставляет это устройство бесплатно на период действия договора эквайринга с торговым предприятием. Если кассовый аппарат (ККМ) оснащен собственным ридером для магнитной полосы, то в период первоначальной разработки/отладки кассового ПО можно обойтись без ПИН-клавиатуры. Для окончательного тестирования (а также для реальной работы в торговом предприятии) ПИН-клавиатура понадобится обязательно.

Глава 2

Порядок вызова функций библиотеки

При оплате (возврате) покупки по банковской карте кассовая программа должна вызвать из библиотеки Сбербанка функцию card_authorize(), заполнив поля TType и Amount и указав нулевые значения в остальных полях. По окончании работы функции необходимо проанализировать поле RCode. Если в нем содержится значение «0» или «00», авторизация считается успешно выполненной, в противном случае – отклоненной. Кроме этого, необходимо проверить значение поля Check. Если оно не равно NULL, его необходимо отправить на печать (в нефискальном режиме) и затем удалить вызовом функции GlobalFree(). При закрытии смены кассовая программа должна вызвать из библиотеки Сбербанка функцию close_day(), заполнив поле TType = 7 и указав нулевые значения в остальных полях. По окончании работы функции необходимо проверить значение поля Check. Если поле Check не равно NULL, его необходимо отправить на печать (в нефискальном режиме) и после этого удалить вызовом функции GlobaFree().

Глава 3

Выполнение платежа с аварийной отменой

В случае, если внешняя программа не обеспечивает гарантированной печати карточного чека из поля Check, она может использовать следующую логику:

1. Выполнить функцию card_authorize().
2. После завершения работы функции card_authorize(), если транзакция выполнена успешно, вызвать функцию SuspendTrx() и приступить к печати чека.
3. Если чек напечатан успешно, вызвать функцию CommitTrx().
4. Если во время печати чека возникла неустранимая проблема, вызвать функцию RollbackTrx() для отмены платежа.
5. Если в ходе печати чека произойдет зависание ККМ или сбой питания, то транзакция останется в «подвешенном» состоянии. При следующем сеансе связи с банком она автоматически отменится.

Для корректного выполнения операций с функцией аварийной отмены необходимо придерживаться следующих правил:

1. Функция SuspendTrx() должна вызываться, только если оплата завершилась успешно – если возвращен код ответа 0.
2. Если вызов функции SuspendTrx() завершился ошибкой, операция должна считаться не выполненной, RollbackTrx() или CommitTrx() не должны вызываться. Если необходимо отменить операцию – то только ручной обработкой через письмо в банк с пояснением ситуации и приложением имеющихся документов. Причинами ошибок при выполнении функции SuspendTrx() могут быть:
   (a) Не верная сумма операции (см. п 4.);
   (b) Отсутствие связи с пинпадом;
   (c) Ошибки в системном ПО пинпада (например, сбой файловой системы).
3. Функции CommitTrx() или RollbackTrx() должны вызываться, только если был успешный вызов SuspendTrx().
4. В функции SuspendTrx(), CommitTrx(), RollbackTrx() обязательно должна передаваться корректная сумма операции.

5. Если касса работает по описанному выше алгоритму и функция CommitTrx() вернет ошибку – откатываем операцию на кассе, товар не отдаем, если клиент считает, что деньги списаны – предлагаем клиенту написать претензию или сами пишем в Сбербанк с просьбой отменить операцию.

6. Если касса работает по описанному выше алгоритму и функция RollbackTrx() вернет ошибку – необходимо обратиться в Сбербанк с информацией об операциях (чеки, контрольные ленты) с просьбой отменить эту операцию.

7. Авариная отмена предназначена только для операции «Оплата».

Например:

```
//оплата на 10 руб (1000 коп).
struct auth_answer aa;
memset(&aa, 0, sizeof(aa));
aa.TType=OP_PURCHASE;
aa.Amount=1000;
carderr=card_authorize(NULL, &aa);
if (carderr==0){
    carderr = SuspendTrx(1000, NULL);
    //если carderr!=0, считаем операцию не выполненой,
    //CommitTrx() или RollbackTrx() не вызываем товар не отдаем, если клиент считает, что
    //деньги списаны предлагаем клиенту написать претензию или сами пишем в
    //Сбербанк с просьбой отменить операцию
    if (действия кассы по печати чека, сохранению операции и т.п. успешны){
        if (carderr==0){
            carderr=CommitTrx(1000, NULL);
            if (carder!=0){
                //Отменяем операцию на кассе, товар не отдаем, если клиент считает,
                //что деньги списаны предлагаем клиенту написать претензию или сами пишем в Сбербанк с
      просьбой отменить операцию
            }
        }
        //если carderr!=0, то подтверждать не надо, т.к. suspend отработал с ошибкой.
    }else{
        // действия кассы по печати чека, сохранению операции и т.п. НЕ УСПЕШНЫ
        if (carderr==0){
            carderr=RollbackTrx(1000, NULL);
            if (carderr!=0){
                //пишем письмо в Сбербанк с просьбой отменить эту операцию
            }
        }else{
            //пишем письмо в Сбербанк с просьбой отменить эту операцию (см. п.2)
        }
    }
}
//Не забываем освободить поле auth_answer::Check
if (aa.Check)
    ::GlobalFree((HGLOBAL)aa.Check);
```

Глава 4

Выполнение операций без карты

Операции возврат, отмена, завершение расчета или отмена предавторизации могут быть выполнены без считывания карты клиента. Это возможно, если исходные операции находятся в той же смене. Для проведения таких операций терминал должен быть соответствующим образом зарегистрирован в процессинговой системе Сбербанка и настроен. Выполнение операций возможно только на тех версиях функций card_authorize(), которые принимают RRN в качестве параметра.

При выполнении операции возврат без карты терминал:

1. Выполнит полную отмену, если сумма возврата равна сумме исходной операции.
2. Выполнит частичную отмену , если сумма возврата меньше суммы исходной операции и настройки терминала позволяют частичные отмены.
3. Вернет код ошибки 4118 "Операции не найдены", если сумма превышает сумму исходной операции.
4. Может быть отменена любая операция, за исключением подтверждения взноса наличных и отмены предавторизации.
5. Частичные отмены возможны только для операции оплата. Для всех остальных операций частичная отмена невозможна.
6. В цепочке из предавторизации и нескольких добавочных предавторизаций допускается полная отмена любой из добавочных предавторизаций. Отмена основной предавторизации без отмены добавочных предавторизаций невозможна.

Оплата бонусами «Спасибо», выполненная самим терминалом, а не кассовым ПО, не может быть отменена полностью или частично. Для таких операции допускается только возврат, поэтому при попытке выполнить возврат без карты по такой операции терминал вернет ошибку 4118. Данное ограничение действует, если терминал самостоятельно выполняет обработку бонусов "Спасибо", без возврата в кассовое ПО промежуточного кода ответа 4353. Ограничение не распространяется на ваше ПО, если оно получает от терминала промежуточный код ответа 4353, передает хэш карты на сервер "Спасибо" и уменьшает сумму операции при повторном вызове card_authorize().

Для выполнения операций возврат, отмена, завершение расчета или отмена предавторизации без карты клиента при вызове функции card_authorize() должны быть указаны следующие параметры:

1. Аргумент Track2 – должно содержать слово «QSELECT»;

2. Поле Amount - сумма операции, если 0 – копируется из исходной операции и кассиру предлагается подтвердить или изменить сумму, если выполняемая операция это допускает. Для операции завершение расчета сумма не должна быть равна 0, для операции отмена предавторизации сумма должна быть равна 0;

3. Поле RRN - номер ссылки на исходную операцию.

4. Поля Department и CurrencyCode, при их отличии от 0 , будут использоваться как дополнительный фильтр при поиске исходных операций. При их отсутвии терминал скопирует из исходной операции код валюты и номер отдела

Терминал вернет код ошибки 4118, если по переданным входным параметрам не сумеет найти исходную операцию.

Например:

```
//оплата на 1000 руб (100000 коп).
struct auth_answer14 aa;
char rrn[RRN_LEN];  //здесь сохраним номер ссылки между оплатой и возвратом

memset(&aa, 0, sizeof(aa));
memset(&rrn, 0, sizeof(rrn));

aa.TType=OP_PURCHASE;
aa.Amount=100000;
carderr=card_authorize14(NULL, &aa);

if (carderr==0){
  //сохраняем RRN
  strcpy(rrn, aa.RRN);
  //печатаем чек, отпускаем топливо
  //все оплаченное топливо не вошло в бак, клиент уехал
  //выполняем возврат без карты

  if (aa.Check)
    ::GlobalFree((HGLOBAL)aa.Check);

  memset(&aa, 0, sizeof(aa));
  aa.TType=OP_RETURN;
  aa.Amount=2000;       //возврат 20 руб
  strcpy(aa.RRN, rrn);  //исходная операция будет найдена по ее ссылочному номеру

  carderr=card_authorize14("QSELECT", &aa);
  if (carderr==0){
    //возврат успешен, оформляем его в учетной системе
  }else{
    //что-то пошло не так. Пишем письмо в Сбербанк с описанием ситуации и просьбой вернуть клиенту 20 руб
  }
}

//Не забываем освободить поле auth_answer::Check
if (aa.Check)
  ::GlobalFree((HGLOBAL)aa.Check);
```

Глава 5

Алфавитный указатель групп

5.1 Группы

Полный список групп.

Глава 6

Алфавитный указатель структур данных

6.1 Структуры данных

Структуры данных с их кратким описанием.

Глава 7

Группы

7.1 Финансовые операции

Структуры данных

- struct auth_answer
- struct auth_answer2

  Расширение для получения кода авторизации успешной операции.
- struct auth_answer3

  Расширение для получения кода авторизации успешной операции и номера карты.
- struct auth_answer4

  Расширение для получения кода авторизации успешной операции, номера карты, кода ответа, времени операции и номера операции на кассе.
- struct auth_answer5

  Расширение для получения кода авторизации успешной операции и номера ссылки (RRN).
- struct auth_answer6

  Расширение для получения данных как при выполнении auth_answer5 и auth_answer4.
- struct payment_info_item

  This structure blah blah blah...
- struct auth_answer7

  Расширение для получения кода авторизации успешной операции, номера карты и признака принадлежности карты Сбербанку.
- struct auth_answer8

  Расширение для получения данных как при выполнении auth_answer5 и auth_answer4, а также номера карты и срока действия в шифрованном виде.
- struct preauth_rec

  Структура для описания одной операции в списке, по которой нужно выполнить завершение расчета.
- struct auth_answer9

  Расширение для получения кода авторизации успешной операции, номера карты, хеша номера карты и признака принадлежности карты Сбербанку.
- struct auth_answer10

  This structure blah blah blah...
- struct auth_answer11

  Расширение для получения кода авторизации успешной операции, номера карты, кода ответа, даты, времени и номера операции, хеша номера карты, признака принадлежности карты Сбербанку, третьей дорожки карты.

- struct auth_answer12

  Расширение card_authorize11 возможностью указать номер отдела и задать/получить номер ссылки.
- struct auth_answer13

  Расширение card_authorize11 возможностью указать номер отдела, код вылюты и получить способ чтения карты, имя клиента карты и AID.
- struct auth_answer14

  Расширение card_authorize13 возможностью указать информацию о товаре.

Макросы

- #define MAX_PAYMENT_ITEM 128
- #define MAX_ENCR_DATA 32
- #define MAX_AID_ASCII_LEN (32+1)
- #define MAX_CARD_NAME_LEN (32+1)
- #define MAX_FULL_ERROR_TEXT 128
- #define MAX_GOODS_DATA 25

Перечисления

- enum CardTypes {
  CT_USER = 0, CT_VISA = 1, CT_EUROCARD = 2, CT_CIRRUS = 3,
  CT_AMEX = 4, CT_DINERS = 5, CT_ELECTRON = 6, CT_PRO100 = 7,
  CT_CASHIER = 8, CT_SBERCARD = 9, CT_MIR = 10 }
- enum OpetationTypes {
  OP_PURCHASE = 1, OP_CASH = 2, OP_RETURN = 3, OP_BALANCE = 4,
  OP_FUNDS = 6, OP_ADD_AUTH = 42, OP_CANC_AUTH = 43, OP_PREAUTH = 51,
  OP_COMPLETION = 52, OP_CASHIN = 53, OP_CASHIN_COMP = 54, OP_PILOT_ST↩
  ART = 55,
  OP_PILOT_STATUS = 56, OP_PILOT_STOP = 57, OP_SETPIN = 58, OP_CHANGEPIN
  = 59 }

Функции

- PILOT_NT_API int card_authorize (char *track2, struct auth_answer *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize2 (char *track2, struct auth_answer2 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize3 (char *track2, struct auth_answer3 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize4 (char *track2, struct auth_answer4 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize5 (char *track2, struct auth_answer5 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize6 (char *track2, struct auth_answer6 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize6_ext (char *track2, struct auth_answer6 *auth_answer, struct payment_info_item *payinfo)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize7 (char *track2, struct auth_answer7 *auth_answer)

  Выполнение операций по картам

- PILOT_NT_API int card_authorize8 (char *track2, struct auth_answer8 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_complete_multi_auth8 (char *track2, struct auth_answer8 *auth_↩ ans, struct preauth_rec *pPreAuthList, int NumAuths)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize9 (char *track2, struct auth_answer9 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize10 (char *track2, struct auth_answer10 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize11 (char *track2, struct auth_answer11 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize12 (char *track2, struct auth_answer12 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize13 (char *track2, struct auth_answer13 *auth_answer)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize14 (char *track2, struct auth_answer14 *auth_answer, struct payment_info_item *payinfo)

  Выполнение операций по картам
- PILOT_NT_API int card_authorize15 (char *track2, struct auth_answer14 *auth_answer, struct payment_info_item *payinfo, CONTEXT_PTR dataIn, CONTEXT_PTR dataOut)

  Выполнение операций по картам

7.1.1 Подробное описание

7.1.2 Макросы

7.1.2.1 #define MAX_AID_ASCII_LEN (32+1)

Максимальный размер длинны AID чиповой карты (в виде ASCIIZ-строки)

7.1.2.2 #define MAX_CARD_NAME_LEN (32+1)

Максимальный размер длинны имени клиента на карте

7.1.2.3 #define MAX_ENCR_DATA 32

Размер буфера шифрованого номера карты и срока действия

7.1.2.4 #define MAX_FULL_ERROR_TEXT 128

Максимальный размер текста ошибки, возвращаемый UPOS

7.1.2.5 #define MAX_GOODS_DATA 25

Максимальный размер наименования или кода товара, включая завершающий ноль

7.1.2.6 #define MAX_PAYMENT_ITEM 128

Максимальный размер значения тега платежной системы.

7.1.3 Перечисления

7.1.3.1 enum CardTypes

Типы обслуживаемых карт

Элементы перечислений

CT_USER Выбор типа карты из меню, либо автоматически
CT_VISA Карта Visa.
CT_EUROCARD Карта Eurocard/Mastercard.
CT_CIRRUS Карта Cirrus/Maestro.
CT_AMEX Карта Amex.
CT_DINERS Карта DinersCLub.
CT_ELECTRON Карта VisaElectron.
CT_PRO100 Карта PRO100.
CT_CASHIER Карта кассира
CT_SBERCARD Карта Сберкарт
CT_MIR Карта МИР

7.1.3.2 enum OpetationTypes

Типы операций

Элементы перечислений

OP_PURCHASE Оплата покупки
OP_CASH Выдача наличных
OP_RETURN Возврат либо отмена покупки
OP_BALANCE Баланс
OP_FUNDS Безнал.перевод
OP_ADD_AUTH Добавочная предавторизация
OP_CANC_AUTH Отмена предавторизации
OP_PREAUTH Предавторизация
OP_COMPLETION Завершение расчета
OP_CASHIN Взнос наличных
OP_CASHIN_COMP Подтверждение взноса
OP_PILOT_START Запуск мониторинга
OP_PILOT_STATUS Опрос статуса мониторинга
OP_PILOT_STOP Остановка мониторинга
OP_SETPIN Установка ПИНа
OP_CHANGEPIN Смена ПИНа

7.1.4 Функции

7.1.4.1 PILOT_NT_API int card_authorize (char * track2, struct auth_answer * auth_answer)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	Основные параметры операции. См. auth_answer

Возвращает

int Код ошибки.

7.1.4.2 PILOT_NT_API int card_authorize10 (char * track2, struct auth_answer10 * auth_answer)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer10

Возвращает

int Код ошибки.

7.1.4.3 PILOT_NT_API int card_authorize11 (char * track2, struct auth_answer11 * auth_answer)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer11

Возвращает

int Код ошибки.

7.1.4.4 PILOT_NT_API int card_authorize12 (char * track2, struct auth_answer12 * auth_answer)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer12

Возвращает

int Код ошибки.

7.1.4.5 PILOT_NT_API int card_authorize13 (char * track2, struct auth_answer13 * auth_answer)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer13

Возвращает

int Код ошибки.

7.1.4.6 PILOT_NT_API int card_authorize14 (char * track2, struct auth_answer14 * auth_answer, struct payment_info_item * payinfo)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer14
in,out	payinfo	Информация для платежной системы

Возвращает

int Код ошибки.

7.1.4.7 PILOT_NT_API int card_authorize15 (char * track2, struct auth_answer14 * auth_answer, struct payment_info_item * payinfo, CONTEXT_PTR dataIn, CONTEXT_PTR dataOut)

Выполнение операций по картам

Заметки

Функция дополнительно принимает входной и выходной контексты операции. Во входном контексте в библиотеку передаются дополнительные параметры операции в выходном контексте возвращается расширенный результат операции. Параметры, которые могут переданы или полученны через контекст операции перечислены в EParameterName.

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer14
in,out	payinfo	Информация для платежной системы
in	dataIn	см. ctxAlloc
in	dataOut	см. ctxAlloc

Возвращает

int Код ошибки.

7.1.4.8 PILOT_NT_API int card_authorize2 (char * track2, struct auth_answer2 * auth_answer)

Выполнение операций по картам

Аргументы

out	track2	- данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	- см. auth_answer2

Возвращает

int Код ошибки.

7.1.4.9 PILOT_NT_API int card_authorize3 (char * track2, struct auth_answer3 * auth_answer)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer3

Возвращает

int Код ошибки.

7.1.4.10 PILOT_NT_API int card_authorize4 (char * track2, struct auth_answer4 * auth_answer)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer4

Возвращает

int Код ошибки.

7.1.4.11 PILOT_NT_API int card_authorize5 (char * track2, struct auth_answer5 * auth_answer)

Выполнение операций по картам

Аргументы

out	track2	- данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	- см. auth_answer5

Возвращает

int Код ошибки.

7.1.4.12 PILOT_NT_API int card_authorize6 (char * track2, struct auth_answer6 * auth_answer)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer6

Возвращает

int Код ошибки.

7.1.4.13 PILOT_NT_API int card_authorize6_ext (char * track2, struct auth_answer6 * auth_answer, struct payment_info_item * payinfo)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer6
in,out	payinfo	см. payment_info_item

Возвращает

int Код ошибки.

7.1.4.14 PILOT_NT_API int card_authorize7 (char * track2, struct auth_answer7 * auth_answer)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer7

Возвращает

int Код ошибки.

7.1.4.15 PILOT_NT_API int card_authorize8 (char * track2, struct auth_answer8 * auth_answer)

Выполнение операций по картам

Заметки

Функция удваивает количество воззвращаемых чеков. При работе по протоколу ТТК поле EncryptedData не заполняется.

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer8

Возвращает

int Код ошибки.

7.1.4.16 PILOT_NT_API int card_authorize9 (char * track2, struct auth_answer9 * auth_answer)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_answer	см. auth_answer9

Возвращает

int Код ошибки.

7.1.4.17 PILOT_NT_API int card_complete_multi_auth8 (char * track2, struct auth_answer8 * auth_ans, struct preauth_rec * pPreAuthList, int NumAuths)

Выполнение операций по картам

Аргументы

in	track2	данные дорожки карты с магнитной полосой. Если NULL, то будет предложено считать карту.
in,out	auth_ans	см. auth_answer8
in,out	pPreAuthList	Массив структур с описанием, по которым требуется выполнить "Завершение расчета". См. preauth_rec
in	NumAuths	Общее количество операций, по которым требуется выполнить "Завершение расчета".

Возвращает

int Код ошибки.

7.2 Служебные операции

Функции

- PILOT_NT_API int GetTerminalID (char *pTerminalID)
  Получить номер терминала как строку
- PILOT_NT_API int ServiceMenu (struct auth_answer *auth_answer)
  Открыть в техническое меню на пинпаде
- PILOT_NT_API int close_day (struct auth_answer *auth_answer)
  Закрытие дня.
- PILOT_NT_API int close_day_info (struct auth_answer *auth_answer, const char *iParams)
  Закрытие дня.
- PILOT_NT_API int get_statistics (struct auth_answer *auth_answer)
  Получение текущего отчета за текущую смену
- PILOT_NT_API unsigned int GetVer ()
  Получение номера версии библиотеки pilot_nt.dll.
- PILOT_NT_API int SetConfigData (const char *pConfData)
  Установить конфигурационные параметры в pinpad.ini.
- PILOT_NT_API void Done ()
  Деинициализация библиотеки pilot_nt.dll.
- PILOT_NT_API int EnableReader (HWND hDestWindow, UINT message)
  Асинхронное включение чтения карты
- PILOT_NT_API int DisableReader ()
  Асинхронное выключение чтения карты
- PILOT_NT_API int TestPinpad ()
  Проверка готовности пинпада
- PILOT_NT_API int SuspendTrx (DWORD dwAmount, char *pAuthCode)
  Функция переводит последнюю успешную транзакцию в «подвешенное» состояние. Если транзакция находится в этом состоянии, то при следующем сеансе связи с банком она будет отменена..
- PILOT_NT_API int CommitTrx (DWORD dwAmount, char *pAuthCode)
  Функция возвращает последнюю успешную транзакцию в «нормальное» состояние. После этого транзакция будет включена в отчет и спроцессирована как успешная. Перевести ее снова в «подвешенное» состояние будет уже нельзя.
- PILOT_NT_API int RollbackTrx (DWORD dwAmount, char *pAuthCode)
  Функция вызывает немедленную отмену последней успешной операции
- PILOT_NT_API int AbortTransaction ()
  Функция прерывает работу функций card_authorizeX()

7.2.1 Подробное описание

7.2.2 Функции

7.2.2.1 PILOT_NT_API int AbortTransaction ( )

Функция прерывает работу функций card_authorizeX()

Заметки

Внешнее ПО может вызвать эту функцию из отдельного треда, чтобы досрочно прекратить выполнение любой из функций card_authorize...(). При этом функция card_authorize...() завершится с кодом ошибки 2000 (операция прервана).

Возвращает

int Код ошибки.

7.2.2.2 PILOT_NT_API int close_day (struct auth_answer * auth_answer)

Закрытие дня.

Заметки

Поля TType, Amount, CType заполнять не нужно.

Аргументы

in,out	auth_answer	см. auth_answer

Возвращает

int Код ошибки.

7.2.2.3 PILOT_NT_API int close_day_info (struct auth_answer * auth_answer, const char * iParams)

Закрытие дня.

Заметки

Поля auth_answer::TType, auth_answer::Amount, auth_answer::CType заполнять не нужно. char* Check

Содержит образ отчета по картам, который вызывающая программа должна отправить на печать, а затем освободить вызовом функции GlobalFree().

Может иметь значение NULL. В этом случае никаких действий с ним вызывающая программа выполнять не должна.

@param[in,out] auth_answer см. ::auth_answer
@param[in,out] iParams Дополнительная информация для передачи на хост Сбербанка (например, о наличном обороте кассы за смену).
В текущей версии поддерживается следующий формат строки:
1;число_оплат_за_наличные;сумма_оплат_за_наличные;номер кассовой смены
Число 1 указывает, что следующие два параметра содержат количество наличных платежей и их сумму в копейках.
Пример: «1;55;20010000;2334» - означает, что за смену было 55 оплат наличными на общую сумму 200100.00 руб, номер кассовой смены 23
Параметр номер смены ккм используется для контроля уникальности, если в течении смены выполнялось несколько сверок итогов.
@return int Код ошибки.

7.2.2.4 PILOT_NT_API int CommitTrx (DWORD dwAmount, char * pAuthCode)

Функция возвращает последнюю успешную транзакцию в «нормальное» состояние. После этого транзакция будет включена в отчет и спроцессирована как успешная. Перевести ее снова в «подвешенное» состояние будет уже нельзя.

Заметки

Функция сверяет переданные извне параметры (сумму и код авторизации) со значениями в последней успешной операции, которая была проведена через библиотеку. Если хотя бы один параметр не совпадает, функция возвращает код ошибки 4140 и не выполняет никаких действий.

Аргументы

dwAmount	Сумма операции (в копейках)
pAuthCode	Код авторизации.

Возвращает

int Код ошибки.

7.2.2.5 PILOT_NT_API int DisableReader ( )

Асинхронное выключение чтения карты

Возвращает

int Код ошибки.

7.2.2.6 PILOT_NT_API void Done ( )

Деинициализация библиотеки pilot_nt.dll.

Заметки

Выполняется отключение от библиотеки gate.dll

7.2.2.7 PILOT_NT_API int EnableReader (HWND hDestWindow, UINT message)

Асинхронное включение чтения карты

Аргументы

in	hDestWindow	Хендл окна, которое будет получать сообщения о чтении карты
in	message	Идентификатор сообщение о чтении карты

Возвращает

int Код ошибки.

7.2.2.8 PILOT_NT_API int get_statistics (struct auth_answer * auth_answer)

Получение текущего отчета за текущую смену

Аргументы

in,out	auth_answer	См. auth_answer. При значении поля TType = 0 формируется краткий отчет, иначе - полный

Возвращает

int Код ошибки.

7.2.2.9 PILOT_NT_API int GetTerminalID (char * pTerminalID)

Получить номер терминала как строку

Заметки

Функция не поддерживается по протоколу ТТК для PCI-DSS решений.

Аргументы

out	pTerminalID	Строковое представление номера терминала. 9 байт.

Возвращает

int Код ошибки.

7.2.2.10 PILOT_NT_API unsigned int GetVer ( )

Получение номера версии библиотеки pilot_nt.dll.

Возвращает

Версия как целое число в формате 0x00VVRRBB. VV - версия, RR - релиз, BB - сборка

7.2.2.11 PILOT_NT_API int RollbackTrx (DWORD dwAmount, char * pAuthCode)

Функция вызывает немедленную отмену последней успешной операции

Заметки

Операция может быть предварительно возможно, ранее переведенную в «подвешенное» состояние, хотя это и не обязательно). Если транзакция уже была возвращена в «нормальное» состояние функцией CommitTrx(), то функция RollbackTrx() завершится с кодом ошибки 4141, не выполняя никаких действий.Функция сверяет переданные извне параметры (сумму и код авторизации) со значениями в последней успешной операции, которая была проведена через библиотеку. Если хотя бы один параметр не совпадает, функция возвращает код ошибки 4140 и не выполняет никаких действий.

Аргументы

dwAmount	Сумма операции (в копейках)
pAuthCode	Код авторизации.

Возвращает

int Код ошибки.

7.2.2.12 PILOT_NT_API int ServiceMenu (struct auth_answer * auth_answer)

Открыть в техническое меню на пинпаде

Заметки

По завершении поле auth_answer.Check может содержать образ документа для печати.

Аргументы

out	auth_answer	Буфер результата операции

Возвращает

int Код ошибки.

7.2.2.13 PILOT_NT_API int SetConfigData (const char * pConfData)

Установить конфигурационные параметры в pinpad.ini.

Аргументы

in	pConfData	Строка формата param1=value1;...paramN=valueN;

Возвращает

int Код ошибки.

7.2.2.14 PILOT_NT_API int SuspendTrx (DWORD dwAmount, char * pAuthCode)

Функция переводит последнюю успешную транзакцию в «подвешенное» состояние. Если транзакция находится в этом состоянии, то при следующем сеансе связи с банком она будет отменена..

Заметки

Функция сверяет переданные извне параметры (сумму и код авторизации) со значениями в последней успешной операции, которая была проведена через библиотеку. Если хотя бы один параметр не совпадает, функция возвращает код ошибки 4140 и не выполняет никаких действий.

Аргументы

dwAmount	Сумма операции (в копейках)
pAuthCode	Код авторизации.

Возвращает

int Код ошибки.

7.2.2.15 PILOT_NT_API int TestPinpad ( )

Проверка готовности пинпада

Заметки

Функция проверяет наличие пинпада. При успешном выполнении возвращает 0 (пинпад подключен), при неудачном – код ошибки (пинпад не подключен или неисправен).

Возвращает

int Код ошибки.

7.3 Чтение карты

Функции

- PILOT_NT_API int ReadCard (char *Last4Digits, char *Hash)

  Чтение карты
- PILOT_NT_API int ReadCardSB (char *Last4Digits, char *Hash)

  Функция позволяет прочитать банковскую карту для использования в скидочной системе «Спасибо от Сбербанка».
- PILOT_NT_API int ReadCardWithType (char *Last4Digits, char *Hash, int *pCardType, int *pIsOwnCard)

  Чтение карты
- PILOT_NT_API int ReadMaskedCardWithType (char *PAN, char *Hash, int *pCardType, int *pIsOwnCard)

  Чтение карты
- PILOT_NT_API int ReadTrack2 (char *Track2)

  Чтение полной второй дорожки карты
- PILOT_NT_API int ReadCardTrack3 (char *Last4Digits, char *Hash, char *pTrack3)

  Чтение полной третьей дорожки карты
- PILOT_NT_API int ReadSbercard (char *CardNo, char *ClientName)

  Чтение карты Сберкарт
- PILOT_NT_API int ReadCardAndName (char *pTrack2, char *ValidThru, char *pName)

  Чтение карты Сберкарт
- PILOT_NT_API int ReadCardFull (char *CardNo, char *ValidThru)

  Чтение карты
- PILOT_NT_API int ReadCardContext (CONTEXT_PTR dataOut)

  Чтение карты

7.3.1 Подробное описание

7.3.2 Функции

7.3.2.1 PILOT_NT_API int ReadCard (char * Last4Digits, char * Hash)

Чтение карты

Аргументы

Last4Digits	Буфер, куда функция записывает четыре последних цифры номера карты. Размер буфера должен быть не менее 5 байт.
Hash	Хеш от номера карты, в формате ASCII с нулевым байтом в конце. Размер буфера должен быть не менее 41 байта.

Возвращает

int Код ошибки.

7.3.2.2 PILOT_NT_API int ReadCardAndName (char * pTrack2, char * ValidThru, char * pName)

Чтение карты Сберкарт

Аргументы

out	pTrack2	Track2
out	ValidThru	Срок действия карты в формате YYMM
out	pName	имя фамилия клиента эмброссированные на карте

Возвращает

int Код ошибки.

7.3.2.3 PILOT_NT_API int ReadCardContext (CONTEXT_PTR dataOut)

Чтение карты

Аргументы

dataOut	Контекст операции, в которую копируются pan, хэш, тип карты, признак "Карта выпущена Сбербанком" и номер программы лояльности

Возвращает

int Код ошибки.

7.3.2.4 PILOT_NT_API int ReadCardFull (char * CardNo, char * ValidThru)

Чтение карты

Заметки

Чтобы использовать результат для авторизации, их нужно будет сформатировать так "Card↩ No=ValidThru". Функция не поддерживается по протоколу TTK для PCI-DSS решений.

Аргументы

CardNo	Номер карты. Может иметь длину от 13 до 19 цифр.
ValidThru	Срок действия карты в формате YYMM

Возвращает

int Код ошибки.

7.3.2.5 PILOT_NT_API int ReadCardSB (char * Last4Digits, char * Hash)

Функция позволяет прочитать банковскую карту для использования в скидочной системе «Спасибо от Сбербанка».

Заметки

Карта должна быть выдана Сбербанком, в противном случае функция вернет ошибку.

Аргументы

Last4Digits	Буфер, куда функция записывает четыре последних цифры номера карты. Размер буфера должен быть не менее 5 байт.
Hash	Хеш от номера карты, в формате ASCII с нулевым байтом в конце. Размер буфера должен быть не менее 41 байта.

Возвращает

int Код ошибки.

7.3.2.6 PILOT_NT_API int ReadCardTrack3 (char * Last4Digits, char * Hash, char * pTrack3)

Чтение полной третьей дорожки карты

Заметки

Данные третьей дорожки могут иметь длину до 40 символов.

Аргументы

out	Last4Digits	4 последние цифры карты
out	Hash	Хеш от номера карты, в формате ASCII с нулевым байтом в конце. Размер буфера должен быть не менее 41 байта.
out	pTrack3	третья дорожка карты

Возвращает

int Код ошибки.

7.3.2.7 PILOT_NT_API int ReadCardWithType (char * Last4Digits, char * Hash, int * pCardType, int * pIsOwnCard)

Чтение карты

Аргументы

Last4Digits	Буфер, куда функция записывает четыре последних цифры номера карты. Размер буфера должен быть не менее 5 байт.
Hash	Хеш от номера карты, в формате ASCII с нулевым байтом в конце. Размер буфера должен быть не менее 41 байта.
pCardType	Тип карты. См. CardTypes
pIsOwnCard	Признак "Карта выпущена Сбербанком"

Возвращает

int Код ошибки.

7.3.2.8 PILOT_NT_API int ReadMaskedCardWithType (char * PAN, char * Hash, int * pCardType, int * pIsOwnCard)

Чтение карты

Аргументы

PAN	Буфер, куда функция записывает PAN карты
Hash	Хеш от номера карты, в формате ASCII с нулевым байтом в конце. Размер буфера должен быть не менее 41 байта.
pCardType	Тип карты. См. CardTypes
pIsOwnCard	Признак "Карта выпущена Сбербанком"

Возвращает

int Код ошибки.

7.3.2.9 PILOT_NT_API int ReadSbercard (char * CardNo, char * ClientName)

Чтение карты Сберкарт

Аргументы

out	CardNo	полный номер карты
out	ClientName	имя фамилия клиента эмброссированные на карте

Возвращает

int Код ошибки.

7.3.2.10 PILOT_NT_API int ReadTrack2 (char * Track2)

Чтение полной второй дорожки карты

Заметки

Данные второй дорожки могут иметь длину до 40 символов. Вторая дорожка имеет формат: nnnn...nn=yymmddd...d где '=' - символ-разделитель nnn...n - номер карты yymm - срок действия карты (ГГММ) ddd...d - служебные данные карты

Аргументы

out	Track2	Буфер на 41 байт.

Возвращает

int Код ошибки.

7.4 Вендинговые операции

Функции

- PILOT_NT_API int SetGUIHandles (int hText, int hEdit)
  Установить элементы для вывода на экран
- PILOT_NT_API int EjectCard ()
  Извлечь карту. Команда 5020.
- PILOT_NT_API int CaptureCard ()
  Перемещение карты в лоток для задержанных карт. Команда 5021.
- PILOT_NT_API int TestCard ()
  Проверка наличия карты в чиповом считывателе.
- PILOT_NT_API int OpenKeyboard ()
  Открыть клавиатуру в нешифрованом режиме
- PILOT_NT_API int CloseKeyboard ()
  Закрыть клавиатуру
- PILOT_NT_API int ReadKeyboard ()
  Получить код нажатого символа без ожидания
- PILOT_NT_API unsigned char TestHardware ()
  Протестировать готовность оборудования

7.4.1 Подробное описание

7.4.2 Функции

7.4.2.1 PILOT_NT_API int CaptureCard ( )

Перемещение карты в лоток для задержанных карт. Команда 5021.

Возвращает

int Код ошибки.

7.4.2.2 PILOT_NT_API int CloseKeyboard ( )

Закрыть клавиатуру

Возвращает

int Код ошибки.

7.4.2.3 PILOT_NT_API int EjectCard ( )

Извлечь карту. Команда 5020.

Возвращает

int Код ошибки.

7.4.2.4 PILOT_NT_API int OpenKeyboard ( )

Открыть клавиатуру в нешифрованом режиме

Возвращает

int Код ошибки.

7.4.2.5 PILOT_NT_API int ReadKeyboard ( )

Получить код нажатого символа без ожидания

Возвращает

int Возвращается 0, если нажатия клавиши не было.

7.4.2.6 PILOT_NT_API int SetGUIHandles (int hText, int hEdit)

Установить элементы для вывода на экран

Аргументы

in	hText	Элемент вывода текста.
in	hEdit	Элемент ввода текста.

Возвращает

int Код ошибки.

7.4.2.7 PILOT_NT_API int TestCard ( )

Проверка наличия карты в чиповом считывателе.

Возвращает

int Код ошибки. 0 - карта внутри ридера, готова к работе; 254 - карты нет; 251 - карта в устье ридера, ожидает, что ее заберет клиент

7.4.2.8 PILOT_NT_API unsigned char TestHardware ( )

Протестировать готовность оборудования

Возвращает

unsigned char Возвращает битовую маску.

- 0x00 ни пинпад, ни картридер не готов
- 0x01 готов только пинпад
- 0x02 готов только картридер
- 0x03 готов и пинпад, и картридер

7.5 Функции работы с контекстом

Функции

- PILOT_NT_API CONTEXT_PTR ctxAlloc (void)

  Создать контекст операции. Функция создает пустой контекст операции.
- PILOT_NT_API void ctxFree (CONTEXT_PTR ctx)

  Удалить контекст операции
- PILOT_NT_API void ctxClear (CONTEXT_PTR ctx)

  Отчистить контекст. Функция удаляет все параметры из контекста.
- PILOT_NT_API int ctxSetString (CONTEXT_PTR ctx, EParameterName name, const char *str)

  Записать в контекст значение строкового параметра.
- PILOT_NT_API int ctxSetInt (CONTEXT_PTR ctx, EParameterName name, int val)

  Записать в контекст значение целочисленного параметра.
- PILOT_NT_API int ctxSetBinary (CONTEXT_PTR ctx, EParameterName name, unsigned char *val, int sz)

  Записать в контекст значение параметра в виде последовательности байт.
- PILOT_NT_API int ctxGetString (CONTEXT_PTR ctx, EParameterName name, char *str, int sz)

  Считать из контекста значение переменной в виде строки.
- PILOT_NT_API int ctxGetInt (CONTEXT_PTR ctx, EParameterName name, int *pVal)

  Считать из контекста значение переменной в виде целого числа.
- PILOT_NT_API int ctxGetBinary (CONTEXT_PTR ctx, EParameterName name, unsigned char *pVal, int *pOutSz, int MAXSIZE)

  Считать из контекста значение переменной в виде последовательности байт.

Переменные

- int TType
- unsigned long Amount
- char RCode [3]
- char AMessage [16]
- int CType
- char * Check
- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- struct auth_answer auth_answ
- char RRN [RRN_LEN]
- char AuthCode [AUTH_CODE_LEN]
- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]

- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- char RRN [RRN_LEN]
- DWORD dwTag
- char Value [MAX_PAYMENT_ITEM]
- BYTE Flags
- void * pNextItem
- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int SberOwnCard
- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- char RRN [RRN_LEN]
- char EncryptedData [MAX_ENCR_DATA *2+1]
- unsigned long Amount
- char RRN [RRN_LEN]
- char Last4Digits [5]
- unsigned short ErrorCode
- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int SberOwnCard
- char Hash [CARD_HASH_LEN]
- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- int SberOwnCard
- char Hash [CARD_HASH_LEN]
- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- int SberOwnCard
- char Hash [CARD_HASH_LEN]
- char Track3 [CARD_TRACK3_LEN]
- unsigned long RequestID
- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- int SberOwnCard
- char Hash [CARD_HASH_LEN]

- char Track3 [CARD_TRACK3_LEN]
- unsigned long RequestID
- DWORD Department
- char RRN [RRN_LEN]
- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- int SberOwnCard
- char Hash [CARD_HASH_LEN]
- char Track3 [CARD_TRACK3_LEN]
- DWORD RequestID
- DWORD Department
- char RRN [RRN_LEN]
- DWORD CurrencyCode
- char CardEntryMode
- char CardName [MAX_CARD_NAME_LEN]
- char AID [MAX_AID_ASCII_LEN]
- char FullErrorText [MAX_FULL_ERROR_TEXT]
- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- int SberOwnCard
- char Hash [CARD_HASH_LEN]
- char Track3 [CARD_TRACK3_LEN]
- DWORD RequestID
- DWORD Department
- char RRN [RRN_LEN]
- DWORD CurrencyCode
- char CardEntryMode
- char CardName [MAX_CARD_NAME_LEN]
- char AID [MAX_AID_ASCII_LEN]
- char FullErrorText [MAX_FULL_ERROR_TEXT]
- DWORD GoodsPrice
- DWORD GoodsVolume
- char GoodsCode [MAX_GOODS_DATA]
- char GoodsName [MAX_GOODS_DATA]

7.5.1 Подробное описание

7.5.2 Функции

7.5.2.1 PILOT_NT_API CONTEXT_PTR ctxAlloc (void)

Создать контекст операции. Функция создает пустой контекст операции.

Заметки

Для входного контекста значения параметров устанавливаются вызывающей программой, для выходного контекста значения параметров устанавливаются библиотекой. При повторном использовании контекста, если переменная уже есть в контексте, она будет перезаписана. Рекомендуется либо создавать и удалять контексты при каждой операции, либо очищать контекст вызовом функции ctxClear перед повторным использованием.

Возвращает

CONTEXT_PTR Идентификатор контекста операции или 0, если произошла ошибка

7.5.2.2 PILOT_NT_API void ctxClear (CONTEXT_PTR ctx)

Отчистить контекст. Функция удаляет все параметры из контекста.

Аргументы

in	ctx	Идентификатор контекста.

Возвращает

нет

7.5.2.3 PILOT_NT_API void ctxFree (CONTEXT_PTR ctx)

Удалить контекст операции

Аргументы

in	ctx	Идентификатор контекста.

Возвращает

нет

7.5.2.4 PILOT_NT_API int ctxGetBinary (CONTEXT_PTR ctx, EParameterName name, unsigned char * pVal, int * pOutSz, int MAXSIZE)

Считать из контекста значение переменной в виде последовательности байт.

Заметки

Для строковой переменной будет значение будет возвращено без изменения, для целочисленной переменной функция вернет последовательность из четырех байт с прямым порядком байт.

Аргументы

in	ctx	Идентификатор контекста.
in	name	Идентификатор параметра.
in	pVal	Указатель на буфер результата.
in	pOutSz	Количество байт, скопированных в буфер.
in	MAXSIZE	Максимально возможное количество байт.

Возвращает

int Код ошибки.

7.5.2.5 PILOT_NT_API int ctxGetInt (CONTEXT_PTR ctx, EParameterName name, int * pVal)

Считать из контекста значение переменной в виде целого числа.

Заметки

Для строковой переменной будет выполнено преобразование строки в число, для последовательности байт функция преобразует первые четыре байта последовательности в целое число с прямым порядком байт.

Аргументы

in	ctx	Идентификатор контекста.
in	name	Идентификатор параметра.
in	pVal	Указатель на число.

Возвращает

int Код ошибки.

7.5.2.6 PILOT_NT_API int ctxGetString (CONTEXT_PTR ctx, EParameterName name, char * str, int sz)

Считать из контекста значение переменной в виде строки.

Заметки

Для целочисленной переменной вы получите ее строковое представление, а для последовательности байт функция вернет hex строку.

Аргументы

in	ctx	Идентификатор контекста.
in	name	Идентификатор параметра.
in	val	Указатель на строку.
in	sz	Максимально возможная длина строки.

Возвращает

int Код ошибки.

7.5.2.7 PILOT_NT_API int ctxSetBinary (CONTEXT_PTR ctx, EParameterName name, unsigned char * val, int sz)

Записать в контекст значение параметра в виде последовательности байт.

Аргументы

in	ctx	Идентификатор контекста.
in	name	Идентификатор параметра.
in	val	Указатель на буфер.
in	sz	Длина буфера.

Возвращает

int Код ошибки.

7.5.2.8 PILOT_NT_API int ctxSetInt (CONTEXT_PTR ctx, EParameterName name, int val)

Записать в контекст значение целочисленного параметра.

Аргументы

in	ctx	Идентификатор контекста.
in	name	Идентификатор параметра.
in	val	Значение параметра.

Возвращает

int Код ошибки.

7.5.2.9 PILOT_NT_API int ctxSetString (CONTEXT_PTR ctx, EParameterName name, const char * str)

Записать в контекст значение строкового параметра.

Аргументы

in	ctx	Идентификатор контекста.
in	name	Идентификатор параметра.
in	str	Указатель на строку.

Возвращает

int Код ошибки.

7.5.3 Переменные

7.5.3.1 char AID[MAX_AID_ASCII_LEN]

[out] Application ID чиповой карты (в виде ASCIIZ-строки)

7.5.3.2 char AID[MAX_AID_ASCII_LEN]

[out] Application ID чиповой карты (уже в виде ASCIIZ-строки)

7.5.3.3 char AMessage[16]

[out] текст результата авторизации

7.5.3.4 unsigned long Amount

[in] сумма в копейках

7.5.3.5 unsigned long Amount

[in] сумма в копейках

7.5.3.6 auth_answer ans

[in, out] Основные параметры операции. См. auth_answer

7.5.3.7 auth_answer ans

[in, out] Основные параметры операции. См. auth_answer

7.5.3.8 auth_answer ans

[in, out] Основные параметры операции. См. auth_answer

7.5.3.9 auth_answer ans

[in, out] Основные параметры операции. См. auth_answer

7.5.3.10 auth_answer ans

[in, out] Основные параметры операции. См. auth_answer

7.5.3.11 auth_answer ans

[in, out] Основные параметры операции. См. auth_answer

7.5.3.12 struct auth_answer auth_answ

[in] Основные параметры операции. См. auth_answer

7.5.3.13 struct auth_answer auth_answ

[in, out] Основные параметры операции. См. auth_answer

7.5.3.14 struct auth_answer auth_answ

[in, out] Основные параметры операции. См. auth_answer

7.5.3.15 struct auth_answer auth_answ

[in] Основные параметры операции. См. auth_answer

7.5.3.16 struct auth_answer auth_answ

[in, out] Основные параметры операции. См. auth_answer

7.5.3.17 struct auth_answer auth_answ

[in, out] Основные параметры операции. См. auth_answer

7.5.3.18 struct auth_answer auth_answ

[in, out] Основные параметры операции. См. auth_answer

7.5.3.19 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.20 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.21 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.22 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.23 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.24 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.25 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.26 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.27 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.28 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.29 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.30 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.31 char AuthCode[AUTH_CODE_LEN]

[out] Код авторизации. 7 байт.

7.5.3.32 char CardEntryMode

[out] Способ чтения карты ('D'-магн.полоса, 'M'-ручной ввод, 'C'-чип, 'E'-бесконтакт EMV, 'R'-бесконтакт magstripe, 'F'-fallback)

7.5.3.33 char CardEntryMode

[out] Способ чтения карты ('D'-магн.полоса, 'M'-ручной ввод, 'C'-чип, 'E'-бесконтакт EMV, 'R'-бесконтакт magstripe, 'F'-fallback)

7.5.3.34 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт. Для международных карт все символы, кроме первых 6 и последних 4, будут заменены символами '*'.

7.5.3.35 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт.

7.5.3.36 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт.

7.5.3.37 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт.

7.5.3.38 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт.

7.5.3.39 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт.

7.5.3.40 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт. Для международных карт все символы, кроме первых 6 и последних 4, будут заменены символами '*'.

7.5.3.41 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт. Для международных карт все символы, кроме первых 6 и последних 4, будут заменены символами '*'.

7.5.3.42 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт. Для международных карт все символы, кроме первых 6 и последних 4, будут заменены символами '*'.

7.5.3.43 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт. Для международных карт все символы, кроме первых 6 и последних 4, будут заменены символами '*'.

7.5.3.44 char CardID[CARD_ID_LEN]

[out] Идентификатор карты. 25 байт. Для международных карт все символы, кроме первых 6 и последних 4, будут заменены символами '*'.

7.5.3.45 char CardName[MAX_CARD_NAME_LEN]

[out] Название типа карты

7.5.3.46 char CardName[MAX_CARD_NAME_LEN]

[out] Название типа карты

7.5.3.47 char* Check

[out] образ чека, должен освобождаться GlobalFree в вызывающей программе

7.5.3.48 int CType

[in,out] тип карты

7.5.3.49 DWORD CurrencyCode

[in] Международный код валюты (810, 643, 840, 978 и т.д.)

7.5.3.50 DWORD CurrencyCode

[in] Международный код валюты (810, 643, 840, 978 и т.д.)

7.5.3.51 DWORD Department

[in] Порядковый номер отдела от 0 до 14-ти, включительно. При установке номера отдела в 0x↩ FFFFFFFF, номер отдела будет запрошен через интерфейс терминала после вставки карты. Если номер отдела будет указан вне настроенного диапазона, то терминал вернет код ошибки 4191.

7.5.3.52 DWORD Department

[in] Порядковый номер отдела от 0 до 14-ти, включительно. При установке номера отдела в 0x↩ FFFFFFFF, номер отдела будет запрошен через интерфейс терминала после вставки карты. Если номер отдела будет указан вне настроенного диапазона, то терминал вернет код ошибки 4191.

7.5.3.53 DWORD Department

[in] Порядковый номер отдела от 0 до 14-ти, включительно. При установке номера отдела в 0x↩ FFFFFFFF, номер отдела будет запрошен через интерфейс терминала после вставки карты. Если номер отдела будет указан вне настроенного диапазона, то терминал вернет код ошибки 4191.

7.5.3.54 DWORD dwTag

Тег платежной системы

7.5.3.55 char EncryptedData[MAX_ENCR_DATA ∗2+1]

[in, out] шифрованый номер карты и срок действия.

7.5.3.56 int ErrorCode

[out] Код ошибки.

7.5.3.57 int ErrorCode

[out] Код ошибки.

7.5.3.58 int ErrorCode

[out] Код ошибки.

7.5.3.59 unsigned short ErrorCode

[out] Код ошибки.

7.5.3.60 int ErrorCode

[out] Код ошибки.

7.5.3.61 int ErrorCode

[out] Код ошибки.

7.5.3.62 int ErrorCode

[out] Код ошибки.

7.5.3.63 int ErrorCode

[out] Код ошибки.

7.5.3.64 int ErrorCode

[out] Код ошибки.

7.5.3.65 BYTE Flags

must be 0x40 for immediate sending

7.5.3.66 char FullErrorText[MAX_FULL_ERROR_TEXT]

[out] Полный текст сообщения об ошибке

7.5.3.67 char FullErrorText[MAX_FULL_ERROR_TEXT]

[out] Полный текст сообщения об ошибке

7.5.3.68 char GoodsCode[MAX_GOODS_DATA]

[in] Код товара во внешней системе

7.5.3.69 char GoodsName[MAX_GOODS_DATA]

[in] Наименование товара во внешней системе

7.5.3.70 DWORD GoodsPrice

[in] Цена за единицу товара, коп (34.99->3499)

7.5.3.71 DWORD GoodsVolume

[in] Количество товара, в тыс. долях (30.234->30234)

7.5.3.72 char Hash[CARD_HASH_LEN]

[in, out] хеш SHA1 от номера карты, в формате ASCII с нулевым байтом в конце. 40 байт.

7.5.3.73 char Hash[CARD_HASH_LEN]

[in, out] хеш от номера карты, в формате ASCII с нулевым байтом в конце

7.5.3.74 char Hash[CARD_HASH_LEN]

[in, out] хеш SHA1 от номера карты, в формате ASCII с нулевым байтом в конце. 40 байт.

7.5.3.75 char Hash[CARD_HASH_LEN]

[in, out] хеш SHA1 от номера карты, в формате ASCII с нулевым байтом в конце. 40 байт.

7.5.3.76 char Hash[CARD_HASH_LEN]

[in, out] хеш SHA1 от номера карты, в формате ASCII с нулевым байтом в конце. 40 байт.

7.5.3.77 char Hash[CARD_HASH_LEN]

[in, out] хеш SHA1 от номера карты, в формате ASCII с нулевым байтом в конце. 40 байт.

7.5.3.78 char Last4Digits[5]

[in] последние 4 цифры номера карты

7.5.3.79 void* pNextItem

??

7.5.3.80 char RCode[3]

[out] код результата авторизации

7.5.3.81 unsigned long RequestID

[in,out] Уникальный номер операции. Только PCI DSS решения.

7.5.3.82 unsigned long RequestID

[in,out] Уникальный номер операции. Только PCI DSS решения.

7.5.3.83 DWORD RequestID

[in,out] Уникальный номер операции. Только PCI DSS решения.

7.5.3.84 DWORD RequestID

[in,out] Уникальный номер операции. Только PCI DSS решения.

7.5.3.85 char RRN[RRN_LEN]

[in,out] Номер ссылки операции, присвоеный хостом. Используется для операций возврат и множественной авторизации. Содержит уникальный 12-значный ссылочный номер. При предавторизации это поле является выходным (его заполняет библиотека pilot_nt.dll), а при завершении расчета – входным (значение должно быть заполнено вызывающей программой; оно должно совпадать со значением, возвращенным при предавторизации).

7.5.3.86 char RRN[RRN_LEN]

[in,out] Номер ссылки операции, присвоеный хостом. Используется для операций возврат и множественной авторизации. Содержит уникальный 12-значный ссылочный номер. При предавторизации это поле является выходным (его заполняет библиотека pilot_nt.dll), а при завершении расчета – входным (значение должно быть заполнено вызывающей программой; оно должно совпадать со значением, возвращенным при предавторизации).

7.5.3.87 char RRN[RRN_LEN]

[in,out] Номер ссылки операции, присвоеный хостом. Используется для операций возврат и множественной авторизации. Содержит уникальный 12-значный ссылочный номер. При предавторизации это поле является выходным (его заполняет библиотека pilot_nt.dll), а при завершении расчета – входным (значение должно быть заполнено вызывающей программой; оно должно совпадать со значением, возвращенным при предавторизации).

7.5.3.88 char RRN[RRN_LEN]

[in,out] Номер ссылки операции, присвоеный хостом. Используется для операций возврат и множественной авторизации. Содержит уникальный 12-значный ссылочный номер. При предавторизации это поле является выходным (его заполняет библиотека pilot_nt.dll), а при завершении расчета – входным (значение должно быть заполнено вызывающей программой; оно должно совпадать со значением, возвращенным при предавторизации).

7.5.3.89 char RRN[RRN_LEN]

[in,out] Номер ссылки операции, присвоеный хостом. Используется для операций возврат и множественной авторизации. Содержит уникальный 12-значный ссылочный номер. При предавторизации это поле является выходным (его заполняет библиотека pilot_nt.dll), а при завершении расчета – входным (значение должно быть заполнено вызывающей программой; оно должно совпадать со значением, возвращенным при предавторизации).

7.5.3.90 char RRN[RRN_LEN]

[in,out] Номер ссылки операции, присвоеный хостом. Используется для операций возврат и множественной авторизации. Содержит уникальный 12-значный ссылочный номер. При предавторизации это поле является выходным (его заполняет библиотека pilot_nt.dll), а при завершении расчета – входным (значение должно быть заполнено вызывающей программой; оно должно совпадать со значением, возвращенным при предавторизации).

7.5.3.91 char RRN[RRN_LEN]

[in,out] Номер ссылки операции, присвоеный хостом. Используется для операций возврат, множественной авторизации и завершения расчета. Содержит уникальный 12-значный ссылочный номер. При предавторизации это поле является выходным (его заполняет библиотека pilot_nt.dll), а при завершении расчета – входным (значение должно быть заполнено вызывающей программой; оно должно совпадать со значением, возвращенным при предавторизации).

7.5.3.92 int SberOwnCard

[out] Флаг принадлежности карты Сбербанку

7.5.3.93 int SberOwnCard

[out] Флаг принадлежности карты Сбербанку

7.5.3.94 int SberOwnCard

[out] Флаг принадлежности карты Сбербанку

7.5.3.95 int SberOwnCard

[out] Флаг принадлежности карты Сбербанку

7.5.3.96 int SberOwnCard

[out] Флаг принадлежности карты Сбербанку

7.5.3.97 int SberOwnCard

[out] Флаг принадлежности карты Сбербанку

7.5.3.98 int SberOwnCard

[out] Флаг принадлежности карты Сбербанку

7.5.3.99 char Track3[CARD_TRACK3_LEN]

[out] Третья дорожка карты

7.5.3.100 char Track3[CARD_TRACK3_LEN]

[out] третья дорожка карты

7.5.3.101 char Track3[CARD_TRACK3_LEN]

[out] третья дорожка карты

7.5.3.102 char Track3[CARD_TRACK3_LEN]

[out] третья дорожка карты

7.5.3.103 char TransDate[TRANSDATE_LEN]

[out] Дата и время операции

7.5.3.104 char TransDate[TRANSDATE_LEN]

[out] Дата и время операции

7.5.3.105 char TransDate[TRANSDATE_LEN]

[out] Дата и время операции

7.5.3.106 char TransDate[TRANSDATE_LEN]

[out] Дата и время операции

7.5.3.107 char TransDate[TRANSDATE_LEN]

[out] Дата и время операции

7.5.3.108 char TransDate[TRANSDATE_LEN]

[out] Дата и время операции

7.5.3.109 char TransDate[TRANSDATE_LEN]

[out] Дата и время операции

7.5.3.110 char TransDate[TRANSDATE_LEN]

[out] Дата и время операции

7.5.3.111 int TransNumber

[out] Номер операции за опер. день, см. номер на чеке

7.5.3.112 int TransNumber

[out] Номер операции за опер. день, см. номер на чеке

7.5.3.113 int TransNumber

[out] Номер операции за опер. день, см. номер на чеке

7.5.3.114 int TransNumber

[out] Номер операции за опер. день, см. номер на чеке

7.5.3.115 int TransNumber

[out] Номер операции за опер. день, см. номер на чеке

7.5.3.116 int TransNumber

[out] Номер операции за опер. день, см. номер на чеке

7.5.3.117 int TransNumber

[out] Номер операции за опер. день, см. номер на чеке

7.5.3.118 int TransNumber

[out] Номер операции за опер. день, см. номер на чеке

7.5.3.119 int TType

[in] тип транзакции. см OpetationTypes

7.5.3.120 char Value[MAX_PAYMENT_ITEM]

Значение тэга платежной системы. 128 байт. MAX_PAYMENT_ITEM

7.6 Коды ошибок

Макросы

- #define ERR_OK 0
- #define SPS_OK ERR_OK
- #define SPS_WAITING 1
- #define SPS_PROCESSING 3
- #define SPS_BADLRC 4
- #define SPS_SUPPRESS_REPLY 5
- #define SPS_WRONGCMD 12
- #define SPS_WRONGPAR 13
- #define SPS_UEPSERR 14
- #define SPS_BAD_REQUEST 15
- #define SPS_NOCARD 21
- #define SPS_BADFRAME 23
- #define SPS_READFAIL 34
- #define SPS_CMFAIL 35
- #define SPS_PINERR 36
- #define SPS_MACERR 37
- #define SPS_INCOMPL 38
- #define SPS_NEED_EDITOR 96
- #define SPS_NEED_RESTART 98
- #define SPS_NOLINK 99
- #define ERR_NOT_SUPPORTED 101
- #define ERR_SUPPRESSED 112
- #define ERR_LIBRARY_LOAD 113
- #define ERR_FUNCTION_NOT_FOUND 114
- #define ERR_LIBRARY_BUSY 115
- #define ERR_CTX_GET 116
- #define ERR_CTX_SET 117
- #define ERR_MCL_ERROR 230
- #define ERR_ABORT_STUPID 238
- #define DRV_OK ERR_OK
- #define DRV_WRONGPAR 249
- #define DRV_CANCEL 250
- #define DRV_BADMEDIA 251
- #define DRV_NOTSUPP 252
- #define DRV_FAILURE 253
- #define DRV_TIMEOUT 254
- #define ERR_CANCEL 2000
- #define ERR_TIMEOUT 2002
- #define ERR_HOT_CLIENT 2004
- #define ERR_HOT_REGION 2005
- #define ERR_HOT_STAFF 2006
- #define ERR_HOT_BANK 2007
- #define ERR_BAD_OPER_MASK 2008
- #define ERR_BAD_HOTLIST 2020
- #define ERR_CANT_WRITE_HOTLIST 2021
- #define ERR_BLOCK_PIN1 2404
- #define ERR_BLOCK_PIN2 2405
- #define ERR_BLOCK_BOTH_PINS 2406
- #define ERR_WRONG_LOY_PWD 2601
- #define ERR_CLIENT_NOT_IN_LOY 2606

- #define ERR_LOW_FUNDS 3001
- #define ERR_BIG_TROUBLES 3002
- #define ERR_OPERDAY_NOT_OPEN 3019
- #define ERR_NO_ACCOUNT 3138
- #define ERR_NO_FUNDS_TO_LOAD 3168
- #define ERR_IV_FAIL 4046
- #define ERR_MK_LOAD_FAIL_0 4060
- #define ERR_MK_LOAD_FAIL_1 4061
- #define ERR_MK_LOAD_FAIL_2 4062
- #define ERR_MAC_FAIL 4063
- #define ERR_FINGER_NOT_FOUND 4070
- #define ERR_FINGER_FAILURE 4071
- #define ERR_BIO_FAILED_SESSION 4072
- #define ERR_BIO_NO_SCANNER 4073
- #define ERR_BIO_CANCEL 4074
- #define ERR_BIO_NOT_SUPPORTED 4075
- #define ERR_BIO_NO_FINGERS 4076
- #define ERR_BIOLINK_ERR 4077
- #define ERR_BAD_SCAN 4078
- #define ERR_BAD_MATCHER 4079
- #define ERR_NO_LINK 4100
- #define ERR_SBER_NO_HOTLIST 4101
- #define ERR_SBER_NO_PERCENT 4102
- #define ERR_EMV_BAD_AC2 4103
- #define ERR_WRONG_REPLY 4104
- #define ERR_PIN_ABSENT 4105
- #define ERR_PIN_WRONG 4106
- #define ERR_PIN_LOCKED 4107
- #define ERR_BAD_CHECK_DIGIT 4108
- #define ERR_MERCH_BATCH_FULL 4110
- #define ERR_MERCH_BATCH_EXPIRED 4111
- #define ERR_WRONG_HOTLIST 4112
- #define ERR_FIXED_LIMIT_EXCEEDED 4113
- #define ERR_ONLINE_FAILED 4114
- #define ERR_NO_MANUAL_ENTRY 4115
- #define ERR_WRONG_4DIGITS 4116
- #define ERR_PIN_REFUSED 4117
- #define ERR_UNABLE_FIND_REC 4118
- #define ERR_V2R_NOLINK 4119
- #define ERR_V2R_UNABLE_MAC 4120
- #define ERR_V2R_INT_ERR 4121
- #define ERR_V2R_KEY_ERROR 4122
- #define ERR_NO_SKEYS 4123
- #define ERR_NO_MKEYS 4124
- #define ERR_NEED_CHIP 4125
- #define ERR_BAD_TRX_LIST 4126
- #define ERR_NEED_SPLIT_TRX 4127
- #define ERR_BAD_CONFIG_PARAMS 4128
- #define ERR_NO_LOG_FILE 4129
- #define ERR_DISK_FULL 4130
- #define ERR_NEED_PARAMS 4131
- #define ERR_REFUSED_BY_CARD 4132
- #define ERR_INVALID_RC 4133
- #define ERR_TIME_FOR_TOTALS 4134
- #define ERR_BAD_DEPT_CFG 4135

- #define ERR_NEED_NEW_PINPAD 4136
- #define ERR_WRONG_NEW_PWD 4137
- #define ERR_CARDS_ARE_SAME 4138
- #define ERR_NO_SUITABLE_CONN 4139
- #define ERR_INVALID_TRX_DATA 4140
- #define ERR_SH_FILE_NOT_FOUND 4141
- #define ERR_ONLINE_CANC_FAILED 4142
- #define ERR_OLD_M_HOTLIST 4143
- #define ERR_BAD_M_HOTLIST_3 4144
- #define ERR_BAD_M_HOTLIST_2 4145
- #define ERR_BAD_M_HOTLIST_1 4146
- #define ERR_BAD_M_HOTLIST 4147
- #define ERR_CARD_IN_HL 4148
- #define ERR_NAME_NOT_FOUND 4149
- #define ERR_FLOOR_LIMIT_EXCEEDED 4150
- #define ERR_CARD_EXPIRED 4151
- #define ERR_NO_MPAD_HISTORY 4152
- #define ERR_BAD_MPAD_HISTORY_FMT 4153
- #define ERR_CHIP_REQUIRED 4154
- #define ERR_EMPTY_HISTORY 4155
- #define ERR_UNABLE_TO_CANCEL 4156
- #define ERR_OPER_LIMIT_EXCEEDED 4157
- #define ERR_PAYPASS_NO_PIN 4158
- #define ERR_MULTI_CUR_NOT_SUPP 4159
- #define ERR_APDU_ERR 4160
- #define ERR_NO_CERT_EMUL_FILE 4161
- #define ERR_RSA_PRIVATE_FAIL1 4162
- #define ERR_RSA_PRIVATE_FAIL2 4163
- #define ERR_BSC_BAD_SKEY 4164
- #define ERR_BIO_BAD_EA1 4165
- #define ERR_BIO_BAD_EA2 4166
- #define ERR_BIO_BAD_EA3 4167
- #define ERR_BIO_BAD_TEMPLATE1 4168
- #define ERR_BIO_BAD_TEMPLATE2 4169
- #define ERR_BIO_NO_IAPPD 4171
- #define ERR_BAD_OPER_TYPE 4172
- #define ERR_CLASS_NOT_SUPP 4173
- #define ERR_BMP_NOT_FOUND 4174
- #define ERR_BMP_TOO_BIG 4175
- #define ERR_VIVO_NO_VERINFO 4176
- #define ERR_NOT_OWN_CARD 4180
- #define ERR_OLD_PP_VERSION 4181
- #define ERR_CA_KEY_BAD 4182
- #define ERR_SPASIBO_1 4183
- #define ERR_SPASIBO_2 4184
- #define ERR_WRONG_ADMIN_CARD 4185
- #define ERR_KEY_ALREADY_PRESENT 4186
- #define ERR_WRONG_PAN 4187
- #define ERR_WRONG_VT 4188
- #define ERR_WRONG_VALUE 4189
- #define ERR_WRONG_DEPARTMENT 4191
- #define ERR_RDL_START_AGAIN 4200
- #define ERR_RDL_VERSION_OK 4201
- #define ERR_RDL_BAD_OFFSET 4202
- #define ERR_RDL_UNKNOWN_DEV 4203

- #define ERR_RDL_ERROR 4204
- #define ERR_RDL_BAD_HOST 4205
- #define ERR_RDL_BAD_CLIENT 4206
- #define ERR_RDL_BAD_MESSAGE 4207
- #define ERR_RDL_BAD_DATABASE 4208
- #define ERR_RDL_BAD_DATA 4209
- #define ERR_RDL_BAD_CRYPT 4210
- #define ERR_RDL_KEY_MISSING 4211
- #define ERR_RDL_BAD_PROC 4212
- #define ERR_RDL_NO_REGION 4220
- #define ERR_RDL_KKM_LINK_FAIL 4221
- #define ERR_PIL_PAR_MISSING 4300
- #define ERR_PIL_OT_INVALID 4301
- #define ERR_PIL_CT_INVALID 4302
- #define ERR_PIL_CT_NOT_SERVICED 4303
- #define ERR_PIL_WRONG_REPLY 4304
- #define ERR_INCORRECT_VERSION 4305
- #define ERR_NOT_INITIALISED 4306
- #define ERR_RESERVED 4307
- #define ERR_GENERAL 4308
- #define ERR_NO_DOCUMENT 4309
- #define ERR_PIL_BAD_TRACK2 4310
- #define ERR_PIL_NO_FILES 4311
- #define ERR_PIL_NO_MORE_TASKS 4312
- #define ERR_PIL_CARD_NOT_SAME 4313
- #define ERR_PIL_NEED_SBER 4314
- #define ERR_PIL_CARD_NOT_SAME_2 4315
- #define ERR_PIL_AUTOCANCEL 4316
- #define ERR_PIL_NOT_CLOSED 4317
- #define ERR_PIL_NOT_PROCESSED 4318
- #define ERR_PIL_AMOUNT_TOO_BIG 4319
- #define ERR_PIL_CARD_NOT_SAME_3 4320
- #define ERR_PIL_REMOVE_OLD_CARD 4321
- #define ERR_PIL_PRINT_ERROR 4322
- #define ERR_PIL_CARD_NOT_SAME_4 4323
- #define ERR_PIL_ENROLL_FAILED 4324
- #define ERR_PIL_NO_AMOUNT 4325
- #define ERR_PIL_BAD_TRACK1 4326
- #define ERR_NO_GOODS_FOR_DISPLAY 4327
- #define ERR_NO_GOODS_IN_ARGS 4328
- #define ERR_PIL_TOO_MANY_GOODS 4329
- #define ERR_NO_GOODS_FOUND 4330
- #define ERR_GENERAL1 4331
- #define ERR_PIL_INVALID_ARG 4332
- #define ERR_GENERAL3 4333
- #define ERR_READ_CARD 4334
- #define ERR_GENERAL5 4335
- #define ERR_INVALID_CURRENCY 4336
- #define ERR_GENERAL7 4337
- #define ERR_GENERAL8 4338
- #define ERR_GENERAL9 4339
- #define ERR_GENERAL_10 4340
- #define ERR_GENERAL_11 4341
- #define ERR_GENERAL_12 4342
- #define ERR_GENERAL_13 4343

- #define ERR_ADMIN_CARD 4350
- #define ERR_NO_TLV_FILES 4351
- #define ERR_CARD_READY 4352
- #define ERR_THIS_IS_SBER 4353
- #define ERR_SONDA_CHECK_FINGER 4354
- #define ERR_SONDA_FINGER_DUP 4355
- #define ERR_SONDA_USER_DUP 4356
- #define ERR_SONDA_INSUFF_Q 4357
- #define ERR_SONDA_FAILED 4358
- #define ERR_SONDA_UNKNOWN_ERR 4359
- #define ERR_SONDA_ENROLL_FAILED 4360
- #define ERR_SONDA_NOT_FOUND 4361
- #define ERR_PP_NOT_WORKING 4362
- #define ERR_AMOUNT_GT_ORIGINAL 4363
- #define ERR_SIGN_CAPTURE_NOT_SUPPORTED 4365
- #define ERR_TOO_LOW_DISCOUNT_VALUE 4366
- #define ERR_RKL_INVALID_REQ_FMT 4367
- #define ERR_RKL_NO_CA_KEY 4368
- #define ERR_RKL_NO_HOST_CERT 4369
- #define ERR_RKL_NO_CA_PK 4370
- #define ERR_RKL_KEYGEN_FAIL 4371
- #define ERR_RKL_B2B_FAIL 4372
- #define ERR_CASHINFO_NO_NUM_TRX 4375
- #define ERR_CASHINFO_NO_AMOUNT 4376
- #define ERR_CASHINFO_NO_SHIFT_ID 4377
- #define ERR_CASHINFO_BAD_AMOUNT 4378
- #define ERR_DAY_SHOULD_BE_CLOSED 4380
- #define ERR_QR_BAD_FORMAT 4381
- #define ERR_VOLUME_TOO_BIG 4382
- #define ERR_CANT_OPEN_SCANER 4383
- #define ERR_V2R_AUTH_DECLINED 4400 /* reserved up to 4699 */
- #define ERR_V2R_AUTH_CALL_ISS 4401
- #define ERR_VV_ERROR 4700

7.6.1 Подробное описание

7.6.2 Макросы

7.6.2.1 #define DRV_BADMEDIA 251

Внутренняя ошибка: Ошибка записи данных на диск

7.6.2.2 #define DRV_CANCEL 250

Внутренняя ошибка: Операция отменена

7.6.2.3 #define DRV_FAILURE 253

Внутренняя ошибка: Что-то пошло не так...

7.6.2.4 #define DRV_NOTSUPP 252

Внутренняя ошибка: Операция не поддерживается

7.6.2.5 #define DRV_OK ERR_OK

Успешно

7.6.2.6 #define DRV_TIMEOUT 254

Внутренняя ошибка: Истекло время ожидания

7.6.2.7 #define DRV_WRONGPAR 249

Внутренняя ошибка: На терминал передана команда не содержащая обязательные параметры

7.6.2.8 #define ERR_ABORT_STUPID 238

Отправка команды отменена. TODO: Check error text.

7.6.2.9 #define ERR_CANCEL 2000

Операция отменена клиентом или кассиром

7.6.2.10 #define ERR_CTX_GET 116

Ошибка чтения параметра

7.6.2.11 #define ERR_CTX_SET 117

Ошибка установки параметра

7.6.2.12 #define ERR_FUNCTION_NOT_FOUND 114

Указанная функция не найдена в динамической библиотеке

7.6.2.13 #define ERR_HOT_BANK 2007

Obsolete duet error

7.6.2.14 #define ERR_HOT_CLIENT 2004

Obsolete duet error

7.6.2.15 #define ERR_HOT_REGION 2005

Obsolete duet error

7.6.2.16 #define ERR_HOT_STAFF 2006

Obsolete duet error

7.6.2.17 #define ERR_LIBRARY_BUSY 115

Библиотека занята другим процессом, требуется подождать его завершения

7.6.2.18 #define ERR_LIBRARY_LOAD 113

Ошибка загрузки динамической библиотеки

7.6.2.19 #define ERR_MCL_ERROR 230

Ошибка чтения карты. TODO: Check error text.

7.6.2.20 #define ERR_NOT_SUPPORTED 101

Операция не поддерживается на текущем решении

7.6.2.21 #define ERR_OK 0

Успешно

7.6.2.22 #define ERR_SUPPRESSED 112

Требуется подавить ответ на комманду

7.6.2.23 #define ERR_TIMEOUT 2002

Истекло время ожидания

7.6.2.24 #define SPS_BAD_REQUEST 15

Задача с указанным номером не найдена в очереди задач

7.6.2.25 #define SPS_BADFRAME 23

На терминал передана команда не полная команда

7.6.2.26 #define SPS_BADLRC 4

На терминал передана команда с неверной контрольной суммой

7.6.2.27 #define SPS_CMFAIL 35

Ошибка чтения ключей пинпада

7.6.2.28 #define SPS_INCOMPL 38

Локальный протокол обмена запрашивает следующий блок данных

7.6.2.29 #define SPS_MACERR 37

Ошибка создания MAC-ключа

7.6.2.30 #define SPS_NEED_EDITOR 96

Для выполнения операции требуется запуск утилиты "Редактор Параметров"

7.6.2.31 #define SPS_NEED_RESTART 98

Для выполнения операции требуется перезапуск кассы

7.6.2.32 #define SPS_NOCARD 21

Ответ на команду CMD_CARD_TEST при отсутствии карты

7.6.2.33 #define SPS_NOLINK 99

Нет связи с пинпадом или терминалом

7.6.2.34 #define SPS_OK ERR_OK

Успешно

7.6.2.35 #define SPS_PINERR 36

Ошибка ввод пин-кода

7.6.2.36 #define SPS_PROCESSING 3

Задача в процессе выполнения

7.6.2.37 #define SPS_READFAIL 34

Ошибка чтения магнитной полосы карты

7.6.2.38 #define SPS_SUPPRESS_REPLY 5

Подождать с ответом. Скорее всего терминал выполняет перезагрузку

7.6.2.39 #define SPS_UEPSERR 14

Ответ терминала содержит код ошибки

7.6.2.40 #define SPS_WAITING 1

Задача поставлена в очередь

7.6.2.41 #define SPS_WRONGCMD 12

На терминал передана не корректная команда

7.6.2.42 #define SPS_WRONGPAR 13

На терминал передана команда не содержащая обязательные параметры

Глава 8

Структуры данных

8.1 Структура auth_answer

Поля данных

- int TType
- unsigned long Amount
- char RCode [3]
- char AMessage [16]
- int CType
- char * Check

8.1.1 Подробное описание

Основные параметры операции Структура, используемая для описания операции и получения результатов выполнения операции.

8.2 Структура auth_answer10

This structure blah blah blah...

Поля данных

- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- int SberOwnCard
- char Hash [CARD_HASH_LEN]

8.2.1 Подробное описание

This structure blah blah blah...

8.3 Структура auth_answer11

Расширение для получения кода авторизации успешной операции, номера карты, кода ответа, даты, времени и номера операции, хеша номера карты, признака принадлежности карты Сбербанку, третьей дорожки карты.

Поля данных

- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- int SberOwnCard
- char Hash [CARD_HASH_LEN]
- char Track3 [CARD_TRACK3_LEN]
- unsigned long RequestID

8.3.1 Подробное описание

Расширение для получения кода авторизации успешной операции, номера карты, кода ответа, даты, времени и номера операции, хеша номера карты, признака принадлежности карты Сбербанку, третьей дорожки карты.

8.4 Структура auth_answer12

Расширение card_authorize11 возможностью указать номер отдела и задать/получить номер ссылки.

Поля данных

- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- int SberOwnCard
- char Hash [CARD_HASH_LEN]
- char Track3 [CARD_TRACK3_LEN]
- unsigned long RequestID
- DWORD Department
- char RRN [RRN_LEN]

8.4.1 Подробное описание

Расширение card_authorize11 возможностью указать номер отдела и задать/получить номер ссылки.

8.5 Структура auth_answer13

Расширение card_authorize11 возможностью указать номер отдела, код вылюты и получить способ чтения карты, имя клиента карты и AID.

Поля данных

- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- int SberOwnCard
- char Hash [CARD_HASH_LEN]
- char Track3 [CARD_TRACK3_LEN]
- DWORD RequestID
- DWORD Department
- char RRN [RRN_LEN]
- DWORD CurrencyCode
- char CardEntryMode
- char CardName [MAX_CARD_NAME_LEN]
- char AID [MAX_AID_ASCII_LEN]
- char FullErrorText [MAX_FULL_ERROR_TEXT]

8.5.1 Подробное описание

Расширение card_authorize11 возможностью указать номер отдела, код вылюты и получить способ чтения карты, имя клиента карты и AID.

8.6 Структура auth_answer14

Расширение card_authorize13 возможностью указать информацию о товаре.

Поля данных

- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- int SberOwnCard
- char Hash [CARD_HASH_LEN]
- char Track3 [CARD_TRACK3_LEN]
- DWORD RequestID
- DWORD Department
- char RRN [RRN_LEN]
- DWORD CurrencyCode
- char CardEntryMode
- char CardName [MAX_CARD_NAME_LEN]
- char AID [MAX_AID_ASCII_LEN]
- char FullErrorText [MAX_FULL_ERROR_TEXT]
- DWORD GoodsPrice
- DWORD GoodsVolume
- char GoodsCode [MAX_GOODS_DATA]
- char GoodsName [MAX_GOODS_DATA]

8.6.1 Подробное описание

Расширение card_authorize13 возможностью указать информацию о товаре.

8.7 Структура auth_answer2

Расширение для получения кода авторизации успешной операции.

Поля данных

- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]

8.7.1 Подробное описание

Расширение для получения кода авторизации успешной операции.

8.8 Структура auth_answer3

Расширение для получения кода авторизации успешной операции и номера карты.

Поля данных

- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]

8.8.1 Подробное описание

Расширение для получения кода авторизации успешной операции и номера карты.

8.9 Структура auth_answer4

Расширение для получения кода авторизации успешной операции, номера карты, кода ответа, времени операции и номера операции на кассе.

Поля данных

- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber

8.9.1 Подробное описание

Расширение для получения кода авторизации успешной операции, номера карты, кода ответа, времени операции и номера операции на кассе.

8.10 Структура auth_answer5

Расширение для получения кода авторизации успешной операции и номера ссылки (RRN).

Поля данных

- struct auth_answer auth_answ
- char RRN [RRN_LEN]
- char AuthCode [AUTH_CODE_LEN]

8.10.1 Подробное описание

Расширение для получения кода авторизации успешной операции и номера ссылки (RRN).

8.11 Структура auth_answer6

Расширение для получения данных как при выполнении auth_answer5 и auth_answer4.

Поля данных

- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- char RRN [RRN_LEN]

8.11.1 Подробное описание

Расширение для получения данных как при выполнении auth_answer5 и auth_answer4.

8.12 Структура auth_answer7

Расширение для получения кода авторизации успешной операции, номера карты и признака принадлежности карты Сбербанку.

Поля данных

- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int SberOwnCard

8.12.1 Подробное описание

Расширение для получения кода авторизации успешной операции, номера карты и признака принадлежности карты Сбербанку.

8.13 Структура auth_answer8

Расширение для получения данных как при выполнении auth_answer5 и auth_answer4, а также номера карты и срока действия в шифрованном виде.

Поля данных

- struct auth_answer auth_answ
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int ErrorCode
- char TransDate [TRANSDATE_LEN]
- int TransNumber
- char RRN [RRN_LEN]
- char EncryptedData [MAX_ENCR_DATA *2+1]

8.13.1 Подробное описание

Расширение для получения данных как при выполнении auth_answer5 и auth_answer4, а также номера карты и срока действия в шифрованном виде.

8.14 Структура auth_answer9

Расширение для получения кода авторизации успешной операции, номера карты, хеша номера карты и признака принадлежности карты Сбербанку.

Поля данных

- auth_answer ans
- char AuthCode [AUTH_CODE_LEN]
- char CardID [CARD_ID_LEN]
- int SberOwnCard
- char Hash [CARD_HASH_LEN]

8.14.1 Подробное описание

Расширение для получения кода авторизации успешной операции, номера карты, хеша номера карты и признака принадлежности карты Сбербанку.

8.15 Структура payment_info_item

This structure blah blah blah...

Поля данных

- DWORD dwTag
- char Value [MAX_PAYMENT_ITEM]
- BYTE Flags
- void * pNextItem

8.15.1 Подробное описание

This structure blah blah blah...

8.16 Структура preauth_rec

Структура для описания одной операции в списке, по которой нужно выполнить завершение расчета.

Поля данных

- unsigned long Amount
- char RRN [RRN_LEN]
- char Last4Digits [5]
- unsigned short ErrorCode

8.16.1 Подробное описание

Структура для описания одной операции в списке, по которой нужно выполнить завершение расчета.

Предметный указатель